Informed Audio Source Separation with Deep Learning in Limited Data Settings

Kilian Schulze-Forster

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 20XXIPPAXXXX

Thèse de doctorat

Informed Audio Source Separation with Deep Learning in Limited Data Settings

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique, données, intelligence artificielle

Thèse présentée et soutenue à Palaiseau, le 09/12/2021, par

KILIAN SCHULZE-FORSTER

Composition du Jury :

Emmanuel Vincent Directeur de recherche, Inria Nancy - Grand Est	Président
Xavier Serra Professeur, Universitat Pompeu Fabra	Rapporteur
Laurent Girin Professeur, Grenoble-INP, Institut Polytechnique de Grenoble	Rapporteur
Hélène-Camille Crayencour Chargée de recherche, CNRS	Examinatrice
Roland Badeau Professeur, Télécom Paris	Directeur de thèse
Gaël Richard Professeur, Télécom Paris	Co-directeur de thèse
Clément S. J. Doire Senior research scientist, Sonos Inc.	Invité

Acknowledgements

The past three years were an exciting, intense, challenging, and amazing journey which helped me grow in several ways. I am deeply grateful for the support I received all along the way.

I was very fortunate to have a fantastic team of supervisors. Thank you, Roland and Gaël, for your guidance, constructive advice, as well as the freedom and trust you gave me. Thank you, Clément, for many nice discussions, valuable feedback, and for being so involved beyond my stay at Audionamix.

It was a pleasure to be a part of the training network *MIP-Frontiers*[1] which funded my research and provided a great program of inspiring talks and workshops with many opportunities to connect with the international music information research community. A big thanks to the whole consortium! I was also very glad to go on this journey together with my Paris-based MIP-Frontiers colleagues Giorgia, Ondřej, Karim, and Javier. It was always fun to work, travel, and hang out with you and to share the ups and downs of living in Paris. Further, I want to thank the whole ADASP team at Télécom Paris for creating a stimulating and supportive work environment. I also had a great time being a visiting researcher at Audionamix, thanks to the whole team for being so welcoming!

The corona crisis presented some unexpected challenges during the second half of my PhD project. MIP-Frontiers and the ADASP team made it possible to stay connected with peers nevertheless which was extremely helpful. I am also grateful for the invaluable support of my friends and family during this time.

Thanks to my amazing friends from Berlin, Gothenburg, and Paris for many enjoyable moments helping me to disconnect from research from time to time. A special thanks goes to Santi who actually found the ad for this PhD project online and encouraged me to apply. Without you, this journey would have never started!

Thanks to my family for your care, advice, help with moving to Paris, several care packages, visits, and encouragement. I am happy that I can always count on you.

Finally, I want to express my deep gratitude to my fiancée Sara. I would have never accomplished this project without your endless support, patience, and love. Thank you for helping me through the most difficult times and for being the best partner to celebrate moments of happiness.

مشتاقانه در انتظار آینده مشترکمون هستم. دوستت دارم.

[1]This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowsa-Curie grant agreement No. 765068.

Abstract

Audio source separation is the task of estimating the individual signals of several sound sources when only their mixture can be observed. It has several applications in the context of music signals such as re-mixing, up-mixing, or generating karaoke content. Furthermore, it serves as a pre-processing step for music information retrieval tasks such as automatic lyrics transcription. State-of-the-art performance for musical source separation is achieved by deep neural networks which are trained in a supervised way. For training, they require large and diverse datasets comprised of music mixtures for which the target source signals are available in isolation. However, it is difficult and costly to obtain such datasets because music recordings are subject to copyright restrictions and isolated instrument recordings may not always exist.

In this dissertation, we explore the usage of prior knowledge for deep learning based source separation in order to overcome data limitations.

First, we focus on a supervised setting with only a small amount of available training data. It is our goal to investigate to which extent singing voice/accompaniment separation can be improved when the separation is informed by lyrics transcripts. To this end, we propose a general approach to informed source separation that jointly aligns the side information with the audio signal using an attention mechanism. We perform text-informed speech-music separation and joint phoneme alignment to evaluate the approach. Results show that text information improves the separation quality. At the same time, text can be accurately aligned with the speech signal even if it is highly corrupted. In order to adapt the approach to the more challenging task of text-informed *singing voice* separation, we propose DTW-attention. It is a combination of dynamic time warping and attention that encourages monotonic alignments of the lyrics with the audio signal. The result is a novel lyrics alignment method which requires a much smaller amount of training data than state-of-the-art methods while providing competitive performance. Furthermore, we find that exploiting *aligned* phonemes can improve singing voice separation, but precise alignment and accurate transcripts are required. Modifications of the input text result in modifications of the separated voice signal. For our experiments we transcribed the lyrics of the MUSDB corpus and made them publicly available for research purposes.

Finally, we consider a scenario where only mixtures but no isolated source signals are available for training. We propose a novel unsupervised deep learning approach to musical source separation. It exploits information about the sources' fundamental frequencies (F0) which can be estimated from the mixture. The method integrates domain knowledge in the form of differentiable parametric source models into the deep neural network. Experimental evaluation on a vocal ensemble separation task shows that the proposed method outperforms F0-informed learning-free methods based on non-negative matrix factorization and an F0-informed supervised deep learning baseline. Combining data-driven and knowledge-based components, the proposed method is extremely data-

efficient and achieves good separation quality using less than three minutes of training data. It makes powerful deep learning based source separation usable in domains where labeled training data is expensive or non-existent.

Résumé

La séparation de sources audio est la tâche consistant à estimer les signaux individuels de plusieurs sources sonores lorsque seul leur mélange peut être observé. Elle a plusieurs applications dans le contexte des signaux musicaux, comme le remixage, l'*up-mixing* ou la génération de contenu karaoké. En outre, elle sert d'étape de prétraitement pour les tâches de recherche d'informations musicales telles que la transcription automatique de paroles de chansons. Les performances de l'état de l'art en séparation de sources musicales sont obtenues par des réseaux neuronaux profonds entraînés de manière supervisée. Pour leur entraînement, on a besoin de grandes bases de données diversifiées composées de mélanges musicaux pour lesquels les signaux sources cibles sont disponibles de manière isolée. Cependant, il est difficile et coûteux d'obtenir de telles bases de données car les enregistrements musicaux sont soumis aux restrictions de droits d'auteur et les enregistrements d'instruments isolés n'existent pas toujours.

Dans cette thèse, nous explorons l'utilisation d'informations supplémentaires pour la séparation de sources par apprentissage profond, afin de s'affranchir d'une quantité limitée de données.

D'abord, nous considérons un cadre supervisé avec seulement une petite quantité de données d'entraînement disponibles. Notre objectif est d'étudier dans quelle mesure la séparation voix chantée/accompagnement peut être améliorée lorsque la séparation est informée par la transcription des paroles. À cette fin, nous proposons une approche générale de séparation de sources informée qui aligne les informations secondaires avec le signal audio pendant la séparation grâce à un mécanisme d'attention. Nous effectuons une séparation parole-musique informée par le texte conjointement avec un alignement des phonèmes pour évaluer l'approche. Les résultats montrent qu'information textuelle améliore la qualité de la séparation. En même temps, le texte peut être aligné avec précision avec le signal vocal même s'il est fortement perturbé. Afin d'adapter l'approche à la tâche plus difficile de la séparation de la voix chantée informée par le texte, nous proposons la technique de *DTW-attention*. Il s'agit d'une combinaison de *dynamic time warping* (déformation temporelle dynamique) et d'attention qui encourage les alignements monotones des paroles avec le signal audio. Le résultat est une nouvelle méthode d'alignement des paroles qui nécessite une quantité de données d'entraînement beaucoup plus faible que les méthodes de l'état de l'art tout en offrant des performances compétitives. En outre, nous constatons que l'exploitation des phonèmes alignés peut améliorer la séparation de la voix chantée, mais un alignement précis et des transcriptions exactes sont nécessaires. Les modifications du texte d'entrée entraînent des modifications du signal vocal séparé. Pour nos expériences, nous avons retranscrit les paroles du corpus MUSDB et les avons rendues publiques à des fins de recherche.

Enfin, nous considérons un scénario où seuls des mélanges, mais aucun signal source isolé, sont disponibles pour l'apprentissage. Nous proposons une nouvelle approche d'apprentissage profond non supervisé pour la séparation de sources musicales. Elle exploite les informations sur

les fréquences fondamentales (F0) des sources qui peuvent être estimées à partir du mélange. La méthode intègre des connaissances du domaine sous la forme de modèles de sources paramétriques différentiables dans le réseau neuronal profond. L'évaluation expérimentale d'une séparation d'un ensemble vocal montre que la méthode proposée surpasse les méthodes sans apprentissage informées par F0 et basées sur la factorisation de matrices non négatives, ainsi qu'une approche d'apprentissage profond supervisé informée par F0. En combinant des approches guidées par les données avec des approches basées sur la connaissance, la méthode proposée est particulièrement efficace en terme de données et atteint une bonne qualité de séparation en utilisant moins de trois minutes de données d'entraînement. Elle rend la séparation de sources par apprentissage profond exploitable dans les domaines où les données d'entraînement étiquetées sont coûteuses ou inexistantes.

Contents

List of Figures

List of Tables

Notation

a, A	real-valued scalars
$\underline{a}$	complex-valued scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix or sequence of column vectors
$a_{i,j}$	a is the entry in the i-th row and j-th column of a matrix
$\mathbf{a}_n$	The vector $\mathbf{a}$ is the n-th element in a sequence of vectors
$\mathbf{AB}$	Matrix product $\mathbf{AB} = \mathbf{C}$ with $c_{i,j} = \sum_r a_{i,r} b_{r,j}$
$*$	Convolution operation
$\mathbb{R}$	Set of real numbers

Abbreviations

AE	Absolute Error
ASR	Automatic Speech Recognition
BLSTM	Bidirectional Long Short-Term Memory
BRNN	Bidirectional Recurrent Neural Network
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
DTW	Dynamic Time Warping
F0	Fundamental frequency
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
LSF	Line Spectral Frequency
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MFA	Montreal Forced Aligner
MIREX	Music Information Retrieval Evaluation eXchange
MLP	Multi-Layer Perceptron
NMF	Nonnegative Matrix Factorization
PCAS	Percentage of Correctly Aligned Segments
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RPCA	Robust Principle Component Analysis
SAR	Source-to-Artifacts Ratio
SATB	Soprano, Alto, Tenor, Bass
SDR	Source-to-Distortion Ratio
SI-SDR	Scale-Invariant Source-to-Distortion Ratio
SIR	Source-to-Interference Ratio
SNR	Signal-to-Noise Ratio
STFT	Short Time Fourier Transform
VAD	Voice Activity Detector
WER	Word Error Rate

Chapter 1

Introduction

Most music recordings are mixtures of several sound sources such as musical instruments or human voice. The signals of the individual instruments are usually not accessible in isolation. If the instruments are recorded separately, the isolated signals exist but are usually not distributed. This is often the case for popular music productions. For many other genres such as jazz, classical music, or folk, it is common practice that the musicians perform together in the same room and only the mixture of the instrument signals is recorded. In this case, no isolated recordings exist. Furthermore, the individual instrument recordings of many old music productions are lost today.

However, having access to the separate instrument signals is necessary for many use cases of music recordings. For example, one may wish to remove the singing voice from a song to obtain a karaoke version. Some artists want to extract one instrument from a given mixture to remix it with other recordings to build a new music piece. Play-along tracks can be created for musicians by removing their instrument from a song they want to practice. A mixture has usually two channels which is the standard in the consumer audio industry. One may wish to *upmix* a recording for playback on systems with more channels and change the spatial location of an instrument. Moreover, automatic analysis of recordings is facilitated through access to the individual sound sources. Examples comprise the retrieval of the singer identity, the instrumentation or language of a song, and the transcription of the lyrics or the musical score.

Recovering one or more sound sources from their mixture is referred to as *audio source separation*. This dissertation deals with *musical* source separation with a focus on estimating the singing voice from a mixture.

1.1 Motivation and aim

The first successful approaches to musical source separation enabling some of the above mentioned applications were knowledge-driven. The knowledge they exploited can be divided into *model-based knowledge* and *side information*. Model-based knowledge comprises models of source properties and the mixing process. Side information is additional data which contains information about one or more sources. Under this knowledge-driven paradigm the term *informed* audio source separation was coined [115, 201]. A sketch of the general workflow of knowledge-driven source separation is shown in Figure 1.1. Note that not all approaches comprise all of the shown components. The processing steps and possible parameter updates are performed for each mixture to be separated individually. Prior knowledge is essential in one or more stages of the process.

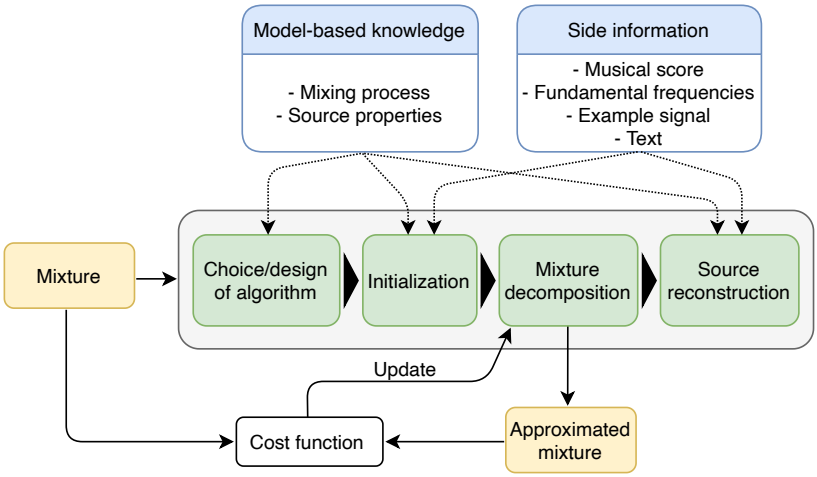


Figure 1.1: General procedure of knowledge-driven audio source separation.

In the last decade, great progress was made regarding the application of deep learning to a wide variety of machine learning tasks. This was partly due to the availability of larger datasets and more computational power. Deep learning has revolutionized fields such as computer vision [99], natural language processing [216], and audio data analysis [60]. Using Deep Neural Networks (DNN) for audio source separation led to immense performance gains [78]. As a consequence, state-of-the-art methods are data-driven today and rely on supervised learning. The general learning procedure of such methods is presented in Figure 1.2. The main challenge of data-driven methods is the need for large sets of high quality labeled data.

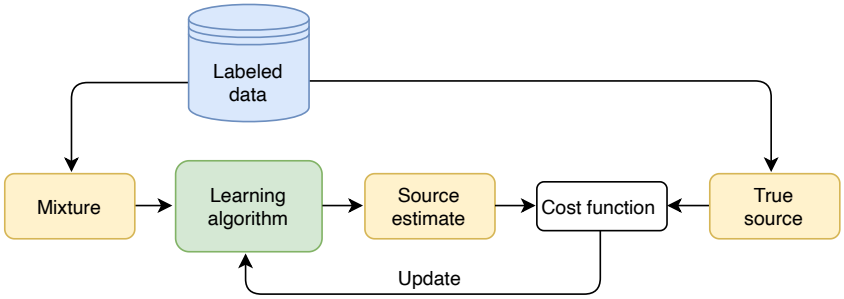


Figure 1.2: Learning procedure of data-driven audio source separation.

One simple way to improve the performance of DNNs would be to use more high quality data for training. In fact, it has been shown that training generic DNN architectures on extremely large datasets can lead to state-of-the-art performance in music source separation [86] and other tasks such as music transcription [67] and version identification [34]. However, access to such large amounts of data is usually limited in the music domain. In general, music recordings are subject to copyright restrictions which impedes their free distribution and usage as training data. This is a challenge that all data-driven approaches in music informatics have in common.

Another challenge is the creation of task specific labels for supervised learning. This is particularly difficult for music source separation because in this context labels or training targets are isolated instrument or voice recordings corresponding to the available mixtures. Such targets cannot be created manually as opposed to labels for other tasks. They have to be obtained from the right holders along with the mixtures. As explained above, isolated instrument recordings are

usually not distributed or may not even exist. Special recording sessions may be arranged in order to record signals in isolation as for example done in [25] with a choir. However, this is not only extremely costly but also leads to unnatural conditions for the musicians of certain music styles.

At the same time, complementary data such as musical scores or lyrics transcripts may be more readily available. This kind of side information was exploited by many knowledge-driven source separation methods but is usually ignored by data-driven methods. Furthermore, while separate instrument signals are extremely difficult to obtain, mixtures can be obtained much more easily. Therefore, it would be useful to develop methods that can learn from only mixture signals or exploit available side information in addition to labeled data.

The aim of this dissertation is to explore the usage of prior knowledge in data-driven approaches to music source separation. To develop informed deep learning based methods, we take inspiration from the knowledge-driven separation procedure. The underlying hypothesis is that this may lead to less dependence on labeled data and more efficient methods.

First, we focus on the usage of side information to complement data for supervised learning. Specifically, it is investigated to which extent the separation of singing voice from the instrumental accompaniment can be supported by lyrics transcripts as side information. Lyrics are widely available and easier to obtain than audio data. Furthermore, they can be transcribed by users without specific musical or technical expertise. Compared to musical scores they have received little attention as side information.

One challenge is that text transcripts do not contain information about the temporal alignment of the words or phonemes with a corresponding audio signal. Therefore, it is also investigated which level of alignment between lyrics and a mixture signal is required to inform the separation. Automatic lyrics alignment methods either do not provide reliable alignments at a fine scale or suffer from very high data requirements [182, 64]. For these reasons, it is an additional goal of this dissertation to lower the data needs of data-driven fine scale alignment approaches. Aligning text with mixtures is more challenging than with solo singing recordings. Therefore, we explore if the alignment and text-informed separation can be performed jointly and if this leads to benefits.

Lastly, a scenario is considered where no separate target source signals are available for training. It is explored how prior knowledge can be exploited in order to learn source separation using only mixture signals. The goal is to make data-driven source separation applicable for the wide range of music genres where instruments are not recorded in isolation. The usage of side information in the form of fundamental frequency (F0) trajectories and model-based knowledge in the form of generative source models is investigated. Finally, we evaluate experimentally how such an informed unsupervised learning approach performs compared to knowledge-based learning-free methods using the same prior knowledge and supervised methods.

1.2 Structure of the dissertation

The document is divided into two parts. Part I contains an overview of relevant concepts and methods as well as a review of previous work on musical source separation. In Part II the proposed approaches are described, put in context of related work, and experimentally evaluated.

Part I: Background

- **Chapter 2 – Technical background:** This chapter provides a technical introduction to audio source separation and a brief review of concepts and methods which are used in Part II.

- **Chapter 3 – Related work:** This chapter provides a review of previous work on audio source separation with a focus on music signals. It includes a discussion of the limitations of recent approaches and the resulting research opportunities.

Part II: Contributions

- **Chapter 4 – Weakly informed audio source separation:** This chapter addresses the problem of how to use non-aligned side information for supervised audio source separation with deep learning. An experimental proof of concept of the proposed approach is provided performing singing voice separation with synthetic side information. Finally, the method is evaluated on text-informed speech-music separation and text-to-audio alignment.
- **Chapter 5 – Text-informed singing voice separation and lyrics alignment:** In this chapter the method of the previous chapter is improved so that it can perform phoneme level lyrics alignment and text-informed singing voice separation. Extensive experimental evaluation is provided and the benefits and limitations of text as side information are discussed.
- **Chapter 6 – Unsupervised audio source separation:** In this chapter it is shown how fundamental frequency information and generative source models can be used to enable deep learning based source separation when only mixtures but no separate source signals are available for training. The proposed method is evaluated on a vocal ensemble separation task and compared to learning-free and supervised methods.
- **Chapter 7 – Conclusion:** This chapter concludes the thesis by summarizing the work, discussing its limitations, and proposing possible directions for future work.

1.3 Contributions and publications

The overall contribution of this dissertation is to explore the integration of prior knowledge into data-driven approaches to audio source separation. We follow the principles of open and reproducible science and make all published articles as well as the code and data produced for this work publicly available under permissive licenses as far as possible. The main contributions are detailed below.

- **Chapter 4:** a novel DNN architecture for supervised informed audio source separation is proposed. It can exploit side information which is not aligned with the mixture because it performs separation and alignment jointly. We show that the alignment can be learned using an attention mechanism as a side outcome of training the DNN with a separation objective. Moreover, two new evaluation metrics for audio source separation are proposed. They complement the standard metrics on frames with a silent target or estimated source. The work described in this chapter led to the following publications:
  - Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, Roland Badeau "Weakly Informed Audio Source Separation". In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics,* 2019.
  - Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, Roland Badeau "Joint Phoneme Alignment and Text-Informed Speech Separation on Highly Corrupted Speech" In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2020.

- **Chapter 5:** the method introduced in Chapter 4 is further improved with a novel *monotonic* attention mechanism. It incorporates the prior knowledge that text and audio sequences follow a left-to-right temporal structure. This enables the model to perform phoneme level lyrics alignment and text-informed singing voice separation. The result is a new lyrics alignment method which works well on mixtures and is competitive with the state-of-the-art while using less data for training. Moreover, new insights into text-informed singing voice separation are provided. The musical source separation corpus MUSDB was extended by lyrics transcripts and other annotations which are made publicly available for research purposes. The work described in this chapter led to the following publication:

  – Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, Roland Badeau
  "Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation."
  IEEE/ACM Transactions on Audio, Speech and Language Processing, 2021.

- **Chapter 6:** a novel unsupervised deep learning approach for audio source separation is proposed. It exploits side information in the form of F0 trajectories and model-based knowledge in the form of parametric source-filter models. A new differentiable procedure to estimate stable time-varying all-pole filters with a DNN is proposed in order to integrate the source-filter models in the training pipeline. The proposed approach outperforms learning-free and supervised baselines on vocal ensemble separation. It is also extremely data efficient because it does not only learn from unlabeled data but also from a small amount of such data. Furthermore, it provides a parameterization of the mixture signal which can be exploited for downstream tasks. The work described in this chapter led to the following paper:

  – Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, Roland Badeau
  "Unsupervised audio source separation using differentiable parametric source models."
  Currently under review.

Part I

Background

Chapter 2

Technical Background

Summary

In this chapter we formally introduce the task of audio source separation. We also summarize the key concepts which are the foundation of the domain knowledge and signal information which will be integrated into data-driven source separation in this thesis. Finally, a basic introduction to deep neural networks is given.

Contents

2.1 Introduction to audio source separation

Let us assume we observe an audio signal $x(t)$ which is a mixture of J sound source signals. The goal of audio source separation is to estimate the individual signal of one or more of those sources. For a comprehensive overview of audio source separation we refer the reader to [203]. In the following, a brief introduction is given based on this reference.

Several scenarios as to how the sound sources are recorded and how the mixture signal is created can be differentiated. In general, the mixing process can be described in two steps. First, each source signal $s_j(t)$ is transformed into a source spatial image signal $v_j(t)$ with a filter $a_j(t)$:

$$v_j(t) = a_j(t) * s_j(t). \tag{2.1}$$

The impulse response $a_j(t)$ may describe the acoustic effect of the room in which the source is recorded or artificial sound effects. In the notation above we consider all signals to have one channel. However, a mixture signal with I channels may be obtained by using $I \times J$ different filters which may represent several microphone positions or artificial spatialization effects. As a second step, the mixture is created by the sum

$$x(t) = \sum_{j=1}^{J} v_j(t). \tag{2.2}$$

The sources can be recorded simultaneously with the same microphone(s) or separately and then mixed artificially. Both cases may be described by the sum above. The mixture $x(t)$ might be processed further. For example, in music production it is common to apply dynamic range compression or a reverberation filter. Such effects are not considered in most source separation approaches including those proposed in this thesis.

The mixture is said to be overdetermined (or determined) if there are no spatially diffuse sources and the number of sources is smaller than (or equal to) the number of channels. Otherwise, it is underdetermined.

In the context of music and speech signals, we usually face underdetermined mixtures. In several applications of musical source separation it is the goal to obtain the source spatial image signals $v_j(t)$ which include reverberation and other audio effects. However, the clean source signals $s_j(t)$ are required to change the source location. In speech processing one is usually interested in the clean source signals, for example to use them as input to automatic speech recognition systems. For this more comprehensive enhancement problem also dereverberation and echo cancellation might be applied. In the multi-channel case, if the sources are recorded with spatially distributed microphones, one can exploit the spatial information for the separation. However, in this dissertation we focus on the single-channel case where this is not possible.

2.1.1 Time-frequency masking

A common approach to single-channel audio source separation is to apply a time-varying filter in the time-frequency domain. To this end, complex-valued time-frequency coefficients $\underline{x}(f, n)$ are computed by a Short Time Fourier Transform (STFT) of the mixture. The frequency bins are indexed by f and the time frames are indexed by n. Complex variables are denoted with an underline. Then, a typically (but not necessarily) real-valued filter $w_j(f, n)$ is applied to obtain an estimate of the target source coefficient $\hat{\underline{v}}_j(f, n)$:

$$\hat{\underline{v}}_j(f, n) = w_j(f, n)\underline{x}(f, n). \tag{2.3}$$

The filter $w(f, n)$ is called a *binary mask*, if it takes only the values zero and one, and a *soft mask* if it takes values in the interval $[0, 1]$.

There are many approaches to obtain such masks with different properties. One widely applied way is to estimate the magnitudes of the sources in the time-frequency domain and to construct the filter as

$$w_j(f,n) = \frac{\hat{p}_j(f,n)^\alpha}{\sum_{j'} \hat{p}_{j'}(f,n)^\alpha} \tag{2.4}$$

where $p_j(f,n) = |\underline{v}_j(f,n)|$ and the hat indicates an estimate [114]. When $\alpha = 2$, the mask $w_j(f,n)$ is called a single-channel Wiener filter which is the minimum mean square error estimator under the assumption that the source signals are uncorrelated locally stationary Gaussian processes with zero mean [203]. However, this assumption does not hold for all audio sources and $\alpha = 1$ is preferred for music signals. This is theoretically justified under the assumption that the signals are locally stationary stable harmonizable processes [114]. Estimates of all sources are required to build such a mask. When only an estimate of one target source magnitude is available it is simply combined with the mixture phase to obtain the corresponding complex-valued source estimate.

2.1.2 Evaluation

Different objective evaluation metrics for audio source separation exist. The most widely used metrics have been proposed by Vincent et al. [202]. They decompose a source estimate $\hat{s}_j$ as $\hat{s}_j = s_{target} + e_{interf} + e_{noise} + e_{artif}$, with the target source term $s_{target} = f(s_j)$ being a function of the true source (or source image). The function f may be a time-variant or constant scaling term or allowed distortion filter. The terms e_{interf}, e_{noise}, and e_{artif} describe interferences from other sources, noise, and algorithmic artifacts, respectively. The proposed evaluation metrics are Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), Source-to-Noise Ratio, and Source-to-Artifacts Ratio (SAR) which are energy ratios of the target source and the error terms. Whereas the SIR, Source-to-Noise Ratio, and SAR focus on specific error types, the SDR computes the general error and is defined as

$$\text{SDR} = 10\log_{10}\frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} = 10\log_{10}\frac{\|f(s_j)\|^2}{\|f(s_j) - \hat{s}_j\|^2}. \tag{2.5}$$

It has recently been argued that f as a filter is too permissive and that it should only scale the true source (image) to make the metric invariant to the scale of the estimate [106]. For this case the metric has been called explicitly scale-invariant SDR (SI-SDR) [106]. For music signals it became common practice to compute the metrics on one second long non-overlapping frames to take the strongly time-varying nature of such signals into account [185].

Moreover, there are some metrics designed to measure the quality of speech signals. The Perceptual Evaluation of Speech Quality (PESQ) [156] evaluates the overall error with a psychoacoustic model and the Short-Time Objective Intelligibility score (STOI) [189] measures intelligibility and was shown to be correlated with intelligibility assessed by humans in listening tests [189, 73].

2.2 Key concepts

2.2.1 Human voice production

Since we are concerned with singing voice and speech signals in this thesis, it is useful to understand the basics of the human voice production mechanism and the resulting properties of voice signals. A comprehensive treatment of the topic can be found in [48] and [51].

A *voiced* sound such as a vowel is produced as follows. Air flows from the lungs via the trachea through the tensed vocal chords which vibrate as a result. The vocal chords open and close rapidly due to their vibration and transform the airflow into quasi-periodic pulses which form sound waves. The pulses then propagate through the vocal tract which consists of the throat cavity, oral cavity, and nasal cavity. Finally, sound waves are radiated from the mouth and nose. The position of the jaw, tongue, velum, lips, and mouth (articulators) determines the shape of the vocal tract and thus its resonance frequencies also called *formants*. The formant frequencies shape the spectral envelop of the produced sound which thus varies over time with the vocal tract shape. When the vocal chords are not tensed, the air flow enters the vocal tract unaltered. There, it may hit a partial constriction, which results in turbulent air flow producing an *unvoiced* sound (e.g. the first sound of the word "sing"). Otherwise, it may hit a total constriction, pressure builds up and a transient sound is produced when it is released (e.g. the first sound of the word "put"). The properties of such sounds are also determined by the articulators [149].

Although this description is simplified, we can conclude that a voice signal is a sequence of time-varying sounds which may be periodic, noise-like, or transient [149]. The primary goal of a speech signal is to encode an implicit message that should be transmitted. In contrast, singing voice signals also intend to convey artistic expression and melodies so that intelligibility is not the main priority.

Source-filter model of human voice production

The source-filter model is based on the observation that a voice sound wave is essentially a signal from a sound source – vibrating vocal chords or turbulent air flow – which is modified by a filter, the vocal tract. Expressed in the language of signal processing, a voice signal $s(t)$ can thus be modeled as the response of a time-varying filter $h(t)$ to an excitation signal $e(t)$ which is periodic for voiced sounds and white noise for unvoiced sounds [48]:

$$s(t) = e(t) * h(t). \tag{2.6}$$

Linear predictive coding models speech signals as autoregressive processes which suggests that the vocal tract filter is an all-pole filter [120]. The transfer function of the filter can be written in the z-domain as

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 - \sum_{k=1}^{K} a_k z^{-k}} \tag{2.7}$$

where G is a gain parameter, a_k are the filter coefficients, and K is the filter order.

The source-filter model greatly simplifies the human voice production process. Nevertheless, it is the foundation of many speech [149, 154] and singing voice synthesis methods [10, 133, 29]. It may also be used to model musical instruments [40, 68].

2.2.2 Phonemes

The concept of phonemes originated in linguistics. Although the exact definition is a subject of debate in this field, a commonly applied definition is the following. A phoneme is a *distinctive* unit of a word. This means that the replacement of one phoneme by another leads to a new word with a new meaning. For example, we can take the word 'bin' and replace the first phoneme 'B' by a 'P' and obtain the new word 'pin' [32]. It results from this definition that phonemes are the smallest

distinctive sound units of a language. In the context of audio processing, we are interested in the distinct spectral properties of phonemes.

There are different notation systems for phonemes including the International Phonetic Alphabet (IPA) [4] and the ARPAbet representation proposed by the Advanced Research Projects Agency (ARPA) as a more computer-friendly alternative [95]. In Table 2.1, a condensed list of 39 phonemes of American English is shown in 2-characters ARPAbet notation along with an example word and their phonetic class. Since each phoneme has distinct spectral characteristics, a phonetic transcription provides a link between written words and the corresponding speech signal. To be precise, a phoneme contains the information whether the corresponding sound is predominantly periodic (voiced phoneme) or noise-like (unvoiced phoneme), and information about the overall spectral shape since each phoneme is produced by a distinct vocal tract configuration. It does not contain information about the fundamental frequency of voiced sounds (cf. Section 2.2.3).

It is important to note that phonetic transcriptions derived from orthographic transcripts follow an idealized pronunciation rule. Nevertheless, there are several ways to pronounce a word and the actual acoustic realization heavily depends on the speaking person. Moreover, the exact pronunciation of a phoneme may differ depending on the preceding and following phoneme, an effect called co-articulation [149].

In this thesis, we will use phonetic transcripts as a source of information about the expected spectral properties of a corresponding voice signal which should be separated from other sound sources.

Phoneme	Example	Class
IY	b**ee**t	vowels
IH	b**i**t	
EH	b**e**t	
AE	b**a**t	
AA	B**o**b	
AH	b**u**t	
AO	b**ough**t	
OW	b**oa**t	
UH	b**oo**k	
UW	b**oo**t	
ER	b**ir**d	
EY	b**ai**t	diphthongs
AW	d**ow**n	
AY	b**uy**	
OY	b**oy**	
Y	**y**ou	semivowels
W	**w**it	
R	**r**ent	
L	**l**et	

Phoneme	Example	Class
M	**m**et	nasals
N	**n**et	
NG	sing	
P	**p**at	unvoiced stops
T	**t**en	
K	**k**it	
B	**b**et	voiced stops
D	**d**ebt	
G	**g**et	
HH	**h**at	whisper
F	**f**at	unvoiced fricatives
TH	**th**ing	
S	**s**at	
SH	**sh**ut	
V	**v**at	voiced fricatives
DH	**th**at	
Z	**z**oo	
ZH	a**z**ure	
CH	**ch**urch	affricates
JH	**j**udge	

Table 2.1: Set of 39 phonemes for American English in 2-character ARPAbet notation. The example words are found in [149].

2.2.3 Fundamental frequency

Certain sounds are composed of several sinusoidal components with different frequencies. These sinusoids are called *partials*. The frequency of the lowest partial is the *fundamental frequency*. The

other partials are called *harmonics* if their frequencies are integer multiples of the fundamental frequency. The fundamental frequency is also referred to as F0 because it can be seen as the zeroth harmonic.

Voiced sounds in speech and singing voice as well as sounds from musical instruments consist usually of F0 and a number of harmonics which are not always perfectly but roughly integer multiples of F0. The term *pitch* refers to the perceived tone when listening to a sound. It is thus a subjective descriptor. In the context of music signals, the pitch is usually determined by the fundamental frequency [134].

2.2.4 Line Spectral Frequencies

Line Spectral Frequencies (LSF) were introduced in [83] as an alternative representation of linear prediction coefficients and have been widely used in speech coding. They provide a convenient and to some degree interpretable parameterization of stable minimum-phase all-pole filters. They also allow the formulation of constraints to control the filter response and can be interpolated [21]. They will be used in Chapter 6 to parameterize vocal tract filters. In the following, a brief introduction of LSFs is given. Comprehensive overviews can be found in [21, 180, 88].

