



**Integration of Neural Networks
and Probabilistic Spatial Models
for Acoustic Blind Source Separation**

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

M. Sc. Lukas Drude

Erster Gutachter: Prof. Dr.-Ing. Reinhold Häb-Umbach

Zweiter Gutachter: Prof. Dr.-Ing. Timo Gerkmann

Tag der mündlichen Prüfung: 18.12.2020

Paderborn 2020

Diss. EIM-E/358

Abstract

Despite a lot of progress in speech separation, enhancement, and automatic speech recognition realistic meeting recognition is still fairly unsolved. Most research on speech separation either focuses on spectral cues to address single-channel recordings or spatial cues to separate multi-channel recordings and exclusively either rely on neural networks or probabilistic graphical models. Integrating a spatial clustering approach and a deep learning approach using spectral cues in a single framework can significantly improve automatic speech recognition performance and improve generalizability given that a neural network profits from a vast amount of training data while the probabilistic counterpart adapts to the current scene. This thesis at hand, therefore, concentrates on the integration of two fairly disjoint research streams, namely single-channel deep learning-based source separation and multi-channel probabilistic model-based source separation. It provides a general framework to integrate spatial and spectral cues in which neural networks and probabilistic graphical models complement each other in achieving state of the art performance in blind source separation on noisy, reverberant data. The efficacy of the proposed approaches is evaluated on simulated artificial mixtures as well as real recordings of simultaneously active speakers. The key findings are (1) a cascade integration in which a neural network initializes a probabilistic graphical model provides substantial improvement, (2) spatial cues can be used for unsupervised training of neural networks, (3) tight integration, an integration in which a joint agreement between both modalities and models is found, leads to lowest word error rates and best generalizability to unseen real mixtures.

Acknowledgments

First and foremost I thank my supervisor Prof. Dr.-Ing. Reinhold Haeb-Umbach for granting me the academic freedom to explore topics of speech enhancement and recognition without restrictions in mind. Being able to freely decide led to a few dead ends but ultimately allowed me to explore the field of robust automatic speech recognition more holistically, resulted in many ideas outside of the beaten tracks, and a thesis topic worth working on. Many published scientific contributions not directly related to this thesis find their roots in this freedom and the encouragement to stay curious.

I thank Prof. Dr.-Ing. Timo Gerkmann for being part of the committee and contributing to the outcome of the thesis with just the right amount of constructive criticism.

I thank my colleagues for their collaboration, fruitful discussions, and uncountable coffee chats. A big thank-you belongs to Jahn Heymann with whom I shared almost the entire time span of my employment at the Paderborn University. Due to Jahn I have grown as a researcher, have become more self-critical, and have become more agile when adopting new technological advancements. Working with him led to many joint scientific contributions that go well beyond what one would have done alone. The joint participation in the CHiME challenges is one highlight in which mutual support resulted in great outcomes and a lot of fun. I thank Christoph Boeddeker who joined our research group later for the endless discussions which really let into the details of why things are as they are, ranging from topics such as LaTeX formatting to dereverberation.

I thank Prof. Bhiksha Raj for hosting me during my research exchange at the Language Technologies Institute (LTI) at Carnegie Mellon University (CMU), Pittsburgh, USA. I really enjoyed my stay with him who did not stop surprising me with unconventional ideas and loopholes in my work on complex backpropagation. I enjoyed his way of teaching with much more focus on the intuition behind the methods which served as a complementary view to the rigid proof-oriented way I was more used to.

I would like to cordially thank Tomohiro Nakatani, Shoko Araki, Keisuke Kinoshita, Marc Delcroix, Nabutako Ito, and Takuya Higuchi from NTT Communication Science Laboratories, Kyoto, Japan for their great support leading to and during my internship in their department. They made me feel very welcome in an entirely new research environment. To me, they are one of the most important research groups in our field and I enjoyed their great focus on scientific details although researching in a corporate setup.

I am grateful for the additional funding by a Google Research Grant and two consecutive research projects with NTT Communication Science Laboratories, Kyoto, Japan. Computational resources were provided by the Paderborn Center for Parallel Computing.

I thank my family for their help and a place to come back to when I needed to forget about my thesis. Finally, I thank my wonderful wife for supporting me during my time at Paderborn University and beyond. I could not have done it without you.

Contents

Abstract	i
Acknowledgments	ii
1 Introduction	1
2 Prerequisites	3
2.1 Notation	3
2.2 Signal model	4
2.3 Overview table of variable names	5
2.4 Random variables	5
2.5 Latent variable models and the expectation maximization algorithm	7
2.5.1 Latent variable models	7
2.5.2 Mixture models	7
2.5.3 Expectation maximization algorithm	8
3 Blind source separation principles	19
3.1 Principles of single-channel approaches	20
3.1.1 Shallow methods	20
3.1.2 Deep-learning methods	21
3.1.2.1 DC: Deep clustering	22
3.1.2.2 DAN: Deep attractor network	23
3.1.2.3 PIT: Permutation invariant training	25
3.1.3 Discussion of single-channel deep-learning methods	25
3.2 Principles of multi-channel approaches	26
3.2.1 Probabilistic spatial mixture models	28
3.2.1.1 Frequency permutation problem	29
3.2.1.2 Initialization	29
3.2.1.3 Influence of the mixture weight	30
3.2.1.4 Complex Watson mixture model	30
3.2.1.5 Complex Bingham mixture model	33
3.2.1.6 Full-Bayesian complex Watson mixture model	34
3.2.1.7 Time-variant complex Gaussian mixture model	35
3.2.1.8 Complex angular central Gaussian mixture model	36
3.2.1.9 Guided source separation	38
3.2.2 Spatial features for neural networks	39
3.3 Principles of source extraction	39
3.3.1 Spectral subtraction/ masking	39

3.3.2	Spatial filtering/ beamforming	40
3.3.2.1	Spatial covariance matrix estimation	41
3.3.2.2	MaxSNR/GEV	42
3.3.2.3	MVDR	43
3.3.2.4	Linearly constrained minimum variance beamformer	44
3.3.2.5	Weighted multi-channel Wiener filter	45
3.3.2.6	Magnitude and phase normalization of beamforming vectors	46
3.3.3	Combination of beamforming and masking	46
4	Integration of neural networks and probabilistic graphical models	47
4.1	Existing integration approaches	48
4.2	Cascade approach: Integration by initialization	50
4.3	Tight integration of spatial and spectral features	51
4.3.1	vMFcACGMM	53
4.3.2	Additional constraints	55
4.4	Unsupervised training using multi-channel features	56
5	Evaluation	58
5.1	Performance metrics	58
5.2	Database design	59
5.2.1	WSJ0-2mix	59
5.2.2	WSJ-BSS	60
5.2.3	WSJ-MC	63
5.3	Acoustic model training	63
5.4	Deep-learning methods	65
5.4.1	Deep clustering	65
5.4.2	Deep attractor network	70
5.4.3	Permutation invariant training	73
5.4.4	Comparison with reference publications on WSJ0-2mix	75
5.5	Probabilistic spatial mixture models	76
5.5.1	Type of the spatial observation model	76
5.5.2	Parameter choice for the cACGMM	78
5.6	Source extraction	81
5.7	Integration of neural networks and probabilistic graphical models	85
5.7.1	Weak integration: A cascade approach	86
5.7.2	Strong integration	87
5.7.3	Comparison of integration models with single-/ multi-channel encoder	89
5.8	Unsupervised training of deep clustering	92
5.9	Overview of all methods on WSJ-BSS	94
5.9.1	Analysis of splits of the WSJ-BSS database	96
5.9.2	Analysis with matched training of the acoustic model	100
5.10	Overview of all methods on WSJ-MC	102
5.11	Reproducibility and statistical significance	104
6	Conclusion	111

A	Appendix	113
A.1	Properties of the complex Bingham distribution	113
A.1.1	Eigenvalue shift in the normalization term	113
A.1.2	Eigenvalue shift in the distribution	113
A.2	Non-negativity of the Kullback-Leibler divergence	113
A.3	Mixture weights without Lagrange’s method	114
A.4	Remarks on complex derivatives	115
A.5	GEV/MaxSNR beamformer	116
A.5.1	Solution with constraint optimization	117
A.5.2	Solution without constraint optimization	117
A.6	MVDR beamformer	118
A.7	Permutation formalism	118
A.8	Comparison of WSJ-BSS and SMS-WSJ	119
A.9	More detailed evaluation results	119
	Glossary	123
	List of peer-reviewed publications with own contributions (OC)	124
	Bibliography	127

1 Introduction

Blind source separation addresses the problem to separate signal components originating from different sources, while only the mixture single can be observed. In the audio domain, when multiple speakers are active simultaneously, humans are able to concentrate fairly well on a particular speaker and get the idea of what is being said. The problem of separating overlapping speech was coined *cocktail party problem* most likely in 1953 by Colin Cherry. Since then, many researchers have addressed simplifications of this problem. Early work concentrated on instantaneous mixtures and later got extended to cover convolutive mixtures, i.e., acoustic conditions in which a room impulse response due to the multi-path transmission in an acoustic enclosure causes a temporal smearing effect of the source signals. While blind source separation (BSS) systems were analyzed on their own for most of the time, more recently – mainly due to improved performance of separation methods and improved robustness of acoustic models – researchers started addressing the more challenging problem of multi-speaker automatic speech recognition (ASR). In fact, BSS for human listeners poses its own challenges such as the demand for low latency, avoidance of audible artifacts, and naturalness of the separation result.

The goal of this thesis is to propose and describe new methods to separate speech sources and automatically transcribe each utterance present in a mixture. Although there are quite many attempts at improving recognition of overlapped speech, the focus of this work is on two distinct aspects: (1) the integration of probabilistic graphical models and deep neural networks, and (2) the integration of spatial and spectral cues. The key motivating factors why integration along both aspects is promising are summarized in Figure 1.1.

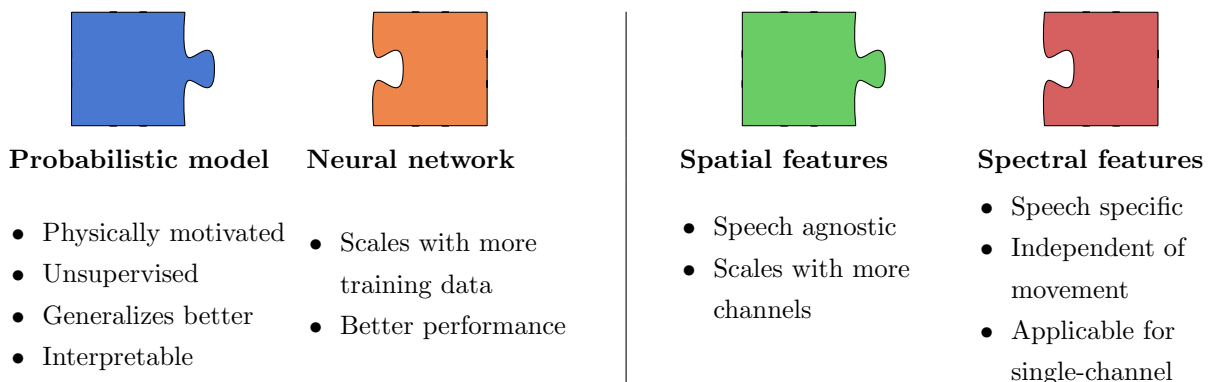


Figure 1.1: Key motivating pieces which illustrate why an integration framework is promising.

Probabilistic graphical models and neural networks have very complementary strengths. While deep neural networks (DNNs) are purely data-driven approaches and therefore contain very little priors introduced by a possibly error-prone human being, probabilistic models allow capturing a physical understanding of the world. Depending on the model choice, they may encapsulate our human understanding of the physics of wave propagation while a neural network has to guess all relevant statistical dependencies from data only. Interestingly, the data-driven approach has beaten statistical models by a great margin in very many domains. Nevertheless, their generalizability is often limited and performance on an unseen database can hardly be predicted. In contrast, while the base performance of unsupervised probabilistic graphical models may be inferior, they shine when it comes to new databases since they are just as unaware of that one as they were on the primary database of interest. The first integration aspect worth to analyze and capitalize upon is the complementarity of neural networks and probabilistic graphical models.

Addressing the second integration aspect, spatial features have long been the feature of choice in a multi-channel setup: Phase- and level differences between microphones can be very informative and lead to high separability given that the geometry is fairly static and the transfer characteristics of each source are sufficiently different. On the contrary, spectral features are either derived from or learned based on individual speakers' speech. The performance heavily depends on the discriminability of the speakers' voices and movement within limits does not impact performance at all.

Given these two dimensions (1) modeling paradigm and (2) feature modality, it is almost self-explanatory that an integration framework is to be sought after and can provide gains in terms of generalizability and overall performance.

To reach this goal and lead through fundamentals towards an integrated solution, the thesis is structured as follows: Chapter 2 introduces some fundamental concepts and sets up the notation and terminology used throughout this work. Chapter 3 provides a broader overview of source separation approaches and then quickly focuses on three deep learning-based separation concepts and a limited number of spatial clustering models, which all serve as a baseline as well as potential candidates to be used within an integration framework. Chapter 3 finishes with a review of source extraction methods, namely masking and beamforming. While a short overview of other integration approaches is compiled at the beginning of Chapter 4, it also develops the key aspects of this thesis, namely the cascade integration and the tight integration for blind source separation. Chapter 4 also details how a neural network-based source separation system can be trained without supervision. Chapter 5 contains an extensive evaluation not just of the proposed framework but also of the underlying integration components and the baseline systems – to prove that the proposed framework actually is an advancement, it is particularly important to demonstrate that the baseline systems are carefully tuned. Each part of Chapter 5 contains a brief exposition of the key findings to more easily capture the essence of the evaluation. Finally, conclusions and remarks on future directions of study are located in Chapter 6.

2 Prerequisites

This chapter introduces the notation and signal model in Section 2.1 and Section 2.2 and continues with an overview in Section 2.3. Most importantly, it introduces random variables and probabilistic graphical models in Section 2.4 and Section 2.5, respectively. Most importantly, Section 2.5 contains estimation techniques which are used in most systems proposed within this work.

2.1 Notation

- Scalars, vectors and matrices are distinguished by using small characters, bold characters and bold capital characters, e.g., x , \mathbf{x} and \mathbf{X} , respectively. More abstract sets of values or variables, without necessarily specifying the shapes of the set elements are denoted by calligraphic symbols such as \mathcal{V} .
- Whenever it becomes necessary to distinguish random variables from their realization a breve symbol is used, e.g., \check{x} , $\check{\mathbf{x}}$, $\check{\mathbf{X}}$, $\check{\mathcal{X}}$. However, this is avoided in the following by using the shorthand notation $p(x)$ instead of $p_{\check{x}}(\check{x} = x)$ if possible.
- To more quickly identify corresponding indices and boundaries, the indices are denoted by lower case characters and are upper-bound by the corresponding upper case character, e.g., $t \in \{0, \dots, T - 1\}$ or $t \in \{1, \dots, T\}$.
- Within the scope of this thesis probability density functions for continuous and discrete random variables as well as probability mass functions for discrete random variables share the same notation, e.g., $p(x)$. The most important reasons are: (a) The probability density function and the probability mass function for a discrete random variable are both defined with the same set of parameters – the probability of each class. Thus, they are used synonymously anyway. (b) Joint distributions of discrete and continuous random variables are not an edge case anymore, they can simply be written as $p(x, y)$.
- As common in an engineering context, this thesis does not distinguish between, e.g., a function f and the value $f(x)$ obtained by calling the function on some input x .
- The expected value of a random variable is denoted as $\mathbb{E}\{\cdot\}$. In case the distribution under which the expected value is calculated is unclear, this is denoted with the distribution/ probability density function (PDF) as a subscript: $\mathbb{E}_{q(\cdot)}\{\cdot\}$.
- Limits for sum or product symbols are simplified to just the index in case the sum or product is carried out over all possible values.

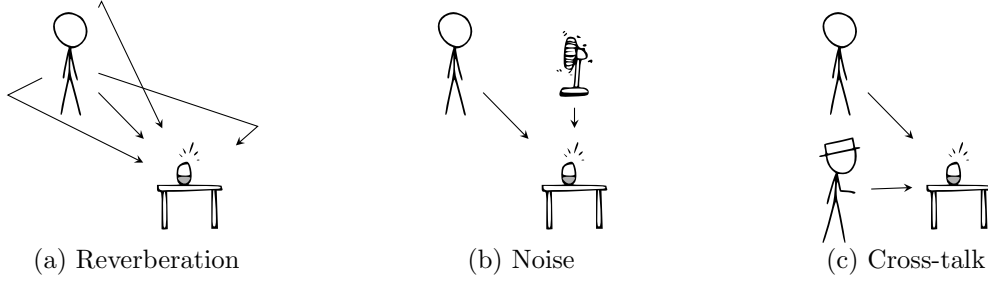


Figure 2.1: Illustration of typical signal quality impairments. In a real-world setting it is practically impossible to obtain reverberation-free (dry) recordings or entirely noise-free (clean) recordings. Some image elements are created and their reuse is permitted by Randall Munroe (xkcd.com).

2.2 Signal model

The typical obstacles for far-field ASR are first and foremost reverberation [1], noise sources [1], and interfering speakers. Although this thesis focuses on source separation, correctly handling reverberation and background noise is crucial when assessing the real-world applicability of the proposed algorithms. Figure 2.1 illustrates these impairments.

The signal propagation process, namely the reverberation of the source signals is caused by an infinite amount of reflections and potentially the direct transmission path. This process can be modeled as a convolution of a room impulse response (RIR) with the source signal in time domain [2, Equation 3]. If we further suppose that the dominant part of the RIR fits approximately into one analysis window, we can conveniently model the entire reverberation process as a multiplication in the short time Fourier transform (STFT) domain [3, Page 8 ff.]. This simplification is often called narrowband approximation [2, Section II.B]. Since the wave equation can be considered linear for room acoustics [2, Section II.A], we can deduct that the whole mixing process is sufficiently well modeled by a sum of all source images and the noise received at each of the D microphones:

$$\mathbf{y}_{t,f} = \sum_k \mathbf{h}_{k,f} s_{k,t,f} + \mathbf{n}_{t,f} = \sum_k \mathbf{x}_{k,t,f} + \mathbf{n}_{t,f}. \quad (2.1)$$

Here, $s_{k,t,f}$ represents the source signal of each of the $k \in \{1, \dots, K\}$ speakers, $\mathbf{h}_{k,f}$ represents the vector of acoustic transfer functions (ATFs) for each speaker [2, Section II.B], and $\mathbf{n}_{t,f}$ is the noise vector summarizing each noise signal at each of the microphones. Further, the indices $t \in \{1, \dots, T\}$ and $f \in \{1, \dots, F\}$ specify the time frame and frequency bin, respectively. The speech image, which is the reverberant version of the source signal as it is received by each microphone is denoted by $\mathbf{x}_{k,t,f}$ while the mixed signal is written as $\mathbf{y}_{t,f}$. Correspondingly, each of the vectors contains a complex-valued scalar for each of the D microphone channels (see Table 2.1 for an overview). It is worth noting that inter-frame and inter-band convolution effects are neglected [4] when the mixing process is modeled in the STFT domain as in Equation 2.1. In the context of this thesis, no distinction between the noise and the noise image is made and both terms are used synonymously. Depending on the application it is now desired to either find an estimate of the speech image $\mathbf{x}_{k,t,f}$ or the underlying source signal $s_{k,t,f}$. The signals are later reconstructed using the corresponding inverse transform either for a human listener or a subsequent speech recognizer.

2.3 Overview table of variable names

Table 2.1: Overview table of most frequently used variables. Variables with a more limited scope are only introduced in the corresponding chapters and not summarized here.

Variable	Values	Description
l	$\{1, \dots, L\}$	Sample index in the discrete time domain
d	$\{1, \dots, D\}$	Channel (sensor) index
t	$\{1, \dots, T\}$	Time frame index in the STFT domain
f	$\{1, \dots, F\}$	Frequency bin index in the STFT domain
k	$\{1, \dots, K\}$	Class index (e.g., for mixture models) or speaker index
i	$\{1, \dots, I\}$	Iteration index
e	$\{1, \dots, E\}$	Embedding dimension index
$s_{t,f}$	\mathbb{C}	Speech source signal in the STFT domain
$h_{k,d,t,f}$	\mathbb{C}	ATF in the STFT domain
$\mathbf{h}_{k,t,f}$	\mathbb{C}^D	Vector of ATFs in the STFT domain
$x_{k,d,t,f}$	\mathbb{C}	Speech image in the STFT domain
$\mathbf{x}_{k,t,f}$	\mathbb{C}^D	Vector of speech image channels in the STFT domain
$n_{d,t,f}$	\mathbb{C}	Noise image in the STFT domain
$\mathbf{n}_{t,f}$	\mathbb{C}^D	Vector of noise image channels in the STFT domain
$y_{t,f,d}$	\mathbb{C}	Mixture in the STFT domain
$\mathbf{y}_{t,f}$	\mathbb{C}^D	Vector of mixture channels in the STFT domain
$\tilde{\mathbf{y}}_{t,f}$	$\mathbb{C}S^{D-1}$	Unit-length normalized vector of mixture channels
$\mathbf{e}_{t,f}$	\mathbb{R}^E	Embedding vector (e.g., in a DC system)
\mathbf{w}_f	\mathbb{C}^D	Beamforming vector in the STFT domain
$\hat{x}_{k,t,f}$	\mathbb{C}	Predicted speech signal
$c_{k,t,f}$	$\{0, 1\}$	Indicator variable is 1 if slot (t, f) is dominated by class k .
$\mathbf{c}_{t,f}$		One-hot vector indicating which class dominates the observation.
$\gamma_{k,t,f}$	$[0, 1]$	Posterior affiliation/ estimated mask

2.4 Random variables

Although basic knowledge about random variables might be common sense for the like-minded, it is very worth to point out different notational variations which often turn out to be the main nuisance factor in teaching.

Let us first address discrete random variables by acknowledging that the following statements all share the same information:

- The scalar random variable \check{c} can have the realization k . The probability of this event is given by $p_{\check{c}}(\check{c} = k) = p(k)$. Due to the second Kolmogorov axiom, all probabilities have to sum up to one, i.e., the probability that one element of the sample space Ω occurs is one:

$$p(\Omega) = \sum_k p(k) = 1.$$

- The probability of the one-hot random vector $\check{\mathbf{c}}$ having the realization \mathbf{c} is denoted by $p_{\check{\mathbf{c}}}(\check{\mathbf{c}} = \mathbf{c}) = p(\mathbf{c})$. The one-hot vector is then defined such that the k -th entry is 1, when $c = k$, otherwise 0. Again, the second Kolmogorov axiom has to hold:

$$p(\Omega) = \sum_{\mathbf{c}} p(\mathbf{c}) = 1.$$

- We can describe $p(k)$ as a categorical distribution¹ [5, Page 35] parameterized by the parameter vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top$:

$$p(k) = p(\check{k} = k) = p(\check{c}_k = 1) = p(\check{\mathbf{c}} = \mathbf{c}) = \text{Cat}(\check{k} = k; \boldsymbol{\pi}), \quad \text{with} \quad \pi_k = p(k).$$

- We may also interpret $p(k)$ as a probability density function. As long as the concept of $p(k)$ just encodes how likely certain outcomes are, it is not further necessary to distinguish between a probability mass function and a probability density function written with Dirac pulses:

$$p(k) = \pi_1 \delta(k - 1) + \dots + \pi_K \delta(k - K) = \sum_{k'} \pi_{k'} \delta(k - k').$$

By not distinguishing between the notation of probability mass functions and probability density functions, we avoid a common notational issue with respect to mixed distributions. The joint distribution of a continuous and a discrete random variable can simply be written as follows:

$$p(x, k) = p_{\check{x}, \check{\mathbf{c}}}(\check{x} = x, \check{c}_k = 1).$$

By extension of the notation for expected values above, we here write entropy, cross-entropy or Kullback-Leibler divergence in terms of the probability density functions they evaluate much rather than the random variable:

$$\begin{aligned} H(p(\check{x})) &= -\mathbb{E}_{p(x)} \{\ln p(\check{x})\} = -\int_{-\infty}^{\infty} p(x) \ln p(x) \, dx, \\ \text{CE}(p(\check{x}), q(\check{x})) &= -\mathbb{E}_{p(x)} \{\ln q(\check{x})\} = -\int_{-\infty}^{\infty} p(x) \ln q(x) \, dx, \\ \text{KL}(p(\check{x}) \| q(\check{x})) &= \text{CE}(p(\check{x}), q(\check{x})) - H(p(\check{x})) = \mathbb{E}_{p(x)} \left\{ \ln \frac{p(\check{x})}{q(\check{x})} \right\} = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \, dx. \end{aligned}$$

¹ The term *Multinoulli distribution*, an alternative name for the categorical distribution, apparently stems from Gustavo Lacerda [5, Page 35].

2.5 Latent variable models and the expectation maximization algorithm

Since latent variable models play an integral role throughout this thesis this section introduces latent variable models, mixture models, and methods to obtain corresponding parameters.

2.5.1 Latent variable models

Probabilistic models $p(\mathcal{Y}, \boldsymbol{\theta})$ ideally represent the distribution of the observation \mathcal{Y} as accurately as possible with a parameterization captured in $\boldsymbol{\theta}$. Often times, when the observations share a common cause or are a manifestation of an underlying process it is advisable to design a model with latent (or hidden) random variables [5, Page 337]. These latent variables \mathcal{Z} then influence the distribution of the observations, often introduce some hierarchy of the random variables, and may serve as an information bottleneck [5]. In particular, in unsupervised learning, this information bottleneck is of main interest as a condensed representation of the observation itself [5] and may then be used in a downstream task.

Most latent variable models exhibit a parameter identifiability problem [5, Section 11.3.1], [6, Section 11.2]. Reciting [6, Definition 11.2.1] a parameter $\boldsymbol{\theta}$ for a family of distributions $\{p(x|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathbb{D}_{\boldsymbol{\theta}}\}$ is *identifiable* if distinct values of $\boldsymbol{\theta}$ correspond to distinct distributions. In the case of discrete latent random variables, this may also be called label switching problem [5, Page 341] or permutation problem. This problem can also be understood as multimodality, in the sense of multiple peaks, of the likelihood in the parameter space: different locations in the parameter space lead to equal likelihood values. How this manifests in BSS applications is addressed in more detail in Section 3.2.1.1.

2.5.2 Mixture models

A particular instance of latent variable models are mixture models. In a mixture model one assumes the following generative process:

1. A class affiliation is sampled from, e.g., a categorical distribution $p(\mathbf{c}_n) = \text{Cat}(\mathbf{c}_n; \boldsymbol{\pi})$, where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top$ summarizes all class probabilities.
2. The observation itself is sampled from a class-conditional distribution. The class-conditional distribution is also termed class-dependent observation model.

The marginal distribution is then a weighted sum of class-conditional distributions:

$$p(\mathbf{y}_n; \boldsymbol{\theta}) = \sum_k \pi_k p(\mathbf{y}_n | c_{k,n}=1) = \sum_k \pi_k p(\mathbf{y}_n; \boldsymbol{\theta}_k), \quad (2.2)$$

where k is a class index and $\boldsymbol{\theta}_k$ contains the class-dependent parameters of the class-conditional distribution $p(\mathbf{y}_n | \boldsymbol{\theta}_k)$. The mixture weights $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top$ with $0 \leq \pi_k \leq 1$ sum up to one such that $p(\mathbf{c}_n) = \text{Cat}(\mathbf{c}_n; \boldsymbol{\pi})$ is a valid probability mass distribution.

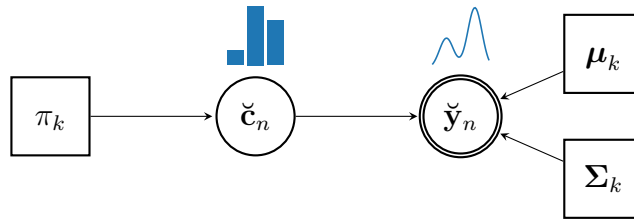


Figure 2.2: Graphical model of a Gaussian mixture model (GMM) as an example of a mixture model. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

In case the observation model is a Gaussian distribution, this model is called a GMM as illustrated in Figure 2.2:

$$p(\mathbf{y}_n; \boldsymbol{\theta}) = \sum_k \pi_k \mathcal{N}(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_k \pi_k \frac{1}{\sqrt{\det(2\pi \boldsymbol{\Sigma}_k)}} e^{-\frac{1}{2}(\mathbf{y}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_n - \boldsymbol{\mu}_k)}, \quad (2.3)$$

where π_k , $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are the class-dependent mixture weight, mean vector, and covariance matrix, respectively. For compactness reasons, all parameters are captured in

$$\boldsymbol{\theta} = \{\boldsymbol{\theta}_k | k \in \{1, \dots, K\}\} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | k \in \{1, \dots, K\}\}. \quad (2.4)$$

It was already in 1894 that Karl Pearson proposed methods to identify parameters of a GMM. At that time, due to a lack of today's nomenclature, he called this process *dissection of abnormal frequency curves into normal curves* [7]. The next section introduces the most common approaches used nowadays to estimate parameters of a mixture model, e.g., a GMM.

2.5.3 Expectation maximization algorithm

The expectation maximization (EM) algorithm is quite a famous² method to obtain maximum likelihood estimates of a latent variable model when an explicit estimation formula cannot be derived [8]. It is by no means the only way to approach this problem, e.g., Everitt mentions and compares a whole list of algorithms just to estimate parameters of a GMM [9]. The EM algorithm can be seen as a special case of the variational expectation maximization (VEM) algorithm as well as a special case of the majorize-minimization or minorize-maximization (MM) algorithm [10, Section 4.2].

Different authors tend to prefer different ways to motivate the EM algorithm. This work summarizes a selection of approaches: (a) A direct derivation which does not rely on an external formalism but rather heuristically defines cutting points to create a two-step iterative algorithm. (b) One version which motivates the auxiliary function by arguing that the marginal likelihood is just intractable. This version still leaves it rather unclear where the auxiliary function stems from and why optimizing it is indeed maximizing the likelihood we

² It is without doubt famous, since the work [8] alone has already received more than 50 000 citations.

were originally interested in. (c) A derivation which starts by decomposing the marginal likelihood and therefore is the most rigid approach arguing from first principles only. (d) An approach which shows that the EM is a special case of the VEM. (e) An approach motivating the EM as a special case of a MM algorithm.

a) Direct derivation The likelihood of a latent variable model which is compactly parameterized with $\boldsymbol{\theta}$ can be written as follows:

$$L = p(\mathcal{Y}; \boldsymbol{\theta}) = \int_{\mathcal{Z}} p(\mathcal{Y}|\mathcal{Z}; \boldsymbol{\theta})p(\mathcal{Z}; \boldsymbol{\theta})d\mathcal{Z}, \quad (2.5)$$

where \mathcal{Y} represents all realizations of observable random variables and \mathcal{Z} captures all realizations of hidden or latent random variables.³ Likewise, we may prefer to work with the log-likelihood function which often turns out slightly easier to handle and implement:

$$\ell = \ln p(\mathcal{Y}; \boldsymbol{\theta}) = \ln \int_{\mathcal{Z}} p(\mathcal{Y}|\mathcal{Z}; \boldsymbol{\theta})p(\mathcal{Z}; \boldsymbol{\theta})d\mathcal{Z}. \quad (2.6)$$

To find a maximum of the likelihood function, the necessary condition $\partial\ell/\partial\boldsymbol{\theta} \stackrel{!}{=} \mathbf{0}$ leads to a system of equations which needs to hold for the likelihood to be maximized. Therefore, we may simply calculate all derivatives first. Then, if we do not obtain explicit parameter estimation formulas directly, we may identify cutting points to partition the equations into a multi-step iterative algorithm. A similar argumentation can be found in [11, Section 9.2.2] and [12, Page 104].

Here, we use a GMM as an illustrative example because we may later reuse some results when motivating the mixture models applied to speech data. The log-likelihood of a GMM for the observations \mathbf{y}_n with the observation index $n = 1, \dots, N$ is denoted by

$$\ell = \sum_n \ln \sum_k \pi_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{with} \quad \sum_k \pi_k = 1. \quad (2.7)$$

To first find an equation for the mixture weights π_k , we modify the objective function ℓ by introducing the sum-1 condition according to the Lagrange method [13]:⁴

$$\ell' = \sum_n \ln \sum_k \pi_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \lambda \left(\sum_k \pi_k - 1 \right). \quad (2.8)$$

³ One may be willing to write $p_{\tilde{\mathcal{Y}}}(\tilde{\mathcal{Y}} = \mathcal{Y}; \boldsymbol{\theta})$ but we here use the simplified notation $p(\mathcal{Y}; \boldsymbol{\theta})$.

⁴ An alternative would have been to parameterize the categorical distribution with $\alpha_k / \sum_{k'} \alpha_{k'}$, where the parameter α_k is then unconstrained. See Section A.3 for a brief derivation.

We obtain the necessary optimality conditions by setting the derivatives to zero:

$$\begin{aligned}
\frac{\partial \ell'}{\partial \pi_k} &= \sum_n \frac{p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} p(\mathbf{y}_n; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} + \lambda \stackrel{!}{=} \mathbf{0} \quad \Bigg| \cdot \pi_k \\
\Leftrightarrow \quad &\sum_n \underbrace{\frac{\pi_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} p(\mathbf{y}_n; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}}_{=:\gamma_{k,n}} + \lambda \pi_k = \mathbf{0} \\
\Leftrightarrow \quad &\sum_n \gamma_{k,n} + \lambda \pi_k = \mathbf{0}
\end{aligned} \tag{2.9}$$

When we now sum up Equation 2.9 for each k , we obtain an expression for λ :

$$\sum_{k,n} \gamma_{k,n} + \lambda \underbrace{\sum_k \pi_k}_{=1} = 0 \Leftrightarrow N + \lambda = 0 \Leftrightarrow \lambda = -N. \tag{2.10}$$

With this result we can now separate π_k in Equation 2.9:

$$\pi_k = -\frac{1}{\lambda} \sum_n \gamma_{k,n} = \frac{1}{N} \sum_n \gamma_{k,n}. \tag{2.11}$$

With a similar procedure we find two more necessary conditions which have to hold for a likelihood maximum [11, Equation 9.17 and 9.19] given by

$$\boldsymbol{\mu}_k = \sum_n \gamma_{k,n} \mathbf{y}_n / \sum_n \gamma_{k,n}, \tag{2.12}$$

$$\boldsymbol{\Sigma}_k = \sum_n \gamma_{k,n} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^\top / \sum_n \gamma_{k,n}. \tag{2.13}$$

We may now identify an iterative two-step algorithm by first evaluating $\gamma_{k,n}$ and then updating the parameters as in Equations 2.11 – 2.13. However, this does neither tell us whether this is a maximum or a minimum nor are there any convergence guarantees visible. Evaluating second-order derivatives allows us to examine if the solution indeed corresponds to a maximum.

It is worth noting that the direct derivations of the update equations up to this point did not require explicit handling of latent random variables.

b) Auxiliary function An alternative to the direct approach is given as follows. First, we state the log-likelihood ℓ of the observations. Then, we introduce the latent random variables as a reverse marginalization [11, Equation 9.29] resulting in

$$\ell = \ln p(\mathcal{Y}; \boldsymbol{\theta}) = \ln \left(\int_{\mathcal{Z}} p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d\mathcal{Z} \right). \tag{2.14}$$

One may now argue that the integral in the logarithm (or the sum in case of discrete latent random variables) is overcomplicating the matter and one rather wishes to optimize the likelihood of the complete data $(\mathcal{Y}, \mathcal{Z})$ – a tuple of the observations and realizations of latent random variables. The alternative maximum likelihood term can then be written in the form

$$\ell' = \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}). \quad (2.15)$$

However, we neither know the realizations of the latent random variables nor do we know their distributions. At most, we may have a guess for the parameters $\boldsymbol{\theta}^{\text{old}}$ from a previous step or from an initialization. Therefore, we can make use of (a possibly wrong or premature) posterior distribution $p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})$ and calculate the expected value of the log-likelihood of the complete data under the assumption that $p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})$ is true, which is called a rather *heuristic idea* in [8, Page 6]. Nevertheless, a presentation like this is fairly common and also appears in Moon's comparably popular tutorial [14] as well as McLachlan's book on EM algorithms [15, Page 19]. The auxiliary function is then given by

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \{ \ell' \} \\ &= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathcal{Y}, \check{\mathcal{Z}}; \boldsymbol{\theta}) \right\} \\ &= \int_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}}) \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d\mathcal{Z}. \end{aligned} \quad (2.16)$$

We may now read this integral as a more structured way of dissecting the algorithm:

- In the expectation step (E-step) we evaluate $p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})$ – the distribution under which the expectation is calculated. This step is called expectation step because in principle we evaluate the expectation operator and then have a single expression for the auxiliary function without the expectation operator.
- In the maximization step, we maximize the resulting expression for the auxiliary function with respect to the new parameters $\boldsymbol{\theta}$ by calculating the derivative of the auxiliary function with respect to the parameters and equating this to zero.

For our concrete GMM example the posterior distribution of the latent random variables then turns out to be

$$p(c_{k,n} = 1 | \mathbf{y}_n; \boldsymbol{\theta}^{\text{old}}) = \gamma_{k,n} = \frac{\pi_k^{\text{old}} p(\mathbf{y}_n; \boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}})}{\sum_{k'} \pi_{k'}^{\text{old}} p(\mathbf{y}_n; \boldsymbol{\mu}_{k'}^{\text{old}}, \boldsymbol{\Sigma}_{k'}^{\text{old}})}. \quad (2.17)$$

First, we recognize that \mathbf{y}_n only depends on \mathbf{c}_n and neither on other observations $\mathbf{y}_{n'}$ nor other latent affiliations $\mathbf{c}_{n'}$, where $n' \neq n$:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathcal{Y}, \check{\mathcal{Z}}; \boldsymbol{\theta}) \right\} \\ &= \sum_n \mathbb{E}_{p(\mathbf{c}_n|\mathbf{y}_n; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathbf{y}_n, \check{\mathbf{c}}_n; \boldsymbol{\theta}) \right\}. \end{aligned} \quad (2.18)$$

This results in the expectation-free expression of the auxiliary function:

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \sum_{k,n} p(c_{k,n} = 1 | \mathbf{y}_n; \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\mu}^{\text{old}}, \boldsymbol{\Sigma}^{\text{old}}) \ln p(\mathbf{y}_n, c_{k,n} = 1; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \sum_{k,n} \gamma_{k,n} \ln p(\mathbf{y}_n, c_{k,n} = 1; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \sum_{k,n} \gamma_{k,n} \underbrace{\ln p(c_{k,n} = 1; \boldsymbol{\pi})}_{\ln \pi_k} \underbrace{\ln p(\mathbf{y}_n | c_{k,n} = 1; \boldsymbol{\mu}, \boldsymbol{\Sigma})}_{\ln p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \\
&= \sum_{k,n} \gamma_{k,n} \left(\ln \pi_k - \frac{1}{2} \ln((2\pi)^D \det \boldsymbol{\Sigma}_k) - \frac{1}{2} (\mathbf{y}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_k) \right). \quad (2.19)
\end{aligned}$$

When we now calculate derivatives of the auxiliary function from Equation 2.19 with respect to the parameters $\boldsymbol{\theta}$, we obtain the same update equations as in the direct derivations (compare Equations 2.11 – 2.13), for example:

$$\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})}{\partial \boldsymbol{\mu}_k} &= \frac{1}{2} \sum_n \gamma_{k,n} \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_k) \stackrel{!}{=} 0 \\
\stackrel{\exists \boldsymbol{\Sigma}_k^{-1}}{\Leftrightarrow} \quad \sum_n \gamma_{k,n} \mathbf{y}_n &= \boldsymbol{\mu}_k \sum_n \gamma_{k,n} \\
\Leftrightarrow \quad \boldsymbol{\mu}_k &= \sum_n \gamma_{k,n} \mathbf{y}_n / \sum_n \gamma_{k,n}. \quad (2.20)
\end{aligned}$$

Interestingly, although it seems like using the expectation operator in Equation 2.16 is an approximation, this led us to the same parameter updates as before. Furthermore, this derivation led us to two distinct processing steps and made their naming rather intuitive.

We now briefly prove that maximizing the auxiliary function in the M-step indeed maximizes the likelihood and thereby roughly follow the suggestions in [8, beginning of Section 3] with updated notation. To do so, we first decompose the auxiliary function into the log-likelihood term and a negative cross entropy term by making use of the definition of conditional probabilities and the linearity of the expectation operator:

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathcal{Y}, \check{\mathcal{Z}}; \boldsymbol{\theta}) \right\} \\
&= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathcal{Y}; \boldsymbol{\theta}) + \ln p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}) \right\} \\
&= \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\mathcal{Y}; \boldsymbol{\theta}) \right\} + \mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}) \right\} \\
&= \ln p(\mathcal{Y}; \boldsymbol{\theta}) + \underbrace{\mathbb{E}_{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}})} \left\{ \ln p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}) \right\}}_{- \text{CE}(p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}}), p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}))}. \quad (2.21)
\end{aligned}$$

Hence, it follows:

$$\ln p(\mathcal{Y}; \boldsymbol{\theta}) = Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \text{CE} \left(p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}^{\text{old}}), p(\check{\mathcal{Z}}|\mathcal{Y}; \boldsymbol{\theta}) \right). \quad (2.22)$$

We can now calculate the difference between the log-likelihood after the M-step and before the M-step using a short hand notation for the cross entropy:

$$\begin{aligned}\Delta\ell &= \ln p(\mathcal{Y}; \boldsymbol{\theta}) - \ln p(\mathcal{Y}; \boldsymbol{\theta}^{\text{old}}) \\ &= \left(Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) - Q(\boldsymbol{\theta}^{\text{old}}, \boldsymbol{\theta}^{\text{old}}) \right) + \underbrace{\left(\text{CE}(\boldsymbol{\theta}^{\text{old}}, \boldsymbol{\theta}) - \text{CE}(\boldsymbol{\theta}^{\text{old}}, \boldsymbol{\theta}^{\text{old}}) \right)}_{\text{KL}(\boldsymbol{\theta}^{\text{old}} \parallel \boldsymbol{\theta})}. \quad (2.23)\end{aligned}$$

The difference of auxiliary functions increases by definition of the M-step – after all, the M-step is defined as maximizing the auxiliary function with respect to the parameters $\boldsymbol{\theta}$. The difference of cross entropy terms is positive or zero, because the expected code length under the wrong distribution $\text{CE}(\boldsymbol{\theta}^{\text{old}}, \boldsymbol{\theta})$ is always larger than the entropy $\text{CE}(\boldsymbol{\theta}^{\text{old}}, \boldsymbol{\theta}^{\text{old}})$ or equal when the distributions are equal. This is also known as the Gibb’s inequality which can itself be proven by applying Jensen’s inequality (compare Section A.2). Consequently, $\Delta\ell \geq 0$ and the M-step for the auxiliary function indeed maximizes the likelihood.⁵

c) Decomposition of the likelihood Alternatively, we may decompose the likelihood into a lower bound and a Kullback-Leibler divergence. This view of the EM algorithm is closest to [16] although Neal et al. do not demonstrate the decomposition. Similarly to the decomposition in the last section we first introduce a latent random variable \mathcal{Z} by multiplying the log-likelihood with the integral of the PDF of the latent random variable, here denoted as $q(\mathcal{Z})$ [17, Equation 3.1]:

$$\begin{aligned}\ell &= \ln p(\mathcal{Y}; \boldsymbol{\theta}) = \ln p(\mathcal{Y}; \boldsymbol{\theta}) \int_{\mathcal{Z}} q(\mathcal{Z}) \, d\mathcal{Z} \\ &= \int_{\mathcal{Z}} q(\mathcal{Z}) \left(\ln p(\mathcal{Y}; \boldsymbol{\theta}) + \ln \left(\frac{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) - \ln \left(\frac{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) \right) d\mathcal{Z} \\ &= \int_{\mathcal{Z}} q(\mathcal{Z}) \left(\ln \left(\frac{p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) - \ln \left(\frac{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) \right) d\mathcal{Z} \\ &= \underbrace{\int_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) d\mathcal{Z}}_{=: F(q, \boldsymbol{\theta})} - \underbrace{\int_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) d\mathcal{Z}}_{= \text{KL}(q \parallel p)}.\end{aligned}$$

We have now obtained a decomposition of the log-likelihood into a lower bound $F(q, \boldsymbol{\theta})$ and a Kullback-Leibler divergence $\text{KL}(q \parallel p)$:

$$\ell = \ln p(\mathcal{Y}; \boldsymbol{\theta}) = F(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p). \quad (2.24)$$

The term $F(q, \boldsymbol{\theta})$ is a lower bound of the log-likelihood since the Kullback-Leibler divergence is always greater or equal to zero (compare Section A.2).

⁵ It is worth noting that this actually proves a bit more: It also proves that a generalized EM algorithm improves the likelihood even when the M-step did not find the maximum and just improved the auxiliary function a bit [8], [16].

It is now possible to maximize the likelihood purely by optimizing the lower bound $F(q, \boldsymbol{\theta})$. Neal et al. formulated the corresponding EM algorithm as follows and proved that this is indeed equal to the auxiliary function optimization as stated before [16]:

- Find a posterior distribution q which maximizes the lower bound $F(q, \boldsymbol{\theta})$ while keeping the parameters $\boldsymbol{\theta}$ fixed.
- Maximize the lower bound $F(q, \boldsymbol{\theta})$ with respect to the parameters $\boldsymbol{\theta}$ under the assumption that the posterior q is the true distribution.

