NORTHWESTERN UNIVERSITY

Auditory-inspired Approaches to Audio Representation and Analysis for Machine Hearing

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Audio Signal Processing

By

Fatemeh Pishdadian

EVANSTON, ILLINOIS

November 2020

ABSTRACT

Auditory-inspired Approaches to Audio Representation and Analysis for Machine Hearing

Fatemeh Pishdadian

The study and design of machines that are able to analyze the auditory scene and organize sound into parts that are perceptually meaningful to humans is referred to as *machine hearing*. Such machines are expected to distinguish between different sound categories (e.g., speech, music, background noise), focus on a sound source of interest coming from a certain direction or accompanied by many different sources (the famous cocktail party problem), and suppress unimportant sounds (e.g., air conditioner humming noise or the background music in a restaurant). The tasks performed by current hearing machines are typically handled by algorithms that are developed separately and independently from one another. Audio source separation, e.g., separating the singing voice in a song from background music, speech recognition in noisy environments or multi-speaker scenarios, and environmental sound detection and classification, e.g., recognizing dog barking or traffic noise are a few examples of such tasks.

A common feature of all these sound processing algorithms is that their performance and the difficulty of combining them with other algorithms are heavily affected by the audio representation they receive as input. If the input representation has fundamental limits, the algorithm may not be able to extract the information required for the task, no matter how intelligent it is. Moreover, if the information required for a particular auditory task is deeply buried in a representation, the algorithm performing this task will inevitably grow very complex and task-dependent in its feature extraction stage. Combining a set of single-task algorithms into a multi-task auditory scene analysis system would be both non-trivial and computationally inefficient if each algorithm applies its own task-specific and complex feature extraction stage to a low-level representation shared by all algorithms.

Audio source separation refers to the task of estimating n individual sound sources given an m-channel recording of a complex auditory scene. Supervised source separation methods, specifically those using deep neural networks have become very popular over the past decade, due to their successful performance in a variety of denoising and source separation tasks including speech enhancement, speech separation, and music separation. A major challenge faced by supervised masking-based separation approaches is that they typically require a large dataset of isolated sound sources to generate target time-frequency masks used in model training. Obtaining the isolated sources that compose an audio mixture may be expensive or require complicated recording setups. In some scenarios, it may not even be possible to record sounds in isolation, e.g., recording a bird song in a forest.

Parsing the auditory scene into meaningful components and focusing on the most informative sound sources are tasks which biological audio processing systems have evolved to perform efficiently. The mammalian auditory system, for instance, has been shown to extract the information required for the analysis of complex auditory scenes very effectively. An interesting example is the existence of neurons in the primary auditory cortex of mammals that respond to a variety of spectro-temporal modulation patterns. Moreover, natural audio-processing systems do not require isolated sources in order to learn to analyze auditory scenes. Humans hardly ever hear sounds in perfect isolation, but still can learn to identify different types of sounds and to focus on them if necessary. Based on the capability of natural auditory systems to extract the characteristics of individual audio sources from everyday complex auditory scenes, one can argue that the knowledge about the presence of sounds in a mixture recording could be sufficient information for training a separation system.

In this dissertation, I propose methods for audio signal representation and for training deep models that are inspired by biological auditory systems, addressing two major challenges in the field of audio source separation: i) separation of sources with a high level of energy overlap in both time and frequency domains, and ii) training deep models on the source separation task when ground truth isolated sources are not available. I develop a biologically-inspired audio representation that explicitly encodes spectro-temporal modulation patterns, and hence disentangles audio sources that overlap in time and frequency in a way that is practically useable for source separation and sound object recognition. I further propose a novel approach to training an audio separation system in the absence of strongly labeled auditory scenes. In this approach an audio classification system guides the separation training.

To Maman and Baba

To Hamid

And to the memory of my parents, Afsaneh Afzalnia, and Abbas Pishdadian – my guiding stars – who made the ultimate sacrifice with their young lives in the battle with religious fascism.

Acknowledgments

First and foremost, I would like to thank Bryan Pardo for being a fantastic advisor and an inspiring mentor. Thank you for giving me the chance to be a member of the Interactive Audio Lab family. Thank you for all the invaluable research, as well as professional development and career advice. Thank you for believing in me and always encouraging me to become an independent thinker. Thank you for pushing me out of my comfort zone when necessary and making me learn new skills to catch up with the fast-moving world of computer science. Thank you for constantly supporting women's presence in one of the most gender-imbalanced fields of study. Thank you for all the political/philosophical discussions at the end of our research meetings (sometimes ending in a very dark view of the world), which I am going to miss the most. I look forward to future collaborations and to many more intellectually stimulating discussions.

I would like to thank the rest of my committee, Thrasos Pappas and Oliver Coissart. Thank you for your support and for taking the time to provide me with valuable feedback.

I would like to thank my internship supervisors and collaborators at MERL: Gordon Wichern and Jonathan Le Roux. Thank you for such an amazing learning opportunity and professional experience. I would like to thank John Woodruff, my internship supervisor at Knowles and my future collaborator at Apple. Thank you for a great internship and all the valuable career advice. I would also like to thank Antoine Liutkus of Inria for all the great research discussions that led to an exciting collaboration.

I would like to thank my current and former labmates and colleagues at the Interactive Audio Lab: Prem Seetharaman, Bongjun Kim, Mark Cartwright, Ethan Manilow, and Max Morrison. Thank you for all the productive research discussions, for sharing your cool ideas and exciting experimental results with me, and for giving me great work-/life-related tips and advice throughout the past five years. I would also like to thank all the undergraduates that I had the privilege to meet and work with: Trent Cwiok, Kristen Amaddio, Julia Wilkins, Brian Margolis, Michael Donovan.

Thanks to my friends at Northwestern, Hara Iakovidou, Rawan Alharbi, Irina Rabkina, Kristen Sudman, and Ettore Trainiti for all the lovely moments and interesting discussions we had over the years.

I would like to thank the National Science Foundation (NSF) for supporting my work via grants 1420971 and 1617497. Of course, there are many more people to thank for their kindness and help throughout my academic life. I am sincerely grateful to every one of you.

My especial thanks go to my late grandparents, Fakhri Azimi, and Akbar Afzalnia, the strongest human beings I have had the honor of knowing in my life. Thank you for raising me with all the love and care you could muster, for always being there for me in spite of your own pain and grief. Thank you for giving me a reason to believe that humans are capable of dignity, decency, and kindness even when tragedy strikes. I would like to thank my aunt, Farzaneh Afzalnia. Thank you for your auntly/motherly/sisterly love and support for me throughout my life.

Finally, thanks to Hamid Charkhkar, the love of my life, my best friend, and my spouse. You have always believed in me and stood by me throughout the ups and downs of our lives over the past many years. I could never have overcome the hardships without your understanding, love, and support. Thank you!

List of Abbreviations

CFT: Common Fate Transform. 12, 13, 15, 26, 33, 34, 40–44, 50, 51, 53, 55–62, 64, 100, 113, 114

CQT: Constant-Q Transform. 10, 12–14, 16, 20, 21, 27, 30–33, 50, 51, 53, 55, 56, 58, 59, 61, 91, 96–106, 109, 112–114

ICFT: Inverse Common Fate Transform. 15, 41

ICQT: Inverse Constant-Q Transform. 51

IMCFT: Inverse Multi-resolution Common Fate Transform. 15, 51

ISTFT: Inverse Short-time Fourier Transform. 66, 73

L-MCFT: Light MCFT. 14, 17, 23, 25, 95, 96, 99–112, 114, 115

MCFT: Multi-resolution Common Fate Transform. 12–15, 23, 25, 26, 34, 41, 45, 50, 51, 53–56, 58–64, 91, 95, 96, 98, 100–103, 111–115

MFCCs: Mel Frequency Cepstral Coefficients. 20

SED: Sound Event Detection. 68–71, 78, 81, 82, 107, 116

STFT: Short-time Fourier Transform. 11–14, 17, 20, 27, 33, 40–45, 50, 53, 55, 56, 58, 59, 61, 66–68, 72, 74, 85, 86, 90, 91, 94, 95, 103–106, 108–113, 115

STRFs: Spectro-temporal Receptive Fields. 12, 32, 46–49

Glossary

audio classification: The task of dividing a set of sounds (available in a dataset) into distinct categories and assigning every audio excerpt to at least one category. 20, 23, 24

audio source separation: The process of estimating n source signals from m channel mixtures. Example: estimating the sound of individual instruments from a multi-channel recording of a piece of music. 19–21, 23, 24, 26, 27, 113

auditory object identification: Recognizing the type of a sound among a group of known (or even unknown) sound objects. 20, 23

auditory stream: A perceived sequence of sound components that is the result of segregation and grouping processes performed by the auditory system and is distinct form other co-occurring sequences. Example: a melody (sequence of notes) played by a violin in a piano-violin duet or a sentence (sequence of words) uttered by one speaker. 18, 32

auditory stream segregation: One of the key concepts of Albert Bregman's auditory scene analysis model referring to a perceptual decomposition of sounds into individual components. For example chords can be heard as one auditory object/stream or as a combination of individual notes. 20

machine hearing: The study and design of machines that are able to analyze the auditory scene and organize sound into parts that are perceptually meaningful to humans. 19, 113

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

> *If we had machines that could hear as humans do, we would expect them to be able to easily distinguish speech from music and background noises, to pull out the speech and music parts for special treatment, to know what direction sounds are coming from, to learn which noises are typical and which are noteworthy. Hearing machines should be able to organize what they hear; learn names for recognizable objects, actions, events, places, musical styles, instruments, and speakers; and retrieve sounds by reference to those names.*

Richard F. Lyon, *Machine Hearing: An Emerging Field*

Parsing the auditory scene into meaningful components and focusing on the most informative sound sources are tasks which biological audio processing systems (e.g., mammalian or avian auditory systems) have evolved to perform efficiently. To get an idea of how brilliantly our own auditory system works, imagine being in a busy coffee place, one of many complex and highly dynamic soundscapes we might encounter in daily life. In such a location, you are surrounded by people speaking, potentially in different languages. There is usually music playing in the background. Baristas shout ready orders, on top of the clicks and clacks of utensils and dishes. Occasionally the sound of someone laughing loudly, or a glass breaking, or a dog barking, or a baby crying might be added to the mix. In spite of all this information bombarding your auditory system, you are capable of having a conversation with your friend who is sitting across from you. A number of complicated processes are carried out by your auditory system so that you are able to engage in this conversation naturally and without any conscious effort, including: focusing on auditory streams coming from the direction of your gaze and suppressing those coming from the left, right, or behind you; identifying the pitch and timbre of your friend's voice among a group of speech streams that might all be coming from

the direction of your gaze; ignoring (tuning out) all sources of auditory information that are not important to you at the moment, including the background music, clicks and clacks of the dishes and the crying baby running around the next table. Not to mention that all the while you are capable of keeping an *ear* out for important sound events that might occur shortly, like your ready order of soy latte being announced. It has taken a whole research community, decades of work, and huge amounts of computational resources only to understand the mechanisms involved in one of these auditory tasks (e.g., speech enhancement, timbre recognition, source localization) and to engineer systems that can perform them half as effectively as humans.

The study and design of machines that are able to analyze the auditory scene and organize sound into parts that are perceptually meaningful to humans is referred to as *machine hearing* [5]. Such machines are expected to distinguish between different sound categories (e.g., speech, music, background noise), focus on a sound source of interest coming from a certain direction or accompanied by many different sources (the famous cocktail party problem [29] [71]), and suppress unimportant sounds (e.g., air conditionner humming noise or the background music in the coffee place example).

The tasks performed by current hearing machines are typically handled by algorithms that are developed separately and independently from one another. Audio source separation, e.g., separating the singing voice in a song from background music [87][85][55][21][32][93][66], speech recognition in noisy environments or multi-speaker scenarios [14][29][11][72], and environmental sound detection and classification, e.g., recognizing dog barking or traffic noise [109][53][10][40] are a few examples of such tasks.

A common feature of all these sound processing algorithms is that their performance and the difficulty of combining them with other algorithms are heavily affected by the audio representation they receive as input. If the input representation has fundamental limits, the algorithm may not be able to extract the information required for the task, no matter how intelligent it is. Moreover, if the information required for a particular auditory task is deeply buried in a representation, the algorithm performing this task will inevitably grow very complex/task-dependent in its feature extraction stage. Combining a set of single-task algorithms into a multi-task auditory scene analysis system would be both non-trivial and computationally inefficient if each algorithm applies its own task-specific and complex feature extraction stage to a low-level representation shared by all algorithms.

Most available audio processing algorithms are applied to minimally processed representations which will be referred to as *low-level* audio representations in this work [1]. Research has been mainly focused on the performance improvement of individual algorithms thus far and less attention has been paid to the audio representations used as input. For instance, most source separation algorithms, regardless of their structural complexity or cues they use, receive as input some variant of time-frequency representations [87][85][55][21][32], e.g., STFT, CQT [91], and Mel Frequency Cepstral Coefficients (MFCCs) [56]. Most approaches deal with the complexities of the source separation task at the algorithmic level and at the representation stage focus only on extracting basic cues from these low-level representations.

The superiority of natural audio processing mechanisms might not be merely due to the way extracted information from the input representation is analyzed and stored, but also to the use of superior audio representations. The mammalian auditory system, for instance, has been shown to extract the information required for the analysis of complex auditory scenes very effectively [62][75][9]. An interesting example is the existence of neurons in the primary auditory cortex of mammals that respond to a variety of spectro-temporal modulation patterns [13][9]. Furthermore, recent studies have found evidence on the plasticity of the feature extraction stages in the mammalian auditory system with respect to the attended auditory task [22][23][28]. That is, as a mammal attends to a sound, the audio representations in their auditory system adapt, on the fly, to make that sound more prominent and easy to distinguish. These findings can inspire the design of audio representations that make the information required by machine hearing algorithms more easily accessible and show flexibility with respect to the requirements of different audio processing tasks.

The first part of this dissertation deals with the development of an audio representation explicitly encoding the spectro-temporal modulation patterns as additional dimensions. In the second part, first a multi-task framework combining audio classification and audio source separation (two crucial machine hearing tasks that are closely related to perceptual auditory tasks: auditory object identification and auditory stream segregation [2]) is proposed for training source separation systems when ground truth sound sources are not available to be used as training targets, and then the new representation will be used as input to this joint separation-classification system.

---

[1] It should be noted that deep learning methods such as deep clustering [32], which map a low-level representation, e.g., STFT into an embedding space for mask inference, still perform source separation in the STFT domain.

[2] Throughout this document distinct terminology will be used for automated/machine hearing algorithms versus perceptual tasks that could inspire the design of those algorithms.

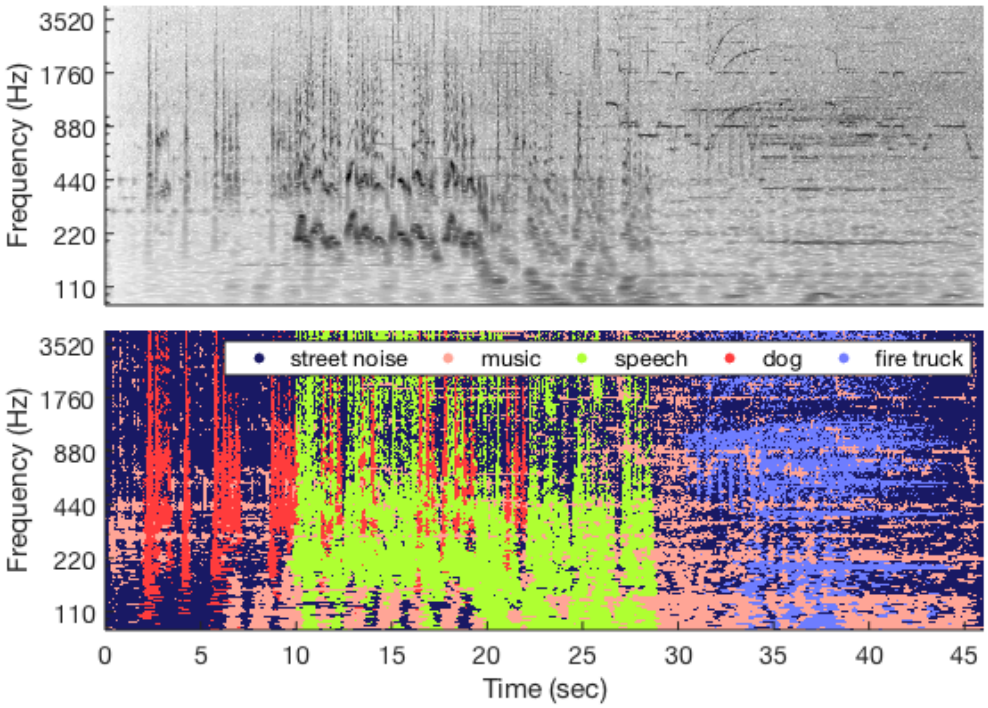

Figure 1.1. The time-frequency representation (magnitude CQT) of a complex auditory scene (top). There are five sources in this scene: street noise, music, speech, dog bark, and fire truck siren. Darker colors mean higher magnitudes. The bottom plot shows the ideal binary masks, which are generated by assigning the mixture energy to the most salient source at each time-frequency bin. Each color corresponds to one audio source.

Audio source separation refers to the task of estimating n individual sound sources given an m-channel recording of a complex auditory scene, also known as *mixture*. The top panel of Figure 1.1 shows the magnitude CQT of a recording of the auditory scene in a busy street including five distinct sources: street noise, music, speech, dog bark, and fire truck siren. Isolating the speech or dog bark in this mixture of sounds is an example of the separation task. Separation via *mask inference* [113][76][50] is a common approach to solving the under-determined source separation problem, in which the number of sources exceeds the number of recorded channels, i.e., $n > m$, and thus the spatial/inter-channel information is not sufficient to perform source separation via beamforming [74] or Independent Component Analysis (ICA) [64]. The bottom panel of Figure 1.1 illustrates the *ideal binary masks* corresponding to the sound sources available in the audio mixture presented in the top panel The masks are generated by assigning the mixture energy at each time-frequency point to the source with the highest energy at that point.

Supervised mask inference methods, specifically those using deep neural networks have become very popular over the past decade, due to their successful performance in a variety of denoising and source separation tasks including speech enhancement [17][57][115][120][122], speech separation [32][35][44][50][58][96][117][126], and music separation [60][106][92][49]. A major challenge faced by supervised masking-based separation approaches is that they typically require a large dataset of isolated sound sources to generate target time-frequency masks used in model training. Obtaining the isolated sources that compose an audio mixture may be expensive or require complicated recording setups. In some scenarios, it may not even be possible to record sounds in isolation, e.g., recording a bird song in a forest or recording the sound of a machine part that only occurs when a machine is running. In cases where isolated sources are not available for mask estimation training, it is also unrealistic for humans to manually label the audio mixtures at the granularity level of time-frequency bins (imagine creating the binary masks in the bottom plot of Figure 1.1 manually!).

Natural audio-processing systems (e.g, the mammalian auditory system) on the other hand, do not require isolated sources in order to learn to analyze auditory scenes. Humans hardly ever hear sounds in perfect isolation, but still can learn to identify different types of sounds and to focus on them if necessary (segregate a single audio source from the rest of the auditory scene). Based on the capability of natural auditory systems to extract the characteristics of individual audio sources from everyday complex auditory scenes, one can argue that the knowledge about the presence of sounds in a mixture recording could be sufficient information for training a separation system. A separation system that can learn from audio mixtures relying only on the information about the present sound types would make the dataset generation much easier, since recording a mixture of sounds in complex auditory scenes is much simpler and less expensive than recording single sound sources in perfect isolation. It is also reasonable to assume that even non-experts can produce limited labels for the activity of sound sources within some time range, and thus, any mixture recording along with such annotations can be used for training.

I will take an auditory-inspired approach to supervised audio source separation training. The differences between my approach and current methods are: i) the separation system receives as input the auditory-inspired representation developed in the first part, which is shown to provide higher separability than commonly used representations, ii) rather than ideal masks, the information about presence or absence of certain types of sounds in a short segment of the input mixture is used as target for training. It is important to note that since the separation system still outputs estimated sources in the representation domain, this approach

requires a system that maps the source energies in the mixture or isolated source representations to sound activities over time.

Auditory object identification, a central task in auditory scene analysis, refers to recognizing the type of a sound among a group of known (or even unknown) sound objects. It is closely related to audio classification in the sense that the identified source can be regarded as a member of a *class* of sounds, for instance, recognizing a dog bark in the recording of a busy street, where the set of present sound types/classes may include speech, street music, cars passing, construction noise, etc. In this work, an audio classification/identification system will be employed to perform the mapping between audio mixtures or estimated sources (by the separation system) to sound presence information. It will be demonstrated that if this classification system is pre-trained to identify sound types in audio mixtures, it can be used for training the separation system in the absence of isolated sources.

1.1. Contributions

In this dissertation, I develop approaches to audio representation as well as joint identification and separation of auditory objects. The main contributions of this work include:

- A bio-inspired audio representation, explicitly encoding spectro-temporal modulation patterns. This representation, termed MCFT, has been shown to improve audio source separation [77][81] and audio classification [78], two of the most important tasks in machine hearing (Chapter 2).
- A novel approach to training an audio source separation system in the absence of strongly labeled auditory scenes [79]. In this approach, an audio classification system guides the separation training (Chapter 3).
- A subsampled version of the MCFT, termed L-MCFT with a significantly smaller size that can be easily used in the context of deep learning, particularly as the input to the joint separation classification framework developed as part of this dissertation (Chapter 4).

1.2. Broader Impact

Biological systems perform many auditory tasks better than any automated systems developed to date. Most automated audio processing systems are not applied to audio representations similar to those used in biological systems. An audio representation that duplicates the functionality of biological representations

in their elaborate feature extraction would make it easier to combine different audio processing tasks (e.g., audio classification and audio source separation) and thus is of interest to many fields of research.

Speech processing and music information retrieval can benefit from such a representation in the development of more flexible and efficient technology. The proposed representation can be used in a broad range of applications, including audio source separation [87][85][55][21][32], speech recognition [94], speaker identification [86], classification of human vocalization [78], musical timbre analysis [6], and automatic labeling of environmental sounds [7][40][41]. Privacy preserving audio encoding, e.g., *speech blurring*, is another potential application, where explicit encoding of speech related characteristics makes it possible to blur out the speech when encountered, without harming the information in the rest of the scene.

Typically, findings in the fields of neuro-/psychoacoustics inspire the design of better audio signal processing tools, e.g. Mel-scale spectrograms [100][34], which are widely used in speech processing [47][110][26]. In return, the signal processing knowledge and advances provide necessary tools for the development of more accurate auditory models used in neuro-/psychoacoustics studies, e.g., the use of filterbank design techniques in auditory system modeling [63]. The proposed work may thus suggest new directions for creation of methods that close the loop between audio signal processing and neuro-/psychoacoustics.

Auditory-inspired audio processing methods, e.g., audio source separation and speech enhancement algorithms using auditory-inspired feature extraction stages could be helpful in the advancement of hearing aid technology. Such algorithms facilitate the segregation of auditory scene by selective amplification of important sound sources [83]. The performance of digital assistants such as Amazon Echo and Google Home could be improved by employing algorithms that combine multiple auditory tasks, e.g., audio classification and audio source separation with a less complex structure. The design of human-like sound processing systems based on the proposed representation and combination of auditory tasks can be of great assistance to Artificial Intelligence (AI) and robotics research. Navigating a space through sound stimuli could become easier for robots with human-like sound processing mechanisms.

The development of model training techniques that work in the absence of strongly labeled auditory scenes would render the data acquisition stage in speech and music processing faster and less expensive. Moreover, such techniques would have a significant impact on fields where recording perfectly isolated sources is almost impossible, e.g., in urban or natural (birds/wildlife) soundscapes. Less burdensome data collection

would result in the development of models that are trained on larger and more diverse datasets, can be updated with new data more frequently, and thus have a higher generalization power.

1.3. Structure of the Dissertation

The remainder of the dissertation and the relevant publications are organized as follows:

- Chapter 2 presents the MCFT, a new auditory-inspired audio representation [77][81].
- Chapter 3 presents a novel approach to supervised audio separation training [80] [79].
- Chapter 4 presents a subsampled version of the MCFT, named L-MCFT and the joint separation-classification system that receives the L-MCFT as input.
- Chapter 5 summarizes the observations, draws conclusions, and lays out future directions.

CHAPTER 2

Multi-resolution Common Fate Transform

In this chapter, I present my work on the development of the MCFT [77][81], an audio representation whose design was inspired by the *common fate* principle (see Section 2.1). The MCFT was initially developed in the context of audio source separation, however, it has been proven to be also useful for classification of sounds with distinct spectro-tempral modulation characteristics [78]. In addition to temporal and spectral information, the MCFT encodes spectro-temporal modulation patterns as explicit dimensions, and thus increases the *separability* of mixtures of multiple audio signals that overlap in both time and frequency domains (see Section 2.3). The MCFT combines the *invertibility* (i.e., being able to reconstruct a time-domain audio signal given its representation in another domain) of a state-of-the-art representation, the CFT, and the multi-resolution property of the cortical stage output of an auditory model. Since the MCFT is computed based on a fully invertible complex time-frequency representation, separation of audio sources with high time-frequency overlap may be performed directly in the MCFT domain, where there is less overlap between sources than in the time-frequency domain. The MCFT circumvents the resolution issue of the CFT by using a multi-resolution 2D filter bank instead of fixed-size 2D windows. The remainder of this chapter:

- Describes the MCFT and discusses its properties with the aid of illustrative examples.
- Provides definitions and objective measures for two desirable representation properties: *separability* of source signals and *clusterability* of components of each signal.
- Presents and compares the results of source separation via ideal binary masking in different representation domains, on a comprehensive dataset of audio mixtures of musical tones played in unison, including audio samples from a wide pitch range and a variety of instruments/playing techniques.

2.1. Background

Audio source separation refers to the task of estimating n individual sound sources given an m-channel recording of a complex auditory scene, also known as *mixture*. It is an important enabling technology to a variety of applications, including: automatic speaker identification in a multi-speaker scenario [11, 29], speech recognition in noisy environments [14], musical instrument recognition in polyphonic audio [30], music remixing [124], music transcription [82], upmixing of stereo recordings to surround sound [20, 37], and lyric-music synchronization [24].

Underdetermined source separation is an important scenario, where the number of sources exceeds the number of recording channels, i.e., $n > m$. Separation via *mask inference* is a common approach to solving the under-determined source separation problem [61][119][114], where the spatial/inter-channel information is not sufficient to perform source separation via beamforming [74] or Independent Component Analysis (ICA) [64]. In this work, I focus on one of the most common underdetermined scenarios: performing separation on monophonic or stereo recordings of mixtures of two or more sound sources.

In a typical mask inference approach, first the raw audio signal is transformed into a representation with a higher level of *separability*, i.e., less energy overlap between sources, and ideally better *clusterability*, i.e., a mapping of mixture components such that components belonging to a single source are close to one another and far from the components of other sources. In this representation domain, each source is isolated by setting the components of other sources to zero, in other words, masking the energy coming from other sources. In the final step, the time-domain version of each isolated source is estimated by applying an inverse transform to the corresponding masked representations.

The time-frequency representation domain (e.g., STFT or CQT) is most commonly used for masking-based source separation, where the mixture energy at each time-frequency bin is either assigned to a single source (via binary masking) or distributed between different sources (via soft masking). Dealing with high levels of energy overlap between sources is a major challenge faced by source separation algorithms that use time-frequency representations as their input. In general, regardless of the algorithm, if the input mixture is represented in the time-frequency domain, performance degrades as the time-frequency energy overlap between sources increases.

A number of source separation approaches map a time-frequency representation to another representation domain, so that the source separation problem can be solved through distance-based clustering. Clusters,

which are assumed to correlate with sources, are then used to create time-frequency masks. Examples include approaches that perform the mapping with a mathematical formula, such as DUET [87] and Kernel Additive Modeling [55], as well as methods that learn a higher-dimensional embedding from data, such as Deep Clustering [32]. In all these cases, the final masking is performed in the time-frequency domain, which leaves the issue of time-frequency energy overlap unresolved.