The polynomial $A(z) = 1 - \sum_{k=1}^{K} a_k z^{-k}$, which is found in the denominator of the vocal tract filter in (2.7), can be decomposed into the polynomials $P(z)$ and $Q(z)$ which are symmetric and antisymmetric, respectively, and have the order $K + 1$:

$$A(z) = \frac{P(z) + Q(z)}{2}. \tag{2.8}$$

It can be shown that if the roots of $P(z)$ and $Q(z)$ alternate on the unit circle, the corresponding filter $\frac{1}{A(z)}$ is stable and minimum-phase [180]. The unit circle in the z-plane is described by $z = e^{-j\omega}$ where ω is the phase angle in radiants. Hence, ω describes the location of the roots. If K is even, $P(z)$ has a root at $z = -1$ and $Q(z)$ has a root at $z = +1$. The remaining roots occur in complex conjugate pairs. Therefore, it is sufficient to consider only the roots on the upper semicircle. The angles ω_k defining the locations of these complex roots are called LSFs. Two to three LSFs tend to be close together when a filter pole is close to the unit circle in their proximity which corresponds to a peak in the frequency response. Hence LSFs have a frequency domain interpretation. If K is even, $P(z)$ and $Q(z)$ have $K/2$ complex roots on the upper unit semicircle each, for which the following relation holds:

$$0 < \omega_k < \omega_{k+1} < \pi. \tag{2.9}$$

When k is odd, ω_k defines a root of $P(z)$; when k is even, it defines a root of $Q(z)$ for $k \in \{1, ..., K\}$. If K is odd, $Q(z)$ has two real roots (at $z = +1$ and $z = -1$) and $(K - 1)/2$ pairs of complex conjugate zeros. $P(z)$ has $(K + 1)/2$ pairs of complex conjugate zeros in this case and (2.9) still holds. To sum up, a stable minimum-phase filter $\frac{1}{A(z)}$ of order K is defined by K LSFs fulfilling the relation in (2.9).

LSFs can be converted to the filter coefficients a_k using Algorithm 1 [88, 124][1].

[1]The formulation of Algorithm 1 which is presented here was presented in [124]. Some equations in the main body of [124] contain errors but the Matlab code in the Appendix is correct. A less general formulation is found in [21, Ch. 8]. The conversion was formally introduced in [88].

Algorithm 1 Compute filter coefficients a_k from LSFs ω_k [88, 124]

Input: $(\omega_k)_{k=1:K}$
  Define: $x_k = \cos(\omega_k)$
  Initialize: $p'_{-1} = q'_{-1} = 0; \quad p'_0 = q'_0 = 1$
  Initialize: $p'_1 = -2x_1; \quad q'_1 = -2x_2$
  for $k = 2$ **to** $K/2$ **do**
    $p'_k = -2p'_{k-1}x_{2k-1} + 2p'_{k-2}$
    $q'_k = -2q'_{k-1}x_{2k} + 2q'_{k-2}$
    for $i = (k-1)$ **to** 1 **do**
      $p'_i = p'_i - 2p'_{i-1}x_{2k-1} + p'_{i-2}$
      $q'_i = q'_i - 2q'_{i-1}x_{2k} + q'_{i-2}$
    end for
  end for
  for $k = 1$ **to** $K/2$ **do**
    $p_k = p'_k + p'_{k-1}$
    $q_k = q'_k - q'_{k-1}$
  end for
  for $k = 1$ **to** $K/2$ **do**
    $a_k = (p_k + q_k)/2$
    $a_{(K/2+k)} = (p_{(K/2-k+1)} - q_{(K/2-k+1)})/2$
  end for
Output: $(a_k)_{k=1:K}$

2.3 Deep neural networks

Deep neural networks (DNN) are powerful models for supervised machine learning. In the last decade, they have set new standards in multiple domains such as computer vision [99], natural language processing [216], and also audio processing [60].

Nevertheless, it is important to note that they have some drawbacks. Perhaps the biggest one is that they require large amounts of labeled data for supervised learning. This is a challenge in domains such as music where copyright issues arise and manual annotations in large amounts are extremely costly to produce. This aspect motivates the work presented in this dissertation. Furthermore, their internal representations are difficult to interpret and the outputs are hard to explain [214]. In fact, their vulnerability to adversarial attacks [3] suggests that they may rely on spurious patterns in the training data which are not relevant for the task at hand. Like most machine learning techniques they do not learn causal relationships [164]. Lastly, the development, training, and deployment of DNNs require massive computational resources which leads to a large carbon footprint [187, 170].

The task of a DNN is to approximate a function $f^*(\mathbf{X}) = \mathbf{Y}$ mapping from some input data $\mathbf{X}$ to some target data $\mathbf{Y}$. A DNN defines a function $f_\theta(\mathbf{X}) = \hat{\mathbf{Y}}$ with parameters θ which are optimized to obtain the best estimate $\hat{\mathbf{Y}}$. The function $f_\theta = f^{(L)} \circ ... \circ f^{(2)} \circ f^{(1)}$ is composed of L different functions which are also referred to as layers. $f^{(1)}$ is called *input layer*, $f^{(L)}$ is the *output layer*, and the other functions are called *hidden layers*. The output of one layer serves as input to the subsequent layer. In the following, we introduce the most important layer types and deep learning concepts which are used in this dissertation. More comprehensive overviews are available in [57] regarding deep learning in general, and in [148] regarding its application to audio data.

2.3.1 Fully connected layer

A fully connected layer is the simplest building block of a DNN. It computes the output $\mathbf{y} \in \mathbb{R}^{D_{out}}$ from an input $\mathbf{x} \in \mathbb{R}^{D_{in}}$ as

$$\mathbf{y} = g(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.10}$$

where $\mathbf{W} \in \mathbb{R}^{D_{out} \times D_{in}}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^{D_{out}}$ is a bias vector, and g is a non-linear activation function. Such a layer may also be called *dense layer*. Multiple dense layers composed together are referred to as Multi-Layer Perceptron (MLP) [57].

2.3.2 Recurrent neural networks

Recurrent Neural Networks (RNN) [158] are designed to process sequential data. Hence, the input and output are sequences of N vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]$, respectively. A simple RNN is defined by the function

$$\mathbf{y}_n = g(\mathbf{W}\mathbf{y}_{(n-1)} + \mathbf{U}\mathbf{x}_n + \mathbf{b}) \tag{2.11}$$

where $\mathbf{U} \in \mathbb{R}^{D_{out} \times D_{in}}$ is a weight matrix. The same parameters are used across the entire input sequence and (2.11) has a causal structure. The vector $\mathbf{y}_n$ is also referred to as *hidden state* because, besides being an output of the recurrent layer at step n, it is also an input to the computation at step $n + 1$.

In many cases, it may be beneficial to make the output depend on the whole input sequence and not only on past observations. This can be done using a *bidirectional* RNN [169] which consists of one RNN as in (2.11) and another RNN processing the input sequence in reverse order. Their outputs are then concatenated before being processed by the next layer.

In practice, RNNs are not applied in the simple form of (2.11) because they suffer from vanishing and exploding gradients due to the recurrent dependencies [74, 8]. Gated RNNs such as Long Short-Term Memory (LSTM) cells [75, 55] and the Gated Recurrent Unit (GRU) [19] solve this problem by introducing mechanisms to reset the internal memory of past (or future) observations. They are also used in this thesis.

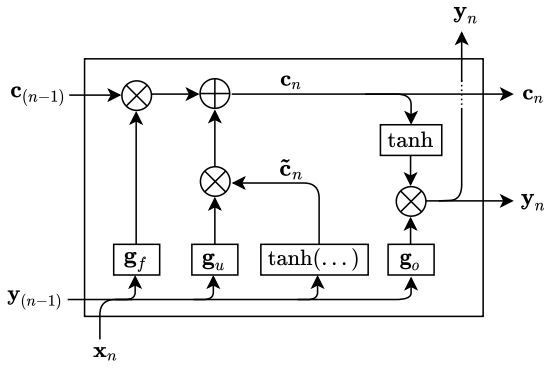


Figure 2.1: Long Short-Term Memory (LSTM) cell.

An LSTM layer is defined by the following set of equations:

$$\tilde{\mathbf{c}}_\mathbf{n} = \tanh(\mathbf{W}_c \mathbf{y}_{(n-1)} + \mathbf{U}_c \mathbf{x}_n + \mathbf{b}_c), \tag{2.12a}$$
$$\mathbf{g}_u = \sigma(\mathbf{W}_u \mathbf{y}_{(n-1)} + \mathbf{U}_u \mathbf{x}_n + \mathbf{b}_u), \tag{2.12b}$$
$$\mathbf{g}_f = \sigma(\mathbf{W}_f \mathbf{y}_{(n-1)} + \mathbf{U}_f \mathbf{x}_n + \mathbf{b}_f), \tag{2.12c}$$
$$\mathbf{g}_o = \sigma(\mathbf{W}_o \mathbf{y}_{(n-1)} + \mathbf{U}_g \mathbf{x}_n + \mathbf{b}_o), \tag{2.12d}$$
$$\mathbf{c}_n = \mathbf{g}_u \circ \tilde{\mathbf{c}}_n + \mathbf{g}_f \circ \mathbf{c}_{(n-1)}, \tag{2.12e}$$
$$\mathbf{y}_n = \mathbf{g}_o \circ \tanh(\mathbf{c}_n), \tag{2.12f}$$

where $\circ$ denotes the element-wise product and σ is the Sigmoid activation function (cf. Section 2.3.4). $\mathbf{W}_c, \mathbf{W}_u, \mathbf{W}_f, \mathbf{W}_o \in \mathbb{R}^{D_{out} \times D_{out}}$ and $\mathbf{U}_c, \mathbf{U}_u, \mathbf{U}_f, \mathbf{U}_o \in \mathbb{R}^{D_{out} \times D_{in}}$ are weight matrices, $\mathbf{b}_c, \mathbf{b}_u, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^{D_{out}}$ are bias vectors, $\mathbf{x}_n \in \mathbb{R}^{D_{in}}$ is the input vector and $\mathbf{y}_n \in \mathbb{R}^{D_{out}}$ is the output. $\mathbf{c}_n$ represents the internal memory and $\mathbf{g}_u, \mathbf{g}_f, \mathbf{g}_o \in]0, 1[^{D_{out}}$ are called update gate, forget gate, and output gate, respectively. A diagram of an LSTM cell is shown in Figure 2.1. A GRU is a simplified LSTM cell which does not have an output gate and is computationally more efficient [19]. The performance of LSTMs and GRUs was found to be on par for sequence modeling tasks [22].

2.3.3 Convolutional neural networks

Convolutional Neural Networks (CNN) [107] are designed to process data with a grid-like structure such as regularly sampled time series (1-D grid), images (2-D grid), or videos (3-D grid) [57]. The input of a convolutional layer is convolved with kernels whose entries are learnable parameters. This means that a kernel is slid across the input and at each position its values are multiplied with the overlapping input values. The sum of the results is the corresponding output value. This way local patterns can be recognized independently of their location in the input representation.

While the approaches proposed in this thesis are based on RNNs, CNNs are also widely used in audio processing and audio source separation. Waveforms are processed with 1-D convolutions and time-frequency representations are processed with the 2-D version. A comprehensive introduction to CNNs can be found in [37] and [57].

2.3.4 Activation functions

DNNs are employed to approximate highly non-linear mappings. Therefore, non-linear functions g are applied element-wise to the outputs of the affine transformations defined by the layers introduced above. These functions are called activation functions. Popular activation functions are the hyperbolic tangent tanh(), the Sigmoid or logistic function $\sigma()$ defined as

$$\sigma(x) = \frac{1}{1 - e^{-x}}, \tag{2.13}$$

and the Rectified Linear Unit (ReLU) [119] defined as

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.14}$$

It is difficult to say with certainty which activation function is the best choice for a certain DNN or data. However, it should be noted that they have different derivatives and thus behave differently

in gradient-based optimization. Moreover, they determine the interval in which the output values lie. This is especially relevant for the output layer of a DNN because its output values must be able to cover the value range of the target data.

2.3.5 Attention mechanism

The attention mechanism has been introduced by Bahdanau et al. [6] for neural machine translation with recurrent encoder-decoder models. It enables sequence-to-sequence models to evaluate the relevance of each element in one sequence with respect to the elements of another sequence by means of a learned scoring scheme. In the context of machine translation, it provides an efficient way to deal with the fact that different languages have different sentence structures. Consider for example the English-French sentence pair "The agreement on the European Economic Area was signed in 1992" and "L'accord sur la zone économique européenne a été signé en 1992". While the first few words can be translated word by word from left to right, the phrase "European Economic Area" is reversed in French.

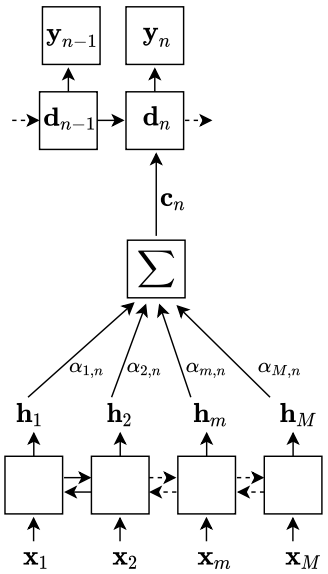


Figure 2.2: Recurrent encoder-decoder model with attention mechanism.

A basic recurrent encoder-decoder model with attention is shown in Figure 2.2. The output $\mathbf{y}_n$ (e.g. a French word) is computed based on the hidden state of the decoder $\mathbf{d}_n = f(\mathbf{d}_{n-1}, \mathbf{y}_{n-1}, \mathbf{c}_n)$ which is a function of the previous hidden state $\mathbf{d}_{n-1}$, the previous output $\mathbf{y}_{n-1}$, and a context vector $\mathbf{c}_n$. The context vector summarizes the hidden representation $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M]$ of the input sequence (e.g. an English sentence). It is computed as

$$\mathbf{c}_n = \sum_{m=1}^{M} \alpha_{m,n} \mathbf{h}_m. \tag{2.15}$$

The attention weights $\alpha_{m,n}$ are computed as

$$\alpha_{m,n} = \frac{e^{s_{m,n}}}{\sum_{k=1}^{M} e^{s_{k,n}}} \tag{2.16}$$

where $s_{m,n} = a(\mathbf{d}_{n-1}, \mathbf{h}_m)$ is a score computed with a scoring mechanism a. In [6] a fully connected layer was used as scoring mechanism but alternatives have been proposed in [118]. Equation (2.16) is a softmax operation which ensures that all M attention weights sum up to one. The attention weights constitute a soft alignment between the input and output sequence.

Attention became a widely adopted concept and has shown to be useful in a wide range of tasks and DNN architectures [15]. Beyond that, a neural network architecture called Transformer was introduced which processes data only with attention mechanisms instead of using RNN or CNN layers [199].

2.3.6 Training deep neural networks

Training a DNN means to adjust its parameters θ so that it returns the desired estimate $\hat{\mathbf{Y}}$ for a given input. A loss function $\mathcal{L}(\theta)$ is defined which measures the difference between the network output $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$ for given θ. The loss is minimized with a variant of gradient descent using a set of training examples by updating the parameters as follows:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta) \tag{2.17}$$

where α is the learning rate. In practice, the gradient is not computed for the entire training set due to high computational costs. Instead, batches of examples are sampled from the training set and the updates are done using gradients computed on the batches which is called stochastic gradient descent [57]. In this thesis, we use the Adam optimization algorithm [90], which is an improved version of stochastic gradient descent and the most widely used algorithm in applied deep learning today. It keeps an average of past gradients to determine the direction of the next update – a concept known as momentum – and adapts the learning rate for each model parameter. These improvements accelerate convergence and lower the dependence on the learning rate which is a hyperparameter that is usually difficult to set [57].

A validation set is usually used to tune hyperparameters and to decide when training is stopped to avoid overfitting on the training data. Training DNNs often involves heuristics and finding an optimal solution is not guaranteed in most applications. However, DNNs solve many tasks exceedingly well despite a lack of theoretical guarantees [57]. In this thesis, deep learning is applied as a method to solve tasks in the audio domain. Hence, the work is more of experimental than of theoretical nature.

Chapter 3

Related Work

Summary

In this chapter we give a broad overview of work on audio source separation with a focus on music signals. We review knowledge-driven and data-driven approaches and discuss their strengths and limitations to motivate the work in this dissertation. Publications which are directly related to a chapter of this thesis are reviewed in the respective chapter.

Contents

In general, source separation can be seen as a linear system identification and inversion problem. However, in the audio domain this perspective is too limited because the sources may be spatially diffuse, the observed mixtures are usually underdetermined, and separation up to an unknown permutation is not sufficient [201]. Two paradigms for audio source separation have emerged in the last decades which are *knowledge-driven* and *data-driven* [152]. They are reviewed in Sections 3.1 and 3.2 respectively, followed by a discussion in Section 3.3.

3.1 Knowledge-driven audio source separation

A strictly blind source separation scenario where absolutely nothing is known about the mixing process or the sources is not applicable in the audio domain. Therefore, information or assumptions are always integrated to some degree in the separation methods for audio signals [203]. Such information may have different levels of detail ranging from rather general *model-based knowledge*, concerning for example source properties or the mixing process, to *side information* being specific to an observed mixture or source signal. In Figure 1.1 (Page 2), an overview of the general workflow of knowledge-driven source separation was presented. Reviews of such methods can be found in [115] and [201].

3.1.1 Model-based knowledge

Some general assumptions are usually made about the sound sources or the mixing process. For example, the Wiener filter makes assumptions about the sources' probability distribution as explained in Section 2.1.1. However, exploiting more specific knowledge further improves or actually enables the separation of audio sources.

Some efforts have been made to exploit information about the spatial properties of the recording setup [201]. Knowledge about the source location can be used in combination with beamforming if the distance between the source and the microphone is known. If also the distance between microphones is known, one can additionally account for the spatial width of the source with a full-rank spatial covariance matrix [38]. Approaches to explicitly model the reverberation were proposed using knowledge about the reverberation time or even room impulse response measurements [201]. The spatial information improves the separation quality, but such approaches are limited to scenarios where detailed information about the recording setup is available. This may be applicable in telephone conferencing systems but excludes most musical mixtures.

A popular approach to model music and speech sources in the time-frequency domain is to describe them as a linear combination of spectral basis vectors which have a varying activity over time. Nonnegative Matrix Factorisation (NMF) [108] approximates magnitude or power spectrograms as a matrix product of two low-rank matrices containing spectral templates and their activations, respectively [203]. NMF and its probabilistic counterparts [49, 135] have been used extensively for audio source separation and denoising. For these tasks they exploit knowledge about spectral properties of the sources.

For example, the fact that transient sounds result in vertical lines and tonal components in horizontal lines in otherwise sparse time-frequency representations was included in various separation approaches via sparsity constraints [98] and temporal continuity constraints [205, 49]. This knowledge was also exploited in a non-parametric approach to separating percussive from harmonic sounds [50]: Median filters are applied along the time and frequency dimension of a magnitude spectrogram of a mixture to separate vertical and horizontal lines. Also the knowledge that only one phoneme is active at once in speech or a few tones in music was used through sparsity constraints on the activations of spectral templates [201, 205, 7].

However, further information is required in order to achieve separation with methods based on spectral basis vectors. When the basis spectra and its activations are estimated based on the observed mixture to be separated, the spectral templates need to be assigned to the sound sources. Unsupervised clustering and supervised classification algorithms were proposed to assign spectral templates to sources [200]. They integrate knowledge through the manual selection of features for clustering and isolated source signals as training examples for classification. These approaches had limited success, though [205].

An alternative is to learn the basis spectra beforehand from appropriate clean source recordings [7, 135, 92, 102]. This step itself is in fact data-driven. Nevertheless, restricting the templates to the learned ones during the separation is knowledge-driven: it requires knowledge about the source types in the mixture and the templates carry information about the spectral source properties. It was also proposed to cluster the spectral components jointly with the mixture decomposition through explicit modeling of the mixing process [140].

The spectrum of most musical sources can be assumed to have a harmonic structure. This was accounted for by imposing a harmonic structure on spectral templates [36, 70, 40]. In [70], the

amplitude of the harmonics is controlled through specific parameters whereas in [40] a source-filter model is formulated for the target source within an NMF framework. However, singing voice also contains unvoiced sounds without harmonic structure. This was taken into account in [84] by learning templates for noise-like sounds and including them in the separation for frames where no voice pitch was detected. Finally, a general NMF framework for audio source separation was developed [143] which allows to flexibly integrate various kinds of prior knowledge and generalizes most approaches to use model-based knowledge.

Robust Principle Component Analysis (RPCA) also showed promising results for singing voice separation [77, 215]. Like NMF, it assumes that the accompaniment has a low-rank structure. Unlike NMF, it assumes that the voice signal is sparse and does not have a low rank. In [215], improvements were achieved by imposing harmonicity priors on the accompaniment model.

A non-parametric method to separate singing voice from the instrumental accompaniment exploits the observation that the accompaniment often consists of patterns that are repeated many times while the vocals are not repeated as much [153].

To conclude, the use of model-based knowledge has enabled and improved audio source separation in various ways. The more specific the exploited information is, the higher is the potential for increased separation quality as long as the underlying assumptions are valid. However, when assumptions do not hold the separation quality is negatively impacted.

3.1.2 Side information

Side information is an additional input to the separation system and contains specific information about one or more source signals comprised in the mixture to be separated. Examples of side information are musical scores, F0 trajectories, voice activity information, and example signals.

Musical scores contain information about the number and types of instruments, the required spectral templates to model the sources and their activations. This is valuable information for NMF based source separation [46]. Scores were used to initialize the spectral templates, enforcing harmonic structure corresponding to the played notes, and the activation matrix which leads to improved decompositions [45, 70]. Moreover, they contain the information required to assign templates and their activations to the sources for the actual separation so that no pre-trained bases, clustering, or classification is required. However, in order to exploit scores in such a way it is required that they are aligned with the mixture. In [175] it is proposed to use information from non aligned scores by modelling simultaneously occurring notes as common factors between score and audio in a tensor factorization framework.

F0 trajectories were used in a similar way to guide source separation with NMF [206, 40] and RPCA [82]. Usually, F0 is estimated from the mixture and, hence, no alignment problem arises.

Improvements on singing voice separation with RPCA were achieved by exploiting vocal activity information which can also be obtained from the mixture [144]. Specifically, it was observed that the optimal hyperparameter setting for the RPCA optimization algorithm is different for segments with and without vocals. Choosing an appropriate value based on vocal activity information makes the approach more adaptive and increases performance.

Another type of side information which was used for source separation is an example audio signal which is related to the target source. Such audio signals can be isolated sources from a cover version of the mixture to be separated [54] or a signal recorded by the user imitating the target source [176, 69]. Moreover, speech signals were synthesized from text transcripts and their

similarity to the target source was exploited for speech music separation [105]. For the case when isolated sources are available at an encoding stage and should be estimated from their mixture at a decoding stage, similarities and synergies between informed source separation and source coding have been established [141].

In general, signal information can greatly improve the separation. However, it is required that the side information accurately describes the corresponding audio signals and is temporally aligned. Errors in the side information or its alignment lead to decreased separation quality.

3.2 Data-driven audio source separation

Data-driven methods learn spectral templates or the separation task on data and can thus avoid making strong assumptions which are a weakness of knowledge-driven methods. As mentioned above, spectral bases were learned from clean source recordings for separation with NMF and similar methods which are mainly knowledge-driven [142, 177]. With the availability of more computation power and data, deep learning based separation approaches emerged. They provide state-of-the-art performance for music and speech source separation and denoising in many scenarios today [152, 207].

A sketch of the typical supervised learning procedure was shown in Figure 1.2 (Page 2). DNNs are usually trained to map from a representation of the mixture signal to a representation of one or more sources or spectral masks. This does not require much prior knowledge apart from crafting the input features. However, the approach requires large datasets of mixture signals for which the corresponding source signals are available in isolation. The separation performance strongly depends on the quality and quantity of the training data and on how representative the data are for unseen test mixtures. In the speech domain, hundreds of hours of audio data are publicly available [24]. For musical source separation public datasets are much smaller due to copyright restrictions and the fact that isolated instrument recordings are usually not distributed or do not exist.

For example, the largest public dataset for music source separation is MUSDB [151]. It comprises about 10 hours of mainly western music such as rock, pop, and hip hop. Next to the mixtures, it contains isolated recordings of the vocals, bass, drums, and a mixture of all remaining instruments. Data for other music styles or instruments are scarcer.

Most works focused on separating the singing voice from the instrumental accompaniment and estimate magnitude spectrograms or masks for the target source from a mixture spectrogram as an input. However, the question of which source can be estimated depends mainly on the available training data and not as much on the DNN architecture. In fact, in the first works on DNN-based music source separation the task was solved with networks comprising mainly RNNs [78], only fully connected layers [196], or CNNs [14].

Subsequent work explored ways to improve upon those first promising results using DNNs. Better performance was obtained through more advanced network architectures [86, 190, 191], more data [86, 71], data augmentation strategies [197, 23] and combining several DNNs or their predictions [186, 197]. It was also proposed to incorporate phase information by processing and predicting complex spectrograms [219, 109]. An influential work applied the U-Net architecture [157], which was originally developed for medical image processing, to singing voice separation using a large proprietary dataset [86]. The work showed that very good separation results can be obtained by training a generic DNN on large databases. The U-Net is a CNN consisting of an encoder and a decoder. The encoder takes a magnitude spectrogram as input and processes it with several

two-dimensional convolutional layers. Each layer downsamples the dimensions corresponding to time and frequency and increases the number of channels along the third dimension. The decoder consists of transposed convolutional layers [37] which upsample the time and frequency dimension and decrease the number of channels until the input spectrogram size is reached again. The output of each encoder layer is concatenated with the corresponding representation in the decoder having the same size. This facilitates the gradient-based optimisation and assures that no high resolution information has to be encoded in layers with lower resolution. This multi-scale processing was shown to be very effective for source separation and other audio processing tasks.

Another line of research developed DNNs which process the mixture and predict the sources directly in the time domain. This end-to-end approach has the theoretical advantages that phase information is not neglected and no decisions concerning hand-crafted input features need to be taken. The latter aspect makes such approaches even more data-driven. First, the U-Net was adapted to time domain processing by replacing 2D with 1D convolutional layers [184]. Another approach was based on the time domain speech synthesis network Wavenet [116, 139]. However, both models did not perform better than their counterparts in the time-frequency domain [184, 116]. The architecture of [184] was then improved in [28] by including LSTM layers and increasing the stride and number of channels in convolutional layers. This improved the separation performance but also greatly increased the number of parameters making the model computationally expensive. End-to-end models were also proposed for speech separation and surpassed separation performance of oracle time-frequency masks [117]. Such a result remains to be shown for music signals. This may be due to the fact that end-to-end models require amounts of training data which are not widely available for music.

Usually, one instance of a DNN is trained to separate exclusively one specific target instrument from a mixture. Two works have shown independently that this is an inefficient use of the network's capacity [126, 161]. They separated several different instruments with one DNN which was extended by a control mechanism to select the desired target source with a one-hot vector as input. The separation quality was on par with instrument-specific networks.

To sum up, data-driven deep learning approaches have the potential to provide better separation quality than knowledge-driven methods under the condition that enough computational resources and data are available. The latter is not always given in the music domain. The increased performance comes at the cost of higher resource demands and a lack of interpretability of the model parameters.

3.3 Discussion

We have seen that both knowledge-driven and data-driven methods come with their own strengths and limitations. Knowledge-driven approaches are based on precise algebraic problem formulations or signal models and, hence, their behaviour is often straightforward to analyze. They are often computationally relatively light and do not depend strongly on the availability of training data. The methods are either learning-free, when all parameters are estimated based on the test mixture, or learn spectral templates on small amounts of data. On one hand, the usage of prior knowledge enables good separation results. On the other hand, it implies strong assumptions. Especially when dealing with music signals it is challenging to formulate models which are precise enough to be useful for the separation and at the same time do not constitute an oversimplification and keep an appropriate degree of flexibility.

In contrast, data-driven deep learning approaches avoid explicit modeling and strong assumptions. They enabled massive performance improvements using high capacity learning models trained on large datasets. However, the fact that performance gains can be achieved through larger models and/or more data has led to a rise in resource demands. In fact, in most research fields which were revolutionized by deep learning the focus has been almost exclusively on performance while efficiency was often neglected [170]. Audio source separation is not an exception. At the same time, available prior knowledge is ignored by most data-driven methods.

There lie opportunities in the combination of data and knowledge based information for source separation. Integrating prior knowledge into deep learning approaches may lead to better performance, higher efficiency, less dependence on training data quality and quantity, and in general more robust methods. In this context, data quality does not only concern the audio quality of recordings but also includes the question to which extent labels or isolated signals are available for supervised training.

Some combined approaches were already proposed. For example, in [79] the computation and application of soft masks was included in the optimization procedure for RNN training. It enforced the constraint that the estimated sources add up to the mixture. Side information was also used in music source separation with DNNs. In [129] the mixture spectrogram is filtered with a soft mask derived from an aligned musical score and the filtered mixture is then processed by a CNN to make the final source estimation. In [47] score information is used to impose structure on hidden representations in a DNN trained for source separation. Another interesting way of using model-based knowledge was to let a DNN estimate parameters of a synthesis model to generate the target source signal instead of predicting masks for filtering the mixture [12]. The synthesis model effectively limits the output space of the estimation problem but the synthesis quality is highly sensitive to errors in the estimated parameters.

More works on data-driven methods which exploit prior knowledge are reviewed in the related chapters of this thesis. This line of research has already shown promising results. We hypothesize that there is still potential to explore new ways of using side information and/or model-based knowledge for music source separation with deep learning. The goal of this dissertation is to make contributions to this research direction.

Part II

Contributions

Chapter 4

Weakly Informed Audio Source Separation

Summary

In this chapter a generic DNN architecture for informed audio source separation is proposed which exploits side information which is only coarsely aligned. We conduct singing voice separation experiments using artificial side information as proof of concept. Furthermore, we test the model on text-informed speech-music separation with joint text alignment. This chapter builds a basis for the next chapter dealing with text-informed singing voice separation. It is based on the publications *Weakly Informed Audio Source Separation* [166] and *Joint Phoneme Alignment and Text-Informed Speech Separation on Highly Corrupted Speech* [167].

Contents

4.1 Introduction

A challenge for the usage of any side information for audio source separation is that it needs to be temporally aligned with the mixture signal. However, musical scores or text transcripts usually come without any alignment information. Therefore, they are considered as weak side information.

Automatic alignment of side information with mixtures is especially difficult because the audio signals do not only contain the signal corresponding to the side information but also other sound sources. For example, methods for text-to-speech alignment are developed for rather clean speech and do not perform as well on corrupted speech [89, 11]. Also, automatic lyrics alignment methods assume isolated vocals [62] or require prohibitively large databases for training [182].

This results in a chicken and egg problem: alignment is required for text-informed voice separation and clean voice signals are required for high quality automatic alignment. Our hypothesis is that performing both tasks jointly leads to mutual benefits. The separation component facilitates alignment on mixtures while the alignment makes the text information more useful for the separation task. Apart from the separation task, aligning text at phoneme level on mixed speech or singing voice has interesting applications such as generating training data for robust speech or lyrics recognition systems or aligning subtitles for movies.

We propose a novel deep learning based separation method that employs weak side information using an attention mechanism (cf. Section 2.3.5). The model has a sequential encoder-decoder architecture where the decoder is connected to the side information via attention. The whole side information sequence is thus accessible to the decoder at all time steps. During training, the model learns to evaluate the relevance of the side information elements with respect to the separation task. Hence, only the most informative elements are taken into account for the separation at each time frame. The relevance is reflected in the attention weights from which alignment information can be retrieved. Therefore, the model allows to perform informed separation and alignment jointly.

In this chapter, we evaluate the proposed method in two sets of experiments. As a proof of concept, we first perform informed singing voice separation using artificial side information with different levels of expressiveness. Thereafter, we perform a second set of experiments on text-informed speech-music separation with joint text alignment. Written words are decomposed into phonemes (cf. Section 2.2.2) which contain information about the sounds produced by a speaker or singer, e.g. if they are voiced or unvoiced, their phonetic class, and order of appearance. We show that the quality of the separated speech can be improved through text information without pre-alignment. Beyond, as a method for phoneme level text-to-speech alignment, the model achieves good results on clean speech as well as on strongly corrupted speech with a Signal-to-Noise Ratio (SNR) of -5 dB.

Using the model on speech signals is an important step towards text-informed singing voice separation with joint lyrics alignment which will be addressed in Chapter 5. It is uncertain if the phonetic information of a text transcript represents exactly the phonetic content of a voice signal. The probability that a speech signal is well represented by a text transcripts is higher than for a singing voice signal. This is because intelligibility is the primary goal of speech signals while singing voice also focuses on artistic expression following a certain melody and rhythm.

Therefore, we assume that the model for text-informed voice separation must first work on speech signals in order to be also applicable to the more challenging case of singing voice. Moreover, there is no publicly available dataset for text-informed singing voice separation. This is an additional motivation to validate our approach first using existing datasets for speech before

creating such a dataset for singing voice.

Finally, speech-music separation is an interesting task in its own right. Speech enhancement research focuses mainly on noise as interfering source [207]. However, musical sound sources also often corrupt speech signals. For example, speech-music separation is used to separate dialogues from background music in movies or to make voice commands intelligible in the presence of music which is important for voice controlled home speakers. The speech-music separation task has mainly been studied in simplified settings so far [30, 105].

4.2 Related work

4.2.1 Weakly labeled data

Training models with weakly labeled data remains a challenging problem for a variety of audio related tasks [47, 129, 162, 101]. In this context, multi-instance learning has been applied to singing voice detection [162] and acoustic event detection [101] to gradually refine the labeling during supervised training, but with limited effectiveness [47, 162]. For informed source separation, it has been proposed to approach training with weak labels in an unsupervised fashion [47]. The side information is then used to enforce structure on the latent representation of the mixture within an autoencoder model. While the approaches above aim for training with weakly labeled data only, we intend to complement supervised strong label training with additional weaker side information. In [129], a tolerance window allows for misalignment of around 0.2 seconds during score-informed source separation. The model proposed in this work allows for much coarser alignments. Instead of explicitly guiding the network regarding how to use the side information as done in [47, 129], the proposed model learns the best use of the side information for the separation task from data.

It has been tested in [105] if the alignment of side information can be improved during text-informed source separation with NMF. No improvement over the pre-alignment could be reported, while the authors stated that it would have been beneficial for the separation quality. We show in experiments that our model can indeed improve the alignment by a considerable extent.

4.2.2 Text-informed speech separation

Text-informed speech-music separation has been studied first in [104]. An example speech signal is synthesized from the text transcript and then aligned with the observed mixture using Dynamic Time Warping (DTW). The separation is done with a variant of NMF exploiting similarities between the target speech and the example speech signal. The results show that text information is beneficial for the separation. In [91], text-informed speech enhancement is done using a DNN. A sequence of phonemes is forced-aligned with noisy speech and then fed to the DNN together with the audio features. The authors show that the text information improves the separation in terms of cepstral distance to clean speech. Information about phoneme identities is exploited for speech separation in [209] and [16] without using text-transcript as additional input. Instead, the phonemes are recognized from the input signal using Automatic Speech Recognition (ASR) techniques. Then, pre-trained phoneme-specific networks perform the separation. Additional effort is required to compensate for the limited performance of ASR on corrupted speech [209, 16].