Arguably, it might still be a bit nebulous how to maximize the lower bound $F(q, \boldsymbol{\theta})$ (a functional) with respect to a function. In general, it is rather complicated to find a function q without implying further constraints. Keeping in mind that $F(q, \boldsymbol{\theta})$ is a lower bound for the log-likelihood and that the Kullback-Leibler divergence is always positive or zero, it becomes clear that the lower bound can at most reach the log-likelihood. Then, the Kullback-Leibler divergence equals zero, i.e., q and p coincide [18, Page 135]:

$$q(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta}). \quad (2.25)$$

At least for our timeless GMM example we may now maximize the lower bound with respect to the candidate distribution q without arguing via the Kullback-Leibler divergence, i.e., by directly differentiating with respect to the parameters of the candidate distribution $q(c_{k,n}=1) = \text{Cat}(k, \boldsymbol{\gamma}_n)$, where $\boldsymbol{\gamma}_n = (\gamma_{1,n}, \dots, \gamma_{K,n})^\top$. We start by adding a constraint that the distribution has to sum up to one by introducing a Lagrange multiplier [13] for each observation indexed by n :

$$\begin{aligned} F' &= F(q, \boldsymbol{\theta}) + \sum_n \lambda_n \left(\sum_k \gamma_{k,n} - 1 \right) \\ &= \sum_{k,n} \gamma_{k,n} \ln p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta}) - \sum_{k,n} \gamma_{k,n} \ln \gamma_{k,n} + \sum_{k,n} \lambda_n \gamma_{k,n} + \sum_n \lambda_n. \end{aligned} \quad (2.26)$$

We now differentiate with respect to the parameters of the candidate distribution q :

$$\begin{aligned} \frac{\partial F'}{\partial \gamma_{k,n}} &= \ln p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta}) - (1 + \ln \gamma_{k,n}) + \lambda_n \stackrel{!}{=} 0 \\ \Leftrightarrow \ln \gamma_{k,n} &= \ln p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta}) + \lambda_n - 1 \\ \Leftrightarrow \gamma_{k,n} &= p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta}) \cdot e^{\lambda_n - 1}. \end{aligned} \quad (2.27)$$

By using the constraint again, we can identify the constant $e^{\lambda_n - 1}$:

$$\frac{\partial F'}{\partial \lambda_n} \stackrel{!}{=} 0 \Rightarrow \sum_k p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta}) \cdot e^{\lambda_n - 1} = 1 \Leftrightarrow e^{\lambda_n - 1} = \frac{1}{\sum_k p(\mathbf{y}_n, c_{k,n}=1, \boldsymbol{\theta})}. \quad (2.28)$$

Plugging Equation 2.28 into Equation 2.27 confirms our previous observation that the lower bound is indeed maximized when the candidate distribution q coincides with the posterior distribution $p(\mathcal{Z}|\mathcal{Y}; \boldsymbol{\theta})$:

$$q(c_{k,n}=1) = \gamma_{k,n} = \frac{\pi_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} p(\mathbf{y}_n; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}. \quad (2.29)$$

It remains to be argued that maximizing the lower bound $F(q, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is, in fact, equal to the M-step of the auxiliary function approach described in Section 2.5.3. To do so, the lower bound is rewritten slightly:

$$\begin{aligned} F(q, \boldsymbol{\theta}) &= \int_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta})}{q(\mathcal{Z})} \right) d\mathcal{Z} \\ &= \underbrace{\int_{\mathcal{Z}} q(\mathcal{Z}) \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d\mathcal{Z}}_{= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})} - \underbrace{\int_{\mathcal{Z}} q(\mathcal{Z}) \ln q(\mathcal{Z}) d\mathcal{Z}}_{= H(q(\check{\mathcal{Z}}))}. \end{aligned} \quad (2.30)$$

The last term, which turns out to be the entropy of the candidate distribution, is constant with respect to the parameters. Consequently, since the first term is the auxiliary function and the entropy does not depend on the parameters, maximizing the lower bound with respect to the parameters $\boldsymbol{\theta}$ is indeed equivalent to maximizing the auxiliary function with respect to the same parameters. In [18, Equation 10] Tzikas et al. argue similarly, however, in contrast to their presentation we actually did not need to plug in the result of the E-step to prove our point.

d) Special case of the VEM algorithm Based on the likelihood decomposition approach in Section 2.5.3 we argue that the likelihood is maximized when the lower bound is maximized (repetition of Equation 2.24):

$$\ell = \ln p(\mathcal{Y}; \boldsymbol{\theta}) = F(q, \boldsymbol{\theta}) + \text{KL}(q||p).$$

To derive a VEM algorithm, we, therefore, strive to maximize the lower bound with respect to q . To do so, no constraints on q are imposed in general. Specifically, we do not impose any functional form such as *polynomial*. Much rather, we want to perform a free-form optimization (also called variational optimization) of the functional F with respect to the function q . The only structural choice necessary for the derivation in the following is that the variational posterior factorizes with respect to each latent random variable or at least with respect to subgroups of latent random variables. Here, m indexes the latent variable subgroup:

$$q(\mathcal{Z}) = \prod_m q(\mathcal{Z}_m). \quad (2.31)$$

According to this argumentation the successive derivation loosely follows [17, Section 3.2] and [18, Equation 15] with adjusted notation and slightly more focus on transparent intermediate

steps. To do so, we need to insert the factorization in Equation 2.31 into the lower bound. For clarity, we make use of the decomposition in Equation 2.30 and insert the factorization into the auxiliary function and into the entropy independently.

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \int_{\mathcal{Z}} q(\mathcal{Z}) \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d\mathcal{Z} \\
&= \int_{\mathcal{Z}} \prod_m q(\mathcal{Z}_m) \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d\mathcal{Z} \\
&= \int_{\mathcal{Z}_m} q(\mathcal{Z}_m) \underbrace{\int_{\mathcal{Z} \setminus \mathcal{Z}_m} \prod_{m' \neq m} q(\mathcal{Z}_{m'}) \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) d(\mathcal{Z} \setminus \mathcal{Z}_{m'}) d\mathcal{Z}_m}_{=: \ln \tilde{p}(\mathcal{Y}, \mathcal{Z}_m; \boldsymbol{\theta})}, \tag{2.32}
\end{aligned}$$

where $d(\mathcal{Z} \setminus \mathcal{Z}_{m'})$ should be read as the product of all but $d\mathcal{Z}_{m'}$. We identify $\tilde{p}(\mathcal{Y}, \mathcal{Z}_m; \boldsymbol{\theta})$ as that distribution which contains only one of the latent variable subgroups: all others have disappeared due to the marginalization.

We now continue by inserting the factorization in Equation 2.31 into the entropy term on the very right hand side of Equation 2.30:

$$\begin{aligned}
H(q(\check{\mathcal{Z}})) &= - \int_{\mathcal{Z}} q(\mathcal{Z}) \ln q(\mathcal{Z}) d\mathcal{Z} \\
&= - \int_{\mathcal{Z}} \prod_{m'} q(\mathcal{Z}_{m'}) \ln \prod_m q(\mathcal{Z}_m) d\mathcal{Z} \\
&= - \int_{\mathcal{Z}} \prod_{m'} q(\mathcal{Z}_{m'}) \sum_m \ln q(\mathcal{Z}_m) d\mathcal{Z}. \tag{2.33}
\end{aligned}$$

Applying the distributive law, we can now move the product over m' into the summation over m : the multiplication is distributive over addition. Finally, the integration and the summation are switched and the known property that the entropy of independent sources is additive is obtained:

$$\begin{aligned}
H(q(\check{\mathcal{Z}})) &= - \sum_m \int_{\mathcal{Z}} \prod_{m'} q(\mathcal{Z}_{m'}) \ln q(\mathcal{Z}_m) d\mathcal{Z} \\
&= - \sum_m \int_{\mathcal{Z} \setminus \mathcal{Z}_m} \prod_{m' \neq m} q(\mathcal{Z}_{m'}) \underbrace{\int_{\mathcal{Z}_m} q(\mathcal{Z}_m) \ln q(\mathcal{Z}_m) d\mathcal{Z}_m d(\mathcal{Z} \setminus \mathcal{Z}_m)}_{= H(q(\check{\mathcal{Z}}_m))} \\
&= \sum_m H(q(\check{\mathcal{Z}}_m)) \underbrace{\int_{\mathcal{Z} \setminus \mathcal{Z}_m} \prod_{m' \neq m} q(\mathcal{Z}_{m'}) d(\mathcal{Z} \setminus \mathcal{Z}_m)}_{=1} \\
&= \sum_m H(q(\check{\mathcal{Z}}_m)). \tag{2.34}
\end{aligned}$$

We can now combine both previous findings and apply them to the lower bound:

$$\begin{aligned}
F(q, \boldsymbol{\theta}) &= \int_{\mathcal{Z}_m} q(\mathcal{Z}_m) \ln \tilde{p}(\mathcal{Y}, \mathcal{Z}_m; \boldsymbol{\theta}) d\mathcal{Z}_m + H(q(\mathcal{Z}_m)) + \sum_{m' \neq m} H(q(\check{\mathcal{Z}}_{m'})) \\
&= \underbrace{\int_{\mathcal{Z}_m} q(\mathcal{Z}_m) \ln \tilde{p}(\mathcal{Y}, \mathcal{Z}_m; \boldsymbol{\theta}) d\mathcal{Z}_m + \int_{\mathcal{Z}_m} q(\mathcal{Z}_m) \ln q(\mathcal{Z}_m) d\mathcal{Z}_m}_{= -\text{KL}(q(\check{\mathcal{Z}}_m) \| \tilde{p}(\mathcal{Y}, \check{\mathcal{Z}}_m; \boldsymbol{\theta}))} + \text{const.} \quad (2.35)
\end{aligned}$$

Now, it can be deduced that the lower bound can be maximized by adjusting $q(\mathcal{Z}_m)$ for each m independently by minimizing the Kullback-Leibler divergence between the approximate posterior $q(\mathcal{Z}_m)$ and the partially marginalized distribution $\tilde{p}(\mathcal{Y}, \mathcal{Z}_m; \boldsymbol{\theta})$. Since the Kullback-Leibler divergence is always positive or zero and only zero when both distributions coincide, the approximate posterior has to be equal to the partially marginalized distribution in order to maximize the lower bound. To make it more clear, we can obtain the candidate distribution for a latent variable \mathcal{Z}_m by evaluating the expectation operator under all distributions q but the candidate distribution [11, Equation 10.9], [18, Equation 16], [17, Equation 3.16]:

$$\ln q_m(\mathcal{Z}_m) = \mathbb{E}_{q(\check{\mathcal{Z}} \setminus \mathcal{Z}_m)} \left\{ \ln p(\mathcal{Y}, \check{\mathcal{Z}}, \boldsymbol{\theta}) \right\} + \text{const.}, \quad (2.36)$$

where $\check{\mathcal{Z}}$ are all latent random variables and $\check{\mathcal{Z}}_m$ is a single latent random variable or a subset of latent random variables and $\check{\mathcal{Z}} \setminus \check{\mathcal{Z}}_m$ stands for all latent random variables but $\check{\mathcal{Z}}_m$. Therefore, our new E-step is to cycle through this equation for each latent random variable $\check{\mathcal{Z}}_m$ and evaluate the expectation operator to obtain $q(\mathcal{Z}_m)$.

The VEM algorithm can then be summarized [17, Page 13]:

E-step Cycle through Equation 2.36 for each latent random variable.

M-step Maximize the lower bound $F(q, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

If there is only one group of latent variables, the expectation operator in Equation 2.36 becomes obsolete: there is no other latent random variable. Consequently, we can simplify the expression and capture all terms not depending on \mathcal{Z} in the additive constant:

$$\begin{aligned}
\ln q(\mathcal{Z}) &= \ln p(\mathcal{Y}, \mathcal{Z}; \boldsymbol{\theta}) + \text{const.} \\
&= \ln p(\mathcal{Z} | \mathcal{Y}; \boldsymbol{\theta}) + \ln p(\mathcal{Y}; \boldsymbol{\theta}) + \text{const.} \\
&= \ln p(\mathcal{Z} | \mathcal{Y}; \boldsymbol{\theta}) + \text{const.} \quad (2.37)
\end{aligned}$$

For the classic GMM example this results in the expected solution:

$$\ln q(\mathbf{c}_n) = \ln p(\mathbf{c}_n | \mathbf{y}_n; \boldsymbol{\theta}) + \text{const.} \quad \Leftrightarrow \quad q(\mathbf{c}_n) = p(\mathbf{c}_n | \mathbf{y}_n; \boldsymbol{\theta}) = \gamma_{k,n}. \quad (2.38)$$

The M-step is the same optimization problem as discussed in Section 2.5.3 before. In summary, the EM algorithm is de facto a special case of the VEM algorithm.

e) Special case of the MM algorithm The EM algorithm can be identified as a special case of the very general MM algorithm [10], [19]. This is mentioned here for reasons of completeness but is not shown in detail. Falling back to the MM algorithm is advantageous when the variational approximation is not sufficient to find a solution.⁶ An example in which Azcarreta et al. applied the MM algorithm to spatial clustering of multi-channel speech signals can be found in [20].

⁶ An example in which MM has not been applied yet, but in which it might lead to a better model is a prior distribution for the concentration parameter of a complex Watson distribution. Although first derivations were performed to find such a prior, efficient parameter updates were unavailable at that time.

3 Blind source separation principles

This chapter introduces the relevant foundations of BSS upon which this thesis is built. It is organized as follows: First, a general overview of source separation algorithms and their goal is provided. Second, single-channel as well as multi-channel approaches that yield an intermediate signal, e.g., a time-frequency mask, are introduced in Section 3.1 and Section 3.2, respectively. Third, two instances of source extraction categories which are informed by the aforementioned masks are discussed in Section 3.3. Although fixed (data-independent) approaches are known, we constrain the discussion to data-dependent source extraction techniques, namely masking and spatial filtering/ beamforming.

BSS aims at estimating the underlying signal components of an observed mixture without access to parts of the source signals or the transfer characteristics from the source to the receiving sensor. Although its applicability generally ranges from medical applications such as disentangling electroencephalography (EEG) signals to seismographic signals to detect earthquakes, we here want to constrain ourselves to approaches particularly derived for speech mixtures.

When separating speech mixtures, the goal is to either obtain an estimate of each speech image or an estimate of each speech source signal. The goal is sometimes defined rather vaguely, since a source separation system may inherently also partially dereverberate a signal and, therefore, might approximate a scaled and dereverberated version of the source signal better than the speech image itself.

The term *blind* refers to the degree of how much a priori knowledge is assumed to be available to a system. Generally speaking, no BSS system is entirely blind. Most systems make some assumptions about the source signal, here speech, which informs how the system itself is designed. This may include the assumption that speech is rather sparse in the STFT domain [21], [22] which led to sparsity-based methods or the assumption that speech is limited in frequency, leading to systems operating on only up to 8 kHz or 16 kHz. More practically, source separation systems within the scope of this thesis are considered blind when they do not use knowledge of the geometry of the setup, e.g., the source locations or the array geometry, when they do not make use of externally provided diarization information, speaker identity or any other kind of speaker information, and do not assume knowledge of interference signals beforehand.

In case signals do not overlap, it seems natural to select the relevant signal simply by masking. In particular, this is used in digital communications in which a communication channel can be occupied by multiple participants using, e.g., time-division multiplexing or frequency-division multiplexing. Yilmaz et al. nicely demonstrated that a sparsity assumption is well valid for speech signals in the STFT domain (with a certain orthogonality measure, namely W-disjoint orthogonality, [22, Section II.]). In other words: although speech signals may overlap in

the time domain the assumption that speech signals overlap rarely in the STFT domain approximately holds [21], [22]. Rickard demonstrates in [23, Figure 8.4] that the W-disjoint orthogonality is maximized for $K = 2$ up to $K = 8$ speakers for a discrete Fourier transform (DFT) window size of 1024 (64 ms) for a 16 kHz speech signal.

Although being aware of time-domain source separation, we here constrain the discussion to frequency domain approaches which, at least to some degree, rely on sparseness in the STFT domain or derived domains. The general processing pipeline, therefore, consists of (1) obtaining a speech mixture from a multi-channel microphone array, (2) analyzing the waveform with a STFT to obtain an STFT signal, (3) applying a clustering or separation algorithm in the STFT domain that yields some form of masks, (4) extracting each source in the STFT domain by either using the mask directly (compare Section 3.3.1) or within a mask-based beamforming step (compare Section 3.3.2), and finally (5) a synthesis of the waveform with an inverse short time Fourier transform (ISTFT). The reconstruction result in the time-domain can then be evaluated with signal-level metrics or be transcribed in a speech recognizer to obtain word error rates (WERs).

3.1 Principles of single-channel approaches

This section addresses single-channel approaches to BSS. To be able to separate sources given only one channel one needs to rely on additional constraints such as independence of the source signals, sparseness in a particular domain, or sophisticated source models. This section is organized into shallow methods in Section 3.1.1 more heavily relying on the aforementioned assumptions or grouping principles and Section 3.1.2 containing selected DNN-based approaches in which less structure is enforced and knowledge is first and foremost obtained through excessive training on dedicated databases.

3.1.1 Shallow methods

In an entirely blind setup and without additional constraints, it is generally impossible to separate sources in a single-channel recording [24]. Such constraints that enable separation are independence constraints [25], sparseness constraints, or constraints on the source itself [26] leading to different well-known algorithms, as briefly introduced in the following.

Constraints on the source itself can be encapsulated into a source model. Ellis groups these into explicit source models, which aim to memorize realistic source signals and implicit source models, in which particular cues encoded into functions are tested against the source signal [24]. Factorial models [27] treat the separation problem as an inference problem, in which the sources are assumed to be well modeled one just needs to figure out how they explain the given mixture. However, this requires learned models for each source beforehand. Given additional sequential constraints, these can also be speaker-independent, e.g., by introducing a (hard) transition matrix between states as nicely illustrated in [24, Figure 3.2] or state transition probabilities resulting in factorial hidden Markov models (HMMs) [28], [29]. Slightly differently, computational auditory scene analysis (CASA) models encode

knowledge about source properties into functions applied to the signal [30]–[32]. Such a grouping principle can be for example proximity in time and frequency, harmonicity, common onsets, or other hand-crafted properties.

In contrast, while the aforementioned models assume independence since no cross-source statistics are captured, independent component analysis (ICA) explicitly maximizes some measure of independence [33], [34], e.g., by approximating uncorrelatedness of any function of the source signals (not just uncorrelatedness of the signals themselves). Smaragdis compares ICA with CASA and goes so far as to argue that the rather heuristically obtained grouping principles in CASA are better captured with independence assumptions naturally leading to ICA [35].

Sparse decomposition can be seen as a generalization of independence and sparsity. Source signals themselves are not directly assumed to be independent or sparse. Much rather, it is assumed that their decomposition coefficients given a possibly overcomplete dictionary are independent and sparse [36]. In that sense, nonnegative matrix factorization (NMF) is a decomposition technique into a dictionary of bases and corresponding activations [37]. Subsequently, NMF has been extensively applied to mixtures of audio signals [38] adding additional constraints or generalizing to convolutive NMF [39].

Although the aforementioned approaches are not directly compared in the remainder of this work, a brief understanding of the underlying assumptions helps to better understand other algorithms referred to in the following. Especially when discussing other integration approaches, e.g., in Section 3.2 or more extensively in Section 4.1 beyond the ones derived in this thesis, the aforementioned shallow approaches are going to reappear either in a generalized form or as a sub-model.

3.1.2 Deep-learning methods

Motivated by the success of deep learning in domains such as speech recognition, image recognition, and segmentation researchers have applied neural networks to speech enhancement as well as speech separation. Early deep learning approaches to source separation trained a separate network for each target speaker which maps from the mixture signal to the speech signal of the target speaker [40] and were, thus, only applicable when the speaker at test time had already been seen during training (closed condition). Tu et al. introduced training networks with a mean squared error (MSE) loss not just for the target speaker but also for the interfering speaker and claim that this training scheme leads to better generalizability when it comes to unseen interferers [41]. They encourage temporal continuity by using neighboring context frames for separation and already encourage multi-style training with varying mixture ratios. That work and its follow-up improvement can certainly be seen as a generalization of [40] in the sense that although the target test speaker needs to be known during training the interference test speaker can differ from training interferences [41, Section 3.3], [42].

The development of deep clustering (DC) is an important breakthrough as it can operate with entirely unseen speakers at test time (open condition) [43]. Not much later Yu et al. published the first successful system, which addresses the source separation problem similarly

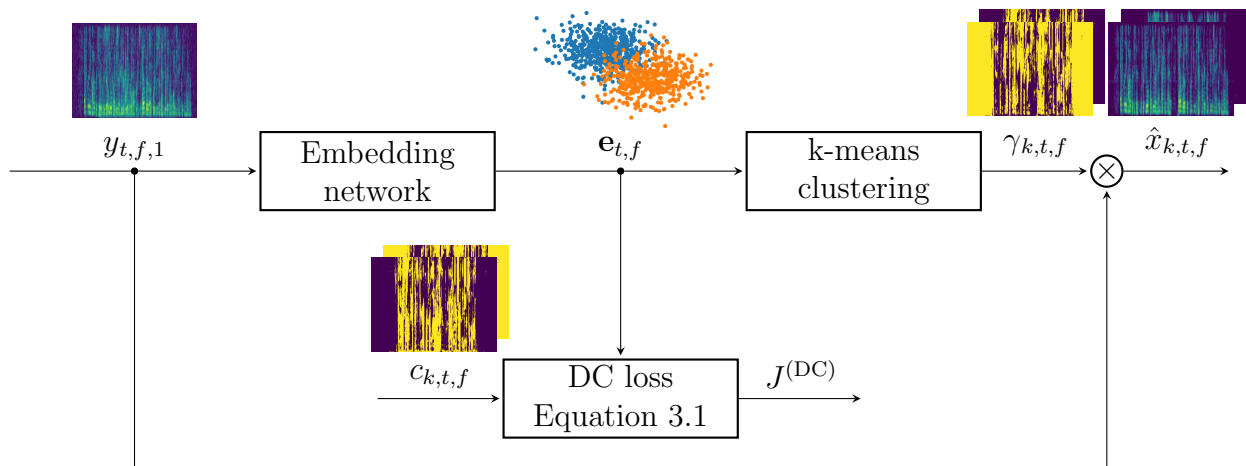


Figure 3.1: Overview of the DC processing steps. The vanilla implementation operates on a single-channel which in this case is, without loss of generality, set to $d = 1$. Furthermore, $c_{k,t,f}$ represents a ground-truth supervision mask such as an ideal binary mask (IBM).

to [41] while allowing the prediction to be arbitrarily permuted [44], [45]. This concept, allegedly solving the permutation problem, was already mentioned briefly in [43] while not being successfully applied. Deep attractor networks (DANs) are an interesting variant of DC in the sense that they infer a latent embedding representation which is intrinsically permutation invariant while enforcing separability with a reconstruction loss without the need for a permutation invariant training (PIT) strategy [46].

3.1.2.1 DC: Deep clustering

DC is a method to separate a single-channel speech mixture into an estimate for each speaker's contribution to the mixture [43]. It is probably the first neural network-based source separation approach that addresses the label ambiguity problem: one is interested in separated source estimates but the actual ordering (e.g., the youngest speaker first) does not matter. The key component is to train a neural network that translates an amplitude spectrogram into an embedding vector per time-frequency bin. Then, applying a clustering algorithm such as k-means on the embedding vectors allows one to obtain a posterior mask for each source. The underlying idea is that the DNN transforms the information in such a way that separability by a simple clustering algorithm is alleviated. Figure 3.1 summarizes these processing steps.

Training embeddings, e.g., to group similar images or to find similar words can be seen as weakly-supervised training: instead of using the class label as supervision, one just uses the information that objects belong or do not belong to the same class. The standard loss is a contrastive loss which encourages to move embeddings belonging to the same class closer together and embeddings belonging to different classes further apart. Figure 3.2 illustrates this behavior.

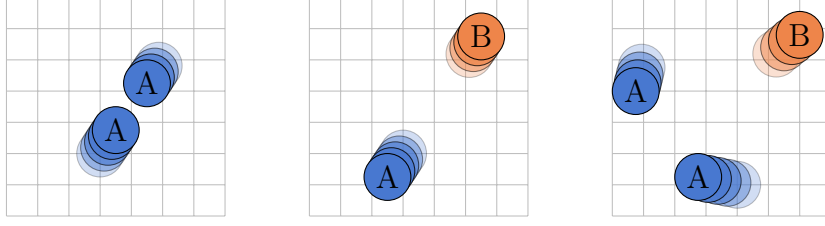


Figure 3.2: Different sketches to illustrate a contrastive embedding loss. From left to right: two embedding vectors from the same class attract each other, two embedding vectors from different classes repel each other, both effects complement each other.

Although there are many variants to the contrastive loss, such as triplet loss, Hershey et al. decided to use a very specific variant of the contrastive loss which can be reformulated in a fairly resource-efficient way. The DC loss is given as follows [43, Equation 1]:

$$J^{(\text{DC})} = \left\| \hat{\mathbf{A}} - \mathbf{A} \right\|_{\text{F}}^2 = \left\| \mathbf{E}\mathbf{E}^{\text{T}} - \mathbf{C}\mathbf{C}^{\text{T}} \right\|_{\text{F}}^2. \quad (3.1)$$

Here, $\hat{\mathbf{A}}$ and \mathbf{A} are the estimated and the ground truth affinity matrices with shape $TF \times TF$. The entries in \mathbf{A} encode if two time-frequency slots belong to the same class. The matrices \mathbf{E} and \mathbf{C} consist of the stacked embedding vectors and the stacked ground-truth class affiliation vectors, respectively:

$$\mathbf{E} = (\mathbf{e}_{1,1} \quad \dots \quad \mathbf{e}_{T,F})^{\text{T}} \in \mathbb{R}^{TF \times E}, \quad \mathbf{C} = (\mathbf{c}_{1,1} \quad \dots \quad \mathbf{c}_{T,F})^{\text{T}} \in \mathbb{R}^{TF \times K}. \quad (3.2)$$

Regardless of the fact that Equation 3.1 suggests that $(TF)^2$ inner products need to be calculated, [43] demonstrates that the effective number of inner products is much lower when an appropriate reformulation of Equation 3.1 is performed.

An early extension added a subsequent mask refinement network and trained that jointly with the embedding network [47]. Other noteworthy extensions of DC are alternative objective functions [48] and multi-channel DC as better explained in Section 3.2.2 [49]. Another larger performance boost was achieved by combining the DC loss with an additional permutation invariant mask inference loss (compare Section 3.1.2.3) [48], [50]. Further, DANs are a noteworthy variant of DC which allow a signal reconstruction loss. DANs are explained in the next section.

3.1.2.2 DAN: Deep attractor network

Just as DC, DANs were developed to separate speech from a single-channel mixture [46]. To do so, a DAN relies on an embedding network similar to DC but avoids the DC loss function. First, the embedding vectors $\mathbf{e}_{t,f}$ are calculated by a DNN. Second, a weighted average of the embedding vectors is calculated using an oracle class affiliation mask to obtain mean embeddings:

$$\boldsymbol{\mu}_k = \sum_{t,f} c_{k,t,f} \mathbf{e}_{t,f} / \sum_{t,f} c_{k,t,f}. \quad (3.3)$$

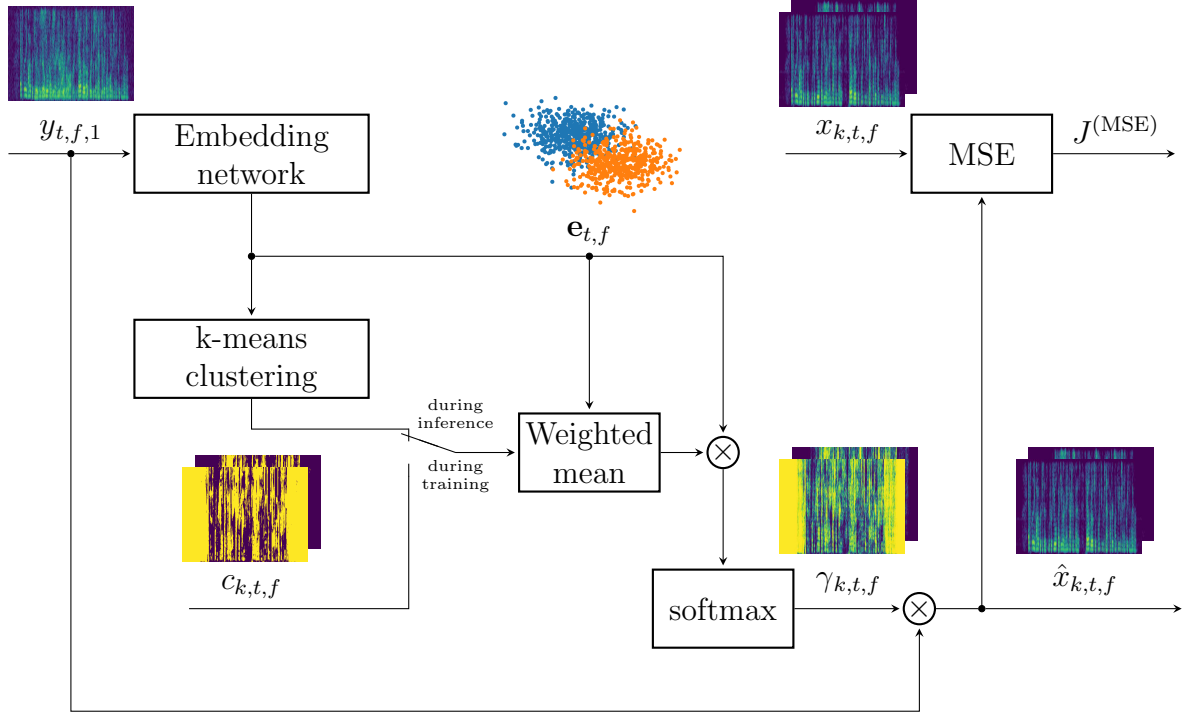


Figure 3.3: Overview of the DAN processing steps with a signal reconstruction loss. During training, an oracle class affiliation mask is used to calculate attractors similar to an M-step in GMM clustering.

Referring to alleged brain mechanisms, these mean vectors are called *attractors* in the original work [46]. Interestingly, although the authors of [46] did not mention this connection, this is reminiscent of supervised GMM parameter estimation: Given the latent class affiliations the mean vectors of a GMM are calculated just as in Equation 3.3. Third, an inner product between each embedding vector and each attractor is calculated to obtain logits,¹ which can then be mapped to probabilities with a softmax function over k [46]:

$$\gamma_{k,t,f} = \text{softmax}_k(\boldsymbol{\mu}_k^\top \mathbf{e}_{t,f}) = e^{\boldsymbol{\mu}_k^\top \mathbf{e}_{t,f}} / \sum_{k'} e^{\boldsymbol{\mu}_{k'}^\top \mathbf{e}_{t,f}}. \quad (3.4)$$

Finally, the embedding network can be trained either with a mask loss (such as cross entropy (CE) comparing a predicted mask with an oracle mask) or with a signal level loss by applying the mask to the mixture spectrogram first [46]. Figure 3.3 shows the training and inference steps for a reconstruction loss.

Albeit avoiding the DC loss function and being able to train with a reconstruction loss, the training conditions still differ slightly from the test conditions: During test time, the oracle masks are replaced by the output of the k-means clustering algorithm. This mismatch can be addressed by training a mask-refinement network similar to [47] or by retraining the downstream task, e.g., ASR as in [51, Table 3].

¹ Here, *logit* refers to the log-probability before applying the sigmoid or softmax function. The term goes back to 1944 when Joseph Berkson used the *logistic function* to map probabilities in $[0, 1]$ to $[-\infty, \infty]$. However, there is no logistic function involved here.

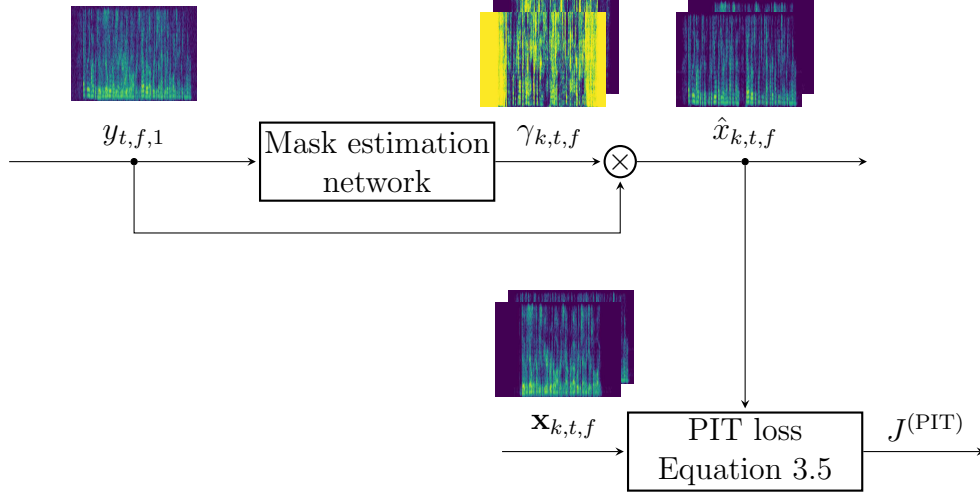


Figure 3.4: Overview of the PIT processing steps. The PIT loss calculates a given metric, e.g., MSE for each possible permutation of classes and then just forwards the minimal loss.

3.1.2.3 PIT: Permutation invariant training

PIT [44], [45] is a different approach to address the global speaker label ambiguity which greatly streamlined the processing scheme and led to many top-performing separation models such as TasNet [52] or DPRNN [53]. The idea is to skip the embedding representation entirely and train a network that directly estimates posterior masks from the observation. In doing so, the neural network cannot be aware of the order in which the different target signals appear. Ideally, one would like to take the oracle permutation and just calculate a mask loss such as CE or a signal reconstruction loss such as MSE on the accordingly permuted signals. One option is to externally calculate the ideal assignment, possibly with a different metric. A much more self-contained solution is to calculate the desired loss for each possible permutation and then use that loss for backpropagation which led to the lowest value [44], [45], where Π is any permutation of $(1, \dots, K)$ (compare Appendix A.7):

$$J^{(\text{PIT})} = \underset{\Pi}{\operatorname{argmin}} \sum_{k,t,f} \text{MSE}(\hat{x}_{\Pi(k),t,f}, x_{k,t,f}). \quad (3.5)$$

When training with an additional noise class, one can keep the position of the noise output fixed to somewhat reduce the number of permutations to test. However, although the number of permutations needed for Equation 3.5 is the factorial of the number of classes, the number of MSE losses which actually need to be calculated scales proportional to the square of the number of classes: One can first calculate the quadratic matrix containing the loss of each image $x_{k,t,f}$ with each prediction $\hat{x}_{k',t,f}$ and then pick those precomputed losses from that matrix to evaluate Equation 3.5.

3.1.3 Discussion of single-channel deep-learning methods

The main advantage of DC and DANs over PIT is that the embedding network is entirely independent of the number of speakers, i.e., the same weights can be used to infer two and

Table 3.1: Conceptual comparison of single-channel deep learning-based approaches to BSS each appearing as baseline systems in Chapter 5.

	DC	DAN	PIT
Embedding	✓	✓	✗
Allows reconstruction loss	✗	✓	✓
Number of sources arbitrary during inference	✓	✓	✗
Computational complexity of the loss	high	low	low

three speaker mixtures [43, Table 3]. However, the number of speakers has to be known before the k-means step during inference, or at least estimated by an external system. The advantage of DANs and PIT over DC is that they allow a signal reconstruction loss: the resulting masks during training can be multiplied with the input spectrogram and the resulting speech estimate can be compared with the ground truth source signals. However, DC variants such as [47] and Chimera++ [48] again allow to train with a signal reconstruction loss, regardless. Another advantage of PIT over DC and DANs is that it is easier to design a frame-online algorithm using PIT than DC or DANs. This is caused by the fact that DC and DANs both require k-means to run on a certain minimum signal length. However, all methods can be generalized to block processing by, e.g., separating blocks independently and then solving the block permutation problem (the problem that the speaker index may be inconsistent across blocks) later. Table 3.1 roughly summarizes the key differences between DC, DANs, and PIT.

3.2 Principles of multi-channel approaches

The main cue for multi-channel separation are cross-channel features: the spatial characteristics of each source can be obtained by using inter-channel phase differences (IPDs) and inter-channel level differences (ILDs). Figure 1 and Figure 3 in [55] conceptually visualize IPDs and ILDs, respectively. Although single-channel approaches such as DC may already profit from additional channels due to the added redundancy (see, e.g., Table 5.26 for a limited comparison), cross-channel dependencies add an additional source of information and, depending on the complexity of the scenario, can lead to good source separation results.

Multi-channel approaches consist of a variety of methods developed in parallel [54]. Figure 3.2 shows an overview of selected algorithms.

Widely known methods include multi-channel NMF [56], [57], a method bending the term *nonnegative* to some degree: To also capture IPDs multi-channel NMF is, in fact, a semi-nonnegative modeling approach in which the latent source power is modeled with NMF and the mixing conditions are addressed with other means [58, Page 74]. An introduction to multi-channel NMF can be found in [58, Chapter 4].

Additional methods are independent vector analysis (IVA) [59]–[61] and independent low-rank matrix analysis (ILRMA) [62] (an approach unifying multi-channel NMF and IVA) which

Table 3.2: Conceptual comparison of multi-channel approaches to BSS. The visualization closely follows [54, Figure 1]. The approaches marked with an asterisk require an additional dimensionality reduction. Approaches highlighted with a blue box are discussed in more detail in this work. Other single-channel approaches are briefly introduced in Section 3.1.1, while other multi-channel approaches are referenced in the introduction of Section 3.2.

Training data	Single-channel	Multi-channel		
	$D = 1$	Underdetermined $K > D$	Determined $K = D$	Overdetermined $K < D$
\times			ICA	ICA*
\times			ILRMA	ILRMA*
\times		Spatial clustering in Section 3.2.1		
\times	NMF	Multi-channel NMF		
\checkmark				
\checkmark	DNN-based methods in Section 3.1.2			

are focusing on the determined case: the number of speakers coincides with the number of channels. These can be generalized to overdetermined scenarios, i.e., situations in which the number of channels exceeds the number of speakers, by preprocessing the observations with a dimensionality reduction. Nevertheless, in the following, we will not address these independence assumption-based methods in any more detail.

A more versatile class of multi-channel separation methods are clustering-based formulations [22], [55], [63]–[66]. These concepts in principle allow to address the underdetermined case (fewer sensors than sources) by assuming that the observations stem from a structured generative model: first, class labels are sampled which indicate which observation belongs to which source or noise, second, the observations are sampled from differently parameterized distributions based on their class affiliation. Although the underdetermined case is supported, in practice, these methods are often applied with six or more channels – more channels result in better separation performance nonetheless.

A rather early overview of spatial features for clustering-based techniques can be found in [23, Section 9.4]. Although Table 9.2 in [23] compares many heuristically motivated spatial features, it already provides features quite similar to normalized observations as used in modern spatial clustering approaches (compare Section 3.2.1). The choice of the actual spatial feature, potentially a transformation of the observation vector $\mathbf{y}_{t,f}$, heavily depends on the separation approach at hand. For probabilistic mixture model-based approaches, the feature choice depends on the availability of adequate probability density functions and the complexity of the parameter estimation process. In contrast, for multi-channel neural network-based approaches, features are typically selected in such a way that the range of possible values is limited and that discontinuities are avoided: to name an example, phase

difference is often avoided due to the discontinuity at the wrapping point, whereas sine and cosine of the phase difference are continuous and bound to $[-1, 1]$.

Quite a different approach, not further analyzed within this work, is to precalculate a set of fixed beamforming vectors for a given microphone array. Then a subsequent system just selects the channels which lead to the best source separation results. To name an example, Chen et al. suggests to apply a DAN to a limited number of fixed beamformer outputs and then uses an additional system to select the best separation result [67].

First, Section 3.2.1 introduces spatial clustering approaches with their corresponding spatial features. Then, Section 3.2.2 highlights, how spatial features can be used with neural networks. Although beamforming is often introduced as a multi-channel source separation approach (see, e.g., [68]), it is here introduced as an instance of source extraction methods alongside masking approaches in Section 3.3.

3.2.1 Probabilistic spatial mixture models

Probabilistic spatial mixture models are a way to address the cocktail party problem based on the sparseness assumption of speech in the STFT domain: speech occupies a small fraction of time-frequency bins in the STFT domain. This assumption holds well for the instantaneous mixing of a few sources in a noise-free scenario. However, in more realistic scenarios reverberation effects cause the different source signals to cover more time-frequency bins. In particular, background noise almost always violates the sparseness assumption.

However, this sparseness assumption can be relaxed: it is sufficient to assume that each time-frequency bin is dominated by one source or noise. By doing so, one can again assign class labels to each time-frequency bin and model the multi-channel observation with a weighted sum of component distributions:

$$p(f(\mathbf{y}_{t,f})) = \mathbb{E}_{p(\mathbf{c}_{t,f})} \{p(f(\mathbf{y}_{t,f}), \mathbf{c}_{t,f})\} = \sum_k p(c_{k,t,f}=1) p(f(\mathbf{y}_{t,f}) | c_{k,t,f}=1), \quad (3.6)$$

where $f(\cdot)$ is a feature extraction method and $c_{k,t,f}$ is the latent class affiliation modeling from which class a particular time-frequency bin stems: each class models either a source or the background noise.

All algorithms in the remaining part of this chapter either use the observation vectors $\mathbf{y}_{t,f}$ directly or use normalized features $\tilde{\mathbf{y}}_{t,f}$ [69, Equation 12]:

$$\mathbf{y}_{t,f} = (y_{t,f,1}, \dots, y_{t,f,D})^\top \in \mathbb{C}^D, \quad (3.7)$$

$$\tilde{\mathbf{y}}_{t,f} = \mathbf{y}_{t,f} / \|\mathbf{y}_{t,f}\| \in \mathbb{C}\mathcal{S}^{D-1} \quad \text{with} \quad \mathbb{C}\mathcal{S}^{D-1} = \{\tilde{\mathbf{y}} \in \mathbb{C}^D : \tilde{\mathbf{y}}^H \tilde{\mathbf{y}} = 1\}. \quad (3.8)$$

This normalization ensures that neither the power in a particular time-frequency bin nor the scaling of the entire input signal influences the clustering result. This elegantly incorporates the scale ambiguity [2, Section III.C] but at the same time avoids using the power information, e.g., as a reliability cue.

The sparseness assumption can be relaxed further by assuming a nonsparse (not to say dense) noise source while sticking to sparse speech sources at the cost of increased computational requirements [70]. Another way is to improve sparseness by either applying a dereverberation algorithm such as weighted prediction error (WPE) [71], [72] first, or formulating a joint separation and dereverberation algorithm [73].

The remainder of this section is organized as follows: Section 3.2.1.1 introduces the frequency permutation problem. Section 3.2.1.2 addresses different initialization schemes and Section 3.2.1.3 discusses particular choices of mixture weights. The remaining sections discuss concrete manifestations of spatial mixture models and corresponding parameter update rules.

3.2.1.1 Frequency permutation problem

The aforementioned spatial mixture models neglect frequency dependencies. Although, under some conditions, a normalization of the features in such a way that all can be processed frequency-independently is possible [74] almost all recent publications suggest to operate at least in part independently of frequency. Thus, when clustering is performed without any kind of guidance, it will yield a solution in which the speaker index is inconsistent over frequency bins. This issue is the so-called frequency permutation problem [75]. This can be addressed by a variety of approaches [2, Page 14] which may or may not be already incorporated into the optimization process: a linear constraint such as in minimum variance distortionless response (MVDR) beamforming can be applied [76], a similarity measure between neighboring beamforming vectors/ relative transfer functions (RTFs) may be used [77], or a similarity to the corresponding anechoic steering vector can be considered [78]. However, within the scope of this work permutation alignment is used as a separate post-processing step which maximizes the correlation of neighboring frequency bins [75] [66], [79], [80]. In particular, a greedy maximization of the similarity of the posterior mask of neighboring frequency bins as proposed by Tran Vu [81, Section 5.6] is used where indicated. It is mainly influenced by [66], [75], [80], [82] and nicely visualized in [83, Section 4.4].

3.2.1.2 Initialization

Probabilistic spatial mixture models tend to be very susceptible to initialization. The two possibilities are to either initialize with values for the class affiliation posteriors $\gamma_{k,t,f}$ or to initialize with values for the class-dependent parameters and the mixture weights.

To be able to initialize with values for the class-dependent parameters one often needs external knowledge such as the approximate source positions [84]. It is also possible to distribute the initial values randomly in their domain of definition, e.g., in [17, Page 33] complex Watson mode vectors were drawn from a uniform distribution on the surface of the complex unit hypersphere.

Initialization with values for the class affiliation posteriors $\gamma_{k,t,f}$ tends to be easier to implement because knowledge of the class-dependent distribution is not required. One option is to sample the class affiliation posteriors $\gamma_{k,t,f}$ i.i.d. from, e.g., a Dirichlet distribution or even from

a uniform distribution with a subsequent normalization. Alternatively, one may randomly assign wider vertical stripes to each of the classes, which alleviates the frequency permutation problem a bit and encourages the class-dependent parameters to initially be more spread out.

A more elaborate way is to better analyze the observation first. Tran Vu et al. proposed a deflation scheme that selects high energy regions first and then calculates class-dependent parameters on those for initialization [85]. Duong et al. proposed a hierarchical agglomerative clustering approach for initialization [86]. Further, one can perform some form of preclustering, e.g., with an online preclustering as done in [87] or simply with k-means [88] or variants thereof [89].

Keeping in mind that a good separation result is, of course, an excellent initialization, we here refer to the proposed weak integration in Section 4.2. A comparison of different initialization schemes is given in [85, Figure 2], [83, Section 6.8], and in Section 5.5.2.

3.2.1.3 Influence of the mixture weight

The formulation in Equation 3.6 intentionally omits the particular shape of the mixture weight $p(c_{k,t,f}=1) = \pi_{k,t,f}$. However, in a given instance of this formulation, the a priori distribution is often set to be constant along at least one of the indices (k , t , or f). Although in terms of notational burden, this is a minor change, it can have an interesting impact on the overall solution. A constant mixture weight $\pi_{k,t,f} := 1/K$ tends to lead to a more even distribution and, thus, avoids to cancel all speakers when a large portion of the signal is noise-only. A time-independent mixture weight $\pi_{k,t,f} := \pi_{k,f}$ is the most common choice. It allows us to entirely model each frequency bin independently, which may be advisable since a few frequencies tend to be purely noise. A frequency-independent mixture weight $\pi_{k,t,f} := \pi_{k,t}$ is seen less often but has the desirable property that it alleviates the permutation problem to some degree [90]. It is also possible to use a Dirichlet prior for the mixture weight to more carefully control how many observations are assigned to one class on average [17], [66].

3.2.1.4 Complex Watson mixture model

The complex Watson mixture model (cWMM) is a spatial clustering model which is used to separate different sound sources based on spatial cues, namely IPDs and ILDs [91], [92]. These cues are encoded in the normalized complex-valued observation vector which is defined on the complex unit hypersphere \mathbb{CS}^{D-1} which itself is a subset of the complex domain \mathbb{C}^D according to Equation 3.8. The motivation for operating on the normalized observation vectors $\tilde{\mathbf{y}}_{t,f}$ is to maintain as much information in the signal which is related to the acoustic mixing conditions but to remove all audio source related properties.

A cWMM is a spatial mixture model with complex Watson distributions [93] as class conditional distributions. Typically, it is applied independently per frequency bin f [91],

[92]. In its generic form, the distribution is the marginal of the complete data distribution:

$$p(\tilde{\mathbf{y}}_{t,f}) = \sum_k p(c_{k,t,f}=1)p(\tilde{\mathbf{y}}_{t,f}|c_{k,t,f}=1), \quad (3.9)$$

where $p(c_{k,t,f}=1)$ is a categorical distribution which is parameterized with the mixture weights $\boldsymbol{\pi}_f = (\pi_{1,f}, \dots, \pi_{K,f})^\top$ and $p(\tilde{\mathbf{y}}_{t,f}|\mathbf{c}_{t,f})$ is a complex Watson distribution.

The complex Watson distribution is defined as follows [93, Equation 1]:

$$p(\tilde{\mathbf{y}}_{t,f}|c_{k,t,f}=1) = \mathcal{CW}(\tilde{\mathbf{y}}_{t,f}; \kappa_{k,f}, \mathbf{w}_{k,f}) = \frac{1}{c_W(\kappa_{k,f})} e^{\kappa_{k,f} |\mathbf{w}_{k,f}^H \tilde{\mathbf{y}}_{t,f}|^2}, \quad (3.10)$$

with the class-conditional concentration parameters $\kappa_{k,f}$ and mode vectors $\mathbf{w}_{k,f}$ for each frequency bin. The mode vectors are constrained to unit-length and define the direction around which the observations accumulate. A potential precursor of the cWMM is the line-orientation model in [66, Section III.A]. The concentration parameters influence how much the observations are concentrated around the mode vectors and are forced to be nonnegative. A concentration of zero implies that the observations are distributed uniformly on the complex unit hypersphere. It is worth noting that the distribution is circularly-symmetric, i.e., $p(\tilde{\mathbf{y}}_{t,f}|c_{k,t,f}=1) = p(\tilde{\mathbf{y}}_{t,f}e^{j\varphi}|c_{k,t,f}=1)$. This elegantly addresses the absolute phase ambiguity of an observation received at a sensor array: the time of flight from the source to the array is unknown anyway and, therefore, the absolute phase should not be used for clustering.