The type of processing performed in the human auditory system can inspire the development of richer (higher dimensional) representations that capture more information about the properties of audio signals as additional dimensions, and thus inherently increase the chance of better separation. Such rich representations would reduce the burden of feature extraction on the algorithmic side of source separation and thus allow the design of low-complexity algorithms that can be combined more easily. Consider, for instance, the *common fate* principle, a concepts borrowed from Gestalt psychology and applied to the domain of psychoacoustics by Albert Bregman [5]. The Gestalt principle of common fate states that humans tend to perceive visual components that move in the same direction and/or with the same velocity as being more related than stationary components or those moving in different directions. Bregman (in [5], page 249) describes an example of the common fate principle as follows:

> “*Let us imagine that we had a photograph taken of the sky. It shows a large number of birds in flight. Because they are all facing the same direction and seem to be at the same distance from the camera, we think that they are a single flock. Later we are shown a motion picture taken of that same scene. On looking at this view, we see that there were two distinct flocks rather than only one. This conclusion becomes evident when we look at the paths of motion. One group of birds seem to be moving in a set of parallel paths describing a curve in the sky. Another group is also moving in a set of parallel paths but this path is different from that of the first group. The common motion within each subgroup binds the group together perceptually and, at the same time, segregates it from other group. The common motion within each group is an example of the Gestalt principle of common fate.*”

Figure 2.1 illustrates a simple example of the common fate principle. In the left picture, all of the thirteen arrows point to the same direction. As a result, they are perceptually bound together as one group of visual

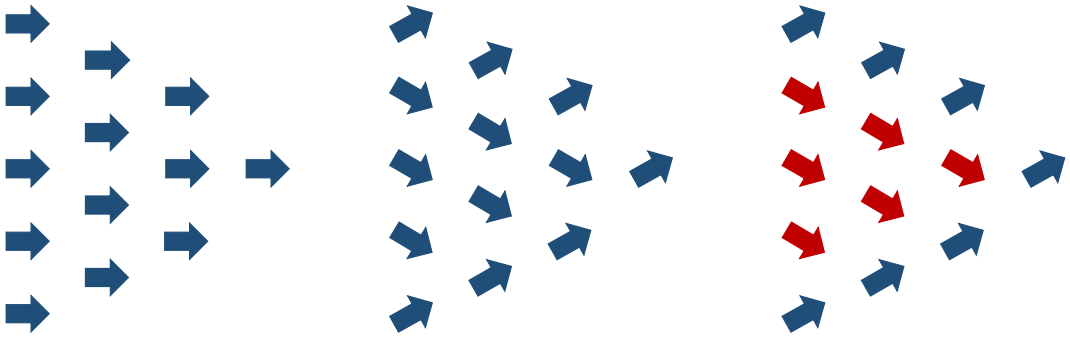

Figure 2.1. An illustrative example of the Gestalt principle of common fate. In the left picture, all visual elements (arrows) point to the same direction, and thus they are perceived as parts of one group. In the middle picture, the arrows are perceptually divided into two groups based on the direction to which they are pointing. In the right picture, the two groups are distinguished with different colors.

elements. In the middle picture, on the other hand, the six inner arrows point in a downward direction while the seven outer arrows point upward. These arrows, therefore, are "perceived" as belonging to two different groups. If these pictures are presented to a computer vision algorithm performing the same arrow-grouping task, the algorithm should clearly be able to make inferences based on the concept of direction in a two-dimensional space. Now imagine using a richer representation instead, where directionality is also explicitly encoded as an additional dimension, e.g., color, as is shown in the right picture. The same results can be achieved in this case by a simpler algorithm that groups visual objects merely based on their color.

Bregman (in [5], page 250) explains how the Gestalt principle of common fate can be translated from the visual domain to the domain of auditory scene analysis:

> *The principle of common fate also has an application in audition. Suppose it was found that two frequency components were changing synchronously by proportional amounts. This would seem to be very unlikely by chance. It is much more reasonable to assume that the two are parts of the same sound, that is, that they have arisen from the same physical disturbance in the environment. It is likely in our world that those frequency components that arise from a single acoustic source will go on and off at more or less the same time, will glide up and down in frequency together, will swell and decline in intensity together, and so on.*

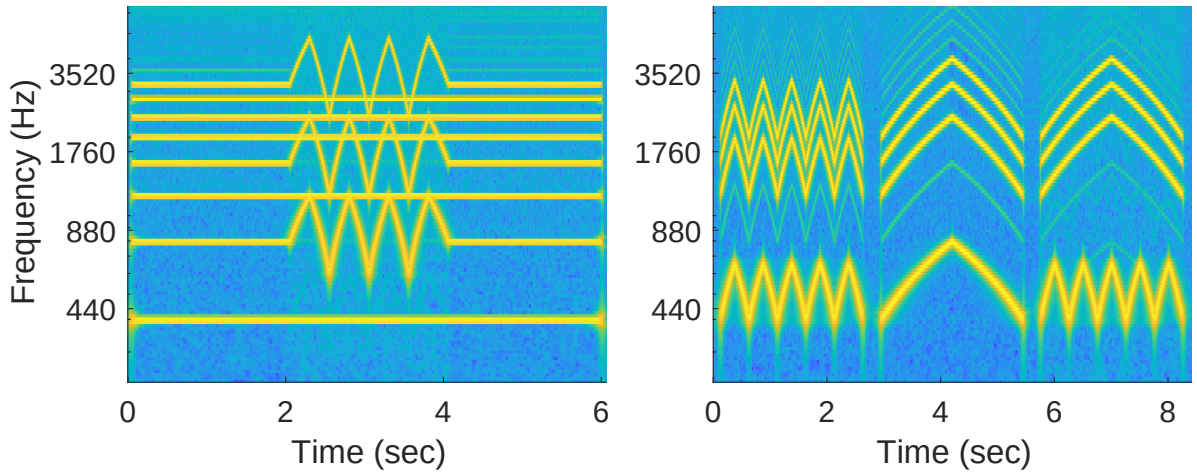

Figure 2.2. The magnitude CQT of two audio examples from *Auditory scene analysis: The perceptual organization of sound* [5] (No. 19 and 20): demonstrations of fusion based on common frequency modulation.

Based on this description, spectral components with the same spectro-temporal modulation properties (i.e., components moving up and down together in the time-frequency domain with a logarithmic frequency scale) are more likely to be grouped together and perceived as a single audio stream by human listeners. Figure 2.2 depicts the magnitude CQT (a widely used time-frequency representation) of two audio examples from *Auditory scene analysis: The perceptual organization of sound* [5], demonstrating perceptual fusion of spectral components based on common frequency modulation. The left panel shows a synthesized harmonic sound composed of eight partials. In the beginning, none of the partials is modulated and thus they are all fused and perceived by humans as one stream. In the middle section, a subset of partials are modulated, resulting in the perception of two different sound streams, one with a steady pitch and one with a pitch that moves up and down. When the modulation is turned off in the last part, the audio signal sounds like a single source again. In the right panel, audio signals with the same set of partials but two different modulation patterns are shown. When all partials are modulated with the same pattern, be it the periodic sinusoidal-like pattern or up-/down-ward glides, they fuse and are perceived as a single stream. However, dividing the partials into two subsets and modulating them differently results in the perception of two streams.

The human voice is an everyday example of a harmonic sound featuring a wide variety of modulation patterns. Figure 2.3 shows two examples of human vocalization. A short excerpt of spoken words is presented in the left panel. It can be clearly observed that the movements of overtones in the time-frequency domain follow that of the fundamental frequency at each instant of time. For instance, the first word, “row" (0 sec

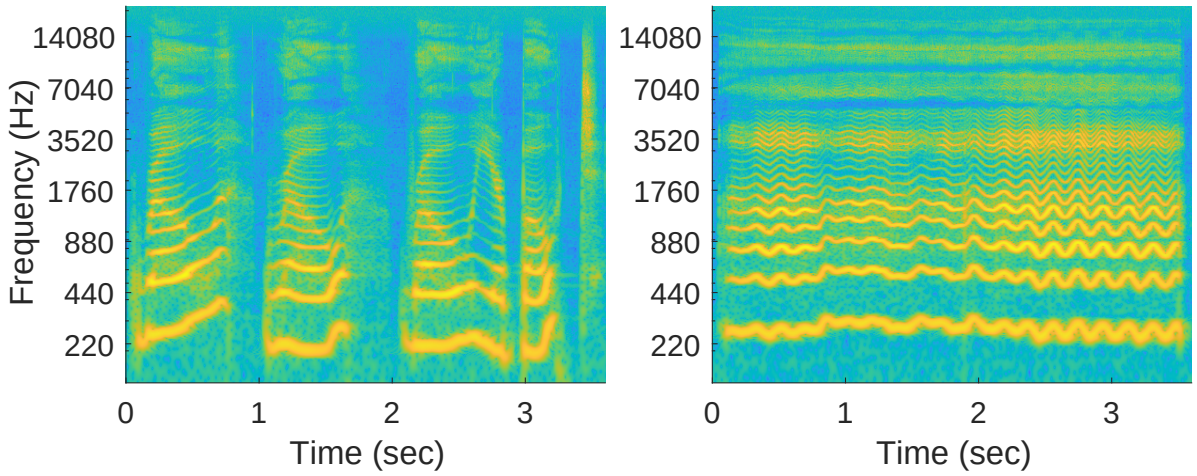

Figure 2.3. The magnitude CQT of recordings of human voice. The left panel shows a short excerpt of a spoken phrase (*row, row, row your boat*). The right panel shows an example of a singing female voice (singing *ah* using the long trill technique).

to 1 sec), is uttered with an upward gliding pitch, that is, a quick upward movement of all partials over a wide range of frequencies, while the harmonic relationship between them is preserved at each instant of time. The right panel shows an excerpt of a female singing voice. The singer is trying to maintain a steady pitch while using the long trill technique. The frequency modulation in this case has a sinusoidal pattern and stays within a much narrower range of frequencies centered around each partial. Again, an interesting observation is that the partials move up and down synchronously and thus their harmonic relationship is preserved throughout. These two examples demonstrate why the human auditory system uses the common fate principle as a grouping strategy in perceptually fusing the partials in any kind of human vocalization and making their combination sound like a single stream.

Harmonic musical sounds are yet other examples of physical systems with overtones that co-vary in time and frequency, making a common-fate assumption useful. Figure 2.4 shows two musical sound examples. An excerpt of a violin sound playing the note C4 (with a fundamental frequency of 261.63 Hz) with the vibrato technique is depicted in the left panel and an excerpt of a trombone sound playing the same note with the tremolo technique in the right panel. The vibrato modulation is very subtle as opposed to the tremolo modulation which produces quite pronounced and intricate spectral patterns. What these two example sounds have in common, however, is the perceptual fusion of their many partials into a single sound stream by human listeners.

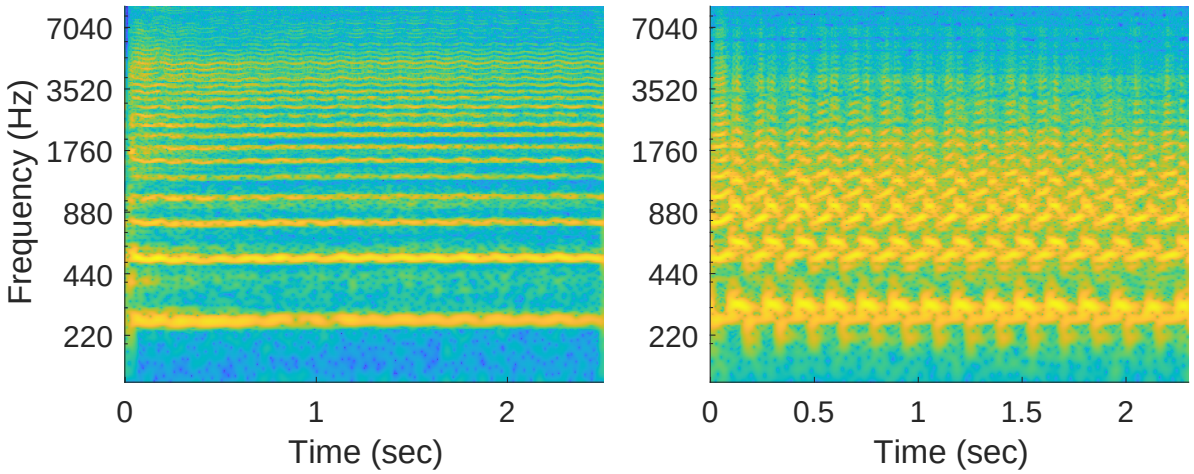

Figure 2.4. The magnitude CQT of recordings of the musical note C4 (261.63 Hz), played by two different instruments and with two different techniques: violin-vibrato (left) and trombone-tremolo (right).

In addition to the results of psychoacoustics experiments on humans, indicating the important role common modulation patterns play in auditory stream formation, there is now physiological evidence for this phenomenon from experiments on the auditory system of small mammals (e.g., ferrets). Recent studies on the primary auditory cortex of small mammals have also shown the importance of spectro-temporal modulation patterns in audio perception and streaming [13] [15] [9]. These studies suggest that the neuronal response of the primary auditory cortex to different spectro-temporal modulation patterns, termed STRFs can be regarded as a bank of two-dimensional filters, each focusing on a particular modulation pattern and segregating all auditory elements with that type of modulation (e.g., all partials of a musical note) from the rest of the auditory scene.

As mentioned earlier, when there are multiple sources in an auditory scene, the spectral components that may belong to one stream according to the common fate principle are not easily distinguishable in time-frequency representations, due to overlap. Accounting for spectro-temporal modulation properties as explicit dimensions in a representation would facilitate the extraction of this information without requiring the source separation algorithms to grow too complex. A representation that explicitly encodes spectro-temporal modulations will be useful in processing sounds with a rich modulation profile, such as human vocalization (e.g., speech and singing) and music (see Figures 2.3 and 2.4). In the next section, I provide a brief review of prior work on commn-fate-based audio source separation and audio representation design.

2.2. Related work

The Short-time Fourier Transform (STFT) and Constant-Q Transform (CQT) [91] are two examples of general time-frequency representations commonly used for audio processing. A major shortcoming of these low-level representations is that they do not explicitly encode the spectro-temporal modulation information, a very important cue in audio perception. As a result, the burden of extracting modulation patterns falls on algorithms that rely on such information and use time-frequency representations as input.

The common fate principle has been employed by some methods such as Non-negative Tensor Factorization (NTF) [21] [12] at the algorithmic level, while leaving the underlying audio representation (magnitude spectrogram) unchanged. The method proposed by Abe et al. [1] is an early work that exploits modulation properties for source separation.

In a recent attempt to address the difficulty in the separation of same pitch (unison), frequency-modulated sources, Stöter et al. [103] proposed a 4D representation, named the Common Fate Transform (CFT), which explicitly captures common fate. The CFT is computed by dividing the complex STFT of an audio signal into overlapping 2D windows and then analyzing each windowed segment by the 2D Fourier transform. The CFT is fully invertible and presents time, frequency, and modulation information as explicit dimensions. Stöter et al. demonstrated that, compared to standard time-frequency representations, the CFT can provide higher separability for harmonic sounds with close fundamental frequencies, but different modulation patterns. However, a shortcoming of the CFT is that it uses a fixed-size 2D window in the time-frequency domain for capturing local modulation patterns, which could range from very slow to very fast. The choice of the fixed window size limits the transform-domain resolution, and hence affects the separation results for sources with close modulation patterns. To achieve maximal performance for a particular situation, a knowledgeable user must select the appropriate window size. It would, however, be preferable to attain good separation results without having the need for hand-tuning the window size. The resolution issue of the CFT can be addressed by using a multi-resolution approach, i.e., analyzing the time-frequency representation over a range of window sizes, or equivalently through a filter bank.

The auditory model proposed by Chi et al. [9] emulates the important aspects of the cochlear and cortical processing stages in the auditory system of small mammals. It transforms the audio signal into a 4D representation based on a multi-resolution analysis approach. The output representation captures spectro-temporal modulation patterns as two additional dimensions, named *scale* and *rate*. It should be noted that

the main goal in the design of these auditory models is to replicate the outputs of natural audio analysis stages as closely as possible. They, therefore, create lossy representations that do not contain all the detail needed to achieve perfect (mathematically zero-loss) reconstruction of the original waveform (also known as *invertibility*). Iterative reconstruction algorithms approximating the waveform within some error margin have been proposed [9][97][98]. However, the quality of the output audio still suffers from some distortion, which hinders their use in machine hearing tasks that require perfect reconstruction of time-domain signals, e.g., audio source separation.

Krishnan et al. [46] proposed a source separation algorithm that uses the output of Chi's auditory model to build time-frequency-domain masks, but since it is forced to apply masking in the time-frequency domain (because Chi's model is not invertible), it remains susceptible to time-frequency overlap between sources. Mesgarani et al. [69] proposed a speech enhancement method based on filtering the noisy signal in the full 4D representation domain. The method is able to suppress noise with distinctive modulation patterns even in cases where there is time-frequency overlap between the speech and noise. However, to recover the acoustic signal they use the signal estimation algorithm accompanying the auditory model, which despite preserving the intelligibility of speech signals suffers from poor reconstruction quality.

The remainder of this chapter is organized as follows: I introduce separability and clusterability, two important properties of audio representations affecting the performance of source separation algorithms in Sections 2.3 and 2.4. The precursors to this work, the Common Fate Transform (CFT) and Chi's auditory model, are then studied in detail in Sections 2.5 and 2.6, respectively. In Section 2.7, I present the Multiresolution Common Fate Transform (MCFT) and discuss its important properties. Experimental results showing the separability and clusterability of a variety of representations are presented in Section 2.8. Section 2.9 concludes the chapter and briefly discusses the significance of this work.

2.3. Audio representation and separability

In this section, I introduce the concept of *separability* as a measurable property of audio mixtures. The separability of two signals depends on the properties of the signals and also the properties of the representation domain. What seems inseparable in one representation domain may be easy to separate in another. Figure 2.5 shows four simple mixtures represented in different domains. The first example, displayed in Panels (a) and (b), is a mixture of two single-frequency sinusoids in the time and frequency

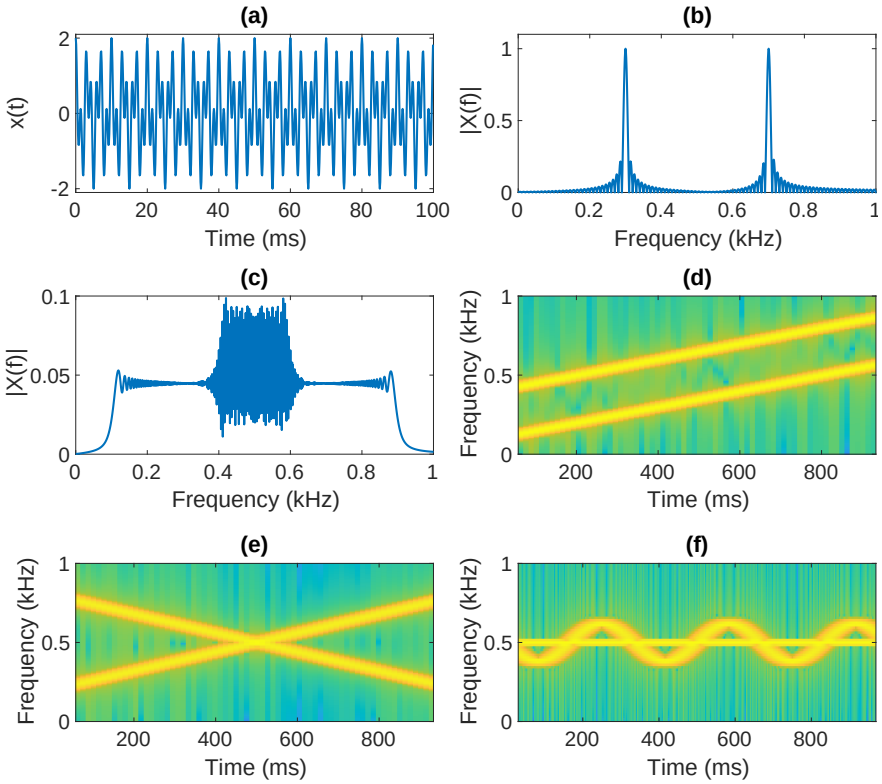

Figure 2.5. Examples of simple audio mixtures with different levels of separability based on the representation used. Panels (a) and (b) display a mixture of two single frequency sinusoids, represented in the time domain and in the frequency domain respectively. In panels (c) and (d), a mixture of two linear chirps is demonstrated, in the frequency domain and in the time-frequency domain respectively. Panels (e) and (f) show mixtures with overlapping energy in the time-frequency domain (a mixture of two linear crossing chirps and a mixture of two sinusoids with different frequency modulation patterns).

domains respectively. Clearly, the separation task is very difficult in the former case, while quite easy in the latter. Panels (c) and (d) present the second example, a mixture of two linear chirps, in the frequency and time-frequency domains respectively. Going from the frequency domain to the time-frequency domain decreases the energy overlap between the two sources, and thus helps the separation. The two examples in Panels (e) and (f) demonstrate that even the time-frequency domain is not immune to overlap between sources as mixtures become more complex. As a matter of fact, the example of Panel (f), where the mixture consists of sources with significant time-frequency overlap, presents one of the most challenging scenarios for the source separation task.

I now provide basic definitions for an underdetermined, linear mixture source separation problem. Let $x(t)$ denote a mixture of N time-domain audio signals, that is

$$x(t) = \sum_{j=1}^{N} s_j(t), \tag{2.1}$$

where t is the time index, $s_j(t)$ is the amplitude of the j^{th} source in the mixture at time t, and $u_j(t)$ indicate the sum of all sources interfering with $s_j(t)$, i.e.,

$$u_j(t) = \sum_{i=1, i\neq j}^{N} s_i(t). \tag{2.2}$$

Assume a linear transform (such as the Fourier transform), denoted by $\mathcal{T}$, is applied to the audio mixture and its constituent sources, taking them from the time domain to a k-dimensional representation domain $\mathcal{D}$. Let $\mathbf{d} = (d_1, d_2, ..., d_k) \in \mathcal{D}$ be an arbitrary point in $\mathcal{D}$, and let $S_j(\mathbf{d}) = \mathcal{T}\{s_j(t)\}$ and $U_j(\mathbf{d}) = \mathcal{T}\{u_j(t)\}$ indicate the transformed versions of $s_j(t)$ and $u_j(t)$, respectively. In general, we assume transformed signals to be complex valued.

An ideal binary mask separating the j^{th} source from the rest of the mixture in the transform domain can be defined as [88]

$$M_{j,\gamma}(\mathbf{d}) = \begin{cases} 1 & if \;\; 20\log_{10}\left(\frac{|S_j(\mathbf{d})|}{|U_j(\mathbf{d})|}\right) > \gamma \\ 0 & otherwise, \end{cases} \tag{2.3}$$

where γ, measured in deciBels (dB) is the masking threshold. Note that for the above formula to be valid, both $|S_j|$ and $|U_j|$ values are assumed to be nonzero. In practice, the expression $20\log_{10}\left[(|S_j(\mathbf{d})| + \epsilon)/(|U_j(\mathbf{d})| + \epsilon)\right]$ with $\epsilon << 1$ is used to avoid numerical errors.

Equation (2.3) simply states that the total mixture energy at each point in the representation is assigned to the j^{th} source if it dominates the total interference from other sources by γ dB. In other words, the j^{th} source "loses" its energy at a given point if the energy ratio between the source and the interference does not pass the masking threshold. It is possible to have points where none of the sources is dominant. The values of all masks at such points are set to zero, and thus the mixture energy is not assigned to any of the sources. Figure 2.6 presents an example of binary masks calculated from Equation 2.3. The top row shows two harmonic signals with the same fundamental frequency, with and without frequency modulation, along with their mixture. The binary masks corresponding to the two sources and their sum are shown in the

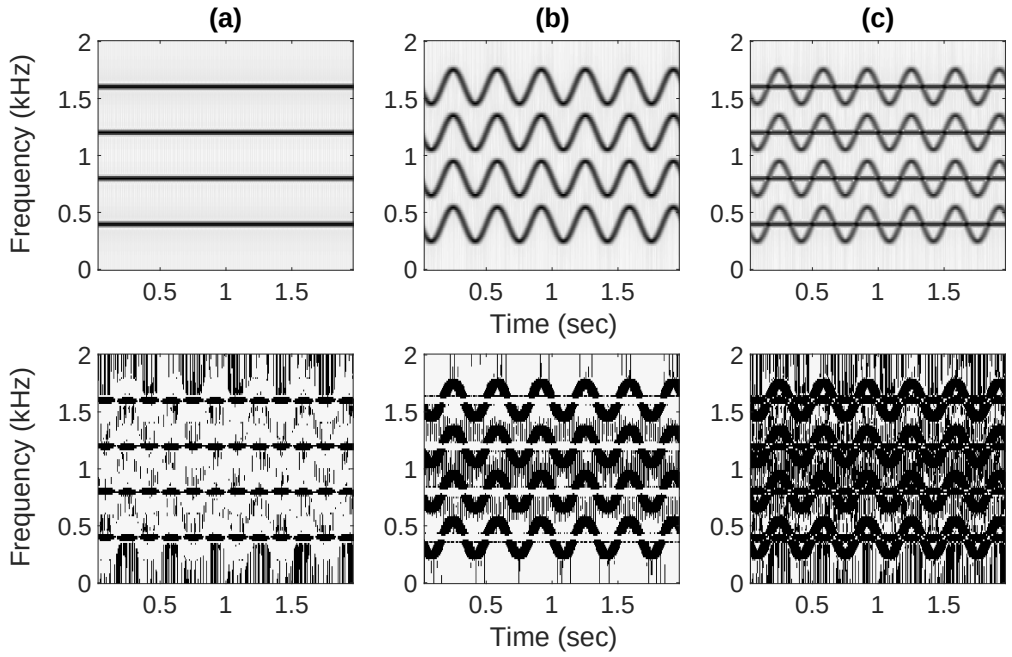

Figure 2.6. Two harmonic signals, with and without frequency modulation, their mixture, and ideal binary masks calculated from Equation (2.3), using $\gamma = 5$ dB. The two signals have the same fundamental frequency (400 Hz). In the top row, columns (a) and (b) show the two separate signals and column (c) shows their mixture. In the bottom row, columns (a) and (b) show the ideal binary masks corresponding to the sources in the top row and column (c) shows the sum of ideal binary masks. Signals and masks are represented in the time-frequency domain. In all panels, darker colors mean higher values. In the bottom row, black and white correspond to 1 and 0, respectively.

bottom row. It can be observed that if the masks are applied to the mixture, each source would lose some of its energy at points where the two sources overlap. Moreover, the sum of masks (column (c), bottom row) contains a number points with a value of zero. At these points, the mixture energy is not assigned to either of the sources.

A measure of separability in a representation domain can be defined as the energy portion of the j^{th} source preserved through masking, normalized by the total energy of the original source, where both the original and masked signals are placed within that representation domain. A version of such a measure, named approximate W-disjoint orthogonality (WDO), introduced by Rickard et al. [88], is calculated by placing the mixture into a time-frequency representation. It should be noted that the use of this energy ratio measure is only appropriate when comparing different mixtures represented in the same domain. Due to the dimensionality mismatch between the representation domains discussed throughout this work and different

types of analysis methods and parameters involved in their computation (e.g., fixed-size windowing versus multi-resolution filtering), the outputs of such a measure are not comparable across representations.

To measure how well different representations naturally separate sources in a mixture, I take an alternative approach, which makes comparison of different domains possible. Instead of measuring the preserved energy ratio in the transform domain, the separability is inferred based on the quality of the time-domain reconstructed sources that were separated via ideal binary masking in different representation domains. Since the main assumption in computing ideal binary masks for a representation domain is the dominance of at most one source at each point, the quality of the separated sources using such masks would be highly correlated with the level of separability provided by the representation. For time-domain evaluation of the separation performance, the BSS-Eval [111] objective measures Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR), and Source to Artifact Ratio (SAR) are used.