Text-to-speech alignment faces two challenges: very long audio signals and corrupted speech. While some approaches cope with the former [89, 11], the latter is far from being solved for low SNRs. The method based on probabilistic kernels in [11] can align text with long audio signals but

performance decreases when the speech is mixed with music. The approach in [132] applies ASR on a long speech signal and aligns a given text transcript with the recognized text. The process is iterated with an updated vocabulary and language model for regions that have not been aligned with high confidence in previous iterations. It can deal with noisy speech with an SNR of 15 dB. In [89], this approach is further improved by also updating the acoustic model on non-aligned regions leading to good alignment results up to an SNR of 10 dB. The Montreal Forced Aligner (MFA) [122] is a more advanced alignment method. It uses a Gaussian Mixture Model (GMM) Hidden Markow Model (HMM) ASR system and is trained in three iterative steps. First monophone, then triphone GMMs are trained iteratively, as in [89], to generate alignments on which acoustic feature transforms for speaker adaptation are learned as a third step.

The alignment capabilities of the attention mechanism have been already observed in [20] on a speech recognition task but have not been evaluated for alignment. Attention has been recently used to cope with non-aligned training data for a singing voice transcription task in [138].

4.3 Proposed method

Let $x(t)$ be the observed single-channel mixture signal at discrete-time t. Let $\mathbf{Y} \in \mathbb{R}^{D \times M}$ be a side information sequence with feature dimension D and M time steps. If phonemes are used as side information, $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{M-1}]$ is a sequence of M phonemes represented as D-dimensional one-hot vectors $\mathbf{y}_m$ being a precise transcription of the utterance contained in $x(t)$. Our goal is to separate $x(t)$ into a target source image $v(t)$ and a mixture of all remaining source images $a(t)$. Moreover, we aim to predict the onset time of each side information element (e.g. a phoneme) in the mixture signal.

The proposed model takes as inputs the magnitude of the mixture's STFT $|\mathbf{X}| \in \mathbb{R}^{F \times N}$ with F frequency bands and N time frames as well as the information $\mathbf{Y}$. The output is an estimate of the target's magnitude STFT $|\hat{\mathbf{V}}| \in \mathbb{R}^{F \times N}$. An inverse STFT of $|\hat{\mathbf{V}}|$ combined with the mixture phase is performed to obtain the target estimation $\hat{v}(t)$ in the time domain. Assuming a linear mixture model, the estimation of remaining sources $\hat{a}(t)$ is obtained as $\hat{a}(t) = x(t) - \hat{v}(t)$.

4.3.1 Base model

The proposed model comprises four building blocks, namely a mixture encoder, a side information encoder, an attention mechanism, and a target source decoder as shown in Figure 4.1. A PyTorch implementation of the model is available online[1].

The mixture encoder is a two-layer deep Bidirectional Recurrent Neural Network (BRNN) [169] with Long Short-Term Memory (LSTM) cells [75]. Given $|\mathbf{X}|$ which is viewed as a sequence of N column vectors $[\mathbf{x}_0, ..., \mathbf{x}_{N-1}]$, it computes the matrix $\mathbf{G} = [\mathbf{g}_0, ..., \mathbf{g}_{N-1}] \in \mathbb{R}^{E \times N}$ which we call the mixture encoding. It has feature dimension E and length N over time.

The side information encoder has the same architecture as the mixture encoder. Given the sequence of side information frames $\mathbf{Y}$, it computes $\mathbf{H} = [\mathbf{h}_0, ..., \mathbf{h}_{M-1}] \in \mathbb{R}^{J \times M}$ which we call the side information encoding with feature dimension J.

The target source decoder gets as inputs the mixture encoding $\mathbf{G}$ and a representation of the side information encoding denoted $\mathbf{C}$, which is computed by the attention mechanism as will be explained further below. Both inputs are concatenated along the feature dimension, which

[1] `https://github.com/schufo/wiass`

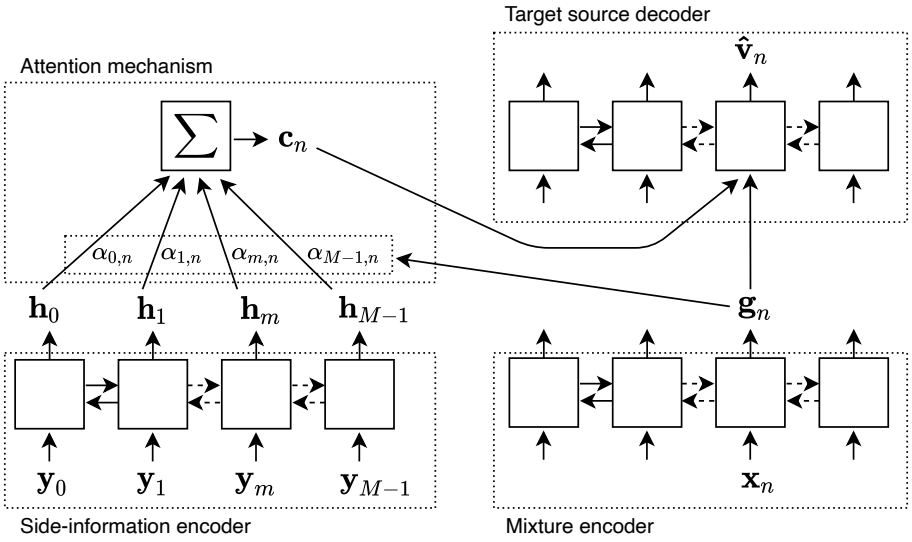


Figure 4.1: Schematic model architecture and workflow of the attention mechanism to compute the n-th prediction frame $\hat{\mathbf{v}}_n$.

is denoted by $[\mathbf{c}_n, \mathbf{g}_n]$. The decoder computes one time frame of the target source estimation $|\hat{\mathbf{v}}_n|$ through the following layers. $\mathbf{W}$ and $\mathbf{b}$ are learnable weights and biases respectively in the equations below. First, a fully connected layer computes the hidden representation $\mathbf{q}_n^{(1)}$:

$$\mathbf{q}_n^{(1)} = \tanh(\mathbf{W}_1[\mathbf{c}_n, \mathbf{g}_n] + \mathbf{b}_1). \tag{4.1}$$

Then, a two layers deep BRNN with LSTM cells, as used in the encoders, computes the hidden representation $\mathbf{q}_n^{(2)}$. Finally, another fully connected layer with ReLU activation [119] computes the estimation:

$$\hat{\mathbf{v}}_n = \max(0, \mathbf{W}_2\mathbf{q}_n^{(2)} + \mathbf{b}_2). \tag{4.2}$$

Predicting time-frequency masks as in [128] instead of magnitude spectrograms directly did not lead to better results in our experiments.

The attention mechanism [6] identifies the relevant elements in the side information sequence for each time step n of the target source decoding and summarizes them in a context vector $\mathbf{c}_n$. Consequently, the decoder can find at every time step the relevant side information elements no matter where they are placed in the sequence. This makes a pre-alignment redundant. We closely follow the attention mechanism proposed in [6] and refined in [118] (cf. Section 2.3.5).

For time step n of the decoder, the vector $\mathbf{c}_n$ is computed as follows. A score $s_{m,n}$ is calculated representing the similarity between the mixture encoding $\mathbf{g}_n$ and each of the M side information encoding steps $\mathbf{h}_m$:

$$s_{m,n} = \mathbf{g}_n^\top \mathbf{W}_s \mathbf{h}_m \;\; \forall m \in \{0, 1, ...M-1\} \tag{4.3}$$

where $\mathbf{W}_s \in \mathbb{R}^{E \times J}$ is a matrix of learnable weights. Then, attention weights $\alpha_{m,n}$ are computed

from the scores by a softmax operation:

$$\alpha_{m,n} = \frac{e^{s_{m,n}}}{\sum_{k=0}^{M-1} e^{s_{k,n}}}. \tag{4.4}$$

Each side information element $\mathbf{h}_m$ thus has a dedicated weight $\alpha_{m,n}$ reflecting its importance for the decoder time step n as a probability. The context vector $\mathbf{c}_n$ is the weighted sum of all side information encoding elements:

$$\mathbf{c}_n = \sum_{m=0}^{M-1} \mathbf{h}_m \alpha_{m,n}. \tag{4.5}$$

The target source estimate is then computed from the context vector and the mixture encoding $\mathbf{g}_n$ as described above. The alignment between mixture and side information is reflected in the attention weights $\alpha_{m,n}$ and is learned without any additional term in the loss function. Note that attention was originally proposed to align decoder hidden states with the input sequence (cf. Section 2.3.5). In contrast, we use it to align two input sequences.

4.3.2 Adaptation for text-informed speech-music separation

We derive three versions from the base model introduced above by modifying the way the phoneme sequence $\mathbf{Y}$ is processed, which we identified as a crucial point for the use of text as side information. It is worth mentioning that the phoneme encoding $\mathbf{H}$ serves two distinct purposes: (1) being an input to the attention mechanism identifying which phoneme is relevant at which mixture time frame and (2) being an input to the target source decoder in the form of $\mathbf{c}_n$ to inform the separation process.

For Version 1 (V1) we reduce the number of LSTM-RNN layers in the side information encoder to one and thereby limit its capacity. This leads to a more general representation of phonemes in $\mathbf{H}$ making it more applicable to fulfill its two purposes at once. Moreover, it reduces overfitting in limited data settings. Version 2 (V2) is identical to V1 except for an unidirectional LSTM-RNN in the phoneme encoder. This further reduces the number of learnable parameters and forces the model to process the phonemes strictly from left to right. Version 3 (V3) is equal to V1 but $\mathbf{h}_m$ is processed by a linear layer l before going into $\mathbf{c}_n$. This changes equation (4.5) to

$$\mathbf{c}_n = \sum_{m=0}^{M-1} l(\mathbf{h}_m) \alpha_{m,n} \tag{4.6}$$

and means the model can learn two different representations of phonemes for their two purposes.

4.3.3 Retrieving phoneme onsets from attention weights

Given a sequence of phonemes $\mathbf{Y}$ and a corresponding audio signal $x(t)$ containing speech, the goal of phoneme-to-audio alignment is to estimate the onset times of each phoneme in the audio signal. We retrieve onsets from the attention weights using the DTW algorithm [204].

The attention weights can be represented collectively as attention matrix $\mathbf{A}$ with shape (M, N) as shown in Figure 4.2. With DTW, we find the optimal path through $\mathbf{A}$ from $(0, 0)$ to $(M-1, N-1)$ indicating which phoneme is active in which spectrogram frame. It maximizes the sum of attention weights it passes being restricted to only two possible moves, namely $(m, n+1)$ and $(m+1, n+1)$. This means we assume that all phonemes are pronounced and given in the correct order. An

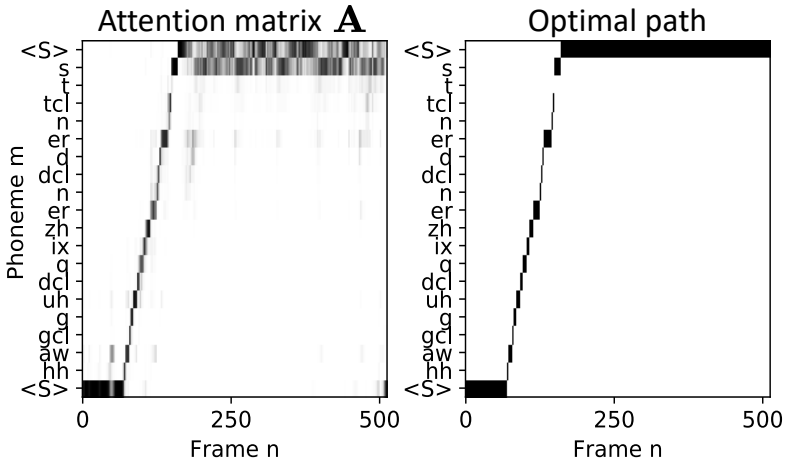


Figure 4.2: Attention matrix (left) and DTW optimal path (right). Darker color represents higher values. All values are in [0, 1].

optimal path obtained this way is shown in Figure 4.2.

Knowing the hop size of the STFT that has been performed on $x(t)$ we know the exact position in time of each time frame. The estimated onset time of a phoneme is the mid-point of the first time frame it has been assigned to by the optimal path.

Since our approach learns the alignment as a side outcome of learning speech separation, it can cope with much lower SNRs than other alignment methods which learn acoustic models from clean speech data. Training data with annotated phoneme onsets are not required.

4.4 Separation evaluation of silent frames

The metrics SDR, SAR, and SIR are typically computed on non-overlapping frames for which $t \in \{0, 1, ..., T' - 1\}$ and T' is chosen so that the frame has a length of one second. The median over the frame scores is taken to represent the performance on the whole signal [185].

However, for frames with a silent true source ($v(t) = 0 \ \forall t$) or prediction ($\hat{v}(t) = 0 \ \forall t$), the metrics are undefined [202]. The MUSDB test set [151] has 2600 frames with silent vocals and 103 frames with silent accompaniment. As a result, at least about 45 out of 210 minutes are systematically ignored during evaluation, with potentially more frames being ignored when the prediction is silent. This issue has also been observed in [183], where the authors suggest reporting the root mean square energy of the prediction for frames with silent ground truth. Inspired by this suggestion, we propose the Predicted Energy at Silence (PES) score. It is the energy in the predictions at those frames with silent ground truth:

$$\text{PES} = \begin{cases} 10 \log_{10} \sum_{t=0}^{T'-1} \hat{v}^2(t) & \text{if} \quad \sum_{t=0}^{T'-1} v^2(t) = 0 \\ n.d. & \text{otherwise} \end{cases} \tag{4.7}$$

The PES reflects a method's capability to not get confused by other sources while the target is not active.

In order to include every single test frame in the evaluation, we also need to evaluate frames for which silence is predicted while the ground truth is not silent. Therefore, we propose also the Energy at Predicted Silence (EPS) score, which is (the mean) of the ground truth energy of all

frames with silent prediction:

$$\text{EPS} = \begin{cases} 10 \log_{10} \sum_{t=0}^{T'-1} v^2(t) & \text{if} \quad \sum_{t=0}^{T'-1} \hat{v}^2(t) = 0 \\ n.d. & \text{otherwise} \end{cases} \tag{4.8}$$

The EPS reflects a method's capability to predict silence at the correct time. We suggest to report the mean of the PES and EPS over all frames where they are defined. Note that in contrast to SDR, SAR, and SIR, the scores for PES and EPS are the better the lower they are.

We made a Python script to compute the metrics publicly available[2].

4.5 Experimental proof of concept

We perform monaural singing voice separation with the proposed model using artificial side information about the singing voice with different levels of expressiveness.

4.5.1 Experimental setup

Data

We use the publicly available dataset MUSDB [151] comprising a 100 tracks training set and a 50 tracks test set containing various genres. We split the training set into 80 tracks for training and 20 tracks for the validation set. All songs are converted to mono, downsampled to 16 kHz, and cut into fragments of 8.2 seconds. The STFT is computed on each fragment with a Fast Fourier Transform (FFT) length of 1024, Hamming window, and hop size of 512 leading to magnitude spectrograms of size $(F \times N) = (513 \times 256)$. Each magnitude spectrogram is divided by its maximum value to normalize it to the range $[0; 1]$.

As data augmentation we set the energy ratio between vocals and accompaniment to a value uniformly drawn from the ± 2 dB range around the original energy ratio. We also shift the mixture's pitch by w half tone steps, with w being uniformly drawn from $[-2; 2]$. These random operations are repeated four times on each original fragment leading to 8152 fragments for training in total.

We use this limited amount of publicly available data to make our results easier to reproduce. However, it is not straightforward to evaluate whether performance of data-driven methods is limited by the model's architecture or the amount of training data [185]. We therefore repeat all experiments with additional training data (65 rock-pop song excerpts with 96 minutes total length) to test if performance is scalable.

Training

We train the model on batches of 128 spectrograms randomly drawn from the training set. The loss function is the L1 loss. The Adam optimizer [90] is used with learning rate 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay rate 0.001. We set both the size E of the mixture encoding and size J of the side information encoding to 513. We select the model with the lowest validation cost after 100 epochs without improvement of the validation cost.

[2] `https://github.com/schufo/wiass`

Side information

The side information $\mathbf{Y}$ has length M, which can be equal to or different from the mixture length N. We use side information with feature size $D = 1$.

We use two baseline models. The first one (BL1) only consists of the mixture encoder and target decoder. It does not use any side information. As second baseline (BL2) we use the full proposed architecture, which is also used in all subsequent experiments, and provide only meaningless side information: a sequence of ones. This allows us to investigate to which extent the added learning capacity of the attention mechanism and side information encoder improves performance. Next, we investigate whether performance can be further improved with meaningful side information.

First, we provide the total vocals Magnitude (M) for each time frame as side information. It is derived from the ground truth spectrograms by summing the magnitudes of all frequencies at each time step: $\mathbf{Y} = \sum_{f=0}^{F-1} |v_{f,n}|$ where $v_{f,n}$ is the time frequency bin of the true source spectrogram at frequency index f and time frame n. It is considered as very strong information, since it has the same length as the mixture ($M = N = 256$) and is numerically closely related to the ground truth. We call it M1. We then derive M2 from it by padding both sides of the sequence so that $M = 300$. We use 100 as padding value and randomly choose the padding length on both sides for each batch. As a result, M2 conveys the same strong information as M1 but is less synchronized to the mixture. The position of relevant information varies from batch to batch during training and from example to example during testing.

Binary sequences indicating vocal Activity (A) and non-activity are derived from M1 by setting all time steps with total magnitude values below 0.1 to zero and all other steps to 1. In practice, such weak information can be obtained by vocal activity detection methods [163]. For experiment A1 we pad the binary sequence to length $M = 300$ keeping the padding value 100 following the procedure of M2 to de-synchronize it from the mixture.

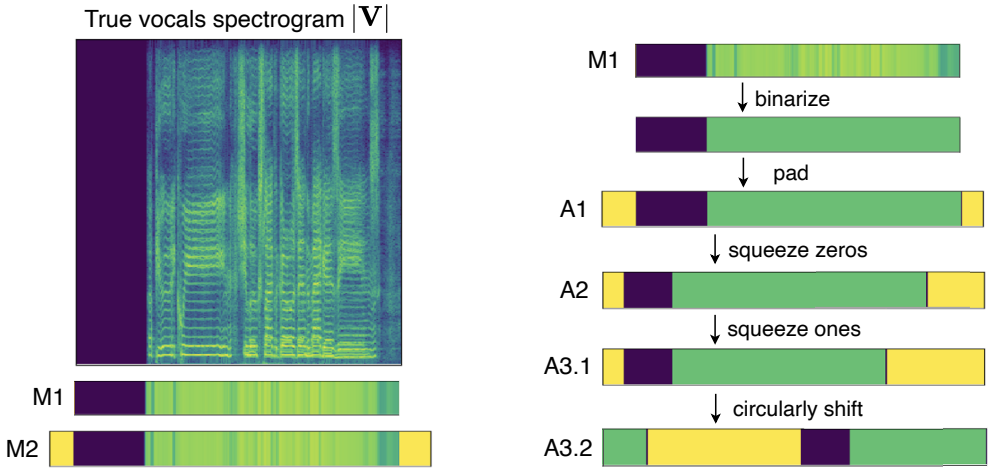


Figure 4.3: Visualization of the artificial side information used in the experiments. Dark blue indicates zero, padding is shown in yellow.

For experiment A2, we further weaken the information by deleting a random number w of zeros in each sub-sequence of zeros in the binary sequence. We draw w uniformly from $[1; L/2]$ for each example, where L is the length of a sub-sequence of zeros. We pad the remaining binary sequence to length $M = 300$ as above. The sequence $\mathbf{Y}$ now only contains information about the number of appearances of silent parts in the vocals and their position relative to non-silent parts. Information

about the silence length is almost completely lost.

For experiment A3, we weaken the binary information even further by additionally reducing the length of sub-sequences of ones with the same rule as applied to zeros in A2. We also pad to length $M = 300$. Now, $\mathbf{Y}$ carries only information about the alternations between vocal activity and silence. We test the model trained with side information A3 in two different inference settings. First, with the same side information as seen during training (A3.1), then with this side information circularly shifted by 100 steps (A3.2). An overview of the different side information types is presented in Figure 4.3.

4.5.2 Results and discussion

The evaluation results for models trained only on MUSDB are shown in Figure 4.4. The results for models trained using additional data are shown in Figure 4.5. The use of additional data is indicated by the '+' added to the experiment name.

Each data point in the boxplots represents the median over all evaluation frames of one test song following the procedure described in Section 4.4. The box extends from lower to upper quartile with the line inside representing the median. The whiskers extend over the whole data range. Note that for the proposed PES and EPS metric lower values are better, while for the standard metrics higher values are better.

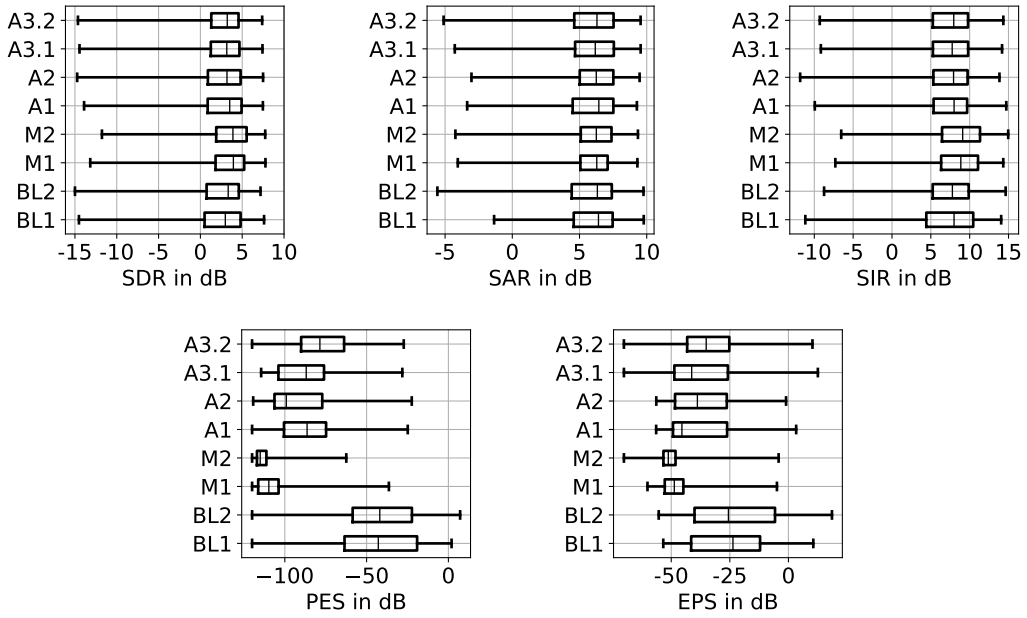


Figure 4.4: Boxplots of the source separation evaluation results for experiments using only MUSDB data. For SDR, SAR, SIR higher values are better, while for PES and EPS lower values are better. BL: baseline, M: vocal magnitude side information, A: vocal activity side information.

The relative results do not change substantially when using more training data. The baselines BL1 and BL2 achieve a median SDR of 3.0 dB and 3.33 dB respectively, which, given the amount of training data and simplicity of the model, can be considered an appropriate baseline. The improvement of BL2 over BL1 shows that the proposed model can leverage the additional capacity even with meaningless side information. Adding only 96 minutes of training data (BL2+) improves

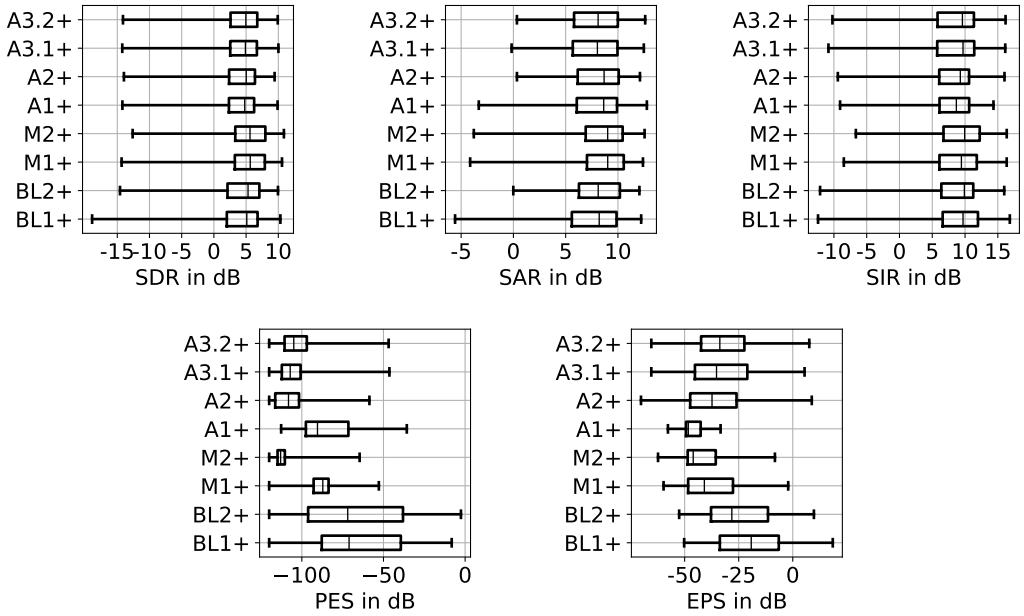


Figure 4.5: Boxplots of the source separation evaluation results for experiments using MUSDB and additional data. For SDR, SAR, SIR higher values are better, while for PES and EPS lower values are better. BL: baseline, M: vocal magnitude side information, A: vocal activity side information.

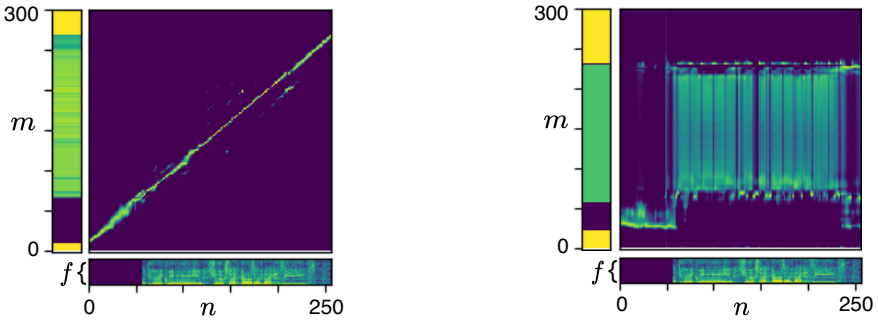
performance on all metrics so that the baseline would have only been outperformed by models trained on much more data in the Signal Separation Evaluation Campaign 2018 [185].

The use of all types of meaningful side information considerably improves performance on silent vocal frames resulting in a much lower PES and predicting silence at the right time resulting in a lower EPS. In case of M1 and M2, the SDR and SIR are also improved, while with the binary vocal activity side information the standard metrics do not change much compared to the baselines. These observations are in line with [183]. For frames with high vocal energy, a lot of information about the vocals is already contained in the mixture. Consequently, the binary side information does not add information for these frames, while the vocal magnitude information does. For frames with silent or near-silent vocals, any other source can potentially be mistaken as vocals leading to wrong predictions. In this case, the binary information is useful to understand the alternations between vocal activity and non-activity. The fact that M2 performs slightly better than M1 can be explained by the data augmentation effect of the random padding in M2.

In general, it is not surprising that additional information leads to better separation results. Our contribution lies rather in the fact that the proposed model can exploit such information despite its weakness. Note that the binary side information types carry less information than a musical score. Audio examples are available online[3].

In addition to improving source separation performance by exploiting weak side information, the proposed model also provides an estimate of the alignment between the side information and the mixture through the attention weights. In Figure 4.6 the attention weight matrix $\mathbf{A}$ is shown for experiments M2 and A3.1 for one fragment of the MUSDB test track *Schoolboy Fascination*, which is also available as audio example. On the left of each matrix $\mathbf{A}$ the corresponding side information is depicted vertically with time step index m. Dark blue indicates a zero value, while

[3] `https://schufo.github.io/publications/2019-WASPAA`



(a) Experiment M2 (b) Experiment A3.1

Figure 4.6: Attention weights **A** containing alignment information. The side information is shown vertically on the left of **A** and the true vocals spectrogram below. Lighter color indicates higher values.

padding is shown in yellow. Below **A** the true vocals spectrogram is shown with frequency bin index f and time frame index n. The lighter the color at point (m, n) in **A** the more the side information element at m is taken into account for producing the prediction at time step n. For M2 a very exact alignment to the mixture is learned, it becomes a bit blurry at the silent vocal part, where the side information contains low and therefore similar values. For A3.1 the model learned to look at ones and zeros at the right time, although the sub-sequences are much shorter than the corresponding parts in the true vocals. The model learned to never look at the padding values. The attention weights show that the model has indeed learned to find the relevant side information at each time step without any pre-alignment.

4.6 Experiments on text-informed speech-music separation with joint text-to-speech alignment

4.6.1 Experimental setup

We perform text-informed speech-music separation with joint text-to-speech alignment with the models V1, V2, V3 described in section 4.3.2. As baseline (BL) for the separation task, we use a model with the same configuration as V1. It resembles the speech separation model in [17] which is a four-layer LSTM-RNN with a linear output layer. Compared to [17], the BL has more expressive power through the attention mechanism and the phoneme encoder. It gets, instead of phonemes, a sequence of ones as side information, which does not convey any additional information about the speech signal to be separated. At the same time, BL has the same computational capacity as the models under test. This allows us to observe the exact effect of text as side information. We share the code of all models and experiments online[4].

Data

We use the instrumental accompaniments of the MUSDB dataset [151] as music signals and mix them with speech signal of the TIMIT corpus [53]. All music signals are converted to mono, downsampled to 16 kHz, and cut into snippets of 8.2 seconds, which is longer than all available

[4] `https://github.com/schufo/tisms`

speech signals. For training, we mix snippets of 80 MUSDB tracks with 4320 TIMIT utterances. The validation set contains 20 music tracks and 240 utterances and the test set 50 music tracks and 1344 utterances. The start time of the utterance within the 8.2 seconds of music is chosen randomly and differs for every example and every epoch. During training, we mix speech and music with a SNR uniformly drawn from $[-8, 0]$ dB. For the validation and test set, we mix with SNR = -5 dB. The SNR is calculated only on the signal parts where the speech is active. There is no utterance or speaker overlap between the training, validation, and test set. The STFT is computed with Fast Fourier Transform length 512, Hamming window, and hop length of 256 leading to magnitude spectrograms of size $(F \times N) = (257 \times 512)$. Each magnitude spectrogram is divided by its maximum value to normalize it to the range $[0, 1]$.

As text information, we use the available phoneme level transcripts for the TIMIT speech recordings. The phonetic alphabet used in TIMIT is an extended version of the ARPAbet (cf. Section 2.2.2). It comprises 60 different phoneme symbols to which we add a silence token (<S>) and a padding token for batching. The silence token is added to the start and end of each phoneme sequence because the speech is not active at the beginning and end of the mixture signal.

Training

We use the L1 loss, batch size 32, and the Adam optimizer [90] with learning rate 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$. The learning rate is reduced to 10^{-5} for the first 200 epochs. We stop training after 200 consecutive epochs without a decrease in validation cost.

4.6.2 Results and discussion

Speech-music separation results

We evaluate the predicted speech signals in terms of the objective separation quality metrics SDR, SAR, and SIR [202]. We also compute the Perceptual Evaluation of Speech Quality (PESQ) [156] and the Short-Time Objective Intelligibility (STOI) measure [189]. A brief introduction to the metrics was given in Section 2.1.2. The SDR, SAR, and SIR are computed on non-overlapping frames of 1 second length and the median value is taken to represent performance on one test example.

In Table 4.1, the median over the test set is presented for all metrics. Given the difficulty of the task (the SNR is -5 dB), BL performs well. The median STOI and PESQ of the corrupted speech in the test set are 0.64 and 1.48, respectively, which BL improves considerably. V1, V2 and V3 improve the PESQ over BL. This indicates that text information can improve the perceived quality of separated speech signals. The SDR, SAR, and SIR are only slightly changed compared to BL.

To test the upper bound of separation quality improvement through phoneme information in our experiment setting, we take our best model V2 and input the Optimal Attention (OA) weights during training and testing instead of learning them. For OA, we set $\alpha_{m,n}$ to 1 if phoneme m is active in frame n and to 0 otherwise based on the true phoneme onsets available with TIMIT. We can see in Table 4.1 that the SDR, SAR, and PESQ for OA improves over BL. This result shows that text information can improve speech separation further when the alignment is provided instead of learned jointly.

	SDR	SAR	SIR	STOI	PESQ
BL	8.81	**10.60**	14.53	0.87	2.66
V1	8.64	10.39	14.44	0.87	2.72
V2	**8.86**	10.57	**14.55**	**0.88**	**2.74**
V3	8.76	10.47	14.53	**0.88**	**2.74**
OA	*8.93*	*10.70*	*14.58*	*0.88*	*2.84*

Table 4.1: Separation quality evaluation results, all values are medians over the test set. SDR, SAR, SIR are shown in dB. BL: Baseline, V1-3: Version 1-3, OA: Optimal Attention weights.

We also provide audio examples online[5]. In informal listening tests we observed that while some word endings are not audible in baseline predictions, they are clearly audible in the predictions of V2 and OA.

Text-to-audio alignment results

As baseline for the phoneme alignment task, we use the Montreal Forced Aligner (MFA) [122]. It is an open source trainable alignment model based on the speech recognition toolkit Kaldi [147]. It learns an acoustic model using GMM-HMMs. To train it, we follow closely the procedure described in [122] which leads to advantageous conditions for the MFA: It is trained on all available data (training, validation and test data), it gets the speaker identity of each utterance to perform speaker adaptation, and each example is cut at the start and end of the utterance for training and testing (no long "music-only" parts). We test all methods on the test set for two cases: clean speech and SNR = -5 dB. The MFA is trained on clean and corrupted speech respectively to learn appropriate acoustic models.

We evaluate the Mean Absolute Error (MAE) on each test example. It is the mean of the absolute differences between the true and predicted phoneme onsets in milliseconds (ms). The mean and median MAE over the test examples are shown in Table 4.2. We see that V2 is not suited for phoneme alignment, whereas it performed best on the separation task. On clean speech, the MFA and V1 perform almost equally well. V1's median is better indicating that its alignments are more accurate when neglecting outliers. V1's mean is worse indicating that it produces more severe outliers. This can be explained by the dependence of our alignment method on somewhat sharp attention weights. When the model focuses on many phonemes at each time step n, i.e. α is not sharp, an optimal path indicating accurate phoneme onsets cannot be found.