The normalization constant $c_W(\kappa_{k,f})$ can be obtained by integration of the PDF over the surface \mathbb{CS}^{K-1} of the complex unit hypersphere and then separating $c_W(\kappa_{k,f})$. The resulting analytic form is given as follow [93, Equation 1]:

$$c_W(\kappa_{k,f}) = \frac{2\pi^D {}_1F_1(1, D, \kappa_{k,f})}{(D-1)!}, \quad (3.11)$$

where ${}_1F_1(\cdot, \cdot, \cdot)$ is the confluent hypergeometric function [94, Equation 13.2.2]. Issues regarding numerical stability can be found in [17, Appendix 1].²

To obtain a maximum likelihood estimate of all involved parameters, an EM algorithm is used (compare Section 2.5.3). The E-step evaluates the posterior of the class labels $\gamma_{k,t,f} = p(\tilde{k}_{t,f} = k|\tilde{\mathbf{y}}_{t,f})$ according to, e.g., Equation 2.25:

$$\ln \gamma_{k,t,f} = \ln \pi_{k,f} - \ln {}_1F_1(1, D, \kappa_{k,f}) + \kappa_{k,f} \mathbf{w}_{k,f}^H \tilde{\mathbf{y}}_{k,f} \tilde{\mathbf{y}}_{k,f}^H \mathbf{w}_{k,f} + \text{const.} \quad (3.12)$$

During the M-step the lower bound is maximized with respect to the parameters. We obtain the following updates [93, Section 4], [81, Appendix B.5], [83, Section 2.4.2]:

$$\pi_{k,f} = \frac{1}{T} \sum_t \gamma_{k,t,f}, \quad (3.13)$$

² A numerical implementation in Matlab can be found in `libDirectional` [95]. A Python implementation can be found in `pb.bss` (<https://github.com/fgnt/pb.bss>).

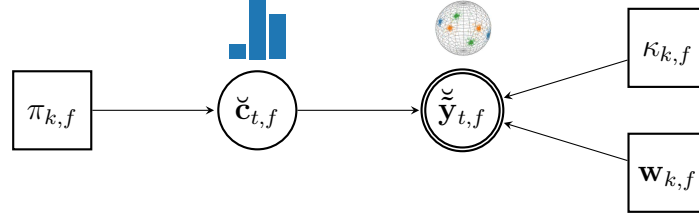


Figure 3.5: Graphical model of a cWMM. The complex-valued distribution is visualized by its real-valued counterpart. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

$$\mathbf{w}_{k,f} = \mathcal{P}(\Phi_{\tilde{\mathbf{y}}\tilde{\mathbf{y}},k,f}) \quad \text{with} \quad \Phi_{\tilde{\mathbf{y}}\tilde{\mathbf{y}},k,f} = \sum_t \gamma_{k,t,f} \tilde{\mathbf{y}}_{t,f} \tilde{\mathbf{y}}_{t,f}^H / \sum_t \gamma_{k,t,f}, \quad (3.14)$$

$$\frac{{}_1F_1(2, D+1, \kappa_{k,f})}{D {}_1F_1(1, D, \kappa_{k,f})} = \mathbf{w}_{k,f}^H \Phi_{\tilde{\mathbf{y}}\tilde{\mathbf{y}},k,f} \mathbf{w}_{k,f}, \quad (3.15)$$

where $\mathcal{P}(\cdot)$ calculates the principle component of the provided matrix.

The update equation for the concentration parameter $\kappa_{k,f}$ is implicit and an explicit update is not available. One can either rely on an approximation, e.g., for high concentration values [93, Equation 9] or evaluate the hypergeometric ratio for a certain number of points to obtain a lookup table. A cubic spline interpolation turned out to be sufficiently fast and accurate both in terms of initialization of the spline parameters as well as in terms of evaluation time.³ Furthermore, it is worth noting that the right hand side of Equation 3.15 is exactly the largest eigenvalue of the covariance matrix $\Phi_{\tilde{\mathbf{y}}\tilde{\mathbf{y}},k,f}$.

Complex spherical k-mode clustering This section describes a clustering algorithm for sound sources based on spatial cues. It is a simplified version of the complex Watson mixture model by enforcing shared concentration parameters for each class, i.e., $\kappa_{k,f} = \kappa > 0$ and a one-hot selection instead of the soft assignment during the E-step in a regular complex Watson mixture model maximum likelihood estimation [89].

Consequently, the E-step can be evaluated rather quickly:

$$\hat{c}_{k,t,f} = 1 \quad \text{for} \quad k = \underset{k'}{\operatorname{argmax}} |\tilde{\mathbf{y}}_{t,f}^H \mathbf{w}_{k',f}|^2. \quad (3.16)$$

During the M-step we just need to estimate the mode directions. The update coincides with Equation 3.14.

In terms of computational complexity, the complex spherical k-mode clustering is more expensive than k-means on complex observation vectors but especially the E-step is considerably faster than the E-step of the complex Watson mixture model. The contribution [89] evaluates the algorithm in comparison to k-means clustering and different complex Watson mixture model variants.

³ An implementation of the spline interpolation can be found in `pb_bss` whereas `libDirectional` uses a linear lookup table interpolation to obtain the concentration parameter.

3.2.1.5 Complex Bingham mixture model

Ito et al. proposed a complex Bingham mixture model to cluster sound sources in a multi-channel recording based on spatial cues [96]. The features are again normalized observation vectors as in Equation 3.8. The complex Bingham distribution is defined as follows [97, Equation 1]:

$$p(\tilde{\mathbf{y}}_{t,f} | c_{k,t,f}=1) = \mathcal{CB}(\tilde{\mathbf{y}}_{t,f}; \mathbf{B}_{k,f}) = \frac{1}{c_B(\mathbf{B}_{k,f})} e^{\tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f} \tilde{\mathbf{y}}_{t,f}}. \quad (3.17)$$

Ito et al. argue that the complex Watson distribution has limited expressiveness due to the reduction of the covariance matrix to a single mode direction [96, Section 3.3]: The complex Watson distribution is a special case of the complex Bingham distribution for the parameterization $\mathbf{B}_{k,f} = \kappa \mathbf{w}_{k,f} \mathbf{w}_{k,f}^H$.

An analytic expression for the normalization constant can be found by integrating $p(\tilde{\mathbf{y}}_{t,f} | \mathbf{c}_{t,f})$ over the complex unit hypersphere [97, Equation 2.3]:

$$\begin{aligned} c_B(\mathbf{B}_{k,f}) &= 4\pi {}_1F_1\left(\frac{1}{2}, \frac{3}{2}, \mathbf{B}_{k,f}\right) \\ &= 2\pi^D \sum_d a_{k,f,d} e^{\lambda_{k,f,d}} \quad \text{with} \quad a_{k,f,d}^{-1} = \prod_{d \neq d'} (\lambda_{k,f,d} - \lambda_{k,f,d'}), \end{aligned} \quad (3.18)$$

where $\lambda_{k,f,d}$ are the eigenvalues of the parameter matrix $\mathbf{B}_{k,f}$. The latter expression assumes that the eigenvalues are distinct. However, in many cases, some eigenvalues may coincide. It is an interesting finger exercise to derive these special cases via l'Hospital's rule. In practice, however, it is often sufficient to guarantee distinct eigenvalues by slightly modifying the eigenvalues. Moreover, the entire expression can be evaluated using a symbolic toolbox to generate functions depending on D .

It is worth noting that the normalization constant depends only on the eigenvalues of the parameter matrix. More precisely, a constant eigenvalue offset can be factored out of the normalization constant (compare Appendix A.1.1). Similarly, offsetting all eigenvalues with a constant does not change the shape of the distribution (compare Appendix A.1.2). Consequently, for numerical stability, the eigenvalues can be normalized such that the maximum eigenvalue is zero:

$$\lambda_{d'} = \lambda_d - \max_d \lambda_d. \quad (3.19)$$

Maximum likelihood estimates can again be obtained using the EM algorithm. The E-step turns out to be:

$$\ln \gamma_{k,t,f} = \ln \pi_{k,f} - \ln {}_1F_1\left(\frac{1}{2}, \frac{3}{2}, \mathbf{B}_{k,f}\right) + \tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f} \tilde{\mathbf{y}}_{t,f} + \text{const}. \quad (3.20)$$

During the M-step the parameters have to be estimated [97, Section 3]:

$$\pi_{k,f} = \frac{1}{T} \sum_t \gamma_{k,t,f}, \quad (3.21)$$

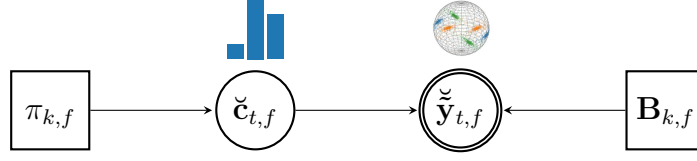


Figure 3.6: Graphical model of a complex Bingham mixture model (cBMM). The complex-valued distribution is visualized by its real-valued counterpart. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

$$\Phi_{\tilde{y}\tilde{y},k,f} = \sum_t \gamma_{k,t,f} \tilde{y}_{t,f} \tilde{y}_{t,f}^H / \sum_t \gamma_{k,t,f}, \quad (3.22)$$

$$\mathbf{B}_{k,f} = \mathbf{V}_{k,f}^H \Lambda_{k,f} \mathbf{V}_{k,f}, \quad (3.23)$$

where $\mathbf{V}_{k,f}$ coincides with the eigenvector matrix of $\Phi_{\tilde{y}\tilde{y},k,f}$ and the eigenvalues $\lambda_{k,f,d}$ have to be solved numerically.⁴ The following equation relates the eigenvalues $l_{k,f,d}$ of $\Phi_{\tilde{y}\tilde{y},k,f}$ to the eigenvalues $\lambda_{k,f,d}$ of $\mathbf{B}_{k,f}$:

$$\frac{\partial \ln c_B(\Lambda_{k,f})}{\partial \lambda_{k,f,d}} = l_{k,f,d}. \quad (3.24)$$

3.2.1.6 Full-Bayesian complex Watson mixture model

Drude et al. proposed a full-Bayesian complex Watson mixture model [17], [98]. This model is a generalization of the complex Watson mixture model in the sense that it treats the mixture weights as a random variable by introducing a Dirichlet prior and treats the mode vectors as a random variable by introducing a complex Bingham prior. The main advantage of this model is that it has the tendency to determine the number of speakers automatically. Mixture components which do not represent an audio source tend to have a mixture weight of almost zero. The EM update equations can be found in [98, Equation 8 – 14] and [17, Table 4.1]:

$$\ln \gamma_{k,t,f} = \mathbb{E}_{q(\pi_k)} \{ \ln \check{\pi}_k \} - {}_1F_1(1, D, \kappa_{k,f}) + \kappa_{k,f} \mathbb{E}_{q(\mathbf{w}_{k,f})} \{ \check{\mathbf{w}}_{k,f}^H \tilde{y}_{t,f} \tilde{y}_{t,f}^H \check{\mathbf{w}}_{k,f} \}, \quad (3.25)$$

$$N_{k,f} = \sum_t \gamma_{k,t,f}, \quad (3.26)$$

$$\alpha_{k,f} = \alpha_{0,k,f} + N_{k,f}, \quad (3.27)$$

$$\Phi_{\tilde{y}\tilde{y},k,f} = \frac{1}{N_{k,f}} \sum_t \gamma_{k,t,f} \tilde{y}_{t,f} \tilde{y}_{t,f}^H, \quad (3.28)$$

$$\mathbf{B}_{k,f} = \kappa_{k,f} N_{k,f} \Phi_{\tilde{y}\tilde{y},k,f} + \mathbf{B}_{0,k,f}, \quad (3.29)$$

$$\frac{{}_1F_1(2, D+1, \kappa_{k,f})}{D {}_1F_1(1, D, \kappa_{k,f})} = \mathbb{E}_{q(\mathbf{w}_{k,f})} \{ \check{\mathbf{w}}_{k,f}^H \Phi_{\tilde{y}\tilde{y},k,f} \check{\mathbf{w}}_{k,f} \}. \quad (3.30)$$

⁴ For example, by nonlinear least-squares fitting as in `libDirectional`.

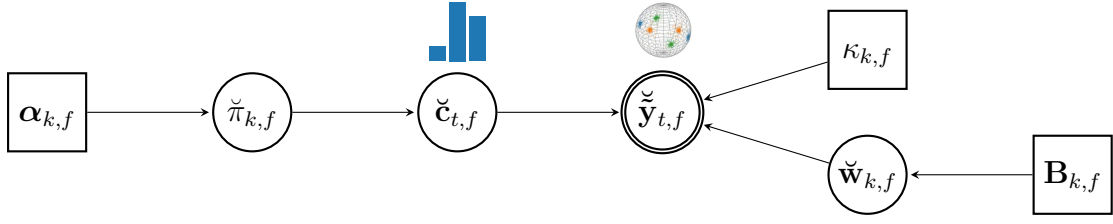


Figure 3.7: Graphical model of a full Bayesian cWMM. The complex-valued distribution is visualized by its real-valued counterpart. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

The expected value $\mathbb{E}_{q(\pi_k)} \{\ln \check{\pi}_k\}$, which is the negative entropy $-H(q(\pi_k))$ is evaluated using the digamma function $\psi(\cdot)$ [11, Equation B.21], [17, Equation 4.40]:⁵

$$\mathbb{E}_{q(\pi_k)} \{\ln \check{\pi}_k\} = \psi(\alpha_{k,f}) - \psi\left(\sum_{k'} \alpha_{k',f}\right). \quad (3.31)$$

Calculating the remaining expected values is a bit more convoluted and corresponding equations can be found in [98, Appendix] or [17, Section 4.2].⁶

3.2.1.7 Time-variant complex Gaussian mixture model

Different authors have introduced a time-variant complex Gaussian mixture model (TV-cGMM) to separate sound sources based on spatial cues. Févotte and Cardoso used a factorized covariance matrix for the source images [100]. Vincent and Gribonval introduced it as a *local Gaussian* model. They argue that this model is suited for speech mixtures due to the local stationarity but global sparseness of speech signals in the STFT domain [101, Section 3]. Although [102] already presented an integration of a TV-cGMM with an NMF-based source model more widespread use can probably be attributed to [73] due to its more accessible notation and [103] due to its successful application in context of the CHiME 3 challenge [104]. The TV-cGMM is related to the full-rank model by Duong, Vincent, and Gribonval [86] but assumes sparse mixing instead of additive mixing.

The TV-cGMM directly uses the observation vectors $\mathbf{y}_{t,f}$ as in Equation 3.7. The distribution of the mixture is a marginalization over the latent class affiliations:

$$p(\mathbf{y}_{t,f}) = \sum_k p(c_{k,t,f}=1)p(\mathbf{y}_{t,f}|c_{k,t,f}=1), \quad (3.32)$$

⁵ The digamma function is defined as the derivative of the logarithm of the Gamma function [99, Page 258]:

$$\psi(x) = \frac{d}{dx} \ln(\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)}$$

⁶ A numerical implementation is publicly available in `libDirectional`.

where the class labels are categorically distributed and the class-conditional observation distribution is a time-variant complex Gaussian distribution:

$$p(\mathbf{y}_{t,f} | c_{k,t,f}=1) = \mathcal{CN}(0, \sigma_{k,t,f} \mathbf{B}_{k,f}) = \frac{1}{\det(\pi \sigma_{k,t,f} \mathbf{B}_{k,f})} e^{-\mathbf{y}_{t,f}^H \sigma_{k,t,f}^{-1} \mathbf{B}_{k,f}^{-1} \mathbf{y}_{t,f}}. \quad (3.33)$$

It differs from the widely-used parameterization of a complex circularly symmetric Gaussian [105, Theoreme 3.1] only by the factorization of the covariance matrix.

During the E-step the posterior distribution of the class labels is given by

$$\ln \gamma_{k,t,f} = \ln \pi_k - \ln \det(\pi \sigma_{k,t,f} \mathbf{B}_{k,f}) - \mathbf{y}_{t,f}^H \sigma_{k,t,f}^{-1} \mathbf{B}_{k,f}^{-1} \mathbf{y}_{t,f} + \text{const}. \quad (3.34)$$

During the M-step the mixture weights, the time-dependent scalar as well as the time-independent matrix are updated using

$$\pi_{k,f} = \frac{N_{k,f}}{T} \quad \text{with} \quad N_{k,f} = \sum_t \gamma_{k,t,f}, \quad (3.35)$$

$$\sigma_{k,t,f} = \frac{1}{D} \mathbf{y}_{t,f}^H \mathbf{B}_{k,f}^{-1} \mathbf{y}_{t,f}, \quad (3.36)$$

$$\mathbf{B}_{k,f} = \frac{1}{N_{k,f}} \sum_t \gamma_{k,t,f} \frac{\mathbf{y}_{t,f} \mathbf{y}_{t,f}^H}{\sigma_{k,t,f}}. \quad (3.37)$$

It is worth noting that the M-step updates are dependent on each other. Although this may require a nonlinear solver, in practice, it turned out to be sufficient to update each parameter only once per M-step and initialize, e.g., the time-dependent scalar $\sigma_{k,t,f}$ with ones. Moreover, the correlation matrix inverse can be obtained via an eigenvalue decomposition thereby clipping the eigenvalues for additional stability (see [106] and `pb_bss` for details).

3.2.1.8 Complex angular central Gaussian mixture model

The real-valued angular central Gaussian distribution was introduced by Tyler [107] as an alternative to the Bingham distribution due to the rather complicated normalization constant and maximum likelihood estimation for the Bingham parameter matrix [108, Page 182]. To illustrate the real-valued angular central Gaussian distribution Figure 3.8 shows samples for different parameterizations. The maximum likelihood parameter estimates for a real-valued angular central Gaussian can be found in [107].

The complex angular central Gaussian mixture model (cACGMM) is a probabilistic spatial mixture model that can be used to separate sound sources in a multi-channel recording. The probabilistic dependencies are illustrated in Figure 3.9. It was proposed by Ito et al. in 2016 [111] with a proof that the update equations coincide with the update equations of the TV-cGMM [111, Appendix]. Consequently, the EM algorithms for each are identical. A similar proof is that when $\mathbf{y}_{t,f}$ is TV-cGMM distributed, the normalized features $\tilde{\mathbf{y}}_{t,f}$ are cACGMM distributed.

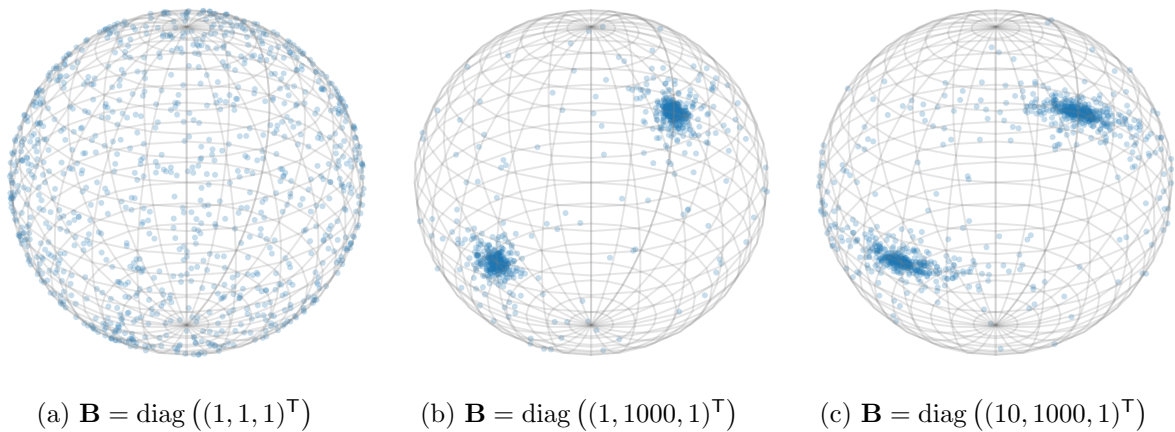


Figure 3.8: Samples from a real-valued angular central Gaussian distribution with different parameters. The first parameter set yields a uniform distribution, the second parameter set yields a symmetric distribution around the mode axis and the third parameter set results in an elliptically symmetric distribution around the mode axis. Visualizations in [109] and [110] inspired this figure.

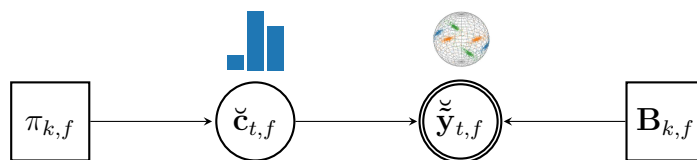


Figure 3.9: Graphical model of a cACGMM. The complex-valued distribution is visualized by its real-valued counterpart. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

It operates on normalized observation vectors defined on the complex unit hypersphere \mathbb{CS}^{D-1} and, therefore, can only capture IPDs and ILDs:

$$\tilde{\mathbf{y}}_{t,f} = \mathbf{y}_{t,f} / \|\mathbf{y}_{t,f}\| \in \mathbb{CS}^{D-1}. \quad (3.38)$$

The class conditional distribution (observation model) is a complex angular central Gaussian distribution [112, Equation 3.1], for which all class-dependent parameters are summarized in the correlation matrix $\mathbf{B}_{k,f}$:

$$p(\tilde{\mathbf{y}}_{t,f} | c_{k,t,f}=1) = \text{cACG}(\tilde{\mathbf{y}}_{t,f}; \mathbf{B}_{k,f}) = \frac{(D-1)!}{2\pi^D \det \mathbf{B}_{k,f}} \frac{1}{(\tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f}^{-1} \tilde{\mathbf{y}}_{t,f})^D}. \quad (3.39)$$

This parametric form can be obtained by applying a random variable transformation, in this case $\mathbf{y}/\|\mathbf{y}\|$, to a multi-variate complex circularly-symmetric Gaussian random variable \mathbf{y} [110, Page 12f].

The maximum likelihood estimates of the parameters of a cACGMM can be determined by applying the EM algorithm. The posterior probabilities in the E-step are then obtained

via

$$\ln \gamma_{k,t,f} = \ln \pi_k - \ln \det \mathbf{B}_{k,f} - D \ln (\tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f}^{-1} \tilde{\mathbf{y}}_{t,f}) + \text{const.} \quad (3.40)$$

The update equations in the M-step result from calculating the derivative of the auxiliary function with respect to all parameters. While this does not lead to a closed form solution, the parameter updates are then given by [111, Equation 13 and 14], for which [110] provides a step-by-step derivation of the updates:

$$\pi_{k,f} = \frac{N_{k,f}}{T} \quad \text{with} \quad N_{k,f} = \sum_t \gamma_{k,t,f}, \quad (3.41)$$

$$\mathbf{B}_{k,f} = \frac{D}{N_{k,f}} \sum_t \gamma_{k,t,f} \frac{\tilde{\mathbf{y}}_{t,f}^H \tilde{\mathbf{y}}_{t,f}}{\tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f}^{-1} \tilde{\mathbf{y}}_{t,f}}. \quad (3.42)$$

3.2.1.9 Guided source separation

Although the previous models clearly classify as blind source separation approaches, often times external knowledge is available. We may have a priori knowledge about the array geometry or may have a previous algorithm estimating voice activity detection (VAD) information. This kind of side information either obtained through some oracle or estimated from the signal can guide the source separation algorithm and significantly influence its performance [113].

An obligatory example for side information is saliency maps indicating which observations should be trusted more [114, Page 127], [17, Equation 6.3]. It is typical for many speech mixture databases that the mixture recording is already cut in such a way that the beginning contains a few noise frames as well as that the mixture ends with a few noise frames. The authors of the CHiME 5 challenge went a bit further and provided start and stop timings for each speaker. This can be exploited by influencing the separation process with soft weights [115] or even forcing the mixture weights of each speaker to zero when the annotation claims that the speaker should indeed be inactive [116, Equation 5]:

$$\pi_{k,t,f} = \pi_{k,f} a_{k,t} \bigg/ \sum_{k'} \pi_{k',f} a_{k',t}, \quad (3.43)$$

where $a_{k,t}$ encodes if speaker k is active during time frame t , e.g., it is one when the speaker is active and zero otherwise. The noise class can be assumed to be active in all time frames. Consequently, a minimum value for $a_{k,t}$ is not required.

This is a great example of external guidance resulting in almost completely avoiding the permutation problem (see Section 3.2.1.1) and boosting the performance of a cACGMM further. Its efficacy was shown with promising WERs on the CHiME 5 dataset [116], [117]. Further, it is worth noting that many integration approaches better covered in Chapter 4 can be interpreted as guided source separation since the spectral cues guide the spatial model.

Clearly, the interpretability of probabilistic spatial mixture models helps to integrate external side-channel information. It is much less clear how to integrate, e.g., time annotations in a pretrained neural network-based source separator.

3.2.2 Spatial features for neural networks

It is worth noting that a very valid alternative to probabilistic spatial models is to use spatial features for neural network-based source separation. In that case, the training encourages DNN to understand and use spatial diversity for source separation. Multi-channel deep clustering [49] in particular demonstrated that even simple IPD features can boost the source separation performance quite a bit over single-channel approaches.

Typical spatial features employed in the context of neural network-based speech enhancement are the sine and cosine of IPDs as used in [49]:

$$\cos\text{IPD}_{t,f,d,d'} = \cos(\arg y_{t,f,d} - \arg y_{t,f,d'}) = \cos \arg(y_{t,f,d'}^* y_{t,f,d}), \quad (3.44)$$

$$\sin\text{IPD}_{t,f,d,d'} = \sin(\arg y_{t,f,d} - \arg y_{t,f,d'}) = \sin \arg(y_{t,f,d'}^* y_{t,f,d}). \quad (3.45)$$

An alternative is to use generalized cross-correlation (GCC) features obtained by comparing the actual complex vectors to precalculated steering vectors obtained according to the array geometry [49, Equation 5], [55]. A comparison of proposed systems and baseline systems with and without spatial features can be found in Section 5.7.3.

3.3 Principles of source extraction

Source extraction refers to techniques developed to extract a single speech source. We here constrain ourselves to spectral masking and spatial filtering in the frequency domain. A common factor of both approaches presented here is that they rely on a previously calculated mask possibly stemming from a blind source separation algorithm.

3.3.1 Spectral subtraction/ masking

Spectral subtraction was originally developed as a noise reduction scheme [118]. Already in Boll's 1979 formulation an STFT estimate is obtained as the product of a mask and the observation STFT [118, Section III.F]:

$$\hat{x}_{k,t,f} = \gamma_{k,t,f} y_{t,f,d}, \quad (3.46)$$

where d identifies an arbitrarily selected reference channel, e.g., $d = 1$. Although the original work relied on a first estimate of the distortion power, we here just adopt the idea to extract a mask first, instead of designing a system which directly outputs the speech spectrogram.

Barker et al. suggested already early on to limit the range of the mask, e.g., to $[0, 1]$ by employing a squashing function [119]. In 2004 Seltzer et al. used a Bayesian approach to generate masks in the form of posterior probabilities grounded on a set of hand-crafted features laying the foundations for data-driven approaches [120].

Interestingly, the idea to generate masks first has survived the neural revolution and neural network-based speech enhancement systems often perform better when they are tasked to

provide a mask instead of directly providing the speech spectrogram. This can mainly be addressed to the limited dynamic range the neural network has to produce when estimating masks only.

To train a system with the goal of predicting a mask suitable to extract one or more sources, one needs to rely – in almost all cases – on a supervision mask which is obtained, e.g., from the oracle source signal and the oracle noise signal during training. For further reading, Erdogan et al. present a structured overview of different ideal masks as targets for a learning-based approach to mask estimation [121] and argue, why masking is likely to work better than directly predicting the spectrogram.

3.3.2 Spatial filtering/ beamforming

Spatial filtering is a widely-used technique to extract a desired signal by combining the different channels of a multi-channel observation either using linear or nonlinear approaches with its origins mainly in radar, radio astronomy, sonar, communications, and seismology (see, e.g., [122, Section 1.2] or [123, Section 1.2] for some historic insights). Linear spatial filtering, namely beamforming, extracts a particular source signal by linearly filtering different microphones (channels) so that the desired parts of a mixture signal positively interfere while undesired parts cancel out [2], [124]. In the STFT domain this results in complex-valued linear filter vectors $\mathbf{w}_{k,f} = (w_{k,1,f}, \dots, w_{k,D,f})^\top$ which lead to a speech estimate by calculating an inner product of the beamforming vector with the observation vector:

$$\hat{x}_{k,t,f} = \mathbf{w}_{k,f}^H \mathbf{y}_{t,f}. \quad (3.47)$$

The term *beamforming* originally stems from the geometric interpretation in which a beam was steered towards a particular direction of arrival (DoA) [2]. However, a beamformer may exploit early reflections or in some other way contain values that extract sources but does not have a geometric counterpart. This even allows us to separate sources with equal DoAs but different distances to the array [125, Section 5].

Beamformer design criteria can be categorized into fixed beamforming and data-dependent beamforming.⁷ Fixed beamforming relies on a priori knowledge of the DoA or even RTFs and requires knowledge of the exact sensor geometry and channel gain. Fixed beamforming with, e.g., superdirective beamformers [126] can be used for source separation systems, e.g., by cleverly switching between output channels of a set of fixed beamforming vectors [67]. In contrast, data-dependent beamformers rely on statistics obtained from the current observation, e.g., second-order statistics for the sources and noise [2, Section V]. Therefore, they avoid the need for exact knowledge of the sensor array geometry as well as the need for additional gain adjustment of each channel [127, Page 56].

The remainder of this section introduces how the necessary second-order statistics, namely the spatial covariance matrices, can be obtained. Moreover, it introduces a limited number of beamforming approaches suitable to extract a speaker from a speech mixture.

⁷ Data-dependent beamforming is often named *adaptive* beamforming. This term is avoided here since it does not clearly differentiate between offline data-dependent beamforming and online data-dependent beamforming.

3.3.2.1 Spatial covariance matrix estimation

Each element of the observation vector is assumed to be a circularly symmetric zero-mean complex random variable.⁸ Relevant second-order statistics which can be extracted from the observed mixture are the observation spatial covariance matrix $\Phi_{\mathbf{y}\mathbf{y},f}$, the target covariance matrix $\Phi_{\mathbf{x}\mathbf{x},k,f}$ and the interference covariance matrix $\Phi_{\mathbf{n}\mathbf{n},k,f}$. The observation spatial covariance matrix $\Phi_{\mathbf{y}\mathbf{y},f}$ can be obtained by approximating the expected value with a time average implicitly assuming at least wide-sense stationarity:

$$\Phi_{\mathbf{y}\mathbf{y},f} = \mathbb{E} \{ \check{\mathbf{y}}_{t,f} \check{\mathbf{y}}_{t,f}^H \} \approx \frac{1}{T} \sum_t \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H. \quad (3.48)$$

Different authors have proposed a variety of variants to estimate the target covariance matrix for each source. One variant provides an estimate of the spatial covariance matrix normalized to the entire utterance and ensures that the target and interference spatial covariance matrices add up to the observation spatial covariance matrix as long as the masks $\gamma_{1,t,f}, \dots, \gamma_{K,t,f}$ sum up to one. The covariance matrix estimate is then obtained with:

$$\Phi_{\mathbf{x}\mathbf{x},k,f} = \mathbb{E} \{ \check{\mathbf{x}}_{t,f} \check{\mathbf{x}}_{t,f}^H \} \approx \frac{1}{T} \sum_t \gamma_{k,t,f} \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H. \quad (3.49)$$

An alternative variant – which is closer to how covariance matrices in an EM algorithm for, e.g., GMMs are calculated – scales up the spatial covariance matrix as if the spatial covariance matrix is estimated only on the active bins:

$$\Phi_{\mathbf{x}\mathbf{x},k,f} = \mathbb{E} \{ \check{\mathbf{x}}_{t,f} \check{\mathbf{x}}_{t,f}^H \} \approx \sum_t \gamma_{k,t,f} \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H / \sum_t \gamma_{k,t,f}. \quad (3.50)$$

In [128, Equation 13] and [129, Equation 3] we propose no normalization altogether and argue that some beamforming design criteria are independent of the scale of the covariance matrix. However, in retrospect, one has to state more carefully, how the particular implementation at hand reacts to differently scaled matrices.

Additionally, one may argue that the noise is not that sparse and an additional subtraction of the interference spatial covariance matrix from the target spatial covariance matrix is needed [130, Equation 7]. This, however, would require additional measures to avoid negative eigenvalues. In other words, it validates the expectation that all spatial covariance matrices are Hermitian and positive semi-definite.

Another variant is to estimate a masked signal first and then feed this into the spatial covariance matrix estimation [131]. This effectively results in the mask $\gamma_{k,t,f}$ being replaced by the quadratic term $\gamma_{k,t,f}^2$:

$$\Phi_{\mathbf{x}\mathbf{x},k,f} = \mathbb{E} \{ \check{\mathbf{x}}_{t,f} \check{\mathbf{x}}_{t,f}^H \} \approx \frac{1}{T} \sum_t (\gamma_{k,t,f} \mathbf{y}_{t,f}) (\gamma_{k,t,f} \mathbf{y}_{t,f})^H = \frac{1}{T} \sum_t \gamma_{k,t,f}^2 \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H. \quad (3.51)$$

⁸ A complex random variable \check{z} is circularly symmetric when $p_{\check{z}}(z) = p_{\check{z}}(e^{j\varphi} z)$.

The interference covariance matrix, which in general may represent interfering speakers as well as the noise signal, can be calculated similarly. In correspondence to Equation 3.49, it can be obtained as follows:

$$\begin{aligned}\Phi_{\mathbf{nn},k,f} &= \mathbb{E} \{ \check{\mathbf{n}}_{t,f} \check{\mathbf{n}}_{t,f}^H \} \approx \frac{1}{T} \sum_t \left(\sum_{k',k' \neq k} \gamma_{k',t,f} \right) \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H \\ &= \sum_k \gamma_{k,t,f} = 1 \frac{1}{T} \sum_t (1 - \gamma_{k,t,f}) \mathbf{y}_{t,f} \mathbf{y}_{t,f}^H,\end{aligned}\quad (3.52)$$

which assumes that the masks sum up to one. The formulation here does not necessarily require one of the masks representing the noise signal.

Even though, in this case, we constrain ourselves to offline processing, it can be beneficial to allow a time-dependent interference covariance matrix. For example Kubo et al. suggested changing the interference covariance matrix depending on how likely the interfering speakers are active [132]. It may also be beneficial in an offline setup when the scenario geometry is likely to change, e.g., the speakers might move in a sufficiently long recording.

All results reported in Chapter 5 were obtained with masks normalized as in Equation 3.50 to imitate the covariance matrices used in mixture models.

3.3.2.2 MaxSNR/GEV

MaxSNR beamforming, also called generalized eigenvalue (GEV) beamforming, is an instance of statistically optimal data-dependent beamforming [133]. The two names stem from the fact that MaxSNR has long been seen as inappropriate for speech processing since the narrowband signal to noise ratio (SNR) maximization may introduce arbitrary signal distortions [134, Page 2]. However, this did not stop Warsitz et al. to introduce it to the speech community (under the name GEV beamforming) anyway [134], alleviating the effect of distortions using a blind analytic normalization (BAN) postfilter (see Section 3.3.2.6). In our subsequent research, the terms GEV beamforming and MaxSNR beamforming have been used synonymously. However, to avoid additional confusion, GEV is the preferred term throughout the remainder of this thesis.

In the framework of GEV beamforming, the optimal filter coefficients are obtained by maximizing the expected output SNR after applying a beamforming vector \mathbf{w}_f to the observation signal:

$$\mathbf{w}_{k,f} = \operatorname{argmax}_{\mathbf{w}_{k,f}} \frac{\mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{x}}_{t,f} \right|^2 \right\}}{\mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{n}}_{t,f} \right|^2 \right\}} = \operatorname{argmax}_{\mathbf{w}_{k,f}} \frac{\mathbf{w}_{k,f}^H \Phi_{\mathbf{xx},k,f} \mathbf{w}_{k,f}}{\mathbf{w}_{k,f}^H \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f}}. \quad (3.53)$$

The latter term is a generalized Rayleigh quotient.⁹ The maximization can be addressed either with a constrained optimization problem (see Section A.5.1) or as an unconstrained

⁹ The Rayleigh quotient is defined as $\mathbf{x}^H \mathbf{A} \mathbf{x} / \mathbf{x}^H \mathbf{x}$ [135, Section 4.2]. Therefore, the additional matrix in the denominator is a generalization.

optimization problem (see Section A.5.2). Either way, one obtains the optimal coefficients with a generalized eigenvalue decomposition:

$$\Phi_{\mathbf{xx},k,f} \mathbf{w}_{k,f} = \lambda_{k,f} \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f} \Rightarrow \mathbf{w}_{k,f} = \mathcal{P}\{\Phi_{\mathbf{nn},k,f}^{-1} \Phi_{\mathbf{xx},k,f}\}, \quad (3.54)$$

where $\mathcal{P}\{\cdot\}$ extracts the principal component (the eigenvector corresponding to the largest eigenvalue) with a yet undefined scale.

However, it is worth noting that an explicit inverse of the interference covariance matrix is not necessary since some generalized eigenvalue decomposition algorithms avoid this step entirely, e.g., one may apply spatial whitening first and then extract the principal component of the resulting covariance matrix as detailed in [136, Section A.5]. The thesis [137] contains a detailed analysis of the effect of the exact algorithm on noise reduction performance.

Another crucial point is that the aforementioned eigenvalue decomposition does not constrain the scale of the beamforming vector altogether. Multiplying the solution with a complex-valued scalar does not change its validity:

$$\Phi_{\mathbf{xx},k,f} \mathbf{w}_{k,f} = \lambda_{k,f} \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f} \Leftrightarrow \Phi_{\mathbf{xx},k,f} (c \mathbf{w}_{k,f}) = \lambda_{k,f} \Phi_{\mathbf{nn},k,f} (c \mathbf{w}_{k,f}) \text{ with } c \in \mathbb{C}.$$

Some implementations normalize each eigenvector to unit length. In the context of this work, I rely on an implementation that obtains the principal eigenvector by first performing a Cholesky decomposition of the interference covariance matrix. Then, the beamforming vector is obtained as the product of the inverse of such a decomposition and the principal component of the target covariance matrix. This coincides with the aforementioned constraint $\mathbf{w}_{k,f}^H \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f} = 1$ but interestingly, this contradicts with [133, Equation 61.16]. See Section 3.3.2.6 for further normalization concepts.

The GEV beamformer has proven to be quite robust with respect to numerical instabilities in comparison to a MVDR beamformer, e.g., [129, Figure 1] demonstrates that the GEV beamformer performance suffers less from a higher condition number of the interference spatial covariance matrix.

3.3.2.3 MVDR

The MVDR beamformer, also known as the Capon beamformer, is designed to minimize the output variance [138]. To avoid the trivial solution $\mathbf{w}_{k,f} = \mathbf{0}$ the optimization is performed under the constraint that the signal from a particular look direction has unity gain [138]:

$$\mathbf{w}_{k,f} = \underset{\mathbf{w}_{k,f}}{\operatorname{argmin}} \mathbf{w}_{k,f}^H \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f} \quad \text{s.t.} \quad \mathbf{w}_{k,f}^H \mathbf{d}_{k,f} = 1. \quad (3.55)$$

Assuming the intermediate vector $\mathbf{d}_{k,f}$ is given, this constrained optimization problem can again be solved using Lagrange multipliers (see Section A.6). The well-known solution turns out to be just a slight adjustment of the intermediate vector $\mathbf{d}_{k,f}$:

$$\mathbf{w}_{k,f} = \frac{\Phi_{\mathbf{nn},k,f}^{-1} \mathbf{d}_{k,f}}{\mathbf{d}_{k,f}^H \Phi_{\mathbf{nn},k,f}^{-1} \mathbf{d}_{k,f}}. \quad (3.56)$$

It has already become apparent that the precision of the intermediate vector $\mathbf{d}_{k,f}$ greatly impacts performance. Traditionally, the intermediate vector can be obtained by using DoA information and a known array geometry (then rightfully called steering vector). However, to allow blind beamforming – purely data-dependent beamforming – the intermediate vector should be calculated from statistics obtained on the audio signal itself (since it then captures the (possibly scaled) RTF it is called RTF vector occasionally [139], [140]). One alternative is to identify the intermediate vector as the principal component of the speech spatial covariance matrix obtained with Equation 3.49, Equation 3.50, or Equation 3.51, e.g., via

$$\mathbf{d}_{k,f} = \mathcal{P}\{\Phi_{\mathbf{xx},k,f}\}. \quad (3.57)$$

Another alternative is to extract the intermediate vector $\mathbf{d}_{k,f}$ according to the GEV beamforming criterion just as in Equation 3.54 as was proposed by Araki et al. in the context of online meeting recognition [141, Equation 5]. Table 5.23 in the evaluation section compares different intermediate vector estimation variants and lists references with additional details.

Alternative MVDR formulation (Souden-MVDR) To avoid an estimation of an intermediate beamforming vector altogether, Souden et al. proposed a reformulation which depends directly on the second-order statistics and an arbitrarily chosen reference channel [142, Equation 14] and, therefore, avoids the rank-one assumption for $\Phi_{\mathbf{xx},k,f}$:

$$\mathbf{w}_{k,f} = \mathbf{w}_{k,f}(\mathbf{u}_k) = \frac{\Phi_{\mathbf{nn},k,f}^{-1} \Phi_{\mathbf{xx},k,f} \mathbf{u}_k}{\text{tr}\{\Phi_{\mathbf{nn},k,f}^{-1} \Phi_{\mathbf{xx},k,f}\}}, \quad (3.58)$$

where \mathbf{u}_k is a one-hot vector indicating the (possibly class-dependent) reference channel. This alternative MVDR formulation will be called *Souden-MVDR* in the following.

To avoid the rather arbitrary choice of the reference channel and to further improve performance Erdogan et al. suggested to select the reference channel depending on the average expected output SNR [131, Page 3]:

$$\mathbf{u}_k = \underset{\mathbf{u}_k}{\text{argmax}} \frac{\sum_f \mathbf{w}_{k,f}(\mathbf{u}_k)^H \Phi_{\mathbf{xx},k,f} \mathbf{w}_{k,f}(\mathbf{u}_k)}{\sum_f \mathbf{w}_{k,f}(\mathbf{u}_k)^H \Phi_{\mathbf{nn},k,f} \mathbf{w}_{k,f}(\mathbf{u}_k)}. \quad (3.59)$$

Although in the absence of estimation errors, the MVDR result should already be distortionless it is occasionally still beneficial to apply a BAN postfilter [143, Page 2].

3.3.2.4 Linearly constrained minimum variance beamformer

The linearly constrained minimum variance (LCMV) beamformer design criterion was introduced by Frost [144] as a means to enforce one or more constraints on the beamforming filter coefficients. In that sense, it can be seen as a generalization of MVDR beamforming. It was

originally derived in the time-domain assuming that the source location and array geometry are known. The optimization criterion is again applied to minimize the output power or output variance subject to a set of linear constraints compactly written as a vector-valued constraint [144, Equation 16]:

$$\mathbf{w}_{k,f} = \underset{\mathbf{w}_{k,f}}{\operatorname{argmin}} \mathbf{w}_{k,f}^H \boldsymbol{\Phi}_{\mathbf{nn},k,f} \mathbf{w}_{k,f} \quad \text{s.t.} \quad \mathbf{C}_{k,f}^H \mathbf{w}_{k,f} = \mathbf{g}_{k,f}, \quad (3.60)$$

where the right hand side vector $\mathbf{g}_{k,f}$ defines if a source is suppressed or emphasized. Typical linear constraints are either distortionless constraints for one or more speakers or spatial zeros (or small epsilon values in $\mathbf{g}_{k,f}$) to suppress point source interferences. The number of constraints is limited by the number of microphone channels D . It is worth mentioning that there are scenarios in which a controlled attenuation of an interference speaker is desired, e.g., Aroudi et al. suggest to not suppress the interfering speaker entirely to still allow auditory attention switching for cognitive-driven hearing aids [145].

Again with the help of Lagrange multipliers the optimal filter coefficients are obtained:

$$\mathbf{w}_{k,f} = \boldsymbol{\Phi}_{\mathbf{nn},k,f}^{-1} \mathbf{C}_{k,f} (\mathbf{C}_{k,f}^H \boldsymbol{\Phi}_{\mathbf{nn},k,f} \mathbf{C}_{k,f})^{-1} \mathbf{g}_{k,f}. \quad (3.61)$$

Although the LCMV is used rather rarely in combination with a DNN or DNN-based mask estimator Chazan et al. successfully employ a neural network-based LCMV beamformer [146].

3.3.2.5 Weighted multi-channel Wiener filter

The multi-channel Wiener filter (MWF) is an optimal filtering approach in the sense that it minimizes the mean squared error between the estimated signal and the desired signal at a reference microphone [147, Equation 8], [140, Equation 4–5]:

$$\begin{aligned} \mathbf{w}_{k,f} &= \underset{\mathbf{w}_{k,f}}{\operatorname{argmin}} \mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{y}}_{t,f} - \check{x}_{k,d,t,f} \right|^2 \right\} \\ &= \underset{\mathbf{w}_{k,f}}{\operatorname{argmin}} \left(\mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{x}}_{k,t,f} - \check{x}_{k,d,t,f} \right|^2 \right\} + \mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{n}}_{k,t,f} \right|^2 \right\} \right), \end{aligned} \quad (3.62)$$

where $\check{\mathbf{n}}_{k,t,f}$ contains all but the target speaker.

Following [140, Equation 6] a distortion weight μ can be introduced to control the influence of both terms. Higher values result in better noise suppression at the cost of more speech distortion. The MWF is obtained for $\mu = 1$. The optimization problem is now given as follows [148]:

$$\mathbf{w}_{k,f} = \underset{\mathbf{w}_{k,f}}{\operatorname{argmin}} \left(\mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{x}}_{k,t,f} - \check{x}_{k,d,t,f} \right|^2 \right\} + \mu \mathbb{E} \left\{ \left| \mathbf{w}_{k,f}^H \check{\mathbf{n}}_{k,t,f} \right|^2 \right\} \right). \quad (3.63)$$

The resulting beamformer is then called speech distortion-weighted MWF or, more concisely, weighted multi-channel Wiener filter (WMWF). The optimal solution can be found by

differentiating the optimization criterion with respect to $\mathbf{w}_{k,f}$ and setting the result equal to zero [140, Equation 7], [148]:

$$\mathbf{w}_{k,f} = (\Phi_{\mathbf{x}\mathbf{x},k,f} + \mu\Phi_{\mathbf{n}\mathbf{n},k,f})^{-1} \Phi_{\mathbf{x}\mathbf{x},k,f} \mathbf{u}_k, \quad (3.64)$$

where again \mathbf{u}_k is a one-hot vector indicating an arbitrary reference channel. Similarly to the Souden-MVDR mentioned above, the WMWF does not require an explicit RTF vector calculation. However, the optimal reference channel can either be fixed or estimated based on expected output SNR (compare Equation 3.59).

3.3.2.6 Magnitude and phase normalization of beamforming vectors

Instead of arbitrarily scaling the beamforming vectors, one may resort to an analytically-motivated normalization. Warsitz et al. proposed a BAN postfilter to compensate the distortion introduced by GEV beamforming which relies on the already obtained second-order statistics of the noise signal [134, Equation 17]:

$$g_{k,f} = \frac{\sqrt{\mathbf{w}_{k,f}^H \Phi_{\mathbf{n}\mathbf{n},k,f} \Phi_{\mathbf{n}\mathbf{n},k,f} \mathbf{w}_{k,f} / D}}{\mathbf{w}_{k,f}^H \Phi_{\mathbf{n}\mathbf{n},k,f} \mathbf{w}_{k,f}}. \quad (3.65)$$

Although this does not solve the issue of an arbitrary absolute phase $e^{j\phi}$, it reduces the frequency-dependent distortions effectively. One way to solve this is to enforce a zero-phase-sum constraint [81, Equation 2.13]. An alternative is to arbitrarily set the phase of a reference channel to zero [149, Equation 21] or to minimize the group delay introduced by the filter [149, Equation 22]. For a comparison of phase corrections, please again refer to [137].