2.4. Audio representation and clusteribility

In this section, I present a measure of *clusterability*, an important property of representations that, to my knowledge, has been little studied in the context of source separation. Clusterability can be defined as the tendency of a representation domain to map the energy of audio sources such that the distance between points belonging to one source (*intra-cluster* distance) is considerably smaller than the distance between points from two different sources (*inter-cluster* distance). This is known as *distance-based* clusteribility [16]. A representation that enhances the distance-based clusterability would make the source separation task more straightforward, as a simple distance-based clustering algorithm (e.g., Gaussian Mixture Models [128]) could be used to assign energy to sources. This insight has been exploited in multiple source separation approaches (e.g., Kernel Additive Modeling [55], DUET [87], Deep Clustering [32]).

In their approach to image segmentation as a graph partitioning problem, Shi et al. [95] proposed the *normalized cut*, a criterion that simultaneously measures the total similarity between nodes belonging to the same group and the total dissimilarity between nodes in different groups. Bach et al. [4] derived a loss function based on the normalized cut for *spectral clustering*, a graph partitioning technique, which relies on the eigenstructure of the similarity matrix in order to assign nodes with high similarity to the same cluster and those with low similarity to different clusters. In this work, I use the normalized-cut-based loss function of Bach et al. [4] as a measure of the clusterability offered by a representation. The ideal binary masks in

a representation are considered the outputs of an ideal clustering algorithm for that representation. The mask points with a value of one are treated as the nodes of an undirected weighted graph. The pairwise distance in the representation space defines edge weights. This allows the computation of the value of the normalized cut for the partitioning of the high energy points produced by ideal binary masks corresponding to sources in an audio mixture. Low normalized cut values for a given representation imply high levels of distance-based clusterability.

In practice, treating every point as equally important can be problematic. Since the only criterion for passing a masking threshold is the dominance of the target source energy and not the absolute energy level, there can be a large number of low-energy points in each estimated source representation that can be counted in the source cluster without contributing much to the total signal energy. Therefore, magnitude thresholding is applied to masks in order to remove low-energy points. A second motivation for the use of this thresholding stage is to lower the computational burden in the calculation of similarity matrices by removing points that contribute little. In the experiments, the threshold is set to 20 dB below the maximum magnitude value for each estimated source.

Let W denote the similarity matrix for a given set of high-energy points in a k-dimensional representation domain, $\mathcal{D}$. Following the framework in Bach et al. [4], the similarity between two arbitrary points $\mathbf{d}_i$ and $\mathbf{d}_j$ can be assumed to be a diagonally scaled Gaussian function of the distance between the two points, i.e.,

$$W_{ij} = \exp(-(\mathbf{d}_i - \mathbf{d}_j)^\top diag(\alpha)(\mathbf{d}_i - \mathbf{d}_j)), \tag{2.4}$$

where W_{ij} indicates the value on the i^{th} row and j^{th} column of the similarity matrix, $\alpha \in \mathbb{R}^k$ is a vector of positive weights, and $diag(\alpha)$ is a $k \times k$ diagonal matrix with diagonal α.

Next, I present a formulation of the normalized-cut-based loss function from Bach et al. [4], which is employed in this dissertation as a measure of the clusterability offered by a representation. Let $v_n \in \mathbb{R}^m$ be the indicator vector for the n^{th} cluster, i.e., $v_n \in \{0,1\}^m$ only has nonzero values for points belonging to the n^{th} cluster. With $V = (v_1, ..., v_N) \in \mathbb{R}^{m \times N}$ denoting the set of all indicator vectors associated with the N clusters, the loss function can be written as

$$\mathcal{L}(V, W) = \frac{1}{N-1} \sum_{n=1}^{N} \frac{v_n^\top (D - W) v_n}{v_n^\top D v_n}, \tag{2.5}$$

where $D = diag(W\mathbf{1})$, ($\mathbf{1} = (1, 1, ..., 1)^\top \in \mathbb{R}^m$), is a diagonal matrix, whose i^{th} diagonal element is the sum of all elements in the i^{th} row of W. The value of $\mathcal{L}(V, W)$ is always between zero and one, with lower values indicating higher clusterability. A more intuitive objective measure can be defined as $1 - \mathcal{L}(V, W)$, such that higher values are associated with better clusterability.

2.5. Common Fate Transform

In this section, I provide a brief introduction to the Common Fate Transform (CFT), proposed by Stöter et al. [103] and study the prominent characteristics of audio representation in this transform domain, which make its use beneficial for the task of audio source separation.

As mentioned earlier, the CFT maps the signal energy from the time-frequency domain into a 4D space based on the common fate principle. The time-frequency components are, therefore, grouped based on their moving directions and mapped into different points in the target domain. Such a grouping property, which arises from the use of the 2D Fourier transform is in particular advantageous when dealing with mixtures of frequency-modulated harmonic signals. Since components of harmonic signals move up and down together in the time-frequency domain, they are likely to be mapped into the same locations in the scale-rate domain, causing harmonic elements of the same signal to group together in this representation. Such a mapping potentially increases the separability and/or clusterability of the data points, and hence makes it easier to isolate only those sound components belonging to the target source.

To formulate the transform, let us denote a single channel time-domain audio signal by $x(t)$ and its complex time-frequency-domain representation by $X(\omega, \tau) = |X(\omega, \tau)|e^{j\angle X(\omega,\tau)}$, where ω, τ, $|.|$, and $\angle(.)$ respectively denote frequency, time, the magnitude and phase operators. In the original version of CFT [103], $X(\omega, \tau)$ is defined as the STFT of $x(t)$. Due to the Hermitian symmetry of the Fourier transform of real signals, only the values of $X(\omega, \tau)$ for positive frequencies are stored for future processing.

In the following step, 2D windows, overlapped along both frequency and time axes are applied to $X(\omega, \tau)$. The 2D Fourier transforms of windowed segments are then computed and concatenated to form a 4D tensor. To keep the terminology and notation consistent throughout this work, the 2D Fourier transform domain will be referred to as the *scale-rate* domain. The scale and rate dimensions explicitly encode the spectro-temporal modulation information, where the former captures the spectral spread and the latter the modulation velocity

Transform	Input	Computation Steps	Output
CFT	$x(t)$	STFT $\rightarrow$ 2D windows centered at (Ω, T) $\rightarrow$ $\mathcal{FT}_{2D}$	$Y(s, r, \Omega, T)$
ICFT	$Y(s, r, \Omega, T)$	$\mathcal{IFT}_{2D}$ $\rightarrow$ 2D overlap and add $\rightarrow$ ISTFT	$x(t)$

Table 2.1. An overview of the computation steps in CFT and ICFT.

over time (see Section 2.7). Let $Y(s, r, \Omega, T)$ denote the 4D representation generated by the CFT. Here, (s, r) indicates the scale-rate coordinate pair and (Ω, T) the 2D window centers along the frequency and time axes.

It should be noted that the CFT is perfectly invertible. The single-sided complex STFT, $X(\omega, \tau)$, can be reconstructed from $Y(s, r, \Omega, T)$ by taking the 2D inverse Fourier transform of all patches and then performing 2D overlap-and-add. Subsequently, the time-domain signal, $x(t)$, can be obtained by taking the 1D inverse Fourier transform of all time-frames and performing 1D overlap-and-add. The operations performed in the CFT and the ICFT computation are summarized in Table 2.1.

In the remainder of this section, I present illustrative examples of taking the 2D Fourier transform of a time-frequency representation. This will provide the reader a more intuitive understanding of this domain. In these examples, I consider the 2D representation domains in isolation and compare their properties. This approach is taken mainly due to the difficulty of higher-dimensional visualization. However, it is important to note that merely going from the time-frequency domain to the scale-rate domain does not necessarily result in better separability or clusteribility. The power of the 4D representations studied in this work (CFT and MCFT) lies in combining the information from the scale-rate domain and the time-frequency domain. An analogy can be drawn to the example of Figure 2.5, Panels (c) and (d). Panel (c) shows that merely going from the time domain to the frequency domain does not completely solve the problem of overlapping energy. High separability is achieved when the time-domain information is processed over short windows and then combined with the frequency-domain information, resulting in a higher dimensional representation, displayed in Panel (d).

Figure 2.7 shows two harmonic signals, one with and one without frequency modulation, and their mixture. Both signals have a fundamental frequency of 400 Hz, and hence overlap significantly in the time-frequency domain. The magnitude STFTs of the three signals are depicted in the top row and the 2D Fourier transforms of magnitude STFTs in the bottom row. As it can be seen, the energy of the non-modulated source, represented by horizontal lines in the time-frequency domain is mapped into the zero-rate

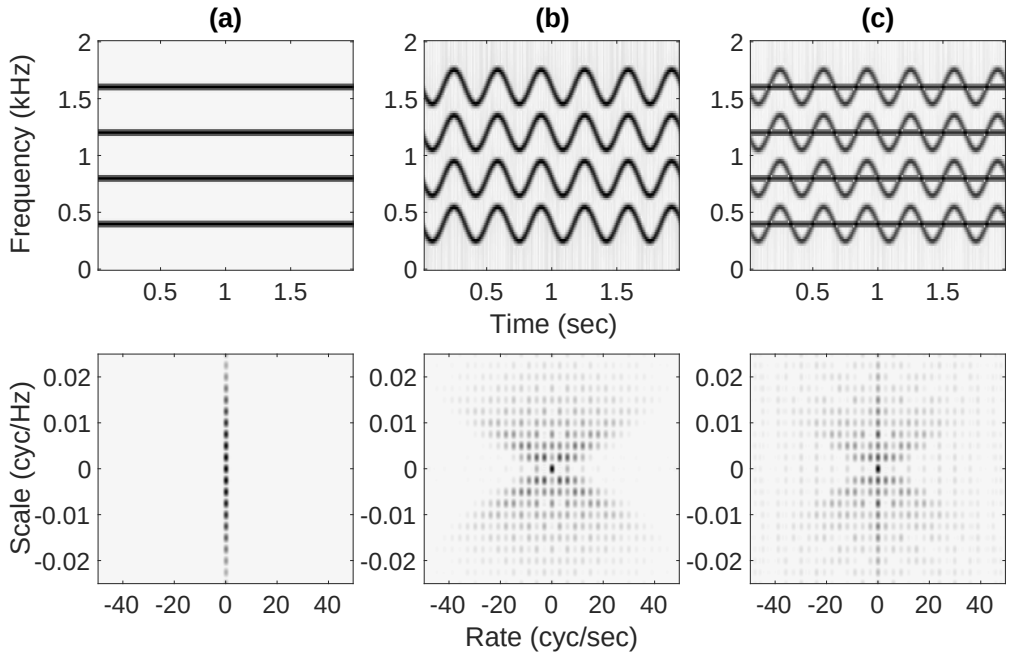

Figure 2.7. Two harmonic signals, with and without frequency modulation and their mixture. The two signals have the same fundamental frequency (400 Hz). Columns (a) and (b) show the two separate signals and column (c) shows their mixture. Signals are represented in the time-frequency domain (top row) and in the scale-rate domain (bottom row).

line, whereas the energy of upward or downward moving ripples of the modulated source is mapped to points scattered over non-zero rate values.

Figure 2.8 illustrates two crossing linear group-chirps moving in opposite directions and their mixture. Each group of linear chirps with the same slope is considered as one source. The two sources overlap at various points in the time-frequency domain. The plots in columns (a) and (b) show the two sources in the time-frequency domain (top row) and scale-rate domain (bottom row) and the plots in column (c) show the mixture. Each line in the "X" shape pattern that emerges in the scale-rate-domain representation of the mixture corresponds to one moving direction. In this case, going from the time-frequency domain to the scale-rate domain increases separability to some extent by remapping the components based on their moving directions, and thus reducing the number of overlapping points down to one. One might argue that the clusterability is also increased since the energy from parallel lines in the time-frequency domain, regardless of their relative spacing, is mapped into a single line in the scale-rate domain.

In the above examples, only the effect of applying the 2D Fourier transform to the magnitude STFT is considered. Nevertheless, it should be noted that the CFT is computed from the complex STFT, where the

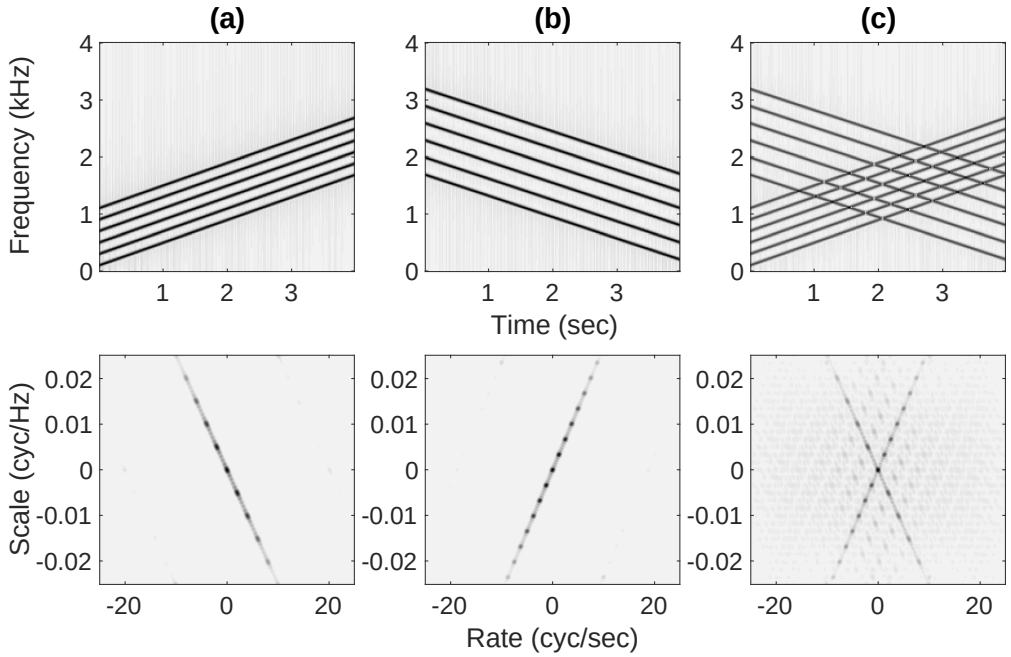

Figure 2.8. Two linear group-chirps (linear chirps with the same slope) with upward and downward moving directions, each considered a separate source, and their mixture. The two sources have energy overlap at various time-frequency points. Columns (a) and (b) show the two separate signals and column (c) shows their mixture. Signals are represented in the time-frequency domain (top row) and in the scale-rate domain (bottom row). We note that the opposing directions of signal representations with respect to the vertical axis in the original and transform domains (i.e. upward in one and downward in the other) is adopted to be in consistence with the image processing literature, although we admit that it might seem counterintuitive to readers that are not familiar with multi-dimensional signal processing concepts.

inclusion of the phase would alternate previously observed patterns in the scale-rate domain. This is what renders the time-frequency-domain audio representation more challenging to analyze through the 2D Fourier transform than photographic images, which are typically 2D real signals.

Figure 2.9 shows the same example as in Figure 2.7 along with the scale-rate-domain representation of the complex STFT. It can be observed that including the time-frequency-domain phase results in a shift in the scale-rate domain. The scale-rate-domain representation is still expected to offer more separability for the components overlapping in the time-frequency domain, although it seems to have lost the nice clusterability property of the magnitude-only case. The experimental findings discussed in Section 2.8 confirm this expectation. That is, in going from the *complex* STFT domain to the CFT domain the results show an increase in separability, although there is a possibility for the loss of clusterability. A general study of the time-frequency phase is beyond the scope of this work and would be the subject of future research.

Similar to the frequency resolution of the STFT which is determined by the time-domain window size, the scale and rate resolutions are determined by the dimensions of the time-frequency-domain windows. Consequently, the choice of the 2D window size has a direct impact on the representation quality of the CFT in terms of the provided separability.

The effect of the window size on the transform-domain resolution is illustrated in Figure 2.10. Panel (a) presents the magnitude STFT of a frequency modulated harmonic signal with a fundamental frequency of 200 Hz. The 2D Fourier transforms of four windowed segments with different window dimensions are depicted in Panels (b)-(e). As it is clearly observed in the plots, an increase in the window size along the time or frequency axis results in an increased resolution along the rate or the scale axis respectively. In the case with the lowest resolution in both directions shown in Panel (b), the scale-rate-domain representation of the windowed segment is quite blurry and only a large peak at the center can be detected, whereas in higher resolution cases, e.g., Panel (e), a number of lower peaks associated with upward and downward moving components also appear in the plot. It can also be seen that each window, depending on its duration over

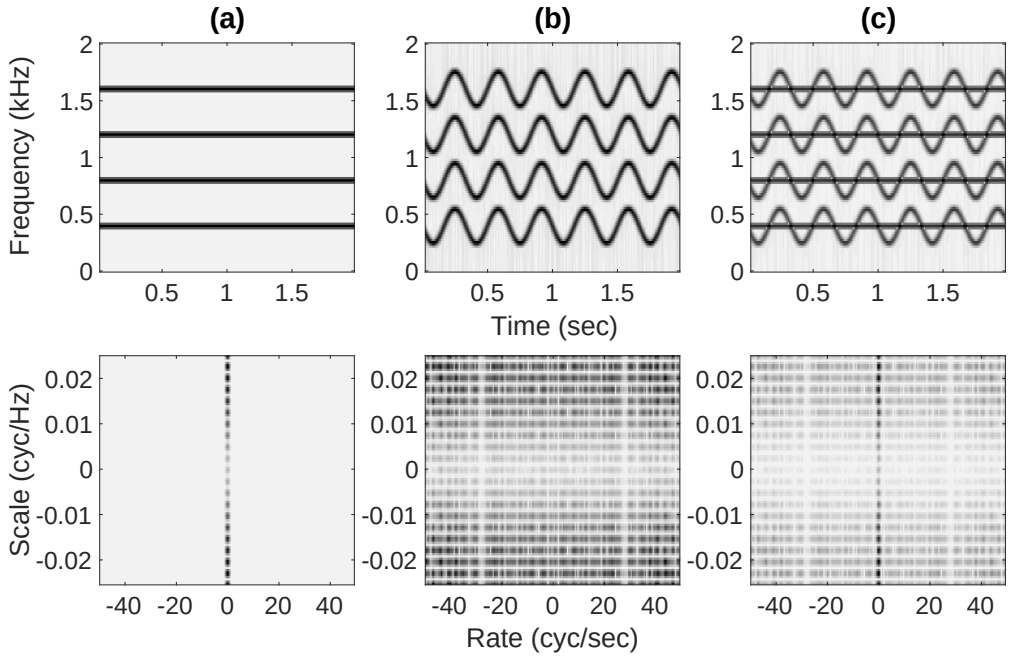

Figure 2.9. The top row shows the two sources and their mixture in the time-frequency domain. The bottom row shows the 2D Fourier transform magnitude of the *complex* STFT of the signal. Compare this to Figure 2.7, which shows the 2D Fourier transform of the *magnitude* STFT of the same signal.

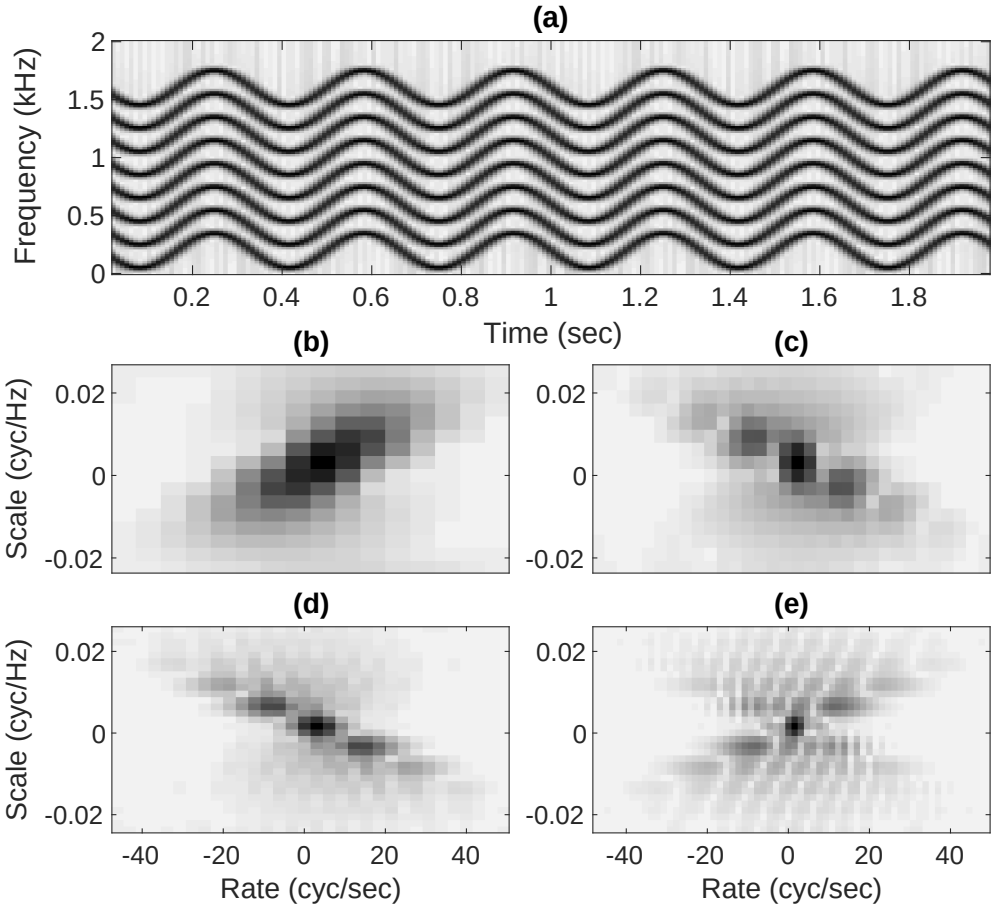

Figure 2.10. The effect of 2D-window size on resolution of the scale-rate-domain representation of the magnitude STFT. Panel (a) shows the magnitude STFT of a harmonic, frequency-modulated signal with a fundamental frequency of 200 Hz. Panels (b-e) show the 2D Fourier transform magnitude of the signal over window sizes of 16×16, 16×32, 32×32, and 32×64 respectively.

time captures one or both moving directions. For instance, the upward direction is not emphasized by the short window of Panel (b) as strongly as it is by the longer windows of Panels (c) and (d).

No general guideline for choosing the window size is proposed by Stöter et al. [103], as the ideal window clearly depends on the signal content. In the two following sections, I show how my proposed multi-resolution approach in computing the time-frequency representation as well as the 4D representation largely eliminates the need to select the right window size.

2.6. The auditory model of Chi et al.

The design of the MCFT was inspired by the multi-resolution aduitorty model of Chi et al. [9]. Recent studies on the primary auditory cortex of small mammals have shown the important role spectro-temporal

modulation patterns play in audio perception and streaming [13] [15] [9]. In [9], Chi et al. present a computational model of early and central stages of the auditory system. The model outputs a 4D multi-resolution representation capturing spectro-temporal modulation patterns.

Their auditory model is composed of two stages: cochlear and cortical. The cochlear stage, as the name suggests, emulates the cochlear filter bank in performing spectral analysis on the input time-domain audio signal. The filter bank model is composed of 128 overlapping constant-Q bandpass filters, with logarithmically-spaced center frequencies. The collective passband of filters covers approximately 5.3 octaves. The goal of the cochlear stage in the model is to replicate, as accurately as possible, the time-frequency-domain representation of the audio signal generated by the cochlea and termed *auditory spectrogram*. To this end, additional operations such as high-pass filtering, nonlinear compression, half-wave rectification, and integration are performed on the output of the filter bank. These operations model the effect of processes taking place between the inner ear and midbrain.

The cortical stage replicates the type of analysis performed by the primary auditory cortex. The neuronal response of the primary auditory cortex to different spectro-temporal modulation patterns, termed Spectro-temporal Receptive Fields (STRFs), can be regarded as a bank of 2D filters. The role of the filter bank is to extract the spectro-temporal modulation patterns from the auditory spectrogram. Each filter within the filter bank is tuned to a particular modulation pattern. The time-frequency-domain impulse responses of the filters in the auditory model are modeled after STRFs ([13]).

STRFs are mainly characterized by: 1) their spectral spread (broad/narrow), referred to as *scale* 2) their frequency modulation velocity over time (slow/fast), referred to as *rate* 3) their moving direction in the time-frequency plane (upward/downward). Spectro-temporal modulation patterns are, therefore, described in terms of their scale and rate values, measured in cycles per octave and cycles per second, respectively. Scale and rate form the two additional dimensions (besides time and frequency) in the 4D output of the auditory model. The STRF models proposed in [9] play the central part in the multi-resolution analysis of modulation patterns. It is, therefore, important to go into some technical detail in this section about the computation of the model.

Let us denote an STRF that is tuned to an arbitrary scale-rate parameter pair (S, R) by $h(\omega, \tau; S, R)$ with ω and τ denoting the frequency and time respectively. Note that S and R are constant (scalar) values for a single filter and determine the filter characteristics (i.e., spectral spread, frequency modulation velocity,

and moving direction). We denote the 2D Fourier transform of the STRF by $H(s, r; S, R)$, where the pair (s, r) indicates an arbitrary point in the transform (scale-rate) domain. The parameter pair (S, R), which is the same for h and H indicates the filter center in the scale-rate domain.

Mainly due to their diagonal movement in the time-frequency plain, STRFs cannot be modeled as separable functions of frequency and time, that is, $h(\omega, \tau)$ cannot be stated as $h(\omega, \tau) = f(\omega) \cdot g(\tau)$. In other words, more than one principal component would be required for describing the time-frequency-domain representation of an STRF. Nevertheless, the 2D Fourier transforms of STRFs are quadrant separable, meaning that they are separable functions of scale and rate in each quadrant of the scale-rate domain.

To derive the filter impulse response, first the spectral and temporal seed functions are to be defined. Chi et al. modeled the spectral seed function as a Gabor-like filter

$$f(\omega; S) = S \cdot (1 - 2(\pi S\omega)^2)e^{-(\pi S\omega)^2}, \tag{2.6}$$

and the temporal seed function as a gammatone filter,

$$g(\tau; R) = R \cdot (R\tau)^2 e^{-\beta R\tau} \sin(2\pi R\tau). \tag{2.7}$$

The dilation factors of the Gabor-like and gammatone filters in the above equations, S and R, are in fact the filter centers in the scale-rate domain. The dropping rate of the temporal envelop, or equivalently the filter bandwidth in the scale-rate domain, is controlled by the time constant of the exponential term, β. Since STRFs are not separable functions of frequency and time, the moving direction (up/down) of the time-frequency-domain components cannot be captured by a simple product of the seed functions. However, the quadrant separability of these functions allows computing their 2D Fourier transform as the product of the 1D Fourier transforms of the seed functions. This operation can be formulated as

$$F(s; S) = \mathcal{FT}_{1D}\{f(\omega; S)\}, \tag{2.8}$$

$$G(r; R) = \mathcal{FT}_{1D}\{g(\tau; R)\}, \tag{2.9}$$

$$H(s, r; S, R) = F(s; S) \cdot G(r; R), \tag{2.10}$$

where $\mathcal{FT}_{1D}$ denotes the 1D Fourier transform.

To generate the time-frequency-domain representation of an up-/down-ward moving filter, the value of H over a pair of opposing quadrants must be set to zero. The scale-rate domain response of the upward-moving filter, indicated by $H^{\Uparrow}(s, r; S, R)$ is defined as

$$H^{\Uparrow}(s, r; S, R) = \begin{cases} H(s, r; S, R) & (s \geq 0, r \leq 0) \\ H(s, r; S, R) & (s < 0, r > 0) \\ 0 & otherwise. \end{cases} \tag{2.11}$$

Similarly, the response of the downward filter, $H^{\Downarrow}(s, r; S, R)$ can be defined as

$$H^{\Downarrow}(s, r; S, R) = \begin{cases} H(s, r; S, R) & (s \geq 0, r \geq 0) \\ H(s, r; S, R) & (s < 0, r < 0) \\ 0 & otherwise. \end{cases} \tag{2.12}$$

In the next step, the impulse responses are computed as

$$h^{\Uparrow}(\omega, \tau; S, R) = \Re\{\mathcal{IFT}_{2D}\{H^{\Uparrow}(s, r; S, R)\}\} \tag{2.13}$$

$$h^{\Downarrow}(\omega, \tau; S, R) = \Re\{\mathcal{IFT}_{2D}\{H^{\Downarrow}(s, r; S, R)\}\} \tag{2.14}$$

where $\Re\{.\}$ denotes the real part of a complex value, and $\mathcal{IFT}_{2D}\{.\}$ the 2D inverse Fourier transform.