	Clean speech		SNR = -5 dB	
	mean	median	mean	median
MFA	**16.3**	15.7	**38.4**	26.0
V1	22.5	**12.9**	39.0	**16.1**
V2	326.2	75.4	355.0	120.2
V3	48.1	14.4	69.0	17.9

Table 4.2: Mean Absolute Error (MAE) of phoneme onset predictions in ms averaged over the test set. MFA: Montreal Forced Aligner, V1-3: Version 1-3.

On corrupted speech with SNR = -5 dB, V1 clearly outperforms the MFA. The mean of both

[5] `https://schufo.github.io/publications/2020-ICASSP`

methods is very similar while V1's median MAE is almost 10 ms lower. In general, V3 does not perform as well as V1 but still gives accurate predictions and outperforms the MFA regarding median MAE on clean and corrupted speech.

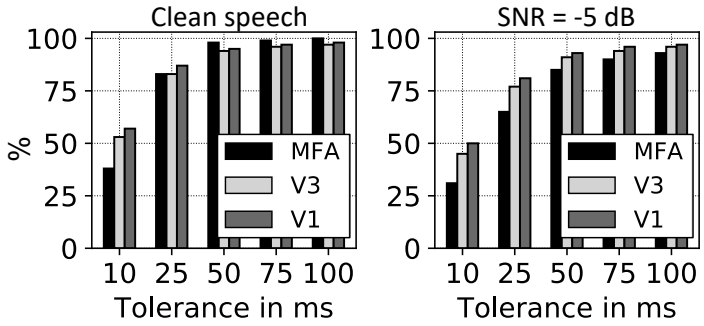


Figure 4.7: Percentage of correctly aligned phonemes with different tolerances. MFA: Montreal Forced Aligner, V1-3: Version 1-3.

We also compute the percentage of correctly aligned phonemes within a tolerance around the true onsets. The results are shown in Figure 4.7. They confirm that V1's and V3's alignment accuracy is not much affected by strong speech corruption while the MFA's accuracy decreases. Moreover, V1 and V3 estimate more than 50 % of all phoneme onsets with less than 10 ms error compared to the true onsets on clean speech. Even for the case of SNR = -5 dB, V1 aligns 50 % of the phonemes within the 10 ms tolerance.

Overall, the results show that there are mutual benefits when performing both tasks, informed separation and alignment, jointly. However, the gain in separation quality through phoneme sequences that are aligned jointly is quite small. The separation can be improved further if the alignment is done beforehand. In contrast, the alignment benefits substantially from the separation component when dealing with corrupted signals.

4.7 Conclusion

In this chapter, we introduced a deep learning model which exploits weak side information via attention for supervised audio source separation. It also provides an alignment of the side information with the audio mixture.

We evaluated the model in two sets of experiments. First, the concept of using attention between two encoders with a shared decoder was validated on a singing voice separation task with artificial side information such as vocal magnitude and vocal activity. The results showed that non-aligned side information can be exploited and aligned with the audio data using an attention mechanism when the model is trained only with a source separation objective. In the second set of experiments, text was used as side information to separate speech from music in low SNR mixtures. The text was represented as a sequence of phonemes. Joint phoneme alignment and speech separation led to benefits for both tasks. However, the separation is further improved if the text is aligned beforehand. Phonemes are accurately aligned even on highly corrupted speech signals.

It can be concluded that non-aligned text can be used with the proposed approach to inform audio source separation. In the next chapter we will further develop the approach in order to use it for text-informed singing voice separation.

Chapter 5

Text-Informed Singing Voice Separation and Lyrics Alignment

Summary

In this chapter the model introduced in the previous chapter is further improved with a new monotonic attention mechanism. It enables the model to perform phoneme-level lyrics alignment and text-informed singing voice separation. The result is a new data-efficient lyrics alignment method and new insights into the usage of text information for singing voice separation. Moreover, the musical source separation corpus MUSDB was extended by lyrics transcripts and other annotations. This chapter is based on the publication *Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation* [168].

Contents

5.1 Introduction

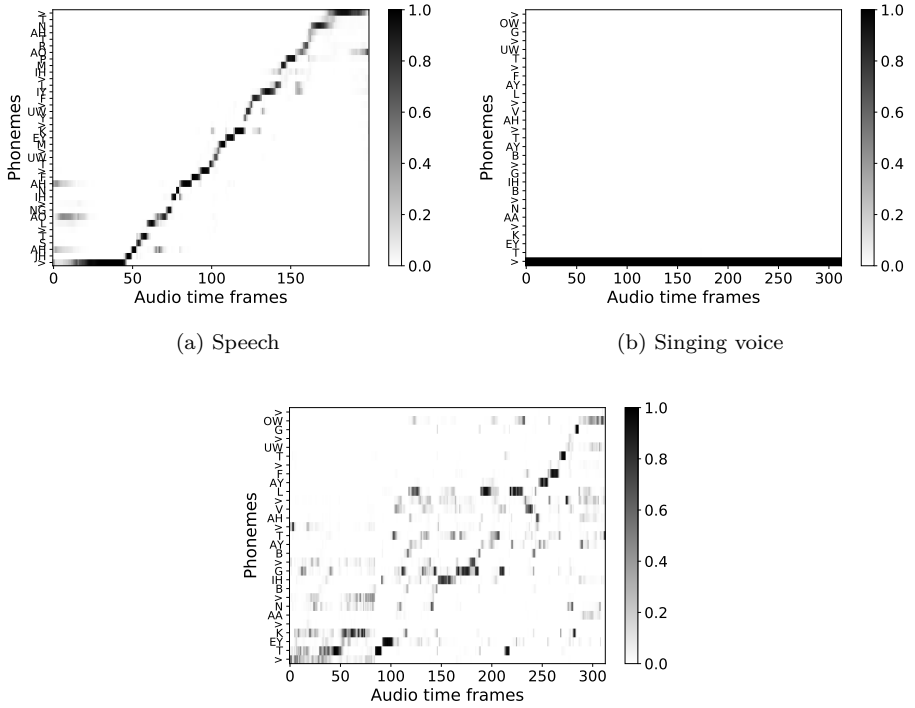
The goal in this chapter is to perform text-informed singing voice separation and phoneme level lyrics alignment building upon the approach introduced in the previous chapter. The motivation for using text as side information for deep learning based singing voice separation is that training data are scarce and expensive due to copyright restrictions. The question arises to which extend separation quality can be improved without access to more audio data by using complementary information such as text which is decomposed into phonemes (cf. Section 2.2.2).

We assume that phonemes need to be aligned with the observed mixture in order to inform the separation process. While great progress has been made regarding lyrics alignment at word level using resource intensive methods [182, 64], phoneme level alignment is rarely addressed although the methods in [182, 64] could be adapted to it. In fact, when phoneme alignment is required, they are often aligned manually [35, 56] or tools such as [122] are used [9, 100, 64] which employ acoustic models based on GMM-HMMs and do not work well on mixed singing voice as will be shown in Section 5.5.2. Aligning lyrics at phoneme level with audio mixtures has applications such as generating karaoke content, singing voice analysis, and creating training data for automatic lyrics transcription models.

Instead of adapting existing alignment methods to phoneme level alignment, we introduce a new approach to lyrics alignment building upon the previous chapter. Hence, the alignment will be performed jointly with text-informed singing voice separation. However, the model proposed in Chapter 4 was only evaluated on speech and does not work on singing voice without modifications which are introduced in this chapter. In Figure 5.1, the attention weights of the model of Chapter 4 are shown at test time for three different scenarios: a) trained and tested on speech-music mixtures, b) trained and tested on singing voice and accompaniment mixtures, and c) pre-trained on speech-music mixtures, then trained and tested on singing voice and accompaniment mixtures. While the correct monotonic alignment is learned for speech, the correspondence between singing voice and phonemes is not learned by the model. With pre-training on speech, some correspondence between text and audio is learned. However, the alignment is not monotonic as it should be because the phonemes are uttered in correct order from left to right. Hence, no alignment information can be derived from the attention weights and the usefulness of the text information for the separation is limited.

Therefore, in this chapter we also integrate prior knowledge in the alignment procedure. The knowledge concerns the fact that both text and audio sequences follow a logical and temporal structure from left to right. We include it by integrating Dynamic Time Warping (DTW) in the attention mechanism. This allows to obtain monotonic alignments. As additional modification, we use the state-of-the-art musical source separation model Open Unmix [186] as target source decoder in order to evaluate improvements of the separation quality through text.

As an alignment method, the model achieves competitive performance although much less training data are used than for state-of-the-art methods [64, 182]. However, the separation performance does not improve compared to non-informed methods using the joint separation and alignment ap-



(a) Speech

(b) Singing voice

(c) Pre-training on speech, then training on singing voice

Figure 5.1: Attention weights of the approach introduced in Chapter 4 for three cases: training and testing on (a) speech-music mixtures, (b) singing voice/accompaniment mixtures, and (c) pre-training on speech-music, then training and testing on singing voice mixtures

proach. Therefore, we also investigate a sequential approach where the phonemes are first aligned with our method and then fed as side information to a dedicated separation model. This leads to improvements through the text-information.

5.2 Related work

In this section, related work on lyrics alignment, monotonic attention, and informed source separation is reviewed.

5.2.1 Lyrics alignment

Most approaches to automatic lyrics alignment are based on acoustic models that estimate text unit (e.g. phoneme, character) probabilities given acoustic input features. When no large dataset of music recordings with corresponding lyrics was available, some acoustic models have been trained on speech and then adapted to singing voice [125, 100], but with limited success. Some works proposed to take additional information next to acoustic features into account such as chord labels [121] or phoneme durations inferred from a musical score [41]. These methods achieved good performance when aligning lyrics at phrase level on mixtures.

Recently, deep learning based approaches have exploited larger data resources for acoustic modeling on singing voice [182, 64]. They achieved high accuracy for word level alignment with

mean absolute alignment errors below one second on mixed singing voice. The method of Stoller et al. [182] learns an acoustic model on time domain signals to estimate character probabilities over audio frames. It is trained on 39,232 songs with line level aligned lyrics using the Connectionist Temporal Classification (CTC) loss [59]. The data intensive nature can be explained by the end-to-end approach and the fact that character sequences are ambiguous regarding the word pronunciation.

Gupta et al. [64] proposed to learn three genre-specific acoustic models for the broad classes pop, hip hop, and metal on mixtures. Genre-specific models for non-vocal segments are learned to improve the performance on long instrumental parts. It requires a training corpus with genre labels and enough data per genre class to train all acoustic models. In total, 3913 songs are used for training. Acoustic modeling and alignment are done using the open source speech recognition toolkit Kaldi [147] with a duration-based pronunciation lexicon for singing voice [61]. The performance seems to rely on a very large beam width during Viterbi decoding [63] as mentioned in the previous work [173] which is computationally expensive.

Instead of adapting the data intensive methods [182, 64] for phoneme alignment, we propose a novel alignment approach. The proposed model is actually trained for informed source separation and learns the acoustic model without direct supervision as a side effect. It has the potential to reduce the amount of required training data compared to [182, 64] because the task it solves during training is simpler. It has to *match* the observed phoneme sequence with the observed audio frames, whereas the other models need to *classify* observed audio frames into phonemes. However, multitrack data are required for training of the proposed method.

The Montreal Forced Aligner (MFA) [122] is a tool that can be used to learn GMM-HMM acoustic models and to align phonemes with audio signals. As initial alignment it assumes that all given phonemes belonging to a short audio example have the same length. On such an alignment a monophone GMM-HMM is trained while iteratively re-estimating the alignment. Then, triphone models are trained iteratively starting from the alignment provided by the monophone model. Speaker adaptation is performed as a last step if the speaker identities are known. The implementation is based on Kaldi [147]. Such a tool is commonly used to align phonemes with singing voice to prepare training data for other tasks [9, 100, 64]. Therefore, it will serve as one of the baselines for phoneme alignment.

5.2.2 Monotonic attention

In some cases, the alignment to be computed with an attention mechanism is known to be monotonic. Modifications to the attention mechanism have been proposed in the context of speech recognition [20, 150, 194] and machine translation [118, 194, 150] in order to enforce monotonic alignments which can help to disambiguate repeated elements in the sequences. We refer to such modified mechanisms as monotonic attention. One important difference between existing monotonic attention models and our model is that they consist of only one encoder and one decoder like the original attention model shown in Figure 2.2 (Page 18). Hence, the attention mechanism aligns the encoder output with hidden states of the decoder. The hidden states are computed autoregressively and cannot be observed all at once whereas we can observe both sequences to be aligned entirely because they are both inputs to the proposed model.

Chorowski et al. [20] proposed to consider the attention weights for the previous decoder time step in the scoring function for the current time step. This enables the model to learn a monotonic

alignment but does not enforce monotonicity explicitly. Luong et al. [118] and Tjandra et al. [194] use a sliding window over the encoder output sequences and only compute attention weights for elements within this window. They explore both shifting the window monotonically from left to right over the encoder output and learning to predict the window position for each decoder step.

Raffel et al. [150] proposed a monotonic attention mechanism for online scenarios where the input to the encoder is observed step-by-step. They define a stochastic process modeling the dependency of the matching decision on previous time steps. It provides a hard alignment at test time and the model is trained using soft alignments which reflect the expected outcome of this process.

The sliding window approach and the stochastic process in [150] make the alignment decision at a certain time step dependent on decisions at previous steps. An incorrect matching at some time step can therefore lead to many incorrect matches at subsequent steps. Our approach relaxes the dependence of attention weights across time steps during training. At test time, DTW finds a *globally* optimal alignment which considers all elements of both sequences. Moreover, in autoregressive models the computation of attention weights cannot be parallelized for the decoder time steps. DTW-attention allows for more parallel computations.

Another difference to the typical single encoder-decoder attention mechanism (cf. Section 2.3.5) is that in our model the information coming from the text is not essential (but potentially useful) in order to minimize the loss function, i.e. to learn the separation. Since the alignment is learned driven only by the separation objective, we observed that too strong constraints on the attention mechanism result in vanishing gradients for the text encoder and the attention mechanism so that no alignment is learned, while the separation is still learned. Therefore, the approaches proposed in [150] and [194] do not work in the context of this work. The proposed DTW-attention mechanism is able to learn the alignment while incorporating monotonicity constraints.

Cuturi et al. [27] proposed soft DTW which enables computing the DTW distance between two sequences with different lengths in a way that is differentiable and well-suited for gradient-based optimization. It allows using the DTW distance as a loss function but recovering the optimal *alignment path* is not possible. Therefore, soft DTW is not applicable in the context of this work and we propose DTW-attention to approximate the DTW *alignment path* in a differentiable way.

5.2.3 Informed audio source separation

Recently there has been an interest in including side information such as pitch [146, 85] or phonetic content [192, 13, 127, 87] in deep learning based separation in order to make it more robust in challenging scenarios. It has also been proposed to learn auxiliary tasks jointly, e.g. instrument activation detection [80] in order to cope with a larger number of musical sources to be separated. Most related to our work are four approaches that consider phonetic and linguistic information for singing voice separation.

Takahashi et al. [192] use deep features from an End-to-End Automatic Speech Recognition (E2EASR) model as side information for voice separation. The assumption is that the features contain phonetic *and* linguistic information because E2EASR combines acoustic and language modelling within one model. The side information leads to big improvements on speech separation in challenging conditions. The improvement for singing voice separation is considerably smaller. A possible reason is that the E2EASR model is trained on speech data and not adapted to singing voice.

Chandna et al. [13] train an encoder to extract content embeddings from mixtures. The target content embeddings are obtained with a speaker conversion method and contain phonetic information. From the embedding, a decoder estimates vocoder features which, along with a fundamental frequency estimate, are used to re-synthesise the voice signal from a mixture. The results show that the intelligibility of synthesized vocals is improved through phonetic features, but the overall subjective audio quality is lower than for filtering based separation methods.

An advantage of the approaches in [192, 13] is that no alignment method is required because phonetic information is extracted directly from the mixtures. On the other hand, the phonetic information is rather implicit and the mixture remains the only source of information. We consider explicit phoneme sequences from lyrics transcripts as additional model input that is independent from the mixture.

Two approaches to lyrics-informed singing voice separation have been developed in parallel to our work. In contrast to our work, they assume the availability of aligned lyrics.

Meseguer-Brocal and Peeters [127] use lyrics transcripts aligned at word level to condition singing voice separation using a U-Net [86]. Words are represented as bag of phonemes (without any temporal information at phoneme level) from which parameters are estimated to transform deep features in the U-Net encoder. Improvements over the classic U-Net are reported. However, it is not clear whether they are caused by the higher number of parameters in the conditioned U-Net, the voice activity information inherent in aligned text, or by the phonetic information. Since only word level alignment is available, the phonetic information of the text cannot be exploited entirely. Jeon et al. [87] condition singing voice separation on lyrics manually aligned at syllable level. They use a deep text encoder consisting of 1-D-convolutional highway layers [181]. The approach is evaluated on a private dataset of Korean amateur solo singing recordings mixed with unrelated accompaniments. To our understanding, only one singer sings at a time (no background singers, no multi-pitch singing). This facilitates learning the relation between phonemes and audio during training and the usage of text-information at test time. However, real commercial music recordings often contain multiple voices making the use of lyrics for separation less straightforward.

In contrast to [127] and [87], we address the lyrics alignment problem which allows us to use lyrics aligned at *phoneme* level. Furthermore, we provide extensive experimental evaluation using publicly available realistic mixtures with multiple singers and correlated accompaniments. We conduct a thorough analysis of the separation performance regarding the number of simultaneously present singers and phonemes and regarding the SNR of the voice-accompaniment mixtures.

5.3 Proposed method

Let $x(t) = v(t) + a(t)$ be a time domain single-channel mixture signal of singing voice $v(t)$ and instrumental accompaniment $a(t)$ where t refers to the discrete time index. In the following, $v(t)$ and $a(t)$ are assumed to be source images. Let $\mathbf{y} \in \{0,1\}^I$ be a one-hot vector representing one out of I considered phonemes and let $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M] \in \{0,1\}^{I \times M}$ be a matrix treated as a sequence of M one-hot vectors indexed by m representing the phonemes pronounced by the singing voice in the mixture.

The goal of text-informed singing voice separation is to separate $x(t)$ into $v(t)$ and $a(t)$ given $x(t)$ and $\mathbf{Y}$ as inputs. The goal of lyrics alignment is to estimate the onset time of each phoneme represented in $\mathbf{Y}$.

An overview of the proposed model is shown in Figure 5.2. The model is an improved version

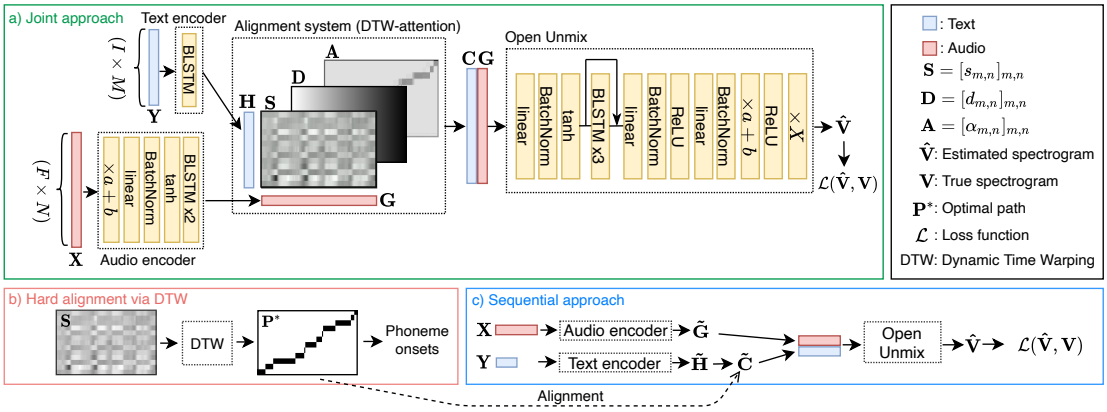


Figure 5.2: Overview of the proposed model. a) With the joint approach, alignment and separation are learned by optimizing the separation objective. b) At lyrics alignment test time, the phoneme onsets can be obtained from the score matrix via DTW. c) In the sequential approach, alignments are not learned but provided by some alignment method, e.g. the joint approach model.

of the one introduced in Chapter 4 and consists of four parts. A text encoder and an audio encoder which are detailed in section 5.3.1, an alignment system with a new monotonic attention mechanism explained in Section 5.3.2, and Open Unmix [186] as a separation model described in Section 5.3.3. Despite some overlap with the previous chapter, we briefly describe all parts in the following in order for this chapter to be self-contained. A PyTorch implementation of the model and our experiments is available online[1].

5.3.1 The encoders

The text encoder is a single Bidirectional Long Short-Term Memory (BLSTM) layer [75, 55]. It transforms $\mathbf{Y}$ into the hidden phoneme representation $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_M] \in \mathbb{R}^{R \times M}$ where R is the number of hidden features.

In the audio encoder, the Short Time Fourier Transform (STFT) of the mixture signal $x(t)$ is computed and we denote its magnitude $\mathbf{X} \in \mathbb{R}_{\geq 0}^{F \times N}$ where F is the number of frequency components and N is the number of time frames which are indexed by $n = 1, ..., N$. Each time frequency bin is scaled and shifted by learnable scalars which are initialized by the standard deviation and mean over the training data, respectively, as in the Open Unmix model [186]. The audio encoder transforms the input with a fully connected layer with *tanh* activation followed by two BLSTM layers into the audio representation $\mathbf{G} = [\mathbf{g}_1, ..., \mathbf{g}_N] \in \mathbb{R}^{S \times N}$ where S is the number of hidden features.

5.3.2 The alignment system

The alignment system learns to align the vector sequences $\mathbf{H}$ and $\mathbf{G}$. An alignment can be formalized as a path which we denote as sequence $\mathbf{P} = (\mathbf{p}_1, ..., \mathbf{p}_L)$ of length L where $\mathbf{p}_l$ are tuples with $\mathbf{p}_l = (m_l, n_l) \in [1 : M] \times [1 : N]$. The path satisfies the following conditions [134]:

$$\mathbf{p}_1 = (1, 1) \quad \text{and} \quad \mathbf{p}_L = (M, N) \tag{5.1}$$

[1] https://github.com/schufo/plla-tisvs

$$\mathbf{p}_{l+1} - \mathbf{p}_l \in \{(0,1),(1,1)\}. \tag{5.2}$$

Each path tuple $\mathbf{p}_l$ matches one phoneme with one audio frame. The step size condition in (5.2) is chosen so that each audio frame is matched with exactly one phoneme, whereas the same phoneme can be assigned to several audio frames. It follows that $L = N$. The condition also implies that the alignment path is monotonic and continuous, i.e. we assume that the phonemes are pronounced in the given order and no phoneme is skipped. The goal is to find the path that provides the correct matching between audio and text.

When phonemes are to be aligned with speech, a standard attention mechanism can learn such a monotonic alignment [167]. However, when working on singing voice, we found it to be crucial to enforce monotonicity explicitly in order to learn an alignment. This is probably due to the wider range of possible acoustic realisations of phonemes in singing due to a wider pitch range and artistic expressiveness. Therefore, we propose DTW-attention, a combination of DTW and attention to obtain monotonic alignments.

First, we compute a pair-wise matching score $s_{m,n}$ between all elements of the sequences $\mathbf{G}$ and $\mathbf{H}$ as

$$s_{m,n} = \mathbf{g}_n^\top \mathbf{W} \mathbf{h}_m \tag{5.3}$$

with the learned weight matrix $\mathbf{W} \in \mathbb{R}^{S \times R}$ as typically done in attention mechanisms [118]. It evaluates how likely it is that the m-th phoneme is pronounced in the n-th audio frame regardless of the position of $\mathbf{g}_n$ and $\mathbf{h}_m$ in their respective sequence.

Then, we incorporate the conditions (5.1) and (5.2) by computing the accumulated score matrix $\mathbf{D} = [d_{m,n}]_{m,n} \in \mathbb{R}^{M \times N}$ as typically done in DTW as follows [134, 160]:

$$d_{m,n} = s_{m,n} + \max(d_{m,n-1}, d_{m-1,n-1}) \tag{5.4}$$

with

$$d_{0,0} = b \quad \text{and} \quad d_{0,n} = d_{m,0} = -\infty \quad \forall m, n > 0 \tag{5.5}$$

where b is a sufficiently large number. Note that in (5.4) the objective is to maximize the accumulated *score*, whereas classical DTW usually minimizes a *distance* [160]. The reason for this is that stronger similarity between a phoneme and an audio frame results in a *higher* score $s_{m,n}$ while it would produce a *lower* distance value. The value $d_{m,n}$ is the accumulated score of the optimal alignment path starting at $(1,1)$ and ending in (m,n) respecting the step size condition (5.2). The optimal path in the DTW sense is the one with the highest accumulated score. The DTW step in (5.4) helps disambiguate identical phonemes appearing several times in the sequence, which could have the same score at a given time frame, by explicitly taking their order into account. It can be implemented efficiently by parallelizing computations of entries on the anti-diagonal of $\mathbf{D}$ or those lying on a line parallel to it because they are mutually independent.

Using classical DTW, the actual optimal path could now be found by path backtracking [134]. However, such hard alignment, where one audio frame is matched with exactly one phoneme, is not differentiable [27, 150] and thus not applicable in a deep learning model during training. Instead, we will use a soft alignment strategy during training. When phoneme onsets are to be retrieved at test time, we are able to compute $\mathbf{P}$ using the scores $s_{m,n}$ and classical DTW to obtain hard alignments.

Soft alignment during training

We compute attention weights α by a column-wise softmax operation on $\mathbf{D}$ as typically done in attention mechanisms [6]:

$$\alpha_{m,n} = \frac{e^{d_{m,n}}}{\sum_{k=1}^{M} e^{d_{k,n}}}. \tag{5.6}$$

The M attention weights corresponding to audio frame n can be interpreted as a probability distribution over all phonemes for this time frame and hence provide a soft alignment. The phoneme with the highest accumulated score in frame n has the highest probability α. This is a local approximation of the globally optimal path that would be obtained by DTW. It assumes that the phoneme with the highest accumulated score at frame n will be part of the optimal path. As we explain in Section 5.6.2, this is true for 84% of the frames in our test set. Equations (5.4) and (5.6) put a soft constraint on the attention weights to be monotonic, i.e. respecting (5.2). It is *soft* because the dependence between time frames is reflected only in (5.4) whereas the attention weights are computed for each frame independently in (5.6). This is in contrast to other methods for monotonic attention, which we reviewed in Section 5.2.2, and avoids error propagation from previous frames at the cost that there is no guarantee for strict monotonic paths during training. We found this trade-off to be appropriate in order for the model to learn the correspondence between phonemes and spectrogram frames of (mixed) singing voice. It also allows for efficient parallel computation of attention weights. The attention mechanism does not require training data with aligned phonemes. However, if such data were available they could be exploited trough a supervised loss term on the scores or attention weights.

The text information corresponding to an audio frame is then computed as

$$\mathbf{c}_n = \sum_{m=1}^{M} \mathbf{h}_m \alpha_{m,n} \tag{5.7}$$

and a new text sequence $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_N] \in \mathbb{R}^{R \times N}$ which has the same length N as the audio sequence $\mathbf{G}$ is obtained. Finally, $\mathbf{C}$ and $\mathbf{G}$ are concatenated along the feature dimension and this combined text and audio representation is then processed further by the separation model as explained in Section 5.3.3.

Hard alignment at test time for lyrics alignment

Once the model is trained, the scores $s_{m,n}$ can be used as a similarity measure between a given phoneme sequence and the spectrogram frames. A globally optimal alignment $\mathbf{P}^*$ can then be found by DTW which consists of (5.4) and path backtracking [134]. The path $\mathbf{P}^*$ is a hard alignment as it assigns exactly one phoneme to each audio frame. While a hard alignment is required to infer phoneme onsets at test time, the soft alignment provided by (5.4) and (5.7) can be used to inform the separation model at test time in order to have the same behaviour as during training. The estimated phoneme onset is the start time of the first frame it has been assigned to. An example of the scores and a DTW path is shown in Figure 5.3.

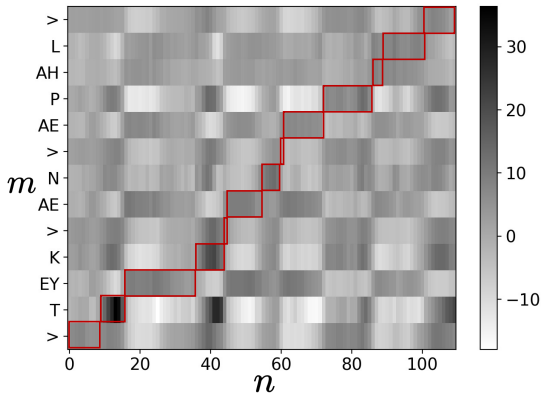


Figure 5.3: Example of a score matrix $\mathbf{S} = [s_{m,n}]_{m,n}$ with optimal DTW path in red which assigns one phoneme to each audio frame.

5.3.3 The separation model

This part consists of the source separation model Open Unmix [186]. The input is the combined text and audio representation (cf. 5.3.2) from which the estimate $\hat{\mathbf{V}} \in \mathbb{R}_{\geq 0}^{F \times N}$ of the singing voice's magnitude spectrogram is computed. The model comprises a fully connected layer with tanh activation, three layers of BLSTM with a skip connection, and two fully connected layers with ReLU activation. The output is multiplied with the mixture magnitude spectrogram $\mathbf{X}$ and yields $\hat{\mathbf{V}}$. The architecture details are visualized in Figure 5.2. In order to obtain the vocals estimate in the time domain, $\hat{\mathbf{V}}$ is combined with the mixture phase and an inverse STFT is applied. In this study, we do not consider additional models to estimate the other sources because the focus is on the effect of text-information for the vocals estimate.

The estimated vocals magnitude $\sum_{f=1}^{F} \hat{v}_{f,n}$ for each time frame n can be used as a Voice Activity Detector (VAD). If the magnitude is below a threshold, it can be assumed that no voice is active in the given frame. At test time, the scores of phoneme tokens that represent silence between words (cf. Section 5.4.2) can be set to a high value for such frames before applying DTW. This reduces the probability that phonemes are assigned to frames without vocals which can happen especially on long instrumental parts.

5.3.4 Joint vs. sequential approach

The model described above performs separation and alignment jointly. However, it can be beneficial for the separation quality to perform these tasks sequentially. For a sequential approach, two different, specialized versions of the model are employed. The first one (alignment model) corresponds exactly to the model described above. It is responsible for the alignment, which is learned through the separation objective as described. It is trained first and provides the hard alignment paths $\mathbf{P}^*$ for the second version (separation model) which is responsible for the separation and does not have an alignment system. We denote representations in the separation model with a tilde $\tilde{}$. The aligned text representation $\tilde{\mathbf{C}}$ is obtained by assigning an element of $\tilde{\mathbf{H}}$ to each audio frame using $\mathbf{P}^*$ (cf. Figure 5.2 and 5.3). During training and testing of the separation model, the text and audio sequences are fed to both the encoders of the alignment model and the encoders of the separation model (cf. Figure 5.2). The encoders of the separation model can learn representations $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{G}}$ dedicated exclusively to the separation task. In contrast, the representations in the

alignment model and the model for a joint approach have to enable the alignment as well.

5.4 Data annotation and training details

In order to obtain training and testing data for text-informed singing voice separation, we annotated the most popular singing voice separation dataset, MUSDB [151], with line level aligned lyrics and additional information about the vocals as explained in Section 5.4.1. We detail the training data and procedure in Section 5.4.2 and a study on pre-training and attention is presented in Section 5.4.3.

5.4.1 Annotations of the MUSDB corpus

The dataset comprises 150 songs and is split into a training partition with 100 songs, of which 96 have English lyrics, and a test partition with 50 songs, of which 45 have English lyrics. We transcribed the English lyrics manually by listening to the isolated vocals.

The songs were divided into sections of lengths between 3 and 12 seconds. The priority when choosing the section boundaries was that they correspond to natural pauses and do not cut vocal sounds. Most of the sections do not overlap, some have an overlap of one second. For each section, we annotated the start and end time, the corresponding lyrics as well as a label indicating one of the following four properties:

(a) only one person is singing,

(b) several singers are pronouncing the same phonemes at the same time (possibly singing different notes),

(c) several singers are pronouncing different phonemes simultaneously (possibly singing different notes),

(d) no singing voice.

Differentiating between singing voice examples with these properties allows for a more thorough analysis of the separation results and one could exclude certain segments from the training set, if desired. Segments that are labeled with the property (b) or (c) do not necessarily have this property over the whole segment duration. As soon as somewhere in a segment several singers are present, label (b) was assigned; as soon as they sung different phonemes somewhere at the same time, label (c) was assigned. Property (a) and (d) are valid for the entire segment. Furthermore, segments with property (c) can contain either some (lead) singer(s) singing some words in the presence of background singers singing long vowels such as 'ah' or 'oh' or they can contain multiple singers who sing different words at the same time. In the latter case, it was very difficult to understand the lyrics and to decide in which order to transcribe words or phrases sung simultaneously. We marked these segments and excluded them from our training and test data. In some difficult cases, e.g. shouting in metal songs or mumbled words, where the words are barely intelligible, we made an effort to make the transcriptions as accurate as possible phonetically and did not prioritize semantically meaningful phrases.

We believe that these annotations are a valuable resource for research on several tasks such as automatic lyrics alignment and transcription, text-informed singing voice separation, and singing

voice analysis. Therefore, we make them publicly available[2].

5.4.2 Training details

We use 82 songs (2289 segments with total length of 4.6 hours) of the annotated MUSDB training set for training. The remaining 14 songs are used as a validation set (487 segments with 0.94 hours total length) for early stopping. The audio signals were downsampled to 16 kHz. As for the original Open Unmix model [186], training is done on short segments to prevent learning difficulties with backpropagation through time [211]. This does not prevent the model to process longer sequences at test time. Preliminary experiments (cf. Section 5.4.3) showed that the attention mechanism requires pre-training with mixtures containing speech signals. We found that pre-training on speech-music mixtures for 66 epochs enables subsequent training on singing voice plus accompaniment mixtures. We use speech recordings sampled at 16 kHz and word level text transcripts from the TIMIT database [53]. The speech is mixed with instrumental music retrieved from Youtube with a SNR uniformly drawn from $[-8,\ 0]$ dB. In total, the speech set consists of 4320 mixtures, which are between 2 and 8 seconds long and have a total length of 4.9 hours.