3.3.3 Combination of beamforming and masking

What can be gained by additional postfiltering if the [...] beamformer already provides the optimum solution for a given sound field? [150]

First of all, the aforementioned beamformers are only optimal for the narrowband they were derived on. In a wideband signal such as speech, it is not guaranteed that they provide the optimal linear spatial filter, e.g., in an SNR sense [150]. Moreover, beamforming is constrained to linear filtering. To allow arbitrary changes to the signal while still leveraging beamforming it is not far-fetched to use beamforming and masking together. Since masking is a single-channel approach, masking can be used as a single-channel postfilter (compare, e.g., [151, Equation 9]):

$$\hat{x}_{k,t,f} = g_{k,t,f} \mathbf{w}_f^H \mathbf{y}_{t,f}, \quad (3.66)$$

where $g_{k,t,f}$ is an arbitrary mask, potentially the posterior mask $\gamma_{k,t,f}$ or a toned-down variant by introducing a minimum gain G_{\min} to retain the noise naturalness during speech absence [152, Page 879]:

$$g_{k,t,f} = \max(\gamma_{k,t,f}, G_{\min}). \quad (3.67)$$

The combination of beamforming and masking was successfully employed in a system proposed for front-end processing for the CHiME 5 challenge [116].

4 Integration of neural networks and probabilistic graphical models

Integration of neural networks and probabilistic graphical models for BSS, as discussed in this chapter, was first proposed in [106]. Although, in general, integration is quite vaguely defined, we here want to focus on the integration of neural network-based models to primarily capture spectral cues and probabilistic models to primarily capture spatial cues. We aim to carefully distinguish cascade approaches from a tight integration and want to briefly mention similarities to ensemble methods.

Integration approaches, in the sense of this work, constitute systems which either address different sub-problems in a single formulation or combine diverse knowledge sources to obtain a single estimate for a given problem. In the chapter at hand source separation systems are addressed which (1) rely on two distinct sources of information, namely spatial and spectral cues and (2) combine a neural network-based algorithm with a probabilistic model-based algorithm. To this effect, the approaches at hand can be seen as instances of information fusion as well as instances of ensemble methods. However, in ensemble methods, systems are typically run independently and the final prediction is obtained through, e.g., majority voting [153], averaging [153], or more advanced domain-specific ensemble methods such as ROVER [154], [155]. In contrast, the integration approaches discussed in Section 4.3 estimate parameters relevant for both input modalities during the prediction step jointly. With this in mind, the discussed integration cannot be seen as an ensemble method realized as a post-processing step.

Integrating different cues helps to improve overall system performance. Ideally, they degrade due to different causes and therefore can lead to meaningful results even when one cue is unreliable. In the work at hand, the aim is to integrate spatial and spectral cues. When speakers are moving slightly or speakers are located close to each other, the spatial cues might be less reliable but the observed spectral cues might very well still lead to sufficient separation results. Vice versa, even in very noisy conditions spatial cues can still be sufficient as demonstrated, for instance, on the CHiME 5 database.

The advantage of combining probabilistic models and neural networks stems from their complementary properties. On the one hand, probabilistic models do not suffer from overfitting to a particular scene or set of speakers when all parameters are estimated on the test utterance. On the other hand, neural networks trained with supervision signals can profit from a vast amount of training data and can model fine-grained dependencies that have otherwise been hard for humans to encode in rules or exact probabilistic models.

The remainder of this chapter is organized as follows: Section 4.1 is a short literature review of existing integration variants. Section 4.2 introduces a cascade approach to integrate spatial and spectral features. Section 4.3 discusses integration approaches with jointly estimated parameters in a single EM framework during the prediction step, whereas Section 4.3.1 introduces a von-Mises-Fisher complex angular central Gaussian mixture model (vMFCACGMM) as a concrete example of the tight integration framework and Section 4.3.2 highlights model specifics. Rounding off, Section 4.4 introduces quite a different interpretation of how knowledge from multi-channel features can be integrated into training single-channel systems resulting in unsupervised training of neural network-based source separation models.

4.1 Existing integration approaches

Due to the diversity of integration variants and the rather loose definition of an *integration system* we here provide a brief literature review of approaches which (1) combine spatial and spectral cues, and (2) do so by relying on pretrained neural networks while estimating observation-dependent parameters on the test mixture. Table 4.1 provides a selection of integration approaches grouped into methods based on a local Gaussian model [86] and models relying on a sparsity assumption and contain a mixture of densities in the sense of the models introduced in Section 3.2.1. The underlying assumption of the former group is that the class-conditional spatial covariance matrices of the spatial observations mix linearly (Compare *linearity of mixing* in [156, Section 2]). The latter group relies on the sparsity assumption (see Section 3.2.1) and the observation distribution is a weighted sum of class-conditional distributions (see Equation 3.6).

One way to elicit an integrated solution to BSS is to design a statistically-motivated source model and a statistically-motivated spatial model. The parameters of both models can then either be partially pretrained on a training database, estimated successively, namely in a cascade approach, or jointly during the prediction step when a joint EM formulation can be found.

According to [58, Page 74], multi-channel NMF, as briefly mentioned in Section 3.2, is the first principled attempt to unify modeling of spatial and spectral cues: [57] relies on a NMF source model and Duong’s full-rank local Gaussian model [86] to handle spatial cues. Ozerov et al. generalize [57] into a generic framework with a wide choice of spectral models [156] ([156, Table 1] is an elaborated overview of integration models based on a local Gaussian model.). A TV-cGMM with a NMF source model and reduced computational demands is proposed in [102] building upon [156]. The system presented in [157] is the first integration of a local Gaussian model and a DNN: a pretrained DNN is applied in each EM-iteration and replaces the source model. Mogami et al. integrate IRLMA [54] with a pretrained DNN [158]. Kameoka et al. employ a variational auto-encoder (VAE) as a source model while relying on a local Gaussian model to characterize the spatial observations [159].

The larger group of integration models in Table 4.1 rely on the sparseness assumption. Early on, Nakatani et al. proposed the DOLPHIN framework which integrates a HMM source model with a Gaussian distribution on IPD features as a spatial observation model [160]. Subsequent

Table 4.1: Overview of a selection of integration methods: all models integrate spatial and spectral models. Fields denoted with an asterisk (*) are ambiguous since more than one option is analyzed.

	Multi-source	Joint estimation	Spectral model					Spatial model	
			Type	Pretrained	Provides deep features	Models local variance	Provides initialization	Type	ICA-based
Local Gaussian models additive in spatial covariance space:									
Arberet et al. [57]	✓	✓	NMF	✗	✗	✓	✗	TV-cG	✗
Ozerov et al. [156]	✓	✓	*	*	✗	✓	✗	TV-cG	✗
Thiemann et al. [102]	✓	✓	NMF	✗	✗	✓	✗	TV-cG	✗
Nugraha et al. [157]	✓	✓	DNN	✓	✗	✓	✓	TV-cG	✗
Mogami et al. [158]	✓	✓	DNN	✓	✗	✓	✓	TV-cG	✓
Kameoka et al. [159]	✓	✓	VAE	✓	✗	✓	✗	TV-cG	✗
Sparseness-based mixture densities additive in PDF space:									
Nakatani et al. [160]	✓	✓	HMM	✓	✗	✓	✗	Gaussian	✗
Nakatani et al. [161]	✓	✓	HMM	✗	✗	✓	✗	cG	✗
Nakatani et al. [162]	✗	✓	GMM	✓	✗	✓	✗	cG	✗
Souden et al. [151]	✓	✓	GMM	✓	✗	✓	✗	cG	✗
Meutzner et al. [163]	✗	✓	DNN	✓	✗	✓	✗	cG	✗
Tran Vu et al. [164]	✓	✓	HMM	✓	✗	✓	✗	cW	✗
Nakatani et al. [165]	✗	✗	DNN	✓	✗	✗	✓	cACG	✗
Drude et al. [106]	✓	✓	DNN	✓	✓	✗	✓	TV-cG	✗
Drude et al. [166]	✓	*	DNN	✓	✓	✗	✓	TV-cG	✗
Drude et al. [167]	✓	✗	DNN	✓	✓	✗	✓	cACG	✗
Drude et al. [51]	✓	*	DNN	✓	✓	✗	✓	cACG	✗

approaches replaced the rather simplistic Gaussian model with a complex Gaussian line orientation model and allowed to train HMM parameters on the test utterance [161], [162]. Souden et al. formalized this approach further into an integrated framework for BSS including statistically optimal beamforming based on the clustering results [151]. The DOLPHIN approach was integrated with a neural mask estimator later to improve speech enhancement results in a single speaker setup [163]. Tran Vu et al. integrated a cWMM with a pretrained 2D-HMM spectral model: both HMM directions allowed spectro-temporal smoothing of the class affiliation posteriors for a multi-source scenario [164].

In 2017 Nakatani et al. presented an integration of a neural mask estimator with a cACGMM as a spatial model: the neural mask estimator provides the initialization as well as the time- and frequency-dependent mixture weight of the subsequent cACGMM for a single-speaker scenario [165]. This model can be seen as the predecessor of the cascade approaches introduced in Section 4.2.

Also in 2017 we introduced the *tight integration* approach [106]. In contrast to the aforementioned DNN integrations, [106] does not use the DNN as a source model. Much rather, the pretrained DNN, e.g., a DC network transforms the observation spectrogram into features that can be grouped better with a rather simple clustering algorithm. In comparison to, for instance, [157] this allows us to run the DNN only once and not in every EM-iteration. The tight integration approach is formalized into a more general framework in [51].

4.2 Cascade approach: Integration by initialization

One of the main weaknesses of probabilistic spatial clustering with the help of EM algorithms is the initialization. Although convergence guarantees for the EM algorithm suggest that the steps never decrease the likelihood, the initial state heavily influences the outcome of the algorithm (see Section 5.5.2 for some examples). Typically, these issues have either been addressed with preclustering [11, Page 427], e.g., with k-means [88], k-means++ [168] or similar approaches (compare, for example, k-mode clustering developed for directional data [89]), or with a heuristic selection of initial mean vectors/ mode directions, for instance, with a deflation algorithm [85].

Given that the single-channel separation algorithms already provide a solid separation result, it is not far to seek that they may serve well as an initialization for spatial clustering. Motivated by the initialization of a spatial clustering model with a mask estimation network in [165]¹, we here analyze this cascade approaches for multi-speaker scenarios.

To initialize a spatial mixture model one can either set initial class affiliation posteriors $\gamma_{k,t,f}$ or set initial values for the class-dependent parameters, e.g., mode vectors or covariance matrices. Since DC, DANs, and PIT use soft masks as an intermediate representation before calculating the separated speech signals, one can set the initial class affiliation posteriors $\gamma_{k,t,f}$ to the masks obtained with the single-channel separation approach. Initializing with

¹ In [165] the mask from the DNN is used as an initialization as well as a time- and frequency-dependent prior.

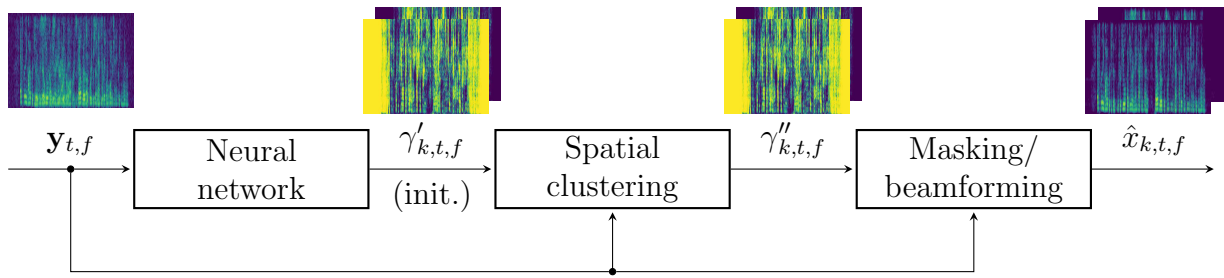


Figure 4.1: Cascade approach to integrated BSS. A neural network-based approach is used to obtain masks $\gamma'_{k,t,f}$ as an initialization to a subsequent spatial clustering. Posterior masks $\gamma''_{k,t,f}$ resulting from the spatial clustering can then be used for source extraction with beamforming and/or masking.

DC masks can cause numerical instabilities since the embeddings are clustered with k-means and hard assignments may result in no observation belonging to a particular class for a given frequency. This can be alleviated by clipping the masks to a limited range, e.g., $[\epsilon, 1 - \epsilon]$, where ϵ is a very small value, e.g., 10^{-6} . Depending on the final output nonlinearity DANs and PIT can lead to masks which do not sum up to one. Proper re-normalization or clipping can again be used to adjust the initialization.

Additionally, since the single-channel results do not suffer from a frequency permutation problem (compare Section 3.2.1.1) the frequency permutation problem of the subsequent spatial clustering is eased albeit not solved entirely (A corresponding evaluation can be found in Table 5.24.).

It is worth noting that the cascade approach does not only allow us to initialize the subsequent model with the result of the first model. To be precise, [165] uses the masks obtained from the neural network to initialize the class affiliation posterior of the spatial clustering model as well as to set a fixed time- and frequency-dependent prior $\pi_{k,t,f}^{(\text{fix})}$. Analogously, [166] contrasts two cascade approaches and a full update model as described in Section 4.3: one cascade approach first runs a spatial clustering model providing an initialization as well as a time- and frequency-dependent prior for a spectral clustering model while the complementary model does just the opposite (Compare [166, Figure 1] for a visualization of all three variants.).

Figure 4.1 shows the cascade processing pipeline. In general, the neural network-based separation step can use single- or multi-channel features. However, in most cases, it is sufficient to obtain the initialization on a single-channel and then use all channels for the subsequent spatial clustering step. Finally, separated signals can be obtained by masking and/or beamforming.

4.3 Tight integration of spatial and spectral features

This section describes *tight integration* as proposed in [51], [106], [166]. The overall processing flow is illustrated in Figure 4.2, in which a neural network calculates deep features, namely embedding vectors $\mathbf{e}_{t,f}$, from the observed spectrogram. In an integrated clustering approach, these spectral features are together with spatial observations to find a consensus for the

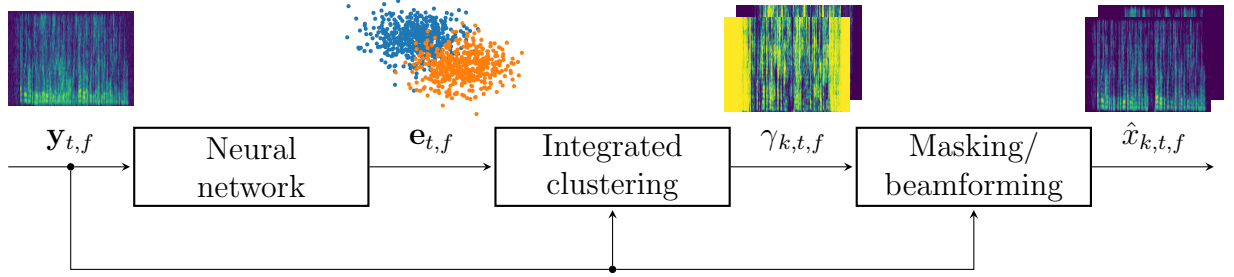


Figure 4.2: Processing steps of a tight integration approach to BSS. A pretrained neural network-based approach is used to obtain embedding vectors as a form of deep features. These embeddings are then jointly clustered together with spatial features. In contrast to Figure 4.1 the embeddings can be seen as deep features which do not just serve as initialization but contribute to the estimation result in each EM step.

clustering result. Consequently, the neural network weights are obtained on separate training data while the parameters of the probabilistic graphical model are estimated on a given mixture at test time. Intuitively, the deep features then incorporate knowledge of a potentially large database while the estimation of the parameters of the probabilistic graphical model on the given mixture at test time facilitates adaptation to unseen conditions.

To concisely capture spatial and spectral cues, one can formulate a single probabilistic model capturing both cues as separate observable random variables. Doing so, the resulting model can be seen as a clustering model with multiple independent observation models (sometimes called *observation heads*) conditional on the latent random class affiliation:

$$\begin{aligned}
 p(f_1(\mathbf{y}_{t,f}), f_2(\mathbf{y}_{t,f}), \dots) &= \mathbb{E}_{p(\mathbf{c}_{t,f})} \left\{ p(f_1(\mathbf{y}_{t,f}), f_2(\mathbf{y}_{t,f}), \dots, \mathbf{c}_{t,f}) \right\} \\
 &= \sum_k p(c_{k,t,f}=1) p(f_1(\mathbf{y}_{t,f}), f_2(\mathbf{y}_{t,f}), \dots | c_{k,t,f}=1) \\
 &\approx \sum_k p(c_{k,t,f}=1) p(f_1(\mathbf{y}_{t,f}) | c_{k,t,f}=1) p(f_2(\mathbf{y}_{t,f}) | c_{k,t,f}=1) \dots, \quad (4.1)
 \end{aligned}$$

in which f_1, f_2, \dots are appropriate feature extraction methods or preprocessing steps. To name an example in line with Figure 4.2 $f_1(\mathbf{y}_{t,f})$ could extract spectral features by calculating embedding vectors $\mathbf{e}_{t,f}$ with, e.g., DC while $f_2(\mathbf{y}_{t,f})$ could capture spatial cues, e.g., by removing the magnitude of the observation as in Equation 3.8.

However, to be able to factorize the conditional distribution into conditional distributions for each observation type, the transformed random variables $f_1(\mathbf{y}_{t,f}), f_2(\mathbf{y}_{t,f}), \dots$ need to be conditionally independent. Although this is possible, in the case of the spatial and spectral features at hand, the conditional independence only holds approximately: intuitively, the variance of the spatial cues is lower when the spectral cues indicate that a speaker is likely to be present. Ignoring this in the following and relying on the conditional independence assumption, the resulting posterior masks may be overly confident.

The advantage of tight integration over the approaches using a single feature is obvious: spectral features alone lack spatial information and vice versa. When using only spatial

Table 4.2: Different manifestations of the tight integration framework for BSS. Different deep features require either a von-Mises-Fisher (vMF) distribution, a Gaussian distribution, or a Beta distribution as a spectral model. In that sense, the abbreviations in the following can be understood as, e.g., GTV-cGMM = (G + TV-cG)-MM, where G abbreviates the spectral and TV-cG the spatial model.

Spatial model	DNN for deep features/ corresponding spectral model		
	DC/ vMF	DAN/ Gaussian (G)	PIT/ Beta (B)
TV-cG	vMFTV-cGMM [106]	GTV-cGMM [166]	
cACG	vMFCACGMM [51]	GcACGMM [51]	BcACGMM [170]

cues, a system is likely to confuse speakers which are very close to each other or even stand behind each other (compare Figure 5.10). When using only spectral cues, it is likely to confuse speakers with similar voices (compare Figure 5.12a or [169] for an analysis of how voice similarity influences DC performance). In comparison to the cascade approach in Section 4.2, the tight integration updates all parameters jointly while the cascade approach can potentially *forget* the spectral information after sufficiently many EM steps. This is conceptually different from ensemble learning [153] in the sense that all the class-dependent parameters are updated independently given the current joint estimate of the posterior affiliation $\gamma_{k,t,f}$. In that sense, each E-step in each iteration can be seen as the ensemble result of the prior, the spectral model from the previous M-step and the spatial model from the previous M-step:

$$\ln \gamma_{k,t,f} = \ln p(c_{k,t,f}=1) + \ln p\left(f_1(\mathbf{y}_{t,f}) \middle| c_{k,t,f}=1\right) + \ln p\left(f_2(\mathbf{y}_{t,f}) \middle| c_{k,t,f}=1\right) + \dots + \text{const.} \quad (4.2)$$

In general, this tight integration framework allows for the use of any spatial observation model. A TV-cGMM was used in [106] and [166], while [51] used a cACGMM for all integration variants. However, there is no conceptual issue with using any of the spatial observation models mentioned in Section 3.2.1 and beyond. While we here compare DC and DANs as neural networks to obtain embedding vectors (or deep features) in [106] and [166], respectively, one may use any neural network which can improve separability of the spectral features over using STFT features. To name an example, [170] directly interprets the soft masks produced by a PIT system as deep spectral features for a tight integration approach. Table 4.2 provides an overview of different variants of the tight integration framework.

4.3.1 vMFCACGMM

Exemplary, this section introduces the vMFCACGMM – a concrete realization of the aforementioned tight integration framework. In a vMFCACGMM, the latent class affiliations $c_{k,t,f}$ represent which time-frequency bin belongs to which speaker or noise. A multivariate vMF distribution [171] models spectral features obtained from a DC model in the form of

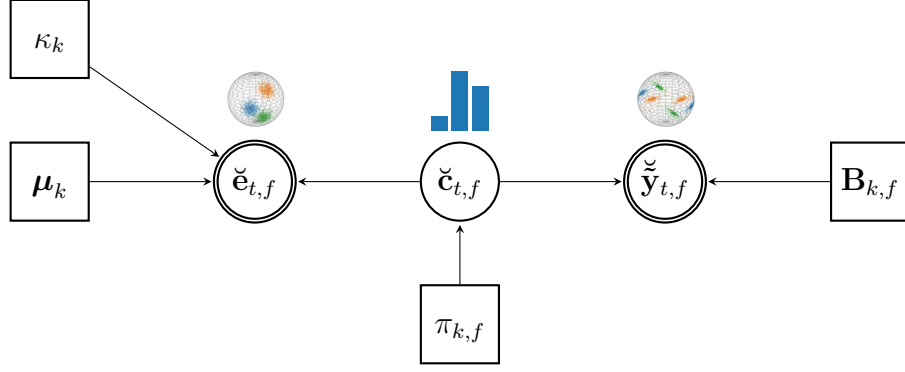


Figure 4.3: Graphical model of a vMFcACGMM. The complex-valued distribution is visualized by its real-valued counterpart. Please note that the spatial model is frequency-dependent while the spectral model is not. Circles depict random variables, while doubly circled elements are observable random variables. Boxes are model parameters which are estimated during test time. Arrows indicate statistical dependencies.

embedding vectors $\mathbf{e}_{t,f}$ conditional on $c_{k,t,f}$:

$$p(\mathbf{e}_{t,f} | c_{k,t,f}=1) = \text{vMF}(\mathbf{e}_{t,f}; \boldsymbol{\mu}_k, \kappa_k) = \frac{1}{c_{\text{vMF}}(\kappa_k)} e^{\kappa_k \boldsymbol{\mu}_k^T \mathbf{e}_{t,f}}, \quad (4.3)$$

where $c_{\text{vMF}}(\kappa_k)$ is an appropriate normalization term [171]. The DC model is trained on separate training data to translate a mixture spectrogram into embedding vectors which are more easily separable by a subsequent clustering algorithm (compare Section 3.1.2.1). The vMF distribution is an adequate distribution to model DC embedding vectors since they are normalized to unit norm and therefore lie on a unit hypersphere. A complex angular central Gaussian (cACG) distribution (compare Section 3.2.1.8) models the normalized spatial observations $\tilde{\mathbf{y}}_{t,f}$ obtained according to Equation 3.8. Typically, due to a frequency-dependent ATF as described in Section 3.2.1.8 parameters for the cACG distribution need to be obtained for each frequency bin independently. In contrast, the embedding vectors obtained with DC are consistent across all time-frequency bins in a given mixture and can, therefore, be modeled with a single vMF distribution per class (each speaker and noise). Figure 4.3 illustrates the underlying probabilistic graphical model. Referring to the processing blocks in Figure 4.2, the neural network, in this case, is the DC network and the integrated clustering model is the vMFcACGMM. Just as in Figure 4.2 a subsequent masking or beamforming step can then be used to obtain speech signals for each source.

Following from Equation 4.1 the probabilistic graphical model is formulated as follows:

$$\begin{aligned} p(\mathbf{e}_{t,f}, \tilde{\mathbf{y}}_{t,f}) &= \sum_k p(c_{k,t,f}=1) p(\mathbf{e}_{t,f} | c_{k,t,f}=1) p(\tilde{\mathbf{y}}_{t,f} | c_{k,t,f}=1) \\ &= \sum_k \pi_{k,f} \text{vMF}(\mathbf{e}_{t,f}; \boldsymbol{\mu}_k, \kappa_k) \text{cACG}(\tilde{\mathbf{y}}_{t,f}; \mathbf{B}_{k,f}). \end{aligned} \quad (4.4)$$

With any of the methods listed in Section 2.5.3 we can now derive an EM algorithm for the vMFcACGMM. In line with the generic formulation of the E-step in Equation 4.2 the class

affiliation posterior $\gamma_{k,t,f}$ can be obtained as follows:

$$\ln \gamma_{k,t,f} = \ln \pi_{k,f} + \ln \text{vMF}(\mathbf{e}_{t,f}; \boldsymbol{\mu}_k, \kappa_k) + \ln \text{cACG}(\tilde{\mathbf{y}}_{t,f}; \mathbf{B}_{k,f}) + \text{const.} \quad (4.5)$$

The M-step consists of the parameter updates, e.g., by maximizing the auxiliary function with respect to the parameters independently. Since the spatial and the spectral observation are conditionally independent given the latent class affiliation, the spatial and the spectral parameters can be estimated one-by-one. Following Equation 2.4, Equation 2.5, and Equation 4.4 in [171] or adapted to an integration model Equation 8 – Equation 9 in [106] the parameters of the vMF distribution can be updated as follows:²

$$\boldsymbol{\mu}_k = \mathbf{r}_k / \|\mathbf{r}_k\| \quad \text{with} \quad \mathbf{r}_k = \sum_{t,f} \gamma_{k,t,f} \mathbf{e}_{t,f}, \quad (4.6)$$

$$\kappa_k = \frac{\bar{r}_k E - \bar{r}_k^3}{1 - \bar{r}_k^2} \quad \text{with} \quad \bar{r}_k = \|\mathbf{r}_k\| / \sum_{t,f} \gamma_{k,t,f}, \quad (4.7)$$

where E is the embedding dimension. Just as in Equation 3.41 and Equation 3.42 the mixture weight and the cACG parameter matrix are updated as follows:

$$\pi_{k,f} = \frac{N_{k,f}}{T} \quad \text{with} \quad N_{k,f} = \sum_t \gamma_{k,t,f}, \quad (4.8)$$

$$\mathbf{B}_{k,f} = \frac{D}{N_{k,f}} \sum_t \gamma_{k,t,f} \frac{\tilde{\mathbf{y}}_{t,f}^H \tilde{\mathbf{y}}_{t,f}}{\tilde{\mathbf{y}}_{t,f}^H \mathbf{B}_{k,f}^{-1} \tilde{\mathbf{y}}_{t,f}}. \quad (4.9)$$

4.3.2 Additional constraints

All parameters of the integration model can be estimated on the test mixture. In its generic form, all parameters are unconstrained, e.g., when using a Gaussian complex angular central Gaussian mixture model (GcACGMM), a full covariance matrix for the spectral observation model is possible. However, in practice it turns out that additional constraints such as enforcing a scaled identity instead of a full covariance matrix as in [51] is mandatory – a full covariance model does not converge (compare the evaluation of different constraints in Section 5.4.1).

In [106] the tight integration model is introduced with an exponential weighting of the spectral observation model and the spatial observation model. However, as argued later in [51, Section V.C], an alternative, which is more convenient and better justified, is to use a fixed concentration parameter for the vMF distribution in a vMFcACGMM or a fixed scale parameter for the Gaussian distribution in a GcACGMM.

² The update for the concentration parameter κ_k is already an approximation. An implicit equation needs to be solved for an exact solution [171, Equation 2.5]. In practice, the approximate solution is sufficiently precise.

4.4 Unsupervised training using multi-channel features

An alternative way to make use of spatial information is to employ it for the training of a single-channel system. That way, one mitigates the need of supervision data in contrast to supervised separation networks, e.g., for DC [43], [47], DAN [46], [172], and PIT [44], [45] with their originally proposed training schemes. To be precise, in most cases, artificial mixtures are used to produce parallel clean and noisy data, whereas one thrives to use more realistic data to avoid a mismatch between training and test. To name an example, Zhou et al. argue that the difference in the acoustic conditions between training and test may be severe enough that gains due to a neural network may be more than compensated [173]: depending on the circumstances, a spatial clustering approach might perform better than a neural network on the test data. Therefore, it is worthwhile to explore how multi-channel data can be used to accommodate for the lack of supervision data. Although in a commercial setup one might train on simulated data and then just fine-tune on real data, we here describe entirely unsupervised approaches.

In [173] the authors propose a teacher-student training scheme, in which a TV-cGMM as in [174] is used to generate either binary masks or soft masks as training targets for the student DNN. The student DNN is then later employed as a mask estimator for neural mask-based beamforming [173]. In [167] we present an unsupervised DC system, in which a cACGMM serves as a spatial clustering model and acts as a teacher for a student DC neural network. It turned out that the student system can indeed outperform the teacher when the student system is used to again initialize a cACGMM. In parallel Tzinis et al. trained a student DC system with a k-means clustering algorithm as the teacher [175]: k-means is used to cluster inter-channel phase differences and corresponding posterior masks then act as supervision for DC. In [176] the authors also analyze a teacher-student approach to unsupervised DC and focus on a particular observation weighting scheme to improve training results. Comparing [175], [176], and [167], the former two systems show that even a very weak teacher is able to train a reasonable DC system. In contrast, the latter system is better tuned towards performance relying on a state of the art spatial clustering model and reporting competitive WERs. In [177] we presented a different take on unsupervised mask estimation: the likelihood under the assumption that the data follows a cACGMM is used as a maximization criterion to train a neural network which just provides the initialization to a single EM-step of the cACGMM parameter estimation process. That way, the network is encouraged to provide a mask as initialization which is close to an optimal initialization, thus leading to a higher likelihood.

Table 4.3 compares the aforementioned unsupervised approaches to mask estimation and source separation. An evaluation of unsupervised DC as in [167] is evaluated in Section 5.8.

Table 4.3: Overview of different approaches to unsupervised training of neural network-based speech enhancement and source separation. The training scheme indicates if spatial clustering is used as a teacher or woven into a likelihood term or evidence lower bound (ELBO).

	Application		Training scheme	
	Single-speaker	Multi-speaker	Teacher-student	Likelihood/ ELBO
Zhou et al. [173]	✓		✓	
Drude et al. [177]	✓			✓
Drude et al. [167]		✓	✓	
Tzinis et al. [175]		✓	✓	
Seetharaman et al. [176]		✓	✓	
Bando et al. [178]		✓		✓

5 Evaluation

The evaluation chapter first introduces performance metrics in Section 5.1 and the databases considered in this work in Section 5.2. Section 5.3 – Section 5.6 explain and justify the choice of the system components and baseline systems and the parameters therein. Section 5.7 analyzes proposed integration variants in detail and Section 5.8 briefly addresses unsupervised training of neural network-based source separation. Section 5.9 and Section 5.10 provide an overview of all proposed methods and put these into perspective by comparing with the baseline approaches on two distinct databases. Key aspects are carved out by examining slices of the corresponding datasets.

5.1 Performance metrics

The overall goal in mind here is to minimize WER obtained when comparing the speech recognition results with a ground truth transcription. The WER is obtained by first accumulating insertion, deletion, and substitution errors in each utterance and then finally dividing the result by the total number of words in the dataset.

Due to the fact that we here aim for modular approaches with a meaningful intermediate signal, we are also able to evaluate the quality of the source separation result. To do so, I selected three closely related signal to distortion ratio (SDR) measures and two speech quality measures. SI-SDR, which is here only used in the single-channel instantaneous mixing scenario calculates the ratio of target signal power and estimation error power with compensation for scale mismatch [179, Equations 2 – 5]. It does not account for small filtering effects which would be necessary when evaluating one channel against another. BSS-Eval SDR is a projection-based definition of SDR [180, Equation 13] and does not impose restrictions on the enhancement system itself, e.g., it does not enforce linearity of the enhancement. Furthermore, it allows short filter effects and therefore is applicable for convolutive mixtures. Another SDR metric which requires linear enhancement but, due to that, does not require additional estimation techniques is invasive SDR as defined in, e.g., [181, Equation 7]. See [181] for an extended discussion of the advantages and disadvantages of different SDR measures. All SDR values are reported on a (principally unlimited) decibel scale. The perceptual evaluation of speech quality (PESQ) metric is a speech quality metric developed to assess telephone transmission quality [182]. It is, therefore, only remotely suitable to assess source separation quality but is included here due to its widespread use. To be precise, we here report narrowband PESQ results due to using 8 kHz sampling rate. The values are reported in

terms of mean opinion score (with MOS-LQO mapping) in the range $[1, 5]$.¹ As an alternative perceptual quality metric we here employ short-time objective intelligibility (STOI) [183] with values in the range $[0, 1]$.

5.2 Database design

The conclusions drawn from an evaluation often depend largely on the database used for the evaluation although one preferably intends to investigate and understand properties of models and algorithms and not databases. The single-channel neural network-based source separation systems introduced in Section 3.1.2 were all developed on a rather artificial mixture corpus (WSJ0-2mix) which was introduced alongside the DC publication [43]. Although performance metrics on this particular database have always been increasing and most very recent models are still being evaluated on this database (see, e.g., [179, Table 1]), very little attention has been paid to evaluation in more realistic noisy and reverberant environments. Only fairly recently WHAM!, an artificial mixture database with real noise recordings was released [184]. However, they still do not address reverberation nor do they adequately account for the mismatch between reverberation-free (dry) target speech recordings and the fairly reverberant background noise. Much in contrast to the fairly large databases used to train and evaluate neural network-based source separation systems, probabilistic spatial mixture models are far too often evaluated on just a fraction of the amount of data, e.g., 8 mixtures per reverberation time in [73]. Besides this critical reflection, it is fair to mention that real recordings with parallel oracle speech images are almost impossible to acquire.

We here first describe the details of the WSJ0-2mix database [43]. Using this database is necessary to prove that our baseline models indeed yield comparable results. Then, we introduce the WSJ-BSS database which contains artificially reverberated Wall Street Journal (WSJ) utterances with white background noise and allows acoustic model training. Finally, we describe the WSJ-MC database which contains real mixture recordings. Although the database is rather small, it can serve as an initial proof that the proposed methods indeed generalize to realistic applications.

All evaluation results will be presented for a sampling rate of 8 kHz, a STFT window size of 512 (64 ms), a shift of 128 (16 ms), a DFT size of 512 (64 ms) and a Hann window to control spectral leakage.²

5.2.1 WSJ0-2mix

The WSJ0-2mix database was released alongside the DC publication [43]. It contains six datasets three of which contain two and three of which contain three speakers. The train and development source signals are taken from the WSJ dataset `train_si284` downsampled

¹ We here rely on a particular C implementation. In practice only values approximately in the range $[1.1, 4.6]$ appear. See <http://www.pesq.org/> for additional details and a more recent successor metric.

² Albeit all confusion this name refers to Julius van Hann, an Austrian meteorologist. It refers neither to Hanning nor Hamming. See <https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.hann.html> for an implementation.

Table 5.1: Specifics of the WSJ0-2mix database.

(a) Co-occurrence of speakers.				(b) Number of mixtures in relation to unique utterances.		
Dataset	Train	Dev	Test	Dataset	Mixtures	Utterances
Train	101	101	0	Train	20 000	8 769
Dev	101	101	0	Dev	5 000	3 557
Test	0	0	18	Test	3 000	1 770

Table 5.2: Input metrics of the WSJ0-2mix database.

Dataset	SDR / dB			PESQ	STOI
	SI	BSS-Eval	Invasive		
Dev	0.00	0.15	0.00	1.66	0.72
Test	0.00	0.15	0.00	1.68	0.74

to 8 kHz and the test signals are taken from the WSJ dataset `test_eval92_5k` [185], [186].³ To generate mixtures one can select a `min` configuration, in which the shortest utterance determines the total length of the signal and a `max` configuration, in which the shorter utterances are zero-padded to match the longest utterance in the given mixture. All evaluations in this work related to the WSJ0-2mix database were done on the `min` configuration. Correspondingly, only the shorter utterance can be used for ASR. All two or three source signals are then simply added up with a mixing ratio of -2.5 dB up to 2.5 dB to result in the mixture signal. Since, at that time, it was important to prove that a system does work on unseen speakers (open condition) just as well as on speakers seen during training (closed condition) the authors of that database decided that the development set contains only speakers seen during training while the test set contains only unseen speakers (see Table 5.1a).

Furthermore, the database contains significantly less unique utterances (of which many are truncated as mentioned previously) than mixtures and is therefore not ideal for acoustic model training. Table 5.1b lists the total number of mixtures and the number of unique utterances per dataset.

Table 5.2 summarizes the input metrics for this database. They serve as a reference to, e.g., calculate gains and help understand what the lower limits of each metric are.

5.2.2 WSJ-BSS

The WSJ-BSS database consists of 30 000, 500, and 1 500 six-channel mixtures. Each mixture is obtained by reverberating each of the two source signals with an artificially generated

³The dataset names are taken from the corresponding Kaldi recipes.

Table 5.3: Specifics of the WSJ-BSS database.

(a) Co-occurrence of speakers.				(b) Number of mixtures in relation to unique utterances.		
Dataset	Train	Dev	Test	Dataset	Mixtures	Utterances
Train	283	0	0	Train	30 000	30 000
Dev	0	8	0	Dev	500	491
Test	0	0	10	Test	1 500	333

room impulse response and adding white Gaussian noise with a SNR of 20 dB – 30 dB. The source signals are obtained from the three non-overlapping WSJ datasets `si284`, `dev93`, and `eval92` for training, development and test [185], [186]. The source utterances were selected in such a way that as many unique utterances as possible are covered while avoiding punctuation pronunciation (e.g., spoken *question mark*) to facilitate sequence-to-sequence acoustic model training. Table 5.3a shows how many speakers are in each dataset and whether they appear again in different datasets. Table 5.3b lists how many unique utterances are available for ASR. A variant of this database with simplified simulation conditions, padding to the maximum utterance length, and some removed edge-cases is published as SMS-WSJ with all RIRs, code, and an ASR baseline [181].⁴ A detailed comparison can be found in Appendix A.8.

The total length of the mixture is chosen in such a way that the first source utterance always determines the length of the mixture. The second source utterance is truncated or padded accordingly. The motivation is again to provide sanitized conditions for acoustic model training. Later, only the estimate of the first source utterance is transcribed by the speech recognition system.⁵ The room impulse responses were generated using the image method [187].⁶ The reverberation time was uniformly sampled in the range [200 ms, 500 ms]. To cover a rather large variety of simulation geometries, the room length, width, and height are uniformly sampled from [7.6 m, 8.4 m], [5.6 m, 6.4 m], and [2.6 m, 3.4 m], respectively. Similarly, the source positions are first sampled from a circle with a uniformly sampled radius in the range [1 m, 2 m] centered at the array center and then moved by a random offset again sampled from the range [−0.4 m, 0.4 m] in each coordinate axis. No minimum angular distance was enforced, i.e., two speakers could theoretically stand behind each other [125, Figure 2]. The sampling rate for the database is set to 8 kHz to somewhat reduce computational load and match the WSJ0-2mix conditions. The sensor array itself is simulated as a circular array with radius 10 cm with a random rotation. The geometry of the setup is summarized in Figure 5.1. Table 5.4 lists the input metrics, i.e., the metrics which can be measured when a system simply outputs the reference channel of the observation as a prediction.

⁴ https://github.com/fgnt/sms_wsj

⁵ This is an unnecessary asymmetry, which is avoided in the SMS-WSJ database.

⁶ We here relied on Emanuel Habets’ implementation (<https://github.com/ehabets/RIR-Generator>) with a thin Python wrapper (<https://github.com/boeddeker/rir-generator>).

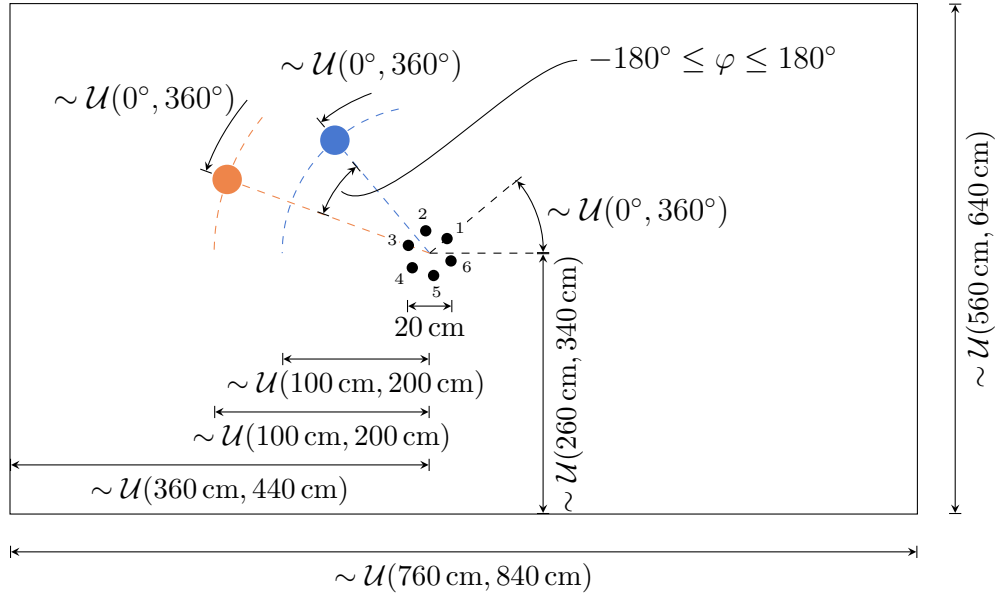


Figure 5.1: Geometry of the WSJ-BSS database. The black dots represent the microphones, the colored dots indicate source positions. Each aspect of the geometry is uniformly sampled from the given range. No minimum angular distance is enforced. The axial center of the sources and the sensor array is independently sampled, i.e., both are not coaxial (not shown in this visualization).

Table 5.4: Input metrics of the WSJ-BSS database.

Dataset	SDR / dB		PESQ	STOI
	BSS-Eval	Invasive		
Dev	-0.37	-0.03	1.62	0.55
Test	-0.38	-0.04	1.48	0.65

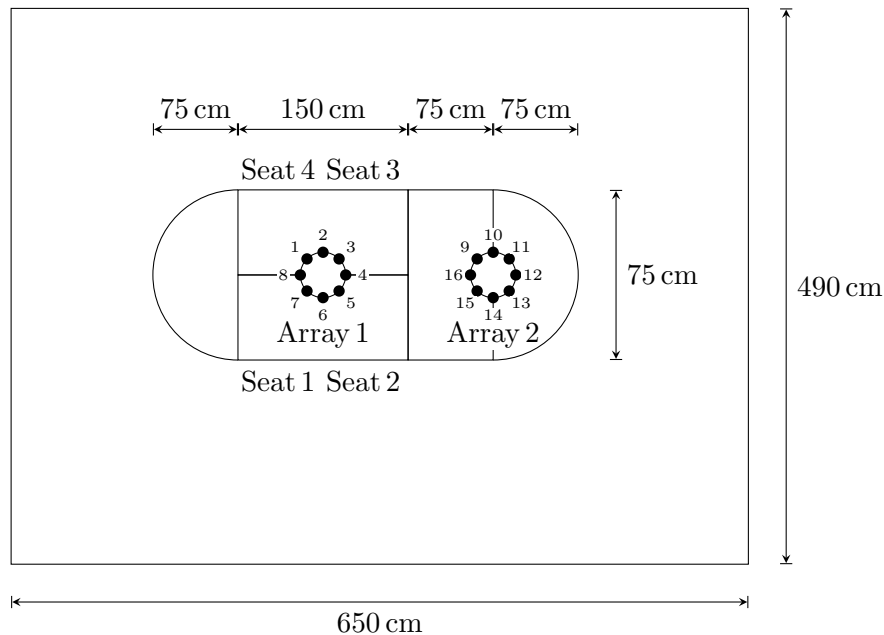


Figure 5.2: Geometry of the WSJ-MC database. Only Array 1 is used for evaluation.

Table 5.5: Input metrics of the WSJ-MC database.

Dataset	BSS-Eval SDR / dB	PESQ	STOI
Dev	-2.29	1.58	0.56
Test	-2.33	1.64	0.58

5.2.3 WSJ-MC

The WSJ-MC database contains 178 real mixtures from the `olap_dev_5k` dataset and 142 real mixtures from the `olap_ev1_5k` dataset of the multi-channel WSJ audio visual corpus [188] which we here resample to 8 kHz. Four British English speakers take turns in reading two sentences from the WSJ simultaneously. In other words, exactly two speakers appear in a single mixture recording. The mixtures are recorded with two 8-channel microphone arrays of which we here only use the first one. The database does not contain an additional train dataset. Thus, all models have to be trained on an artificial database before they can be evaluated on the WSJ-MC database. The database does not contain clean speech recordings, however, the fairly clean headset signal shall serve as a reference signal here. Table 5.5 lists the approximated input metrics when using the headset instead of a clean source signal as a reference. Figure 5.2 illustrates the geometry.

5.3 Acoustic model training

When source separation approaches are optimized with an ASR task in mind, it is important to evaluate with a rather competitive acoustic model (AM). This is necessary to demonstrate that

Table 5.6: Oracle WERs for the WRN on the WSJ-BSS database.

		Test WER / %	
		Image	Noisy image
Train	Image	10.33	10.33
	Noisy image	10.30	10.30

the gains created by modifications to the front-end (the source separation and enhancement) do not get compensated by the back-end (the ASR system). To provide an example, a simple enhancement method might still result in a WER reduction for a GMM-HMM system. A mediocre front-end might lead to a small or no improvement if the back-end is quite strong by itself. We here opted for a rather sophisticated hybrid acoustic model: a wide residual network (WRN) estimates state posterior probabilities and a HMM represents temporal context. This decision is made since although sequence-to-sequence AMs are in principal easier to handle, their performance on limited audio data still lacks behind (compare [189] for a comparison of hybrid and all-neural ASR).

To train the AM, we first train a GMM-HMM system on clean speech following the Kaldi recipe for bootstrapping purposes by using increasingly complex GMM-HMMs.⁷ The final model (**tri4b**) is then used to extract state alignments on the early-arriving speech images of the training and development data. The idea of this is, that the early-arriving speech images are rather similar to clean (and dry) speech while still having the same initial delay due to the simulated time of flight as the speech images of the simulated mixtures. Using these forced alignments on early-arriving speech, a WRN can now be trained on the speech images (containing the full reverberation tail). The WRN used here is a simplification of the wide bi-directional residual network (WBRN) [190] originally developed for the CHiME 4 challenge. The simplified WRN is described in more detail in [137]. The AM here only consists of convolutions with skip connections and a linear layer to produce state posterior logits before entering the softmax nonlinearity. No dropout is used. The Mel filter bank feature extraction and the pooling operations are adjusted to match the reduced sampling rate. The reduced sampling rate and the removal of the bidirectional long short-term memory (BLSTM) layers allow training with a batch size of 4 on a GTX 1070 GPU with convergence in about 24 h. Table 5.6 shows oracle WERs of the WRN on the WSJ-BSS database.

Although a neural network can potentially be warm-started from a previous model, we here avoid this to not overcomplicate the entire pipeline: each AM, also the ones with matched training, i.e., training on the results of a source separation model, are trained from scratch. We do not study the effect of warm-starting here.

In general, we expect that matched training improves the recognition performance, since artifacts produced by the front-end are seen during the training of the back-end. It is worth to keep in mind that, especially in a speech enhancement setup, the matched back-end is exposed to less variability when the front-end already works well on the training data. While this might lead to worse performance in unseen conditions (The hypothesis of the

⁷ Compare the initial part of <https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/run.sh>.

influence of reduced training variability is to some degree supported by [191, Table 3].), we expect this effect to be rather small on a separation task. Matched training is evaluated in Section 5.9.2.

5.4 Deep-learning methods

This section explains how the different neural networks for blind source separation are trained and how particular parameter choices can be justified. First, we identify a few properties of the models on the WSJ0-2mix database. Then, we investigate whether they hold true on the WSJ-BSS database.

Training neural networks involves a substantial number of hyperparameters. For example [192] demonstrate an expensive Bayesian hyperparameter search for source separation neural networks on the WSJ0-2mix database. In general, to limit the search space, it is useful to start close to already established parameter choices on related tasks [193]. Therefore, we first present a parameter overview table for each separation neural network and then describe the models in detail.