Examples of 2D filter impulse responses (STRFs) for different values of S and R are presented in Figure 2.11. Panels (a) and (b) correspond to upward moving filters, both with a rate of 4 cycles per second (a full cycle of the sinusoidal pattern covers 0.25 seconds). Panels (c) and (d) show downward moving filters with a rate of 2 cycles per second. In all panels, the frequency is shown on a logarithmic scale based on a reference frequency f_0, which maps frequencies that are separated by multiple octaves (an octave is a power of 2 relationship between frequencies) to a linear scale. The scale value for Panels (a) and (c) is 0.5 cycles per octave (a full cycle of 2 octaves), while Panels (b) and (d) demonstrate filters with a scale of 2 cycles per octave.

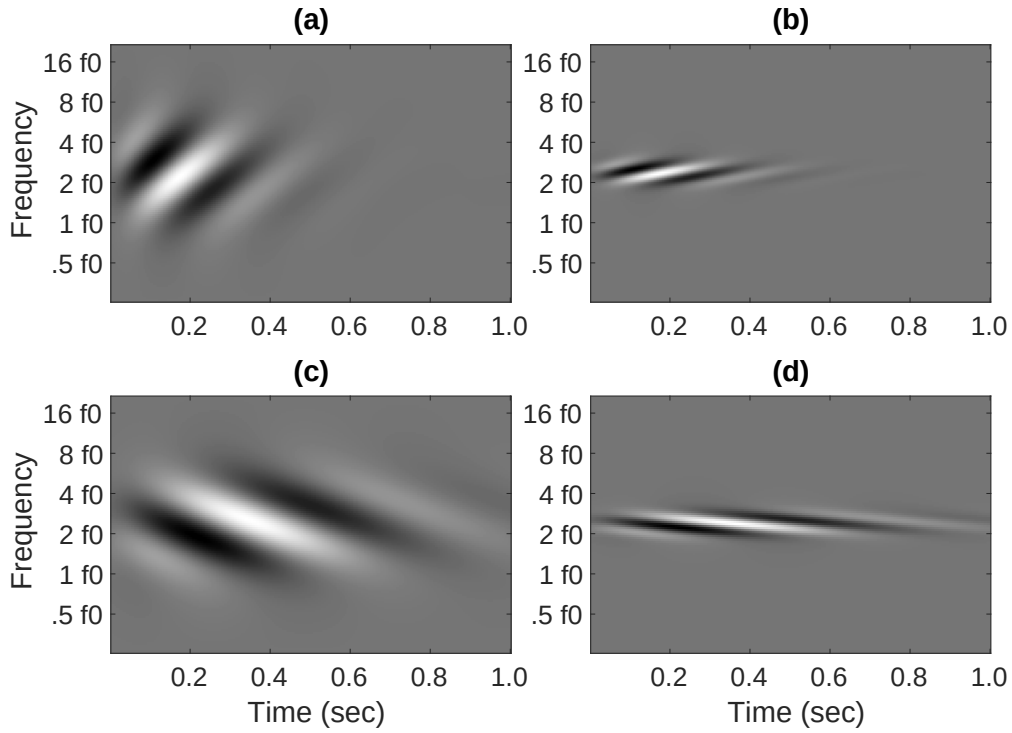

Figure 2.11. Impulse responses, known as Spectro-temporal Receptive Fields (STRFs), of four filters from the 2D filter bank: (a) Upward-moving STRF $h^{\Uparrow}(\omega, \tau; S = 0.5, R = 4)$ (low scale, high rate). (b) Upward-moving STRF $h^{\Uparrow}(\omega, \tau; S = 2, R = 4)$ (high scale, high rate). (c) Downward-moving STRF $h^{\Downarrow}(\omega, \tau; S = 0.5, R = 2)$ (low scale, low rate). (d) Downward-moving STRF $h^{\Downarrow}(\omega, \tau; S = 2, R = 2)$ (high scale, low rate). The frequency is displayed on a logarithmic scale based on a reference frequency f_0.

To compute the output representation, a bank of 2D filters, computed as described for various (S, R) values and different moving directions is applied to the auditory spectrogram. The 4D output of the cortical stage is denoted by $Z(S, R, \omega, \tau)$, where (S, R) give the filter center in the scale-rate domain. Note that since the fast Fourier transform has a lower computational complexity than convolution, filtering can be performed more efficiently in the scale-rate domain.

The main disadvantage of the output representation of the auditory model, which hinders its use in signal processing tasks such as source separation, is the lack of invertibility. The non-linear operations in the cochlear stage and the removal of phase-related information makes perfect reconstruction of the original audio waveform from their representation impossible. An algorithm for estimating the time-domain signal from the 4D output representation is proposed in [9]. Unfortunately, the quality of the estimated audio signal is not acceptable for audio processing applications.

2.7. Multi-resolution Common Fate Transform

In this section, I propose a new representation, which circumvents the shortcomings of the Common Fate Transform (CFT) and the auditory model output and combines their strengths. To address the invertibility issue, the auditory spectrogram is replaced by a fully invertible complex time-frequency representation with log-scale frequency. The Constant-Q Transform (CQT) is a multi-resolution time-frequency representation, where the resolution is progressively more coarse-grained as frequency increases. The log-scale frequency spacing of the CQT is similar to the frequency spacing of the auditory spectrogram in Chi et al. auditory model (see Section 2.6). Unlike the auditory spectrogram, however, the CQT captures the phase. In the implementation, I use the CQT as proposed by Schörkhuber et al. [91], which is fully invertible back to the time domain.

To compute the new 4D representation, the cortical filter bank of the auditory model is applied to the complex CQT of the audio signal. This new representation is termed the Multi-resolution Common Fate Transform (MCFT), and denoted by $\tilde{Z}(S, R, \omega, \tau)$. The MCFT addresses the resolution issues of the CFT in the time-frequency domain as well as the scale-rate domain. The linear-scale frequency of the STFT offers a fixed resolution for the whole range of musical notes. Given that the fundamental frequency of musical notes are distributed on a logarithmic scale, the STFT would not be able to resolve low-frequency notes as effectively as high-frequency notes.

The use of a multi-resolution 2D filter bank instead of fixed size 2D windows in the spectro-temporal modulation analysis stage allows the representation of 2D sinusoidal patterns with high as well as low frequencies at the same time (highly localized and long-term modulation patterns), and thus results in an improvement in the scale-rate domain resolution of the MCFT compared to the CFT (see Figure 2.10). The difference between the modulation analysis stages in the MCFT and CFT is analogous to the difference between the frequency analysis stages in the CQT and STFT, in that one of the transforms performs the short-term analysis through fixed-size windowing in the original domain, while the other by multi-resolution filtering in the transform domain.

The time-domain signal can be reconstructed from $\tilde{Z}(S, R, \omega, \tau)$ in two steps. First, the time-frequency representation is reconstructed from $\tilde{Z}(S, R, \omega, \tau)$ by inverse filtering:

$$\hat{X}(\omega, \tau) = \mathcal{IFT}_{2D}\left\{\frac{\sum_{S,R}^{\Uparrow\Downarrow} \tilde{z}(s, r; S, R) H^*(s, r; S, R)}{\sum_{S,R}^{\Uparrow\Downarrow} |H(s, r; S, R)|^2}\right\}, \tag{2.15}$$

Transform	Input	Computation Steps	Output
MCFT	$x(t)$	CQT $\rightarrow$ $\mathcal{FT}_{2D}$ $\rightarrow$ 2D filters centerd at (S,R) $\rightarrow$ $\mathcal{IFT}_{2D}$	$\tilde{Z}(S,R,\omega,\tau)$
IMCFT	$\tilde{Z}(S,R,\omega,\tau)$	$\mathcal{FT}_{2D}$ $\rightarrow$ 2D inverse filters centered at (S,R) $\rightarrow$ $\mathcal{IFT}_{2D}$ $\rightarrow$ ICQT	$x(t)$

Table 2.2. An overview of the computation steps in MCFT and IMCFT.

where $*$ is complex conjugate, $\tilde{z}(s,r;S,R)$ denotes the 2D Fourier transform of $\tilde{Z}(\omega,\tau;S,R)$ for a particular (S,R), and $\sum_{S,R}^{\Uparrow,\Downarrow}$ indicates summation over the whole range of (S,R) values and all up-/down-ward filters. The time domain signal is then reconstructed from $\hat{X}(\omega,\tau)$ using the Inverse Constant-Q Transform (ICQT) proposed in [91]. Table 2.2 gives a summary of operations performed in the MCFT and IMCFT computation.

In previous sections, mostly the scale-rate-domain behavior of the magnitude of the time-frequency representation was studied. Including the phase in the time-frequency domain results in a shift in the location of scale-rate-domain components (see Figure 2.9). Including the phase, however, allows for invertibility of the CFT and MCFT back to the time domain, which in turn allows separation to be performed in the 4D domain, where separability is improved, compared to the time-frequency domain. The cost is that including phase may introduce some scattering to the patterns in the representation that could potentially reduce clusterability. In practical use, this potential reduction in clusterability is outweighed by the improvement for separability, as illustrated in the experiments in Section 2.8. An in depth study of the phase behavior for all types of audio signals is beyond the scope of this work.

The method used in this work to deal with the effect of phase is shifting the filter components in the scale-rate domain in accordance with the shift in the location of mixture components. This can be achieved through modulating the filters with the phase of the mixture CQT, i.e., using filters with impulse responses equal to $h(\omega,\tau;S,R)e^{j\angle X(\omega,\tau)}$, where $\angle(.)$ denotes the phase operator. Panel (a) in Figure 2.12 shows the magnitude CQT of a mixture of harmonic and non-harmonic signals. Panels (b), (d), and (f) present the output of three filters applied to the magnitude CQT. The upward and downward moving components of the modulated source are clearly separated from the components of the non-modulated source. Panels (c), (e), and (g) demonstrate the outputs of three modulated filters applied to the complex CQT. Although the emerged modulated patterns look slightly different from the output of the original filters in the left column, they are still successfully separated from the non-modulated components. Furthermore, it is worth noting that due to phase preservation, the CQTs in the right column are fully invertible to the time domain, while this is not the case for the CQTs in the left column and not true for the auditory model of Chi et al.

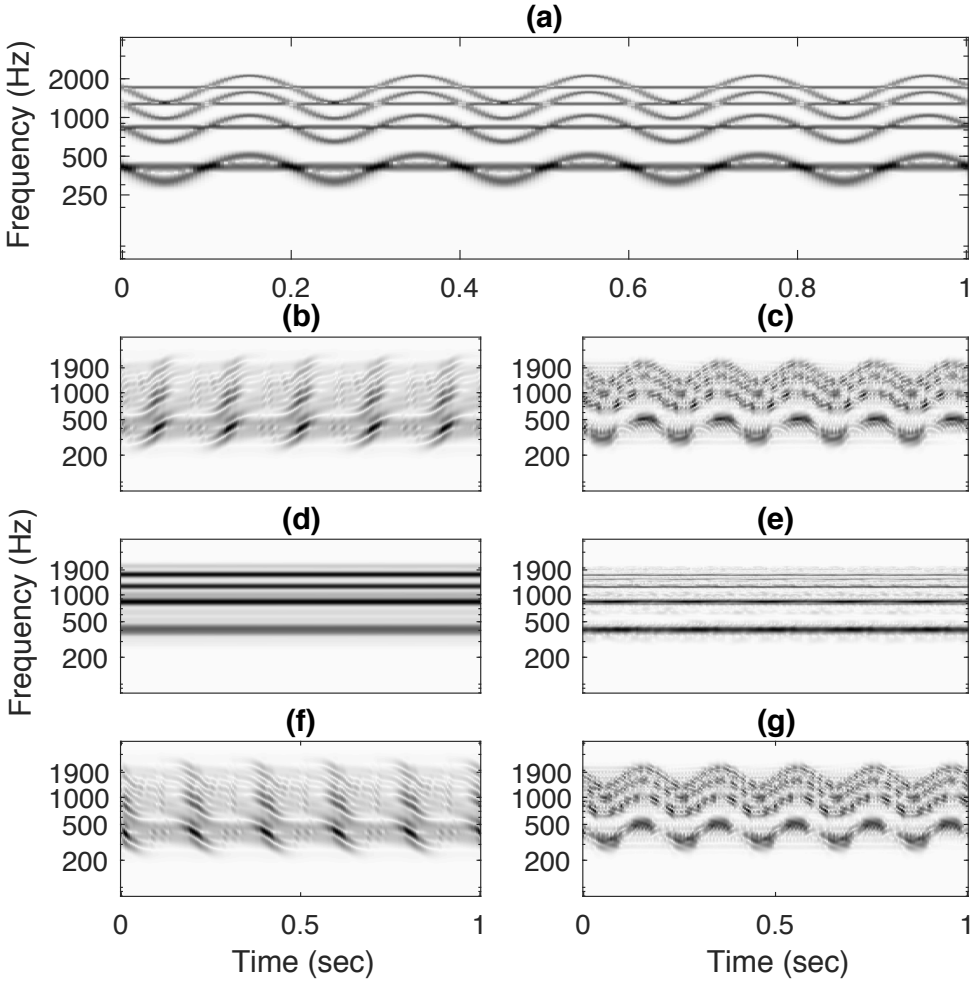

Figure 2.12. (a) Magnitude spectrogram of a mixture of two harmonic sources one with frequency modulation (the sinusoid shaped lines) and one without frequency modulation (the straight lines). (b,d,f) Magnitude spectrograms of the filtered mixture. Filters are applied to the magnitude spectrogram. (c,e,g) Magnitude spectrogram of the filtered mixture. Filters are first modulated with the mixture phase and then applied to the complex spectrogram.

Representation	Output Dimensionality	Basic Auditory Cues	Invertibility	Multi-resolution Frequency Analysis	Common Fate Property	Multi-resolution Scale-Rate Analysis
SP - STFT	2	t, f	✓	-	-	-
SP - CQT	2	t, f	✓	✓	-	-
SP - CFT	4	t, f, tf-mod	✓	-	✓	-
Aud - Chi	4	t, f, tf-mod	-	✓	✓	✓
MCFT	4	t, f, tf-mod	✓	✓	✓	✓

Table 2.3. Audio representations and their properties; Aud. denotes Auditory-model-based. Time, frequency, and spectro-temporal modulation are respectively indicated by t, f, and tf-mod.

2.8. Experimental Validation

In this section, I examine the separability and clusterability of unison mixtures of instrumental sounds played with different techniques when they are encoded in four different representations. Two are commonly used for source separation: the STFT and the CQT. The other two are common-fate-based representations: the CFT and the proposed MCFT. The auditory model of Chi et al. is not included in the experiments because audio encoded in this model cannot be perfectly reconstructed. Table 2.3 summarizes the properties of all the representations discussed in this chapter.

2.8.1. Dataset

In the experiments, I primarily focus on evaluating the efficacy of the MCFT in capturing spectro-temporal modulation patterns as higher dimensions and in using them as source separation cues in cases with high energy overlap in the time-frequency domain. Mixtures of instrumental sound sources played in unison (same pitch) but with different frequency modulation techniques (e.g., vibrato versus trill) are a good example of such cases. Such mixtures also happen to be one of the most challenging cases for state-of-the-art audio source separation algorithms. My next goal is to study the effect of the multi-resolution property of the MCFT, in the frequency domain as well as the scale-rate domain, on the separation quality and to compare its performance to CFT, which has fixed resolution at both stages. To this end, a wide range of musical octaves and a variety of modulation techniques are included in the dataset.

The testing dataset in my prior work [77] included a single pitch from a middle octave (D4 with a fundamental frequency of 293.66 Hz). In this work, the pitch range is extended to two lower and three higher octaves (6 octaves in total). The set of single sources used to generate the mixture dataset is composed of 68 orchestral instrument samples generated by the EastWest Symphonic Orchestra sampler [1], 7 samples selected from the Philharmonia Orchestra [2], and 6 piano samples recorded on a Steinway grand (81 samples in total). All samples are 2 seconds long and are sampled at 44.1 kHz.

The note C was chosen as a representative pitch class over octaves 2 to 7 (65.41 Hz to 2093 Hz). Table 2.4 presents the list of all instruments included in the dataset along with their playing techniques and octave coverage. The playing techniques include vibrato: continuous frequency modulation, trill: frequency modulation alternating between two adjacent pitches in the chromatic scale, and tremolo: amplitude (and

[1] http://www.soundsonline.com/symphonic-orchestra
[2] www.philharmonia.co.uk

Instrument	ModulationTechnique	Note	Instrument	Modulation Technique	Note
piano	-	C2, C3, C4, C5, C6, C7	english horn	vibrato, major trill, minor trill	C4, C5
contrabassoon	vibrato	C2	clarinet	major trill, minor trill	C4, C5, C6
contrabass	vibrato	C2, C3, C4	oboe	vibrato, major trill, minor trill	C4, C5, C6
bassoon	vibrato, major trill, minor trill	C2, C3, C4, C5	trumpet	vibrato	C4, C5, C6
cello	vibrato	C2, C3, C4, C5, C6	saxophone	major trill, minor trill	C5
viola	major trill, minor trill	C3, C4, C5, C6	trombone	tremolo	C5
tuba	minor trill	C3, C4	piccolo trumpet	major trill, minor trill	C5, C6
tuba	major trill	C4	piccolo flute	vibrato, major trill, minor trill	C6, C7
saxophone	tremolo	C4	violin	vibrato, major trill, minor trill	C7
flute	vibrato	C4, C5			

Table 2.4. Single sound sources used in generating the mixture datasets. Instruments are orded by the pitch of the lowest note used.

sometimes frequency) modulation. It should be noted that a *single sound source* in these experiments refers to a single note played by an instrument-technique pair, e.g., a C4-viola-major trill is considered a different source than a C4-viola-minor trill. It can be clearly observed that the number of samples per octave follows a bell-shaped distribution (there are respectively 7, 9, 21, 22, 15, 7 samples in octaves 2 to 7). This is because orchestral instruments have a limited pitch range, as a result of which the number of samples for all pitch classes is much larger in middle octaves than in high/low octaves.

Due to the imbalance in the number of sources per octave, there is a large difference between the number of mixtures per octave. For instance, the total number of two-source mixtures ranges from 21 for the second and seventh octaves to 231 for the fifth octave. In the experiments, the number of mixtures is kept the same for all octaves by randomly selecting 21 mixtures (minimum number) in octaves 3 to 6. This gives rise to a testing dataset of size 126 two-source mixtures. To study the behavior of representations as the number of sources increases, I create three-, four-, and five-source-mixture datasets, each of size 126, following the same procedure described for two-source mixtures.

It should be taken into account that the MCFT is designed to explicitly capture frequency modulation. The testing dataset is thus almost entirely composed of frequency-modulated samples (vibrato, major trill, minor trill), such that the dominant effect on the behavior of separability and clusterability results can be attributed to frequency-modulation. Since tremolo is sometimes defined as amplitude and sometimes as frequency modulation, the MCFT is expected to provide improvement only if there is frequency modulation that is greater than the minimum detectible frequency change, which is controlled by the resolution of the underlying transform.

2.8.2. Audio Representations

In the experiments, the window length and overlap ratio of the STFT are set to 93 ms (4096 samples) and 75% respectively. At its time-frequency representation stage, the CFT uses the same parameter values.

To study the effect of the 2D window on separability and clusterability of the CFT, I experiment with a grid of values of the 2D window sizes including all combinations of $L_\omega \in \{2, 4, 8, 16, 32\}$ (21.6 Hz - 344.5 Hz) and $L_\tau \in \{4, 8, 16, 32, 64, 90\}$ (93 ms - 2 sec), where L_ω and L_τ denote the window widths, along the frequency axis and time axis respectively. I present the results for the best and worst window sizes. There is 50% overlap between windows in both dimensions.

For computation of CQTs, I use the MATLAB toolbox in [91]. The minimum and maximum frequencies are respectively set to 61.74 Hz (note B1) and 4435 Hz (note C$^{\#}$8) to cover the whole range of pitches included in the dataset. The frequency resolution of the CQT is set to 96 bins per octave. One observation in these experiments was that for both time-frequency representations higher frequency resolutions result in better separation results for this particular type of mixture (harmonic sound). Therefore the frequency resolution was increased to a point where the decreasing time resolution starts to harm the performance.

The same CQT parameter values are used in the time-frequency representation stage of the MCFT. In the modulation analysis stage, the MCFT uses a spectral filter bank $F(s; S)$ including a lowpass filter centered at 2^{-4} (cyc/oct), 6 bandpass filters at $2^0, 2^1, ..., 2^5$ (cyc/oct), and a highpass filter at $2^{5.5}$ (cyc/oct). The temporal filter bank $G(r; R)$ is composed of a lowpass filter centered at 2^{-2} (cyc/sec), 5 bandpass filters at $2^0, 2^1, ..., 2^4$ (cyc/sec), and a highpass filter at $2^{4.5}$ (cyc/sec). The time constant parameter, β, is set to 1. The product of F and G gives rise to a 2D filter response, which is then split into two analytic filters (see Section 2.6). This set of parameters is selected to keep the filterbank size and consequently the MCFT size small (low scale and rate resolutions), while covering the entire range of scale and rate values (capturing the entire signal energy). Since one advantage of the MCFT is that it is inherently multi-resolution, I only used the single setting described above, rather than experimenting with 30 settings, as was done with the CFT. An implementation of the MCFT and audio examples from the experimental results have been provided in the accompanying website [3].

[3] https://interactiveaudiolab.github.io/MCFT

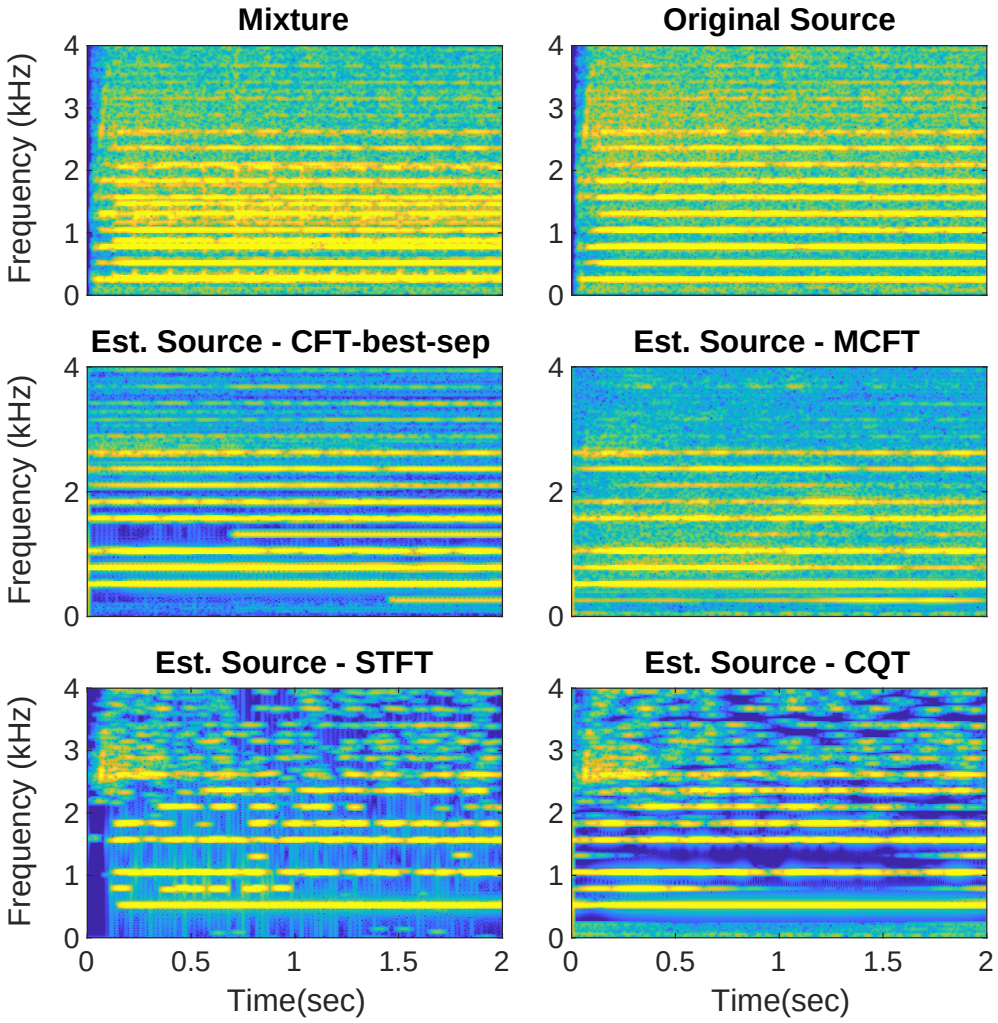

Figure 2.13. An example of separation via ideal binary masking with a threshold of $\gamma = 25$ dB for a mixture of C4-clarinet-major trill and C4-flute-vibrato. Magnitude spectrograms of the mixture (top left) and C4-flute-vibrato (top right). Magnitude spectrograms of the estimated source by applying the mask respectively in the CFT-best-sep (middle left), MCFT (middle right), STFT (bottom left), and CQT (bottom right) domains.

2.8.3. Separability Results

Figure 2.13 presents an example of source separation via ideal binary masking in different representation domains for one of the mixtures in our dataset. For easier visual comparison, all signals are presented in the STFT domain. The top row shows the mixture and the original signal, and the next two rows show the results of separation in the 4D and 2D domains. It can be observed that the MCFT preserves more of the signal energy and harmonic structure, and introduces fewer masking artifacts than other representations.

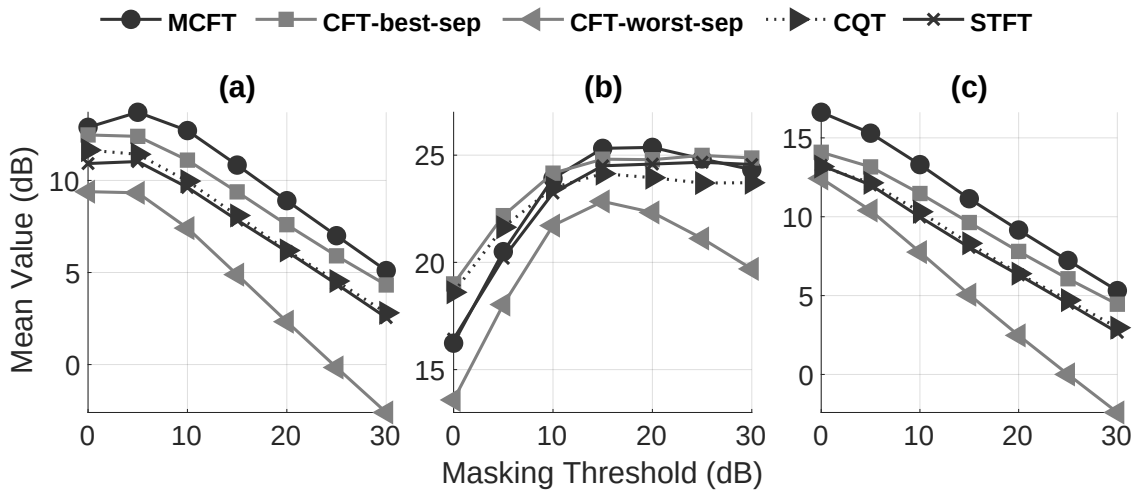

Figure 2.14. Measuring Separability for two-source mixtures as a function of masking threshold. Higher values are better. Mean (a) SDR, (b) SIR, and (c) SAR for 2D and 4D representations versus masking threshold, γ. The results for 2 out of 30 2D window sizes tried in CFT computation are presented: CFT-best-sep (4×64) and CFT-worst-sep (32×4).