All words in the transcripts are translated into phonemes using the CMU LOGIOS Lexicon Tool[3]. Hence, there is no guarantee that the phonetic transcription always reflects the actual word pronunciation in the recordings accurately. We add a space token between each word that represents potential silence in the vocals. Examples without vocals are annotated with only the space token as lyrics.

The model is trained with the objective to minimize the L1 distance between the estimated and true vocals magnitude spectrogram, $\hat{\mathbf{V}}$ and $\mathbf{V}$ respectively. The Adam optimizer [90], a learning rate of 0.001 and a batch size of 16 are used. A STFT with a Hann window of length 512 samples (32 ms) and a hop size of 256 samples (16 ms) is applied to compute the spectrograms. The learning rate is multiplied by 0.3 after 80 consecutive epochs without improvement of the validation loss and training is stopped after 140 consecutive epochs without improvement. Following the Open Unmix procedure, additive mixtures are produced for training by sampling the stems bass, drums, and others (as defined by MUSDB) randomly from different tracks, scaling them by a factor randomly drawn from $[0.25,\ 1.25]$ and adding them to a vocals segment scaled by a factor drawn from $[0.25,\ 0.9]$.

5.4.3 Study on pre-training and attention

In order to illustrate the effect of pre-training on speech-music mixtures, we train the proposed model with and without pre-training. To test the effectiveness of the proposed attention mechanism, we also train the model with a conventional attention mechanism [118] (applying the softmax operation in (5.6) on the scores $\mathbf{S} = [s_{m,n}]_{m,n}$ instead of the accumulated scores $\mathbf{D}$) for comparison. The resulting attention weights matrices for the four studied scenarios are shown in Figure 5.4.

Without pre-training on speech, neither of the attention mechanisms learn an alignment for singing voice. With pre-training, both attention mechanisms learn some correspondence between audio and text, but only the proposed mechanism provides a sharp and nearly monotonic alignment. We can look at the differences between the speech and singing voice data used for training in order to understand why the attention mechanism initially requires speech data. The speech-music

[2] `https://doi.org/10.5281/zenodo.3989267`
[3] `http://www.speech.cs.cmu.edu/tools/lextool.html`

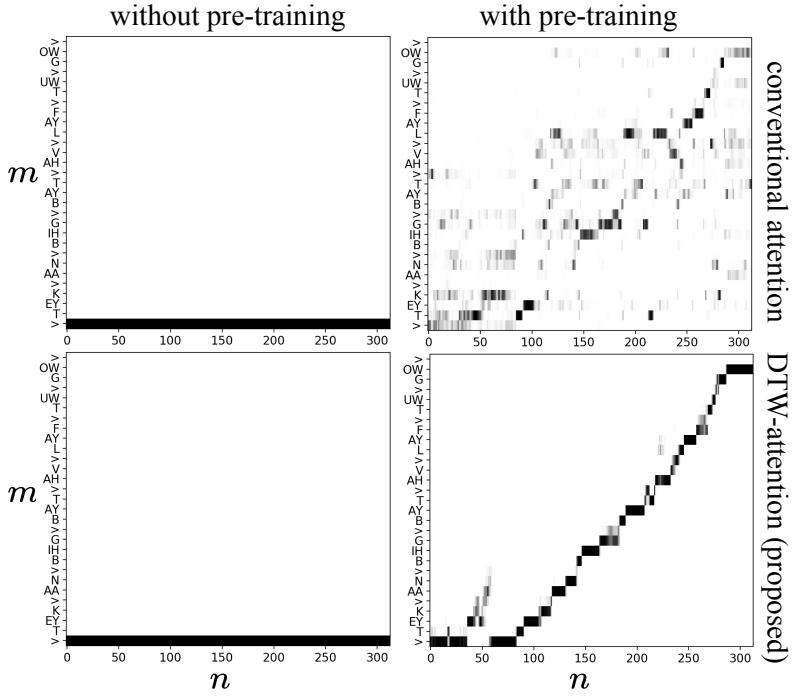


Figure 5.4: Attention weight matrices $\mathbf{A} = [\alpha_{m,n}]_{m,n}$ for four different scenarios. Darker colors represent higher values, all values are in $[0, 1]$.

mixtures have more accurate text transcripts, a lower SNR (making the task more difficult and thus the side information more valuable), a smaller voice pitch range, and more phonemes are uttered in a given time interval compared to singing voice. Also, word pronunciations are altered in singing voice for artistic reasons. We conducted various additional experiments with lower SNRs in the training examples using both the MUSDB data and singing voice recordings with accurate phoneme transcriptions [35]. In none of the settings did the attention mechanism train as desired. Therefore, the pitch range, phoneme rate, and uniform pronunciation in speech are likely to be the factors that enable the proper training in the considered limited data setting. A possible reason for the sensitivity to initialization is that the separation task can be learned by the model even without learning the alignment as discussed in the end of Section 5.2.2.

The advantage of computing the attention weights for each audio frame independently from the other frames while still encouraging monotonicity can be seen in the bottom right plot of Figure 5.4: Although some phonemes are wrongly assigned to some early frames without singing, this mistake does not impede the correct monotonic alignment at later frames.

5.5 Evaluation of lyrics alignment

We explain the experimental design for phoneme and word level lyrics alignment in Section 5.5.1 and present and discuss the results in Section 5.5.2.

5.5.1 Experimental design

Each test song is processed in full length at once by the model, so that no segmentation of audio and text is required, i.e. DTW is done on the score matrix $\mathbf{S} = [s_{m,n}]_{m,n}$ for the whole song.

Phoneme level alignment

We use the NUS-48E Sung and Spoken Lyrics Corpus [35] to assess phoneme level lyrics alignment. It is a collection of 48 solo singing recordings[4] of length between 53 seconds and 3.5 minutes with manually transcribed phonemes and their time stamps. 12 amateur singers sing 4 English songs each, the set comprises 20 unique songs. In order to evaluate phoneme alignment on mixtures, we mix each singing recording with a different instrumental accompaniment of a song of the MUSDB test set.

We train the proposed model as explained in Section 5.4.2 and call it JOINT1. Then, we test if some modifications regarding the training data can improve phoneme alignment. Since pre-training on speech data enabled learning the correspondence between phonemes and audio, speech data might also be beneficial when continuing training on singing voice. Therefore, we add 1000 speech-music mixtures to the MUSDB training data and call the model trained this way JOINT2. For training of the next model, we also add silence to the MUSDB vocals segments before mixing them with the other stems which results in longer instrumental sections in the training examples. This increases the amount of audio frames that correspond to the space token and potentially helps learning a better acoustic model for non-vocal frames. The idea is inspired by Gupta et al. [64] who identified acoustic modeling of non-vocal frames as a crucial aspect of automatic lyrics alignment. Specifically, each vocals signal is zero-padded to length 11 seconds. Padding is done for 50% of the signals at the start and for 50% at the end. The model trained with added silence and added speech is called JOINT3. We also train a model only on speech-music mixtures for comparison. It is called JOINT-SP.

Thereafter, we compare the best performing model from the study above to two baselines using both solo singing and mixtures as audio signals. The first one is the Montreal Forced Aligner (MFA) [122] (cf. 5.2.1) which is a GMM-HMM. The MFA performs acoustic modeling and alignment iteratively and processes the training and test data combined. It is informed by the singer identity of the test songs and performs speaker adaptation. The second baseline is a deep learning model trained with the CTC loss [59]. It consists of three BLSTM layers with 256 hidden units followed by a linear layer mapping to the output size of 44 units (number of phonemes plus CTC's blank token). This architecture is inspired by the work in [198]. After a comprehensive hyperparameter search, we found that the best performance on solo singing is obtained using 13 MFCCs (frame sise 256, 50% overlap) plus their deltas as input features. On mixtures it was best to use Mel-spectrograms (frame sise 512, 50% overlap) with deltas and delta-deltas as inputs. We call these versions CTC-MFCC and CTC-MEL, respectively. The model is trained with batch size 1 and a learning rate of 0.001. Both baselines are trained on our MUSDB training set. They are trained on mixtures for the evaluation on mixtures and on the clean vocals stems for the solo singing evaluation. Pre-training or including speech data or adding extra silence did not improve their performance.

Word level alignment

We evaluate word level lyrics alignment on the Hansen [66] and the Jamendo lyrics [182] dataset. They are widely used for word alignment evaluation on mixtures and comprise 10 and 20 western pop songs in English language, respectively. Also, they have been used in the Music Information

[4] We excluded song 09 of singer ADIZ due to incorrect annotations

Retrieval Evaluation eXchange (MIREX) 2019 lyrics alignment task[5], facilitating comparison between the proposed method and the two best performing methods which were proposed by Gupta et al. (GU) [64] and Stoller et al. (ST) [182] reviewed in section 5.2.1.

While word alignment can be considered as less difficult than phoneme alignment because it is coarser, these two datasets are more challenging than the one we have at our disposal for phoneme alignment evaluation. The reasons are that the accompaniment is correlated with the voice, they contain longer instrumental sections such as intros or solos, and the transcripts are partly incomplete as some vocal sounds such as 'ah' or 'oh' are sometimes neglected. Therefore, we also test using the vocals estimate $\hat{\mathbf{V}}$ as a VAD: when the estimated total vocal magnitude is lower than 20 for a time frame, it is assumed that it is a non-vocal frame and the score s of all space tokens is set to the maximum score obtained for the given song. This method is called JOINT3-VAD. The threshold was selected empirically on the MUSDB test set by visual inspection of the vocals magnitude for some examples. However, the alignment results have a marginal sensitivity regarding the exact threshold value as will be shown in Section 5.5.2 (cf. Figure 5.6).

5.5.2 Results and discussion

Phoneme level alignment

The results of the experiment on training data are shown in Table 5.1. The evaluation metrics are the mean and median Absolute Error (AE), which is the absolute difference between the true and estimated onset averaged over all phonemes of a test song, and the Percentage of Correctly Aligned Segments (PCAS) [52]. In this context, segments are the signal parts between onset time stamps and each segment is labeled with one phoneme. The PCAS measures the percentage of overlap of ground truth and estimated segments over the whole song. The AE compares onsets which are point estimates and does not take the phoneme duration into account whereas the PCAS tells which percentage of the audio signals is labeled with the correct phoneme. This is especially critical when the alignment is used for other downstream tasks such as informed separation in our case. Adding speech examples (+sp.) improves all evaluation metrics. There is less variance in the acoustic realisation of a phoneme in speech signals than in singing, which facilitates learning the relation between audio and phoneme labels statistically. Adding silence (+sil.) reduces the mean AE more than the median AE, and slightly improves the PCAS. As observed in [64], it helps recognizing non-vocal frames and makes the alignment more robust. Training only on speech-music mixtures (JOINT-SP) does not allow to align phonemes on singing voice. As a result of this study, we use the model JOINT3 for comparison with other methods on phoneme and word level alignment.

In Table 5.2, JOINT3 is compared to the baselines MFA, CTC-MFCC, and CTC-MEL. The proposed method outperforms the baselines on solo and mixed singing voice. Note that the baselines have been trained on mixtures for the evaluation on mixtures (cf. 5.5.1). The fact that JOINT3 works well also on mixed singing, even with low SNRs shows the effectiveness of the voice separation component inherent in our alignment approach. In practice, the baselines could be used with voice separation as pre-processing step. However, it is likely that performance is worse than on solo singing. The PCAS of the proposed approach is above 80 % for SNRs of 0dB and higher. This makes it a suitable method to produce phoneme alignments for datasets on which models for other tasks are trained.

[5] https://www.music-ir.org/mirex/wiki/2019:Automatic_Lyrics-to-Audio_Alignment_Results

Method	Training data	mean AE [s]	median AE [s]	PCAS [%]	SNR [dB]
JOINT-SP	sp.	27.9382	26.5665	1.76	solo singing
JOINT1	MUSDB	0.0884	0.0158	81.49	
JOINT2	MUSDB+sp.	0.0611	**0.0149**	85.91	
JOINT3	MUSDB+sp.+sil.	**0.0573**	**0.0149**	**85.94**	
JOINT-SP	sp.	26.0748	23.7819	3.62	5
JOINT1	MUSDB	0.1122	0.0173	79.13	
JOINT2	MUSDB+sp.	0.0638	0.0160	84.41	
JOINT3	MUSDB+sp.+sil.	**0.0631**	**0.0158**	**84.66**	
JOINT-SP	sp.	33.4086	30.8661	0.97	-5
JOINT1	MUSDB	0.2639	0.0360	68.91	
JOINT2	MUSDB+sp.	0.1634	0.0254	75.38	
JOINT3	MUSDB+sp.+sil.	**0.1425**	**0.0247**	**76.02**	

Table 5.1: Phoneme alignment results on NUS-48E corpus. Values are the mean over the test set. AE=Absolute Error, PCAS=Percentage of Correctly Aligned Segments.

Method	mean AE [s]	median AE [s]	PCAS [%]	SNR [dB]
JOINT3	**0.057**	**0.015**	**85.94**	solo singing
MFA	0.073	0.030	77.94	
CTC-MFCC	0.071	0.034	76.49	
JOINT3	**0.063**	**0.016**	**84.66**	5
MFA	1.468	1.089	46.92	
CTC-MEL	0.198	0.078	57.61	
JOINT3	**0.077**	**0.018**	**82.17**	0
MFA	4.523	3.756	25.61	
CTC-MEL	0.513	0.267	46.94	
JOINT3	**0.143**	**0.025**	**76.21**	-5
MFA	7.079	6.172	10.03	
CTC-MEL	1.590	1.087	30.58	

Table 5.2: Phoneme alignment results on NUS-48E corpus. Values are the mean over the test set. AE=Absolute Error, PCAS=Percentage of Correctly Aligned Segments.

Comparing CTC-MFCC and JOINT3 shows that DTW-attention is more efficient in this limited data setting than CTC training to learn an acoustic model for alignment. This may be surprising because DTW-attention performs only a DTW forward pass (followed by finding the locally, frame-wise optimal solution as approximation of the globally optimal path) during training. In contrast, the CTC loss implements a forward-backward algorithm to find *all* alignment paths [59]. Therefore, it is likely that the CTC loss can find optimal alignment paths more often than DTW-attention during training.

However, at training time the goal is not to find optimal alignment paths. The goal is rather to encourage the acoustic model to produce posteriorgrams or score matrices which are accurate with respect to frame/label alignments. From a theoretical perspective, the CTC loss does not have an advantage over the proposed DTW-attention regarding this training goal. There is an intuitive explanation why DTW-attention can learn score matrices with better time-synchronization using gradients which are backpropagated from the separation network.

The objective of the CTC loss is to maximize the likelihood of the target label sequence (phonemes in our case) given acoustic input features. In order to compute the likelihood of the

target label sequence, it marginalizes over *all* possible alignments [59, 65] which are found using the forward-backward algorithm. Therefore, the alignment that provides the correct frame/label synchronization is not preferred over other alignments which output label probabilities delayed (or too early if bidirectional RNNs are used). The internal memory enables the model to remember acoustic states and output the corresponding phoneme probabilities at arbitrary frames. The CTC loss can be minimized as long as the order of the phonemes is correct. Thus, the CTC loss does not maximize the likelihood of any particular alignment. This issue has also been discussed by Sak et al. [159]. It is probably also the reason why Stoller et al. [182] add a constant delay of 180 ms to the alignments of their CTC based model.

On the other hand, in the proposed method there is a strong incentive to learn the correct frame/label synchronization: if the synchronization is bad, the information coming from the phoneme labels is not useful for the separation network. Therefore, the gradients from the separation objective are a strong learning signal for DTW-attention leading to accurate alignments that can be superior to the ones obtained with the CTC loss in the limited data setting considered in the proposed work.

Word level alignment

The word alignment results on the Hansen (H) and Jamendo (J) dataset are shown in Table 5.3. The metrics are the mean and median Absolute Error (AE) (explained in 5.5.2) and the percentage of correctly aligned words within a tolerance of 0.3 seconds.

Method	Songs for training	mean AE [s]		median AE [s]		% within 0.3s	
		H	J	H	J	H	J
ST [182] (SV)	39232	0.39	0.38	0.09	0.10	88	87
ST [182] (MV)	39232	-	0.82	-	0.10	-	85
GU [64]	3913	**0.10**	**0.22**	**0.04**	**0.05**	**97**	**94**
JOINT3	82*	1.47	1.86	0.06	0.10	83	80
JOINT3-VAD	82*	0.79	0.88	0.06	0.08	85	81

*plus 4.9 hours of speech music mixes (equals the length of 98 songs of 3 minutes)

Table 5.3: Word alignment results on the Hansen (H) [66] and Jamendo (J) [182] dataset. Values are the mean over test songs.

A boxplot of the AEs on the Jamendo dataset is shown in Figure 5.5. The mean and median values in the boxplot are taken over all AEs on the whole test set while the values in Table 5.3 are taken per song and are then averaged over all songs following the procedure of MIREX.

Using the VAD reduces the mean AE and barely influences the median AE and overall error distribution. It can be seen in Figure 5.5 that the VAD decreases the largest errors. This happens because using VAD reduces the number of phonemes that are wrongly assigned to frames of long instrumental parts. In Figure 5.6, boxplots of the absolute alignment errors are shown for JOINT3-VAD for different thresholds in the VAD to decide weather a frame contains singing voice or not. It can be seen that the exact threshold value affects the results only marginally.

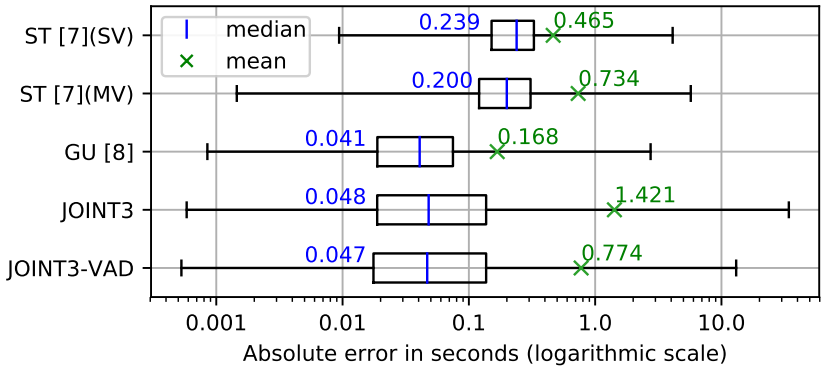


Figure 5.5: Boxplot of the absolute alignment errors on the Jamendo dataset [182]. The boxes extend from the first to the third quartile. The whiskers extend from the first to the 99th percentile.

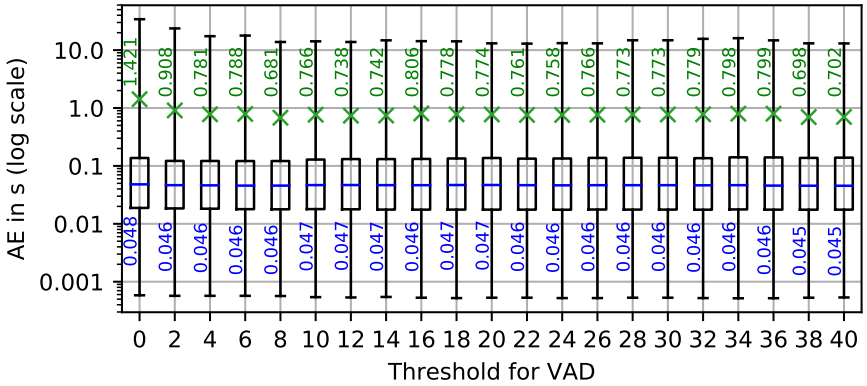


Figure 5.6: Boxplot of the absolute alignment errors on the Jamendo dataset [182] for method JOINT3-VAD for different VAD thresholds. The boxes extend from the first to the third quartile. The whiskers extend from the first to the 99th percentile. The medians are shown as blue lines, the means are shown as green 'x'.

The baseline ST [182] has been evaluated for training and testing on Separated Vocals (SV) and Mixed Vocals (MV) by its authors. In Table 5.3 it can be seen that the baselines used considerably more training data than the proposed method. They have a lower mean AE than our method while the median AE is roughly the same. Figure 5.5 shows that the overall error distribution of the proposed method is very similar to the state-of-the-art method GU [64] with the difference that some larger errors are produced which increase the mean AE. We observed that those outliers occur due to two reasons. Firstly, our method cannot cope well with vocal sounds that are not included in the given lyrics transcript because any vocal sound in an audio frame makes the model assign a higher score to the phonemes than to the space token (cf. equation (5.3)). This can result in assigning the first phoneme of the word after a non-transcribed sound to the frame of this non-transcribed sound and hence to predicting the onset too early. Secondly, the VAD does not capture all non-vocal segments perfectly and the model might confuse similar sounding instruments with vocals and assign high scores to phonemes instead of silence, which influences the DTW path. The baseline models learn more advanced acoustic models (triphone and genre-specific [64] or character level [182]) on more training data than our method. We think that this is the reason why they are more robust to those failure modes. They were the first to produce mean AEs below one second in the MIREX lyrics alignment task on mixtures. Considering the error boxplots in Figure 5.5,

the proposed method can be seen as a less data intensive alternative to the baselines. This is especially interesting for alignment of lyrics in other languages than English for which training data are scarcer.

To conclude the alignment evaluation, it can be said that the proposed method is able to align phonemes accurately on mixed singing voice when accurate transcripts are provided. Performance decreases when challenges such as long instrumental parts or inaccurate transcripts are faced, but performance is not far from the state-of-the-art on word level alignment in this case while less training data are used. DTW-attention trained with the separation objective yields better alignments than CTC training in the considered limit data setting.

5.6 Evaluation of singing voice separation

We explain the experimental design in Section 5.6.1 and present and discuss the results in Section 5.6.2.

5.6.1 Experimental design

We use the 45 songs of the MUSDB [151] test set that are in English language along with their text transcripts for the separation evaluation. In total, our test set comprises 1461 segments with a total length of 2.9 h. The audio signals were downsampled to 16 kHz.

Open Unmix reference and joint approach

As a reference, we train the original Open Unmix model [186] on our MUSDB training data and call it UMX1. We also train it with the exact same training data and procedure as the best alignment model, JOINT3, i.e. pre-training on speech, adding silence to vocals, and adding speech data when training on singing voice (cf. Section 5.5.1), and call it UMX2. In order to evaluate the joint alignment and separation approach, we evaluate JOINT3 which was the version with the best alignment performance as shown in Section 5.5.2.

Sequential approach

For the sequential approach (cf. section 5.3.4 and Figure 5.2), we use JOINT3 as the alignment model, providing alignments for a dedicated text-informed separation model which we call SEQ. Two baselines (BL) are provided. They use the exact same model as SEQ but, instead of one-hot vectors representing phonemes, they get different side information. For SEQ-BL1, every element in $\mathbf{Y}$ is the same one-hot vector and the given alignment path assigns the last element of $\mathbf{H}$ to all audio frames, i.e. $\mathbf{p}_n = (M, n) \forall n$. This means that no information about the singing voice is provided to SEQ-BL1. The second baseline, SEQ-BL2, receives the alignments provided by JOINT3 but all phonemes are represented with the same one-hot vector and the space token (silence) is represented with a different one-hot vector. This means the information of aligned phonemes is reduced to voice activity information for SEQ-BL2. Since the two baselines have the exact same architecture and number of parameters as SEQ, the effect of text as a side information can be evaluated.

Evaluation on mixtures with fixed SNR

For the experiments above, all models are evaluated with the original mixtures of the MUSDB dataset. Beyond that, we evaluate some models again and, this time, we mix the voice and accompaniment with a fixed SNR of 0, -5, and -10 dB. The SNR is computed on each test segment individually. This experiment allows us to investigate the effect of text as a side information on mixtures with different degrees of difficulty for singing voice separation. As reported in [113], lower SNRs usually decrease the separation quality.

5.6.2 Results and discussion

In Table 5.4, the separation evaluation scores are presented. The metrics SDR, SIR, and SAR [202] are computed on one second long non-overlapping evaluation frames using *museval* with *BSSEval* v4 [188] following the Signal Separation Evaluation Campaign [185]. We differentiate between the three annotated vocals properties of the test segments regarding the number of singers and the simultaneous presence of different phonemes (cf. Section 5.4.1). The presented values are the medians over all evaluation frames within a property category. Higher values indicate better performance. To give an idea of the scores' distributions, boxplots are shown for SDR, SIR, SAR in Figures 5.7, 5.8, and 5.9 respectively. Beyond, scores of additional metrics are shown in Table 5.5. The Predicted Energy at Silence (PES) measures the energy of the estimated vocals in evaluation frames where the true vocals are all-zero, and the Energy at Predicted Silence (EPS) measures the energy in the true vocals for evaluation frames where the estimate is all-zero (cf. Section 4.4). The presented values are the mean over *all* evaluation frames and lower values indicate better performance. The SDR, SAR, SIR are not defined for frames with a silent estimate or ground truth, so that the PES and EPS complement them for a complete evaluation. Note that a comparison of the presented performance scores with other models trained and tested on MUSDB is not straightforward because we were limited to the songs with English lyrics for training and testing.

			a)			**b)**			**c)**		
Method	**Training data**	**Side info Y**	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
UMX1	MUSDB	-	4.32	8.62	6.73	4.45	8.73	6.56	3.61	8.38	5.39
UMX2	MUSDB +sp.+sil.	-	4.06	8.62	6.22	4.31	8.30	6.56	3.85	7.87	5.69
JOINT3	MUSDB +sp.+sil.	phonemes	3.69	7.38	6.51	3.92	7.29	6.51	3.92	7.29	**6.17**
SEQ-BL1	MUSDB	constant	4.77	9.52	**7.16**	**4.93**	9.39	**6.91**	**4.20**	9.06	5.77
SEQ-BL2	MUSDB	voice activity	4.74	9.18	6.83	4.56	9.14	6.46	3.75	8.62	5.28
SEQ	MUSDB	aligned phonemes	**5.08**	**10.41**	6.82	4.89	**10.21**	6.51	3.86	**9.82**	5.03

Table 5.4: Separation evaluation results in dB. Values for SDR, SIR, SAR are medians over evaluation frames, higher values are better. The differentiated vocals categories are a) one singer, b) 2+ singers singing the same phonemes simultaneously, c) 2+ singers singing different phonemes simultaneously.

Method	**Training data**	**Side info Y**	PES	EPS
UMX1	MUSDB	-	-72.26	-89.21
UMX2	MUSDB+sp.+sil.	-	-75.74	**-97.06**
JOINT3	MUSDB+sp.+sil.	phonemes	-84.09	-81.96
SEQ-BL1	MUSDB	constant	-93.57	-87.45
SEQ-BL2	MUSDB	voice activity	**-101.39**	-80.51
SEQ	MUSDB	aligned phonemes	-95.63	-85.98

Table 5.5: Separation evaluation results of frames containing silent true or predicted sources in dB. Values for PES and EPS are the mean over evaluation frames and lower values are better.

Open Unmix reference and joint approach

The evaluation scores of UMX1 are lower than those reported for the state-of-the-art version of Open Unmix [186]. The reason is the difference in training data such as the amount (we excluded non-English songs and multi-text segments), sampling rate, number of channels, and augmentation. In the original procedure, different random segments of 6 seconds length are cut out of the tracks at every epoch, whereas we are bound by the segment-wise aligned lyrics. However, this simulates the scenario which we investigate in this work: a limited amount of available audio data. Also, we focus on one model instance with vocals as target in order to investigate the effect of text as side information for the vocals estimate. In [186], four specialized model instances are used to estimate the four MUSDB targets which are combined using generalized Wiener filtering. UMX2 performs worse than UMX1. This indicates that the training data and process used for JOINT3, which enable the model to learn an alignment, decrease the separation performance. The evaluation scores for JOINT3 show that the model has successfully learned the separation task jointly with the alignment. However, the evaluation scores are lower than for the original Open Unmix model (UMX1 and UMX2). In the joint approach, the two encoders have to learn representations that enable both alignment and separation, which is worse for the separation than dedicated representations. JOINT3 was evaluated using the soft alignments provided by DTW-attention. However, using hard alignments of DTW instead has only marginal impact on the results. In fact, DTW-attention selects the same phoneme as the DTW path for 84% of all frames on the MUSDB test set, if we consider the phoneme with the highest weight as the one being selected, which is a reasonable assumption given the sharpness of the distribution (c.f. Figure 5.4). We conclude that joint alignment and separation is possible but not beneficial for the separation quality.

Sequential approach

The evaluation scores of the sequential approach SEQ are better than those for the joint approach, JOINT3. This is an expected result because dedicated representations can be learned by SEQ as discussed above. They are also better than those for the original Open Unmix model trained on the same data, UMX1. This improvement has two potential reasons: Firstly, the proposed model has more capacity because of the two encoders and, secondly, it uses text as additional information. We would like to know to which extent the performance increase is due to the text information. This can be seen when comparing SEQ to SEQ-BL1 and SEQ-BL2 which have all the same capacity. The text-informed model SEQ improves the SIR across all vocals properties compared to the less informed baselines. When only one singer is present (property a)) also the SDR is improved

through the text information. The SAR is decreased when using text information with the decrease becoming stronger over categories a), b), and c). This shows that text information is most useful when only one person is singing. In general, the effects of using text are small in terms of the objective metrics which is illustrated by the boxplots in Figures 5.7, 5.8, and 5.9. We discuss the limitations further below. SEQ-BL2 has the lowest PES, which means it performs best on frames without vocals and thus uses the provided voice activity information. SEQ has the second lowest PES which indicates that it also uses the vocal activity information inherent in aligned text.

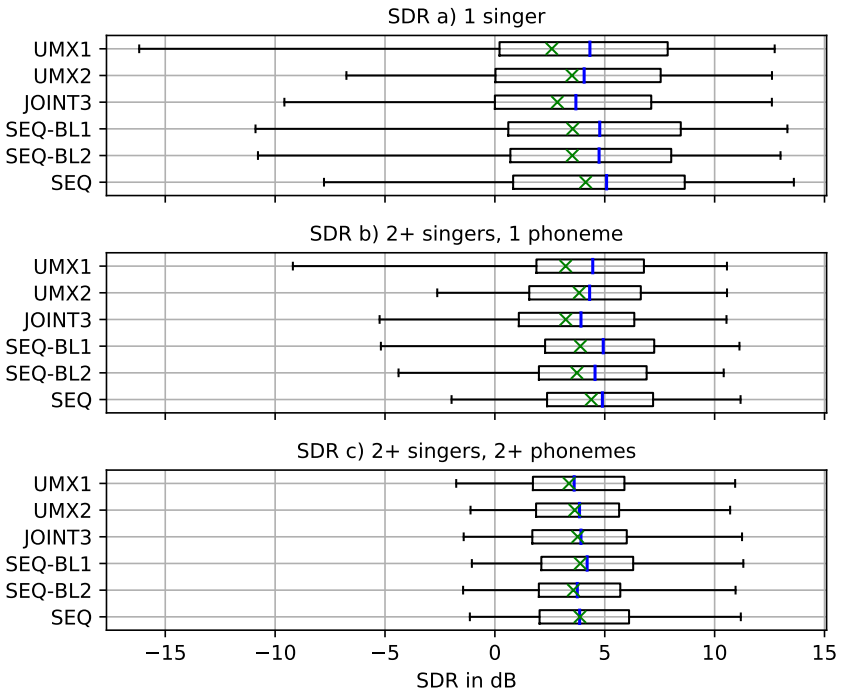


Figure 5.7: Boxplots of the SDR scores for the three vocals categories. Each data point is the score of one evaluation frame of 1 s length. The boxes extend from the first to the third quartile. The whiskers extend from the fifth to the 95th percentile. The medians are shown as blue lines, the means are shown as green 'x'.

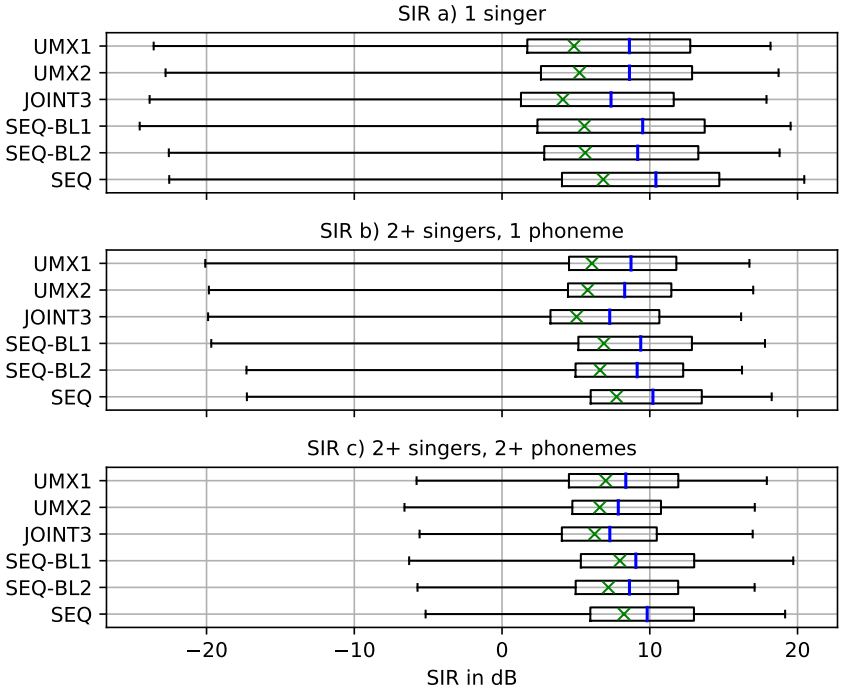


Figure 5.8: Boxplots of the SIR scores for the three vocals categories. Each data point is the score of one evaluation frame of 1 s length. The boxes extend from the first to the third quartile. The whiskers extend from the fifth to the 95th percentile. The medians are shown as blue lines, the means are shown as green 'x'.