5.4.1 Deep clustering

Table 5.7 contains a parameter overview of a selection of publications involving different recipes for training a deep clustering encoder. The table already includes the parameter choice we finally use for further evaluations. The choices are further detailed in the next paragraphs.

All DC models are pretrained with a batch size of 8 of random mixture segments with 32000 samples (4s) each to avoid extensive zero-padding. After the pretraining has converged, training is resumed with a batch size of 8 on entire utterances. Ten buckets based on sequence lengths have been used to group utterances, somewhat homogenize each batch, and reduce zero-padding. The motivation here is that the network is trained in conditions as close as possible to the test stage.

The loss is minimized using the Adam optimization procedure [194] and a step size of $\alpha = 1 \cdot 10^{-3}$. To somehow update gradients more conservatively, the regularizing parameter $\epsilon = 1 \cdot 10^{-4}$ instead of $\epsilon = 1 \cdot 10^{-8}$ is used [194]. All gradients are limited to a maximum length of one. If the length is above one, the gradient is rescaled to length one instead of clipping the offending values. Validation is started after every epoch. For pretraining, optimization is stopped once the validation loss has not decreased for 10 validation runs. Then, the previous best parameters according to the validation loss on the development set are used for evaluation. For fine-tuning a learning-rate decay strategy with automatic back-off is used: if the validation loss does not decrease for 10 epochs the learning rate is halved and the parameters are reset to the previous best parameters according to the validation loss. Once the learning rate is below $4 \cdot 10^{-4}$, the training is stopped and the previous best parameters are used for evaluation.

Table 5.7: Comparison of training details for deep clustering. $4 \cdot 300$ BLSTM units stands for four layers of 300 forward and 300 backward units. Empty cells indicate that the information is not available. Two numbers with an arrow indicate that the parameter is changed after the initial training has converged. The entry marked with an asterisk (*) is taken from [48].

	[43]	[47]	[48]	[106]	[51]	This work
Batch size					4	8
DFT size	256	256	256	512	512	512
DFT window	sqrt. Hann	sine	sqrt. Hann	Blackman	Hann	Hann
Train samples		$6400 \rightarrow 25600$	25600	entire	entire	$32000 \rightarrow$ entire
Features	log mag.	log mag.	log mag.	log mag.	log mag.	log mag.
Input norm		global			sequence	sequence
BLSTM units	2×600	4×300	4×600	4×300	2×600	4×300
Dropout		0.5	0.3	0.5	0.5	0
Rec. dropout	0	0.2	0	0	0	0
Normalization				sequence	sequence	sequence
Stream merge				concat.	concat.	concat.
Embedding dim.	40				20	40
Output nonlin.	tanh					tanh
Loss mask	-40 dB th.	-40 dB th.*	-40 dB th.	98 % quantile		none
Optimizer	M-SGD	RMS-Prop		Adam	Adam	Adam
Learning rate		schedule		schedule	fixed	back-off
Gradient clip		200			1	1
Weight decay				no	no	no
Weight noise	yes			no	no	no

The DC encoder uses log magnitude spectrogram features and consists of 4 BLSTM layers [195], [196] with 300 units in each direction. The forward and backward stream is merged by concatenating the hidden states [196].⁸ All trained models contain a sequence normalization [128, Section 3.1.3] just before each BLSTM layer such that, e.g., each frequency bin is normalized to zero mean and unit variance. In contrast, [47] uses a global mean-variance normalization as a preprocessing step. The main argument for such a normalization here is that the system generalizes better to unknown microphone scaling and is somewhat invariant to equalization effects. This has been done in view of the WSJ-MC database, but not doing so has not been evaluated further. Each long short-term memory (LSTM) cell uses a forget bias of one to discourage forgetting at the beginning of training. A final linear layer is used to map each time-frequency bin to an embedding vector with $E = 40$ dimensions. Neither the input nor the output of the linear layer is normalized. However, a tanh nonlinearity is used after the linear layer. Finally, the embedding vectors are normalized to unit length. The way to regularize the neural network is through the use of early stopping.

⁸Schuster et al. do not discuss concatenation. However, they mention that they split the number of neurons in half. From that follows that they most likely concatenated the outputs.

If not otherwise noted, the networks for the WSJ0-2mix database use the DC loss [43, Equation 1] with $K' = K = 2$ classes while the DC networks for the WSJ-BSS database use an additional noise class: $K' = K + 1 = 2 + 1$. The loss is not masked, although, according to, e.g., [47] masking leads to a slight performance improvement. The reason to not mask the loss is that the additional noise class should sufficiently handle the issue the loss masking was originally intended for.

In contrast to [47], we did not use recurrent dropout [197], [198]. Although this showed improvements in [199] for a phoneme recognition task with BLSTMs, recurrent dropout is avoided to simplify the training recipe.

In derogation from [47], we here use a DFT size of 512 and an accordingly higher shift. This reduces the number of LSTM steps by a factor of two, thus speeding up training significantly while yielding a similar separation performance. Furthermore, we base this parameter choice on the W-disjoint orthogonality analysis in [23, Figure 9.8], in which a DFT size of 512 resulted in a maximum of the W-disjoint orthogonality for three, four, and five speakers given a sampling rate of 8 kHz. A more detailed analysis for 16 kHz with different window functions can be found in [22, Figure 4].

To limit the search space and to somewhat ease reproducibility, all further DC models now share a common architecture, common features, and a common training recipe.

To ensure that the gains of an integration model do not simply stem from a soft clustering of the embedding vectors, it is worth it to first investigate how different latent models influence the performance of a DC system. It is worth keeping in mind that simply switching to a soft clustering model influences the performance of a masking-based extraction more than it would impact a beamforming-based extraction. However, since our final goal is to optimize overall performance, we compare masking results on the WSJ0-2mix database and beamforming results on the WSJ-BSS database.

Table 5.8 compares k-means with other latent models and varying parameterizations. A GMM with an unconstrained (full) covariance per class mostly did not converge for this ($E=40$)-dimensional latent space. Constraining the covariance matrix to a diagonal matrix at least resulted in stable processing but with rather poor separation results. Constraining the covariance matrix further to impose spherical equiprobability surfaces (scaled identity matrix) improves the performance more. Enforcing a fixed scale parameter which itself is optimized by selecting the best invasive SDR validation result for all classes even allows for surpassing the SDR gains of the k-means baseline slightly. A similar observation can be made for a von-Mises-Fisher mixture model (vMFMM) in which the concentration parameter κ seems to have an anti-proportional influence compared to σ^2 for a GMM. It is worth mentioning that this slightly contradicts our previous findings in [106] and [51]: we previously argued that simply using a soft model instead of k-means does not change separation performance. Here, however, we tuned the parameters of the soft clustering model more carefully, which, to some degree, explains the more nuanced results. In conclusion, the k-means algorithm on embedding vectors is already a fairly competitive baseline and can be used for a later comparison with integration models.

Next, we analyze the influence of the mixture weight. Table 5.9 shows that the type of mixture weight as discussed in Section 3.2.1.3 is of minor importance for the embedding

Table 5.8: Comparison of masking results of a k-means clustering, different variants of a GMM, and different variants of a vMFMM on the WSJ0-2mix database. The tuning parameter κ or σ^2 is selected to maximize invasive SDR on the development set.

Initialization	Latent model	Weight type	Parameter	SDR / dB					
				SI-SDR		BSS-Eval		Invasive	
				Dev	Test	Dev	Test	Dev	Test
k-means		$1/K'$		8.83	8.87	9.41	9.43	12.28	12.35
i.i.d.	GMM (diagonal)	$1/K'$	free	3.69	3.63	4.49	4.40	6.81	6.73
i.i.d.	GMM (spherical)	$1/K'$	free	8.44	8.27	9.02	8.83	12.31	12.40
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1/8$	9.20	9.22	9.76	9.75	12.37	12.44
i.i.d.	vMFMM	$1/K'$	free	8.45	8.32	9.03	8.88	12.31	12.39
i.i.d.	vMFMM	$1/K'$	$\kappa = 8$	9.15	9.17	9.73	9.72	12.48	12.52

Table 5.9: Comparison of the influence of the mixture weight type on the WSJ0-2mix database. The free parameter is selected based on best development set invasive SDR.

Initialization	Latent model	Weight type	Parameter	SDR / dB					
				SI-SDR		BSS-Eval		Invasive	
				Dev	Test	Dev	Test	Dev	Test
k-means		$1/K'$		8.83	8.87	9.41	9.43	12.28	12.35
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1/8$	9.20	9.22	9.76	9.75	12.37	12.44
i.i.d.	GMM (spherical)	π_k	$\sigma^2 = 1/8$	9.23	9.23	9.80	9.79	12.53	12.57

clustering model. While this comparison seems to be too detailed for the moment, it is worth pointing out that the performance of models discussed later significantly depends on the choice of the mixture weight.

Table 5.10 compares a GMM and a vMFMM with different initializations. First of all, we observe that the GMM and the vMFMM perform almost equally and both are slightly ahead of the k-means-only result. We can further deduce from Table 5.10 that the embedding mixture models do not profit from an additional k-means initialization, they bootstrap themselves sufficiently well.

Table 5.11 again compares the k-means clustering, the GMM, and the vMFMM but this time on the WSJ-BSS database with beamforming. To be precise, we here use a GEV decomposition as an RTF estimator, a rank-one matrix construction, a Souden-MVDR beamformer, and a BAN postfilter. The underlying DC encoder is now trained with an additional noise class, i.e., $K' = K + 1 = 2 + 1$. Again, we see slight differences between the soft clustering models with the vMFMM once more performing slightly better in terms of invasive SDR. Interestingly, the k-means clustering now performs best. This should suffice

Table 5.10: Comparison of different embedding clustering models and how important proper initialization is on the WSJ0-2mix database. The tuning parameter κ or σ^2 is selected to maximize invasive SDR on the development set.

Initialization	Latent model	Weight type	Parameter	SDR / dB					
				SI-SDR		BSS-Eval		Invasive	
				Dev	Test	Dev	Test	Dev	Test
k-means		$1/K'$		8.83	8.87	9.41	9.43	12.28	12.35
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1/8$	9.20	9.22	9.76	9.75	12.37	12.44
i.i.d.	vMFMM	$1/K'$	$\kappa = 8$	9.15	9.17	9.73	9.72	12.48	12.52
k-means	GMM (spherical)	$1/K'$	$\sigma^2 = 1/8$	9.21	9.22	9.76	9.75	12.37	12.44
k-means	vMFMM	$1/K'$	$\kappa = 8$	9.16	9.17	9.73	9.72	12.48	12.52

Table 5.11: Comparison of latent models for DC on the WSJ-BSS database. Each rows represents beamforming results with a GEV decomposition as an RTF estimator, a rank-one matrix construction, a Souden-MVDR beamformer and a BAN postfilter. The latent model parameter is selected based on maximum invasive SDR on the development set.

Initialization	Latent model	Weight type	Parameter	SDR / dB			
				BSS-Eval		Invasive	
				Dev	Test	Dev	Test
k-means		$1/K'$		9.93	10.28	14.31	14.65
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1/100$	9.74	10.16	14.11	14.52
i.i.d.	GMM (spherical)	π_k	$\sigma^2 = 1/100$	9.52	10.06	13.88	14.41
i.i.d.	vMFMM	$1/K'$	$\kappa = 100$	9.69	10.22	14.03	14.58
i.i.d.	vMFMM	π_k	$\kappa = 100$	9.74	10.19	14.07	14.55

to argue that using k-means for the DC model is a solid baseline for later comparison. Moreover, it becomes apparent that the variance σ^2 is much higher and the concentration κ is much lower than on the WSJ0-2mix database which results in a behavior fairly similar to k-means.

Findings

To finalize this section, we can draw the following conclusions:

- In a DC system applying k-means is already rather competitive.
- The mixture weight has a marginal influence on the embedding clustering models.
- The choice of the concrete soft clustering model for a spectral-only model is of minor importance.

5.4.2 Deep attractor network

Similar to the previous section, we start the analysis of the deep attractor network with a short literature review. Table 5.12 summarizes the parameter choices in selected publications and mentions the parameters used here in the last column. The parameters used here are selected to match the DC parameters as closely as possible. Although this may have led to a slight degradation of the DAN baseline, it facilitates comparison between DC and DAN.

The original motivation for the DAN architecture was to be able to train with a loss function closer to the downstream task (e.g., masking). As detailed in Section 3.1.2.2 the DAN encoder first produces embedding vectors. During training, a supervision mask is then used to calculate weighted means per class (attractors). These attractors can then be used to calculate an inner product with the original embeddings (called DAN decoder in the following) to create a time-frequency mask per class. Table 5.13 compares two different loss functions with the corresponding nonlinearity. The MSE loss as used in the original work is intended to improve masking performance. A possible hypothesis at this point is that CE loss with a softmax nonlinearity may lead to a better initialization of later models. However, at least in our analysis, the CE loss with a softmax nonlinearity (Row 4) already outperforms the MSE loss with a sigmoid nonlinearity (Row 2), when using the DAN decoder also during test time, i.e., when applying Equation 3.4 during inference.

Figure 5.3 shows how the DAN performance depends on the choice of the fixed parameter in terms of invasive SDR. First of all, we notice that a clear optimum is hard to be found. Only when using masking, fixed variance parameters around $\sigma^2 = 1$ lead to an improvement over the k-means result. This can be explained by the smoothing effect of higher variance values. However, when using a beamformer, hard masks are not an issue for performance and k-means avoids any parameter selection at all. Thus, a DAN encoder with k-means, which is arguably easier to handle, and beamforming is already a rather strong baseline.¹⁰

Table 5.14 compares different latent model configurations and different initializations. Similar to the observations with a DC encoder, we observe that an additional k-means initialization is not helpful. The very different selection of the optimal covariance parameter takes place due to the ambiguous maxima as illustrated in Figure 5.3.

¹⁰ The k-means implementation at hand already uses a k-means++ initialization with random restarts.

Table 5.12: Comparison of training details for DANs. $4 \cdot 300$ BLSTM units stands for four layers of 300 forward and 300 backward units. Empty cells indicate that the information was not available.

	[46]	[200]	[166]	[51]	This work
Batch size				4	8
DFT size	256	512	512	512	512
DFT window	sqrt. Hann	Blackman		Hann	Hann
Train samples	6400 \rightarrow 25600	entire	entire	entire	32000 \rightarrow entire
Features	log mag.	log mag.	log mag.	log mag.	log mag.
Input norm		sequence	sequence	sequence	sequence
BLSTM units	4×600	2×600		2×600	4×300
Dropout		0.5		0.5	0
Rec. dropout		0		0	0
Normalization		sequence	sequence	sequence	sequence
Stream merge	concat. ⁹	add	concat.	concat.	concat.
Embedding dim.	20	20		20	40
Output nonlin.			tanh	tanh	
Loss mask	90 % amp. th.	98 % quantile			none
Optimizer	RMS-Prop	Adam	Adam	Adam	Adam
Learning rate	schedule	none		fixed	back-off
Gradient clip		5		1	1
Weight decay		no	no	no	no
Weight noise		no	no	no	no

Table 5.13: Comparison of DAN with different loss functions and different DAN decoder output nonlinearities. Further, the second column indicates whether a DAN decoder was used during inference, i.e., applying Equation 3.4 during inference.

Output nonlinearity	DAN decoder at test time (Equation 3.4)	Loss	SDR / dB					
			SI-SDR		BSS-Eval		Invasive	
			Dev	Test	Dev	Test	Dev	Test
sigmoid	✗	MSE	8.96	8.80	9.55	9.37	12.52	12.39
sigmoid	✓	MSE	9.41	9.28	9.91	9.77	11.29	11.18
softmax	✗	CE	8.71	8.84	9.39	9.45	12.48	12.66
softmax	✓	CE	9.58	9.71	10.16	10.27	12.79	12.96

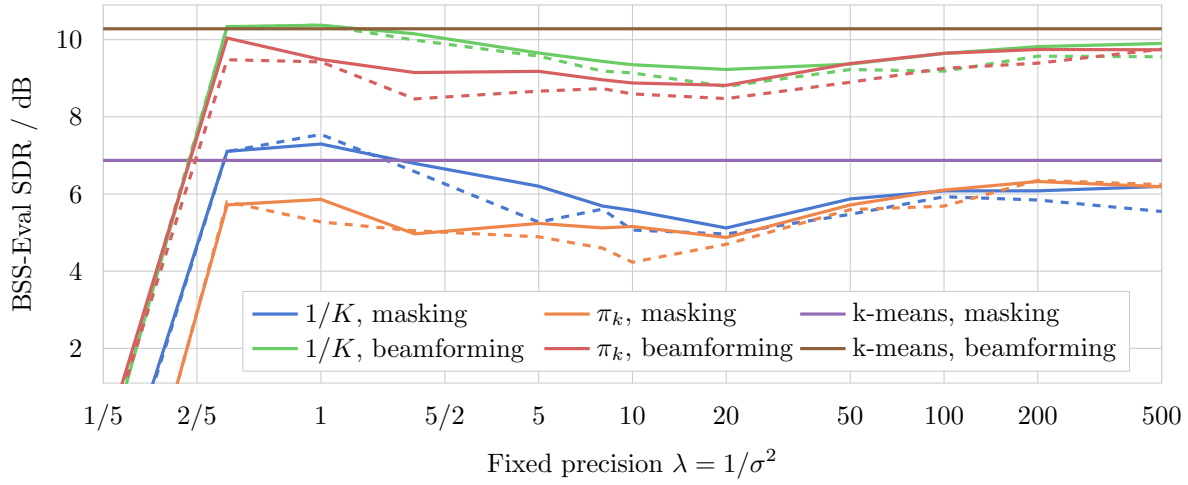


Figure 5.3: Dependency of DAN performance on fixed covariance parameter observed on the WSJ-BSS database. The horizontal axis shows an inverse variance to be better comparable with concentration parameter plots. Dashed lines indicate results on the development set.

Table 5.14: Comparison of different clustering models on DAN embeddings on the WSJ-BSS database. The tuning parameter $\sigma^2 = 1/\lambda$ was chosen to maximize development set invasive SDR. Figure 5.3 provides insights into why the parameter σ^2 differs that much between rows.

Initialization	Latent model	Weight type	Parameter	Extractor	SDR / dB			
					BSS-Eval		Invasive	
					Dev	Test	Dev	Test
k-means		$1/K'$		Masking	7.03	6.88	11.52	11.20
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1$	Masking	7.54	7.30	11.80	11.42
i.i.d.	GMM (spherical)	π_k	$\sigma^2 = 1/200$	Masking	6.36	6.32	10.93	10.67
k-means		$1/K'$		Beamforming	10.24	10.28	14.61	14.58
i.i.d.	GMM (spherical)	$1/K'$	$\sigma^2 = 1$	Beamforming	10.38	10.37	14.69	14.61
i.i.d.	GMM (spherical)	π_k	$\sigma^2 = 1/500$	Beamforming	9.75	9.74	14.04	13.91

Table 5.15: Comparison of training details for PIT. $4 \cdot 300$ BLSTM units stands for four layers of 300 forward and 300 backward units. Empty cells indicate that the information was not available.

	[45]	This work
Batch size	8	8
DFT size	256	512
DFT window		Hann
Train samples	entire	32000 \rightarrow entire
Features	mag.	log mag.
Input norm		sequence
BLSTM units	4×300	
Dropout	0.5	0
Rec. dropout	0	0
Normalization		sequence
Stream merge		concat.
Optimizer		Adam
Learning rate	decay	back-off
Gradient clip		1
Weight decay		no
Weight noise		no

Findings

To finalize this section, we can draw the following conclusions:

- Using k-means to cluster the DAN is a robust baseline and avoids the need for additional parameter tuning.
- A CE loss works well and is less dependent on a DAN decoder during inference.

5.4.3 Permutation invariant training

To begin with, Table 5.15 shows a brief parameter comparison with the original utterance-wise PIT publication [45]. Again, we chose parameters to be consistent with our DC implementation.

One big advantage of PIT over, e.g., DC is its ability to be trained with a reconstruction loss. It had already been shown in previous studies on masking such as [121, Table 4] that a reconstruction loss significantly outperforms a mask approximation loss.

However, it is worth noting that those comparisons focus on direct reconstruction using masking, whereas we here intend to either use the resulting mask to initialize a spatial mixture model or intend to steer a beamforming vector. Both operations are much more indirect and

Table 5.16: PIT results with masking on the WSJ0-2mix database with $K' = K = 2$ classes.

Output nonlinearity	Loss	SDR / dB					
		SI-SDR		BSS-Eval		Invasive	
		Dev	Test	Dev	Test	Dev	Test
sigmoid	NPSMSE	9.04	9.01	9.62	9.58	11.79	11.76
softmax	CE	9.59	9.50	10.16	10.07	12.69	12.63

Table 5.17: PIT results on the WSJ-BSS database. The beamforming rows contain a GEV as an RTF estimator, a rank-one matrix construction, a Souden-MVDR beamformer and a BAN postfilter.

Classes	Output nonlinearity	Loss	Extraction method	SDR / dB			
				BSS-Eval		Invasive	
				Dev	Test	Dev	Test
$K' = K$	sigmoid	NPSMSE	Masking	6.60	6.48	9.51	9.33
$K' = K + 1$	sigmoid	NPSMSE	Masking	6.74	6.59	9.66	9.45
$K' = K + 1$	softmax	CE	Masking	6.37	6.67	9.91	10.03
$K' = K$	sigmoid	NPSMSE	Beamforming	9.28	9.43	13.26	13.37
$K' = K + 1$	sigmoid	NPSMSE	Beamforming	9.15	9.28	13.19	13.25
$K' = K + 1$	softmax	CE	Beamforming	9.71	9.95	14.03	14.25

are conceptually motivated by posterior distributions as masks instead of estimated ratio masks.

Consequently, we here train PIT models with a mask approximation loss as well as a reconstruction loss to then evaluate how this influences end results in different setups.

Table 5.16 shows results on the WSJ0-2mix database using masking. The particular signal level loss is a mean squared error with a nonnegative phase-sensitive mask (NPSMSE) on the magnitude spectrograms [45, Equation 10]. Counterintuitively, the CE loss yields better masking results on the WSJ0-2mix dataset. However, it is fair to point out that the original authors did not evaluate a CE loss and it also may depend severely on the particular neural network.

Table 5.17 shows results on the WSJ-BSS database. While the BSS-Eval SDR results on the development set and the test set seem to contradict, the invasive SDR gains are more consistent. The first row is closest to the configuration presented in the original work [45, Table 3 Row 1]. However, at least in terms of invasive SDR a softmax output nonlinearity with CE loss yields better results. This is favorable since it is expected that all integration models profit from masks resembling posterior distributions more than from masks optimized for a signal reconstruction loss. Also in the context of beamforming, the CE results with an additional noise class work best.

Table 5.18: Comparison of our implementation of DC, DAN, and PIT on the WSJ0-2mix database with reference implementations in the literature. The results with an asterisk were originally reported as gains and are here translated to absolute metrics using the input metrics as in Table 5.2.

Model	Source	SDR / dB					
		SI-SDR		BSS-Eval		Invasive	
		Dev	Test	Dev	Test	Dev	Test
DC	[43, Table 1 Row 4]	5.9	6.0				
DC	[47, Table 7 Row 1]		10.3				
DC	[47, Table 7 Row 3]		10.8				
DC	[48, Table 1 Row 1]	9.6	9.5				
DC	here	8.83	8.87	9.41	9.43	12.28	12.35
DAN	[46, Table 1 Row 6]		10.5				
DAN	[201, Table 1 Row 4]		10.3				
DAN	[179, Table 1 Row 2]		10.4		10.95*		
DAN	here, sigmoid and speech loss	9.41	9.28	9.91	9.77	11.29	11.18
DAN	here, softmax and mask loss	9.58	9.71	10.16	10.27	12.79	12.96
PIT	[45, Table 3 Row 1]			9.65*	9.65*		
PIT	[179, Table 1 Row 3]				10.15*		
PIT	here, sigmoid and speech loss	9.04	9.01	9.62	9.58	11.79	11.76
PIT	here, softmax and mask loss	9.59	9.50	10.16	10.07	12.69	12.63

Findings

To finalize this section, the following conclusions can be drawn:

- PIT is conceptually easier since there is no additional clustering stage involved.
- A CE loss yields mixed results for masking. No clear conclusion can be drawn.
- A CE loss works well with beamforming and, as a consequence, is likely to yield a good initialization for an integration model.

5.4.4 Comparison with reference publications on WSJ0-2mix

Table 5.18 shows an overview of separation results on the WSJ0-2mix database. The trained models here are slightly worse than the reference systems. The main reasons are that all systems here use the same potentially suboptimal hyperparameters to facilitate comparison. Moreover, we here avoided particular tricks geared towards the WSJ0-2mix database such as loss masking. Additionally, a subsequent mask refinement network (Row 2) and end-to-end training (Row 3) were avoided in the baselines in this work as well.

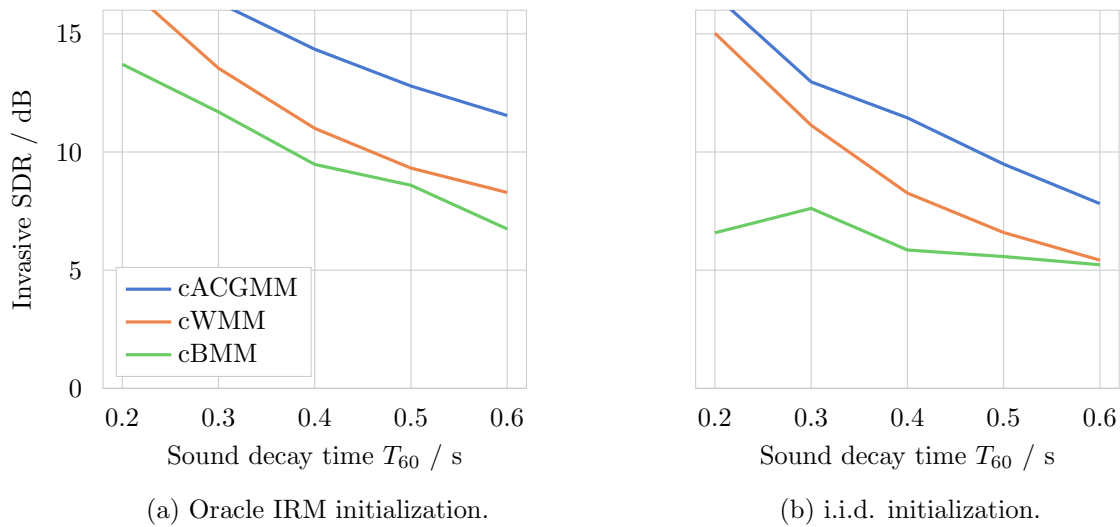


Figure 5.4: Performance of different spatial models on a limited number of examples (here 10) and different sound decay times.

5.5 Probabilistic spatial mixture models

To be able to draw reliable conclusions from experiments with integration models it is important to ensure that the baseline models, which also form the integration components, are well optimized independently. Consequently, this chapter first compares different observation models. Based on that, the cACGMM is selected for further experiments. Parameters, such as the number of classes, different initializations, and permutation alignment solvers are subsequently compared. For simplicity, all beamforming experiments within this chapter use Souden’s MVDR variant without a rank-one approximation (compare Section 3.3.2.3). The distortion matrix used in the beamforming algorithm consists of noise-plus-interference spatial correlations as in Equation 3.52. Experiments comparing different beamforming variants with a fixed clustering algorithm can be found in Section 5.6.

5.5.1 Type of the spatial observation model

First of all, we compare different spatial observation models for spatial clustering-based BSS. Similarly to the experiments in [111], we compare a cWMM, a cBMM, and a cACGMM for different reverberation conditions. In this comparison, all models use a time-dependent mixture weight and an additional noise class. Figure 5.4a shows invasive SDR results for spatial clustering models for each of the three observation models initialized with the oracle ideal ratio mask (IRM). Each data point summarizes ten mixtures with a given sound decay time. To simplify the experiment, ten examples from the WSJ-BSS database were selected and mixed with RIRs with varying reverberation.

Under these oracle conditions, the cACGMM clearly performs best on all depicted reverberation times. The cWMM, which is a special case of the cBMM and, thus, has less class-specific parameters, results in better scores than the cBMM. These results seem to contradict the

findings reported by Ito et al. in [96] and [111]. However, the cited references have to be read with care: [96] only reports single-speaker results for which the cWMM baseline uses a diffuse noise assumption (concentration parameter is zero) and the cBMM again uses a diffuse noise assumption (Bingham parameter matrix is the zero matrix). [111] reports performance differences between the cBMM and the cWMM in Figure 2. However, they obtained these results with $D = 2$ microphones for which, in theory, the complex Bingham distribution and the complex Watson distribution coincide:

$$\begin{aligned}
\mathcal{CB}(\tilde{\mathbf{y}}; \mathbf{B}) &\stackrel{\mathbf{B} \text{ Hermitian}}{=} \mathcal{CB}(\tilde{\mathbf{y}}; \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H) \\
&\stackrel{\mathbf{U} \text{ unitary}}{=} \mathcal{CB}(\mathbf{U}\tilde{\mathbf{y}}; \mathbf{\Lambda}) \\
&\stackrel{D=2}{=} \mathcal{CB}(\mathbf{U}\tilde{\mathbf{y}}; \text{diag}((\lambda_1, \lambda_2)^T)) \\
&\stackrel{\text{Appendix A.1.2}}{=} \mathcal{CB}(\mathbf{U}\tilde{\mathbf{y}}; \text{diag}((\lambda_1 - \lambda_2, 0)^T)) \\
&\stackrel{\mathbf{U} \text{ unitary}}{=} \mathcal{CB}(\tilde{\mathbf{y}}; (\lambda_1 - \lambda_2)\mathbf{u}_1\mathbf{u}_1^H) \\
&= \mathcal{CW}(\tilde{\mathbf{y}}; \mathbf{u}_1, (\lambda_1 - \lambda_2)), \tag{5.1}
\end{aligned}$$

where \mathbf{u}_1 is the first column of \mathbf{U} .

Nevertheless, the findings in terms of the superiority of the cACGMM here agree with [111]. In [106, Section 4] we reported that the implementation of the TV-cGMM is slightly more robust than the implementation of the cACGMM. Without further proof, these differences were traced back to small constants for stability improvements (informally called epsilons). After carefully selecting robust algorithms for the intermediate linear algebra operations without these constants, the TV-cGMM implementation and the cACGMM implementation are now both numerically stable and their results coincide. These findings are in line with the identity proof in the appendix of [111].

Figure 5.4b shows invasive SDR results with an i.i.d.-initialized clustering model. First of all, it becomes evident that the cBMM relies much more on an initialization closer to ideal masks. However, the results also indicate that gains over the cBMM diminish for higher reverberation times. Moreover, by comparing Figure 5.4a and Figure 5.4b, it is expected that the performance of a cascade system, i.e., using a neural network to initialize the mixture model, is likely to fall in between the i.i.d. initialization and the IRM initialization results.

Table 5.19 compares the aforementioned spatial clustering models on the WSJ-BSS database. The key findings are that all models dramatically profit from inline permutation alignment (PA) and again, as already anticipated because of Figure 5.4, the cACGMM performs best. Inline permutation alignment refers to applying a permutation alignment solver after each E-step so that disagreement between the model parameters across frequencies is avoided early on. This nuance is only prevalent when there is at least some coupling between different frequency bins (here due to the time-dependent mixture weights), otherwise one final permutation alignment step coincides with inline permutation alignment. The oracle IRM initialization results can be seen as an upper limit. Interestingly, the cWMM results and the cACGMM results only lack about 1 dB behind the oracle initialization in terms of invasive SDR suggesting that, on the given database, the cACGMM system already serves as a very strong baseline.

Table 5.19: Comparison of different spatial models with and without inline permutation alignment on the WSJ-BSS test set. The results are reported with an i.i.d. initialization and with an oracle IRM initialization.

Latent model	Inline PA	BSS-Eval SDR / dB		Invasive SDR / dB	
		i.i.d. init.	Oracle IRM init.	i.i.d. init.	Oracle IRM init.
cWMM	✗	9.25	11.24	11.23	13.25
cWMM	✓	10.40	11.26	12.33	13.27
cBMM	✗	5.39	10.32	7.30	13.09
cBMM	✓	7.25	10.41	9.68	13.19
cACGMM	✗	11.23	12.92	14.18	16.21
cACGMM	✓	12.17	12.92	15.28	16.21

Findings

To finalize this section, we can draw the following conclusions:

- The cACGMM is the most robust clustering model in all conditions.
- The cBMM is most dependent on a proper initialization.
- All latent models profit greatly from an inline PA.

In summary, the cACGMM yields on average the best performance and, consequently, will be analyzed in more detail and serve as a baseline for all experiments in the following.

5.5.2 Parameter choice for the cACGMM

The credibility of an evaluation largely depends on the choice of the baseline system and how much care was taken to tune it. Thus, we spend this section on establishing a sophisticated baseline by identifying how to best set up a cACGMM.

First of all, Table 5.20 shows source separation results with and without an additional noise class. While the gain from an additional noise class is limited for a neural network-based approach (compare, e.g., PIT results in Table 5.17), an additional noise class is crucial for spatial clustering models. The main reason is that the spatial clustering model needs to assign each time-frequency bin to effectively one class. Even the low-power observations with spurious phase information need to be assigned to one class. If these are assigned to a speaker class, the summary statistics of that class are less reliable and, consequently, the overall performance drops. Thus, all future spatial clustering models analyzed here make use of an additional class to capture all non-speaker time-frequency bins.

Table 5.20: Comparison of a cACGMM with different numbers of classes. $K' = K + 1$ classes indicates that there is an additional noise class for a two-speaker scenario. The cACGMM was initialized by sampling each entry in the affiliation mask i.i.d. from a uniform Dirichlet distribution.

Classes	SDR / dB				PESQ		STOI	
	BSS-Eval		Invasive					
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
$K' = K$	3.96	4.36	5.51	5.92	1.83	1.71	0.57	0.68
$K' = K + 1$	11.20	11.25	14.12	14.19	2.23	2.05	0.67	0.82

Table 5.21 lists source separation results for different initializations with and without inline permutation alignment and different final permutation alignments. Here, *oracle* refers to oracle permutation alignment, while the tick mark refers to a permutation alignment solver variant introduced by Tran Vu [81, Section 5.6]. The flag initialization divides the length of the signal into segments of T/K' length. Each segment is active for one of the classes. One segment, which is likely to become the noise class, is split so that half of it is placed at the start and half of it is placed at the end of the mixture. All inactive areas are set to a tiny float value. The reasoning behind this is to use human-specified prior knowledge to initialize the mixture models. By initializing this way, an initial frequency permutation problem is somewhat reduced. This can be observed by inspecting Row 1 in Table 5.21: even without any permutation alignment a reasonable source separation performance is possible. In contrast, an i.i.d. initialization as in Row 5 heavily relies on permutation alignment. Although flag initialization can lead to very high invasive SDR (e.g., 14.85 dB) without inline permutation alignments, all future experiments will use an i.i.d. initialization. This decision results from the observation that, at least when a time-dependent mixture weight is chosen, the flag initialization should not be able to converge. However, since the inactive parts are chosen to be non-zero, the model recovers from this very tiny numerical value in the order of $1 \cdot 10^{-10}$ and we would like to avoid this numeric oddity in the following.

Table 5.22 compares how different mixture weight types influence the separation result. Here, $1/K'$ is a constant mixture weight, all others are varying for the given index, e.g., π_k is an only-speaker-dependent mixture weight. First of all, we observe that a time-dependent mixture weight $\pi_{k,t}$ with inline permutation alignment leads to best separation results, for example, in terms of BSS-Eval SDR. A time-dependent mixture weight leads to better spectral continuity. If inline permutation alignment is applied, disagreement between the mixture weight and the estimated parameters is resolved early on (Row 6). Reversely, a SDR drop of more than 1 dB occurs, when inline permutation alignment is not used (Row 5). A similar effect, albeit somewhat smaller, can be observed when comparing Row 2 with Row 3. Interestingly, a constant mixture weight $1/K'$ performs better than a frequency- and speaker-dependent mixture weight, at least on this particular database.

The convergence of the cACGMM mostly depends on the initialization. Figure 5.5 illustrates the convergence behavior for two different initialization variants and showcases the effect of an additional inline permutation alignment. Table A.5 in the appendix lists more detailed evaluation results for the convergence behavior. Based on the SDRs for different numbers

Table 5.21: Comparison initializations and permutation alignment methods for a cACGMM.

Initialization	Inline PA	Final PA	SDR / dB				PESQ		STOI	
			BSS-Eval		Invasive					
			Dev	Test	Dev	Test	Dev	Test	Dev	Test
flag	✗	✗	10.14	10.21	12.97	13.23	2.10	1.95	0.64	0.78
flag	✗	✓	11.46	11.71	14.49	14.85	2.19	2.05	0.66	0.81
flag	✗	oracle	12.40	12.33	15.78	15.66	2.26	2.08	0.68	0.83
flag	✓	✓	11.42	11.76	14.37	14.89	2.18	2.04	0.66	0.81
i.i.d.	✗	✗	0.24	0.12	1.79	1.62	1.57	1.44	0.51	0.61
i.i.d.	✗	✓	11.20	11.25	14.12	14.19	2.23	2.05	0.67	0.82
i.i.d.	✗	oracle	11.16	11.30	14.28	14.39	2.23	2.05	0.68	0.82
i.i.d.	✓	✓	12.38	12.22	15.53	15.35	2.26	2.08	0.68	0.82
oracle IRM	✗	✗	13.23	12.92	16.55	16.20	2.30	2.10	0.69	0.83
oracle IRM	✗	✓	13.24	12.92	16.56	16.21	2.30	2.10	0.69	0.83
oracle IRM	✗	oracle	13.04	12.82	16.50	16.21	2.29	2.09	0.69	0.83
oracle IRM	✓	✓	13.25	12.92	16.56	16.21	2.30	2.10	0.69	0.83

Table 5.22: Comparison of a cACGMM with different mixture weight types. The cACGMM with $K' = K + 1$ classes was initialized by sampling each entry in the affiliation mask i.i.d. from a uniform distribution with subsequent normalization to sum up to one.

Weight type	Inline PA	SDR / dB				PESQ		STOI	
		BSS-Eval		Invasive					
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
$1/K'$	✗	11.20	11.30	14.14	14.25	2.22	2.05	0.67	0.82
π_k	✗	11.27	11.29	14.20	14.25	2.23	2.05	0.68	0.82
π_k	✓	11.60	11.51	14.42	14.35	2.24	2.06	0.68	0.82
$\pi_{k,f}$	✗	10.77	10.88	13.45	13.61	2.21	2.04	0.67	0.81
$\pi_{k,t}$	✗	11.20	11.25	14.12	14.19	2.23	2.05	0.67	0.82
$\pi_{k,t}$	✓	12.38	12.22	15.53	15.35	2.26	2.08	0.68	0.82

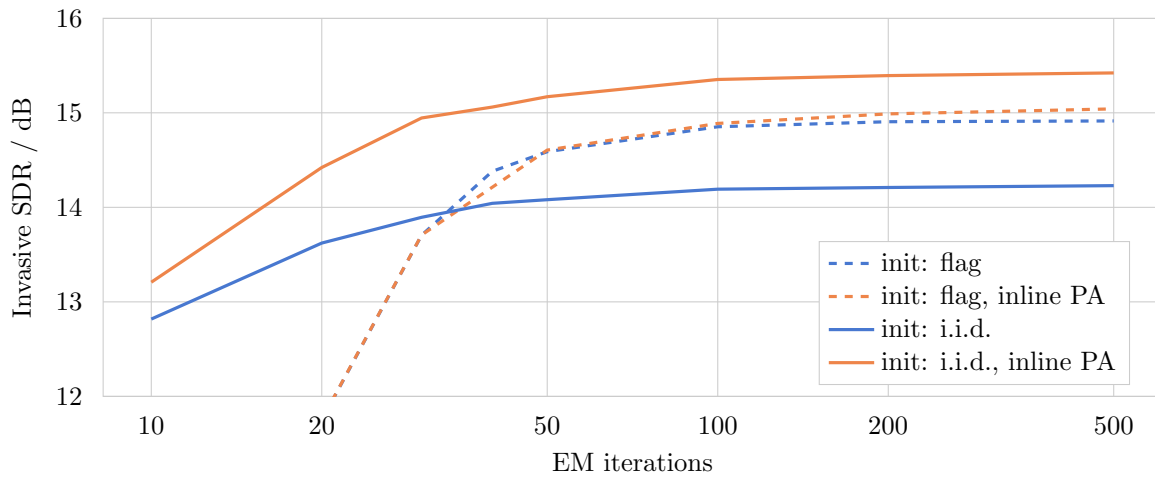


Figure 5.5: Convergence behavior of the cACGMM for different initialization variants and depending on usage of an inline permutation aligner. The iteration axis is logarithmically scaled.

of iterations, we may conclude that the system is already rather saturated at 100 steps. Consequentially, to limit computational costs, all further experiments will be limited to 100 iterations.

Findings

The findings in this section can be summarized as follows:

- Inline permutation alignment, albeit a simple change, is an important tuning method which has not been published anywhere yet.
- Approximately 100 EM-iterations are sufficient for the analyzed spatial clustering algorithms to approximately converge in terms of invasive SDR.
- A time- and speaker-dependent mixture weight results in the best performance.

5.6 Source extraction

This section evaluates different beamforming variants to extract each source from the mixture signal. Since this section covers a wide range of beamformers, different RTF or covariance matrix approximations and other variants, we aim at highlighting key findings concerning source separation and eliminating variants early on. For completeness' sake, a comparison of masking and beamforming can be found later in Table 5.29.

First of all, it is important to note that all beamformers were evaluated with a pretrained acoustic model on single-speaker noisy recordings (i.e., trained on noisy images). Therefore, one expects that a normalization such as BAN helps to match the training conditions. Figure 5.6 shows WERs over different SDR variants. All covariance matrices were obtained with the mask normalization as in Equation 3.50.

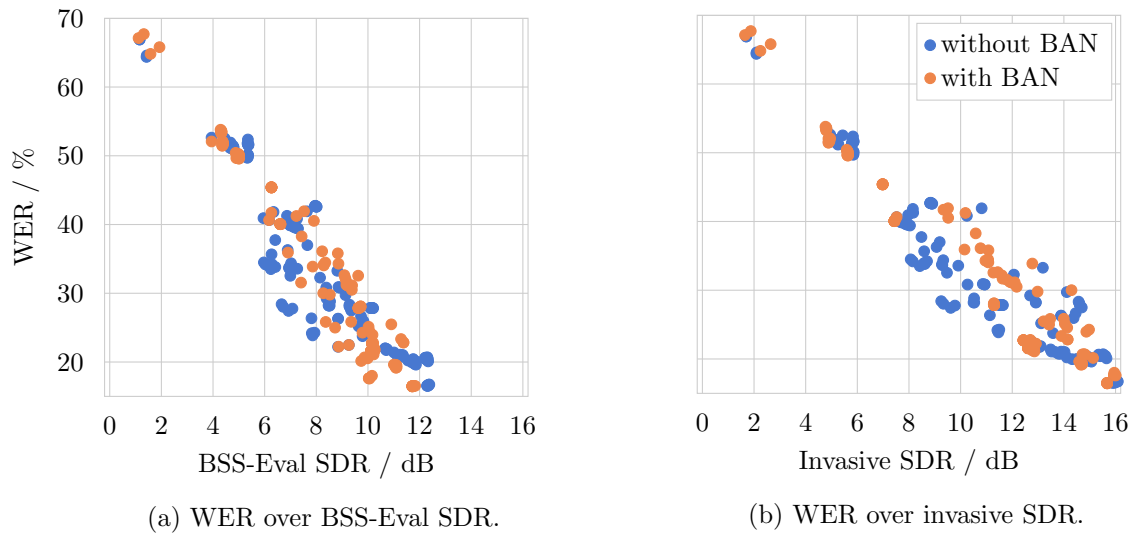


Figure 5.6: Scatter plot of dependency of WER on different performance metrics for a variety of different beamformers. The masks to obtain covariance matrices are either oracle masks or posterior masks obtained with a cACGMM. Therefore, the main purpose of these figures is to demonstrate the correlation of WER with different SDR variants.

In general, both metrics provide a good indication of the quality of the front-end for speech recognition. However, for high SDR values, invasive SDR seems to predict WER slightly more accurately. Further, the figures do not show any evidence whether BAN should or should not be used in general. Therefore, all future results select BAN or no BAN based on the development set WERs instead of simply reporting, e.g., only BAN results. Although we now observed that SDR is a good predictor for WER when using a beamformer, it is not said that the correlation holds just as well for other systems. Finally, an evaluation with pretrained AMs as well as with matched AMs is inevitable and, consequently, is presented in Table 5.31 and Table 5.32.

Next, it is worth analyzing which distortion matrix definition to choose. All beamforming variants besides the principal component analysis (PCA) beamformer require some kind of distortion matrix. In a multi-source scenario, this can either be the noise matrix or the noise-plus-interference matrix as in Equation 3.52. Figure 5.7 shows SDR values as well as WERs for different beamformers in their standard configuration, i.e., RTF estimates (used as intermediate vectors $\mathbf{d}_{k,f}$) are obtained using a PCA when applicable. All masks used to obtain the covariance matrices stem from a cACGMM clustering. The PCA beamformer results (two upmost bars) do not change (up to randomness in the cACGMM initialization), since the PCA beamformer does not depend on a distortion matrix. It turns out that results are substantially better when using a noise-plus-interference matrix instead of a noise-only matrix. Based on these results, all further investigation will be conducted with a noise-plus-interference matrix as a distortion matrix.

Finally, we can analyze different beamformers with different ways to either extract the RTF estimate or to approximate the target covariance matrix. The results are presented in Table 5.23 and contrast either using oracle IRMs or posterior masks provided by a

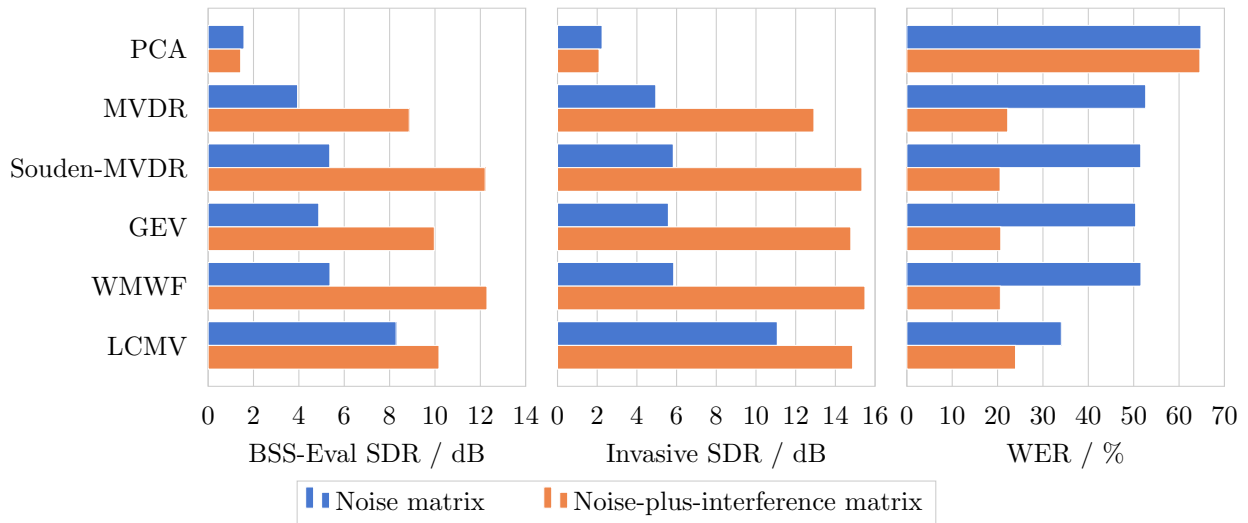


Figure 5.7: Comparison of beamformers using the noise matrix and using the noise-plus-interference matrix as distortion matrix. The MVDR and LCMV beamformer use a PCA to extract the RTF estimate. BAN is enabled or disabled, the distortion weight of the WMWF is selected, as well as the leakage parameter of the LCMV is selected based on minimal WER on the development data. Masks are obtained with an i.i.d.-initialized cACGMM.

cACGMM. The preprocessing column defines how the RTF estimate is obtained from the speech covariance matrix $\Phi_{\mathbf{x}\mathbf{x},k,f}$ for MVDR and LCMV beamforming. Moreover, it defines if and how the speech covariance matrix $\Phi_{\mathbf{x}\mathbf{x},k,f}$ was processed before using it in the beamforming algorithm:

- PCA: The RTF is obtained with a PCA decomposition as in [140, Equation 26].
- GEV: The RTF is obtained with a GEV decomposition and appropriate rescaling using the noise covariance matrix as in [140, Equation 27].
- PCA \rightarrow Rank 1: The target covariance matrix $\Phi_{\mathbf{x}\mathbf{x},k,f}$ is enforced to have rank one as in [140, Equation 25] combined with the PCA decomposition in [140, Equation 26].
- GEV \rightarrow Rank 1: The target covariance matrix $\Phi_{\mathbf{x}\mathbf{x},k,f}$ is enforced to have rank one as in [140, Equation 25] combined with the GEV decomposition in [140, Equation 27].