The separability of the representations is evaluated over the testing dataset through ideal binary masking. The idea binary masking was chosen so that the separation results are only affected by the overlap of sources within the representation domain and not by the source separation algorithm (see Section 2.3). I use a range of threshold values (0 dB to 30 dB with a step of 5 dB) in the computation of ideal binary masks in each representation domain. Separation is performed through ideal binary masking in each representation domain and then the preserved energy level and separation quality of all reconstructed signals are compared in the time domain, so that the comparison is agnostic to the representation used for separation.

For time-domain evaluation of the separation performance, I use the BSS-Eval [111] objective measures: Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR), and Source to Artifact Ratio (SAR), which are commonly used in the literature to evaluate source separation methods. Higher values of these measures indicate better performance. It is worth mentioning that the scale-invariant versions of these measures have recently been introduced and are now commonly used for separation algorithm evaluation [52]. While the scale-invariant measures have been shown to give a more accurate assessment of estimated signals in realistic scenarios, here they show a similar behavior to the BSS-Eval measures since ideal binary masks are used for separation.

The mean SDR values over the whole dataset are used as a measure of separability. In the separability results the “CFT-best-sep" and “CFT-worst-sep" correspond to the window sizes (drawn from the set of 30

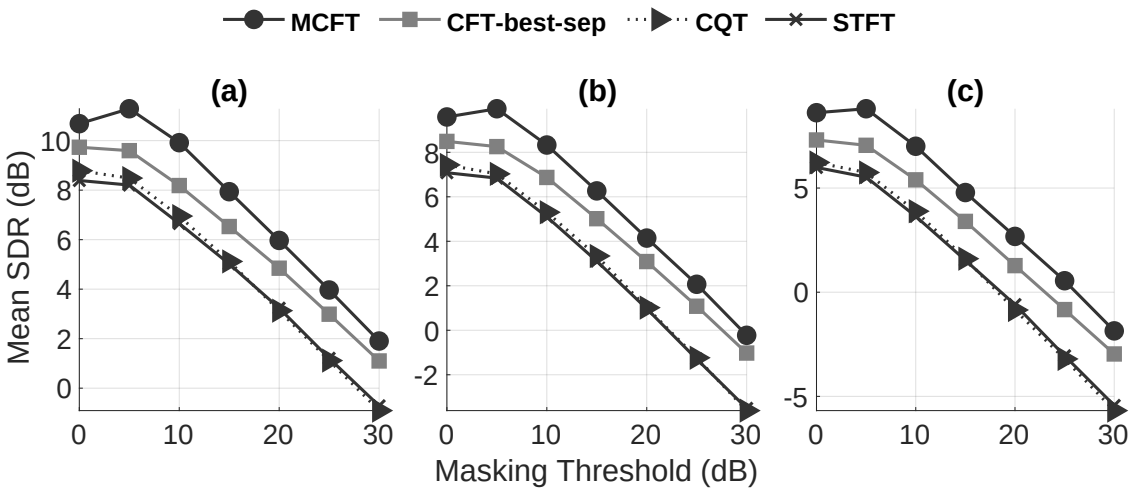

Figure 2.15. Mean SDR versus masking threshold for 2D and 4D representations over (a) three-, (c) four-, and (d) five-source mixture datasets. Higher values are better. Only the results for the best two-dimensional window size used in CFT computation (4×64) are presented.

window sizes mentioned earlier) that resulted in the best and worst mean SDR values. Figure 2.14 presents the mean values of separability metrics as a function of masking threshold used for the ideal binary mask, γ, for two-source mixtures. It can be clearly seen that the MCFT outperforms all other representations in terms of SDR and SAR at all threshold values and in terms of SIR for middle threshold values, e.g., 10 dB (i.e., the source energy must be 10 dB louder than the interference to be included in the ideal binary mask).

The reason why all representations have a better SIR performance for middle thresholds (15-20 dB) can be explained by considering the fact that low threshold values would let in a large amount of noise and interference along with the energy from the target source and high threshold values would remove a significant potion of the target signal energy, both resulting in a decrease in SIR. The performance of the CFT depends heavily on the 2D window size and ranges from much worse than the STFT to better than the CQT. Such dependency makes the use of the CFT less reliable in blind source separation scenarios, since it is highly sensitive to data-dependent settings to achieve maximal performance.

The mean SDR values over masking threshold for datasets with more than two sources per mixture are shown in Figure 2.15. While the performance of all representations degrades in general with an increase in the number of sources, the MCFT stays strictly dominant in all cases.

2.8.4. Clusterability Results

The clusterability measure defined in Section 2.4 is used to measure how well each representation groups together elements of a single source. Higher values mean better clusterability. The Gaussian kernel in Equation (2.5) along with the Euclidean distance measure are used in the computation of similarity values in the experimental results.

An increase in the similarity kernel width assigns higher weights to points farther from the center of each cluster and thus increases the likelihood of mislabeling points from neighboring clusters. On the other hand, increasing the masking threshold means removing lower-energy points, which are presumably located towards the boundaries of neighboring classes and therefore producing wider inter-cluster margins. I study the effect of the similarity kernel width, α, as well as the masking threshold, γ. Figure 2.16 demonstrates the mean clusterability values for two-source mixtures versus these two parameters. As it can be observed in Figure 2.16, an increase in the similarity width results in a drop in clusterability values for all representations, whereas an increase in masking threshold causes an increase in clusterability values. The MCFT seems to outperform the 2D representations over α values larger than 2 and γ values below 30 dB. The performance of the CFT is again dependent on the window size and can vary dramatically as shown by the results for the best- and worst- performing window sizes, where it goes from outperforming to underperforming all the other representations.

An interesting difference between the MCFT curve and others is that it almost levels out after 20 dB while the others keep increasing. This behavior is not unexpected since the MCFT tends to project the signal energy to a larger number of points in the higher-dimensional space and thus preserve a much larger portion of signal energy for higher thresholds compared to other representations (see Figure 2.14). This behavior is more noticeable in Figure 2.17 for mixtures composed of more than two sources.

Note, however, that higher separability (i.e., sources are not overlapped) is a neccessary precursor to high-quality source separation while higher clusterability (i.e., sources are in separate regions of the representation space) is strongly desireable, but not technically neccessary, depending on the sophistication of the separation algorithm.

The mean SDR and mean clusterability over the whole two-source dataset and all parameter values are respectively shown on the y-axis and x-axis in Figure 2.18. The plot depicts the mean performance for the STFT, CQT, MCFT, and CFT (all 30 2D window sizes). The bold dashed lines delimit the range of values

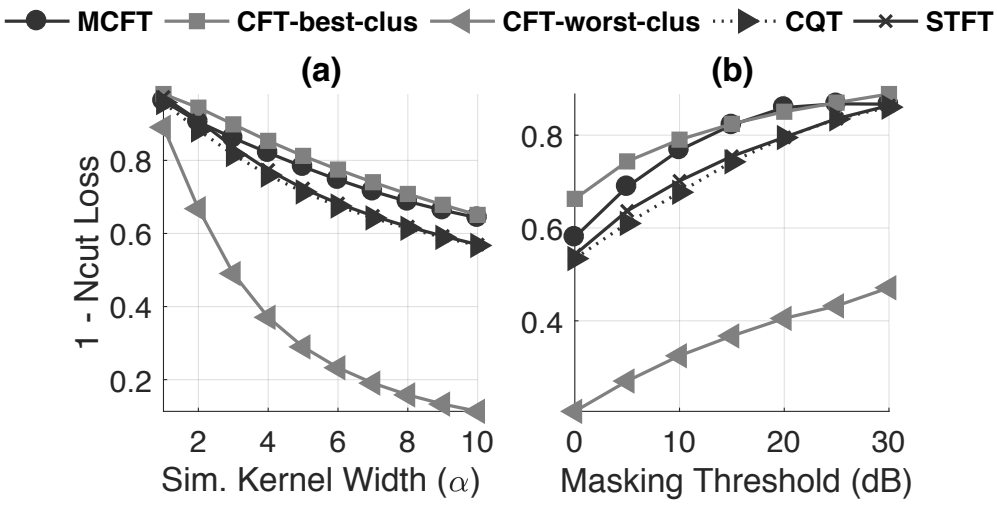

Figure 2.16. Mean clusterability for 2D and 4D representations versus similarity kernel width, α (a) and masking threshold, γ (b). Higher values are better. The results for 2 out of 30 2D window sizes tried in CFT computation are presented: CFT-best-clus (2×90) and CFT-worst-clus (32×8).

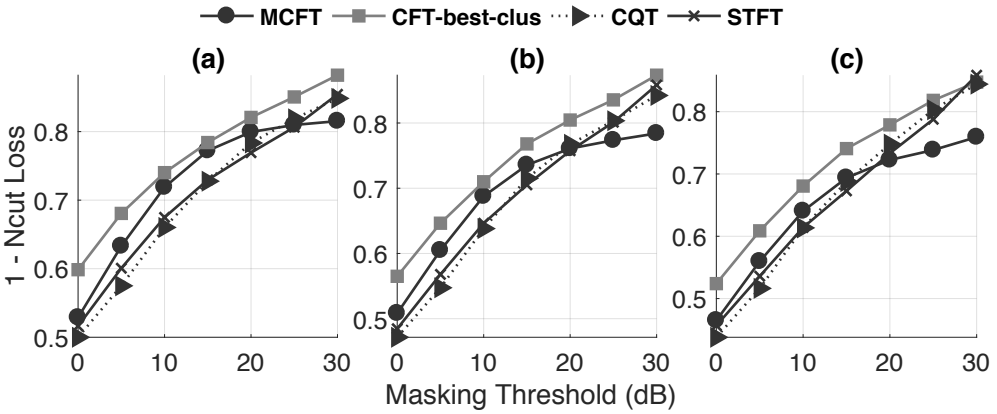

Figure 2.17. Mean clusterability versus masking threshold for 2D and 4D representations over (a) three-, (b) four-, and (c) five-source mixture datasets. Higher values are better. Only the results for the best 2D window sizes used in CFT computation (2×90) are presented.

that are inferior to the MCFT performance across both dimensions. The MCFT clearly outperforms all other representations in terms of separability and only underperforms the CFT in terms of clusterability for 2 out of 30 different window sizes. Even in these two cases, the clusterability is similar between CFT and MCFT, while MCFT strongly dominates the CFT on separability.

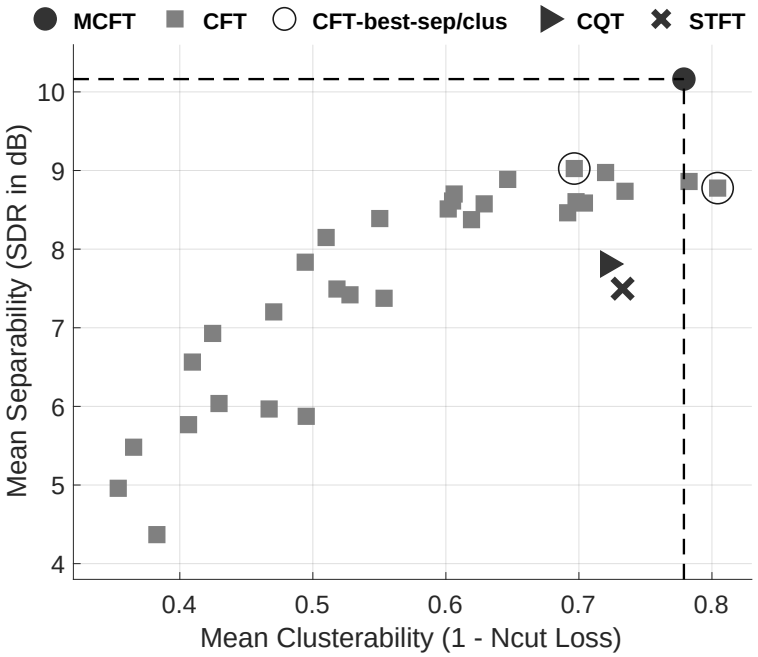

Figure 2.18. Mean SDR versus mean clusterability over all samples and masking thresholds for two-source mixtures. The results for all 30 2D window sizes used in CFT computation are presented, along with the results for the MCFT, CQT and STFT. Higher values are better in both dimensions.

2.8.5. Statistical Testing Results

To compare the distributions of separability and clusterability results for different representations, the Wilcoxon rank sum test was used. The results of statistical significance tests are presented in Tables 2.5 to 2.8. The null hypothesis in all statistical tests is that the median of the results for the MCFT is less than or equal to the median of other representations, or equivalently, the MCFT does not provide any improvement in separation performance compared to other representations.

In separability statistical tests, $n =$ *number of mixtures* $\times$ *number of masking thresholds* $= 126 \times 7 = 882$, and in clusterability statistical tests, $n =$ *number of mixtures* $\times$ *number of masking thresholds* $\times$ *number of similarity kernel widths* $= 126 \times 7 \times 10 = 8820$. In all tables, *median diff = median(MCFT) − median(other representation)* (positive values indicate improved performance for the MCFT).

The SDR values for the MCFT show significant improvement over all other representations and for all mixture types with $p \leq 0.0001$ in all cases. The MCFT performs significantly better on clusterability than CFT-best-sep for all mixture types with $p \leq 0.0001$, significantly better than CQT for two-, three-, and four-source mixtures with $p \leq 0.05$ in the worst case, and significantly better than the STFT for two-,

Table 2.5. SDR - Wilcoxon rank sum test results ($n = 882$)

		STFT	CQT	CFT best-sep	CFT best-clus
2src	median diff (dB)	+2.18	+2.11	+1.05	+1.27
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
3src	median diff (dB)	+2.42	+2.32	+1.22	+1.55
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
4src	median diff (dB)	+2.68	+2.60	+0.98	+1.32
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
5src	median diff (dB)	+2.80	+2.78	+1.20	+1.58
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001

Table 2.6. SIR - Wilcoxon rank sum test results ($n = 882$)

		STFT	CQT	CFT best-sep	CFT best-clus
2src	median diff (dB)	+0.16	+0.15	−0.57	−0.41
	p-value	> 0.05	> 0.05	> 0.05	> 0.05
3src	median diff (dB)	+0.79	+0.60	−0.13	+0.02
	p-value	≤ 0.0001	≤ 0.0001	> 0.05	> 0.05
4src	median diff (dB)	+1.09	+0.87	+0.11	+0.19
	p-value	≤ 0.0001	≤ 0.0001	> 0.05	> 0.05
5src	median diff (dB)	+1.52	+1.44	+0.46	+0.73
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.05	≤ 0.0001

and three-source mixtures with $p \leq 0.0001$. Although MCFT outperforms most other representations on clusterability, the CFT-best-clus improves on the MCFT in all cases with $p \leq 0.0001$.

Note that the superior clustering of CFT-best-clus is due to a careful selection of the window size in the presence of ground truth, which is typically not possible in real-world use. Moreover, separability for CFT-best-clus remains worse than the separability of the MCFT. Therefore, even if it is easier to cluster

Table 2.7. SAR - Wilcoxon rank sum test results ($n = 882$)

		STFT	CQT	CFT best-sep	CFT best-clus
2src	median diff (dB)	+2.59	+2.48	+1.46	+1.65
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
3src	median diff (dB)	+2.66	+2.62	+1.43	+1.74
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
4src	median diff (dB)	+2.67	+2.68	+1.02	+1.35
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001
5src	median diff (dB)	+2.83	+2.82	+1.32	+1.58
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	≤ 0.0001

Table 2.8. Clusterability - Wilcoxon rank sum test results ($n = 8820$)

		STFT	CQT	CFT best-sep	CFT best-clus
2src	median diff (dB)	+0.088	+0.100	+0.130	0.003
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	> 0.05
3src	median diff (dB)	+0.041	+0.040	+0.13	−0.031
	p-value	≤ 0.0001	≤ 0.0001	≤ 0.0001	> 0.05
4src	median diff (dB)	+0.012	+0.008	+0.098	−0.057
	p-value	> 0.05	≤ 0.05	≤ 0.0001	> 0.05
5src	median diff (dB)	−0.009	−0.017	+0.098	−0.077
	p-value	> 0.05	> 0.05	≤ 0.0001	> 0.05

energy from a single source, the resulting separation will have a ceiling of performance that is lower for other representations than for the MCFT.

2.9. Conclusion

The efficacy of source separation algorithms can be limited by the representation used for the input audio. A representation that reduces overlap and interspersal of sources can simplify the separation process and improve results. In this chapter, I presented the Multi-resolution Common Fate Transform (MCFT), a representation that is fully invertible and increases the separability of audio signals with significant time-frequency-domain overlap, through explicitly representing spectro-temporal modulation patterns. It was placed in the context of two existing common-fate-based models: the Common Fate Transform (CFT) and the auditory model of Chi et al. The MCFT, by being multi-resolution and fully invertible combines the strengths of both approaches.

I also introduced and provided metrics for two desirable properties of audio representations for source separation: separability and clusterability. Experiments on a dataset of unison mixtures of musical instrumental sounds showed that the MCFT strictly dominates the other representations on separability. It also outperforms other representations on clusterability in the majority of cases, without requiring data-dependant parameter setting to achieve these results. Given these results, the MCFT is a promising representation to be used as the input to source separation algorithms. Moving forward, I will use the MCFT as the input representation to a joint separation-calssification system (Chapter 4) in order to train a separator using weak (e.g., frame-level) class labels.

CHAPTER 3

Learning to Separate Sounds from Weakly Labeled Scenes

3.1. Introduction

In this chapter, I present a new algorithm for supervised audio source separation training, which does not require ground truth isolated sources as training targets and merely depends on weak labels indicating the activity of different sound types over time [79][80]. To bridge the gap between strong and weak labels, the algorithm relies on audio classification results as a metric for assessing the separation performance.

The auditory-inspired aspects of this new approach consists in: i. having a sound identification system to learn different sound types from examples of complex auditory scenes, where there can be considerable spectral and temporal overlap between sound sources (natural auditory systems manage to learn different types of sounds without being exposed to perfectly isolated sources), and ii. enforcing a sound separation system to isolate individual sound sources in a complex auditory scene with the guidance of a sound identification system (natural auditory systems segregate auditory scenes into *identifiable* auditory objects).

The core component of this new approach is a multi-task optimization framework combining an audio event classification objective with a separation-specific objective that enforces the separated sources to sum up to the mixture. The proposed framework is flexible with respect to network architectures used as classification and separation systems. The remainder of this chapter:

- Describes the new joint separation-classification approach to audio source separation.
- Provides a detailed description of the framework components, including objective functions used for labels of different strength levels, as well as different network architectures and training strategies.
- Presents the results of benchmarking the algorithm on synthetic mixtures of overlapping events created from a database of sounds recorded in urban environments.

3.2. Background

Audio source separation aims to isolate individual sound sources in a complex auditory scene. This process plays an essential role in a variety of applications, including speech recognition in noisy environments [14], speaker identification in a multi-speaker scenario [11], and music remixing [124].

As discussed in Chapter 2, mask inference is a common approach to solving the under-determined source separation problem, in which the number of audio sources exceeds the number of recorded channels [61][119][114]. It should be noted that the main focus of this chapter is the development of a new framework for separation training that is flexible with respect to the input audio representation and the separation system architecture. Thus, in order to design better controlled experiments, a most commonly used representation domain, i.e., time-frequency (TF), and a common transform will be adopted here. The use of new representations with the proposed training approach will be discussed in the next chapter. In a TF-domain masking approach, a raw audio mixture is first transformed into an intermediary representation, e.g., the STFT. Each source is then estimated by applying a weighting function with values typically in $[0, 1]$, referred to as a *mask*, to the mixture in the transform domain and then converted back to the time domain via an inverse transform, e.g., the Inverse Short-time Fourier Transform (ISTFT).

Figure 3.1 presents an example audio mixture along with the TF masks corresponding to different audio sources in the mixture. The top panel shows the magnitude STFT of the mixture, which is composed of five sources: street noise, music, speech, dog bark, and fire truck siren. The ideal binary masks are illustrated in the bottom panel. An ideal binary mask corresponding to a source assigns a 1 to every TF bin where the source dominates all the other sources, and assigns 0's to the remaining TF bins.

Supervised mask inference methods, especially those using deep neural networks, have gained much popularity over the past decade, due to their successful performance in a variety of denoising and source separation tasks including speech enhancement [57][122][121][18][115], speech separation [32][36][43][118][59], music separation [60][107][92][102][49], and sound effect separation [39]. A major challenge faced by supervised masking-based separation approaches is that they typically require a large dataset of isolated sound sources to generate target time-frequency masks used in model training. Obtaining the isolated sources that compose a mixture may be expensive, require complicated recording setups, or necessitate the creation of synthetic mixtures that lack a certain amount of realism. In some scenarios, it may not even be possible to record sounds in isolation, e.g., recording a bird song in the woods or recording the sound of a machine part

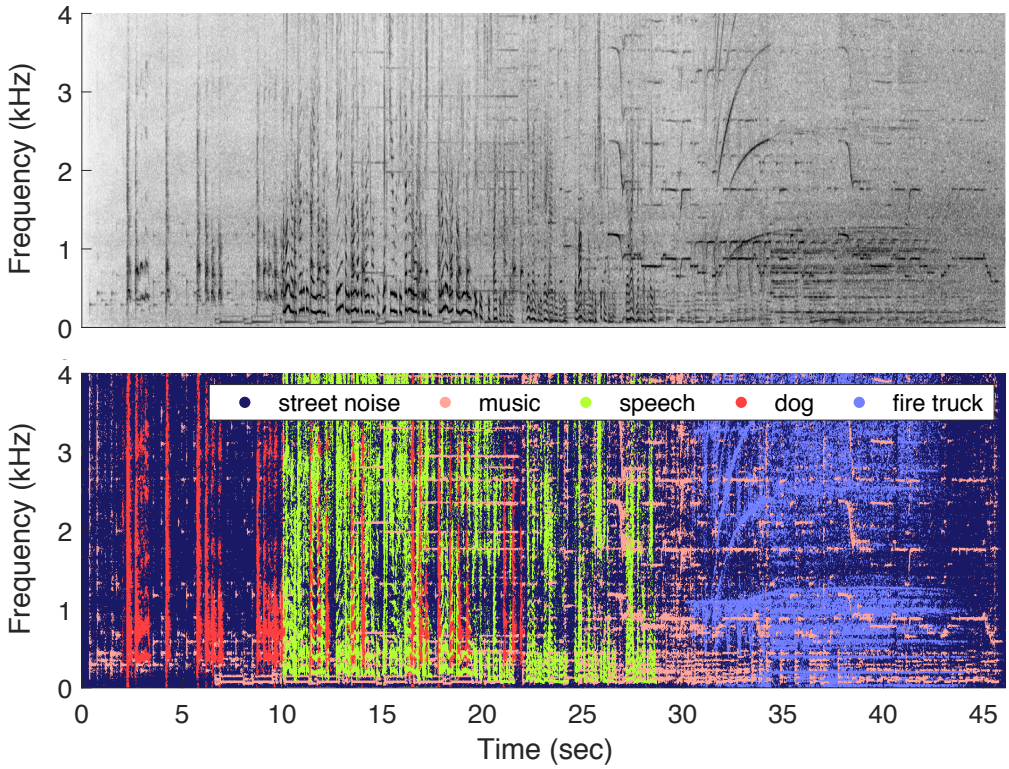

Figure 3.1. The time-frequency representation (magnitude STFT) of a complex auditory scene (top). There are five sources in this scene: street noise, music, speech, dog bark, and fire truck siren. Darker colors mean higher magnitudes. The bottom plot shows the ideal binary masks. An ideal binary mask assigns 1's to time-frequency bins where the associated source dominates all other sources and 0's to the rest of time-frequency bins. Each color corresponds to one audio source.

that only occurs when a machine is running. In cases where isolated sources are not available for training the separation system, it is also unrealistic for humans to use signal processing tools to manually label the audio at the granularity level of TF bins, especially to do so accurately and at scale (imagine creating the binary masks in the bottom plot of Figure 3.1 manually!).

Natural audio-processing systems (e.g, the mammalian auditory system) on the other hand, do not require isolated sources in order to learn to analyze the auditory scene. Humans hardly ever hear sounds in perfect isolation, but still can learn to identify different types of sounds and to focus on them if necessary (segregate a single audio source from the rest of the auditory scene). Based on the capability of natural auditory systems to extract the characteristics of individual audio sources from everyday complex auditory scenes, one can argue that the knowledge about the presence of sounds in a mixture recording could be sufficient information for training a separation system. A separation system that can learn from audio

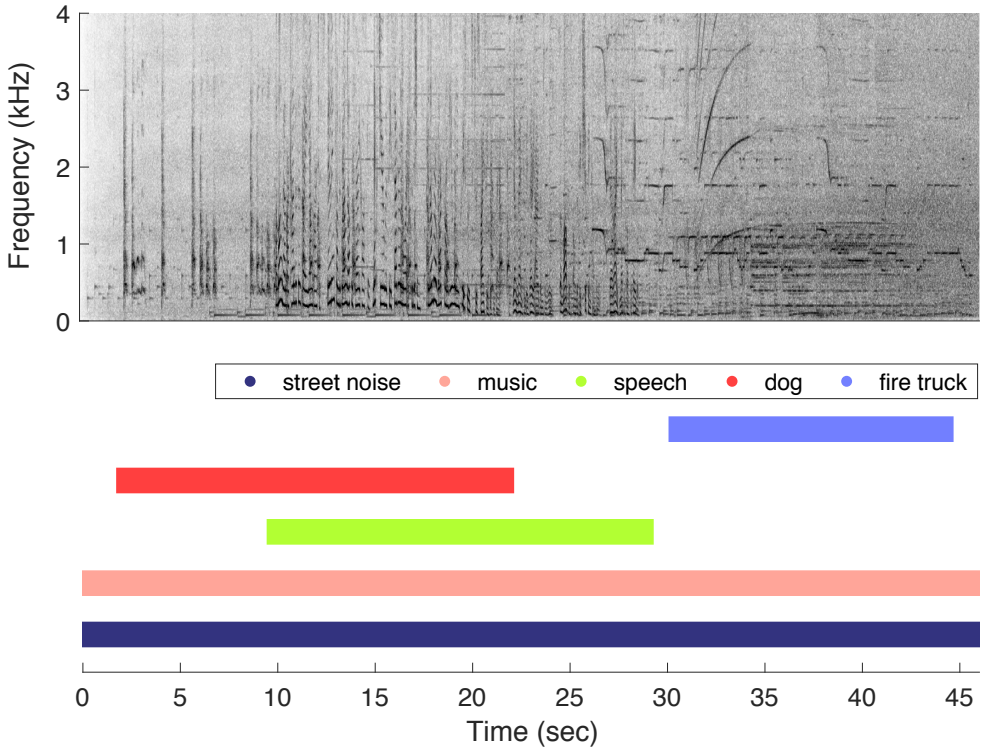

Figure 3.2. The time-frequency representation (magnitude STFT) of a complex auditory scene (top). There are five sources in this scene: street noise, music, speech, dog bark, and fire truck siren. Darker colors mean higher magnitudes. The bottom plot shows the *frame-level* sound labels, which indicate the onset and duration of sounds in the mixture (but do not provide any information about the characteristics, e.g., spectral content of sources). Each color corresponds to one audio source.

mixtures relying only on the information about the present sound types would make the dataset generation much easier, since making recordings of complex auditory scenes (mixture recording) is a much simpler and less expensive task than recording single sound sources in perfect isolation.