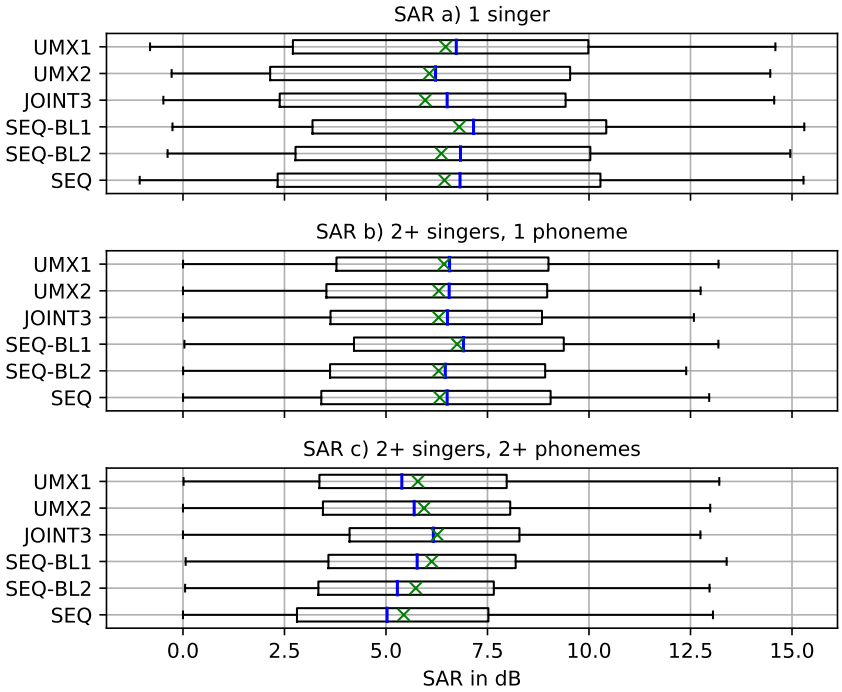


Figure 5.9: Boxplots of the SAR scores for the three vocals categories. Each data point is the score of one evaluation frame of 1 s length. The boxes extend from the first to the third quartile. The whiskers extend from the fifth to the 95th percentile. The medians are shown as blue lines, the means are shown as green 'x'.

In order to evaluate an additional aspect of the separated singing voice signals, we automatically transcribe the lyrics from the voice estimates for the MUSDB test set using the state-of-the-art system proposed by Demirel et al. [31] and compute the Word Error Rate (WER). The transcription system was trained on monophonic solo singing recordings of the Smule Sing! 300x30x2 dataset [1] and uses a language model built from lyrics [31]. The results in Table 5.6 show that the text-informed model SEQ produces a lower WER in each vocals category than the baselines. This means that the given phoneme information helps to preserve the characteristic phoneme properties in the separated voice signals.

Method	**Side info Y**	a)	b)	c)
Mixture		76.06	78.44	89.24
SEQ-BL1	constant	68.30	70.88	83.25
SEQ-BL2	voice activity	63.34	66.81	79.00
SEQ	aligned phonemes	**52.76**	**51.81**	**64.29**
True vocals		37.83	30.85	58.45

Table 5.6: Word Error Rate [%] of the lyrics transcription method proposed in [31].

An illustrative example is given in Figure 5.10. In the shown segment, a female singer sings the words "right there almost got you". The phonetic transcription of this line using 2-letter ARPAbet notation (cf. Section 2.2.2) is "> R AY T > DH EH R > AO L M OW S T > G AA T > Y UW >", where '>' denotes the space token. The unvoiced 's' sound (in "almost") is missing in the estimate of the non-informed model (SEQ-BL1) but when using the text (SEQ) the model is able to separate it. Unvoiced sounds with high energy at high frequencies are difficult to differentiate from drum sounds such as cymbals, which makes text a valuable extra information. It can also be seen that the harmonic structure of the vowels is separated more clearly when using text. This leads to a clean sound and reduces interferences. However, it can also lead to artefacts, especially when multiple singers are present. Listening examples are provided online[6].

Relevance of the phonetic prior information

In order to test what kind of information is derived from the phoneme sequence by the model SEQ, we feed uniform white noise generated in the time domain as input to the audio encoder at test time. We use phoneme sequences of the MUSDB test set as input to the text encoder. The alignment information of the phonemes with respect to their corresponding audio mixture, computed by our alignment model JOINT3, is also provided. In the audio examples[4], it can be observed that the model filters the white noise so that the given phonemes become audible. The experiment shows that the model learned the spectral characteristics of the phonemes and how to use this information for the voice estimation. This explains why the separation with SEQ leads to a lower WER compared to the baselines.

In a second experiment, we test how much the model SEQ relies on the text information if it is conflicting with the observed audio mixture. To this end, we exchange some phonemes in the text after the alignment has been obtained with the original text. Using the example in Figure 5.10, we replaced the 's' in 'almost' with an 'o' so that its phonetic transcription became "AO L M OW OW T" and we replaced the last word 'you' by "S S". In Figure 5.10, it can be seen that the vocals estimate changes accordingly (SEQ (altered text)). The high frequency energy of the 's' sound is

[6] `https://schufo.github.io/plla_tisvs/`

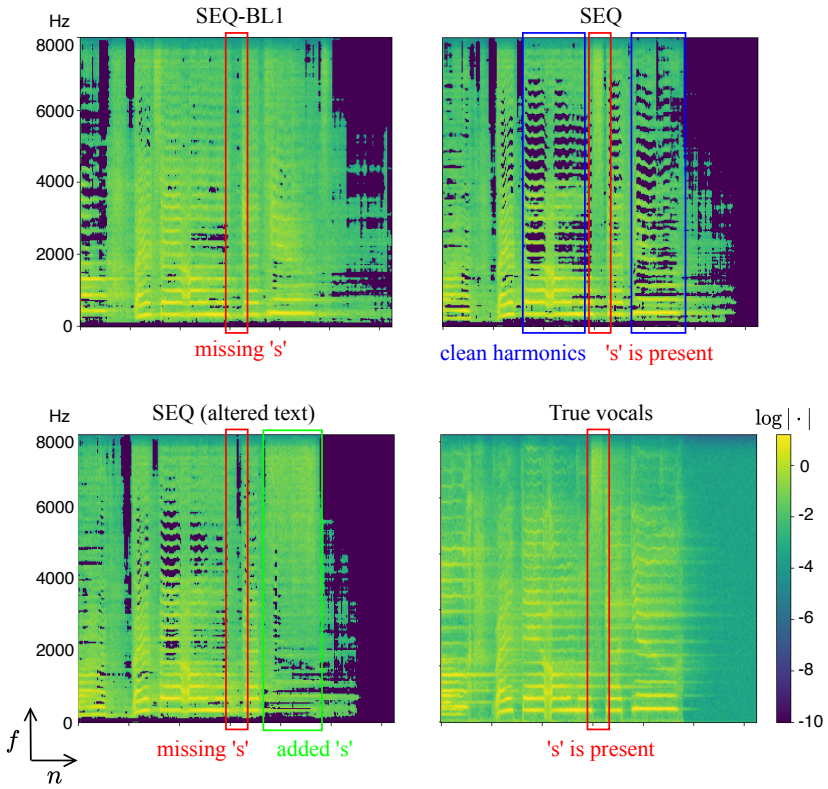


Figure 5.10: Magnitude spectrograms of singing voice estimates obtained with different types of side information. SEQ-BL1: Meaningless side information, SEQ: aligned original phoneme sequence, SEQ (altered text): aligned modified phoneme sequence. At the bottom right, the true vocals are shown for comparison.

now missing where it was correctly estimated before (SEQ) for the word 'almost'. The spectral characteristics of 's' are added in the last frames where the word 'you' was actually pronounced and where a clear harmonic structure was visible when the correct text was used (SEQ). We refer the reader to the audio examples for better illustration. This shows that the text information is actively used to estimate the voice and can even outweigh the information from the observed mixture. This can lead to a better separation as shown above but it can also lead to artefacts when the alignment or the transcription is inaccurate. Editing the phoneme sequence allows us to edit the obtained singing voice signal. This can, for example, be useful to correct small pronunciation mistakes.

Evaluation on mixtures with fixed SNR

In Table 5.7, the separation results evaluated on mixtures with manually fixed SNRs are shown. The three annotated vocals properties of the test segments are differentiated and the values are the median over evaluation frames within each vocals category (a, b, c). For all SNRs and all vocals categories, the text-informed model SEQ achieves higher SIRs than both baselines. The SAR is reduced when using text information on all SNRs. The improvement of the separation through text becomes stronger when the SNR becomes lower. When the SNR is -10 dB also the SDR is clearly improved, even on test segments with multiple singers (b and c). We can conclude that text-informed singing voice separation is more beneficial in challenging conditions whereas it can

lead to degraded performance in very easy conditions.

		a)			b)			c)		
Method	**SNR**	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
UMX1	0	8.00	12.65	10.60	7.16	11.87	9.75	4.90	10.35	6.93
SEQ-BL1		**8.59**	13.62	**11.04**	**7.57**	12.82	**10.00**	**5.43**	11.62	**7.43**
SEQ		8.36	**13.77**	10.27	7.34	**13.04**	9.51	4.94	**11.73**	6.43
UMX1	-5	4.45	8.47	6.93	4.08	8.12	6.18	2.51	6.15	4.17
SEQ-BL1		4.98	9.26	**7.32**	4.56	8.84	**6.55**	3.01	7.00	**4.93**
SEQ		**5.03**	**10.06**	6.83	**4.66**	**9.76**	6.29	**3.08**	**7.95**	4.10
UMX1	-10	0.91	4.01	3.46	0.81	3.39	2.42	0.03	0.65	1.78
SEQ-BL1		1.16	4.07	**3.91**	1.17	3.87	**3.09**	0.21	0.69	**2.42**
SEQ		**1.75**	**6.09**	3.38	**1.86**	**5.83**	2.87	**0.94**	**3.13**	1.74

Table 5.7: Separation evaluation results for mixtures with different SNRs. All values are in dB. Evaluation scores are medians over evaluation frames within a vocal category.

Limitations

The discussions above show that accurate phoneme alignment and correct transcripts are necessary to achieve improvements through text. Otherwise, the vocals estimate will be degraded. In the case of multiple singers singing multiple phonemes (category c), the text contains information only about a part of the target vocals signal, which is defined as mixture of all voice sources in MUSDB. Hence, the text-informed model SEQ as well as the model SEQ-BL2, which is informed by voice activity information derived from aligned text, might suppress the background singers when the lead vocals pause. Since the ground truth vocals contain all singers, this leads to lower evaluation scores. However, it can also be seen as an advantage if only the lead vocals are the source of interest. We refer the reader to the additional audio examples[7] to illustrate the points discussed above.

5.7 Conclusion

The goal of this chapter was to investigate to which extent singing voice separation with deep neural networks can be improved through text information provided by lyrics transcripts. Since lyrics are usually not aligned with the observed mixture signals, we proposed a joint approach to phoneme level lyrics alignment and text-informed singing voice separation.

Experimental evaluation showed that phoneme alignment can benefit from the separation component when the singing voice is mixed with other instruments. Moreover, the proposed alignment method achieved competitive results on two word level alignment test sets although it used less training data than state-of-the-art methods. This is partly due to the learning efficiency of the proposed DTW-attention mechanism compared to conventional approaches such as the CTC loss.

In order to improve the separation performance, lyrics should be aligned first and subsequently be processed by a separation model. With this sequential approach, text information can help to improve the separation quality. The proposed model uses phoneme information actively to shape the spectral content of the voice estimates. This preserves the phonetic properties in the estimates but can also lead to degraded performance in case of inaccurate alignments or transcripts. In

[7] `https://schufo.github.io/plla_tisvs/`

general, the improvement is most noticeable in terms of reduced interference from other sources. Particular cases where text information is especially useful are low SNRs, mixtures with exactly one singer, and ambiguous segments were the spectral content of the voice is similar to other instruments.

However, the overall improvements through text are rather subtle and lead only to small improvements of objective evaluation metrics. At the same time, additional effort is required in order to use the text effectively. Firstly, phonetically accurate transcripts must be obtained. Secondly, they must be precisely aligned with the mixture to be separated. Therefore, it is not advisable to use phoneme sequences by default as side information for singing voice separation. Nevertheless, text is a valuable source of information for the separation of mixture segments which are difficult to separate. Hence, we see specialized use cases for text-informed singing voice separation in challenging separation projects.

Chapter 6

Unsupervised Audio Source Separation

Summary

In this chapter a novel unsupervised deep learning approach to audio source separation is proposed. It exploits fundamental frequency information and parametric generative source models. A neural network is trained to reconstruct the observed mixture as a sum of the sources by estimating the source models' parameters given their fundamental frequencies. At test time, soft masks are obtained from the synthesized source signals. The experimental evaluation on a vocal ensemble separation task shows that the proposed method outperforms learning-free methods based on NMF and a supervised deep learning baseline. Integrating model-based knowledge in the form of source models into a data-driven method leads to high data efficiency: the proposed approach achieves good separation quality even when trained on less than three minutes of mixture signals. The chapter is based on the paper *Unsupervised Audio Source Separation Using Differentiable Parametric Source Models* which is currently under review.

Contents

6.1 Introduction

In this chapter, we consider a more challenging scenario where only mixtures – but no isolated source signals – are available for training and the signals to be separated are produced by sources of the same type. We exploit side information in the form of F0 estimates and integrate model-based knowledge in the form of explicit source models in our separation approach. The motivation for this work are the following two shortcomings of state-of-the-art supervised deep learning methods for musical source separation such as [191, 28, 186].

Firstly, they are not able to separate homogeneous sources. By homogeneous sources we mean sound sources of the same type which have the same sound production mechanism and the same or at least an overlapping frequency range. Homogeneous sources are common in musical mixtures. For example, there may be two guitars in a music group. Musical mixtures may also consist of only one source type such as singing voice in a choir or violins in violin quartets. The separation methods in [191, 28, 186] are able to separate *all* singing voices from an instrumental accompaniment but provide only the mixture of these voices and do not further separate them into the different singer signals. Hence, such methods can neither be used to obtain only the *lead* vocals nor to separate all singers of vocal ensembles or all instruments of violin quartets.

Secondly, they require training data with available ground truth, i.e. mixtures for which target source signals are available in isolation. However, such isolated signals are difficult, sometimes impossible, to obtain for music mixtures as explained in Chapter 1. For example, vocal ensembles usually perform with all singers being in the same room and are also recorded this way. Hence, no separate signal for each singer is recorded. Special sessions may be arranged in order to record signals in isolation [25], however, this is not only extremely costly but also leads to unnatural conditions for the musicians.

Therefore, there is a need for separation methods that do not require ground truth signals for training. Such methods may be learning-free or unsupervised.

Learning-free methods estimate all parameters directly from the test mixture [203]. Hence, they do not require any training data. NMF [108] and its numerous extensions have successfully been used for learning-free musical source separation [203]. Using side information such as musical scores [45, 70] or fundamental frequency [40], NMF-based methods can separate homogeneous sources.

Unsupervised methods have a training stage and require only mixtures (no isolated sources) for learning. At test time, their parameters are fixed. They have the potential to provide better performance than learning-free methods while being less demanding regarding data than supervised methods. Recently proposed unsupervised deep learning methods for audio source separation are based on assumptions such as uncorrelated [213, 212] or non-homogeneous sources [137, 171]. Therefore, they are not applicable to music mixtures where sources are correlated and possibly homogeneous.

We propose and evaluate a novel approach to unsupervised source separation with DNNs which does not make such assumptions. It is hence also applicable but not limited to music mixtures. The approach is inspired by the recent line of research which integrates signal processing models in DNNs to incorporate domain knowledge [174, 42]. Each source is modeled with a differentiable parametric source model. During training, the task of the DNN is to re-synthesize the observed mixture as a sum of the sources by estimating the source parameters. Separation is achieved because the F0s for all sources are estimated from the mixture and assigned to the sources beforehand. This can be done using methods such as [26, 165].

Besides being unsupervised and able to separate homogeneous sources, the approach has further advantages: high data efficiency as well as parametric, hence interpretable and modifiable, source estimates. The source code related to this chapter is available online [1].

6.2 Related work

In this section we review work on homogeneous musical source separation, learning-free and unsupervised source separation, and, finally, on the integration of signal processing models in deep neural networks.

Homogeneous audio sources are not easily distinguishable in the time-frequency domain and pose a permutation problem [72, 217]. While permutation-invariant training is used for supervised speech separation [217, 96], methods for musical homogeneous source separation exploit side information such as F0 estimates [146, 40] or a musical score [58, 45, 70] to guide the separation.

Two deep learning approaches for supervised choir separation were proposed recently. In this context, a choir is composed of four homogeneous sources: a soprano, alto, tenor, and a bass singer. Petermann et al. [146] modified the conditioned U-Net [126] so that the target source can be selected and separated using its F0 information. Results show that this leads to improved objective separation quality compared to using non-informed source-specific models. However, ground truth source signals are needed for training and they are rare for choir recordings. This motivated Gover and Depalle [58] to synthesize choir singing from MIDI files and to use this synthetic data for training of a score-informed DNN. However, when tested on real choir recordings, the model is outperformed by the learning-free, score-informed NMF proposed in [45]. This shows that the performance of supervised DNNs depends strongly on the quality and quantity of the training data.

Therefore, learning-free methods are a powerful alternative in limited data settings. Several separation methods based on NMF are learning-free and can exploit side information to separate homogeneous sources. NMF approximates a spectrogram with a matrix product of two low-rank matrices containing spectral templates and their activations, respectively [203]. Ewert and Müller [45] proposed to initialize both templates and activations using musical score information. This leads to improvements compared to random initialization. Using the score allows even to separate notes played by the left and the right hand in piano recordings. Similarly, Hennequin et al. [70] used a musical score to initialize the activations whereas the templates consist of parametric frequency atoms. Durrieu et al. [40] formulated an advanced signal model using multiple NMF decompositions. The predominant source is modeled with a source-filter model and all other sources are captured by an unconstrained NMF. First, the F0 of the predominant target source is estimated using the signal model. Then, the F0 is used to guide the separation. Nakamura and Kameoka [136] proposed a powerful signal model combining NMF and harmonic-temporal clustering and integrated a source-filter model. It allows for blind, learning-free separation of harmonic sounds. A drawback of NMF-based methods is the low degree of flexibility because only a fixed number of spectral templates is used to describe a signal. This limits their performance, especially when inherent assumptions are violated.

Recently, efforts have been made to make more flexible deep learning based source separation also usable in cases where no mixture-target pairs are available for training. Most works focus on creating learning targets artificially from mixtures or side information in order to train DNNs in a

[1] `https://github.com/schufo/umss`

supervised way in the absence of real targets. Seetharaman et al. [171] obtain targets for singing voice/accompaniment separation by clustering time-frequency bins of mixtures using several simple perceptual cues. Hung et al. [81] obtain harmonic target masks from well-aligned musical scores and further support the training process using score transcription models. Also deep clustering models [72] have been trained for speaker separation without ground truth signals [33, 195]. The targets are obtained by clustering the mixture based on spatial information. The methods above yield good results but require substantial amounts of (unlabeled) training data and cannot separate homogeneous correlated sources.

As an alternative, it has been proposed to train deep generative models on isolated source signals to use them subsequently for source separation [137] or speech enhancement [110]. However, this strategy is challenging for musical source separation because it requires a large amount of isolated source signals and uncorrelated sources.

Lastly, mixture invariant training has been proposed recently in [213] and refined in [212] for unsupervised learning of audio source separation without a need for artificial targets. During training, the sum of two mixtures is given as an input and the DNN has to separate all sources so that, given the respective optimal binary mixing matrices, the two mixtures can be reconstructed individually. Since it is necessary that the sources are uncorrelated [212], this approach is not an option for musical source separation.

The method proposed in this chapter uses F0 information to separate the (possibly homogeneous) sources like the learning free-methods in [45, 40] and the supervised methods in [146, 58]. It provides better performance than learning-free methods and does not require expensive labeled data like supervised methods. Our learning strategy is fundamentally different from other unsupervised methods: it is not limited to uncorrelated sources like [212] and does not rely on artificial source targets which require the availability of aligned scores [81], sufficient spatial information in the mixture [195, 33], or non-homogeneous sources [171]. The proposed training objective is to re-synthesize the mixture with differentiable parametric source models. The only assumptions are that the number of sources is known and that their F0 can be estimated. In contrast to the unsupervised methods reviewed above, the proposed one can separate homogeneous sources, requires only a small amount of unlabeled data, and provides interpretable and modifiable source estimates.

There is a recent line of research that explores the combination of data-driven and knowledge-based methods to take advantage of both paradigms [130, 174, 42]. The integration of differentiable source models in the DNN-based source separation process is inspired by this model-based deep learning research. Specifically related to our work are recent speech synthesis methods which use differentiable parametric voice models and estimate their parameters using DNNs [208, 154]. We use similar voice models but in a different context. Engel et al. [42] implemented a code library for differentiable digital signal processing and showed the advantages of model-based deep learning for tasks such as synthesis, timbre transfer and dereverberation. The DNN architectures and the differentiable signal processing implementations we use in our experiments are inspired by their work. To the best of our knowledge, the proposed method is the first one that uses model-based deep learning for musical source separation.

6.3 Proposed method

We observe the single-channel mixture $x(t) = \sum_{j=1}^{J} v_j(t)$ of J monophonic source image signals $v_j(t)$ where $t \in \{1, ..., T\}$ indexes discrete time samples. Our goal is to estimate all source

image signals v_j. We propose a novel approach to train a DNN for this task without access to any isolated source signals. The sources are modeled with differentiable parametric source models which we describe in Section 6.3.1. The DNN estimates the source parameters given the F0 as explained in Section 6.3.2. The objective of the unsupervised training strategy is to re-synthesize the mixture. Details about training are given in Section 6.3.3 and an overview of the procedure is presented in Figure 6.1.

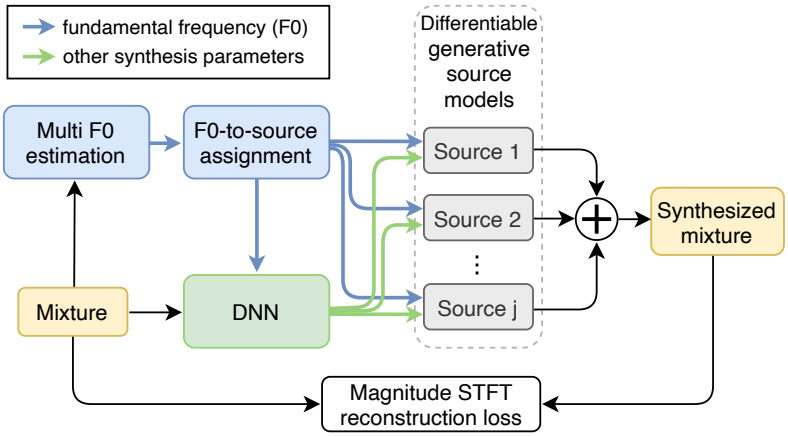


Figure 6.1: Overview of the proposed unsupervised training procedure of a Deep Neural Network (DNN) for audio source separation.

At test time, the synthesized source signals can either be used directly as source estimates or soft masks can be derived from them for Wiener filtering of the mixture. Implementation details are described in Section 6.3.4

6.3.1 Source model

The proposed method is not specific to any particular source model and any parametric model may be used as long as it can be formulated in a differentiable way. This is often facilitated by automatic differentiation software such as TensorFlow [2] or PyTorch [145]. In this work, we use the source-filter model of speech production [48] (cf. Section 2.2.1). It describes a signal as an excitation signal from a sound source (e.g. the glottis) which is modified by a time-varying filter (e.g. the vocal tract). An exemplary visualization of our source-filter model is presented in Figure 6.2.

We do not explicitly model reverberation or other mixing filters which may have been applied. However, if present, such effects will still be captured to a certain degree by the source model. Hence, the model estimates source images and not the clean source signals.

In the following, we assume that the true source image signal $v_j(t)$ is segmented into N frames of length T' samples. The n-th frame is given by

$$v_j(n,t) = v_j(t + nB), \quad t \in \{1, ..., T'\} \tag{6.1}$$

where B is the hop size between frames in samples and $n \in \{1, ..., N\}$. We denote the estimate of the source signal frame generated by the source model using a tilde: $\tilde{v}_j(n,t)$. The source model

may be formulated in the z-domain as

$$\tilde{V}_j(n,z) = E_j(n,z)\frac{1}{A_j(n,z)}. \tag{6.2}$$

$E_j(n,z)$ is the z-transform of the excitation signal $e_j(n,t)$ and $\frac{1}{A_j(n,z)}$ is the transfer function of a time-varying all-pole filter of order K. We drop the source index j for brevity hereafter but we would like to emphasize that each source is modeled with its dedicated model. The filtering process in (6.2) is best described by the difference equation

$$\tilde{v}(n,t) = e(n,t) - \sum_{k=1}^{K} a_k(n) \cdot \tilde{v}(n,t-k) \tag{6.3}$$

where $a_k(n)$ are the filter coefficients for frame n and '$\cdot$' denotes scalar multiplication. We explain how to deal with frame boundaries and other implementation details in Section 6.3.4.

A sinusoids plus noise model is employed to generate the excitation signal $e(n,t)$. It is an expressive synthesis model for music [172] and speech signals [103, 155, 44] which synthesizes sound as a sum of sinusoids and filtered white noise. A differentiable version was recently implemented by Engel et al [42, 43] who showed impressive results using it for model-based deep learning. Since we model a monophonic source, we constrain the sinusoid frequencies to be integer multiples of a fundamental frequency. The model thus reduces to the *harmonics plus noise* model [103, 42] which we formulate as

$$e(n,t) = [\alpha(n,t) \cdot h(n,t)] * r(t) + [w(t) * d(t)] \cdot g(n) \tag{6.4}$$

where $*$ denotes the convolution operator, $\alpha(n,t)$ is the time-varying amplitude of the harmonic signal $h(n,t)$, and $r(t)$ and $d(t)$ are Impulse Responses (IR) of time-invariant finite impulse response (FIR) filters. $w(t)$ is a uniform white noise signal and $g(n)$ is the constant noise gain for frame n.

The harmonic signal $h(n,t)$ is defined as

$$h(n,t) \;=\; \sum_{i=1}^{I} \sin(\phi_i(n,t)) \tag{6.5}$$

$$\phi_i(n,t) = 2\pi \sum_{\tau=1}^{t} i \cdot f_0(n,\tau)/f_s \tag{6.6}$$

where ϕ_i is the instantaneous phase of the i-th harmonic, f_0 is the fundamental instantaneous frequency, and f_s is the sampling frequency. The initial phase is assumed to be zero. Equation (6.6) is a numerical approximation of integration based on *sample and hold* [76, Ch. 4]. Note that the signal $h(n,t)$ is fully parameterized by the time-varying fundamental frequency f_0.

The filter $r(t)$ imposes a fixed spectral shape on $h(n,t)$. Without $r(t)$, all sinusoids have the same amplitude. However, for certain sound sources a specific time-invariant spectral shape can be assumed, e.g. the spectral roll-off of the glottal signal [48]. Alternatively, a specific amplitude parameter may be used for each sinusoid in $h(n,t)$ [172, 103]. However, we choose to make the gain dependent on the frequency and not on the harmonic number. Similarly, $d(t)$ determines the spectral shape of the noise component. Both filters are time-invariant so that they only account for the global spectral shape. Short term variations, e.g. due to articulations of words, are modeled by the all-pole filter $\frac{1}{A(n,z)}$.

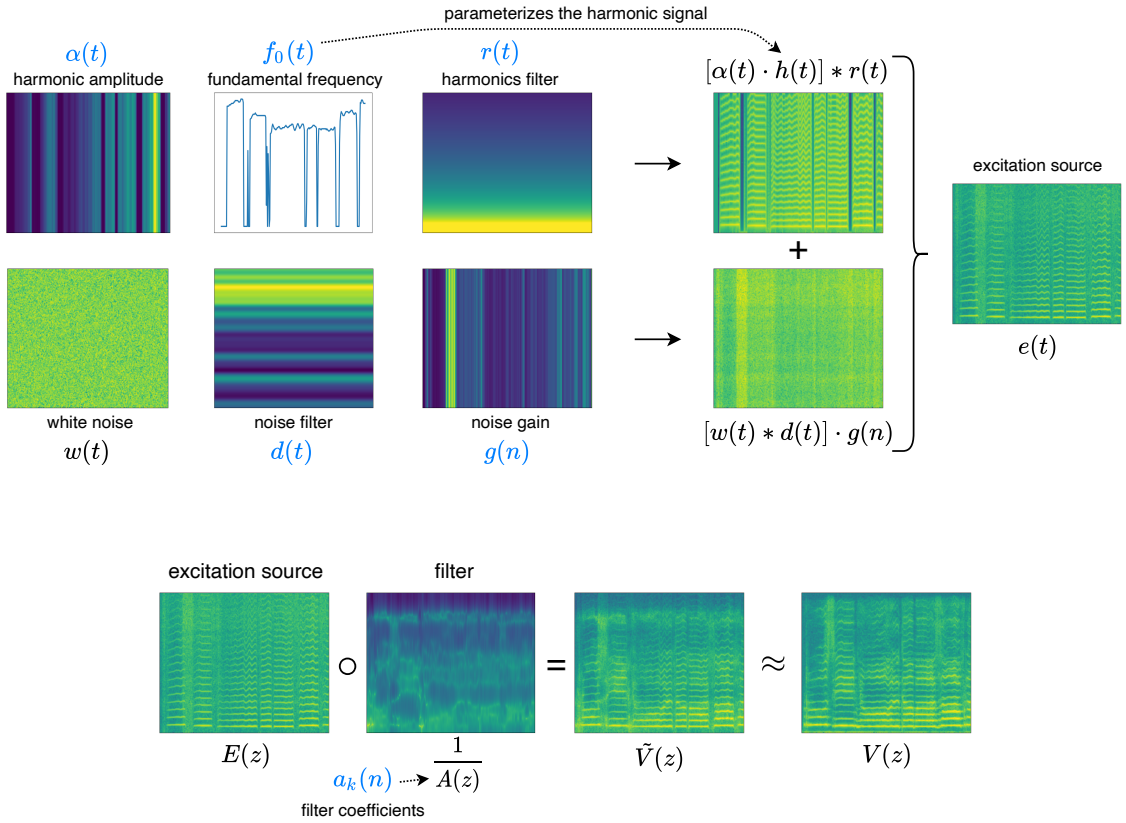


Figure 6.2: Exemplary overview of the source-filter model decomposition. The model parameters are denoted in blue font. The '$\circ$' denotes element-wise multiplication. Although most components are visualized through magnitude spectrograms, processing is not necessarily done in the time-frequency domain.

The source model parameters are $\{a_k(n), \alpha(t), f_0(t), r(t), g(n), d(t)\}$. In the next section, it is explained how they are obtained. α and f_0 need to vary slowly enough over time for the model to be mathematically identifiable. This is indirectly enforced by the way these parameters are estimated which leads to smooth trajectories.

6.3.2 Parameter estimation

We assume that the fundamental frequency trajectories for each of the J sources can be obtained from the mixture signal with a multiple F0 estimation system. Given that many such systems exist [93, 26, 218] and that it is still an active research area, we are confident that this is a reasonable assumption. When all F0s are obtained, each F0 value needs to be assigned to one specific source. Various solutions for the F0-to-source assignment problem have been proposed [18, 123, 165]. Most of them are based on principles such as temporal pitch continuity, low voice crossing probability, and minimal temporal gaps within a voice [123]. In our experiments we use a heuristic based on these principles, cf. Section 6.4.2. F0 estimates are usually provided at a frame rate which is smaller than the sample rate [93, 26, 218]. Therefore, following [42], the source specific F0 time series are upsampled to the sample rate using bilinear interpolation. This leads to smooth trajectories.

In the following, we describe how the remaining synthesis parameters are estimated with a

DNN for each source given its F0. The task the DNN has to solve is similar to the one of NMF in the context of learning-free F0-informed source separation in [45, 40]. Note that the differentiable source models do not put any constraints on the neural network type or architecture which is used to estimate the parameters. Here we use a simple DNN as in [42] and focus on the advantages of including parametric source models in deep learning based separation.

The mixture signal is represented by the logarithmic magnitude of its spectrogram obtained by an STFT of $x(t)$. The spectrogram has F frequency bins and N time frames. Each spectrogram is normalized by subtraction of its mean and division by its standard deviation. Then, each frequency bin is scaled and shifted by dedicated learned scalars. The DNN architecture is similar to the one used in [42]. An overview of the DNN and further processing steps for the parameter estimation is presented in Figure 6.3. We use linear layers and unidirectional Recurrent Neural Networks (RNN) with Gated Recurrent Units (GRUs) [19]. The Multi-Layer Perceptron (MLP) consists of three repetitions of linear layer, layer normalization [5], Leaky ReLU activation [119].

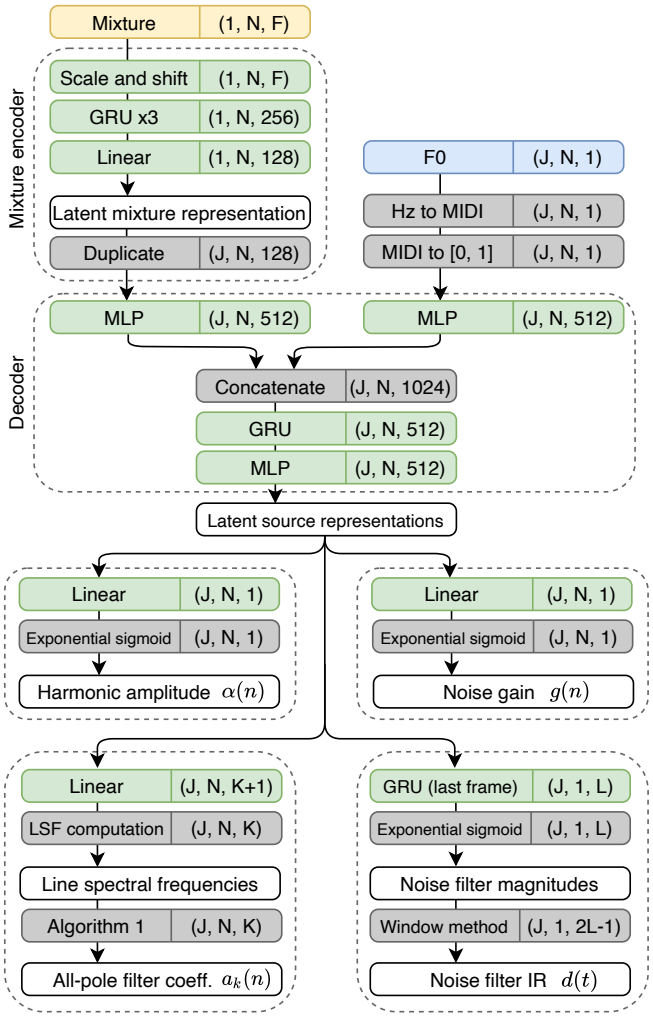


Figure 6.3: Overview of the processing steps for the parameter estimation. Transformations with learnable parameters are shown in green, predefined processing steps in gray, (intermediate) outputs in white boxes. The output shape of a transformation is shown in the right part of the box.