In all cases, applying GEV first and then creating a rank-one matrix by calculating the outer product of the RTF with itself is the best preprocessing in terms of SDR as well as WER. Whether BAN is to be used is again selected based on development set WER (to be precise, selected on IRM WER results). It turns out that a BAN postfilter is beneficial in most cases. The distortion weight μ for the WMWF, as well as the leakage parameter ϵ , is chosen such that WER is minimized on the development set. Overall, the Souden-MVDR, the GEV as well as the WMWF each with a GEV \rightarrow Rank 1 matrix approximation yield the best performance. In the following, the Souden-MVDR is preferred since it does not need further tuning of yet another hyperparameter such as the distortion weight. Although the LCMV was specifically derived for the multi-speaker case, it turned out to be slightly less effective than the previously mentioned approaches although it is in this form already heavily tuned: it contains

Table 5.23: Comparison of different beamformers. Each metric is stated with beamforming based on an IRM oracle and with posterior masks from a cACGMM latent model. The usage of BAN, the optimal weight μ or the optimal leakage parameter ϵ is decided based on IRM WERs results on the development set. Metrics shown are on the test set.

Preprocessing RTF/ $\Phi_{\mathbf{x}\mathbf{x},k,f}$	Beamformer estimation	BAN	Parameter μ or ϵ	BSS-Eval SDR / dB		WER / %	
				Oracle	cACGMM	Oracle	cACGMM
	PCA	✗		1.16	1.43	66.91	64.59
PCA	MVDR	✓		9.26	8.87	22.48	22.23
GEV	MVDR	✓		10.03	9.99	17.63	20.53
	Souden-MVDR	✗		12.26	12.21	20.54	20.56
PCA → Rank 1	Souden-MVDR	✗		10.67	10.71	21.98	21.73
GEV → Rank 1	Souden-MVDR	✓		11.72	11.08	16.46	19.31
	GEV	✓		10.03	9.97	17.63	20.70
PCA → Rank 1	GEV	✓		10.17	10.19	21.67	21.97
GEV → Rank 1	GEV	✓		11.83	11.00	16.53	19.63
	WMWF	✗	0.20	12.28	12.25	20.55	20.45
PCA → Rank 1	WMWF	✗	0.80	11.69	11.26	19.97	21.07
GEV → Rank 1	WMWF	✓	0.60	11.72	11.11	16.49	19.22
PCA	LCMV	✓	0.01	10.05	9.79	24.56	24.27
GEV	LCMV	✓	0.00	10.06	9.95	17.60	20.65

the interference-plus-noise matrix as distortion matrix, the leakage parameter ϵ is carefully selected ¹¹ and a scaled GEV is used to extract the RTF per speaker.

Each metric is stated with an IRM oracle and with a cACGMM latent model (alternating columns). It can be observed that improvements based on the choice of the beamformer on IRMs translate to improvements with imperfect cACGMM posteriors. The three top-performing beamformers are similarly susceptible to mask quality. One can observe that this dependency is similar for all beamforming variants involving GEV decomposition in some way.

Although the GEV and the GEV with an RTF extraction also using the GEV should theoretically coincide up to the absolute phase, we observe a rather dramatic gain using the latter. This indicates that it is well worth investigating further how the absolute phase should be determined. Initial results on phase normalization for GEV beamforming vectors can be found in [149, Section V].

Findings

The findings in this section can be summarized as follows:

- Selecting the right beamforming algorithm is data-dependent and should, therefore, be formalized by selecting an algorithm on a separate development set.
- All beamforming variants profit from using a noise-plus-interference covariance matrix instead of a noise-only covariance matrix. Although this might be an obvious finding, it is rarely clearly stated.
- Souden’s MVDR variant with GEV \rightarrow Rank 1 preprocessing leads to competitive WERs while avoiding yet another tuning parameter.

5.7 Integration of neural networks and probabilistic graphical models

In this section, we evaluate different integration variants to better understand how to choose model parameters, and how they interact with other system components, such as permutation alignment. In Section 5.7.1 we analyze weak integration approaches, in which a DNN provides the initialization for subsequent spatial clustering. In Section 5.7.2 we focus on tight integration approaches, namely approaches in which spatial and spectral information both influence the EM-algorithm throughout the iterations. Finally, in Section 5.7.3 we address the valid critique that multi-channel features for, e.g., a DC network are also an integration of both feature types.

¹¹ Tuning the leakage parameter is an idea which I became aware of due to a discussion with Sharon Gannot during his visit in Paderborn. Although changing the leakage parameter was detrimental or led to marginal improvements in this case, others report more robust beamforming with $\epsilon > 0$.

Table 5.24: Different weak integration results depending on the usage of inline permutation alignment and final permutation alignment. Each row consists of a cACGMM as a spatial clustering model which is either initialized randomly or with a spectral model.

Encoder	Initialization	Inline PA	Final PA	SDR / dB			
				BSS-Eval		Invasive	
				Dev	Test	Dev	Test
	i.i.d.	✗	✓	9.95	10.11	13.11	13.39
	i.i.d.	✓	✓	11.13	11.08	14.72	14.66
DC	k-means	✗	✗	11.95	11.76	15.78	15.56
DC	k-means	✗	✓	12.01	11.79	15.86	15.62
DC	k-means	✓	✓	12.01	11.78	15.87	15.61
DAN	k-means	✗	✗	11.53	11.45	15.27	15.18
DAN	k-means	✗	✓	11.63	11.41	15.51	15.24
DAN	k-means	✓	✓	11.59	11.34	15.41	15.09
PIT		✗	✗	11.96	11.72	15.75	15.51
PIT		✗	✓	12.02	11.74	15.83	15.54
PIT		✓	✓	12.02	11.75	15.83	15.54

5.7.1 Weak integration: A cascade approach

Table 5.24 shows separation results with a cACGMM as a spatial clustering model. For each experiment, the speakers are extracted with a Souden-MVDR beamformer with a GEV \rightarrow Rank 1 preprocessing and subsequent BAN filter in accordance with the findings reported in Table 5.23. The first two rows contain the baseline cACGMM with an i.i.d. initialization. In any case, the spatial model uses a time- and speaker-dependent mixture weight. Particularly Row 2 serves as a very competitive baseline with an additional inline permutation alignment. All other rows present results with a DNN providing an initialization for subsequent spatial clustering. In the case of DC and DANs, the embedding vectors are clustered using k-means and the resulting posterior mask then serves as the initialization for the cACGMM. All DNNs are trained with $K' = K + 1$ classes, i.e., including an additional noise class, to match the cACGMM.

First of all, we observe that the initialization with any of the listed DNNs improves the performance compared to Row 2. Interestingly, the inline permutation alignment, while it improved the performance of the i.i.d. initialized cACGMM, did not improve the performance of weak integration systems. In some cases, e.g., comparing Row 7 with Row 8, the inline permutation alignment was even detrimental. However, the final permutation alignment consistently improved the results in comparison to the systems without permutation alignment. Consequently, we may conclude that the initialization with a DNN almost completely avoids the permutation problem. All further comparisons with weak integration approaches will, therefore, contain a final permutation alignment but omit a permutation alignment step

in each EM-iteration. The best results with a weak integration system are obtained either with a DC encoder and subsequent k-means clustering on the embedding vectors, or with a PIT network directly providing the initialization. While, at first glance, this might seem to contradict the findings in Table 5.18, where the DC system provided the lowest BSS-Eval SDR results, the DC embeddings are not directly used for reconstruction here. Much more indirectly, they only serve as initialization to the spatial clustering system and possible artifacts, which are detrimental when using the masks directly for source extraction, are covered up by, e.g., the mask-based beamforming.

Findings

To finalize this section, we can draw the following conclusions:

- Proper initialization of a spatial mixture model with a DNN, here called weak integration, renders an additional inline permutation alignment obsolete.
- Best results are obtained with a DC encoder or an initialization based on a PIT system. The advantage of the weak integration using a DC encoder is that the number of speakers at inference time is independent of the number of speakers during training. However, the weak integration with a PIT network is conceptually easier to implement.

5.7.2 Strong integration

In this section, we evaluate different strong integration variants consisting of integrated clustering models which are comprised of a spatial and a spectral observation model on the WSJ-BSS database. To do so, we first compare different latent models and different embedding networks, then the emphasis is put on the choice of a fixed concentration or scale parameter.

The reported results in Table 5.25 are obtained with embedding networks trained to form $K' = K + 1 = 3$ distinct clusters during training on $K = 2$ speaker mixtures. Once posterior masks are obtained from the integrated clustering model, the individual speakers are extracted using Souden’s MVDR formulation (compare Section 3.3.2.3) with a $\text{GEV} \rightarrow \text{Rank 1}$ preprocessing of the target speaker covariance matrix (compare Section 5.6). The optimal trade-off parameter $\sigma^2 = 1/\lambda$ in the case of a GcACGMM and κ in the case of a vMFcACGMM is selected based on development set invasive SDR. Each group in Table 5.25 reports results of a unique combination of an encoder network and a latent model. We first observe that a preclustering of the embedding vectors with k-means improves the performance significantly. Consequently, although all information is available to the integrated clustering model, careful preclustering is still inevitable. Furthermore, although all models iterated 100 EM-steps, better initialization leads to faster convergence and, thus, a smaller number of iterations is necessary on average. Just as we observed for weak integration models in Table 5.24 the inline permutation alignment is now obsolete given a k-means preclustering. However, it is worth noting that a final permutation alignment is still performed and separate experiments without a final permutation alignment are not shown.

Table 5.25: Different strong integration results depending on usage of inline permutation alignment and whether the integration model is initialized randomly or from a k-means result of the predecessor model. The tuning parameter κ or σ^2 is selected to maximize invasive SDR on the development set.

Encoder	Initialization	Latent model	Parameter	Inline PA	SDR / dB			
					BSS-Eval		Invasive	
					Dev	Test	Dev	Test
DC	i.i.d.	GcACGMM	$\sigma^2 = 1/20$	✗	10.85	10.93	15.19	15.32
DC	k-means	GcACGMM	$\sigma^2 = 1/8$	✗	11.74	11.52	16.28	15.98
DC	k-means	GcACGMM	$\sigma^2 = 1/8$	✓	11.74	11.52	16.28	15.98
DC	i.i.d.	vMFCACGMM	$\kappa = 20$	✗	10.80	10.76	15.19	15.06
DC	k-means	vMFCACGMM	$\kappa = 5$	✗	11.80	11.58	16.30	16.00
DC	k-means	vMFCACGMM	$\kappa = 5$	✓	11.82	11.59	16.34	16.02
DAN	i.i.d.	GcACGMM	$\sigma^2 = 1$	✗	10.63	10.68	14.99	14.97
DAN	k-means	GcACGMM	$\sigma^2 = 5/2$	✗	11.46	11.39	15.87	15.70
DAN	k-means	GcACGMM	$\sigma^2 = 5/2$	✓	11.45	11.39	15.87	15.71

Figure 5.8 shows how the particular choice of the trade-off parameter $\sigma^2 = 1/\lambda$ or κ influences the separation result under the same evaluation conditions as before. It becomes evident that i.i.d. initialized models heavily depend on the particular choice of the trade-off parameter with no clear explanation for the particular shape of the curves. In contrast, the variability of the results over the trade-off parameter given k-means preclustering is much smaller and almost disappears at the current scaling. Nevertheless, at least for the DC-based system, a small peak around $\kappa = 5$ is visible which is similarly located for the development set as well as for the test set. We may, therefore, conclude that the particular choice is rather stable and obtaining the trade-off parameter on the development set is a reasonable strategy. Not shown in this figure is the observation that the maximum slightly moves towards lower values for the trade-off parameter with more poorly trained embedding models: more emphasis is put on the spatial model when the spectral model is less reliable.

Figure 5.9 shows a closer look of the dependency of the separation performance and the trade-off parameter for a DC-based system with k-means preclustering and inline permutation alignment (in orange). The results are contrasted with the corresponding weak integration model (in blue). We observe that the maximum invasive SDR nicely coincides with the minimum WER both on the development as well as on the test set. In contrast, the BSS-Eval SDR does not show similar behavior and seems to be a less predictive indicator for WER. Arguably, the WER reduction with a tight integration approach over the weak integration approach is limited given the added complexity of the model and the need to properly select the trade-off parameter. However, if an optimal separation performance is required the tight integration approach is a valid choice.

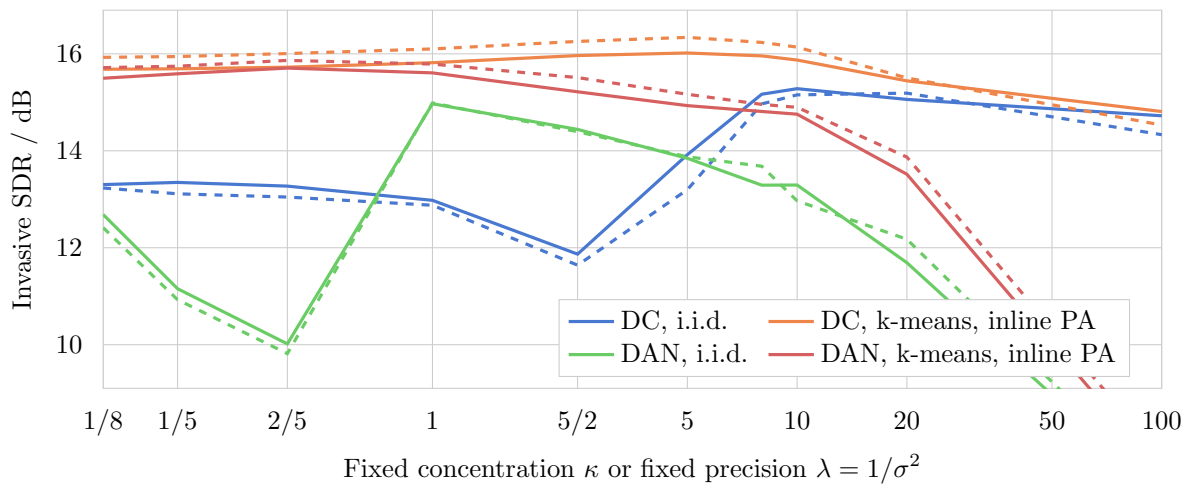


Figure 5.8: Dependency of the strong integration models on a fixed concentration parameter or fixed covariance parameter. The horizontal axis shows precision $\lambda = 1/\sigma^2$ instead of variance to allow better comparison with the concentration κ . Dashed lines indicate development set results.

Findings

To finalize this section, we can draw the following conclusions:

- Selecting an optimal trade-off parameter is an added complexity.
- Preclustering of the embedding vectors leads to significant improvements.
- An additional inline permutation alignment is obsolete given proper preclustering.
- The differences between the different spectral observation models are minor.

5.7.3 Comparison of integration models with single-/multi-channel encoder

Similarly to the evaluation in [51, Table V] we here analyze how a multi-channel encoder, in this case, a multi-channel DC (listed as *Spatial-DC*) network, improves the separation with and without integration approaches. All results listed in Table 5.26 are again reported on the WSJ-BSS database with the same conditions as in the previous two sections. In all previous experiments, we extracted embedding vectors from the spectrogram of a single reference channel. To get a better understanding of how additional channels improve the embedding vectors, we analyze different channel stack modes:

- Reference: The DC embedding network operates on a single fixed reference channel or the Spatial-DC network operates on two fixed reference channels.
- D channels: The embedding network extracts embedding vectors independently on each channel. Before further processing, the embedding vectors are stacked so that the k-means algorithm operates on $D \cdot E$ dimensional data.

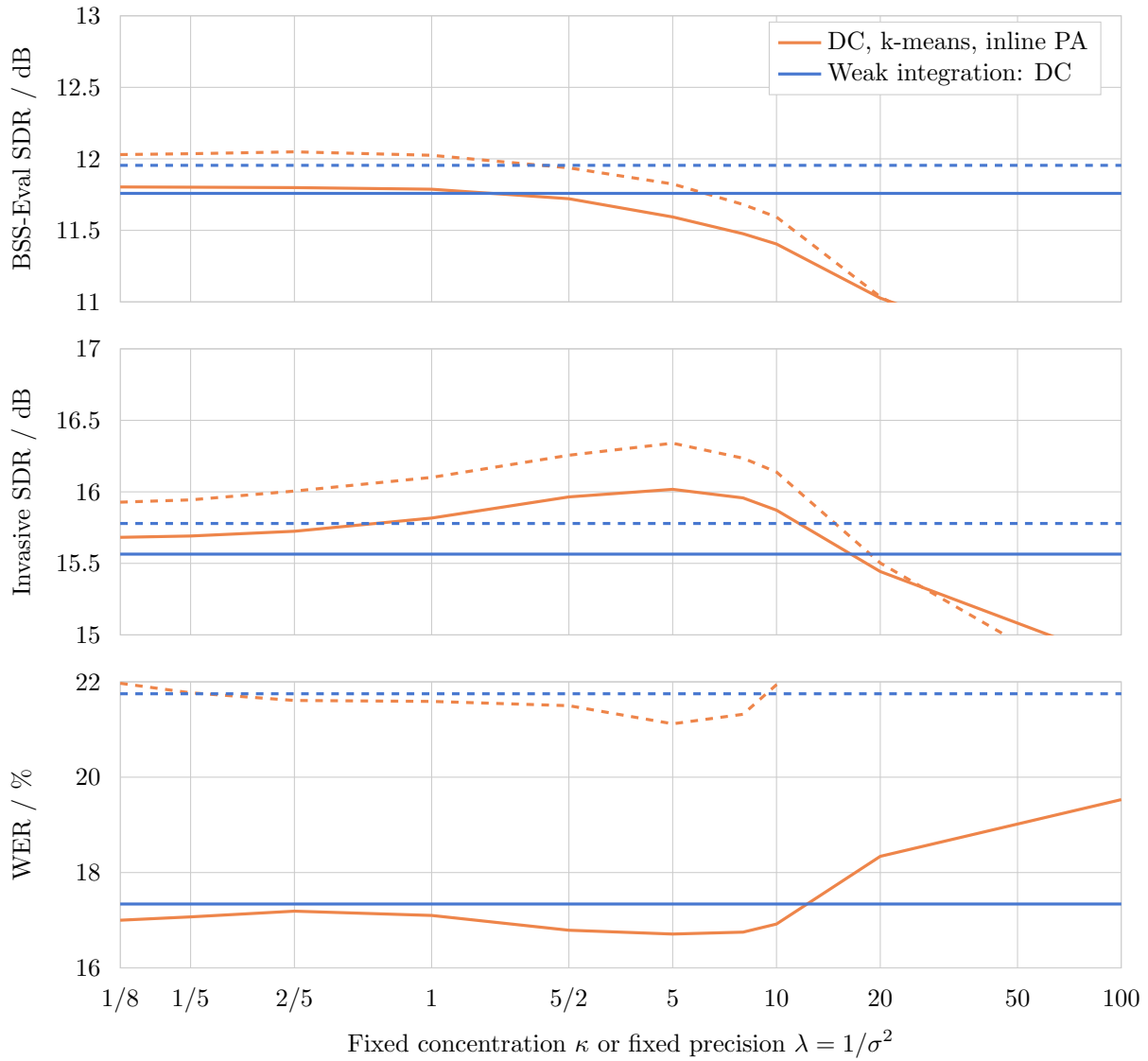


Figure 5.9: Dependency of a strong integration model on a fixed concentration parameter in comparison to a weak integration model. The horizontal axis shows precision $\lambda = 1/\sigma^2$ instead of variance to allow better comparison with the concentration κ . Dashed lines indicate development set results. All results are obtained on the WSJ-BSS database.

Table 5.26: Comparison of different channel stack modes for DC on the WSJ-BSS test set. The embedding vectors are stacked before entering the latent model (here k-means either directly or as a preclustering step). Latent model parameters are selected based on best invasive SDR result on the development set. Results reported are on the test set.

Encoder	Latent model	Channel mode	Parameter	SDR / dB	
				BSS-Eval	Invasive
DC		reference		10.28	14.65
DC		D channels		10.45	14.85
DC	cACGMM	reference		11.78	15.61
DC	cACGMM	D channels		11.79	15.62
DC	vMFcACGMM	reference	$\kappa = 5$	11.59	16.02
DC	vMFcACGMM	D channels	$\kappa = 5$	11.61	16.04
Spatial-DC		reference		11.00	15.45
Spatial-DC		$D - 1$ pairs		11.19	15.65
Spatial-DC	cACGMM	reference		11.80	15.64
Spatial-DC	cACGMM	$D - 1$ pairs		11.80	15.64
Spatial-DC	vMFcACGMM	reference	$\kappa = 5$	11.62	16.08
Spatial-DC	vMFcACGMM	$D - 1$ pairs	$\kappa = 5$	11.64	16.11

- $D - 1$ pairs: The Spatial-DC embedding network extracts embedding vectors independently on $D - 1$ pairs before these embeddings get stacked. Just as in [49] we obtain $D - 1$ pairs by selecting a single reference channel and then calculate inter-channel features of each other channel against this channel.

First of all, the results are in agreement with [49] in the sense that the Spatial-DC model outperforms a single-channel DC model. Moreover, the single-channel DC model also leads to better results when independently applied to each microphone channel. This can be explained with the cross-channel variability even in a rather compact microphone array although the single-channel DC network does not have access to directional information: variability across channels is somewhat averaged out by clustering the stacked embedding vectors. The key finding, however, is that all integration variants profit at least to some degree from a multi-channel embedding network. Nevertheless, once any form of integration is used, stacking embedding vectors almost does not change the results at all: the cross-channel variability is small enough for the additional information to be useless in comparison to the information available to the spatial observation model. The best results in terms of invasive SDR are obtained with a multi-channel encoder operating on all $D - 1$ pairs and a vMFcACGMM tight integration model.

Table 5.27: Results with and without supervision on the WSJ-BSS database.

Encoder	Unsupervised	Latent model	Channel mode	SDR / dB				WER / %	
				BSS-Eval		Invasive			
				Dev	Test	Dev	Test	Dev	Test
	✓	cACGMM	D channels	11.10	11.10	14.67	14.68	25.55	19.51
DC	✗		reference	9.93	10.28	14.32	14.65	26.80	19.79
DC	✗		D channels	10.17	10.45	14.62	14.85	25.62	19.02
DC	✗	cACGMM	reference	12.01	11.79	15.86	15.62	21.59	17.21
DC	✗	cACGMM	D channels	12.03	11.79	15.88	15.62	21.97	17.19
DC	✓		reference	10.17	10.42	13.88	14.12	28.81	21.46
DC	✓		D channels	10.31	10.57	14.08	14.30	27.67	20.52
DC	✓	cACGMM	reference	11.99	11.74	15.78	15.54	21.81	17.45
DC	✓	cACGMM	D channels	11.98	11.74	15.77	15.53	21.83	17.38

Findings

To finalize this section, the following conclusions can be drawn:

- Multi-channel DC indeed outperforms single-channel DC.
- Weak integration, as well as the tight integration approach, still profit from a multi-channel encoder although the spatial information is available directly to the spatial observation head of the integration model. Although the independence assumption between spatial and spectral features clearly does not hold here the results indicate that a possible over-confidence does not hurt overall performance.

5.8 Unsupervised training of deep clustering

This section presents evaluation results comparing supervised and unsupervised training of a DC network. To do so, we first evaluate with the WSJ-BSS database to provide results comparable to the previous sections. Then, we analyze generalizability by reporting results on WSJ-MC, a database with real recordings introduced in Section 5.2.3.

Table 5.27 lists results on the WSJ-BSS database. All results are reported with beamforming as in the previous sections. In Row 1 results of a cACGMM with i.i.d. initialization serve as a baseline relying solely on spatial features and, notably, not requiring a training phase. Row 2 – Row 5 contain supervised DC results with and without a subsequent cACGMM. As seen before, the weak integration outperforms the supervised DC and profits much less from stacking the embedding vectors of the D channels. Row 6 – Row 9 contain unsupervised DC results. The architecture of the unsupervised DC model is identical to the supervised DC

model. However, the model parameters are trained using supervision from a cACGMM as a teacher instead of ideal binary masks.

The resulting unsupervised DC model operating on only one channel in Row 6 lacks behind the unsupervised teacher in Row 1. However, when initializing the cACGMM with the k-means clustering results, obtained on the embedding vectors of the unsupervised DC system, this weak integration outperforms the teacher in Row 1. Although this might appear astonishing at first, given that the weak integration uses the same data as the teacher alone, the effect can be explained as follows: During training of the DC system, the teacher cACGMM often produces only moderate separation results but due to the fact that the separation performance is good enough on average, the training process smoothes out these variations. This is further confirmed by the fact that the unsupervised DC training requires many more steps, possibly due to conflicting gradients resulting from the posterior masks produced by the cACGMM teacher. Another plausible explanation is that the student does not have access to the spatial information and, thus, cannot reliably reproduce the errors the teacher is doing. For example, when the separation quality of the teacher is poor due to speakers being very close to each other the student would not know about this and still produce meaningful separation results. Comparing all supervised systems with all unsupervised systems, it becomes apparent that the supervised systems still outperform the unsupervised system, albeit only by a small margin. However, in a more practical application, one might pretrain a DC system with supervision on a limited training set and then fine-tune on real recordings closer to the test conditions.

The findings reported here agree with our previously reported results in [167]. In tendency, they demonstrate that the results are reproducible with freshly trained embedding networks and acoustic models. Besides the results reported here, [167] lists masking results. We did not analyze unsupervised DC in combination with a tight integration approach because that would require us to obtain a tuned trade-off parameter, which is hard to do when no metrics can be obtained in an unsupervised setting.

Table 5.28 lists the results of the aforementioned experiments on the WSJ-MC database. However, since the WSJ-MC database does not provide a separate training set, the embedding network (supervised and unsupervised) was trained on the WSJ-BSS database. First of all, we realize fairly high WERs which can be explained by the fact that the database contains British English speech whereas the acoustic model was trained on WSJ-BSS containing only American English. Nevertheless, the WER may still serve as an objective performance measure, even if they may only provide an idea of the relative ordering of the proposed systems. The BSS-Eval metrics were obtained by comparing the separation results with headset microphone signals, which is, of course, an approximation to the true source signal: It may still contain a severe portion of cross talk. Invasive SDR are not reported here since they require access to the images (speech and noise separately how they appear at the microphones) which are only available in a simulation environment.

Interestingly, this time the best BSS-Eval SDR results are obtained with a weak integration including an unsupervised DC system. Still, this result does not translate to WERs, for which the gap between both approaches is much closer. Anyhow, given that these results are reported on completely different data, we may conclude that the weak integration approaches generalize well to unseen data. When, in comparison, we inspect Row 2 and Row 6, we realize

Table 5.28: Results with and without supervision on the WSJ-MC database. Due to the nature of the database invasive SDR metrics are not available.

Encoder	Unsupervised	Latent model	Channel mode	SDR / dB		WER / %	
				BSS-Eval			
				Dev	Test	Dev	Test
	✓	cACGMM	D channels	4.25	4.92	29.28	47.25
DC	✗		reference	3.87	3.72	45.93	61.86
DC	✗		D channels	4.50	4.24	38.64	58.77
DC	✗	cACGMM	reference	4.78	4.63	26.07	46.48
DC	✗	cACGMM	D channels	4.83	4.87	26.69	44.32
DC	✓		reference	4.03	4.29	42.44	55.47
DC	✓		D channels	4.19	4.65	39.12	50.21
DC	✓	cACGMM	reference	5.45	5.67	27.11	44.79
DC	✓	cACGMM	D channels	5.54	5.67	26.35	44.83

that the unsupervised DC system without any integration performs significantly better than its supervised counterpart.

Findings

To finalize this section, we can draw the following conclusions:

- An unsupervised DC system can outperform its teacher when used in a weak integration.
- The integration approaches, in particular when comprising an unsupervised DC model generalize better to unseen data.
- Unsupervised DC alone generalizes better to unseen data when compared to its supervised counterpart.

5.9 Overview of all methods on WSJ-BSS

To summarize the findings of the previous evaluation sections and put all variants into perspective, this section compares weak integration, tight integration, and nonintegration variants. To gain further insights, we analyze splits of the dataset or operate on fewer channels and provide speech recognition results with matched training of the acoustic model.

Table 5.29 lists masking results on the WSJ-BSS database. Although the beamforming results are likely to be better, masking is closer to applications for which DC, DANs, and

Table 5.29: Summary of masking results for the WSJ-BSS test set. The acoustic model was trained on noisy source images, i.e., it did not see overlap or system artifacts during training. The tuning parameter κ or σ^2 is selected to minimize WER on the development set.

Encoder	Loss	Latent model	Parameter	Output nonlinearity	SDR / dB		WER / %
					BSS-Eval	Invasive	
		cACGMM			9.22	13.37	24.92
DC					7.06	11.18	45.14
DC		cACGMM			9.91	14.22	22.22
DC		vMFCACGMM	$\kappa = 1/8$		9.93	14.28	22.05
DAN	CE				-4.73	1.53	73.72
DAN	CE			softmax	-0.98	3.23	64.62
DAN	CE	cACGMM			7.73	11.87	31.19
DAN	CE	GcACGMM	$\sigma^2 = 8$		6.87	11.32	36.11
DAN	MSE				6.88	11.20	56.29
DAN	MSE			sigmoid	7.35	9.99	43.76
DAN	MSE	cACGMM			9.36	13.62	24.14
DAN	MSE	GcACGMM	$\sigma^2 = 8$		8.94	13.42	26.75
PIT	CE				6.67	10.03	40.59
PIT	CE	cACGMM			9.88	14.21	22.59
PIT	NPSMSE				6.59	9.45	52.75
PIT	NPSMSE	cACGMM			9.76	14.14	22.70

PIT were originally designed and for which the DAN with a reconstruction loss (MSE) and the PIT system with a reconstruction loss (NPSMSE) were trained in particular. We show DAN results, in which the network was either trained with a mask loss (CE) or with a reconstruction loss (MSE) because, at least for the experiments we performed on the WSJ0-2mix database, the CE system performed better, although the DAN was originally proposed with a reconstruction loss [46, Equation 1].

One key observation is that the cACGMM, which is an unsupervised system and which is not aware of which masks yield particularly good reconstruction, leads to better results than DC, DANs, and PIT on this particular database. One reason surely is that the spatial cues are a very important knowledge source. Additionally, the database consists of a fixed geometry without any head movement, which is helpful for the spatial clustering model but which the single-channel neural networks cannot capitalize on.

Given the simplicity, weak integration already provides significant gains. For example the weak integration, in which DC extracts embeddings and k-means an initialization for a subsequent cACGMM, reduces the WER from 24.92 % and 45.14 % down to 22.22 % from the cACGMM alone or the DC system alone, respectively. Comparing the different weak integration variants the DC-based variant performs best albeit the PIT-based weak integration

is close behind and potentially slightly simpler to implement.

Overall, the best performance in terms of BSS-Eval SDR, invasive SDR and WER was achieved with the tight integration approach consisting of a DC embedding network, k-means preclustering, and a vMFcACGMM integration model. Oddly, the tight integration variants based on DANs are less effective than their weak integration counterparts. This indicates that either, the DC training is significantly better on the given database (further supported by the 7.06 dB BSS-Eval SDR), or that the DC embeddings are more easily clustered due to the DC loss itself.

As a small remark, it is worth highlighting that we here also report results for which the DAN is evaluated with the corresponding output nonlinearity after k-means clustering of the embeddings and calculating the inner product of each embedding vector with each attractor. In previously reported results, we did not add this additional nonlinearity during inference, although it was part of the training process. In both cases (Row 6 and Row 10) the additional nonlinearity improved results significantly and, therefore, can be seen as a better-justified baseline now.

Table 5.30 lists beamforming results for the same systems as in Table 5.29. Most importantly, for an acoustic model that was not trained in matched conditions, i.e., not trained on the separation results of a given separation system, all WERs are lower with beamforming. In part, this can be attributed to the fact that beamforming averages out artifacts such as hard switches in masks, which otherwise result in musical tones and can confuse acoustic models. This goes so far that negative BSS-Eval SDR values for the DAN are compensated and now provide values up to 6.44 dB.

Again, we realize that the tight integration based on DC and a vMFcACGMM performs best, directly followed by a weak integration based on a PIT-based initialization.

5.9.1 Analysis of splits of the WSJ-BSS database

Figure 5.10a and Figure 5.10b show bar charts of invasive SDR results for different separation models grouped by absolute angular distance between the speakers for masking and beamforming, respectively. Both variants are shown to decouple the effect of an increasing angular distance on the clustering model from the gains due to beamforming. Please note that the examples are split into groups of uneven size.

The important point to note here is that the integration approaches degenerate more gracefully when fewer channels are available than, e.g., the cACGMM alone. Since the cACGMM depends solely on spatial information, its results are worst when the speakers are located very close to each other. This effect is even more pronounced when using beamforming to extract individual speakers. Overall, the best results across all groups are obtained with the tight integration approach, even when speakers are closer than 15° apart from each other. All in all the tight integration approach nicely trades off between spectral and spatial information.

Why the single-channel DC results with masking show slight improvements with an increasing angular distance is not explainable within the scope of this thesis. Although it is possible

Table 5.30: Summary of beamforming results for the WSJ-BSS database. The acoustic model was trained on noisy source images, i.e., it did not see overlap or system artifacts during training. The tuning parameter κ or σ^2 is selected to minimize WER on the development set.

Encoder	Loss	Latent model	Parameter	Output nonlinearity	SDR / dB		WER / %
					BSS-Eval	Invasive	
		cACGMM			11.08	14.66	19.53
DC					10.28	14.65	19.76
DC		cACGMM			11.79	15.62	17.21
DC		vMFcACGMM	$\kappa = 5$		11.59	16.02	16.72
DAN	CE				3.75	6.87	56.01
DAN	CE			softmax	6.44	9.19	47.81
DAN	CE	cACGMM			9.71	13.19	24.60
DAN	CE	GcACGMM	$\sigma^2 = 8$		8.86	12.36	28.67
DAN	MSE				10.29	14.58	20.74
DAN	MSE			sigmoid	10.58	14.64	19.60
DAN	MSE	cACGMM			11.40	15.23	17.85
DAN	MSE	GcACGMM	$\sigma^2 = 8$		11.45	15.49	17.67
PIT	CE				9.95	14.25	20.04
PIT	CE	cACGMM			11.74	15.54	17.41
PIT	NPSMSE				9.28	13.25	24.78
PIT	NPSMSE	cACGMM			11.61	15.43	17.61

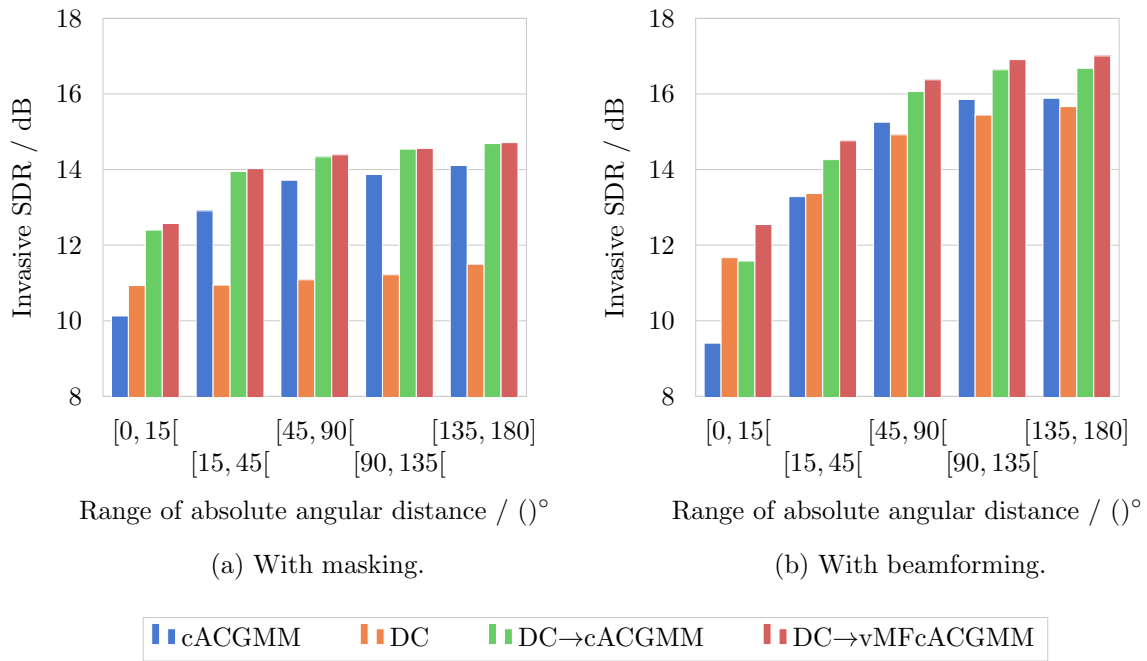


Figure 5.10: Split of separation results based on absolute angular distance. All bars represent slices of the WSJ-BSS test set. Please note that the last group contains about half of the test examples.

that more easily distinguishable RIRs also improve single-channel separability, we will not develop this point further.

Figure 5.11 shows masking results organized in the form of a swarm plot, i.e., a categorical scatter plot with horizontal jitter added in such a way that points do not overlap¹². The WSJ-BSS data is again split into groups of absolute angular distance to get an insight into how system performance degrades when speakers are located close to each other. Although it in principal shows similar results as Figure 5.10a, it puts more emphasis on the outliers and the actual distribution of invasive SDR values. Please note again that the examples are split into groups of uneven size.

We, first of all, emphasize that the highest variability in separation performance occurs with the two baseline systems cACGMM and DC. Both the cACGMM as well as the DC system expose a very high variance with some results even in the negative SDR region. The number of outliers decreases when the speakers are further apart from each other. Most notably, both integration variants do not only show improved mean invasive SDR, they also result in much fewer outliers with a fairly compact distribution for high absolute angular distances. Arguably, although the tight integration approach performs slightly better, the differences to the cascade approach are hardly visible in this visualization.

Figure 5.12a shows average results on the WSJ-BSS database with a split based on the gender composition of a mixture. In accordance with findings on the WSJ0-2mix database analyzed, for example, in [169] the neural network-based approaches degrade quite a bit when separating speakers of the same gender, in particular two female speakers. However, please note that

¹² The idea of a swarm plot originates from the corresponding R package: <http://www.cbs.dtu.dk/~eklund/beeswarm/>

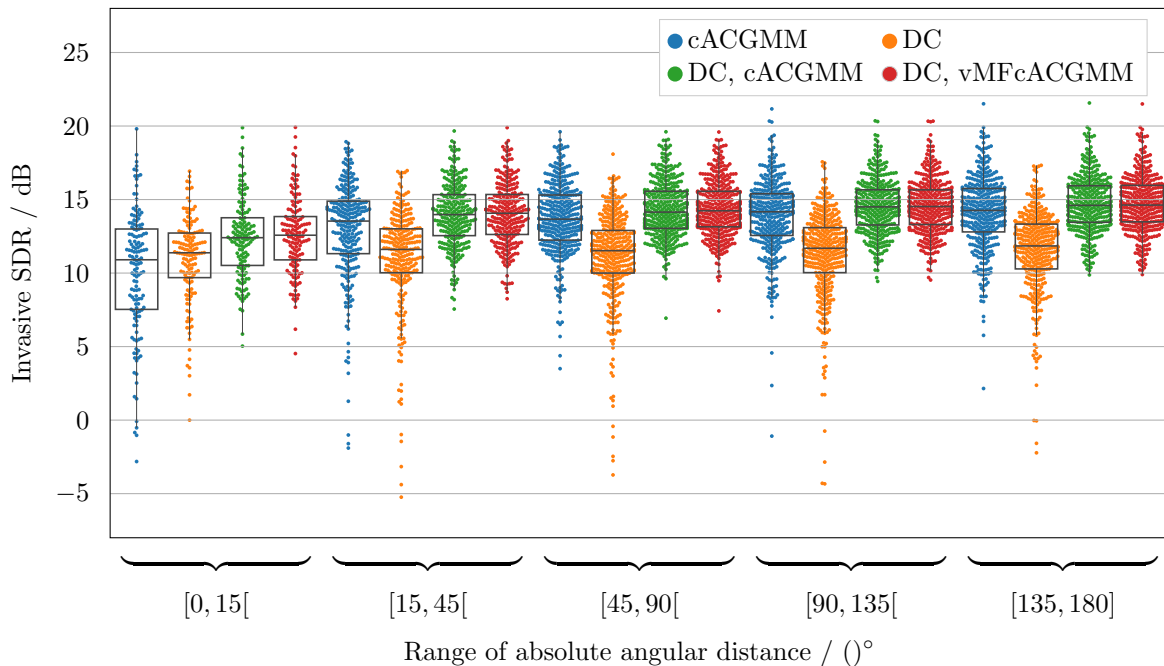


Figure 5.11: Swarm plot of separation results with masking based on absolute angular distance. All groups represent slices of the WSJ-BSS test set. Please note that the last group contains about half of the test examples.

the WSJ-BSS database consists of considerably fewer female speakers. Thus, it is not clear whether female speakers are harder to separate in general or the network simply needs more speech from female speakers for training. Likewise, since the examples are artificially mixed, effects such as the Lombard effect, which occurs in real mixtures, might lead to better separability in a real setup. As expected, the best separation results are obtained on the mixed speaker data, i.e. examples in which a male and a female speaker are mixed. Overall, the tight integration approach performs best in all gender splits.

Figure 5.12b shows a split of the WSJ-BSS test set based on the reverberation time in each example. As expected, the highest invasive SDR values are obtained for low reverberation times. This is reasonable since the W-disjoint orthogonality is highest in an anechoic environment. Further, the temporal smearing effect of long RIRs invalidates the assumption that a RIR fits into a STFT analysis window further. The separation performance of the cACGMM decreases the most with an increasing reverberation time. The integration variants compensate for the effect to some degree due to the additional spectral information. Particularly, the weak integration results highlight that the cACGMM is, in principle, able to separate well if it is initialized close enough to the optimal solution. Even in the highest simulated reverberation conditions, the tight integration approach performs best.

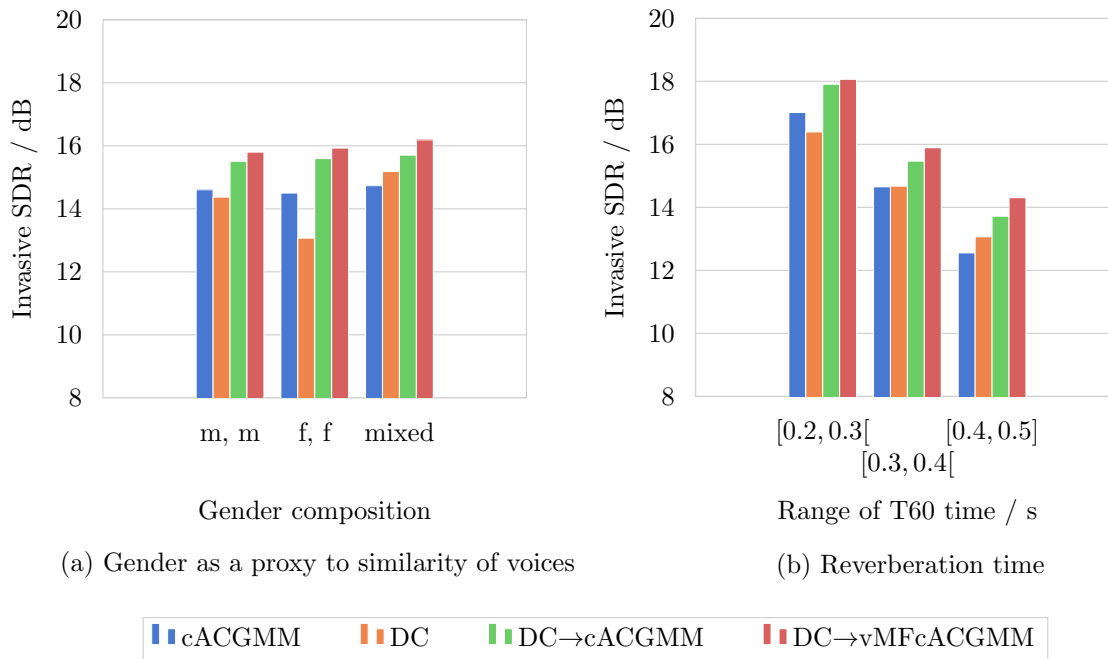


Figure 5.12: Split of separation results based on gender decomposition and sound decay time (T60) range. All bars represent slices of the WSJ-BSS test set.

Findings

- The tight integration approach performs best in all gender splits, reverberation time splits, and absolute angular distance splits.
- The weak integration approach already provides solid results.
- Both variants make use of both data sources.
- The DC system is most susceptible with respect to the gender composition while the cACGMM performance depends most on absolute angular distance and reverberation time.

5.9.2 Analysis with matched training of the acoustic model

This section addresses the often-discussed issue of matched training of the acoustic model. The core idea is that the front-end may produce artifacts or separation results a separately trained AM is not aware of. To understand additional gains due to matched training we report WERs for different acoustic models in this section:

- Image: The acoustic model is trained on noisy images. It has never seen overlap or artifacts produced by a separation module or speech enhancement module before.
- Matched: The acoustic model is trained on the separation results of a given algorithm on the entire train dataset of the WSJ-BSS database. Therefore, it is aware of all

Table 5.31: Comparison of different latent models with a retrained AM on the test set of the WSJ-BSS database. The *Image* column corresponds to an acoustic model that was trained on oracle images with background noise. The *Matched* acoustic model was trained on the separation results of the training dataset. The parameter κ is set to 5 for the vMFcACGMM.

Encoder	Latent model	Extraction method	SDR / dB		WER / %	
			BSS-Eval	Invasive	Image	Matched
DC	cACGMM	GEV→Rank 1→Souden-MVDR	11.06	14.64	19.52	13.68
		GEV→Rank 1→Souden-MVDR	10.28	14.65	19.80	15.58
DC	cACGMM	GEV→Rank 1→Souden-MVDR	11.79	15.62	17.21	13.03
DC	vMFcACGMM	GEV→Rank 1→Souden-MVDR	11.59	16.02	16.72	12.60
Oracle images						10.33

scaling issues, artifacts, and residual interfering speech the corresponding separation model it is later evaluated on is likely to produce.

Performing matched training of the acoustic model is quite costly since it requires the separation of the entire training dataset and an acoustic model training from scratch. Therefore, to experiment with front-end algorithms, all other results outside of this section do not contain matched acoustic model WERs.