Furthermore, it is reasonable to assume that humans can produce limited labels for the activity of sound sources within some time range, and thus, any mixture recording along with such annotations can be used for training. Even non-expert users have successfully provided labels for musical instrument detection [33] and Sound Event Detection (SED) [8], where the labels consisted of the type of audio events as well as the precise time of their occurrence in a given recording. An example of such labels for the audio mixture of Figure 3.1 is presented in the bottom panel of Figure 3.2. Each horizontal bar marks the onset and duration of the activity of one sound type in the mixture. For instance, the green bar indicates that the speech signal becomes active 10 seconds into the mixture recording and stays on for around 20 seconds. The annotation burden can be further reduced, as has been considered in the SED task, by replacing the fine resolution

labels on the precise sound event onsets and offsets (e.g., on the order of 10 ms) by a coarse temporal label indicating the presence or absence of a sound event within a particular audio clip (e.g., on the order of 10 s). Since the fine resolution labels are typically defined at the level of an STFT frame, they are hereafter referred to as frame-level labels, while the coarse labels are referred to as clip-level labels.

In this chapter, my goal is to develop deep-learning-based separation methods, which are trained using weak (frame-level or clip-level) labels, as opposed to TF-bin-level labels typically used in fully supervised setups. It is important to note that since the separation system still outputs estimated sources in the TF domain, this training method requires a system that maps the source energies in the mixture/source representations to sound activities over time. The transition from strong to weak labels can be performed similarly to what has been done in the context of SED.

The SED task is particularly important in this work since: i) I try to address similar challenges in transitioning from strong to weak labels, and ii) I employ an SED mechanism as the critic for separation performance. It should be taken into account that there is an important difference regarding the notion of strength of a label in the context of SED and separation. In SED, the goal is to estimate the type of an audio event together with its precise onset and offset, with the corresponding ground truth referred to as a strong label. In contrast, ground truth limited to presence or absence of a sound within a coarser time range is referred to as a weak label. The interested reader is referred to the description of a weakly labeled SED task in the DCASE 2017 challenge [68]. In the context of source separation, complete ground truth consists of each source's isolated signal, which amounts to having information on each source at the granularity of a TF bin. Strong labels for SED are thus only weak labels for source separation.

To the best of my knowledge, no deep-learning-based source separation system has so far been presented that can be trained under the assumption of sole availability of frame-level labels (let alone clip-level ones) and is able to separate mixtures at test time without side information. **The algorithm proposed in this dissertation allows training a separation system on frame-level and clip-level labels and does not require any side information at the inference time. The important contribution of this approach is that it alleviates the need for isolated sources in separation training and consequently renders the data collection task far easier and less expensive.**

Weakly labeled SED approaches typically leverage multiple instance learning, where an instance-level (i.e., fine temporal resolution) predictor is trained by aggregating or pooling the instance-level predictions to

match the labels at the "bag" level (i.e., a chunk of audio on the order of several seconds, and its associated coarse ground truth label). I would like to use a similar concept for source separation, where the instance-level prediction is now at the level of the TF bin, and the bag level is either that of a frame or that of a whole clip. A different approach to pooling is however needed in SED and source separation systems. In weakly-labeled SED, consecutive time frames will often share the same class labels. In weakly-labeled separation, on the other hand, the structure is much more intricate, as frequency bins sharing the same label may be far from each other, often harmonically spaced in a highly variable manner even among the same types of sounds.

To overcome these difficulties in pooling over the frequency and time dimensions, I propose a form of discriminative pooling, where an SED classifier is employed as the principal metric for loss calculation when training the separator. When applied to a separated source, the classifier is expected to detect that only a single class is present, while all other sources are inactive.

Furthermore, I propose a multi-task learning approach in training the separator, combining the audio event classification objective with an additional separation-specific objective that enforces the separated sources to sum up to the mixture. The model learns to separate based solely on weak labels, either at the frame level or at the clip level. Clip-level labels are equivalents of SED weak labels, which do not require the sound to be active throughout the entire time period for which the label applies. In this chapter, I investigate the contribution of the classification and separation objective function terms to the quality of learned masks, as well as the correlation between classifier and separator performance. I also explore different training strategies, where the classifier and separator are trained jointly, or the classifier is trained first, and then its weights are fixed or fine-tuned while training the separator. Empirical comparison of weakly-labeled separation performance to the strongly-labeled case (when isolated sources are available) is carried out using synthetic mixtures created from the *UrbanSound8K* [89] dataset.

3.3. Related work

As previously mentioned, there has been a resurgence in multiple instance learning approaches for audio following the DCASE 2017 challenge [68], where several studies examine deep network architectures [48][105][125] and/or pooling functions [67][116][42] for the weakly-labeled SED task. There have also been

several applications of multiple instance learning for music, including detecting instruments in mixtures [54], applying artist-/album-level labels to individual tracks [65], and saliency-based singing voice detection [90].

Deep learning based techniques are currently dominant for fully supervised source separation, and typically trained to separate a single source class of interest, such as vocals or a particular instrument type from music mixtures [108][107], or speech from noise [18][116]. An alternative class of techniques such as deep clustering [32] and permutation invariant training [32][43] is required when the source types to be separated are very similar, e.g., separating speech from speech. The fully supervised approaches most relevant to the current study are those that train a single network to separate multiple classes of musical instruments [49][92][99].

Semi-supervised separation methods based on generative adversarial learning were proposed in [101, 127]. The key assumption of these methods is that estimated sources produced by an optimal separator should be indistinguishable from real sound sources, i.e., they should be samples drawn from the same distribution. These adversarial approaches are semi-supervised, since one-to-one correspondence between the mixtures and the real isolated sources used for training is not required. Nevertheless, their training is indeed dependent on the existence of some dataset of isolated sources. However, the need for isolated data can be relaxed when separating a single type of source while only observing isolated background and the target source in the background [70, 104]. Another class of source separation techniques based on weak labels assumes the availability of weak labels at both training and inference time, such as the score-informed approach in [19], the variational auto-encoder in [38], and the audio-visual approach in [25], where the video provides (weak) class labels to guide the audio separation. The approach taken in this dissertation can separate multiple source classes, does not require seeing any sources in isolation, and requires only the audio mixture (no labels) during inference.

Another line of research performs source separation implicitly when training SED systems using either Non-negative Matrix Factorization (NMF) [31] or deep networks [45]. The method in [45] is composed of two stages: first, a segmentation mapping is applied to the TF representation of an audio recording to obtain TF segmentation masks, then a classification mapping is applied to the segmentation masks to estimate the presence probability of sound events. The authors suggest that the separation task can be performed as a byproduct of event detection using the learned segmentation masks. However, their objective function is only event detection cross-entropy and does not include any terms modeling the separation problem explicitly,

such as enforcing each separated mask to belong only to a single source, or enforcing estimated sources to sum up to the mixture as in our approach. Furthermore, they test their method only on isolated sources in background noise, whereas the experiments in this work deal with multiple overlapping sound events.

3.4. Join separation-classification approach

In this section, I describe the joint separation-classification approach to audio source separation. I first provide basic definitions for the mask-based single-channel source separation problem and briefly review the fully supervised separation setup. Then, I present my weakly supervised separation model, formulate the objective function, and discuss the training setup in detail.

3.4.1. Mask-based single-channel audio source separation

Throughout this work, I assume a monaural source separation scenario, where only one recording channel of the mixture is available. Extensions to multi-channel scenario can be considered, but are beyond the scope of this dissertation. The observed audio mixture can be formulated as

$$x(t) = \sum_{i=1}^{n} s_i(t), \tag{3.1}$$

where $x(t)$ and $s_i(t)$ respectively denote the mixture signal and the i-th sound source signal in the time domain, and n is the total number of sound sources in the mixture. Note that each *sound source* is assumed to belong to a distinct *sound class* (e.g., musical instrument, human speech, dog bark, etc.), in other words all instances of the same sound class are considered as a single sound source. These two terms will thus be used interchangeably hereafter.

As mentioned earlier, a common approach to solving the under-determined separation problem is to perform masking on the mixture in a domain where there is less overlap between sources than in the time domain. The time-frequency (TF) domain is used for masking in this chapter. The magnitude TF representation (e.g., magnitude STFT) of the mixture is denoted by $X_{\omega,\tau}$, where ω and τ are frequency and time-frame indices, respectively.

The first step in a typical TF-masking-based method is to estimate source magnitudes by performing element-wise multiplication of the mixture magnitude with a set of estimated masking functions. Let $\hat{M}_{i,\omega,\tau}$ denote a TF mask estimate for the i-th source, taking on values in $[0, 1]$, with $\hat{M}_{i,\omega,\tau}$ being ideally very close

or equal to 0 where the source is inactive and very close or equal to 1 where the source is dominant in the mixture. The masking operation can be formulated as

$$\hat{S}_{i,\omega,\tau} = \hat{M}_{i,\omega,\tau} X_{\omega,\tau}, \tag{3.2}$$

where $\hat{S}_{i,\omega,\tau}$ is the estimated magnitude of the i-th source. The estimated source magnitudes are then typically combined with the mixture phase and converted back to the time domain through an inverse transform (e.g., ISTFT).

3.4.2. Fully supervised separation

The supervised mask inference task aims at training a model to generate estimates of the sources present in a given audio mixture via the estimation of masks to be applied to a TF representation of the mixture. In the fully supervised separation scenario, the time-domain signals of the isolated sources, their TF-domain representations, or TF masks built from them (e.g., the ideal binary mask or the ideal ratio mask [18]) are used as targets in model training. Such targets are referred to as "strong labels", as they provide information about sound classes at the TF-bin level.

Various loss functions have been used in fully supervised mask inference, such as mask approximation (MA), magnitude spectrum approximation (MSA), phase spectrum approximation (PSA), and waveform approximation (WA) [18][51]. While masking approaches that include a phase estimation step are more likely to produce higher quality sound estimates, here, I focus on the MSA objective with L^1 norm for simplicity:

$$\begin{aligned} \mathcal{L}_{\text{mi}} &= \sum_{i,\omega,\tau} |\hat{S}_{i,\omega,\tau} - S_{i,\omega,\tau}| \\ &= \sum_{i,\omega,\tau} |X_{\omega,\tau}\hat{M}_{i,\omega,\tau} - S_{i,\omega,\tau}|, \end{aligned} \tag{3.3}$$

where $S_{i,\omega,\tau}$ denotes the magnitude TF representation of the i-th isolated source and $|\cdot|$ indicates the modulus operator.

In the weakly supervised scenario, TF-bin-level labels (target sources or masks) are no longer available. The target labels instead indicate only the sound class presence at the frame or clip level. The next two sections present my approach to training mask inference networks using only frame- or clip-level sound labels.

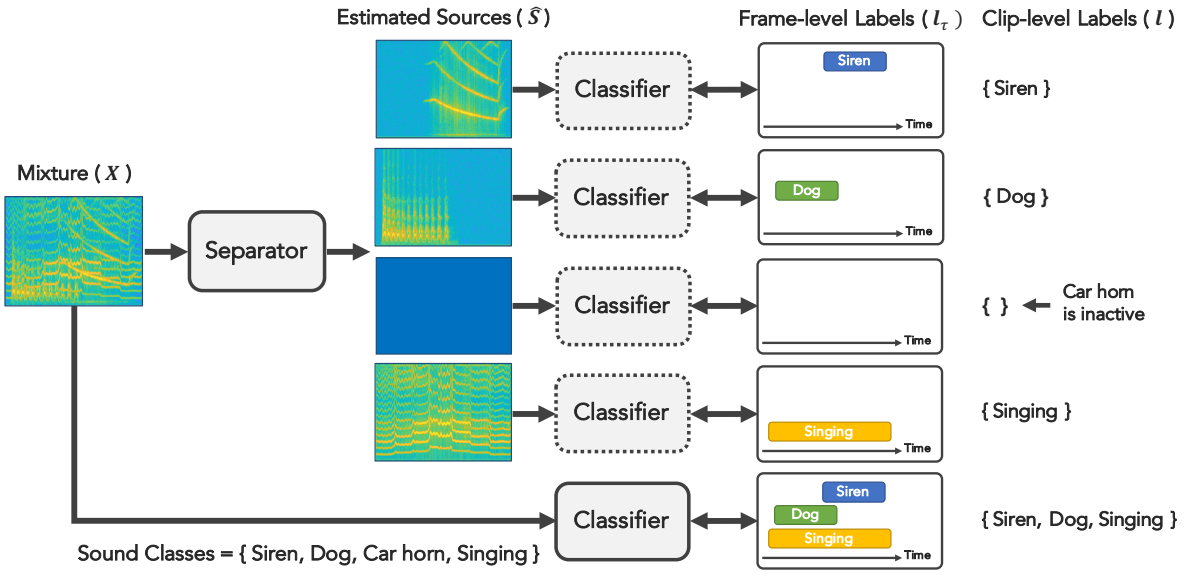

Figure 3.3. The joint separation-classification model. The separator receives an audio mixture and returns source estimates (the blue square is the estimate of an inactive source). The classifier processes separately the mixture and each estimated source (dashed lines indicate shared parameters). When applied to the mixture, the classifier should output the presence probabilities for all classes. The separator is trained such that if any of the source estimates is used as input to the classifier, the classifier output is the presence probability for that source along with zeros for all other sources.

3.4.3. Weakly supervised separation

At a high level, the model is composed of two main blocks: a source separator and an audio event classifier. The block diagram of the entire system is shown in Figure 3.3.

The separator block receives the TF representation (e.g., magnitude STFT) of a mixture and outputs estimates $\hat{\boldsymbol{S}}_i$, $i = 1, \ldots, n$, for each of the sources in the TF domain, where n indicates the total number of sound classes available in a dataset. The number of active sound classes in a given mixture is assumed to range from 1 to n.

The input to the classifier block is also a TF representation. In general, the TF representation used as input to the classifier may be of a different type from the separator output (and the separator input, as it is typically assumed they are in the same domain), as long as it is possible to pass gradients through the transform used to compute it. For instance, the classifier input can be a Mel spectrogram while the separator input and output are a magnitude STFT. Given a TF representation $\boldsymbol{Y}$ as input, the classifier computes frame-level class probabilities $p_{i,\tau}(\boldsymbol{Y})$ for each source class i and time frame τ, representing how likely each

source class is to be active at each time frame in $\boldsymbol{Y}$, or clip-level class probabilities $p_i(\boldsymbol{Y})$ for each source class i, representing how likely each source class is to be active (anywhere) within $\boldsymbol{Y}$.

The frame-level label for the i-th sound source at frame τ is denoted by $l_{i,\tau}$, which indicates whether the source is active at that frame ($l_{i,\tau} = 1$) or not ($l_{i,\tau} = 0$). The clip-level label l_i indicates whether the source i is present anywhere in the clip, i.e., $l_i = 1$ would correspond to the case where $l_{i,\tau} = 1$ for at least one time-frame τ, and $l_i = 0$, otherwise, assuming that frame-level labels were available. Note that $l_{i,\tau}$ may be regarded as the output of a pooling operation across frequency applied to the TF-level labels for the i-th isolated source at frame τ, while l_i would be further pooled across time.

The main idea in this work is that, if one can train a classifier that performs well in predicting source class activities on natural mixtures, where sound classes may sometimes occur in isolation and other times overlap with other classes, that classifier can be used as a critic of the separator's performance, assessing how well the separator is able to separate each source. Weak labels, either at the frame or clip level can thus be used to train the separator through the classifier. For instance, if source i is active at frame τ, passing the estimated source $\hat{\boldsymbol{S}}_i$ as input to the classifier should result in classification outputs such that $p_{i,\tau}(\hat{\boldsymbol{S}}_i) = 1$ and $p_{j,\tau}(\hat{\boldsymbol{S}}_i) = 0$ for all $j \neq i$, because all other sources should be removed from $\hat{\boldsymbol{S}}_i$. This is illustrated in Figure 3.3, where both frame-level labels, with onsets and offsets for each sound class, and clip-level labels where only presence or absence within a clip is indicated are shown.

The classifier can be trained independently or jointly with the separator. The separator, on the other hand, requires the classification results while training, since TF-bin labels are not available and the classifier is required to pool over the TF-bin-level source activity predictions to make predictions at each time frame or for the whole clip. Training the classifier together with the separator could potentially lead to coadaptation of the two systems. That is, the classifier could rely only on a few frequency bins that are most discriminative between sound classes and not penalize the separator's mistakes over the entire frequency range of estimated sources. To investigate the impact of coadaptation between the separator and classifier on the separation performance, three training strategies will be considered in the experiments (see Section 3.5): i) training the separator and classifier jointly from scratch, ii) training the separator through a pre-trained classifier while the classifier is being fine-tuned, and iii) training the separator through a pre-trained and fixed classifier. It

should be noted that the classifier is pre-trained only on mixtures, not on isolated sources, as the latter case would violate the assumption that strong labels are unavailable.

3.4.4. Weakly supervised objective function

3.4.4.1. Mixture loss. The principal goal in training the model is to achieve high quality separation, which requires explicit optimization of mask estimates, even if ground truth TF-bin-level labels are not available. To this end, a key constraint is to enforce the output signals of the separator to add up to explain the input mixture. Indeed, this constraint is critical in preventing the separator from producing masks that solely focus on the most discriminating time-frequency components for classification without fully reconstructing the entire source. The constraint can be enforced through a mixture loss term in the objective function that minimizes the discrepancy between the mixture and the sum of estimated source spectrograms, or between the mixture magnitude and the sum of estimated source magnitudes, assuming that all source estimates are obtained using the mixture phase. A vanilla version of such a term can be formulated using an L^1 loss as

$$\mathcal{L}_{\text{mix,vanilla}} = \sum_{\omega,\tau} |X_{\omega,\tau} - \sum_{i=1}^{n} \hat{S}_{i,\omega,\tau}|. \tag{3.4}$$

Thanks to the information provided by the weak labels, it can in fact further enforced that only the sum over active sources should be equal to the mixture, and all inactive sources should be silent. In the frame-level case, the vanilla loss term in (3.4) can therefore be replaced by a more explicitly constrained version defined in two parts:

$$\mathcal{L}_{\text{f-mix}} = \sum_{\omega,\tau} |X_{\omega,\tau} - \sum_{i\in\mathcal{A}_\tau} \hat{S}_{i,\omega,\tau}| + \sum_{\omega,\tau} \sum_{i\notin\mathcal{A}_\tau} |\hat{S}_{i,\omega,\tau}|, \tag{3.5}$$

where $\mathcal{A}_\tau$ is the set of active source indices in time frame τ. Moreover, given the weak labels, mixture frames where no sources are active can be located and excluded entirely from loss computation. $\mathcal{L}_{\text{f-mix}}$ is referred to as the frame-level mixture loss. In the clip-level case, there is only information available regarding the set $\mathcal{A}$ of active source indices for the whole clip, which can be used to similarly modify (3.4) as:

$$\mathcal{L}_{\text{c-mix}} = \sum_{\omega,\tau} |X_{\omega,\tau} - \sum_{i\in\mathcal{A}} \hat{S}_{i,\omega,\tau}| + \sum_{\omega,\tau} \sum_{i\notin\mathcal{A}} |\hat{S}_{i,\omega,\tau}|, \tag{3.6}$$

which is referred to as the clip-level mixture loss (see Figure 3.3). In the experiments, these refinements to the vanilla mixture loss in (3.4) proved very important for obtaining good mask estimates.

3.4.4.2. Frame-level loss. The sound classes identified by the classifier should match the correct labels, whether it is applied to the input mixture or any of the sources estimated by the separator. This can be achieved by including a binary cross-entropy term between the classifier output and the corresponding true labels. Let $H(l, p)$ denote the binary cross-entropy function defined as

$$H(l, p) = -l \log(p) - (1 - l) \log(1 - p), \tag{3.7}$$

where $l \in [0, 1]$ and $p \in [0, 1]$ respectively denote the true and estimated class probabilities. $\mathcal{L}_{\text{f-class}}(\boldsymbol{Y}, \tau)$ denotes the frame-level classification loss at frame τ for an input spectrogram $\boldsymbol{Y}$ and its associated frame-level weak labels (where labels are left implicit for simplicity of notation). This loss is computed on the mixture $\boldsymbol{X}$ and on each separated source $\hat{\boldsymbol{S}}_i$. For the mixture $\boldsymbol{X}$, the classification loss at frame τ can be classically computed as the sum of binary cross-entropy terms over all sources,

$$\mathcal{L}_{\text{f-class}}(\boldsymbol{X}, \tau) = \sum_{i=1}^{n} H(l_{i,\tau}, p_{i,\tau}(\boldsymbol{X})), \tag{3.8}$$

where $l_{i,\tau} \in \{0, 1\}$ is the true frame-level label for the i-th source at frame τ. For the i-th estimated source $\hat{\boldsymbol{S}}_i$, the associated labels at each frame τ are obtained from the labels for mixture $\boldsymbol{X}$ by keeping only the label $l_{i,\tau}$ for the i-th source, whose activity should be the same as in $\boldsymbol{X}$, and replacing the labels for all other sources with zeros, as they should now be inactive. The loss is thus computed as:

$$\mathcal{L}_{\text{f-class}}(\hat{\boldsymbol{S}}_i, \tau) = H(l_{i,\tau}, p_{i,\tau}(\hat{\boldsymbol{S}}_i)) + \sum_{j \neq i} H(0, p_{j,\tau}(\hat{\boldsymbol{S}}_i)). \tag{3.9}$$

The total frame-level classification loss $\mathcal{L}^{\text{total}}_{\text{f-class}}(\tau)$ at frame τ, where the classifier is applied to the mixture and all the estimated sources, is computed as

$$\mathcal{L}^{\text{total}}_{\text{f-class}}(\tau) = \mathcal{L}_{\text{f-class}}(\boldsymbol{X}, \tau) + \sum_{i=1}^{n} \mathcal{L}_{\text{f-class}}(\hat{\boldsymbol{S}}_i, \tau). \tag{3.10}$$

Combining the mixture loss and the classification loss, the overall frame-level loss function to be minimized can be written as

$$\mathcal{L}_{\text{f}} = \sum_{\tau} \mathcal{L}^{\text{total}}_{\text{f-class}}(\tau) + \alpha \mathcal{L}_{\text{f}-\text{mix}}, \tag{3.11}$$

where $\alpha \geq 0$ is a tunable parameter controlling the contribution of the mixture loss to the total loss.

3.4.4.3. Clip-level loss. When only clip-level weak labels are available, the classifier output is assumed to be a single prediction at the clip level. For example, in this work, a time-pooling operation is applied to the output of the frame classifier to map frame labels to clip labels as is commonly done in the weakly-labeled SED literature [67][116] (see Section **??**). The classification loss given the clip-level labels is formulated as

$$\mathcal{L}^{\text{total}}_{\text{c-class}} = \mathcal{L}_{\text{c-class}}(\boldsymbol{X}) + \sum_{i=1}^{n} \mathcal{L}_{\text{c-class}}(\hat{\boldsymbol{S}}_i), \tag{3.12}$$

with

$$\mathcal{L}_{\text{c-class}}(\boldsymbol{X}) = \sum_{i=1}^{n} H(l_i, p_i(\boldsymbol{X})), \tag{3.13}$$

$$\mathcal{L}_{\text{c-class}}(\hat{\boldsymbol{S}}_i) = H(l_i, p_i(\hat{\boldsymbol{S}}_i)) + \sum_{j \neq i} H(0, p_j(\hat{\boldsymbol{S}}_i)), \tag{3.14}$$

where l_i denotes the clip-level label for the i-th sound class and p_i is the clip-level class probability output by the classifier for the i-th class. Finally, the total loss in the clip-level case is computed as

$$\mathcal{L}_{\text{c}} = \mathcal{L}^{\text{total}}_{\text{c-class}} + \alpha \mathcal{L}_{\text{c-mix}}. \tag{3.15}$$

Table 3.1 summarizes the loss functions used with strong labels, frame-level weak labels, and clip-level weak labels. In the present work, I only consider systems trained exclusively with one of the three label types, i.e., I leave the study of combining available training data with different label-types as a topic of future work. It should be noted that, while the different loss functions in Table 3.1 are interrelated, the decisions they make in separating and classifying sources are not necessarily consistent with each other. For example, a clip-level classifier using max-pooling to combine frame-level decisions could make the right decision at the clip level even though all frame-level decisions are incorrect (i.e., the correct maximum score occurs at a frame that does not have the correct frame-level label).

Table 3.1. Summary of fully-supervised and weakly-supervised loss functions.

Label type	Equation	Loss function
Strong	(3.3)	$\mathcal{L}_{\text{mi}} = \sum_{i,\omega,\tau} \lvert X_{\omega,\tau}\hat{M}_{i,\omega,\tau} - S_{i,\omega,\tau}\rvert$
Frame	(3.5)	$\mathcal{L}_{\text{f-mix}} = \sum_{\omega,\tau} \lvert X_{\omega,\tau} - \sum_{i\in\mathcal{A}_\tau} \hat{S}_{i,\omega,\tau}\rvert + \sum_{\omega,\tau}\sum_{i\notin\mathcal{A}_\tau} \lvert\hat{S}_{i,\omega,\tau}\rvert$
	(3.10)	$\mathcal{L}^{\text{total}}_{\text{f-class}}(\tau) = \sum_i H(l_{i,\tau}, p_{i,\tau}(\boldsymbol{X})) + \sum_i (H(l_{i,\tau}, p_{i,\tau}(\hat{\boldsymbol{S}}_i)) + \sum_{j\neq i} H(0, p_{j,\tau}(\hat{\boldsymbol{S}}_i)))$
	(3.11)	$\mathcal{L}_{\text{f}} = \sum_\tau \mathcal{L}^{\text{total}}_{\text{f-class}}(\tau) + \alpha\mathcal{L}_{\text{f-mix}}$
Clip	(3.6)	$\mathcal{L}_{\text{c-mix}} = \sum_{\omega,\tau} \lvert X_{\omega,\tau} - \sum_{i\in\mathcal{A}} \hat{S}_{i,\omega,\tau}\rvert + \sum_{\omega,\tau}\sum_{i\notin\mathcal{A}} \lvert\hat{S}_{i,\omega,\tau}\rvert$
	(3.12)	$\mathcal{L}^{\text{total}}_{\text{c-class}} = \sum_i H(l_i, p_i(\boldsymbol{X})) + \sum_i (H(l_i, p_i(\hat{\boldsymbol{S}}_i)) + \sum_{j\neq i} H(0, p_j(\hat{\boldsymbol{S}}_i)))$
	(3.15)	$\mathcal{L}_{\text{c}} = \mathcal{L}^{\text{total}}_{\text{c-class}} + \alpha\mathcal{L}_{\text{c-mix}}$

3.4.5. Balancing class weights

In the preceding discussions, all sound sources contribute equally to the total loss value. This is a reasonable setup in cases where all sound sources are equally likely to be active at any given time. However, sound sources may in general occur with very different activity levels in a dataset. For instance, a dataset of urban sounds might include rare, impulsive sound events such as gun shots, as well as sounds that are active over long periods of time such as street music. Therefore, we weight each sound class during training to balance the contribution to the total loss of active and inactive frames for that class, which also equalizes the weight between classes.

Let γ_i denote the probability of the i-th source being active at any given frame. γ_i is computed from the training data as the ratio of the number of frames in the dataset where the i-th source is active to the total number of frames in the dataset. The aim is increasing the contribution of sources occurring less frequently or for very short periods of time (e.g., $\gamma_i = 0.1$) in the total loss, while decreasing the contribution of sources that are active most of the time (e.g., $\gamma_i = 0.9$). This can be achieved by weighting the loss terms corresponding to frames where a source is active by the inverse of the source's prior probability of being active, and similarly for the frames where the source is inactive. The loss weight for the i-th source is defined

as

$$W_{i,\tau} = \begin{cases} \gamma_i^{-1} & i \in \mathcal{A}_\tau, \\ (1-\gamma_i)^{-1} & i \notin \mathcal{A}_\tau, \end{cases} \tag{3.16}$$

where $\mathcal{A}_\tau$ is the set of active source indices in time frame τ. When using such weights in a loss term, the expected number of frames contributing to that loss is not only the same for active and inactive regions of a given source, but also the same across all sources.