The mixture encoder computes a latent representation of the mixture and then creates as many duplicates as there are sources. Each latent mixture copy is then combined with the F0

information of one source by the decoder. The F0 is provided at the frame rate of the mixture STFT. The F0 values are converted from Hertz to MIDI note numbers which are then normalized to the interval $[0, 1]$. The decoder computes a separate latent representation for each source. The source model parameters are obtained from this source representation by one last transformation with learned parameters (linear layer or GRUs) followed by some predefined processing steps. The frame-wise harmonic amplitude $\alpha(n)$ and the noise gain $g(n)$ are computed with a linear layer with an exponential sigmoid activation function [42] defined as

$$y = y_{\max} \cdot \text{sigmoid}(x)^{\log(10)} + 10^{-7} \tag{6.7}$$

where x and y are the input and output value, respectively, and $y_{\max}$ is a scalar determining the upper bound of y. Following [42], the harmonic amplitude is then upsampled to the sample rate using overlapping Hann windows which yields a smooth $\alpha(t)$. The noise gain is only required at frame rate.

The filter with impulse response $d(t)$ is time-invariant. Therefore, the network output from which $d(t)$ is computed should summarize information about the whole source signal. We obtain such an output by processing the latent source representation with a unidirectional RNN with GRUs and then using only the output at the last time frame for further processing. This last output frame is processed with the exponential sigmoid presented in (6.7) which results in a tensor of shape $(J, 1, L)$. The tensor contains L samples of the magnitudes of the single-sided frequency responses of the noise filters for J sources. The samples define a zero-phase FIR filter according to the frequency sampling method [178]. Using the window method [179], we obtain the impulse response $d(t)$ as it is also done in [42].

The impulse response $r(t)$ of the time-invariant harmonics filter can be obtained in the same way as $d(t)$ from a DNN output. One may also wish to make the filters time-varying by using a linear layer for the last transformation or using all GRU outputs. However, for the scope of this work, we fix $r(t)$ manually. More details about $r(t)$ are given in Section 6.4.2 where we describe the experimental setup.

For the estimation of the parameters we addressed so far, practical ways have already been proposed by Engel et al. [42]. More care needs to be taken when obtaining Infinite Impulse Response (IIR) filters such as $\frac{1}{A(z)}$ from DNN outputs because it must be avoided that the filter becomes unstable. The filter $\frac{1}{A(z)}$ of order K is fully defined by the filter coefficients a_k with $k \in \{1, ..., K\}$ (see also the difference equation in (6.3)). However, no condition which guarantees stability can be formulated for the filter coefficients directly.

Different parameterizations of all-pole filters exist which allow for the formulation of stability criteria. One option would be to estimate K reflection coefficients [149] with the DNN. Stability is guaranteed if the coefficients are within the interval $]-1, 1[$. They can be converted to the filter coefficients with a simplified version of the Levinson-Durbin algorithm [112, 39], see also [149]. This approach was used in [154] to define the all-pole vocal tract filter with a DNN for speech synthesis. The drawback of this method is that conclusions about the filter's frequency response can neither be drawn from the reflection coefficients nor the filter coefficients.

Therefore, we choose to parameterize the all-pole filter with Line Spectral Frequencies (LSFs) [83]. LSFs are related to the positions of the filter poles and thus to the frequency response [149]. Hence, they provide an interpretable parametrization. They also allow the formulation of constraints to control the filter response and can be interpolated [21]. An introduction to LSFs

is given in Section 2.2.4. A stable minimum-phase all-pole filter $\frac{1}{A(z)}$ of order K is defined by K LSFs fulfilling the relation

$$0 < \omega_k < \omega_{k+1} < \pi \tag{6.8}$$

where ω_k denotes the k-th LSF.

We obtain such LSFs as follows. The latent source representations are transformed by a linear layer which yields a tensor of shape $(J,\ N,\ K+1)$. It is processed by an exponential sigmoid activation with $y_{max} = 2$. The resulting tensor can be viewed as $J \cdot N$ vectors $\mathbf{u} \in \mathbb{R}^{K+1}$. The vectors are normalized so that their entries u_k sum up to π:

$$\bar{\mathbf{u}} = \frac{\mathbf{u}}{\sum_{k=1}^{K+1} u_k} \cdot \pi. \tag{6.9}$$

The K LSFs respecting (2.9) are then obtained by the cumulative sum

$$\omega_k = \sum_{i=1}^{k} \bar{u}_i \ \text{ for } \ k = 1, ..., K. \tag{6.10}$$

Finally, the LSFs are converted to filter coefficients using Algorithm 1 which is detailed in Section 2.2.4.

To sum up the parameter estimation, F0s are estimated from the mixture and assigned to the sources using existing methods. $a_k(n)$, $\alpha(t)$, $g(n)$, and $d(t)$ are obtained with a DNN and $r(t)$ is fixed manually in this work but may also be estimated by a DNN.

6.3.3 Unsupervised training

The proposed training procedure requires only mixture signals, no isolated source signals are needed. During training, the task of the DNN is to reconstruct the observed mixture by estimating the corresponding parameters of the source models. A schematic overview of the training process is presented in Figure 6.1. The generated mixture estimate $\tilde{x}(t)$ is the sum of the source signals generated by the source models:

$$\tilde{x}(t) = \sum_{j=1}^{J} \tilde{v}_j(t). \tag{6.11}$$

In theory, the source models make it possible to synthesize a mixture estimate $\tilde{x}(t)$ which is perceptually identical to the true mixture $x(t)$. Since absolute phase offsets are irrelevant for human perception, the true and estimated mixtures do not need to have the same phase. Therefore, the reconstruction loss $\mathcal{L}_{rec}$ is formulated as a multi-scale spectral loss [42]

$$\mathcal{L}_c = \|\mathbf{X}_c - \tilde{\mathbf{X}}_c\|_1 + \|\log(\mathbf{X}_c) - \log(\tilde{\mathbf{X}}_c)\|_1 \tag{6.12}$$

$$\mathcal{L}_{rec} = \sum_c \mathcal{L}_c \tag{6.13}$$

where $\mathbf{X}_c$ and $\tilde{\mathbf{X}}_c$ denote the magnitude spectrograms of the input mixture and its estimate, respectively, and $c = [2048, 1024, 512, 256, 128, 64]$ indicates the FFT size used to compute the STFT. The time frames overlap by 75%.

The separation of the sources is essentially ensured by the assignment of the F0s to the sources

similar to score/F0-informed separation with NMF [45, 40]. The DNN has to estimate the remaining parameters for each source in order to minimize the loss.

At test time, the DNN parameters are fixed and a soft mask for source j is obtained by the element-wise division $\tilde{\mathbf{V}}_j / \sum_{j=1}^{J} \tilde{\mathbf{V}}_j$ where $\tilde{\mathbf{V}}_j$ is the magnitude spectrogram of the generated source signal $\tilde{v}_j$. The *final* time domain source estimates, marked with a hat, $\hat{v}_j$ are obtained by Wiener filtering using the soft masks.

6.3.4 Implementation details

We implemented the proposed method using the PyTorch framework [145]. For the differentiable source models, we make use of the DDSP library [42]. We re-implemented it in PyTorch and added extensions such as Algorithm 1 and an all-pole filter. The code is available online[2].

Using an all-pole filter in the proposed framework entails two challenges. Firstly, the autoregressive filtering process is slow because it does not allow for precise parallel processing of frames. Secondly, the filter is time-varying, i.e. its coefficients are different at every frame. Therefore, extra care must be taken to ensure a smooth transition between frames to avoid artefacts. The DNN operates at a frame rate which is determined by the FFT size T' and hop size B used to compute the STFT of the mixture. Hence, the DNN estimates a set of K filter coefficients for each frame. We apply the all-pole filter to all frames in parallel using the difference equation in (6.3) in order to make filtering faster. The initial states $\tilde{v}(n, t)$ with $t \leq 0$ are set to zero for each frame. The output frames are then multiplied with a Hann window and the final output signal is obtained by the overlap-add method. It is therefore important that the hop size B is chosen so that the Hann window respects the constant overlap-add condition. We use $B = T' / 2$ in our experiments. Windowing and 50% overlap make the transition between frames smooth. The errors that are introduced by setting the initial states to zero instead of taking samples of the previous frame into account (which is not possible in parallel processing) are negligible: Firstly, the errors are larger at the start of each frame where their importance is mitigated by the window. Secondly, since the filter coefficients are different at each frame, the importance of samples from the previous frame is reduced.

We found it to be critical to implement Algorithm 1 with double precision (64-bit floating point) because it is more sensitive to rounding errors with increasing filter order, which can lead to unstable filters.

The excitation signal $e(t)$ is computed as follows. The harmonic component $\alpha(t) \cdot h(t)$ and the noise $w(t)$ are generated in the time domain for the entire signal length T. The time-invariant FIR filters $r(t)$ and $d(t)$ and the noise gain $g(n)$ are applied frame-wise in the frequency domain followed by overlap-add.

6.4 Experiments

We evaluate the proposed approach on an *a cappella* vocal ensemble separation task. The goal is to estimate the individual signals of J singers from their mixture. This task is a good choice for evaluation because sources in vocal ensembles are homogeneous and correlated. Moreover, singing voice is a challenging musical source. It has a strongly time-varying spectral envelop and also produces sounds without any harmonic content such as unvoiced consonants. Also, only small

[2] `https://github.com/schufo/umss`

amounts of data for supervised training are available for vocal ensemble separation. This makes unsupervised learning an important alternative.

6.4.1 Data

As training and validation data, we use the Bach Chorals (BC) dataset[3] and the Barbershop Quartet (BQ) dataset[4]. The BC set contains 26 chorals sung by a vocal quartet with the voices Soprano, Alto, Tenor, Bass (SATB). The BQ set contains 22 songs performed by a vocal quartet comprising the voices tenor, lead, baritone and bass. All voices are available in isolation for both sets. This allows us to compare the proposed unsupervised approach to supervised baselines.

We combine the BC and BQ sets to generate what we call the *full* training and validation sets. The *full* validation set comprises songs 8 and 9 of the BC set and songs 8 and 9 of the BQ set and has a total length of 9 minutes and 10 seconds. The remaining songs build the *full* training set with a total length of 91 minutes and 20 seconds. We also build a *small* training set consisting of BC song 1 with a length of 2 minutes and 40 seconds and a *small* validation set consisting of BC song 2 with a length of 2 minutes and 20 seconds. When mixtures with less than four singers are created from the individual voice recordings, all possible combinations of the four voices with the desired number of singers are used with the constraint of using only one singer per voice.

As test data, we use the Choral Singing Dataset [25]. It comprises three songs performed by an SATB choir with four singers per voice. All 16 singer signals are available in isolation which allows to evaluate the separation with objective metrics. We add the signals of individual singers (max. one per voice) to produce the test mixtures. For mixtures of $J = 4$ singers, the test set has a length of 6 minutes and 48 seconds. For mixtures of $J = 2$ singers, the test set has a length of 40 minutes and 48 seconds due to more possible voices combinations.

We resample the training, validation, and test data to a sample rate of 16 kHz. The training examples are excerpts of 4 seconds length which are randomly drawn from the training set. The validation and test set are split into fixed excerpts of 4 seconds length. There is no overlap regarding singers, songs, or recording setup between the test and training data. While the training data contain a considerable amount of reverberation, the test recordings are much less reverberant.

6.4.2 Experimental setup

We perform two sets of experiments: one using mixtures of $J = 2$ singers for training and testing, and a second one using mixtures of $J = 4$ singers.

The F0s are obtained from the mixture signals using the multiple F0 estimation model of Cuesta et al. [26]. We use the pre-trained "Model 3" which is available online[5]. For the F0-to-source assignment on the given data, we found that a simple heuristic is sufficient. It is based on the same principles as more advanced solutions such as temporal pitch continuity, low voice crossing probability, and minimal temporal gaps within a voice [123, 165]. The F0 estimator provides m F0 values at each time frame. First, we process all frames where $m = J$. The F0 values are sorted according to magnitude and assigned to the voices assuming they do not cross. Subsequently, the remaining frames are processed. When $m < J$ we assume that some voices are silent. We assign each F0 value to the source which has the closest F0 value in a previous or subsequent frame (pitch

[3] https://www.pgmusic.com/bachchorales.htm
[4] https://www.pgmusic.com/barbershopquartet.htm
[5] https://github.com/helenacuesta/multif0-estimation-polyvocals

continuity principle). The zero value is assigned to silent sources. In the rare case that $m > J$, we sort the values according to magnitude and select J F0s using the pitch continuity principle and assign them to the sources.

The mixture spectrograms are computed using an FFT size of $T' = 512$ and a hop size of $B = 256$ samples. Hence, they have $F = 257$ frequency bins and $N = 250$ time frames. We fix the impulse response $r(t)$ so that the frequency response of the FIR filter falls off with a rate of 6 dB/octave, with a reference frequency of 200 Hz below which the response is flat. We chose this rate because it accounts for the combined spectral characteristics of the glottal source and lip radiation [48]. We set the order of the all-pole filter to $K = 20$. The spectrograms of the synthesized source signals $\tilde{\mathbf{V}}_j$ to compute the soft masks are computed with an FFT size of 2048 and a hop size of 256 samples.

Training is done with the Adam optimizer [90], a batch size of 16 and a learning rate of 0.0001. Training is stopped after 200 consecutive epochs without improvement of the validation loss.

We train the model with the proposed unsupervised approach on the *full* and on the *small* training set. We call the experiments UnSupervised-Full (US-F) and UnSupervised-Small (US-S), respectively. As a reference, we also train the model in a supervised way on the same data. In this case, the loss is computed for each source estimate individually using its target. The total loss is the sum of the "source losses". We call these experiments SuperVised-Full (SV-F) and SuperVised-Small (SV-S).

6.4.3 Baselines

We compare the proposed unsupervised approach to two learning-free methods and one supervised approach. The baselines also exploit F0 information and compute soft masks for Wiener filtering.

The first learning-free baseline was proposed by Ewert and Müller [45]. It approximates the mixture magnitude spectrogram $\mathbf{X}$ with a simple NMF decomposition:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{W}\mathbf{H} \tag{6.14}$$

where $\mathbf{W} \in \mathbb{R}^{F \times R}$ is a matrix of R spectral templates and $\mathbf{H} \in \mathbb{R}^{R \times N}$ contains their activations over N time frames. In [45], $\mathbf{W}$ and $\mathbf{H}$ are initialized using information from an aligned musical score. One spectral template per semitone is used. In our experiments, we have F0 information available, which is more precise than a semitone scale. Therefore, we use a scale with a precision of $\frac{1}{10}$ of a semitone. The F0 values are converted from Hertz to MIDI numbers which are rounded to one decimal place for this purpose. The F0s are used for initialization and for the separation to determine which activations belong to which source. After testing different combinations, we obtained the best results with an FFT size of 2048 and a hop size of 256 samples to compute the spectrograms. We call this method NMF1.

The second learning-free baseline is the method proposed by Durrieu et al. [40]. The target source is modeled with a source-filter model and the residual sources are modeled with a conventional NMF. The method approximates the power spectrogram of the mixture $\mathbf{X}_{\text{pow}}$ as

$$\mathbf{X}_{\text{pow}} \approx \hat{\mathbf{X}}_{\text{pow}} = \underbrace{(\mathbf{W}^{\Gamma}\mathbf{H}^{\Gamma}\mathbf{H}^{\Phi})}_{\text{filter}} \circ \underbrace{(\mathbf{W}^{F0}\mathbf{H}^{F0})}_{\text{source}} + \underbrace{(\mathbf{W}^{O}\mathbf{H}^{O})}_{\text{residual}} \tag{6.15}$$

where $\circ$ denotes element-wise multiplication. $\mathbf{W}^{\Gamma} \in \mathbb{R}^{F \times P}$ contains P spectral atoms consisting

of shifted Hann windows with 75% overlap so that the whole frequency range is covered across $\mathbf{W}^{\Gamma}$. The matrix $\mathbf{H}^{\Gamma} \in \mathbb{R}^{P \times K}$ contains their activations to combine them to smooth filters and $\mathbf{H}^{\Phi} \in \mathbb{R}^{K \times N}$ contains activations to combine the smooth filters. $\mathbf{W}^{F0} \in \mathbb{R}^{F \times U}$ contains a fixed set of U spectral templates defined by the glottal source model KLGLOTT88 [94]. There is one spectral template for each F0 in steps of $\frac{1}{20}$ semitone between a minimum and a maximum frequency. $\mathbf{H}^{F0} \in \mathbb{R}^{U \times N}$ contains the activations of the spectral templates. In [40], $\mathbf{H}^{F0}$ is initialized using F0 information of the predominant source estimated using the signal model in (6.15). We initialize $\mathbf{H}^{F0}$ using the F0 information we obtained from the multi-pitch estimation [26]. In [40], the spectral templates of the residual sources $\mathbf{W}^{O} \in \mathbb{R}^{F \times R}$ and their activations $\mathbf{H}^{O} \in \mathbb{R}^{R \times N}$ are initialized randomly. We initialize them using the F0 information for the corresponding sources as done in NMF1. This leads to improvements compared to random initialization. We call this baseline NMF2. The parameters to be estimated are $\{\mathbf{H}^{\Gamma}, \mathbf{H}^{\Phi}, \mathbf{H}^{F0}, \mathbf{W}^{O}\mathbf{H}^{O}\}$. For NMF2, we obtained the best results using an FFT size of 1024 and a hop size of 128 samples. To the best of our knowledge, these two baselines are among the best learning-free, informed methods for musical and homogeneous source separation.

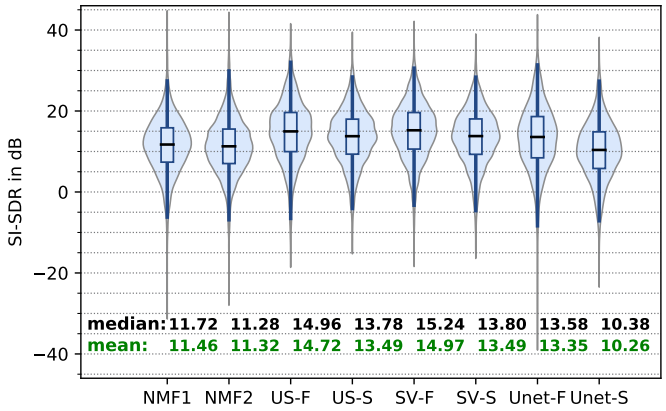
Furthermore, we train the F0-informed supervised deep learning model for vocal ensemble separation proposed by Petermann et al. [146] on our data. They use a classical U-Net architecture with a control mechanism [126]. The F0 information is used to select the target source and to guide the separation. For this baseline, mixture and target source spectrograms are computed using an FFT size of 1024 and a hop size of 256 samples. Wiener filtering is applied at test time using all J source estimates to compute soft masks. It is trained with the Adam optimizer [90], a batch size of 16 and a learning rate of 0.001. We train this baseline on the full and the small training set and call the experiments Unet-F and Unet-S, respectively.

6.5 Results and discussion

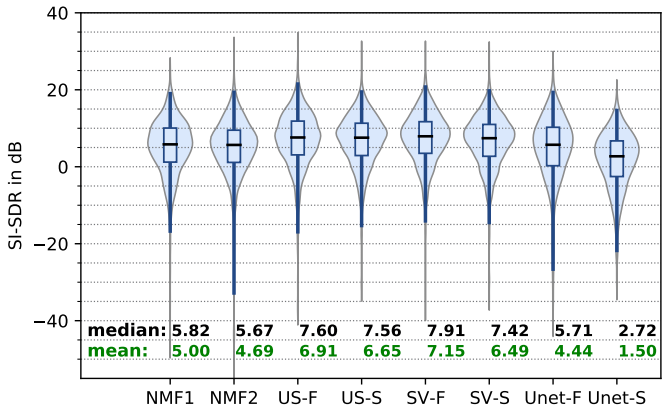
The separation quality was evaluated using the objective metric Scale-Invariant Source-to-Distortion Ratio (SI-SDR) [106]. It is computed on evaluation frames of one second length without overlap as usually done for musical source separation evaluation [185]. The results for the cases of $J = 2$ and $J = 4$ sources are shown in Fig. 6.4 (a) and (b), respectively. The data points for the boxplots and violin plots are the SI-SDR values in dB for all evaluation frames in the test set. Frames in which the target source is silent (the total energy is below 10) are excluded. For methods in which random numbers are involved, the evaluation was run with five different seeds to initialize the pseudorandom number generator. These methods are NMF2 (random initialization of $\mathbf{H}^{\Gamma}$ and $\mathbf{H}^{\Phi}$) and the proposed approach (random white noise) used in experiments US-F, US-S, SV-F, and SV-S.

We conducted two-sided t-tests [131] to assess whether the means of the SI-SDR score distributions are significantly different for each pair of experiments in our study. We used a Levene test [111] to assess whether a pair of SI-SDR score distributions has the same variance or not. If true, the comparison was made with a Student's t-test. If false, Welch's t-test [210] was used. The resulting p-values [131] are shown in Fig. 6.5 (a) and (b) for $J = 2$ and $J = 4$, respectively. Most p-values are extremely small being in the order of 10^{-4} or smaller. This indicates that the corresponding means are significantly different. It can be seen that a few p-values are considerably larger. In this case it is more likely that the true means are not different.

In general, the SI-SDR is higher for the separation of mixtures of two sources compared to the



(a) $J = 2$ sources



(b) $J = 4$ sources

Figure 6.4: Violin plots and boxplots of the SI-SDR values in dB for all evaluation frames. All methods use Wiener filtering for the separation. The boxes extend from the first to the third quartile, the medians are marked with a black horizontal line. The boxplot whiskers (dark blue) extend from the first to the 99th percentile. The violin plots extend over the whole data range. In (b), NMF2 has five outliers between -60 and -80 dB which are not shown.

four sources case. However, the relative performance of the methods is the same for both cases with the exception that Unet-F outperforms NMF1 and NMF2 when $J = 2$ but not when $J = 4$. Listening examples are available online[6].

The proposed unsupervised method (US-F, US-S) performs better than the baselines. Its performance is very close to the one which is reached by the same model trained in a supervised way: SV-F is only slightly better than US-F, while SV-S and US-S have the same performance (p-values of 0.9507 and 0.164 for $J = 2$ and $J = 4$). This means that the proposed method achieves almost the same performance whether isolated target sources are available for training or not. This can be explained by the fact that the F0 information is used very efficiently by the proposed method. The F0 fully parameterizes the harmonic source component $h(t)$ and, hence, defines the corresponding source to a large extent. The DNN has to determine the remaining parameters which, given the F0, can be inferred from the mixture. Hence, isolated source targets do not carry major additional information.

[6] `https://schufo.github.io/umss/`

	NMF1	NMF2	US-F	US-S	SV-F	SV-S	Unet-F	Unet-S
NMF1	-	0.2154	$1.2 \cdot 10^{-164}$	$7.8 \cdot 10^{-72}$	$5.3 \cdot 10^{-193}$	$2.3 \cdot 10^{-69}$	$1.5 \cdot 10^{-31}$	$9.0 \cdot 10^{-16}$
NMF2	0.2154	-	$< 10^{-300}$	$3.2 \cdot 10^{-224}$	$< 10^{-300}$	$1.8 \cdot 10^{-220}$	$6.0 \cdot 10^{-52}$	$5.9 \cdot 10^{-19}$
US-F	$1.2 \cdot 10^{-164}$	$< 10^{-300}$	-	$2.3 \cdot 10^{-70}$	$3.9 \cdot 10^{-4}$	$3.9 \cdot 10^{-70}$	$2.2 \cdot 10^{-24}$	$2.8 \cdot 10^{-275}$
US-S	$7.8 \cdot 10^{-72}$	$3.2 \cdot 10^{-224}$	$2.3 \cdot 10^{-70}$	-	$9.7 \cdot 10^{-109}$	0.9507	0.2858	$3.5 \cdot 10^{-157}$
SV-F	$5.3 \cdot 10^{-193}$	$< 10^{-300}$	$3.9 \cdot 10^{-4}$	$9.7 \cdot 10^{-109}$	-	$3.7 \cdot 10^{-108}$	$6.1 \cdot 10^{-34}$	$< 10^{-300}$
SV-S	$2.3 \cdot 10^{-69}$	$1.8 \cdot 10^{-220}$	$3.9 \cdot 10^{-70}$	0.9507	$3.7 \cdot 10^{-108}$	-	0.3006	$2.3 \cdot 10^{-156}$
Unet-F	$1.5 \cdot 10^{-31}$	$6.0 \cdot 10^{-52}$	$2.2 \cdot 10^{-24}$	0.2858	$6.1 \cdot 10^{-34}$	0.3006	-	$5.7 \cdot 10^{-78}$
Unet-S	$9.0 \cdot 10^{-16}$	$5.9 \cdot 10^{-19}$	$2.8 \cdot 10^{-275}$	$3.5 \cdot 10^{-157}$	$< 10^{-300}$	$2.3 \cdot 10^{-156}$	$5.7 \cdot 10^{-78}$	-

(a) $J = 2$ sources

	NMF1	NMF2	US-F	US-S	SV-F	SV-S	Unet-F	Unet-S
NMF1	-	0.1739	$1.6 \cdot 10^{-18}$	$1.2 \cdot 10^{-14}$	$3.0 \cdot 10^{-23}$	$2.9 \cdot 10^{-12}$	0.0684	$1.5 \cdot 10^{-35}$
NMF2	0.1739	-	$2.1 \cdot 10^{-53}$	$2.0 \cdot 10^{-46}$	$5.3 \cdot 10^{-70}$	$1.2 \cdot 10^{-39}$	0.3357	$1.8 \cdot 10^{-43}$
US-F	$1.6 \cdot 10^{-18}$	$2.1 \cdot 10^{-53}$	-	0.0335	0.0554	$5.2 \cdot 10^{-4}$	$4.0 \cdot 10^{-22}$	$1.3 \cdot 10^{-121}$
US-S	$1.2 \cdot 10^{-14}$	$2.0 \cdot 10^{-46}$	0.0335	-	$2.9 \cdot 10^{-5}$	0.164	$2.0 \cdot 10^{-18}$	$1.5 \cdot 10^{-113}$
SV-F	$3.0 \cdot 10^{-23}$	$5.3 \cdot 10^{-70}$	0.0554	$2.9 \cdot 10^{-5}$	-	$2.5 \cdot 10^{-8}$	$2.2 \cdot 10^{-26}$	$3.0 \cdot 10^{-132}$
SV-S	$2.9 \cdot 10^{-12}$	$1.2 \cdot 10^{-39}$	$5.2 \cdot 10^{-4}$	0.164	$2.5 \cdot 10^{-8}$	-	$4.0 \cdot 10^{-16}$	$1.4 \cdot 10^{-107}$
Unet-F	0.0684	0.3357	$4.0 \cdot 10^{-22}$	$2.0 \cdot 10^{-18}$	$2.2 \cdot 10^{-26}$	$4.0 \cdot 10^{-16}$	-	$2.1 \cdot 10^{-21}$
Unet-S	$1.5 \cdot 10^{-35}$	$1.8 \cdot 10^{-43}$	$1.3 \cdot 10^{-121}$	$1.5 \cdot 10^{-113}$	$3.0 \cdot 10^{-132}$	$1.4 \cdot 10^{-107}$	$2.1 \cdot 10^{-21}$	-

(b) $J = 4$ sources

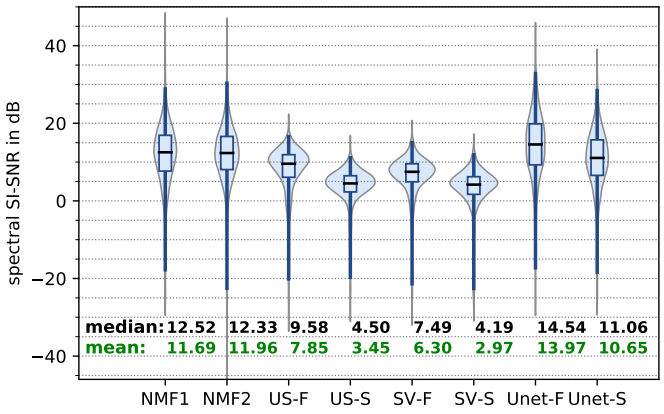
Figure 6.5: The p-values of pair-wise t-tests between the distributions of SI-SDR values for all experiments.

Another interesting observation is that the performance of the proposed method does not drop drastically when the amount of training data is decreased by 97% (US-F vs. US-S and SV-F vs. SV-S). For $J = 2$, a decrease in SI-SDR can be seen but it is much smaller than for the supervised baseline (Unet-F vs. Unet-S). For $J = 4$, the performance difference of the proposed approach is very small when comparing training on the full and the small training set. For the unsupervised version the difference is probably not significant since the p-value of 0.0335 for the comparison of US-F and US-S is larger than most other p-values. In contrast, the SI-SDR of the Unet baseline drops strongly for $J = 4$ as well. This shows that it is beneficial to integrate knowledge in the form of explicit source models in the separation model. The source models limit the output space of the source estimates. It is further narrowed down by the F0 information. This leads to high data efficiency compared to purely data-driven (informed) estimation.

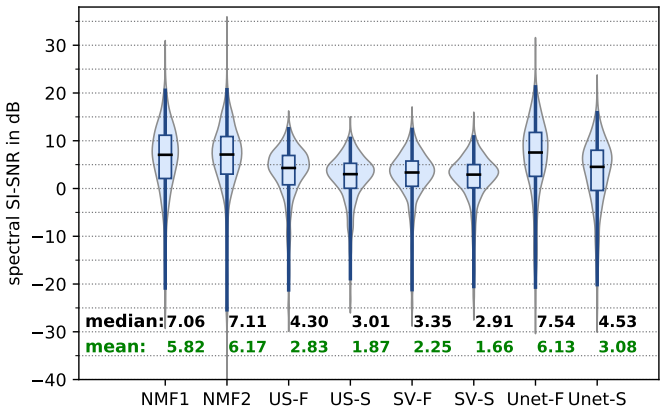
To sum up, the proposed unsupervised model-based deep learning approach to source separation performs better than learning-free and supervised purely data-driven baselines. It is also extremely efficient in learning from data. The method is useful in many scenarios where homogeneous sources need to be separated and/or only a very small amount of data (possibly without ground truth) is available for training. Besides choir separation as in our experiments, such scenarios may be the separation of lead from background vocals or of traditional music with less common instrumentation. Since only mixtures are needed for training, the proposed model may also be trained directly on the mixtures at hand which are to be separated. Given sufficient computational resources, parameter optimization may also be done directly on each test mixture individually, which would make the method learning-free.

6.5.1 Limitations and perspectives

The experimental evaluation showed many advantages of the proposed approach compared to various alternatives. Nevertheless, there are some limitations. As for all F0-informed separation methods, the sources should exhibit mainly harmonic content and be monophonic so that the separation can be guided by the F0 information. It requires that good F0 estimates can be obtained for all sources from the mixture. As shown in the experiments, this is possible with existing methods. Progress in research on multiple F0 estimation may lead to further improvements. An extension of our method to polyphonic sources as well as estimating the F0 jointly with the other source parameters may be an interesting direction for future work. Moreover, audio effects such as reverberation or distortion, which may have been applied to the sources, should be explicitly modeled in the source models and must hence be known beforehand. Lastly, the space complexity grows linearly with the number of sources to be modeled.



(a) $J = 2$ sources



(b) $J = 4$ sources

Figure 6.6: Violin plots and boxplots of the spectral SI-SNR values in dB for all evaluation frames. The source estimates of the methods US-F, US-S, SV-F, and SV-S are the generated signals $\tilde{v}_j$. The other methods used Wiener filtering of the mixture for the separation. The boxes extend from the first to the third quartile, the medians are marked with a black horizontal line. The boxplot whiskers (dark blue) extend from the first to the 99th percentile. The violin plots extend over the whole data range. In (b), NMF2 has five outliers between -60 and -80 dB which are not shown.

In the experiments above, the final source estimates were obtained by Wiener filtering of the

mixture. To this end, soft masks were obtained from the source signals $\tilde{v}_j$ generated by the source models. We also evaluated the quality of the generated signals $\tilde{v}_j$ as source estimates. The metric used for this evaluation was the spectral Scale-Invariant Source-to-Noise Ratio (SI-SNR) [205]. It can be seen as a SI-SDR which is computed on magnitude spectrograms. We used this spectral metric because the phase of the generated signals is known not to be the same as the one of the ground truth signals. This makes a time domain evaluation not applicable. The results are shown in Figure 6.6.

In terms of the spectral SI-SNR, the quality of such source estimates is inferior to the baselines and to $\hat{v}_j$ obtained using soft masks. This is because the synthesis of the signals $\tilde{v}_j$ is less constrained than masking of the mixture. The output of masking is limited by the frequency content of the mixture, since masking can only keep or remove (but not add) such content. In contrast, frequency content which is not present in any source can be contained in $\tilde{v}_j$. In fact, the DNN tends to overestimate the noise content of the sources. While this is clearly audible in $\tilde{v}_j$, no noise is added in $\hat{v}_j$.

Nevertheless, we believe that source estimates generated by parametric models are a worthwhile goal for future research. They provide a complete parameterization of the mixture signal which can be exploited for tasks such as timbre or style transfer, transposition, and melody editing of single sources. We included the generated source signals $\tilde{v}_j$ and their sum $\tilde{x}$ in the audio examples[7]. Moreover, we provide two examples of melody editing for which the mixture parametrization was exploited.

6.6 Conclusion

In this chapter, we presented a method for (musical) audio source separation which overcomes two limitations of state-of-the-art supervised deep learning methods: They do not separate homogeneous sources and require large datasets of mixtures with the individual sources in isolation for training. We proposed a novel unsupervised model-based deep learning approach. It integrates model-based knowledge in the form of differentiable parametric source models in a data-driven method and exploits F0 information. Experiments show that it outperforms learning-free and supervised baselines. Furthermore, the method performs well even when trained on less than three minutes of audio data. It allows to apply powerful deep learning based separation in domains where training data is expensive or nonexistent.

[7] `https://schufo.github.io/umss/`

Chapter 7

Conclusion and Future Work

Summary

This chapter concludes the dissertation. A summary of the main results and contributions is provided followed by a discussion of the limitations and resulting directions for future work.

Contents

In this dissertation we have explored different ways to integrate prior knowledge into data-driven approaches to audio source separation. The aim was to reduce the dependence on labeled data and to exploit complementary data which may be available.

First, we focused on exploiting text as side information for supervised singing voice separation with deep learning. To this end, we also proposed a novel lyrics alignment approach. We found that while using text information can lead to improvements in difficult scenarios, it also brings back some negative aspects of knowledge-driven approaches such as low flexibility.

In a second step, we explored unsupervised learning for source separation in order to leverage unlabeled data which is more readily available than labeled data. Learning only from mixtures was enabled by exploiting F0 trajectories for each source as well as parametric generative source models which were integrated in the training procedure.

In the following, we summarize the contributions of this thesis and possible directions for future work.