Table 5.31 lists separation results for different separation methods, varying latent models, and a fixed extraction method. All results can be compared with the oracle images provided directly to the acoustic model as denoted in the last row. The table allows the following observations: Matched training in general greatly improves performance. That goes so far that the difference between matched and nonmatched (Image) training is larger than the differences between the different separation approaches presented. Further, the biggest WER drop is observed for the cACGMM alone. The WER for a tight integration system with matched training is as low as 12.6 %, which is only 2.27 % points less effective than speech recognition on the oracle speech images.

Table 5.32 showcases matched training results for different source extraction methods. Given that the single-channel DC system was designed with masking in mind, the performance gains using any kind of beamforming is large. A particularly big performance gain, when performing matched training, can be observed when a GEV beamformer is applied to extract the sources. One may conjecture that this is due to the inconsistent scaling for different frequency bins which the acoustic model was not aware of. Using an additional BAN filter greatly improves the acoustic model trained on noisy images while the gain with matched training is much smaller. In accordance with the beamforming evaluation in Section 5.6 the Souden-MVDR with preprocessing of the covariance matrix of the target speaker performs best here.

It is worth pointing out that results with pretraining of the acoustic model on noisy speech images and then just fine-tuning a few epochs on the source separation results are not reported here. Such warm-start strategies are definitely worth investigating as they

Table 5.32: Comparison of different extraction methods with a retrained AM on the test set of the WSJ-BSS database. The *Image* column corresponds to an acoustic model that was trained on oracle images with background noise. The *Matched* acoustic model was trained on the separation results of the training dataset. The parameter κ is set to 5 for the vMFcACGMM.

Encoder	Latent model	Extraction method	SDR / dB		WER / %	
			BSS-Eval	Invasive	Image	Matched
DC	vMFcACGMM	Masking	9.64	14.39	25.97	18.33
DC	vMFcACGMM	GEV	8.04	13.47	30.68	13.83
DC	vMFcACGMM	GEV \rightarrow BAN	10.10	16.20	17.57	13.31
DC	vMFcACGMM	GEV \rightarrow Rank 1 \rightarrow Souden-MVDR	11.59	16.02	16.72	12.60
Oracle images						10.33

promise faster model design iterations as most of the AM training time can most likely be saved.

Findings

The findings in this section can be summarized as follows:

- Although costly, matched training of the acoustic model is crucial for optimal performance to adjust to specifics of a source separation front-end.
- Beamforming is the source extraction method of choice for static geometries. More recent beamforming algorithms developed primarily for speech enhancement apply well to source separation.

5.10 Overview of all methods on WSJ-MC

In contrast to Section 5.9 we here evaluate a selection of separation models on the WSJ-MC database which was briefly introduced in Section 5.2.3. There are two important reasons why we report results on a separate database. First of all, we intend to demonstrate or at least examine the generalizability of the proposed algorithms. Additionally, but potentially even more important, it is necessary to check if trends that appeared on one database still hold on an independent second database. Although all previous model selections and choices were based on the development set, testing again on an independent database renders an analysis more trustworthy.

Most importantly, due to the fact that the database contains realistic recordings of simultaneously spoken utterances, clean speech or speech images are not available. Consequently, the BSS-Eval results are to be read with care, given that they were calculated against a headset microphone as a reference signal. All DNNs in Table 5.33 and Table 5.34 were not retrained. They have been trained once on the WSJ-BSS database and we here purely evaluate, how

Table 5.33: Summary of masking results for the WSJ-MC database. The spectral models as well as the acoustic model are trained on the WSJ-BSS database, i.e., in mismatched conditions. The tuning parameter κ or σ^2 is selected to minimize WER on the development set.

Encoder	Loss	Latent model	Parameter	Output nonlinearity	SDR / dB BSS-Eval	WER
cACGMM					2.96	60.13
DC					1.87	78.64
DC		cACGMM			2.81	64.70
DC		vMFCACGMM	$\kappa = 1$		3.08	63.94
DAN	MSE				2.14	82.42
DAN	MSE			sigmoid	1.91	74.96
DAN	MSE	cACGMM			3.75	57.63
DAN	MSE	GcACGMM	$\sigma^2 = 8$		3.84	55.42
PIT	NPSMSE				2.49	79.24
PIT	NPSMSE	cACGMM			3.87	58.31

they perform in mismatched conditions. The same holds true for the acoustic model, which was trained on the WSJ-BSS speech images with American English whereas the recordings at hand are British English.

First and foremost we observe that the WERs, in general, are much higher than on WSJ-BSS. This is particularly the case for the single-channel systems purely based on DNNs with WERs up to 82.42 % for masking. This increase is also visible in the beamforming results in Table 5.34 indicating that the mismatch did not just cause minor artifacts which could have been smoothed out by the beamforming operation.

However, the purely unsupervised and model-based cACGMM performs the best of all nonintegration variants. Therefore, we may deduce that the main gains in integration systems are to be expected from spatial observations.

The overall best WER is obtained with a tight integration approach consisting of a DAN, a k-means preclustering and a GcACGMM integration model with a subsequent beamforming step. This system improved upon the cACGMM by 4.11 % points WER, but – given the high baseline WER – this is not an astonishing step forward. This differs from the results on the WSJ-BSS database in Section 5.9 in the sense that there the best performing system was a tight integration approach with a DC embedding network.

Overall, we may conclude that the integration methods not just allow for a combination of spatial and spectral cues but, to some degree, also help generalize better to unseen recordings. This finding is consistent with the hypothesis of better generalizability mentioned in the introduction of [165] on integration for speech enhancement.

Table 5.34: Summary of beamforming results for the WSJ-MC database. The spectral models as well as the acoustic model are trained on the WSJ-BSS database, i.e., in mismatched conditions. The tuning parameter κ or σ^2 is selected to minimize WER on the development set.

Encoder	Loss	Latent model	Parameter	Output nonlinearity	SDR / dB BSS-Eval	WER
		cACGMM			5.09	46.06
DC					3.72	61.99
DC		cACGMM			4.63	44.96
DC		vMFCACGMM	$\kappa = 5/2$		5.12	42.71
DAN	MSE				4.77	56.74
DAN	MSE			sigmoid	4.58	55.25
DAN	MSE	cACGMM			5.64	42.84
DAN	MSE	GcACGMM	$\sigma^2 = 8$		5.70	41.95
PIT	NPSMSE				3.85	67.58
PIT	NPSMSE	cACGMM			5.64	43.86

5.11 Reproducibility and statistical significance

This section lists some remarks on reproducibility and statistical significance. The main goal is to emphasize sources of variability when experimenting with neural and probabilistic system components. The applicability of statistical significance is briefly discussed and a limited number of experiments are performed to showcase variability in different stages of the experiment.

First of all, it is worth to identify the main important sources of variability. Table 5.35 shows the simplified processing flow for the multi-channel separation and recognition systems described in this work. Below, it provides a high-level overview of the sources of variability, each of which we are going to discuss in the following briefly.

In each part of the processing pipeline implementation details, implicit assumptions, and hyperparameters that may not have been communicated are the cause of a great deal of variability when applying the same method to the same data. Although researchers thrive for reproducible results in the sense that, by reading a paper one is able to reproduce results up to equal performance metrics it is often practical to resort to reproducibility in the sense of experiments that lead to similar conclusions [202, Section 2]. To facilitate reproducibility publishers start to ask for executable code alongside the main publication [203].

While the variability due to implementation details is hard to quantify, another important aspect is the training dataset. In principle, the unspoken assumption is that the training set is sufficiently large to resemble the totality of situations on which the system is to be applied¹³. This is of course in almost all cases impossible and the training set has to be seen

¹³ One may thrive for an infinite training set but even that is not sufficient when the training examples are statistically dependent and do not represent the totality of possible test mixtures sufficiently well.

Table 5.35: Main sources of randomness in evaluating integration approaches to blind source separation. Tick symbols (✓) indicate where a given type of randomness applies. Items in parenthesis can be a source of variability but are not employed here.

	→	Embedding network	→	Mixture model	→	Beamforming/ masking	→	ASR	→
Implementation/ hyperparameters		✓		✓		✓		✓	
Selection of training data		✓		✗		✗		✓	
Augmentation and dropout		(✓)		✗		✗		(✓)	
Weight initialization		✓		✗		✗		✓	
Train data order		✓		✗		✗		✓	
Selection of dev data		✓		✓		✓		✓	
Initialization at test time		✗		✓		✗		✗	

as one sample from the distribution of the totality of examples. Consequently, researchers often agree on a fixed database consisting of a fixed training, development, and test split well knowing that results may or may not hold on a different database. In the scope of this thesis the WSJ0-2mix database can at this point in time be seen as such a community standard albeit its shortcomings discussed in Section 5.2.1. Besides containing simulated multi-channel data the presented WSJ-BSS database addresses these shortcomings with a larger training set size and substantially more unique speakers in the training set (see Section 5.2.2 for details). In the following, we treat the training set as given. An analysis of the transferability of the trained systems to mismatched real recordings can be found in Section 5.10.

During the training of, e.g., a neural network two important sources of randomness are (1) the random weight initialization and (2) the random order in which examples are presented to the system. Dodge et al. explicitly differentiate between these two factors of variability and report high variability of the test results while the variability due to the random seed for weight initialization and the random seed for the data order is similar [204]. While Dodge et al. are able to draw these conclusions by repeating the same experiment more than a thousand times, we here have to constrain ourselves to a limited number of repetitions and do not differentiate between randomness in weight initialization and data order. The approach to demonstrate the influence of a random seed for weight initialization and data order is here demonstrated by repeating the training (pretraining and fine-tuning) 10 times. Figure 5.13 visualizes different training results as a parallel coordinates plot when repeating the training of the DC system on the WSJ-BSS database as used in the previous parts of the evaluation chapter. First of all, the figure allows us to draw the conclusion that the mean loss values and mean MIR-Eval SDR values indeed differ quite a bit between repeated runs. However,

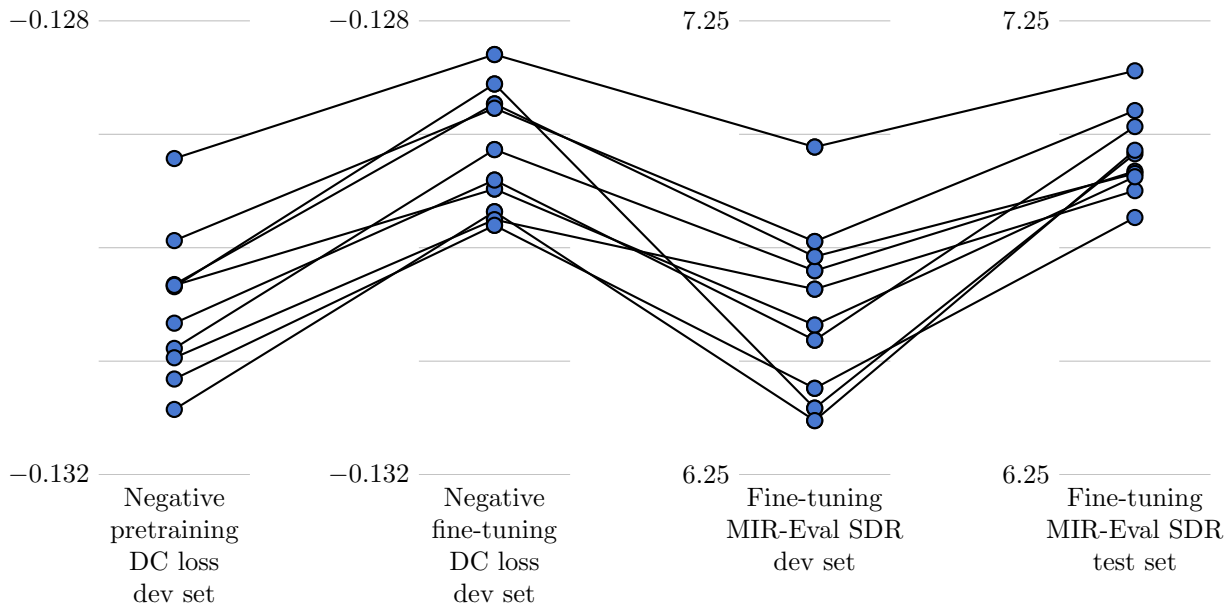


Figure 5.13: Parallel coordinates plot of multiple metrics of pretraining and fine-tuning a DC embedding network. Each column represents a different metric or training stage (with possibly different scaling). The connecting lines help understand how consistent the rank (in the sense of ordering) is over different metrics or training stages. The MIR-Eval SDR values are obtained using mask multiplication instead of beamforming.

we also realize that the variability is much smaller on the test set (1500 examples) than on the development set (500 examples). Further, we may deduce that albeit all variability fine-tuning of the DC system by training on entire utterances is helpful. At first sight, we also realize that models that were better after pretraining tend to be better after fine-tuning. However, this observation will be put into perspective when discussing the corresponding table with rank correlations in the following.

It is worth pointing out that a sophisticated learning rate scheduling combined with patience-based early stopping can increase the variability further. Analyzing the number of training steps (one step corresponds to presenting one mini-batch to the network) reveals that the total number of steps in pretraining as well as in fine-tuning can differ by more than a factor of 2 as indicated in Table 5.36.

To illustrate the variability of the test results for different methods, we here rely on confidence intervals of mean values to characterize the quality of the mean estimate. The 95 % confidence intervals here are based on bootstrapping with 10000 bootstraps for each estimate. In [205, Section 9.1] Chernick argues that rather small sample sizes even below 10 samples (here 10 independently trained DC systems) can be used to calculate bootstrap estimates. However, as a rule of thumb, he recommends that confidence intervals should be trusted when the sample size is at least 30. Since the repetition of neural network training is prohibitively expensive in this context, we need to make the best of our 10 samples and emphasize that the confidence intervals themselves may be unreliable.

Figure 5.14 shows aggregated invasive SDR values for 10 trained DC models. The bars represent mean values over the 10 runs while the black lines at the tip of each bar represent

Table 5.36: Comparison of training duration in steps for repeated training of the same DC system for pretraining and fine-tuning.

Pretraining	Fine-tuning
980 001	135 001
680 000	277 501
780 000	198 750
790 000	180 000
610 000	168 801
780 001	146 301
1 380 001	153 750
760 000	172 501
540 000	150 001
1 150 001	135 001

the confidence intervals obtained with bootstrapping¹⁴. First of all, one can observe that all confidence intervals are quite small. However, given that only 10 repeated trainings were performed the confidence interval obtained through bootstrapping tends to underestimate the uncertainty when compared to infinitely many repeated experiments [207, Page 25]. Please note that the horizontal bar is scaled such that the confidence intervals are best visible. Consequently, the disagreement between models trained with a different seed and data order is almost negligible. The confidence intervals become smaller when using an integration model. This can be explained by the fact that outliers within one run are already less likely because one model component can compensate failures of the other to some degree. This hypothesis is further supported when inspecting, e.g., Figure 5.11. Interestingly, the cACGMM (first row) shows some variability between runs, too. This cannot be explained with a varying model initialization seed or a varying training data order since this experiment does not contain any trained system component. This variability only stems from the random seed when initializing the mixture model before the EM algorithm starts.

Figure 5.15 shows the dependency of a strong integration model on the tuning parameter κ similar to Figure 5.9. Here, Figure 5.15 shows the confidence intervals obtained with 10 trained DC models as shaded areas. It becomes apparent that the gain from the strong integration approach which uses the vMFCACGMM is much larger than the width of each of the confidence intervals. We can, therefore, conclude that it is highly unlikely, that the gains are only caused by randomness.

To further quantify the reliability of the evaluation results the following paragraphs make use of statistical hypothesis tests. We here limit the number of tests to three and highlight that only the three confidence tests were performed to avoid cherry-picking p -values.

¹⁴ We here rely on the Seaborn implementation [206] of the percentile bootstrap approach by using `seaborn.algorithms.bootstrap()` and `seaborn.utils.ci()`.

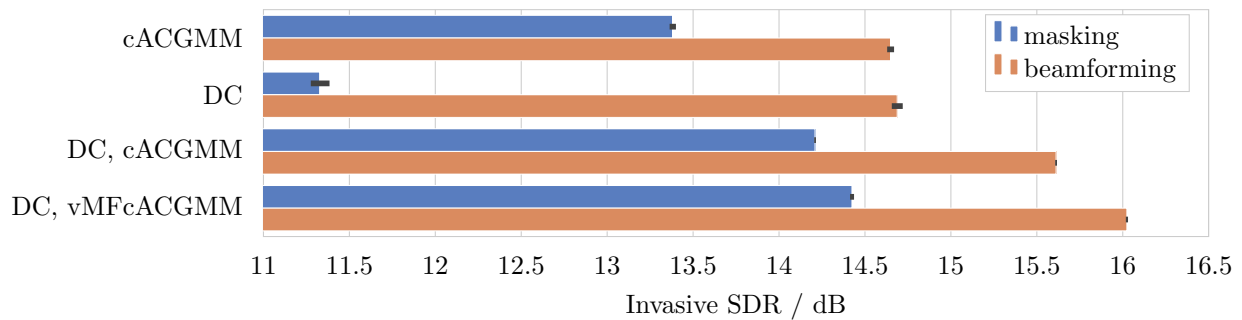


Figure 5.14: Confidence intervals of mean invasive SDR values as black lines for 10 trained DC models. All results are obtained on the WSJ-BSS test set. The horizontal axis is stretched as much as possible to improve the visibility of confidence intervals.

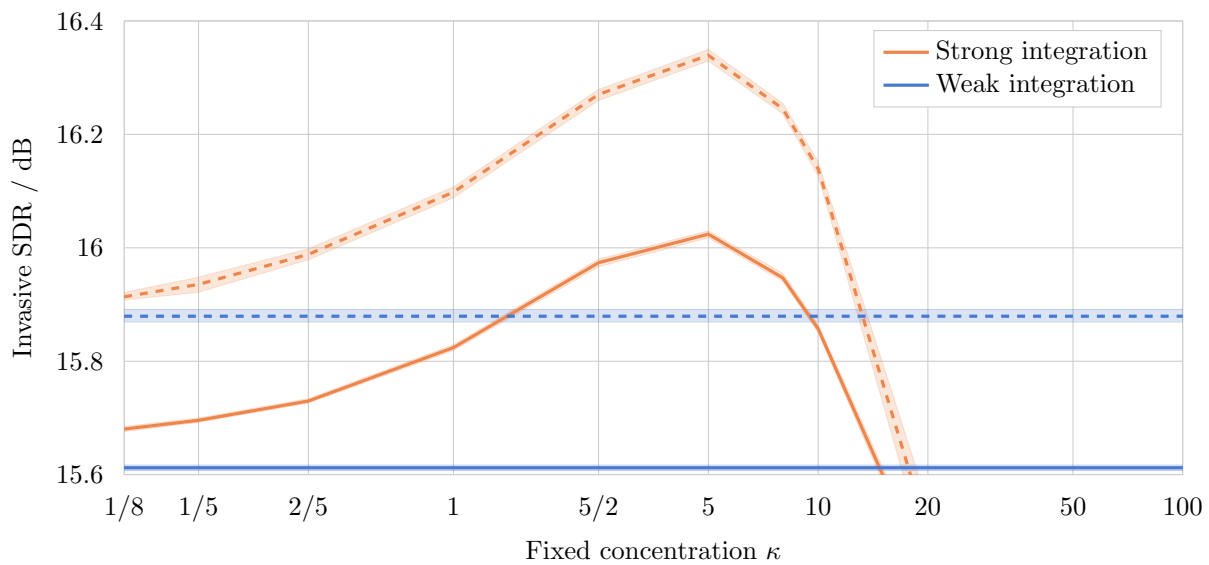


Figure 5.15: Confidence intervals of mean invasive SDR values as shaded areas for 10 trained DC models. Dashed lines indicate development set results. All results are obtained on the WSJ-BSS database and use beamforming. The confidence intervals for the test set are only slightly wider than the line width and therefore barely visible.

Hypothesis: weak integration better than the DC approach For hypothesis testing, we follow the significance testing (p -value approach) as outlined in [208, Page 344] because the p -value approach avoids setting a fixed significance level α in advance. In this approach the p -value in a sense measures the credibility of H_0 : It is the probability of the measurements being at least as extreme as observed given that H_0 is true [209]. In other words, a low p -value provides evidence against H_0 [209]. However, one should be well aware of the possible ways to misinterpret p -values as extensively discussed in [209].

To conduct a hypothesis test we first define the null hypothesis H_0 and the alternative hypothesis H_1 :

- H_0 : There is no difference between both approaches or the weak integration is worse.
- H_1 : Weak integration performs better than the DC approach.

The possible outcomes of this test procedure are:

- Reject H_0 .
- Fail to reject H_0 .

However, it is worth noting that the test is posed such that a negative test outcome does not necessarily imply that H_0 is true.

Next, we define a valid test statistic: Since the variance of the invasive SDR values (actually mean values for one run with one model over one dataset) is unknown, we need to reside to tests in which the variance is estimated. With an estimated variance, the test statistic is Student- t -distributed and given as follows [208, Page 340]:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{I}}, \quad (5.2)$$

where \bar{x} is the mean estimate of test outcomes x_i (x_i being one mean SDR value on the entire dataset under consideration), μ_0 is the assumed mean under the H_0 hypothesis and s is the estimate of the standard deviation of test outcomes x_i . Since each group of samples is derived from the same set of $I = 10$ trained DC models the assumption that both groups are independent does not hold. Therefore, we need to perform a paired test such that here x_i is a difference of a weak integration result and a DC result.

Next, we compute the test statistic t and the corresponding p -value:

$$t = 60.08, \quad p = 2.47\text{e-}13. \quad (5.3)$$

Finally, we can interpret the result. The p -value is extremely small, which means that the probability that a value of $t \geq 60.08$ appeared under the H_0 hypothesis is extremely low. Therefore, we can treat this as evidence to reject H_0 and conclude (since it is the only alternative in a one-sided test) that the weak integration approach is better than the DC approach alone with a very high significance level.

Hypothesis: weak integration better than cACGMM In a similar vein, we can now compare the weak integration approach with the unsupervised cACGMM baseline. The two hypotheses are formulated as follows:

- H_0 : There is no difference between both approaches or the weak integration is worse.
- H_1 : Weak integration performs better than the cACGMM.

In contrast to the last hypothesis test, the 10 repetitions of the weak integration approach (with 10 independently trained DC models) can be considered independent from the 10 repetitions of the cACGMM and therefore a paired test is not necessary. However, just as before, the conducted Student- t test is one-sided. The resulting test statistic t and the corresponding p -value are given as follows:

$$t = 86.72, \quad p = 2.34e-25. \quad (5.4)$$

We observe an extremely low p -value indicating that the observed data is almost irreconcilable with H_0 . This is again good evidence against H_0 . Since H_1 is complementary to H_0 in this case, one may conclude that the weak integration performs better with a high significance level.

Hypothesis: strong integration better than the weak integration Finally, we compare the strong integration approach with the weak integration approach in a paired one-sided Student- t test with the following hypotheses:

- H_0 : There is no difference between both approaches or the strong integration is worse.
- H_1 : Strong integration performs better than weak integration.

The test statistic t and the p -value compute to:

$$t = 267.9, \quad p = 3.57e-19. \quad (5.5)$$

We again observe a very low p -value which indicates that the observed invasive SDR values are very unlikely produced under the null hypothesis. Since H_1 is again chosen to be complementary, we can use it as evidence that H_1 is the better hypothesis.

Findings

The findings in this section can be summarized as follows:

- Given the ten repeated training results the confidence intervals for mean estimates are small in comparison to the differences between different methods.
- Complementarily, using statistical hypothesis tests, the differences between mean estimates of different experiments have been proven to be statistically significant.

6 Conclusion

In conclusion of this thesis the three main contributions *cascade integration*, *unsupervised training*, and *tight integration* are highlighted in the following. Subsequently, remaining challenges and possible indications for future research decisions are discussed quite briefly to round up this contribution.

Main contributions

Cascade integration: The first main contribution is clearly the cascade integration approach, which has been introduced and theoretically justified in Section 4.2, carefully evaluated in Section 5.7.1, and put into perspective in Section 5.9 and Section 5.10. Identifying and acknowledging that the main limitation of probabilistic spatial clustering approaches stems from their initialization and not necessarily their modeling capacity led to the conclusion that neural network-based initialization can yield a significant performance boost. The self-dependent nature of the parameter estimation procedure of spatial clustering naturally led to optimization ending in local optima. A neural network, even when trained on data mediocrely fitting to the test conditions, can provide the necessary hint tipping the estimation process towards an optimum closer to the global one. This is particularly in terms of better initialization per frequency bin but also due to the resulting improved consistency across frequency bins alleviating the permutation problem to a certain degree.

Unsupervised training: While source separation neural networks almost always require artificially generated mixtures for training, we here demonstrated a principled way for unsupervised training, i.e., to train a source separation model when only multi-channel mixtures are observable. The key concept introduced in Section 4.4 and evaluated in Section 5.8 is to apply a spatial clustering model first, which serves as an unsupervised teacher and is, on average, inclined to provide a good separation result. It turned out that this teacher-student training scheme can lead to situations, in which the student in the aforementioned cascade integration is able to outperform the teacher both in terms of objective separation performance metrics and speech recognition metrics.

Tight integration: The third contribution, arguably the core contribution of this thesis, is the tight integration approach, in which a neural network extracts embedding vectors from an observed mixture which then gets jointly modeled with spatial features in a single probabilistic graphical model. The concept is detailed in Section 4.3, evaluated in Section 5.7.2 and compared with other approaches on two different databases in Section 5.9 and Section 5.10. By estimating all spatial and spectral model parameters jointly at test time, a mismatch between both cues is minimized, a possible permutation problem confusing speakers across

frequency bins is alleviated, and overall better generalizability across database boundaries is achieved.

Future work

It is worth pointing out that the current development of speech separation neural networks points towards time-domain separation approaches. While this might sound like separation takes place in time-domain it much rather encapsulates the concept that the neural network extracts high-level features itself and then performs separation in a more abstract but potentially more suitable domain. A task that remains is to better understand how these systems scale to more adverse conditions such as background noise and reverberation and, more in the spirit of this thesis, how spatial features can effectively be used to boost separation performance further.

Continuous meeting recognition implies the idea of dropping many artificial assumptions made in meeting recognition challenges and literature such as externally provided time-annotations, speaker identities, or diarization results. Although the general idea to transcribe meetings automatically has been prevalent for quite some time, just recently researchers are beginning to focus more on actually processing a continuous meeting including speaker counting, tracing, separation and recognition in all its facets.

As a final, more general remark: Within this work DNNs have been used to perform a subordinate task, such as extracting embedding vectors, which are eventually clustered with a statistic model. Just in the same spirit, it is important to point out that neural networks particularly excel, when hand-crafted rules hardly capture the whole picture but, when required, probabilistic graphical models allow to encode human knowledge or even human-defined policy. In times in which one asks for interpretable artificial intelligence, probabilistic models making business decisions on top of neural network-based subordinate processes, such as computer vision or speech recognition can be an answer.

A Appendix

A.1 Properties of the complex Bingham distribution

A.1.1 Eigenvalue shift in the normalization term

Using the analytic expression of the complex Bingham normalization term (Equation 3.18), it can be shown that an eigenvalue shift can be factored out of the expression:

$$\begin{aligned}
c_B(\mathbf{B} + \delta \mathbf{I}_D) &= 2\pi^D \sum_d e^{\lambda_d + \delta} \prod_{d' \neq d} \left((\lambda_d + \delta) - (\lambda_{d'} + \delta) \right)^{-1} \\
&= e^\delta \cdot 2\pi^D \sum_d e^{\lambda_d} \prod_{d' \neq d} (\lambda_d - \lambda_{d'})^{-1} \\
&= e^\delta c_B(\mathbf{B}).
\end{aligned} \tag{A.1}$$

A.1.2 Eigenvalue shift in the distribution

The complex Bingham distribution is invariant to an eigenvalue shift. To be able to reference this invariance, it is demonstrated in the following:¹

$$\begin{aligned}
\mathcal{CB}(\tilde{\mathbf{y}}; \mathbf{B} + \delta \mathbf{I}_D) &= c_B^{-1}(\mathbf{B} + \delta \mathbf{I}_D) \cdot e^{\tilde{\mathbf{y}}^H (\mathbf{B} + \delta \mathbf{I}_D) \tilde{\mathbf{y}}} \\
&\stackrel{\text{Appendix A.1.1}}{=} e^{-\delta} c_B^{-1}(\mathbf{B}) \cdot e^{\tilde{\mathbf{y}}^H \mathbf{B} \tilde{\mathbf{y}}} \cdot e^{\delta \tilde{\mathbf{y}}^H \mathbf{I}_D \tilde{\mathbf{y}}} \\
&\stackrel{\tilde{\mathbf{y}}^H \tilde{\mathbf{y}} = 1}{=} c_B^{-1}(\mathbf{B}) \cdot e^{\tilde{\mathbf{y}}^H \mathbf{B} \tilde{\mathbf{y}}} \\
&= \mathcal{CB}(\tilde{\mathbf{y}}; \mathbf{B}).
\end{aligned} \tag{A.2}$$

A.2 Non-negativity of the Kullback-Leibler divergence

The Kullback-Leibler divergence is always larger or equal to zero. This is a direct consequence of the Jensen's inequality [210, Equation 5 or Equation 5']:

$$\mathbb{E} \{ \varphi(\tilde{x}) \} \geq \varphi(\mathbb{E} \{ \tilde{x} \}) \quad \text{for a convex function } \varphi. \tag{A.3}$$

¹ According to [97] this can be *easily checked*; characteristically this expression appears six times in [97].

When we apply this in the following for the PDFs $p = p(\mathcal{Z})$ and $q = q(\mathcal{Z})$, we apply it to the convex function $\varphi(x) = -\ln x$ since $\ln x$ is concave:

$$\begin{aligned} \text{KL}(q\|p) &= - \int_{\mathcal{Z}} q \ln \frac{p}{q} d\mathcal{Z} = - \mathbb{E} \left\{ \ln \frac{p}{q} \right\} = \mathbb{E} \left\{ - \ln \frac{p}{q} \right\} \\ &\stackrel{\text{Jensen's}}{\geq} - \ln \mathbb{E} \left\{ \frac{p}{q} \right\} = - \ln \int_{\mathcal{Z}} q \cdot \frac{p}{q} d\mathcal{Z} = - \ln \int_{\mathcal{Z}} p d\mathcal{Z} = 0. \end{aligned} \quad (\text{A.4})$$

Alternatively, we can make use of $\ln x < x - 1$ and obtain the result for the PDFs $p = p(\mathcal{Z})$ and $q = q(\mathcal{Z})$ without resorting to Jensen's inequality [17, Equation 3.4]:

$$\text{KL}(q\|p) = - \int_{\mathcal{Z}} q \ln \frac{p}{q} d\mathcal{Z} \geq - \int_{\mathcal{Z}} q \left(\frac{p}{q} - 1 \right) d\mathcal{Z} = - \int_{\mathcal{Z}} p d\mathcal{Z} + \int_{\mathcal{Z}} q d\mathcal{Z} = 0. \quad (\text{A.5})$$

A.3 Mixture weights without Lagrange's method

When maximizing the log-likelihood of a mixture model with respect to the mixture weights, one usually introduces the sum-1 constraint into the optimization function with a Lagrange multiplier. Alternatively, this can be done by replacing π_k with $\alpha_k / \sum_{k'} \alpha_{k'}$ and performing an unconstrained optimization:²

$$\begin{aligned} J &= \sum_n \ln \left(\sum_k \frac{\alpha_k}{\sum_{k'} \alpha_{k'}} p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \\ &= \sum_n \ln \left(\frac{1}{\sum_{k'} \alpha_{k'}} \sum_k \alpha_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \\ &= \sum_n \ln \sum_k \alpha_k p(\mathbf{y}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) - N \ln \sum_{k'} \alpha_{k'}. \end{aligned} \quad (\text{A.6})$$

We can now differentiate with respect to α_k :

$$\begin{aligned} \frac{\partial J}{\partial \alpha_k} &= \sum_n \frac{p(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \alpha_{k'} p(\mathbf{y}; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} - N \frac{1}{\sum_{k'} \alpha_{k'}} \stackrel{!}{=} 0 \quad \Big| \cdot \alpha_k \\ \Leftrightarrow \sum_n \frac{\alpha_k p(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \alpha_{k'} p(\mathbf{y}; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} - N \frac{\alpha_k}{\sum_{k'} \alpha_{k'}} &= 0 \quad \Big| \begin{array}{l} \text{expand first fraction} \\ \text{with } 1 / \sum_{k'} \alpha_{k'} \end{array} \\ \Leftrightarrow \sum_n \frac{\pi_k p(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\underbrace{\sum_{k'} \pi_{k'} p(\mathbf{y}; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}_{\gamma_{k,n}}} - N \frac{\alpha_k}{\underbrace{\sum_{k'} \alpha_{k'}}_{\pi_k}} &= 0 \\ \Leftrightarrow \pi_k = \frac{1}{N} \sum_n \gamma_{k,n}. \end{aligned} \quad (\text{A.7})$$

² This idea originates from a discussion with Christoph Boeddeker during classes.

A.4 Remarks on complex derivatives

Complex-valued functions $g : \mathbb{C} \mapsto \mathbb{C}$ are *complex differentiable* at a given point if and only if the limit of the difference quotient exists, i.e., converges to a single value independent of the path of h [211, Section 1.2.2]:

$$\frac{dg}{dz} = \lim_{h \rightarrow 0} \frac{g(z+h) - g(z)}{h}. \quad (\text{A.8})$$

It can be shown that the limit exists, when the Cauchy-Riemann differential equations hold [211, Equation 1.3], where g_r and g_i are the real and imaginary part of g and z_r and z_i are the real and imaginary part of z :

$$\frac{\partial g_r}{\partial z_r} = \frac{\partial g_i}{\partial z_i}, \quad \frac{\partial g_i}{\partial z_r} = -\frac{\partial g_r}{\partial z_i}. \quad (\text{A.9})$$

If a function is complex differentiable everywhere, such a function is called analytic or holomorph. However, many functions do not fulfill these properties, e.g., $g(z) = \text{Re}\{z\}$, $g(z) = \text{Im}\{z\}$, or $g(z) = z^*$. Most relevant for this work, when the goal is to optimize parameters of a system involving complex numbers, the following problem occurs: The cost function is real-valued, after all it has to be a single value which we optimize for, while one or more intermediate variables or parameters are complex-valued. But, any real-valued function $f : \mathbb{C} \mapsto \mathbb{R}$ only fulfills the Cauchy-Riemann differential equations in Equation A.9 when it is a constant, i.e. $f(z) \equiv \text{const.}$

Consequently, we need to rely on an alternative definition of differentiability. One natural definition is to only require the differentiability of the real and imaginary part of the complex-valued function with respect to the real and imaginary part of the input independently. This property is called *real differentiability* and holds for the aforementioned examples [211, Definition 2].

To motivate partial derivatives for real differentiable functions we may start with relating the real and imaginary part of the input variable z with z and z^* [212, Page 65]:

$$z_r = \frac{z + z^*}{2}, \quad z_i = \frac{z - z^*}{2j}. \quad (\text{A.10})$$

Based on this, we can state the corresponding partial derivatives:

$$\frac{\partial z_r}{\partial z} = \frac{1}{2}, \quad \frac{\partial z_r}{\partial z^*} = \frac{1}{2}, \quad \frac{\partial z_i}{\partial z} = \frac{1}{2j}, \quad \frac{\partial z_i}{\partial z^*} = -\frac{1}{2j}. \quad (\text{A.11})$$

From the chain rule for real-valued intermediate variables, the two partial derivatives with respect to z and z^* directly follow [212, Page 65]:

$$\begin{aligned} \frac{\partial g}{\partial z} &= \frac{\partial g}{\partial z_r} \frac{\partial z_r}{\partial z} + \frac{\partial g}{\partial z_i} \frac{\partial z_i}{\partial z} = \frac{1}{2} \left(\frac{\partial g}{\partial z_r} - j \frac{\partial g}{\partial z_i} \right), \\ \frac{\partial g}{\partial z^*} &= \frac{\partial g}{\partial z_r} \frac{\partial z_r}{\partial z^*} + \frac{\partial g}{\partial z_i} \frac{\partial z_i}{\partial z^*} = \frac{1}{2} \left(\frac{\partial g}{\partial z_r} + j \frac{\partial g}{\partial z_i} \right). \end{aligned} \quad (\text{A.12})$$

This set of rules and the notion that we can calculate partial derivatives of functions with respect to z and z^* independently is called *Wirtinger calculus* [213], [214].³

Alternatively, we may motivate complex derivatives by replacing the complex numbers with their vector representation $z = (z_r \ z_i)^T$. Given that in this sense $f : \mathbb{C} \mapsto \mathbb{R}$ is treated as a function which maps from \mathbb{R}^2 to \mathbb{R} , we can decompose the derivative:

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial (z_r \ z_i)} = \begin{pmatrix} \frac{\partial f}{\partial z_r} \\ \frac{\partial f}{\partial z_i} \end{pmatrix} = \frac{\partial f}{\partial z_r} + j \frac{\partial f}{\partial z_i}. \quad (\text{A.13})$$

The last step makes use of the composition of complex numbers again. Apparently, this definition is in contrast to Equation A.12. Further, it has the charming effect that the derivative using this definition for real-valued functions with complex argument coincides with the derivative of the same function with real-valued argument:

$$\frac{\partial |z|^2}{\partial z} \stackrel{\text{Equation A.13}}{=} 2z \quad \text{coincides with} \quad \frac{\partial |x|^2}{\partial x} = 2x. \quad (\text{A.14})$$

Further, most of the matrix derivatives derived for real-valued matrices, e.g., as listed in the Matrix Cookbook [217] can be used right away by replacing transpose symbols with conjugate transpose symbols.

We here arbitrarily choose the definition as popularized in engineering by [214] and do not thrive to resolve apparent conflicts between Equation A.12 and Equation A.13. It is worth pointing out that in the context of this thesis the additional factor 1/2 does not change the optimization results, i.e., it cancels out when setting derivatives to zero. However, when performing complex-valued backpropagation as, e.g., in our own prior work [136], [177], [218], [219], the additional factor 1/2 indeed impacts the weight updates and may lead to different results.

A.5 GEV/MaxSNR beamformer

The GEV/MaxSNR beamformer maximizes the generalized Rayleigh coefficient given the speech image covariance matrix and noise covariance matrix:

$$J = \frac{\mathbf{w}^H \Phi_{\mathbf{x}\mathbf{x}} \mathbf{w}}{\mathbf{w}^H \Phi_{\mathbf{nn}} \mathbf{w}}. \quad (\text{A.15})$$

Maximizing the generalized Rayleigh coefficient can either be formulated as a constrained optimization problem or a regular optimization problem. While the former is a bit shorter, the latter has the advantage that we do not have to introduce a seemingly arbitrary constraint.

³ The same definition, i.e., including $\frac{1}{2}$ is also used by, besides others, Adali, Haykin, and Schreier [211], [215], [216].

A.5.1 Solution with constraint optimization

One way is to constrain the denominator to a fixed scalar, e.g., $\mathbf{w}^H \Phi_{nn} \mathbf{w} = 1$. Any other choice, such as $\mathbf{w}^H \mathbf{w} = 1$, is just as valid. It will lead to the same maximum for J , but the actual filter coefficients are (in almost all cases) different. Adding this additional constraint with a Lagrange multiplier results in the Lagrangian function:

$$J' = \mathbf{w}^H \Phi_{xx} \mathbf{w} + \lambda(1 - \mathbf{w}^H \Phi_{nn} \mathbf{w}). \quad (\text{A.16})$$

Differentiating this with respect to \mathbf{w} in the sense of Section A.4 and setting the result to zero leads to the following generalized eigenvalue problem:

$$\frac{\partial J'}{\partial \mathbf{w}} = \Phi_{xx} \mathbf{w} - \Phi_{nn} \mathbf{w} \stackrel{!}{=} \mathbf{0} \quad \Leftrightarrow \quad \Phi_{xx} \mathbf{w} = \lambda \Phi_{nn} \mathbf{w}. \quad (\text{A.17})$$

According to this result, any eigenvalue leads to an extremum. However, to argue why the maximum eigenvalue leads to the maximum Rayleigh coefficient, we can multiply the previous statement with \mathbf{w}^H from the left:

$$\mathbf{w}^H \Phi_{xx} \mathbf{w} = \lambda \mathbf{w}^H \Phi_{nn} \mathbf{w}. \quad (\text{A.18})$$

We realize that the left hand side term coincides with J . Given that the quadratic form on the right hand side is fixed due to our constraint, J is maximized by choosing the maximum value for λ , i.e., the maximum eigenvalue.

A.5.2 Solution without constraint optimization

Alternatively, we calculate the derivative with respect to \mathbf{w} in the sense of Section A.4 without any constraint and set it equal to zero. This can be done by applying the quotient rule for real-valued scalars:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}} &= \frac{\Phi_{xx} \mathbf{w} \cdot \mathbf{w}^H \Phi_{nn} \mathbf{w} - \mathbf{w}^H \Phi_{xx} \mathbf{w} \cdot \Phi_{nn} \mathbf{w}}{(\mathbf{w}^H \Phi_{nn} \mathbf{w})^2} \stackrel{!}{=} \mathbf{0} \\ \Leftrightarrow \quad \frac{\Phi_{xx} \mathbf{w}}{\mathbf{w}^H \Phi_{nn} \mathbf{w}} &= \underbrace{\frac{\mathbf{w}^H \Phi_{xx} \mathbf{w}}{\mathbf{w}^H \Phi_{nn} \mathbf{w}}}_{\lambda} \cdot \frac{\Phi_{nn} \mathbf{w}}{\mathbf{w}^H \Phi_{nn} \mathbf{w}} \quad \Bigg| \cdot (\mathbf{w}^H \Phi_{nn} \mathbf{w}) \\ \stackrel{\mathbf{w}^H \Phi_{nn} \mathbf{w} > 0}{\Leftrightarrow} \quad \Phi_{xx} \mathbf{w} &= \lambda \Phi_{nn} \mathbf{w}. \end{aligned} \quad (\text{A.19})$$

Finally, we identify that λ is again exactly the ratio we intended to maximize. The solution to Equation A.19 which maximizes the original cost function is then the eigenvector corresponding to the biggest eigenvalue.

A.6 MVDR beamformer

The MVDR formalism asks for a beamforming vector which minimizes the expected output variance while respecting a distortionless constraint as in

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_t |\mathbf{w}^H \mathbf{x}_t|^2 \quad \text{s.t.} \quad \mathbf{w}^H \mathbf{d} = 1. \quad (\text{A.20})$$

The linear constraint can be incorporated in the cost function by using a Lagrange multiplier resulting in a modified optimization criterion given by

$$J' = \sum_t |\mathbf{w}^H \mathbf{x}_t|^2 + \lambda(1 - \mathbf{w}^H \mathbf{d}) \quad (\text{A.21})$$

$$= \mathbf{w}^H \sum_t \mathbf{x}_t \mathbf{x}_t^H \mathbf{w} + \lambda(1 - \mathbf{w}^H \mathbf{d}) \quad (\text{A.22})$$

$$= \mathbf{w}^H \Phi_{\mathbf{xx}} \mathbf{w} + \lambda(1 - \mathbf{w}^H \mathbf{d}). \quad (\text{A.23})$$

The complex derivative following Section A.4 leads to the following solution:

$$\frac{\partial J}{\partial \mathbf{w}} = \Phi_{\mathbf{xx}} \mathbf{w} - \lambda \mathbf{d} = 0 \quad \Leftrightarrow \quad \Phi_{\mathbf{xx}} \mathbf{w} = \lambda \mathbf{d} \quad \Leftrightarrow \quad \mathbf{w} = \lambda \Phi_{\mathbf{xx}}^{-1} \mathbf{d}. \quad (\text{A.24})$$

Finally, we can plug the result of Equation A.24 into the linear constraint and obtain

$$\lambda \mathbf{d}^H \Phi_{\mathbf{xx}}^{-1} \mathbf{d} = 1 \quad \Leftrightarrow \quad \lambda = \frac{1}{\mathbf{d}^H \Phi_{\mathbf{xx}}^{-1} \mathbf{d}}, \quad (\text{A.25})$$

which when again inserted in Equation A.24 results in the well-known MVDR solution as in Equation 3.56.

A.7 Permutation formalism

When discussing permutations Π , it can become quite confusing if the permutation at hand corrects the ordering or if it is what led to the wrong ordering in the first place.

Let us formalize it using an example. A system \mathcal{S} receives correctly sorted items, e.g., $\mathbf{a} = (a_0, a_1, a_2) = (A, B, C)$. It applies a random *system permutation*:

$$\Pi = (2, 0, 1).$$

The resulting order is

$$\begin{aligned} \mathcal{S} \{(A, B, C)\} &= \Pi \circ (A, B, C) \\ &= (a_{\Pi(0)}, a_{\Pi(1)}, a_{\Pi(2)}) \\ &= (C, A, B). \end{aligned}$$

Table A.1: Comparison of most important features of the WSJ-BSS database used in this work and the successor SMS-WSJ [181].

WSJ-BSS	SMS-WSJ
<ul style="list-style-type: none"> • Approximately each unique utterance equally often • 30000, 500, 1500 mixtures • Exclude verbalized punctuation • Reverberation time (T60): 200 ms to 500 ms • Time of flight compensation jointly over all channels • 20 dB to 30 dB additive white Gaussian noise • Only split in speech and noise component • Fixed 2 speaker • First speaker’s utterance determines length 	<ul style="list-style-type: none"> • Each unique utterance exactly equally often • 33561, 982, 1332 mixtures • Exclude verbalized punctuation • Reverberation time (T60): 200 ms to 500 ms • Time of flight compensation jointly over all channels • 20 dB to 30 dB additive white Gaussian noise • Additionally, early-late split • Randomization approach can be generalized to more speakers • Maximum utterance length determines total length: ASR on both possible

To correct the random permutation, we may use the *inverse permutation*⁴:

$$\begin{aligned}\Pi^{-1} &= \text{argsort } \Pi \\ &= (1, 2, 0).\end{aligned}$$

We obtain the initial ordering, by applying the inverse permutation:

$$\begin{aligned}\Pi^{-1} \circ \Pi \circ (A, B, C) &= \Pi^{-1} \circ (C, A, B) \\ &= (a_{\Pi^{-1}(\Pi(0))}, a_{\Pi^{-1}(\Pi(1))}, a_{\Pi^{-1}(\Pi(2))}) \\ &= (A, B, C).\end{aligned}$$

A.8 Comparison of WSJ-BSS and SMS-WSJ

Table A.1 shows a comparison of the WSJ-BSS database used in this work and the successor SMS-WSJ [181].

A.9 More detailed evaluation results

Table A.2 displays different training parameters for a fixed network topology and a fixed number of epochs. After 1000 epochs training with a mini-batch size of 64 and mixtures

⁴ argsort: <https://docs.scipy.org/doc/numpy/reference/generated/numpy.argsort.html>

Table A.2: Comparison for different DFT sizes while limiting the training to 1000 epochs. All models contain four layers of 300 BLSTM cells. The networks which are trained on the entire mixture are initialized with the parameters obtained from the networks trained on 6400 sample segments. No dropout is used.

DFT size	Embedding dimensions	Train samples	SDR / dB					
			SI-SDR		BSS-Eval		Invasive	
			Dev	Test	Dev	Test	Dev	Test
256	20	6400	6.04	6.02	6.66	6.64	9.64	9.69
256	40	6400	5.83	5.82	6.46	6.44	9.47	9.52
512	20	6400	8.17	8.16	8.78	8.75	11.70	11.72
512	40	6400	8.48	8.46	9.08	9.03	11.99	12.01
256	20	entire	6.98	7.03	7.55	7.59	10.47	10.57
256	40	entire	7.07	7.17	7.64	7.73	10.56	10.70
512	20	entire	9.33	9.39	9.89	9.94	12.74	12.82
512	40	entire	9.54	9.50	10.10	10.05	12.95	12.96

randomly cut to segments of 6400 samples (first four rows) better results are obtained for a DFT size of 512. It can be observed that the embedding dimension has an influence on the SDR gains with $E = 40$ dimensions performing slightly better for a DFT size of 512 samples. This is more pronounced for fine-tuned models (last 4 rows) on the entire utterance. As a consequence, all following recipes use a DFT size of 512 samples and $E = 40$ embedding dimensions.