These weights can be incorporated in the fully supervised mask inference loss (3.3) as

$$\mathcal{L}_{\text{mi},W} = \sum_{i,\omega,\tau} W_{i,\tau} \left| X_{\omega,\tau} \hat{M}_{i,\omega,\tau} - S_{i,\omega,\tau} \right|. \tag{3.17}$$

The weights can also be incorporated in the case of frame-level weak labels, reformulating the classification loss functions from (3.8) and (3.9) as follows:

$$\mathcal{L}_{\text{f-class},W}(\boldsymbol{X}, \tau) = \sum_{i=1}^{n} W_{i,\tau} H(l_{i,\tau}, p_{i,\tau}(\boldsymbol{X})), \tag{3.18}$$

$$\mathcal{L}_{\text{f-class},W}(\hat{\boldsymbol{S}}_i, \tau) = W_{i,\tau} H(l_{i,\tau}, p_{i,\tau}(\hat{\boldsymbol{S}}_i)) + \sum_{j \neq i} W_{j,\tau} H(0, p_{j,\tau}(\hat{\boldsymbol{S}}_i)). \tag{3.19}$$

The total classification loss $\mathcal{L}^{\text{total}}_{\text{f-class}}(\tau)$ in (3.10) and the overall loss function $\mathcal{L}_{\text{f}}$ in (3.11) are modified accordingly, leading to $\mathcal{L}^{\text{total}}_{\text{f-class},W}(\tau)$ and $\mathcal{L}_{\text{f},W}$.

In the clip-level scenario, the prior knowledge regarding sound class activities at a frame-level granularity is no longer available, but similarly clip-level weights can be used if the sound classes are not uniformly distributed at the clip level. This case in not considered here as the dataset is designed such that the sound classes are distributed uniformly at the clip level in all experiments (all classes are equally likely to be active within a clip).

3.4.6. Network architecture

The architecture of the separator block used in the experiments is depicted in Figure 3.4(a). It is composed of a 3-layer bidirectional long short-term memory (BLSTM) network, with each layer including 600 nodes in each direction. A fully connected layer maps the output of the BLSTM network to n masks with the same size as the input mixture. Activation functions of all BLSTM units are *tanh*, while the dense layer outputs go through *sigmoid* functions, so that the mask values are always in [0,1].

To design a frame-level SED classifier, I explored a number of architectures, ranging from very simple, such as a small stack of fully connected layers, to increasingly more sophisticated ones, such as convolutional recurrent neural networks (CRNNs) [7][2]. I will present the results for the two best performing classifier architectures (RNNs and CRNNs) in Section 3.5. The clip-level SED classifier in this work is a simple extension of the frame-level classifier. It is built by adding a max-pooling operator to the output of the frame-level classifier for each sound class, in order to perform frame-level to clip-level mapping of sound presence probabilities. More specifically, if the prediction of the frame-level classifier at frame τ for sound class i with an input spectrogram $\boldsymbol{Y}$ is denoted by $p_{i,\tau}(\boldsymbol{Y})$, the clip-level classifier prediction $p_i(\boldsymbol{Y})$ for class i and input $\boldsymbol{Y}$ is obtained as

$$p_i(\boldsymbol{Y}) = \max_{\tau} p_{i,\tau}(\boldsymbol{Y}). \tag{3.20}$$

The investigation of separation performance for some of the more advanced temporal pooling operations explored in [67] and [116] is left to future work.

Here, I present the two SED classifier architectures that performed best in the separation training experiments (see Section 3.5):

i) *RNN*: A 2-layer BLSTM network, with each layer including 100 nodes in each direction, followed by a fully connected layer that maps the BLSTM output for every time frame to n class probabilities. Activation functions of all BLSTM units are again *tanh*. Since the classifier is expected to detect the presence of multiple overlapping sound classes independently from one another, its output for each class is mapped to probability values through a *sigmoid* function. Figure 3.4(b) illustrates this architecture.

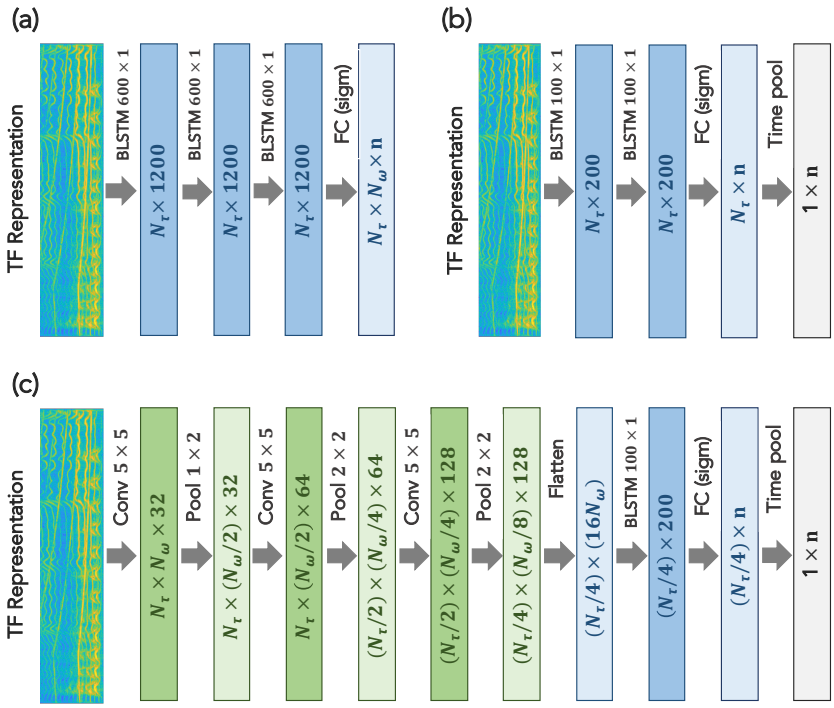

Figure 3.4. Architectures of (a) the separator, (b) the RNN classifier, and (c) the 2D-CRNN classifier. (b) and (c) show the architectures of clip-level classifiers. The frame-level classifiers in both cases can be obtained by removing the last layer (Time pool). N_τ and N_ω denote the number of time frames and frequency bins in the input representation, respectively. n is the total number of sound classes.

ii) *2D-CRNN*: A CRNN architecture composed of a 3-layer 2D convolutional network including max-pooling after each layer, followed by a BLSTM layer and a fully connected layer, which maps the BLSTM output to class probabilities. Activation functions of convolutional, BLSTM, and fully connected layers are *relu*, *tanh*, and *sigmoid*, respectively. The output of each convolutional layer is batch normalized prior to the application of the activation function. Figure 3.4(c) illustrates this architecture in detail. This network is a slightly modified version of the SED model proposed in [116]. Note that the second and third pooling operations in the convolutional network are applied across both frequency and time axes, which results in a downsampled version of frame-level predicted probabilities. To match this coarser time resolution while computing the frame-level loss values, the true weak labels are also downsampled via max-pooling.

3.5. Experiments

In this section, I present the results of audio source separation experiments. The principal research question this set of experiments are designed to answer is whether a separation system can learn to isolate sounds in audio mixtures using only frame-level or clip-level labels (as opposed to TF-bin-level labels). The main results are presented in Section 3.5.3, where the performance of separation systems trained using the proposed weakly supervised method is compared to that of separation systems trained through the fully supervised approach. I further investigate the effect of different components of the proposed optimization framework (loss function terms, training strategies, etc.) on the separation performance in Section 3.5.4.

3.5.1. Sound event dataset

UrbanSound8K[1] [89] is a dataset of 8732 sound excerpts of length ≤ 4 s, taken from field recordings. The dataset contains 10 sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The audio excerpts are labeled based on the sound classes to which they belong as well as their salience in the auditory scene (foreground or background).

Table 3.2. Frame-level prior probabilities of activity γ_i for the five selected sound classes. The probabilities are computed for training datasets with different λ values.

	Sound class				
λ	Car horn	Dog bark	Gun shot	Jackhammer	Siren
5	0.26	0.36	0.27	0.40	0.40
10	0.44	0.57	0.45	0.62	0.63

In this set of experiments, five classes are included in mixture generation: car horn, dog bark, gun shot, jackhammer, and siren. The class selection was made based on two criteria: i) audio examples in one class should contain mostly the sound corresponding to the class label, with a reasonable salience level, and ii) audio examples from different classes should be acoustically distinct enough so that they are at least recognizable as different sounds by human listeners. The air conditioner, children playing, and street music classes do not meet the first criterion, as their examples contain target sounds that are either in the background and barely audible, or accompanied by sounds from other classes. The drilling, jackhammer, and engine idling classes include many examples that sound very similar, thus only one of them was selected.

[1] https://urbansounddataset.weebly.com/urbansound8k.html

Table 3.3. Distribution of frames and clips containing different numbers of sources in training datasets with different λ values.

	Number of sources per frame					
λ	0	1	2	3	4	5
5	0.17	0.28	0.30	0.18	0.06	0.01
10	0.07	0.13	0.21	0.28	0.23	0.08
	Number of sources per clip					
λ	0	1	2	3	4	5
5	0.00	0.06	0.20	0.34	0.30	0.10
10	0.00	0.00	0.02	0.12	0.38	0.48

Audio mixtures in the generated mixture dataset are 4 seconds long and sampled at 16 kHz. Each mixture is composed of at least one *sound event* (i.e., a single occurrence of a sound class) from one of the five selected classes. The total number of sound events per mixture is sampled from a zero-truncated Poisson distribution with an expected value of λ. It is important to note that this number can include multiple sound events from one class, which are grouped together and regarded as one source while generating the weak labels. Thus, the value of λ determines how crowded the auditory scene is. For instance, $\lambda = 10$ means there are on average 10 sound events (from any class) per mixture. For each event, first one of the five classes is selected uniformly at random, and then the actual sound event is sampled from all sounds of that class uniformly as well. Sound events are of arbitrary lengths, ranging from 0.5 s to 4 s, with a start time sampled uniformly at random under the constraint that the event fits entirely in the 4 s clip. The level of each sound event is randomly sampled from a uniform distribution of -30 to -25 loudness units full-scale (LUFS) [27].

UrbanSound8K is distributed with the data split into 10 folds. I use folds 1-6 for creating the training set, folds 7-8 for the validation set, and folds 9-10 for the test set. The training, validation, and test sets include 20K, 5K, and 5K mixtures, respectively. The frame-level prior probabilities of activity γ_i (see Section 3.4.5) for the five sound classes and λ values of 5 and 10 are presented in Table 3.2. Since all classes were sampled uniformly during training, the clip-level prior probabilities of activity are uniformly distributed and thus not reported. To gain an idea of the amount of overlap between sources, I have also computed the distribution of frames and clips containing different numbers of sources in the entire training set. This information is provided in Table 3.3.

3.5.2. Training setup

In all training sessions, the ADAM optimizer was used, with a learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The batch size was set to 10 in all experiments, except in the experiment investigating the effect of a shorter window size (8 ms) where the batch size was set to 8 (see Table 3.10). All networks are trained until the loss on the validation set stops improving for five consecutive epochs, with a maximum of 50 epochs. The separator takes the log-magnitude STFT of a mixture as input using the square root of a Hann window of size 32 ms and a hop size of 8 ms. To provide an upper bound for the separation performance, a separator network was trained as described in Section 3.4.6 on strong labels (i.e., target sources) with the weighted version of the fully supervised mask inference loss function $\mathcal{L}_{\text{mi},W}$ in (3.17).

In the weak label cases, three training strategies were considered: i) training the separator and classifier jointly from scratch using (3.11), ii) pre-training the classifier until convergence using (3.8), then training the separator through the pre-trained classifier while the classifier is being fine-tuned using (3.11), and iii) pre-training the classifier until convergence using (3.8), then training the separator through the pre-trained and fixed classifier using (3.11), where only the term involving the estimated sources contributes to the gradient. The most effective setup used the 2D-CRNN classifier shown in Figure 3.4(c), with linear magnitude STFT features as classifier input, where the classifier was first pre-trained on the mixtures, and then the separator was trained through the fixed pre-trained classifier using a mixture loss weight $\alpha = 100$ in (3.11) and (3.15). I use this as the default setup, and explore the importance of the design choices in Section 3.5.4.

3.5.3. Sound event separation results

The performance of the classifier is evaluated in terms of F-measure $\mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P}+\mathcal{R}}$, the harmonic mean of precision $\mathcal{P} = \frac{\text{TP}}{\text{TP+FP}}$ and recall $\mathcal{R} = \frac{\text{TP}}{\text{TP+FN}}$, where TP, FP, and FN respectively denote the number of true positives, false positives, and false negatives in the classification results. As mentioned earlier in Section 2.8.3, when dealing with source separation algorithms in realistic scenarios (as opposed to performing separation via ideal binary or soft masking), the recently proposed evaluation measure, Scale-invariant Source to Distortion Ratio (SI-SDR) [52][36] has been shown to be more appropriate than the original Source to Distortion Ratio (SDR) [111]. Thus, here I use the SI-SDR to measure the quality of the separated sources. When computing SI-SDR over the test set, I exclude silent sources as well as any mixtures that contain isolated sources, which can happen occasionally for $\lambda = 5$ (see Table 3.3).

Table 3.4. Frame-level sound source classification performance in terms of F-measure. The classifiers are trained and tested on datasets with $\lambda = 5$.

	Sound class				
Classifier	Car horn	Dog bark	Gun shot	Jackhammer	Siren
RNN (mel-40)	0.850	0.813	0.809	0.915	0.811
2D-CRNN (STFT)	0.948	0.870	0.856	0.940	0.876

Tables 3.4 and 3.5 present the average F-measure for frame-level and clip-level sound classification, respectively. The input to the RNN classifier is a magnitude Mel spectrogram with 40 filters and the 2D-CRNN input is a magnitude STFT with linear frequency. It can be observed that, at the frame level, the 2D-CRNN classifier outperforms the RNN classifier by a large margin in identifying all sound sources. The two classifiers perform more similarly at the clip level, with 2D-CRNN working slightly worse than RNN for the jackhammer class, but still better than RNN for all other classes.

Source separation results for strong labels (fully supervised upper bound) and weak labels are shown in Table 3.6 and Figure 3.5, where the weak label results are obtained with the default setup described above, using a pre-trained 2D-CRNN classifier. Table 3.6 presents both means and medians of SI-SDR values, since the former measure is commonly used for reporting separation results in the literature, and the latter is better suited to the non-Gaussian distributions with a large number of outliers, which is particularly the case for the strong label results in Figure 3.5. Since the overall trends between mean and median results in Table 3.6 are similar, for clarity only mean results for the ablation studies in Section 3.5.4 are reported. From these summary statistics, SI-SDR improvements with respect to the mixture is observed for all classes with both frame- and clip-level weak labels. The smallest and largest SI-SDR improvements in Table 3.6 correspond to the siren and gun shot classes, respectively. The siren class in this dataset contains a more diverse set of sounds compared to other classes (e.g., police siren versus ambulance siren), which is likely the reason why it is the most difficult sound type to separate even when strong labels are used.

The scatter plots of separation results, shown in Figure 3.5, allow a more detailed comparison between the performance of separators trained through different types of labels. It should be noted that all test mixtures included in these plots contain at least two sound sources. Each panel shows the amount of SI-SDR improvement versus input SI-SDR for all test set examples of one sound class. The input SI-SDR refers to the SI-SDR obtained when considering the input mixture as the estimate for the target source. One common

Table 3.5. Clip-level sound source classification performance in terms of F-measure. The classifiers are trained and tested on datasets with $\lambda = 5$.

	Sound class				
Classifier	Car horn	Dog bark	Gun shot	Jackhammer	Siren
RNN (mel-40)	0.914	0.915	0.915	0.934	0.864
2D-CRNN (STFT)	0.958	0.924	0.949	0.922	0.914

trend observed in all cases is the downward tilted shape of the data distribution, which is also typically observed in speech separation [36][123]. This pattern indicates that the highest SI-SDR improvement is achieved for low-SI-SDR inputs and the amount of improvement shrinks when using inputs with higher SI-SDR values.

When going from strong to weak labels in all sound classes, an obvious trend is a decrease in the number of points in the higher end of SI-SDR improvement values. For example, in the plot corresponding to the results for the car horn class and strong labels (top row, leftmost panel), there are several points with SI-SDR improvements above 50 dB. When using frame-level labels, the highest SI-SDR improvement drops to around 40 dB, and it decreases even further down to 30 dB when using clip-level labels. Interestingly, however, the high-density regions of the distributions in each class seem to remain largely similar, contrary to what one may have expected given the difference in the strength of labels used for training. Although frame-level labels yield better results than clip-level labels in general, the distribution of output SI-SDRs for these two label types seems to be very similar in all cases. Furthermore, both weak-label distributions seem to have large amounts of overlap with strong-label distributions and to provide significant SI-SDR improvement over the input SI-SDRs.

Table 3.6. Mean/median SI-SDR values (dB) for all sound classes and separators trained using different labels. ΔSI-SDR indicates the SI-SDR improvement. The last column shows the results over all samples and all classes. The 2D-CRNN classifier is used in both weak label cases. Models are trained and tested on datasets with $\lambda = 5$.

	Sound class					
	Car horn	Dog bark	Gun shot	Jackhammer	Siren	Overall
Input SI-SDR	−5.8 / −5.7	−5.4 / −5.4	−5.5 / −5.8	−2.9 / −2.8	−3.0 / −3.0	−4.5 / −4.5
ΔSI-SDR-strong	9.9 / 7.9	10.0 / 9.2	12.5 / 11.2	7.8 / 6.8	4.9 / 6.2	9.0 / 8.3
ΔSI-SDR-frame	7.0 / 5.4	8.3 / 7.7	9.7 / 9.1	5.7 / 4.9	3.1 / 4.3	6.8 / 6.2
ΔSI-SDR-clip	6.5 / 5.7	6.4 / 6.1	8.8 / 8.3	4.6 / 4.0	1.8 / 3.5	5.6 / 5.5

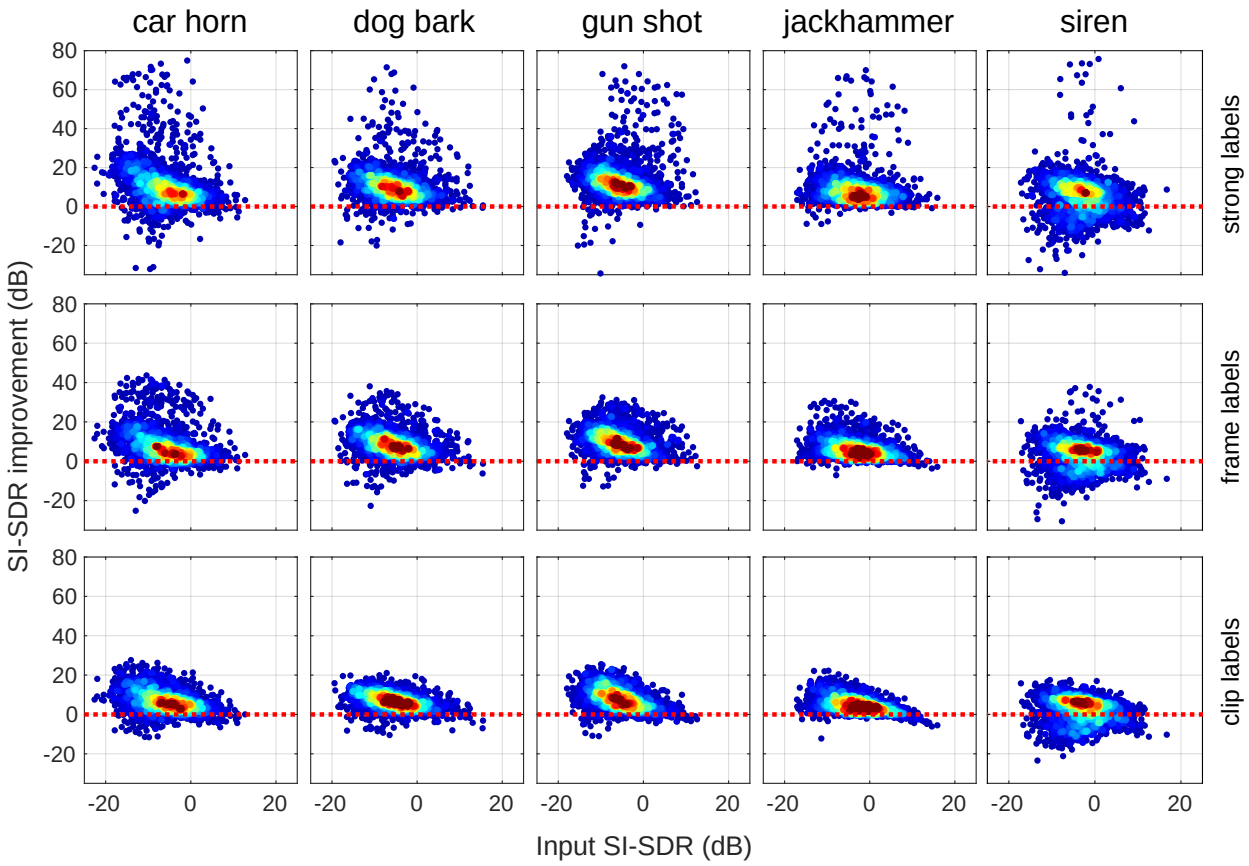

Figure 3.5. Separation results for all sound classes when the separator is trained on strong labels (top row), frame-level weak labels (middle row), and clip-level weak labels (bottom row). All panels show SI-SDR improvement versus input SI-SDR values. The 2D-CRNN classifier and the magnitude STFT input are used in experiments with both frame-level and clip-level labels. There are over 3000 datapoints in each plot (between 3073 for gun shot and 3147 for jackhammer). Warmer colors mean higher densities of data points.

3.5.4. Ablation studies

In this section, I investigate the effect of different components of the proposed weakly supervised framework on the separation performance. I focus on the core components of the proposed algorithm, i.e., the loss function and training setup in Sections 3.5.4.2, 3.5.4.5 and 3.5.4.1. Furthermore, I study the robustness of the separation system with respect to the amount of overlap between sources (Section 3.5.4.4) and the impact of the audio representation parameters on the separation results (Section 3.5.4.3).

3.5.4.1. Training strategies. The effect of different training strategies on separation performance can be observed in Table 3.7. The joint separation-classification model was trained on frame-level weak labels under the three training strategies listed in Section 3.4.3. Regardless of the classifier architecture, the best

separation results are achieved when the classifier is pre-trained on mixtures and its parameters are then fixed when training the separator. Training the separator and classifier jointly from scratch, or fine-tuning the classifier when the separator is being trained always resulted in a worse separation performance in the experiments. One hypothesis is that this behavior is due to the co-adaptation of the two networks, where the classifier can adapt its weights to correctly classify errors made by the separator, rather than forcing the separator to output estimated sources that match the previously learned representation for each sound class. In other words, this co-adaptation weakens the ability of the classifier to objectively assess the performance of the separator.

Table 3.7. Mean SI-SDR improvement (dB) for different training strategies, over all classes. The models are trained on frame-level labels. In all cases, $\alpha = 100$, $\lambda = 5$, and the average input SI-SDR is -4.5 dB.

	Training strategy		
Classifier	Joint	Fine-tune classifier	Fix classifier
RNN (mel-40)	-4.4	5.5	**6.2**
2D-CRNN (STFT)	-0.2	1.3	**6.8**

3.5.4.2. Mixture loss. To investigate the effect of the mixture loss term, the separator network was trained using different α values in the overall frame-level loss of (3.11). The SI-SDR improvement results, presented in Table 3.8, clearly show the importance of this loss term for the separation task. A similar trend is observed for both classifiers. When $\alpha = 0$, only the classification loss is used to train the separator, which leads to poor separation performance as the separator network only needs to isolate the TF features necessary for classification, not signal reconstruction. Conversely, a comparatively very low contribution of the classification loss term (e.g., $\alpha = 10^4$) results in degraded performance as the separator only needs to reconstruct the mixture without isolating the individual sound sources. A good balance between the two loss terms (e.g., $\alpha = 10^2$) is essential to obtain high SI-SDR gains.

3.5.4.3. TF representation. The properties of the audio representation input to the classifier, such as frequency scaling and resolution, proved to have a considerable impact on the separation results in the experiments. The performance of the separator is essentially correlated with the efficacy of the classifier in capturing the spectro-temporal patterns that distinguish each sound class from the others. For instance, a classifier that depends only on a few frequency bins to identify a sound will output accurate class probabilities as long as the separator assigns correct amounts of energy to those bins. Using such a classifier, the model

Table 3.8. Mean SI-SDR improvement (dB) using different mixture loss weights, over all classes. The models are trained on frame-level labels. In all cases, $\lambda = 5$ and the average input SI-SDR is -4.5 dB.

	Mixture loss weight (α)				
Classifier	0	10	10^2	10^3	10^4
RNN (mel-40)	1.9	4.6	**6.2**	5.3	1.9
2D-CRNN (STFT)	0.9	3.9	**6.8**	5.1	1.1

could correctly identify an impulsive, broadband sound (e.g., gun shot) in a mixture, even if the separated source estimate includes only a small portion of the actual spectral content.

Table 3.9. Mean SI-SDR improvement (dB) using different frequency scales and resolutions, over all classes. The models are trained on frame-level labels. In all cases, $\alpha = 100$, $\lambda = 5$ and the average input SI-SDR is -4.5 dB.

	Number of mel filters					
Classifier	10	20	30	40	56	Linear freq.
RNN	5.5	**6.4**	6.0	6.2	5.0	6.1
2D-CRNN	5.3	4.8	5.9	6.0	6.2	**6.8**

One way to address this problem is to force the classifier to produce predictions based on broader frequency ranges by decreasing the frequency resolution of mixtures and estimated sources prior to feeding them to the classifier. To lower the frequency resolution, a Mel-frequency filterbank is applied to the magnitude STFTs to be used as classifier inputs. The Mel-frequency filterbank also has the advantage of changing the frequency resolution logarithmically, with a grid that is finer across lower frequencies, maintaining most of the information necessary to distinguish harmonic sources, and grows coarser as the frequency increases. I investigate the effect of frequency scaling and resolution on the quality of spectral patterns learned by the classifier, which in turn impacts the separation quality, by using two different representations as the classifier input: a linear magnitude STFT and a linear magnitude Mel spectrogram with a varying number of Mel filters. The STFT parameters (window size and hop length) are the same for the separator and classifier inputs.

The amount of SI-SDR improvement for different Mel-frequency filterbanks (featuring different numbers of filters and different center frequencies) are provided in Table 3.9. The results for the linear magnitude STFT, corresponding to the linear frequency case (no filterbank), are reported as "Linear Freq." in the table.

As can be seen, changing the frequency scale and resolution of the classifier input can make a difference of up to 2 dB in the average SI-SDR improvement. The performance of a model using the RNN classifier can be improved up to 0.4 dB by using a Mel spectrogram as input. The best number of Mel filters, however, seems to be difficult to choose without running a grid search. The 2D-CRNN classifier, on the other hand, provides the highest improvement when the original magnitude STFT is used as input. I will investigate the effect of using a multi-resolution time-frequency representation, e.g., CQT, and the proposed audio representation, i.e., MCFT on the separation performance in the next chapter.

Table 3.10. Mean SI-SDR improvement (dB) for different STFT window sizes, over all classes. In all cases, the 2D-CRNN classifier is used with magnitude STFT input features. The models are trained on frame-level labels. In all cases, $\alpha = 100$, $\lambda = 5$, the overlap between windows is 75%, and the average input SI-SDR is -4.5 dB.

	Sound class				
Win. size	Car horn	Dog bark	Gun shot	Jackhammer	Siren
8 ms	5.8	6.8	10.0	4.9	3.0
16 ms	**7.8**	**8.7**	**10.4**	**6.1**	**4.1**
32 ms	7.0	8.3	9.7	5.7	3.2
64 ms	5.1	5.8	7.0	4.1	2.8

It can be observed that the 2D-CRNN classifier consistently outperforms the RNN classifier in terms of separation performance in Tables 3.7-3.9. Since the 2D-CRNN classifier also provided the best classification performance for most classes in Tables 3.4 and 3.5, these results imply that better classification performance is correlated with better separation performance when training a weakly labeled separation system. Further refinements of the classifier may thus lead to improved separation quality.