7.1 Summary of contributions

7.1.1 Weakly informed audio source separation

Signal related side information for musical source separation such as musical scores or lyric transcripts are widely available but usually not aligned with the mixture to be separated. In Chapter 4

we proposed a novel DNN architecture for informed audio source separation which exploits non-aligned side information. We showed that an attention mechanism can be used between two encoders to learn the alignment when the network is trained only with a separation objective. As a proof of concept, the approach was evaluated on a singing voice separation task using synthetic side information with different levels of expressiveness. Further experimental evaluation was performed using text as side information for speech-music separation with low SNR. We found that joint alignment and separation led to benefits for both tasks. The separation component facilitated text alignment despite the low SNR and the phonetic information in the text led to small improvements in separation quality. However, pre-aligned text led to stronger improvements of the separation. The proposed approach builds the basis for a new lyrics alignment method proposed in Chapter 5. We also presented two new evaluation metrics for audio source separation which complement the standard metrics on frames with a silent target or estimated source.

7.1.2 Phoneme level lyrics alignment with DTW-attention

While text-to-speech alignment was performed successfully with the alignment approach introduced in Chapter 4, aligning phonemes with singing voice required further improvements. Therefore, DTW-attention, which is a combination of dynamic time warping and attention was proposed. It incorporates the prior knowledge that text and audio sequences follow a left-to-right temporal structure. The result was a new lyrics alignment method which works well not only for solo singing but also with mixtures. Experimental results showed that DTW-attention is more data-efficient than CTC-based training. Hence, it is an attractive alternative especially for languages where large datasets of audio and text pairs are not available.

For this work, the MUSDB dataset was extended by lyrics transcripts and further annotations which are made publicly available[1]. The provided data can be used for research on automatic lyrics alignment and transcription, text-informed singing voice separation, or singing voice synthesis and analysis.

7.1.3 Text-informed singing voice separation

We provided new insights into the usage of text as side information for singing voice separation with deep learning. Using the proposed joint approach to text-to-audio alignment and text-informed voice separation led to a robust lyrics alignment method but did not lead to improvements for the separation. We found that lyrics should be aligned first and can then be used to inform the separation. Sequences of phonemes as side information are used by the proposed model as a strong prior regarding the spectral shape of the voice estimate. This helps to preserve the phonetic properties in challenging conditions. However, it is required that the text transcript and the translation into phonemes accurately represent the voice signal and that the alignment is precise. Otherwise, the separation performance is negatively influenced by the text prior. It must also be noted that text information is most useful if exactly one singer is present in the mixture. There are specialized use cases where text is a valuable source of information. For example, to separate unvoiced sounds from drum sounds. However, the efforts required to use text effectively do not justify its usage in most singing voice separation problems.

[1] `https://doi.org/10.5281/zenodo.3989267`

7.1.4 Unsupervised audio source separation

We presented an unsupervised deep learning approach to audio source separation. It exploits side information in the form of F0 trajectories and model-based knowledge in the form of parametric source models. It can be trained using only a small amount of mixture signals and does not require separate source signals. In experiments on vocal ensemble separation with two and four singers, the proposed method outperformed F0-informed learning-free approaches based on NMF and an F0-informed supervised deep learning baseline. The high data-efficiency allows its application to the numerous music genres where separate sources for training do not exist or are expensive to obtain. Furthermore, due to the parametric source models it provides a parameterization of the mixture signal which can be exploited for downstream tasks such as timbre or style transfer, transposition, or editing of the melody of single sources.

7.2 Future work

We presented some strategies to integrate prior knowledge into data-driven audio source separation methods with a focus on music signals. Despite promising results, there remain several limitations which may inspire directions for future work.

We showed that text as side information can lead to small improvements in singing voice separation with deep learning compared to a non-informed approach with an identical DNN. While this is an encouraging result, it does not yet fully justify the usage of text as side information. Given that purely data-driven supervised deep learning approaches can reach better performance than our informed approach [97], further analyses are necessary. They should concern the remaining artefacts or failure modes of state-of-the-art data-driven approaches. This may even lead to new evaluation metrics. One example would be to focus on the separation of different phoneme classes such as vowels, unvoiced fricatives, and stops from drum sounds with similar properties. This might allow to define more precisely the scenarios in which lyrics as side information have most value. Another aspect that may be improved is the dependence on accurate alignments which is a limitation of the proposed approach. One could try to exploit the text in a looser sense, for example, using only information about the succession of voiced and unvoiced sounds.

An advantage of the proposed lyrics alignment approach is that it requires only a small amount of training data. However, it also makes an additional demand on the training data: separate voice recordings must be available along with mixture signals because the model is trained with a source separation objective. If alignment with solo singing is the target application and a small amount of solo singing data is available for training, one can produce artificial mixtures using any available instrumental music. Future research may explore other training objectives such as the reconstruction of the observed text and audio sequences which may be done combining attention and the CTC loss. First steps in this direction have been made recently [193].

Furthermore, the unsupervised source separation approach presented in Chapter 6 may be extended in several ways. For example, effects such as reverberation could be modeled explicitly with an additional filter to make the source models more general. However, it should be noted that estimating the reverberation parameters of several sound sources from a mixture in one-shot fashion is extremely challenging. A starting point could be to assume that the mixtures to be separated are available for the unsupervised training procedure and that the sources have the same reverberation parameters in all mixtures. In this case, the parameters can be learned from

several mixtures as it is done in [42] for single monophonic sources.

Another aspect which offers room for improvement is the estimation of the noise component in each source model. The F0 information concerns only the harmonic component and, hence, there remains ambiguity in the assignment of a share of the noise observed in the mixture to each source. A solution could be to formulate a noise model which depends on the harmonic content to a certain degree.

In our work, we used an external multiple F0 estimator to provide the F0 trajectories for the unsupervised learning approach. A challenging yet very interesting extension would be to estimate the F0s jointly with the other synthesis parameters guided by the reconstruction loss. In order to achieve separation one would also need to include the source assignment step. It seems to be promising to further explore the synergies between source separation, fundamental frequency estimation and F0-to-source assignment.

Moreover, we think that synthesis-based approaches to musical source separation have not gained enough attention yet. The idea was explored in [12] and [13] in a supervised setting and we proposed an unsupervised approach. However, the quality of synthesized source estimates is still lower than of those obtained by filtering the mixture. Synthesizing source signals is a flexible way to integrate domain knowledge in data-driven source separation and leads to interpretable and modifiable estimates. Furthermore, synthesized signals do not contain interfering sources and usually have a coherent and perceptually correct phase.

In general, the dependence on training data remains a main obstacle for progress in musical source separation and we believe that integrating knowledge in data-driven methods is one promising avenue to tackle this issue. Alternatively, it may be worthwhile to explore strategies for self-supervised learning of representations which can be used for source separation or to exploit available data for related tasks in multi-task settings or via pre-training.

Bibliography

[1] Smule sing! 300x30x2 dataset. `https://ccrma.stanford.edu/damp/`. Accessed: 2020-01-11.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[3] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[4] I. P. Association, I. P. A. Staff, et al. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[5] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):191–199, 2005.

[8] Y. Bengio, P. Frasconi, and P. Simard. The problem of learning long-term dependencies in recurrent networks. In *IEEE International Conference on Neural Networks*, pages 1183–1188, 1993.

[9] M. Blaauw and J. Bonada. A neural parametric singing synthesizer. In *Proceedings of Interspeech*, 2017.

[10] J. Bonada and X. Serra. Synthesis of the singing voice by performance sampling and spectral models. *IEEE Signal Processing Magazine*, 24(2):67–79, 2007.

[11] G. Bordel, M. Penagarikano, L. J. Rodríguez-Fuentes, A. Álvarez, and A. Varona. Probabilistic kernels for improved text-to-speech alignment in long audio tracks. *IEEE Signal Processing Letters*, 23(1):126–129, 2015.

[12] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez. A vocoder based method for singing voice extraction. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 990–994, 2019.

[13] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez. Content based singing voice extraction from a musical mixture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 781–785, 2020.

[14] P. Chandna, M. Miron, J. Janer, and E. Gómez. Monoaural audio source separation using deep convolutional neural networks. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 258–266. Springer, 2017.

[15] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath. An attentive survey of attention models. *arXiv preprint arXiv:1904.02874*, 2019.

[16] S. E. Chazan, S. Gannot, and J. Goldberger. A phoneme-based pre-training approach for deep neural network with application to speech enhancement. In *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*, pages 1–5, 2016.

[17] J. Chen and D. Wang. Long short-term memory for speaker generalization in supervised speech separation. *The Journal of the Acoustical Society of America*, 141(6):4705–4714, 2017.

[18] E. Chew and X. Wu. Separating voices in polyphonic music: A contig mapping approach. In *International Symposium on Computer Music Modeling and Retrieval*, pages 1–20. Springer, 2004.

[19] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585, 2015.

[21] W. C. Chu. *Speech coding algorithms: foundation and evolution of standardized coders*. John Wiley & Sons, 2004.

[22] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[23] A. Cohen-Hadria, A. Roebel, and G. Peeters. Improving singing voice separation using deep U-Net and Wave-U-Net with data augmentation. In *Proceedings of the IEEE European Signal Processing Conference*, pages 1–5, 2019.

[24] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

[25] H. Cuesta, E. Gómez Gutiérrez, A. Martorell Domínguez, and F. Loáiciga. Analysis of intonation in unison choir singing. In *Proceedings of the International Conference of Music Perception and Cognition*, 2018.

[26] H. Cuesta, B. McFee, and E. Gómez. Multiple f0 estimation in vocal ensembles using convolutional neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 302–309, 2020.

[27] M. Cuturi and M. Blondel. Soft-DTW: a differentiable loss function for time-series. In *Proceedings of the International Conference on Machine Learning*, pages 894–903, 2017.

[28] A. Défossez, N. Usunier, L. Bottou, and F. Bach. Demucs: Deep extractor for music sources with extra unlabeled data remixed. *arXiv preprint arXiv:1909.01174*, 2019.

[29] G. Degottex, P. Lanchantin, A. Roebel, and X. Rodet. Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis. *Speech Communication*, 55(2):278–294, 2013.

[30] C. Demir, M. Saraclar, and A. T. Cemgil. Single-channel speech-music separation for robust asr with mixture models. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(4):725–736, 2012.

[31] E. Demirel, S. Ahlbäck, and S. Dixon. Automatic lyrics transcription using dilated convolutional neural networks with self-attention. In *International Joint Conference on Neural Networks*, pages 1–8, 2020.

[32] B. E. Dresher. The phoneme. In *The Blackwell companion to phonology*, pages 1–26. John Wiley & Sons, 2011.

[33] L. Drude, D. Hasenklever, and R. Haeb-Umbach. Unsupervised training of a deep clustering model for multichannel blind source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 695–699, 2019.

[34] X. Du, Z. Yu, B. Zhu, and X. Chen. Caspernet for cover song identification. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2020.

[35] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang. The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–9, 2013.

[36] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi. Unsupervised single-channel music source separation by average harmonic structure modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(4):766–778, 2008.

[37] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[38] N. Q. Duong, E. Vincent, and R. Gribonval. Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1830–1840, 2010.

[39] J. Durbin. The fitting of time-series models. *Revue de l'Institut International de Statistique*, pages 233–244, 1960.

[40] J.-L. Durrieu, B. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1180–1191, 2011.

[41] G. B. Dzhambazov and X. Serra. Modeling of phoneme durations for alignment between polyphonic audio and lyrics. In *Sound and Music Computing Conference*, 2015.

[42] J. Engel, C. Gu, A. Roberts, et al. DDSP: Differentiable digital signal processing. In *Proceedings of the International Conference on Learning Representations*, 2019.

[43] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne. Self-supervised pitch detection by inverse audio synthesis. *Workshop on Self-supervision in Audio and Speech at the International Conference on Machine Learning*, 2020.

[44] D. Erro, I. Sainz, E. Navas, and I. Hernaez. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):184–194, 2013.

[45] S. Ewert and M. Müller. Using score-informed constraints for NMF-based source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 129–132, 2012.

[46] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, 2014.

[47] S. Ewert and M. B. Sandler. Structured dropout for weak label and multi-instance learning and its application to score-informed source separation. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2277–2281, 2017.

[48] G. Fant. *Acoustic theory of speech production*. Number 2. Walter de Gruyter, 1970.

[49] C. Févotte, L. Daudet, S. J. Godsill, and B. Torrésani. Sparse regression with structured priors: Application to audio denoising. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages III–III, 2006.

[50] D. Fitzgerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects*, volume 13, 2010.

[51] J. L. Flanagan. *Speech analysis synthesis and perception*, volume 3. Springer Science & Business Media, 2013.

[52] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno. LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1252–1261, 2011.

[53] J. S. Garofolo. TIMIT acoustic phonetic continuous speech corpus. *Linguistic Data Consortium*, 1993.

[54] T. Gerber, M. Dutasta, L. Girin, and C. Févotte. Professionally-produced music separation guided by covers. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2012.

[55] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

[56] R. Gong and X. Serra. Singing voice phoneme segmentation by hierarchically inferring syllable and phoneme onset positions. In *Proceedings of Interspeech*, pages 716–720, 2018.

[57] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[58] M. Gover and P. Depalle. Score-informed source separation of choral music. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 231–239, 2020.

[59] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 369–376, 2006.

[60] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[61] C. Gupta, H. Li, and Y. Wang. Automatic pronunciation evaluation of singing. In *Proceedings of Interspeech*, pages 1507–1511, 2018.

[62] C. Gupta, R. Tong, H. Li, and Y. Wang. Semi-supervised lyrics and solo-singing alignment. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2019.

[63] C. Gupta, E. Yılmaz, and H. Li. AutoLyrixAlign: Pre-trained model and script to automatically align lyrics to polyphonic audio. `https://github.com/chitralekha18/AutoLyrixAlign`. Accessed: 2020-12-22.

[64] C. Gupta, E. Yılmaz, and H. Li. Automatic lyrics transcription in polyphonic music: Does background music help? *arXiv preprint arXiv:1909.10200*, 2019.

[65] A. Hannun. Sequence modeling with CTC. `https://distill.pub/2017/ctc`, 2017. Accessed: 2021-04-28.

[66] J. K. Hansen. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *Proceedings of the Sound and Music Computing Conference*, pages 494–499, 2012.

[67] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel. Sequence-to-sequence piano transcription with transformers. *Proceedings of the International Society for Music Information Retrieval Conference*, 2021.

[68] T. Heittola, A. Klapuri, and T. Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2009.

[69] R. Hennequin, J. J. Burred, S. Maller, and P. Leveau. Speech-guided source separation using a pitch-adaptive guide signal model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6672–6676, 2014.

[70] R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 45–48, 2011.

[71] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam. Spleeter: A fast and state-of-the art music source separation tool with pre-trained models. Late-Breaking/Demo ISMIR 2019, November 2019. Deezer Research.

[72] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 31–35, 2016.

[73] G. Hilkhuysen, N. Gaubitch, M. Brookes, and M. Huckvale. Effects of noise suppression on intelligibility. ii: An attempt to validate physical metrics. *The Journal of the Acoustical Society of America*, 135(1):439–450, 2014.

[74] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91(1), 1991.

[75] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[76] P. Horowitz and W. Hill. *The art of electronics*. Cambridge university press Cambridge, 2002.

[77] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 57–60, 2012.

[78] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 477–482, 2014.

[79] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2136–2147, 2015.

[80] Y.-N. Hung and A. Lerch. Multitask learning for instrument activation aware music source separation. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2020.

[81] Y.-N. Hung, G. Wichern, and J. Le Roux. Transcription is all you need: Learning to separate musical mixtures with score as supervision. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 46–50, 2021.

[82] Y. Ikemiya, K. Itoyama, and K. Yoshii. Singing voice separation and vocal f0 estimation based on mutual combination of robust principal component analysis and subharmonic summation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2084–2095, 2016.

[83] F. Itakura. Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America*, 57(S1):S35–S35, 1975.

[84] J. Janer and R. Marxer. Separation of unvoiced fricatives in singing voice mixtures with semi-supervised NMF. In *Proceedings of the International Conference on Digital Audio Effects*, pages 2–5, 2013.

[85] A. Jansson, R. M. Bittner, S. Ewert, and T. Weyde. Joint singing voice separation and f0 estimation with deep U-Net architectures. In *Proceedings of the IEEE European Signal Processing Conference*, pages 1–5, 2019.

[86] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde. Singing voice separation with deep U-Net convolutional networks. *Proceedings of the International Society for Music Information Retrieval Conference*, pages 23–27, 2017.

[87] C.-B. Jeon, H.-S. Choi, and K. Lee. Exploring aligned lyrics-informed singing voice separation. *arXiv preprint arXiv:2008.04482*, 2020.

[88] P. Kabal and R. P. Ramachandran. The computation of line spectral frequencies using Chebyshev polynomials. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 34(6):1419–1426, 1986.

[89] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan. Sailalign: Robust long speech-text alignment. In *Proceedings of the Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[90] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[91] K. Kinoshita, M. Delcroix, A. Ogawa, and T. Nakatani. Text-informed speech enhancement with deep neural networks. *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[92] D. Kitamura, N. Ono, H. Saruwatari, Y. Takahashi, and K. Kondo. Discriminative and reconstructive basis training for audio source separation with semi-supervised nonnegative matrix factorization. In *IEEE International Workshop on Acoustic Signal Enhancement*, pages 1–5, 2016.

[93] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 216–221, 2006.

[94] D. H. Klatt and L. C. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *The Journal of the Acoustical Society of America*, 87(2):820–857, 1990.

[95] A. Klautau. ARPABET and the TIMIT alphabet. `https://web.archive.org/web/20160603180727/http://www.laps.ufpa.br/aldebaro/papers/ak_arpabet01.pdf`, 2001.

[96] 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*, 25(10):1901–1913, 2017.

[97] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang. Decoupling magnitude and phase estimation with deep resunet for music source separation. *arXiv preprint arXiv:2109.05418*, 2021.

[98] M. Kowalski and B. Torrésani. Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video processing*, 3(3):251–264, 2009.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[100] A. M. Kruspe. Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 358–364, 2016.

[101] A. Kumar and B. Raj. Audio event detection using weakly labeled data. *Proceedings of the ACM International Conference on Multimedia*, pages 1038–1047, 2016.

[102] C. Laroche, H. Papadopoulos, M. Kowalski, and G. Richard. Genre specific dictionaries for harmonic/percussive source separation. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2016.

[103] J. Laroche, Y. Stylianou, and E. Moulines. HNM: A simple, efficient harmonic+ noise model for speech. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 169–172, 1993.

[104] L. Le Magoarou, A. Ozerov, and N. Q. Duong. Text-informed audio source separation using nonnegative matrix partial co-factorization. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2013.

[105] L. Le Magoarou, A. Ozerov, and N. Q. Duong. Text-informed audio source separation. example-based approach using non-negative matrix partial co-factorization. *Journal of Signal Processing Systems*, 79(2):117–131, 2015.

[106] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. SDR–half-baked or well done? In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 626–630, 2019.

[107] Y. LeCun et al. Generalization and network design strategies. *Connectionism in Perspective*, 19:143–155, 1989.

[108] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[109] Y.-S. Lee, C.-Y. Wang, S.-F. Wang, J.-C. Wang, and C.-H. Wu. Fully complex deep neural network for phase-incorporating monaural source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 281–285, 2017.

[110] S. Leglaive, L. Girin, and R. Horaud. Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 101–105, 2019.

[111] H. Levene. Robust tests for equality of variances. *Contributions to probability and statistics. Essays in honor of Harold Hotelling*, pages 279–292, 1961.

[112] N. Levinson. The Wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25(1-4):261–278, 1946.

[113] K. W. E. Lin and M. Goto. Zero-mean convolutional network with data augmentation for sound level invariant singing voice separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 251–255, 2019.

[114] A. Liutkus and R. Badeau. Generalized wiener filtering with fractional power spectrograms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 266–270, 2015.

[115] A. Liutkus, J.-L. Durrieu, L. Daudet, and G. Richard. An overview of informed audio source separation. *International Workshop on Image Analysis for Multimedia Interactive Services*, pages 1–4, 2013.

[116] F. Lluís, J. Pons, and X. Serra. End-to-end music source separation: is it possible in the waveform domain? *arXiv preprint arXiv:1810.12187*, 2018.

[117] Y. Luo and N. Mesgarani. Conv-Tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, 2019.

[118] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[119] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning*, volume 30, 2013.

[120] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[121] M. Mauch, H. Fujihara, and M. Goto. Integrating additional chord information into HMM-based lyrics-to-audio alignment. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 20(1):200–210, 2011.

[122] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In *Proceedings of Interspeech*, pages 498–502, 2017.

[123] A. McLeod and M. Steedman. HMM-based voice separation of MIDI performance. *Journal of New Music Research*, 45(1):17–26, 2016.

[124] I. V. McLoughlin. Line spectral pairs. *Signal Processing*, 88(3):448–467, 2008.

[125] A. Mesaros and T. Virtanen. Adaptation of a speech recognizer for singing voice. In *Proceedings of the IEEE European Signal Processing Conference*, pages 1779–1783, 2009.

[126] G. Meseguer-Brocal and G. Peeters. Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations. *arXiv preprint arXiv:1907.01277*, 2019.

[127] G. Meseguer-Brocal and G. Peeters. Content based singing voice source separation via strong conditioning using aligned phonemes. *arXiv preprint arXiv:2008.02070*, 2020.

[128] S. I. Mimilakis, K. Drossos, J. F. Santos, G. Schuller, T. Virtanen, and Y. Bengio. Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 721–725, 2018.

[129] M. Miron, J. Janer Mestres, and E. Gómez Gutiérrez. Monaural score-informed source separation for classical music using convolutional neural networks. *Proceedings of the International Society for Music Information Retrieval Conference*, pages 55–62, 2017.

[130] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[131] D. S. Moore and S. Kirkland. *The basic practice of statistics*, volume 2. WH Freeman New York, 2007.

[132] P. J. Moreno, C. Joerg, J.-M. V. Thong, and O. Glickman. A recursive algorithm for the forced alignment of very long audio segments. In *Fifth International Conference on Spoken Language Processing*, 1998.

[133] M. Morise, F. Yokomori, and K. Ozawa. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, 99(7):1877–1884, 2016.

[134] M. Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.

[135] G. Mysore and M. Sahani. Variational inference in non-negative factorial hidden Markov models for efficient audio source separation. *arXiv preprint arXiv:1206.6468*, 2012.

[136] T. Nakamura and H. Kameoka. Harmonic-temporal factor decomposition for unsupervised monaural separation of harmonic sounds. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:68–82, 2020.

[137] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias. Unsupervised audio source separation using generative priors. In *Proceedings of Interspeech*, pages 2657–2661, 2020.

[138] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii. Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 161–165, 2019.

[139] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[140] A. Ozerov and C. Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2009.

[141] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard. Informed source separation: source coding meets source separation. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 257–260, 2011.

[142] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot. One microphone singing voice separation using source-adapted models. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 90–93, 2005.

[143] 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*, 20(4):1118–1133, 2011.

[144] H. Papadopoulos and D. P. Ellis. Music-content-adaptive robust principal component analysis for a semantically consistent separation of foreground and background in music audio signals. In *Proceedings of the International Conference on Digital Audio Effects*, 2014.

[145] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[146] D. Petermann, P. Chandna, H. Cuesta, J. Bonada, and E. Gomez. Deep learning based source separation applied to choir ensembles. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2020.

[147] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The Kaldi speech recognition toolkit. In *Workshop on Automatic Speech Recognition and Understanding*, 2011.

[148] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019.

[149] L. Rabiner and R. Schafer. *Theory and applications of digital speech processing*. Prentice Hall Press, 2010.

[150] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck. Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the International Conference on Machine Learning*, pages 2837–2846, 2017.

[151] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner. The MUSDB18 corpus for music separation. `https://doi.org/10.5281/zenodo.1117372`, Dec. 2017.

[152] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo. An overview of lead and accompaniment separation in music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(8):1307–1335, 2018.

[153] Z. Rafii and B. Pardo. Repeating pattern extraction technique (REPET): A simple method for music/voice separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1):73–84, 2012.

[154] A. Rao and P. K. Ghosh. SFNet: A computationally efficient source filter model based neural speech synthesis. *IEEE Signal Processing Letters*, 27:1170–1174, 2020.

[155] G. Richard and C. d'Alessandro. Analysis/synthesis and modification of the speech aperiodic component. *Speech Communication*, 19(3):221–244, 1996.

[156] 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 749–752, 2001.

[157] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[158] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[159] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4280–4284, 2015.

[160] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[161] D. Samuel, A. Ganeshan, and J. Naradowsky. Meta-learning extractors for music source separation. *arXiv preprint arXiv:2002.07016*, 2020.

[162] J. Schlüter. Learning to pinpoint singing voice from weakly labeled examples. *Proceedings of the International Society for Music Information Retrieval Conference*, pages 44–50, 2016.

[163] J. Schlüter and T. Grill. Exploring data augmentation for improved singing voice detection with neural networks. *Proceedings of the International Society for Music Information Retrieval Conference*, pages 121–126, 2015.

[164] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[165] R. Schramm, A. McLeod, M. Steedman, E. Benetos, et al. Multi-pitch detection and voice assignment for a cappella recordings of multiple singers. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2017.

[166] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau. Weakly informed audio source separation. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019.

[167] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau. Joint phoneme alignment and text-informed speech separation on highly corrupted speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[168] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau. Phoneme level lyrics alignment and text-informed singing voice separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[169] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[170] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

[171] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo. Bootstrapping unsupervised deep music separation from primitive auditory grouping principles. *Workshop on Self-supervision in Audio and Speech at the International Conference on Machine Learning*, 2020.

[172] X. Serra and J. Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.

[173] B. Sharma, C. Gupta, H. Li, and Y. Wang. Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 396–400, 2019.

[174] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis. Model-based deep learning. *arXiv preprint arXiv:2012.08405*, 2020.

[175] U. Şimşekli and A. T. Cemgil. Score guided musical source separation using generalized coupled tensor factorization. In *Proceedings of the IEEE European Signal Processing Conference*, pages 2639–2643, 2012.

[176] P. Smaragdis and G. J. Mysore. Separation by "humming": User-guided sound extraction from monophonic mixtures. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 69–72, 2009.

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

[178] J. O. Smith. *Spectral Audio Signal Processing: Frequency sampling method*. W3K Publishing, 2011. `https://ccrma.stanford.edu/~jos/sasp/Frequency_Sampling_Method_FIR.html`.

[179] J. O. Smith. *Spectral Audio Signal Processing: Generalized Window Method*. W3K Publishing, 2011. `https://ccrma.stanford.edu/~jos/sasp/Generalized_Window_Method.html`.

[180] F. Soong and B. Juang. Line spectrum pair (LSP) and speech data compression. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 9, pages 37–40, 1984.

[181] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *Advances in Neural Information Processing Systems*, pages 2377–2385, 2015.

[182] D. Stoller, S. Durand, and S. Ewert. End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model. *arXiv preprint arXiv:1902.06797*, 2019.

[183] D. Stoller, S. Ewert, and S. Dixon. Jointly detecting and separating singing voice: A multi-task approach. *International Conference on Latent Variable Analysis and Signal Separation*, pages 329–339, 2018.

[184] D. Stoller, S. Ewert, and S. Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018.

[185] F.-R. Stöter, A. Liutkus, and N. Ito. The 2018 signal separation evaluation campaign. *International Conference on Latent Variable Analysis and Signal Separation*, pages 293–305, 2018.

[186] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji. Open-Unmix - a reference implementation for music source separation. *Journal of Open Source Software*, 4(41):1667, 2019.

[187] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[188] F.-R. Stöter and A. Liutkus. museval 0.3.0. `https://doi.org/10.5281/zenodo.3376621`, Aug. 2019.

[189] 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*, 19(7):2125–2136, 2011.

[190] N. Takahashi and Y. Mitsufuji. Multi-scale multi-band densenets for audio source separation. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 21–25, 2017.

[191] N. Takahashi and Y. Mitsufuji. D3net: Densely connected multidilated densenet for music source separation. *arXiv preprint arXiv:2010.01733*, 2020.

[192] N. Takahashi, M. K. Singh, S. Basak, P. Sudarsanam, S. Ganapathy, and Y. Mitsufuji. Improving voice separation by incorporating end-to-end speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 41–45, 2020.

[193] Y. Teytaut and A. Roebel. Phoneme-to-audio alignment with recurrent neural networks for speaking and singing voice. In *Proceedings of Interspeech*, pages 61–65, 2021.

[194] A. Tjandra, S. Sakti, and S. Nakamura. Local monotonic attention mechanism for end-to-end speech and language processing. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 431–440, 2017.

[195] E. Tzinis, S. Venkataramani, and P. Smaragdis. Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 81–85, 2019.

[196] S. Uhlich, F. Giron, and Y. Mitsufuji. Deep neural network based instrument extraction from music. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2135–2139, 2015.

[197] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 261–265, 2017.

[198] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. D'Alché-Buc. Multilingual lyrics-to-audio alignment. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2020.

[199] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[200] S. Vembu and S. Baumann. Separation of vocals from polyphonic audio recordings. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 337–344, 2005.

[201] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot. From blind to guided audio source separation: How models and side information can improve the separation of sound. *IEEE Signal Processing Magazine*, 31(3):107–115, 2014.

[202] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

[203] E. Vincent, T. Virtanen, and S. Gannot. *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

[204] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, 4(1):52–57, 1968.

[205] T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007.

[206] T. Virtanen, A. Mesaros, and M. Ryynänen. Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music. *Proceedings of Interspeech*, pages 17–22, 2008.

[207] D. Wang and J. Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.

[208] X. Wang, S. Takaki, and J. Yamagishi. Neural source-filter-based waveform model for statistical parametric speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5916–5920, 2019.

[209] Z.-Q. Wang, Y. Zhao, and D. Wang. Phoneme-specific speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 146–150, 2016.

[210] B. L. Welch. The generalization of 'Student's' problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35, 1947.

[211] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[212] S. Wisdom, A. Jansen, R. J. Weiss, H. Erdogan, and J. R. Hershey. Sparse, efficient, and semantic mixture invariant training: Taming in-the-wild unsupervised sound separation. *arXiv preprint arXiv:2106.00847*, 2021.

[213] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey. Unsupervised sound separation using mixture invariant training. *arXiv preprint arXiv:2006.12701*, 2020.

[214] N. Xie, G. Ras, M. van Gerven, and D. Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020.

[215] Y.-H. Yang. On sparse and low-rank matrix decomposition for singing voice separation. In *Proceedings of the ACM International Conference on Multimedia*, pages 757–760, 2012.

[216] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational intelligenCe magazine*, 13(3):55–75, 2018.

[217] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 241–245, 2017.

[218] W. Zhang, Z. Chen, and F. Yin. Multi-pitch estimation of polyphonic music based on pseudo two-dimensional spectrum. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2095–2108, 2020.

[219] Y. Zhang, Y. Liu, and D. Wang. Complex ratio masking for singing voice separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 41–45, 2021.

ECOLE
DOCTORALE

Titre : Séparation de sources audio informée par apprentissage profond avec des données limitées

Mots clés : Apprentissage profond, traitement du signal, séparation de sources audio

Résumé : La séparation de sources audio consiste à estimer les signaux individuels de plusieurs sources sonores lorsque seul leur mélange peut être observé. Des réseaux neuronaux profonds entraînés de manière supervisée permettent d'obtenir des résultats de l'état de l'art pour les signaux musicaux. Ils nécessitent de grandes bases de données composées de mélanges pour lesquels les signaux des sources cibles sont disponibles de manière isolée. Cependant, il est difficile d'obtenir de tels bases de données car les enregistrements musicaux sont soumis à des restrictions de droits d'auteur et les enregistrements d'instruments isolés n'existent pas toujours. Dans cette thèse, nous explorons l'utilisation d'informations supplémentaires pour la séparation de sources par apprentissage profond, afin de s'affranchir d'une quantité limitée de données disponibles.
D'abord, nous considérons un cadre supervisé avec seulement une petite quantité de données disponibles. Nous étudions dans quelle mesure la séparation de la voix chantée peut être améliorée lorsqu'elle est informée par la transcription des paroles. Nous proposons un nouveau modèle d'apprentissage profond pour la séparation de sources informée. Ce modèle permet d'aligner le texte et l'audio pendant la séparation grâce à un nouveau mécanisme d'attention monotone. La qualité de l'alignement des paroles est compétitive par rapport à l'état de l'art, alors qu'une quantité plus faible de données est utilisée. Nous constatons que l'exploitation des phonèmes alignés peut améliorer la séparation de la voix chantée, mais un alignement précis et des transcriptions exactes sont nécessaires. Enfin, nous considérons un scénario où seuls des mélanges sont disponibles pour l'apprentissage. Nous proposons une nouvelle approche d'apprentissage profond non supervisé. Elle exploite les informations sur les fréquences fondamentales (F0) des sources. La méthode intègre les connaissances du domaine sous la forme de modèles de sources paramétriques dans le réseau neuronal profond. L'évaluation expérimentale montre que la méthode surpasse les méthodes sans apprentissage basées sur la factorisation de matrices non négatives, ainsi qu'une approche d'apprentissage profond supervisé. La méthode proposée est extrêmement efficace en terme de données. Elle rend la séparation de sources par apprentissage profond exploitable dans des domaines où les données étiquetées sont coûteuses ou inexistantes.

Title : Informed Audio Source Separation with Deep Learning in Limited Data Settings

Keywords : Deep learning, signal processing, audio source separation

Abstract : Audio source separation is the task of estimating the individual signals of several sound sources when only their mixture can be observed. State-of-the-art performance for musical mixtures is achieved by Deep Neural Networks (DNN) trained in a supervised way. They require large and diverse datasets of mixtures along with the target source signals in isolation. However, it is difficult and costly to obtain such datasets because music recordings are subject to copyright restrictions and isolated instrument recordings may not always exist.
In this dissertation, we explore the usage of prior knowledge for deep learning based source separation in order to overcome data limitations.
First, we focus on a supervised setting with only a small amount of available training data. We investigate to which extent singing voice separation can be improved when it is informed by lyrics transcripts. To this end, a novel deep learning model for informed source separation is proposed. It aligns text and audio during the separation using a novel monotonic attention mechanism. The lyrics alignment performance is competitive with state-of-the-art methods while a smaller amount of training data is used. We find that exploiting *aligned* phonemes can improve singing voice separation, but precise alignments and accurate transcripts are required.
Finally, we consider a scenario where only mixtures but no isolated source signals are available for training. We propose a novel unsupervised deep learning approach to source separation. It exploits information about the sources' fundamental frequencies (F0). The method integrates domain knowledge in the form of parametric source models into the DNN. Experimental evaluation shows that the proposed method outperforms F0-informed learning-free methods based on non-negative matrix factorization and a F0-informed supervised deep learning baseline. Moreover, the proposed method is extremely data-efficient. It makes powerful deep learning based source separation usable in domains where labeled training data is expensive or non-existent.

Institut Polytechnique de Paris
91120 Palaiseau, France