Figure A.3 and Figure A.4 compare different SDR results for DC and DANs, respectively. Both tables present results for a varying number of units per layer, a varying number of layers, different dropout probabilities and whether the systems were trained only on truncated segments, e.g., 6400 samples or the entire mixture. First of all it becomes apparent, that the best configuration is not necessarily equal for both approaches: a DC system appears to be best without dropout while the DAN appears to be slightly better when applying dropout with a dropout probability of 0.2. However, both approaches tend to perform slightly better when using more layers but less LSTM cells per layer. This findings coincide with the observations reported in [47, Table 4]. Further, training on the entire mixture (after pretraining on segments) improved performance in all variants. It is worth keeping in mind that this is on the WSJ0-2mix database which consists of highly overlapped speech. Without reporting further evidence, it is important to note that the effect of training on the entire mixture is more pronounced when training and evaluating on mixtures with less overlap.

Table A.5 lists separation results for a cACGMM with different initializations. In a sense, it can be seen as an extension to Figure 5.5 in which only invasive SDR is displayed over EM-iteration. It can be observed that independent of the initialization method the biggest gains are achieved in the first 100 iterations. Further, the i.i.d. initialization leads to somewhat meaningful results already after 10 iterations. The flag iteration is most promising when no

Table A.3: Comparison of different DC architectures and dropout on the WSJ0-2mix database.

Recurrent units	Dropout	Train samples	SDR / dB					
			SI-SDR		BSS-Eval		Invasive	
			Dev	Test	Dev	Test	Dev	Test
2×600	0.0	6400	7.95	8.16	8.57	8.75	11.48	11.72
2×600	0.2	6400	7.75	7.91	8.37	8.50	11.33	11.49
2×600	0.0	entire	8.92	9.05	9.49	9.61	12.36	12.50
2×600	0.2	entire	8.99	9.08	9.56	9.64	12.43	12.54
4×300	0.0	6400	7.96	7.91	8.58	8.52	11.49	11.49
4×300	0.2	6400	7.05	7.00	7.70	7.64	10.69	10.69
4×300	0.0	entire	9.45	9.45	10.01	10.00	12.86	12.89
4×300	0.2	entire	8.84	8.86	9.42	9.42	12.32	12.35

Table A.4: Comparison of different DAN architectures and dropout on the WSJ0-2mix database.

Recurrent units	Dropout	Train samples	SDR / dB					
			SI-SDR		BSS-Eval		Invasive	
			Dev	Test	Dev	Test	Dev	Test
2×600	0.0	6400	9.20	9.11	9.72	9.62	11.05	11.01
2×600	0.2	6400	9.26	9.22	9.78	9.73	11.23	11.20
2×600	0.0	entire	10.19	10.07	10.66	10.52	11.73	11.62
2×600	0.2	entire	10.31	10.16	10.78	10.63	12.07	11.94
4×300	0.0	6400	9.38	9.22	9.91	9.74	11.36	11.23
4×300	0.2	6400	8.96	8.82	9.52	9.37	11.23	11.11
4×300	0.0	entire	10.25	10.09	10.72	10.56	12.00	11.85
4×300	0.2	entire	10.13	10.06	10.61	10.53	12.10	12.02

Table A.5: Comparison of a cACGMM with different number of iterations. The cACGMM was initialized either by sampling each entry in the affiliation mask i.i.d. from a uniform Dirichlet distribution or with the *flag* initialization tuned towards this particular dataset and consists of $K' = K + 1$ classes.

Initialization	Inline PA	Iterations	SDR / dB				PESQ		STOI	
			BSS-Eval		Invasive					
			Dev	Test	Dev	Test	Dev	Test	Dev	Test
flag	✗	10	5.96	5.76	8.10	7.82	1.84	1.69	0.58	0.71
flag	✗	100	11.46	11.71	14.49	14.85	2.19	2.05	0.66	0.81
flag	✗	500	11.52	11.79	14.59	14.91	2.20	2.05	0.66	0.81
flag	✓	10	5.96	5.76	8.10	7.82	1.84	1.69	0.58	0.71
flag	✓	100	11.42	11.76	14.37	14.89	2.18	2.04	0.66	0.81
flag	✓	500	11.54	11.92	14.53	15.04	2.19	2.05	0.66	0.81
i.i.d.	✗	10	9.73	9.91	12.67	12.82	2.16	2.00	0.66	0.80
i.i.d.	✗	100	11.20	11.25	14.12	14.19	2.23	2.05	0.67	0.82
i.i.d.	✗	500	11.18	11.28	14.10	14.23	2.22	2.05	0.67	0.82
i.i.d.	✓	10	9.92	10.16	12.91	13.21	2.15	2.00	0.66	0.80
i.i.d.	✓	100	12.38	12.22	15.53	15.35	2.26	2.08	0.68	0.82
i.i.d.	✓	500	12.29	12.29	15.42	15.42	2.25	2.08	0.68	0.82

inline permutation alignment is used (Row 3) when comparing to i.i.d. initialization (Row 9). However, flag results are much worse when limiting to 10 iterations only. Overall, an i.i.d. initialization which is easier to implement and assumes less about the structure of the mixture leads to best results when an inline permutation alignment is applied. Consequently, when not otherwise denoted, all presented recipes use this setup.

Glossary

Spectrum A real-valued one-dimensional representation of the amplitude or power of a signal over frequency bins.

Spectrogram A real-valued two-dimensional representation of the amplitude or power of a signal over time frames and frequency bins.

Stationary process A stochastic process is called stationary if all its statistical properties do not change over time.

Wide-sense stationary process A stochastic process of which at least the mean and variance do not change over time. It is a relaxation of a strictly stationary process.

Complex symmetry A complex random variable \tilde{z} is considered to be circularly symmetric when $p_{\tilde{z}}(z) = p_{\tilde{z}}(e^{j\phi}z)$.

Underdetermined, determined, overdetermined While in general these terms are used to discuss the solvability of a system of equations, it is often applied to speech mixtures as follows: *Underdetermined* refers to scenarios with more sources than sensors, *determined* refers to scenarios in which the number of sources and sensors match, and *overdetermined* refers to scenarios in which more sensors than sources are available.

Database A collection of examples, e.g., audio files with transcriptions. Ideally, it consists of predefined dataset splits.

Dataset A dataset is a split of a database into a *train set* used to determine, e.g., network weights, a *development set* used for early stopping or architecture decisions, and a *test set* to report final results. Although the origin of this terminology is unknown, we here refer to [11, Page 33].

List of peer-reviewed publications with own contributions (OC)

- [OC1] L. Drude, A. Chinaev, D. H. Tran Vu, and R. Haeb-Umbach, “Source counting in speech mixtures using a variational EM approach for complex Watson mixture models”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 6834–6838.
- [OC2] L. Drude, A. Chinaev, D. H. T. Vu, and R. Haeb-Umbach, “Towards online source counting in speech mixtures applying a variational EM for complex watson mixture models”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, IEEE, 2014, pp. 213–217.
- [OC3] L. Drude, F. Jacob, and R. Haeb-Umbach, “DOA-estimation based on a complex Watson kernel method”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2015, pp. 255–259.
- [OC4] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge”, in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 444–451.
- [OC5] O. Walter, L. Drude, and R. Haeb-Umbach, “Source counting in speech mixtures by nonparametric bayesian estimation of an infinite gaussian mixture model”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 459–463.
- [OC6] A. Chinaev, J. Heymann, L. Drude, and R. Haeb-Umbach, “Noise-presence-probability-based noise PSD estimation by using DNNs”, in *Speech Communication; ITG Symposium*, VDE, 2016, pp. 1–5.
- [OC7] L. Drude, C. Boeddeker, and R. Haeb-Umbach, “Blind speech separation based on complex spherical k-mode clustering”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 141–145.
- [OC8] L. Drude, B. Raj, and R. Haeb-Umbach, “On the appropriateness of complex-valued neural networks for speech enhancement”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 1745–1749.
- [OC9] T. Glarner, M. M. Momenzadeh, L. Drude, and R. Haeb-Umbach, “Factor graph decoding for speech presence probability estimation”, in *Speech Communication; ITG Symposium*, VDE, 2016, pp. 1–5.
- [OC10] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 196–200.

- [OC11] —, “Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition”, in *International Workshop on Speech Processing in Everyday Environments (CHiME’16)*, 2016, pp. 12–17.
- [OC12] T. Menne, J. Heymann, A. Alexandridis, K. Irie, A. Zeyer, M. Kitza, P. Golik, I. Kulikov, L. Drude, R. Schlüter, et al., “The RWTH/UPB/FORTH system combination for the 4th CHiME challenge evaluation”, in *CHiME-4 workshop*, 2016.
- [OC13] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, “Optimizing neural-network supported acoustic beamforming by algorithmic differentiation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 171–175.
- [OC14] L. Drude and R. Haeb-Umbach, “Tight integration of spatial and spectral features for BSS with deep clustering embeddings”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017.
- [OC15] J. Ebbers, J. Heymann, L. Drude, T. Glarner, R. Haeb-Umbach, and B. Raj, “Hidden markov model variational autoencoder for acoustic unit discovery”, in *INTERSPEECH*, 2017, pp. 488–492.
- [OC16] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, “BEAM-NET: End-to-end training of a beamformer-supported multi-channel ASR system”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [OC17] J. Heymann, L. Drude, and R. Haeb-Umbach, “A generic neural acoustic beamforming architecture for robust multi-channel speech processing”, *Computer Speech & Language*, 2017.
- [OC18] J. Schmalenstroer, J. Heymann, L. Drude, C. Boeddecker, and R. Haeb-Umbach, “Multi-stage coherence drift based sampling rate synchronization for acoustic beamforming”, in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2017, pp. 1–6.
- [OC19] C. Boeddeker, J. Heitkaemper, J. Schmalenstroer, L. Drude, J. Heymann, and R. Haeb-Umbach, “Front-end processing for the CHiME-5 dinner party scenario”, in *CHiME5 Workshop*, 2018.
- [OC20] L. Drude, C. Boeddeker, J. Heymann, R. Haeb-Umbach, K. Kinoshita, M. Delcroix, and T. Nakatani, “Integrating neural network based beamforming and weighted prediction error dereverberation”, in *Interspeech*, 2018, pp. 3043–3047.
- [OC21] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, “NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing”, in *ITG Fachtagung Sprachkommunikation (ITG)*, Oct. 2018.
- [OC22] L. Drude, T. Higuchi, K. Kinoshita, T. Nakatani, and R. Haeb-Umbach, “Dual frequency- and block-permutation alignment for deep learning based block-online blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018.

- [OC23] L. Drude, T. von Neumann, and R. Haeb-Umbach, “Deep attractor networks for speaker re-identification and blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 11–15.
- [OC24] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, “Frame-online DNN-WPE dereverberation”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018.
- [OC25] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, “Listening to each speaker one by one with recurrent selective hearing networks”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5064–5068.
- [OC26] M. Kitza, W. Michel, C. Boeddeker, J. Heitkaemper, T. Menne, R. Schlüter, H. Ney, J. Schmalenstroeer, L. Drude, J. Heymann, et al., “The RWTH/UPB system combination for the CHiME 2018 workshop”, in *CHiME Workshop*, 2018.
- [OC27] L. Drude and R. Haeb-Umbach, “Integration of neural networks and probabilistic spatial models for acoustic blind source separation”, *IEEE Journal of Selected Topics in Signal Processing*, 2019.
- [OC28] L. Drude, D. Hasenklever, and R. Haeb-Umbach, “Unsupervised training of a deep clustering model for multichannel blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019.
- [OC29] L. Drude, J. Heymann, and R. Haeb-Umbach, “Unsupervised training of neural mask-based beamforming”, *arXiv preprint arXiv:1904.01578*, 2019.
- [OC30] J. Ebberts, L. Drude, R. Haeb-Umbach, A. Brendel, and W. Kellermann, “Weakly supervised sound activity detection and event classification in acoustic sensor networks”, in *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, IEEE, 2019.
- [OC31] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, “Joint optimization of neural network-based WPE dereverberation and acoustic model for robust online (asr)”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6655–6659.
- [OC32] G. Kurz, I. Gilitschenski, F. Pfaff, L. Drude, U. D. Hanebeck, R. Haeb-Umbach, and R. Y. Siegwart, “Directional statistics and filtering using libDirectional”, *Journal of Statistical Software*, vol. 89, no. 4, pp. 1–31, 2019.
- [OC33] J. Heitkaemper, D. Jakobeit, C. Boeddeker, L. Drude, and R. Haeb-Umbach, “Demystifying TasNet: A dissecting approach”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020.
- [OC34] T. von Neumann, C. Boeddeker, L. Drude, K. Kinoshita, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, “Multi-talker ASR for an unknown number of sources: Joint training of source counting, separation and ASR”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [OC35] T. von Neumann, K. Kinoshita, L. Drude, C. Boeddeker, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, “End-to-end training of time domain audio separation and recognition”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020.

Bibliography

- [1] M. Wölfel and J. W. McDonough, *Distant speech recognition*. Wiley Online Library, 2009 (cited on p. 4).
- [2] 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 (cited on pp. 4, 28, 29, 40).
- [3] R. E. Crochiere and L. R. Rabiner, *Multirate digital signal processing*. Prentice-Hall Englewood Cliffs, N.J, 1983 (cited on p. 4).
- [4] A. Gilloire and M. Vetterli, “Adaptive filtering in subbands with critical sampling: Analysis, experiments, and application to acoustic echo cancellation”, *IEEE Transactions on Signal Processing*, vol. 40, no. 8, pp. 1862–1875, 1992, ISSN: 1053-587X (cited on p. 4).
- [5] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012 (cited on pp. 6, 7).
- [6] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2 (cited on p. 7).
- [7] K. Pearson, “Contributions to the mathematical theory of evolution”, *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894 (cited on p. 8).
- [8] 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 (cited on pp. 8, 11–13).
- [9] B. Everitt, “Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms”, *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 33, no. 2, pp. 205–215, 1984 (cited on p. 8).
- [10] K. Lange, *MM optimization algorithms*. SIAM, 2016, vol. 147 (cited on pp. 8, 18).
- [11] C. M. Bishop et al., *Pattern recognition and machine learning*. Springer New York, 2006, vol. 1 (cited on pp. 9, 10, 17, 35, 50, 123).
- [12] R. Haeb-Umbach, *Statistical and machine learning – course script*, https://groups.uni-paderborn.de/nt/lehre/slvme/script/script_en.pdf, Accessed: 2019-12-10, 2019 (cited on p. 9).
- [13] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014 (cited on pp. 9, 14).

- [14] T. K. Moon, “The EM algorithm in signal processing”, *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, 1996 (cited on p. 11).
- [15] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382 (cited on p. 11).
- [16] R. M. Neal and G. E. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants”, in *Learning in graphical models*, Springer, 1998, pp. 355–368 (cited on pp. 13, 14).
- [17] L. Drude, “Variational Bayesian inference for complex Watson mixture models applied to blind source separation”, Master’s thesis, Paderborn University, 2014 (cited on pp. 13, 15, 17, 29–31, 34, 35, 38, 114).
- [18] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, “The variational approximation for Bayesian inference”, 2008 (cited on pp. 14, 15, 17).
- [19] Y. Sun, P. Babu, and D. P. Palomar, “Majorization-minimization algorithms in signal processing, communications, and machine learning”, *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2016 (cited on p. 18).
- [20] J. Azcarreta, N. Ito, S. Araki, and T. Nakatani, “Permutation-free CGMM: Complex Gaussian mixture model with inverse Wishart mixture model based spatial prior for permutation-free source separation and source counting”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 51–55 (cited on p. 18).
- [21] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda, “Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones”, *Acoustical Science and Technology*, vol. 22, no. 2, pp. 149–157, 2001 (cited on pp. 19, 20).
- [22] O. Yilmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking”, *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004 (cited on pp. 19, 20, 27, 67).
- [23] S. Makino, T.-W. Lee, and H. Sawada, *Blind speech separation*. Springer, 2007, vol. 615 (cited on pp. 20, 27, 67).
- [24] D. Ellis, in *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006, ch. Model-based scene analysis, pp. 115–146 (cited on p. 20).
- [25] 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 (cited on p. 20).
- [26] F. R. Bach and M. I. Jordan, “Blind one-microphone speech separation: A spectral learning approach”, in *Advances in neural information processing systems (NIPS)*, 2005, pp. 65–72 (cited on p. 20).
- [27] S. T. Roweis, “Factorial models and refiltering for speech separation and denoising”, in *European Conference on Speech Communication and Technology (EUROSPEECH)*, 2003 (cited on p. 20).

- [28] Z. Ghahramani and M. I. Jordan, “Factorial hidden Markov models”, in *Advances in Neural Information Processing Systems (NIPS)*, 1996, pp. 472–478 (cited on p. 20).
- [29] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, “Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system”, in *International Conference on Spoken Language Processing (SLT)*, 2006, pp. 97–100 (cited on p. 20).
- [30] M. Weintraub, “A theory and computational model of auditory monaural sound separation”, PhD thesis, Stanford University, 1985 (cited on p. 21).
- [31] G. J. Brown and M. Cooke, “Computational auditory scene analysis”, *Computer Speech & Language*, vol. 8, no. 4, pp. 297–336, 1994 (cited on p. 21).
- [32] G. Hu and D. Wang, “Monaural speech segregation based on pitch tracking and amplitude modulation”, *IEEE Transactions on neural networks*, vol. 15, no. 5, pp. 1135–1150, 2004 (cited on p. 21).
- [33] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2001 (cited on p. 21).
- [34] P. Comon, “Independent component analysis, a new concept?”, *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994 (cited on p. 21).
- [35] P. Smaragdis, in *Speech separation by humans and machines*. Springer Science & Business Media, 2004, ch. Exploiting redundancy to construct listening systems, pp. 83–95 (cited on p. 21).
- [36] M. Zibulevsky and B. A. Pearlmutter, “Blind source separation by sparse decomposition in a signal dictionary”, *Neural computation*, vol. 13, no. 4, pp. 863–882, 2001 (cited on p. 21).
- [37] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization”, *Nature*, vol. 401, no. 6755, p. 788, 1999 (cited on p. 21).
- [38] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 15, no. 3, pp. 1066–1074, 2007 (cited on p. 21).
- [39] P. Smaragdis, “Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs”, in *International Conference on Independent Component Analysis and Signal Separation*, Springer, 2004, pp. 494–499 (cited on p. 21).
- [40] J. Du, Y. Tu, Y. Xu, L. Dai, and C.-H. Lee, “Speech separation of a target speaker based on deep neural networks”, in *2014 12th International Conference on Signal Processing (ICSP)*, IEEE, 2014, pp. 473–477 (cited on p. 21).
- [41] Y. Tu, J. Du, Y. Xu, L. Dai, and C.-H. Lee, “Deep neural network based speech separation for robust speech recognition”, in *International Conference on Signal Processing (ICSP)*, IEEE, 2014, pp. 532–536 (cited on pp. 21, 22).

- [42] ———, “Speech separation based on improved deep neural networks with dual outputs of speech features for both target and interfering speakers”, in *International Symposium on Chinese Spoken Language Processing*, IEEE, 2014, pp. 250–254 (cited on p. 21).
- [43] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 31–35 (cited on pp. 21–23, 26, 56, 59, 66, 67, 75).
- [44] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017 (cited on pp. 21, 22, 25, 56).
- [45] 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 (cited on pp. 22, 25, 56, 73–75).
- [46] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017 (cited on pp. 22–24, 56, 71, 75, 95).
- [47] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, “Single-channel multi-speaker separation using deep clustering”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016 (cited on pp. 23, 24, 26, 56, 66, 67, 75, 120).
- [48] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, “Alternative objective functions for deep clustering”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018 (cited on pp. 23, 26, 66, 75).
- [49] ———, “Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018 (cited on pp. 23, 39, 91).
- [50] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, “Deep clustering and conventional networks for music separation: Stronger together”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 61–65 (cited on p. 23).
- [51] L. Drude and R. Haeb-Umbach, “Integration of neural networks and probabilistic spatial models for acoustic blind source separation”, *IEEE Journal of Selected Topics in Signal Processing*, 2019 (cited on pp. 24, 49–51, 53, 55, 66, 67, 71, 89).
- [52] Y. Luo and N. Mesgarani, “TasNet: Time-domain audio separation network for real-time, single-channel speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018 (cited on p. 25).
- [53] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation”, *arXiv preprint arXiv:1910.06379*, 2019 (cited on p. 25).

- [54] H. Sawada, N. Ono, H. Kameoka, D. Kitamura, and H. Saruwatari, “A review of blind source separation methods: Two converging routes to ILRMA originating from ICA and NMF”, *APSIPA Transactions on Signal and Information Processing*, vol. 8, 2019 (cited on pp. 26, 27, 48).
- [55] M. I. Mandel, R. J. Weiss, and D. P. Ellis, “Model-based expectation-maximization source separation and localization”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 382–394, 2010 (cited on pp. 26, 27, 39).
- [56] A. Ozerov and C. Févotte, “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2009 (cited on p. 26).
- [57] S. Arberet, A. Ozerov, N. Q. Duong, E. Vincent, R. Gribonval, F. Bimbot, and P. Vanderghelynst, “Nonnegative matrix factorization and spatial covariance model for under-determined reverberant audio source separation”, in *International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, IEEE, 2010, pp. 1–4 (cited on pp. 26, 48, 49).
- [58] S. Makino, *Audio Source Separation*. Springer, 2018 (cited on pp. 26, 48).
- [59] 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 (cited on p. 26).
- [60] 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 (cited on p. 26).
- [61] I. Lee, T. Kim, and T.-W. Lee, “Complex FastIVA: A robust maximum likelihood approach of MICA for convolutive BSS”, in *International Conference on Independent Component Analysis and Signal Separation*, Springer, 2006, pp. 625–632 (cited on p. 26).
- [62] 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 (TASLP)*, vol. 24, no. 9, pp. 1622–1637, 2016 (cited on p. 26).
- [63] A. Jourjine, S. Rickard, and O. Yilmaz, “Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures”, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, vol. 5, 2000, pp. 2985–2988 (cited on p. 27).
- [64] N. Roman, D. Wang, and G. J. Brown, “Speech segregation based on sound localization”, *The Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2236–2252, 2003 (cited on p. 27).
- [65] S. Araki, H. Sawada, R. Mukai, and S. Makino, “Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors”, *Signal Processing*, vol. 87, no. 8, pp. 1833–1847, 2007 (cited on p. 27).

- [66] H. Sawada, S. Araki, and S. Makino, “Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, 2011 (cited on pp. 27, 29–31).
- [67] Z. Chen, J. Li, X. Xiao, T. Yoshioka, H. Wang, Z. Wang, and Y. Gong, “Cracking the cocktail party problem by multi-beam deep attractor network”, in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2017, pp. 437–444 (cited on pp. 28, 40).
- [68] A. A. Nugraha, “Deep neural networks for source separation and noise-robust speech recognition”, PhD thesis, 2017 (cited on p. 28).
- [69] V. G. Reju, S. N. Koh, and Y. Soon, “Underdetermined convolutive blind source separation via time-frequency masking”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 101–116, 2009 (cited on p. 28).
- [70] N. Ito, C. Schymura, S. Araki, and T. Nakatani, “Noisy cGMM: Complex gaussian mixture model with non-sparse noise model for joint source separation and denoising”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 1662–1666 (cited on p. 29).
- [71] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, “Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2008, pp. 85–88 (cited on p. 29).
- [72] —, “Speech dereverberation based on variance-normalized delayed linear prediction”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010 (cited on p. 29).
- [73] N. Ito, S. Araki, T. Yoshioka, and T. Nakatani, “Relaxed disjointness based clustering for joint blind source separation and dereverberation”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2014 (cited on pp. 29, 35, 59).
- [74] S. Araki, H. Sawada, R. Mukai, and S. Makino, “Normalized observation vector clustering approach for sparse source separation”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2006, pp. 1–5 (cited on p. 29).
- [75] H. Sawada, S. Araki, and S. Makino, “Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain BSS”, in *International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2007, pp. 3247–3250 (cited on p. 29).
- [76] 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 (cited on p. 29).
- [77] H. Sawada, R. Mukai, S. de la Kethulle de Ryhove, S. Araki, and S. Makino, “Spectral smoothing for frequency-domain blind source separation”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, Citeseer, 2003, pp. 311–314 (cited on p. 29).

- [78] K. Reindl, Y. Zheng, A. Schwarz, S. Meier, R. Maas, A. Sehr, and W. Kellermann, “A stereophonic acoustic signal extraction scheme for noisy and reverberant environments”, *Computer Speech & Language*, vol. 27, no. 3, pp. 726–745, 2013 (cited on p. 29).
- [79] H. Sawada, S. Araki, R. Mukai, and S. Makino, “Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation”, *Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1592–1604, 2007 (cited on p. 29).
- [80] 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”, *Transactions on speech and audio processing*, vol. 12, no. 5, pp. 530–538, 2004 (cited on p. 29).
- [81] D. H. Tran Vu, “Integrated multi-channel blind speech separation and noise reduction using 2D hidden Markov models”, PhD thesis, Paderborn University, 2015, to be published (cited on pp. 29, 31, 46, 79).
- [82] H. Sawada, S. Araki, and S. Makino, “A two-stage frequency-domain blind source separation method for underdetermined convolutive mixtures”, in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 2007, pp. 139–142 (cited on p. 29).
- [83] C. Boeddeker, “Untersuchungen zur permutationsfreien blinden Quellentrennung”, Master’s thesis, Paderborn University, 2015 (cited on pp. 29–31).
- [84] N. Ito, S. Araki, and T. Nakatani, “Data-driven and physical model-based designs of probabilistic spatial dictionary for online meeting diarization and adaptive beam-forming”, in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, 2017, pp. 1165–1169 (cited on p. 29).
- [85] D. H. Tran Vu and R. Haeb-Umbach, “On initial seed selection for frequency domain blind speech separation”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 1757–1760 (cited on pp. 30, 50).
- [86] N. Q. K. Duong, E. Vincent, and R. Gribonval, “Under-determined reverberant audio source separation using a full-rank spatial covariance model”, *Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010 (cited on pp. 30, 35, 48).
- [87] H. Sawada, R. Ikeshita, N. Ito, and T. Nakatani, “Computational acceleration and smart initialization of full-rank spatial covariance analysis”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2019, pp. 1–5 (cited on p. 30).
- [88] S. P. Lloyd, “Least squares quantization in pcm”, *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982 (cited on pp. 30, 50).
- [89] L. Drude, C. Boeddeker, and R. Haeb-Umbach, “Blind speech separation based on complex spherical k-mode clustering”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 141–145 (cited on pp. 30, 32, 50).

- [90] N. Ito, S. Araki, and T. Nakatani, “Permutation-free convolutive blind source separation via full-band clustering based on frequency-independent source presence priors”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2013, pp. 3238–3242 (cited on p. 30).
- [91] D. H. Tran Vu and R. Haeb-Umbach, “Blind speech separation employing directional statistics in an expectation maximization framework”, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2010, pp. 241–244 (cited on p. 30).
- [92] —, “An EM approach to integrated multichannel speech separation and noise suppression”, in *Proc. Int. Workshop Acoust. Signal Enhancement (IWAENC)*, 2010, pp. 1–4 (cited on p. 30).
- [93] K. V. Mardia and I. L. Dryden, “The complex Watson distribution and shape analysis”, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 4, pp. 913–926, 1999 (cited on pp. 30–32).
- [94] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST handbook of mathematical functions hardback and CD-ROM*. Cambridge university press, 2010 (cited on p. 31).
- [95] G. Kurz, I. Gilitschenski, F. Pfaff, L. Drude, U. D. Hanebeck, R. Haeb-Umbach, and R. Y. Siegwart, “Directional statistics and filtering using libDirectional”, *Journal of Statistical Software*, vol. 89, no. 4, pp. 1–31, 2019 (cited on p. 31).
- [96] N. Ito, S. Araki, and T. Nakatani, “Modeling audio directional statistics using a complex Bingham mixture model for blind source extraction from diffuse noise”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 465–468 (cited on pp. 33, 77).
- [97] J. T. Kent, “The complex Bingham distribution and shape analysis”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 56, no. 2, pp. 285–299, 1994 (cited on pp. 33, 113).
- [98] L. Drude, A. Chinaev, D. H. Tran Vu, and R. Haeb-Umbach, “Source counting in speech mixtures using a variational EM approach for complex Watson mixture models”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 6834–6838 (cited on pp. 34, 35).
- [99] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation, 1965, vol. 55 (cited on p. 35).
- [100] 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 (IWAENC)*, IEEE, 2005, pp. 78–81 (cited on p. 35).
- [101] 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 (ICA)*, Springer, 2009, pp. 775–782 (cited on p. 35).

- [102] J. Thiemann and E. Vincent, “A fast EM algorithm for Gaussian model-based source separation”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2013, pp. 1–5 (cited on pp. 35, 48, 49).
- [103] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, et al., “The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices”, in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 436–443 (cited on p. 35).
- [104] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘chime’ speech separation and recognition challenge: Dataset, task and baselines”, in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 504–511 (cited on p. 35).
- [105] N. R. Goodman, “Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction)”, *The Annals of mathematical statistics*, vol. 34, no. 1, pp. 152–177, 1963 (cited on p. 36).
- [106] L. Drude and R. Haeb-Umbach, “Tight integration of spatial and spectral features for BSS with deep clustering embeddings”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017 (cited on pp. 36, 47, 49–51, 53, 55, 66, 67, 77).
- [107] D. E. Tyler, “Statistical analysis for the angular central Gaussian distribution on the sphere”, *Biometrika*, vol. 74, no. 3, pp. 579–589, 1987 (cited on p. 36).
- [108] K. V. Mardia and P. E. Jupp, *Directional statistics*. John Wiley & Sons, 2009, vol. 494 (cited on p. 36).
- [109] P. Paine, S. P. Preston, M. Tsagris, and A. T. Wood, “An elliptically symmetric angular Gaussian distribution”, *Statistics and Computing*, vol. 28, no. 3, pp. 689–697, 2018 (cited on p. 37).
- [110] D. Hasenklever, “Unsupervised training of neural network-based multichannel blind source separation”, Master’s thesis, Paderborn University, 2019 (cited on pp. 37, 38).
- [111] N. Ito, S. Araki, and T. Nakatani, “Complex angular central Gaussian mixture model for directional statistics in mask-based microphone array signal processing”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2016, pp. 1153–1157 (cited on pp. 36, 38, 76, 77).
- [112] J. T. Kent, “Data analysis for shapes and images”, *Journal of statistical planning and inference*, vol. 57, no. 2, pp. 181–193, 1997 (cited on p. 37).
- [113] E. Vincent, N. Bertin, R. Gribonval, F. Bimbot, et al., “From blind to guided audio source separation”, *IEEE Signal Processing Magazine*, 2013 (cited on p. 38).
- [114] B. Loesch, “Complex blind source separation with audio applications”, PhD thesis, Stuttgart University, 2013 (cited on p. 38).
- [115] N. Q. K. Duong, A. Ozerov, and L. Chevallier, “Temporal annotation-based audio source separation using weighted nonnegative matrix factorization”, in *International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, IEEE, 2014, pp. 220–224 (cited on p. 38).

- [116] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, “Front-end processing for the CHiME-5 dinner party scenario”, in *CHiME5 Workshop*, 2018 (cited on pp. 38, 46).
- [117] N. Kanda, C. Boeddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, “Guided source separation meets a strong ASR backend: Hitachi/paderborn university joint investigation for dinner party ASR”, *arXiv preprint arXiv:1905.12230*, 2019 (cited on p. 38).
- [118] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction”, *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979 (cited on p. 39).
- [119] J. Barker, L. Josifovski, M. Cooke, and P. Green, “Soft decisions in missing data techniques for robust automatic speech recognition”, in *International Conference on Spoken Language Processing (ICSLP)*, 2000 (cited on p. 39).
- [120] M. L. Seltzer, B. Raj, and R. M. Stern, “A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition”, *Speech Communication*, vol. 43, no. 4, pp. 379–393, 2004 (cited on p. 39).
- [121] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015 (cited on pp. 40, 73).
- [122] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004 (cited on p. 40).
- [123] R. A. Monzingo and T. W. Miller, *Introduction to adaptive arrays*. Scitech publishing, 2004 (cited on p. 40).
- [124] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1 (cited on p. 40).
- [125] E. Hadad, F. Heese, P. Vary, and S. Gannot, “Multichannel audio database in various acoustic environments”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2014, pp. 313–317 (cited on pp. 40, 61).
- [126] J. Bitzer and K. U. Simmer, “Superdirective microphone arrays”, in *Microphone arrays*, Springer, 2001, pp. 19–38 (cited on p. 40).
- [127] E. Warsitz, “Mehrkanalige Sprachsignalverbesserung durch adaptive Lösung eines Eigenwertproblems im Frequenzbereich”, Dissertation, University of Paderborn, 2008 (cited on p. 40).
- [128] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge”, in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 444–451 (cited on pp. 41, 66).
- [129] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 196–200 (cited on pp. 41, 43).

- [130] —, “A generic neural acoustic beamforming architecture for robust multi-channel speech processing”, *Computer Speech & Language*, 2017 (cited on p. 41).
- [131] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, “Improved MVDR beamforming using single-channel mask prediction networks”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 1981–1985 (cited on pp. 41, 44).
- [132] Y. Kubo, T. Nakatani, M. Delcroix, K. Kinoshita, and S. Araki, “Mask-based MVDR beamformer for noisy multisource environments: Introduction of time-varying spatial covariance model”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6855–6859 (cited on p. 42).
- [133] B. Van Veen and K. Buckley, “Beamforming techniques for spatial filtering”, *Digital Signal Processing Handbook*, pp. 61–1, 1997 (cited on pp. 42, 43).
- [134] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1529–1539, 2007 (cited on pp. 42, 46).
- [135] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012 (cited on p. 42).
- [136] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, “On the computation of complex-valued gradients with application to statistically optimum beamforming”, *arXiv preprint arXiv:1701.00392*, 2017 (cited on pp. 43, 116).
- [137] J. Heymann, “Robust multi-channel speech recognition with neural network supported statistical beamforming”, PhD thesis, Paderborn University, 2020, to be published (cited on pp. 43, 46, 64).
- [138] J. Capon, “High-resolution frequency-wavenumber spectrum analysis”, *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969 (cited on p. 43).
- [139] S. Markovich, S. Gannot, and I. Cohen, “Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1071–1086, 2009 (cited on p. 44).
- [140] Z. Wang, E. Vincent, R. Serizel, and Y. Yan, “Rank-1 constrained multichannel wiener filter for speech recognition in noisy environments”, *Computer Speech & Language*, vol. 49, pp. 37–51, 2018 (cited on pp. 44–46, 83).
- [141] S. Araki, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, T. Higuchi, T. Yoshioka, D. Tran, S. Karita, and T. Nakatani, “Online meeting recognition in noisy environments with time-frequency mask based MVDR beamforming”, in *Hands-free Speech Communications and Microphone Arrays (HSCMA)*, Mar. 2017, pp. 16–20 (cited on p. 44).
- [142] M. Souden, J. Benesty, and S. Affes, “A study of the LCMV and MVDR noise reduction filters”, *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4925–4935, 2010 (cited on p. 44).

- [143] C. Boeddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, “Exploring practical aspects of neural mask-based beamforming for far-field speech recognition”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 6697–6701 (cited on p. 44).
- [144] O. L. Frost, “An algorithm for linearly constrained adaptive array processing”, *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972 (cited on pp. 44, 45).
- [145] A. Aroudi and S. Doclo, “Cognitive-driven binaural LCMV beamformer using EEG-based auditory attention decoding”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 406–410 (cited on p. 45).
- [146] S. E. Chazan, S. Gannot, and J. Goldberger, “Attention-based neural network for joint diarization and speaker extraction”, in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, IEEE, 2018, pp. 301–305 (cited on p. 45).
- [147] S. Doclo and M. Moonen, “GSVD-based optimal filtering for single and multimicrophone speech enhancement”, *IEEE Transactions on signal processing*, vol. 50, no. 9, pp. 2230–2244, 2002 (cited on p. 45).
- [148] A. Spriet, M. Moonen, and J. Wouters, “Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction”, *Signal Processing*, vol. 84, no. 12, pp. 2367–2387, 2004 (cited on pp. 45, 46).
- [149] J. Schmalenstroeer, J. Heymann, L. Drude, C. Boeddecker, and R. Haeb-Umbach, “Multi-stage coherence drift based sampling rate synchronization for acoustic beamforming”, in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2017, pp. 1–6 (cited on pp. 46, 85).
- [150] K. U. Simmer, J. Bitzer, and C. Marro, “Post-filtering techniques”, in *Microphone Arrays*, Springer, 2001, pp. 39–60 (cited on p. 46).
- [151] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada, “A multichannel mmse-based framework for speech source separation and noise reduction”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1913–1928, 2013 (cited on pp. 46, 49, 50).
- [152] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer handbook of speech processing*. Springer, 2008 (cited on p. 46).
- [153] L. K. Hansen and P. Salamon, “Neural network ensembles”, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 993–1001, 1990 (cited on pp. 47, 53).
- [154] J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)”, in *IEEE Workshop on Automatic Speech Recognition and Understanding*, IEEE, 1997, pp. 347–354 (cited on p. 47).
- [155] B. Hoffmeister, T. Klein, R. Schlüter, and H. Ney, “Frame based system combination and a comparison with weighted ROVER and CNC”, in *Ninth International Conference on Spoken Language Processing*, 2006 (cited on p. 47).
- [156] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1118–1133, 2011 (cited on pp. 48, 49).

- [157] 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 (cited on pp. 48–50).
- [158] 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 (cited on pp. 48, 49).
- [159] H. Kameoka, L. Li, S. Inoue, and S. Makino, “Semi-blind source separation with multichannel variational autoencoder”, *arXiv preprint arXiv:1808.00892*, 2018 (cited on pp. 48, 49).
- [160] T. Nakatani, S. Araki, T. Yoshioka, and M. Fujimoto, “Multichannel source separation based on source location cue with log-spectral shaping by hidden markov source model”, in *Eleventh Annual Conference of the International Speech Communication Association*, 2010 (cited on pp. 48, 49).
- [161] —, “Joint unsupervised learning of hidden Markov source models and source location models for multichannel source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011, pp. 237–240 (cited on pp. 49, 50).
- [162] T. Nakatani, S. Araki, T. Yoshioka, M. Delcroix, and M. Fujimoto, “Dominance based integration of spatial and spectral features for speech enhancement”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2516–2531, 2013, ISSN: 1558-7916 (cited on pp. 49, 50).
- [163] H. Meutzner, S. Araki, M. Fujimoto, and T. Nakatani, “A generative-discriminative hybrid approach to multi-channel noise reduction for robust automatic speech recognition”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5740–5744 (cited on pp. 49, 50).
- [164] D. H. Tran Vu and R. Haeb-Umbach, “Blind speech separation exploiting temporal and spectral correlations using 2D-HMMs”, in *European Signal Processing Conference (EUSIPCO)*, IEEE, 2013, pp. 1–5 (cited on pp. 49, 50).
- [165] T. Nakatani, N. Ito, T. Higuchi, S. Araki, and K. Kinoshita, “Integrating DNN-based and spatial clustering-based mask estimation for robust MVDR beamforming”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017 (cited on pp. 49–51, 103).
- [166] L. Drude, T. Higuchi, K. Kinoshita, T. Nakatani, and R. Haeb-Umbach, “Dual frequency- and block-permutation alignment for deep learning based block-online blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018 (cited on pp. 49, 51, 53, 71).
- [167] L. Drude, D. Hasenklever, and R. Haeb-Umbach, “Unsupervised training of a deep clustering model for multichannel blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019 (cited on pp. 49, 56, 57, 93).

- [168] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding”, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035 (cited on p. 50).
- [169] D. Ditter and T. Gerkmann, “Influence of speaker-specific parameters on speech separation systems”, *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 4584–4588, 2019 (cited on pp. 53, 98).
- [170] T. Cord-Landwehr, “Integration neural networks and probabilistic spatial mixture models for multi-channel blind source separation”, Master’s thesis, Paderborn University, 2019 (cited on p. 53).
- [171] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von Mises-Fisher distributions”, *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005 (cited on pp. 53–55).
- [172] Z. Chen, Y. Luo, and N. Mesgarani, “Speaker-independent speech separation with deep attractor network”, *arXiv preprint arXiv:1707.03634*, 2017 (cited on p. 56).
- [173] Y. Zhou and Y. Qian, “Robust mask estimation by integrating neural network-based and clustering-based approaches for adaptive acoustic beamforming”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 536–540 (cited on pp. 56, 57).
- [174] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5210–5214 (cited on p. 56).
- [175] E. Tzinis, S. Venkataramani, and P. Smaragdis, “Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019 (cited on pp. 56, 57).
- [176] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo, “Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019 (cited on pp. 56, 57).
- [177] L. Drude, J. Heymann, and R. Haeb-Umbach, “Unsupervised training of neural mask-based beamforming”, *arXiv preprint arXiv:1904.01578*, 2019 (cited on pp. 56, 57, 116).
- [178] Y. Bando, Y. Sasaki, and K. Yoshii, “Deep bayesian unsupervised source separation based on a complex gaussian mixture model”, in *International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2019, pp. 1–6 (cited on p. 57).
- [179] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR–half-baked or well done?”, in *ICASSP 2019-2019 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 626–630 (cited on pp. 58, 59, 75).
- [180] 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 (cited on p. 58).

- [181] L. Drude, J. Heitkaemper, C. Boeddeker, and R. Haeb-Umbach, “SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition”, *arXiv preprint arXiv:1910.13934*, 2019 (cited on pp. 58, 61, 119).
- [182] 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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, vol. 2, 2001, pp. 749–752 (cited on p. 58).
- [183] 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, ISSN: 1558-7916 (cited on p. 59).
- [184] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, “WHAM!: Extending speech separation to noisy environments”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Sep. 2019 (cited on p. 59).
- [185] D. B. Paul and J. M. Baker, “The design for the Wall Street Journal-based CSR corpus”, in *Workshop on Speech and Natural Language (HLT)*, Association for Computational Linguistics, 1992, pp. 357–362 (cited on pp. 60, 61).
- [186] J. Garofolo, D. Graff, D. Paul, and D. Pallett, “CSR-I (WSJ0) Complete LDC93S6A”, *Web Download. Philadelphia: Linguistic Data Consortium*, 1993 (cited on pp. 60, 61).
- [187] 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, pp. 943–950, 1979 (cited on p. 61).
- [188] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, “The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): Specification and initial experiments”, in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2005, pp. 357–362 (cited on p. 63).
- [189] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, “RWTH ASR systems for LibriSpeech: Hybrid vs attention – w/o data augmentation”, *arXiv preprint arXiv:1905.03072*, 2019 (cited on p. 64).
- [190] J. Heymann, L. Drude, and R. Haeb-Umbach, “Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition”, in *International Workshop on Speech Processing in Everyday Environments (CHiME’16)*, 2016, pp. 12–17 (cited on p. 64).
- [191] C. Zorila, C. Boeddeker, R. Doddipatla, and R. Haeb-Umbach, “An investigation into the effectiveness of enhancement in asr training and test for chime-5 dinner party transcription”, in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019 (cited on p. 65).

- [192] J. Zegers and H. van Hamme, “CNN-LSTM models for multi-speaker source separation using Bayesian hyper parameter optimization”, *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019 (cited on p. 65).
- [193] *A recipe for training neural networks*, <http://karpathy.github.io/2019/04/25/recipe/>, Accessed: 2019-07-16 (cited on p. 65).
- [194] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014 (cited on p. 65).
- [195] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997 (cited on p. 66).
- [196] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks”, *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997 (cited on p. 66).
- [197] Y. Gal, “A theoretically grounded application of dropout in recurrent neural networks”, *arXiv preprint arXiv:1512.05287*, 2015 (cited on p. 67).
- [198] S. Semeniuta, A. Severyn, and E. Barth, “Recurrent dropout without memory loss”, *arXiv preprint arXiv:1603.05118*, 2016 (cited on p. 67).
- [199] T. Moon, H. Choi, H. Lee, and I. Song, “RNNDROP: A novel dropout for RNNs in ASR”, in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 65–70 (cited on p. 67).
- [200] L. Drude, T. von Neumann, and R. Haeb-Umbach, “Deep attractor networks for speaker re-identification and blind source separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 11–15 (cited on p. 71).
- [201] Y. Luo, Z. Chen, and N. Mesgarani, “Speaker-independent speech separation with deep attractor network”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018 (cited on p. 75).
- [202] J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith, “Show your work: Improved reporting of experimental results”, *arXiv preprint arXiv:1909.03004*, 2019 (cited on p. 104).
- [203] *Guidelines for authors submitting code & software*. [Online]. Available: <https://www.nature.com/documents/GuidelinesCodePublication.pdf> (visited on 05/16/2013) (cited on p. 104).
- [204] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, “Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping”, *arXiv preprint arXiv:2002.06305*, 2020 (cited on p. 105).
- [205] M. R. Chernick, *Bootstrap methods: A guide for practitioners and researchers*. John Wiley & Sons, 2011, vol. 619 (cited on p. 106).

- [206] M. Waskom, O. Botvinnik, J. Ostblom, M. Gelbart, S. Lukauskas, P. Hobson, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, C. Swain, A. Miles, T. Brunner, D. O’Kane, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, and Brian, *Mwaskom/seaborn: V0.10.1 (april 2020)*, version v0.10.1, Apr. 2020. DOI: 10.5281/zenodo.3767070. [Online]. Available: <https://doi.org/10.5281/zenodo.3767070> (cited on p. 107).
- [207] P. Dragicevic, “Fair statistical communication in hci”, in *Modern statistical methods for HCI*, Springer, 2016, pp. 291–330 (cited on p. 107).
- [208] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*. Pearson, 2017, ISBN: 9780134115856 (cited on p. 109).
- [209] R. L. Wasserstein and N. A. Lazar, *The ASA statement on p-values: Context, process, and purpose*, 2016 (cited on p. 109).
- [210] J. L. W. V. Jensen et al., “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”, *Acta mathematica*, vol. 30, pp. 175–193, 1906 (cited on p. 113).
- [211] T. Adali and S. Haykin, *Adaptive signal processing: next generation solutions*. John Wiley & Sons, 2010, vol. 55 (cited on pp. 115, 116).
- [212] R. Remmert, *Theory of Complex Functions*. Springer Science & Business Media, 1991, vol. 122 (cited on p. 115).
- [213] W. Wirtinger, “Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen”, *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, 1927 (cited on p. 116).
- [214] D. H. Brandwood, “A complex gradient operator and its application in adaptive array theory”, in *IEE Proceedings F (Communications, Radar and Signal Processing)*, IET, vol. 130, 1983, pp. 11–16 (cited on p. 116).
- [215] P. J. Schreier and L. L. Scharf, *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals*. Cambridge University Press, 2010 (cited on p. 116).
- [216] T. Adali and P. J. Schreier, “Optimization and estimation of complex-valued signals: Theory and applications in filtering and blind source separation”, *IEEE Signal Processing Magazine*, vol. 5, no. 31, pp. 112–128, 2014 (cited on p. 116).
- [217] K. B. Petersen and M. S. Pedersen, “The matrix cookbook”, *Technical University of Denmark*, vol. 7, p. 15, 2015 (cited on p. 116).
- [218] L. Drude, B. Raj, and R. Haeb-Umbach, “On the appropriateness of complex-valued neural networks for speech enhancement”, in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 1745–1749 (cited on p. 116).
- [219] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, “BEAM-NET: End-to-end training of a beamformer-supported multi-channel ASR system”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017 (cited on p. 116).