I further investigate the effect of frequency and time resolutions by varying the window size of the magnitude STFTs input to both separator and classifier blocks. I run this experiment using the best performing model thus far, which includes a 2D-CRNN classifier with magnitude STFT (linear frequency scale) as input. The results, provided in Table 3.10, demonstrate that decreasing the time resolution by using windows longer than 32 ms degrades the separation performance, while decreasing the window size provides performance improvement. The limit on the improvement, however, seems to be reached when the window size is 16 ms, as using a window size of 8 ms results in worse performance, likely due to the poor frequency resolution.

3.5.4.4. Source density. Next, I investigate how the density of sound sources in the scene impacts the performance of networks trained with different label types. When strong labels are used, the network is provided with information about the presence of sources at a time-frequency-bin level granularity. However, when using weak labels the ability of the classifier to learn the spectral structure of a sound type, and hence how accurate a metric it is for separation training, depends on how often that sound source overlaps with other sources in training examples. It is, therefore important to have an idea about how much the separation performance drops when using weak labels as the number of sound events in the scene and overlap between them increase.

As mentioned in Section 3.5.1, the parameter λ used when creating the mixtures determines the expected number of events in each four-second scene. The left four columns in Table 3.11 compare separation performance between training using strong labels (fully supervised upper bound) and both frame- and clip-level weak labels for different λ values, where only results with the 2D-CRNN are reported for brevity. In the fully supervised case, training with more difficult mixtures ($\lambda = 10$) leads to improved separation performance compared to training with easier mixtures ($\lambda = 5$). However, for frame-level weak labels, training with $\lambda = 10$ leads to slightly worse performance than training with $\lambda = 5$, and for clip-level weak labels training with $\lambda = 10$ causes a larger performance drop. Revisiting Table 3.3, it can be seen that when $\lambda = 10$ the training set contains no clips with only a single active source, compared to 6% of the clips for $\lambda = 5$, while there are some single source frames for both λ values. Therefore, I hypothesize that the drop in clip-level weak-label performance for the $\lambda = 10$ training set in Table 3.11 is due to the lack of any training data containing labeled regions with isolated sources. While the higher SI-SDR numbers in Table 3.11 for both frame-level and clip-level weak labels using the $\lambda = 5$ training set indicate that the network does likely make use of labeled regions containing isolated sources, the method can still work in their absence, as shown by the SI-SDR improvements of 4.9 dB and 4.4 dB for the most difficult case of clip-level weak-labels and the $\lambda = 10$ training set.

3.5.4.5. Class weights. Finally, I investigate the importance of the loss weights, computed based on the prior probability of sound class activities, in separation training (see Section 3.4.5). By increasing the focus on sources that are short or happen less often and decreasing the focus on sources that are active most of the time, class weights help a separator to maintain a balance in learning sound classes with different levels of activity. It is particularly important to study the effect of class weights on the performance of separation

Table 3.11. Mean SI-SDR improvement (dB) for separators trained using strong labels, frame-level weak labels, clip-level weak labels, datasets with different λ values, and weighted (left four columns) or not weighted (right four columns) loss functions. All separators are trained using the 2D-CRNN classifier.

	With class weights				Without class weights			
Training λ	5		10		5		10	
Testing λ	5	10	5	10	5	10	5	10
Input SI-SDR	−4.5	−6.2	−4.5	−6.2	−4.5	−6.2	−4.5	−6.2
ΔSI-SDR-strong	9.0	7.0	9.4	7.3	3.2	1.8	9.3	7.4
ΔSI-SDR-frame	6.8	5.2	6.4	5.4	6.3	4.9	6.1	5.3
ΔSI-SDR-clip	5.6	4.7	4.9	4.4	5.6	4.7	4.9	4.4

systems trained on weak labels as no information about the spectral structure of sounds is provided in separation training. In this training setup, the loss weights are used in the case of strong labels and frame-level labels, in the formulation of the mask inference loss and the classification loss, respectively. The results presented in Table 3.11 (right four columns) are obtained using the same setup as in the previous section, with the only difference that the loss terms are not weighted. Removing the weights when a less dense training dataset (e.g., $\lambda = 5$) is used results in a larger performance drop in both strong and frame-level cases. Intuitively, this behavior is expected as in such scenarios, the prior probabilities are much smaller than 0.5 for sparser classes, such as gun shot (see Table 3.2), and hence these classes are assigned larger weights than less sparse classes, such as jackhammer. However, as the prior probabilities get closer to 0.5 in a dataset with $\lambda = 10$, the difference between class weights becomes smaller and their effect on separation results less noticeable. Further, it can be observed that removing the weights has a more dramatic impact on the strong label than frame-level label results. This can be explained by considering the fact that in the frame-level label case, the weights are only incorporated in the classification loss. Therefore, as long as the classification performance does not degrade significantly (which was the case in these experiments), the separation performance is anticipated not to vary dramatically.

3.5.5. Unsuccessful attempts

In this section, I provide a list of failed attempts in the experiments so that they are avoided (or tried more carefully) in any future work that might build on the proposed method. In addition to using the RNN and 2D-CRNN sound event classifiers shown in Figures 3.4(b) and 3.4(c) for frequency pooling, I explored simple frequency pooling (e.g., average pooling over frequency), a learned linear transform, or a feedforward deep

network (without memory). In all cases, these frequency pooling approaches failed to learn to separate. Furthermore, I experimented with an "idempotent" loss, where an additional loss function term enforced the separation network to pass estimated separated sources unchanged. I found that this loss term only hurt separation performance. My hypothesis is that in most cases this constraint was redundant with information provided by the weak labels. Finally, I explored using log magnitude STFT features (as opposed to linear magnitude) as input to the classifier. Although log features gave slightly better classification performance, the separator networks did not train reliably, even when regularizing the log (i.e., adding a small positive value to the log input).

3.6. Conclusion

In this chapter, I presented an algorithm for training a source separation system with weak labels, where isolated sources are not required for the training process. The proposed algorithm is auditory-inspired in that: i. it includes having a sound identification system to learn different sound types from examples of complex auditory scenes, where there can be considerable spectral and temporal overlap between different sources, and ii. it employs the trained sound identification system as the principal metric for loss calculation while training the separator. The model is trained to minimize an objective function that requires the separator to produce source estimates that are identifiable by the classifier. The objective function also enforces the estimated sources to sum up to the mixture. The experiments with weak labels yielded promising results and showed significant SI-SDR improvement even when using weak labels on a very coarse-resolution time grid (clip-level labels). Since the proposed separation training method is not constrained by having access to ground truth isolated sources, it is an important step towards the design of systems that can be trained on large collections of audio mixtures capturing more realistic scenarios. Systems trained using weak labels would thus be better-suited to use in wearable hearing devices (e.g., hearing aids), as they are much easier to train (and re-train) on complex auditory scenes encountered in everyday life.

CHAPTER 4

Putting it all together

4.1. Introduction

Dealing with high levels of energy overlap between sources is a major challenge faced by source separation algorithms that use time-frequency representations as their input. The MCFT [77][81], an audio representation whose design was inspired by the *common fate* principle (see Chapter 2), was developed with the principle goal of addressing the separability issue in the time-frequency domain. The MCFT is fully invertible (i.e., a time-domain audio signal can be perfectly reconstructed given its representation in the MCFT domain), and increases the separability for highly overlapped sound sources with different spectro-temporal modulation patterns by encoding the modulation patterns as explicit dimensions. However, the original version of the MCFT is high-dimensional and thus it cannot be easily used in realistic scenarios where low computational complexity is desirable. In this chapter, a subsampling method will be presented that significantly reduces the size of the MCFT while maintaining its invertibility, its capability for capturing spectro-tempral modulations, and to a great extent its higher level of separability compared to a time-frequency representation.

The utility of this new representation, termed L-MCFT, will be demonstrated by training deep neural networks to separate mixtures of highly overlapped audio sources with distinct modulation patterns in the L-MCFT domain. Furthermore, it will be shown that similar to a time-frequency representation such as the STFT, the L-MCFT can be employed as input to a joint separation-classification system that learns to separate sounds using weak (e.g., frame-level) labels as target. In summary, the contributions of this chapter include:

- A low-complexity version of the MCFT, which can be easily used for audio source separation in realistic scenarios.
- A series of experiments on urban sound mixtures demonstrating the utility of the L-MCFT as an audio representation for deep-learning-based source separation.

4.2. MCFT Refinement

In the implementation of the MCFT as proposed in Chapter 2, each two-dimensional filter in the multi-resolution filterbank, regardless of its bandwidth, is applied to all time-frequency points of the Constant-Q Transform (CQT). Such an approach results in oversampling for low-scale/low-rate filters while high-scale/high-rate filters are only critically sampled. Thus, the output representation is large and contains a high level of redundancy making it somehow difficult to use in realistic situations. For instance, a one-second-long audio signal sampled at 44.1 kHz, with a frequency resolution of 48 bins/oct and a frequency range of 20 Hz to 22.05 kHz, results in a CQT of size 486×628. A spectro-temporal filterbank for this signal with scale and rate resolutions of 1 (cyc/oct) and 1 (cyc/sec), respectively, can include up to 154 filters (7 spectral and 22 temporal filters). Keeping all the time-frequency points of the filtered sections, therefore, results in a representation of size $7 \times 22 \times 486 \times 628$, that is 154 times larger than the CQT, where the low-scale/low-rate slices are highly oversampled (contain redundant information).

In order to use the MCFT as input to audio processing algorithms, it is important to reduce the amount of redundant information such that the representation is more memory efficient and imposes less computational complexity on the algorithms. To this end, the original spectro-temporal filterbank can be replaced by a subsampled filterbank, where all filters are critically sampled, or they are all sampled with the rate of the filter with the highest scale-rate center. In the latter approach, lower filters would be slightly oversampled compared to the critically sampled case, but they are still downsampled versions of the original filters. Slight oversampling, as will be shown in Section 4.2.2, can be helpful to the separation performance in non-ideal cases where the signal segment under analysis does not contain exactly full cycles of very low frequency components, and hence critical sampling results in some information loss. It will be demonstrated in Section 4.2.2 that critical sampling and downsampling to the highest bandpass filter can respectively results in at least 95% and 80% reduction in the MCFT size while maintaining significant separability gain over the baseline time-frequency representation, i.e., the CQT. It should be noted that here the term *subsampled filterbank* is used instead of the common term *multi-rate filterbank* merely to avoid confusion with *rate* as one dimension of the MCFT. In the following sections, I discuss the development of a subsampled filterbank, which is subsequently used in the implementation of a subsampled version of the MCFT, hereafter referred to as L-MCFT.

4.2.1. Subsampled filter-bank design

The first step in the design of subsampled version of the MCFT is the design of a spectro-temporal filterbank with a sampling frequency that varies with the filter center and bandwidth. In this section, I will first present the fundamentals of the design and implementation of of a one-dimensional subsampled filterbank (e.g., the filterbank used in the CQT computation [91]), and then I will discuss the generalization of the subsampled filterbank to the two-dimensional (e.g., scale-rate) case.

4.2.1.1. One-dimensional subsampled filter-bank. In their work on the development of an efficient, perfectly reconstructable CQT, Schörkhuber et al. [91] proposed a subsampled filterbank design method, which is carried out in the frequency domain (unlike methods applying windows of different lengths to the signal in the time domain). In this section, I briefly cover their approach to the subsampled filterbank design.

Let us denote the multi-resolution short-term frequency transform of a discrete time-domain signal $x(n)$ by $X_{k,n}$ defined as

$$X_{k,n} = \sum_{l=0}^{L-1} x(l) w_k(n-l) e^{i2\pi(n-l)f_k/f_s}, \tag{4.1}$$

where k and n are frequency bin and time indices, respectively, L denotes the length of x, and $w_k(n)$ is a symmetric window function (with respect to $n = 0$). f_s and f_k respectively denote the sampling frequency of x and the center frequency of the kth frequency bin. In a logarithmic frequency scale (e.g., the frequency scale of the CQT), the center frequencies can be obtained from

$$f_k = f_0 \cdot 2^{\frac{k}{b}}, \qquad k = 0, 1, ..., K-1 \tag{4.2}$$

with b denoting the number of frequency bins per octave, f_0 denoting the lowest center frequency and K the total number of frequency bins. In a constant-Q transform, the ratio of the center frequencies to bandwidths, the Q-factor, is set to be constant. Thus, the support of the time-domain window function, $w_k(n)$, is wider for lower center frequencies and becomes narrower as f_k increases. This is equivalent to the frequency-domain support of window functions being narrower for lower frequencies and wider for higher frequencies. The Q-factor can be computed as

$$Q = 2^{\frac{1}{b}} - 2^{-\frac{1}{b}}. \tag{4.3}$$

Note that equation (4.1) presents a formulation of the transform with a hop size of one sample between adjacent windows, that is, the transform is computed for every $n = 0, 1, ..., L-1$, which is computationally inefficient. Reducing the complexity of the transform can be carried out by subsampling in the time domain, i.e., by evaluating the values for every $n = 0, h_k, 2h_k, ..., \frac{L-1}{h_k}$, where h_k is the hop size between adjacent windows corresponding to the kth frequency bin. This is equivalent to subsampling the transform at the kth bin with a sampling rate of $f_s^k = f_s/h_k$.

In a different approach, fast computation of (4.1) through Fourier transform and reducing the computational complexity can be combined and implemented more efficiently by subsampling the transform in the frequency domain. Considering the fact that the time-domain subsampling is equivalent to mapping the entire frequency range of the spectrum, $\left(-\frac{f_s}{2}, +\frac{f_s}{2}\right]$ into the frequency interval $\left(-\frac{f_s^k}{2}, +\frac{f_s^k}{2}\right]$, frequency-domain subsampling of the kth frequency bin can be performed by shifting down the kth filtered segment to be centered around zero, discarding all the frequency bins outside $\left(-\frac{f_s^k}{2}, +\frac{f_s^k}{2}\right]$, and then applying the inverse Fourier transform to the remaining values. As demonstrated in [91], however, merely shifting the kth center frequency results in a phase difference between the frequency-domain and time-domain subsampled versions (even though the magnitude of the coefficients are the same). Although this phase discrepancy does not affect the signal reconstruction in the frequency-domain approach, the phase values produced by the time-domain approach are more intuitive and thus more desirable. The phase discrepancy can be simply resolved by performing the mapping of the values in the interval $\left(-\frac{f_s}{2}, +\frac{f_s}{2}\right]$ into $\left(-\frac{f_s^k}{2}, +\frac{f_s^k}{2}\right]$ using the following function:

$$M(f, f_s^k) = f - \lfloor \frac{f}{f_s^k} \rfloor f_s^k. \tag{4.4}$$

where f denotes the original frequency value, $M(.,.)$ is the function that maps the original frequency into the value after subsampling, and $\lfloor \cdot \rfloor$ indicates rounding towards negative infinity. For more details, please refer to [91].

4.2.1.2. Subsampled spectro-temporal filter-bank. The frequency-domain subsampling approach described in the previous section can be generalized to the two-dimensional space to reduce the complexity of the spectro-tempral filters, and consequently the complexity of the MCFT.

In this context, the time-frequency domain is the original domain and the representation to be filtered and subsampled is the CQT. The scale-rate domain is the transform domain, where the subsampling is

performed (similar to “frequency" domain in the previous section). The spectro-temporal filterbank contains two-dimensional constant-Q filters whose center frequencies are distributed as

$$(s_p, r_q) = (s_0 \cdot 2^{\frac{p}{b_s}}, r_0 \cdot 2^{\frac{q}{b_r}}), \qquad p = 0, 1, ..., N_s - 1, \quad q = 0, 1, ..., N_r - 1, \tag{4.5}$$

where b_s and b_r denote the number of scale and rate filters per octave, respectively, (s_0, r_0) is the center frequency of the closest filter to the origin, and N_s and N_r respectively denote the total number of scale and rate filters. The subsampled filter centered at (s_p, r_q) is computed by mapping the entire scale-rate domain, $\left(-\frac{\eta_s}{2}, +\frac{\eta_s}{2}\right] \times \left(-\frac{\eta_r}{2}, +\frac{\eta_r}{2}\right]$ into the interval $\left(-\frac{\eta_s^p}{2}, +\frac{\eta_s^p}{2}\right] \times \left(-\frac{\eta_r^q}{2}, +\frac{\eta_r^q}{2}\right]$ using the function

$$M(s, \eta_s^p, r, \eta_r^q) = (s - \lfloor \frac{s}{\eta_s^p} \rfloor \eta_s^p, r - \lfloor \frac{r}{\eta_r^q} \rfloor \eta_r^q), \tag{4.6}$$

where η_s and η_r denote the original sampling rate along the scale and rate axes, respectively, η_s^p is the reduced sampling rate corresponding to the $p$the scale filter and η_r^q is the sampling rate of the qth rate filter.

To compute the L-MCFT coefficients associated to the scale-rate bin centered at (s_p, r_q), first the 2D Fourier transform is applied to the CQT to obtain its scale-rate-domain version. Next, the values in the interval $\left(s_p - \frac{\eta_s^p}{2}, s_p + \frac{\eta_s^p}{2}\right] \times \left(r_q - \frac{\eta_r^q}{2}, r_q + \frac{\eta_r^q}{2}\right]$ are multiplied by the 2D filter values and shifted along the scale and rate axes according to the mapping function in equation (4.6). Finally, the filtered section is converted back to the time-frequency domain by applying the inverse 2D Fourier transform. It is worth considering that the computation of spectro-tempral filters can be performed faster by directly implementing the closed-form Fourier transform of the spectral and temporal seed functions (e.g., in equations (2.6) and (2.7)) as it would save the time of two FFT operations for each filter in the filterbank (e.g., in computing a filterbank with 100 filters this would save the time of 200 FFT operations). The formulas for the spectral and temporal seed functions used in this dissertation are provided in Appendix A.

4.2.2. Experimental Validation

In this section, I compare the L-MCFT to MCFT in terms of separation performance and the overall size of the representation. The goal of this set of experiments is to answer the following questions:

(1) Is there a trade-off between the amount of discarded information (the level of dimensionality reduction) and the separability of the L-MCFT in realistic scenarios (a short signal containing frequency components with different start times and potentially incomplete cycles)? How does the L-MCFT

computed with different subsampling methods performs on the source separation task compared to the MCFT?

(2) Does L-MCFT provide better separability for mixtures of highly overlapped sources with distinct spectro-temporal modulation patterns, compared to a representation that does not explicitly capture spectro-temporal modulation patterns as explicit dimensions such as the CQT?

(3) How much parameter tuning is required to improve the separation performance of the L-MCFT (e.g., compared to the CFT)?

The dataset used in Section 2.8 to benchmark the MCFT against other representations is also used in this set of experiments. As a reminder, the dataset includes 126 mixtures of instrumental sounds played in unison (same pitch) but with different frequency modulation techniques (e.g., vibrato versus trill). All mixtures are 2 seconds long and are sampled at 44.1 kHz. The note C is selected as a representative pitch class over octaves 2 to 7 (65.41 Hz to 2093 Hz).

The set of representations to be compared includes the CQT, the original fully-sampled MCFT, and the L-MCFT with two different subsampling methods (critically sampled and sampled to the length of the highest bandpass filter). The minimum and maximum frequencies of the CQT are respectively set to 61.74 Hz (note B1) and 4435 Hz (note $C^{\#}8$) in all cases. The frequency resolution of the CQT is considered as a tunable parameter taking on values in $\{12, 24, 48, 96\}$ (bins/oct). The same parameters are used in the time-frequency representation stage of the MCFT and L-MCFT.

The total number of spectro-temporal filters in the MCFT and L-MCFT filterbanks depends on the scale and rate resolutions, as well as the frequency and time resolutions of the CQT, i.e., the sampling rates of the scale and rate values, respectively. Table 4.1 presents the filter centers of the spectral and temporal filterbanks used in the experiments for different values of the CQT frequency resolution.

Table 4.1. Spectral and temporal filterbank centers for different CQT frequency resolution.

Freq. Resolution	Spectral Filter Centers (cyc/oct)	Temporal Filter Centers (cyc/sec)
12 bins/oct	2^{-4}, 2^{0}, 2^{1}, 2^{2}, $2^{2.5}$	2^{-2}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, 2^{7}, $2^{7.5}$
24 bins/oct	2^{-4}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, $2^{3.5}$	2^{-2}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, $2^{6.5}$
48 bins/oct	2^{-4}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, $2^{4.5}$	2^{-2}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, $2^{5.5}$
96 bins/oct	2^{-4}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, $2^{5.5}$	2^{-2}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, $2^{5.5}$

Table 4.2. Mean SDR (dB) for the results of ideal binary masking on two-source and four-source unison mixtures. Higher values are better. Each number is the overall result over all masking thresholds. “critic" and “bpass" respectively mean critically sampled and sampled to the rate of the highest bandpass filter. “s1r1" means the scale and rate resolutions of the filterbank are 1 (cyc/oct) and 1 (cyc/sec), respectively.

No. Sources	Freq. Resolution	Representation			
		CQT	L-MCFT critic-s1r1	L-MCFT bpass-s1r1	MCFT s1r1
2	12 bins/oct	4.2	4.9	5.1	6.6
	24 bins/oct	5.8	6.0	6.2	8.0
	48 bins/oct	7.1	7.3	7.5	9.0
	96 bins/oct	7.9	8.5	8.6	10.2
4	12 bins/oct	-1.3	-0.2	0.2	1.7
	24 bins/oct	0.5	1.0	1.4	3.1
	48 bins/oct	1.9	2.4	2.7	4.3
	96 bins/oct	2.9	3.8	4.0	5.7

To investigate the effect of different levels of source overlap on the separation performance, two-source and four-source mixtures are considered. In each experiment, ideal binary masks with masking thresholds ranging from 0 dB to 30 dB with a step size of 5 dB are applied to mixtures represented in one of the transform domains and then Source to Distortion Ratio (SDR) values are computed for the estimated sources that are converted back to the time domain.

The separation results are presented in Tables 4.2 and 4.3. In each table, the mean of the SDR values over all mixtures and all masking thresholds are given for each transform-frequency resolution pair. The results in Table 4.2 are obtained using a spectro-temporal filterbank with scale and rate resolutions of 1 (cyc/oct) and 1 (cyc/sec), respectively. It can be observed that even though the L-MCFT provides a lower separability level compared to the original MCFT, it still outperforms the CQT for all frequency resolutions and the performance gain over the CQT increases with an increase in the number of sources per mixture. It can be also observed that a slight oversampling of low-scale/low-rate filters (“bpass" subsampling method) improves the separation performance implying the existence of a trade-off between the amount of discarded information and separation performance in realistic scenarios.

The scale resolution did not show a considerable effect on the separation performance in the experiments, therefore, I will only treat the rate resolution as a tunable parameter for the L-MCFT. Table 4.3 presents the results for a filterbank with scale and rate resolutions of 1 (cyc/oct) and 2 (cyc/sec), respectively. It can be observed that the separability of both MCFT and L-MCFT is improved with an increase in the rate

Table 4.3. Mean SDR (dB) for the results of ideal binary masking on two-source and four-source unison mixtures. Higher values are better. Each number is the overall result over all masking thresholds. “critic" and “bpass" respectively mean critically sampled and sampled to the rate of the highest bandpass filter. “s1r2" means the scale and rate resolutions of the filterbank are 1 (cyc/oct) and 2 (cyc/sec), respectively.

No. Sources	Freq. Resolution	Representation			
		CQT	L-MCFT critic-s1r2	L-MCFT bpass-s1r2	MCFT s1r2
2	12 bins/oct	4.2	6.4	6.8	7.4
	24 bins/oct	5.8	7.2	7.6	8.6
	48 bins/oct	7.1	8.0	8.3	9.5
	96 bins/oct	7.9	8.8	9.0	10.5
4	12 bins/oct	-1.3	1.3	2.0	2.6
	24 bins/oct	0.5	2.2	2.8	3.8
	48 bins/oct	1.9	3.3	3.7	4.8
	96 bins/oct	2.9	4.2	4.5	6.2

resolution and both representations perform significantly better than the CQT. This performance gain is achieved at the cost of increased representation size. However, the rate of growth for the MCFT size is considerably larger compared to the size of the L-MCFT.

Table 4.4 presents the ratio of the L-MCFT size to the MCFT size for all cases included in Tables 4.2 and 4.3. When a filterbank with a rate resolution of 1 (cyc/sec) is used, critical sampling and downsampling to the length of the highest bandpass filter result in representations whose sizes are respectively around 5% and 19% of the original MCFT size. The size reduction is even larger when using a higher rate resolution (around 3% and 14%, of the MCFT size, respectively) implying a higher rate of growth in the computational complexity of the MCFT.

Table 4.4. The ratio of the L-MCFT size to the MCFT size for different frequency resolutions and different subsampling methods. The scale and rate resolutions are 1 (cyc/oct) and 1 (cyc/sec), respectively for all representations.

Filterbank Resolution	Subsampling	Freq. Resolution (bins/oct)			
		12	24	48	96
s1r1	critic	5.7%	5.4%	5.3%	4.7%
	bpass	19.1%	19.1%	18.9%	19.1%
s1r2	critic	3.1%	3.1%	3.2%	2.8%
	bpass	13.5%	13.5%	13.5%	13.7%

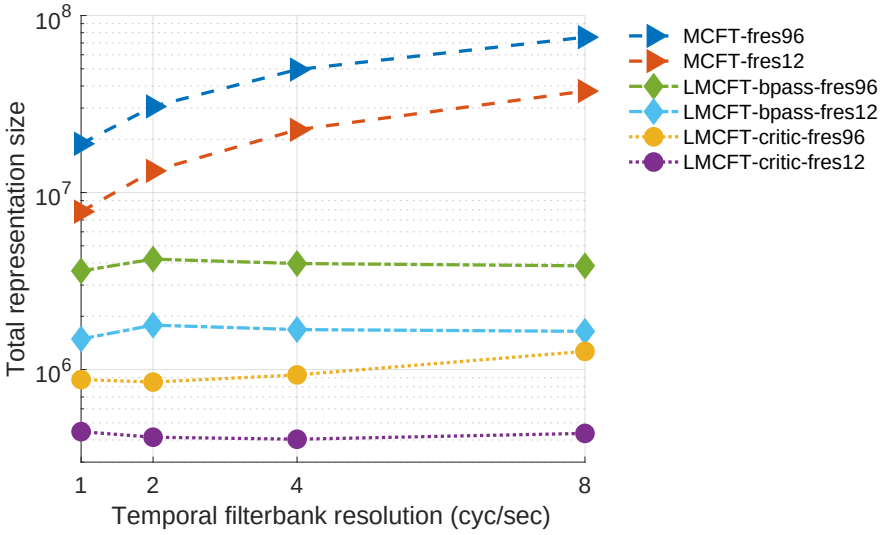

Figure 4.1. Total representation size for the MCFT and L-MCFT versus temporal filter resolution. Each graph corresponds to one value of the CQT frequency resolution.

The sizes of the MCFT and L-MCFT are further compared in Figure 4.1, where the CQT frequency resolution and the rate resolution are tunable parameter taking on values in $\{12, 96\}$ and $\{1, 2, 4, 8\}$, respectively. It can be clearly observed that while the size of both transforms grows with an increase in the frequency or rate resolution, the size of the L-MCFT computed with either subsampling method is significantly smaller than the size of the MCFT with the same parameters. Moreover, the L-MCFT size grows much more slowly with an increase in the parameter values.

4.3. L-MCFT-based source separation training

This section presents a source separation scenario where the L-MCFT is used as the representation domain for separation training. I will consider mixtures of sources with distinct spectro-temporal patterns, which were shown to be captured effectively by the common-fate-based representations. I will compare the performance of a separation system trained on the L-MCFT of audio mixtures to that of a system with comparable characteristics/capacity (e.g., number of learnable parameters) trained on a time-frequency-domain representation, e.g., the STFT of audio mixtures.

4.3.1. Experimental design

In this section, I describe the setup used in the separation training experiments. The goal of this set of experiments is to answer the following question:

(1) Can a deep neural network be trained to perform source separation using the L-MCFT as the input representation? Is training such a network possible with weak labels as well as strong labels?

(2) How does a separator trained on the L-MCFT of the audio signals perform in terms of separation quality compared to a separator trained on a time-frequency representation such as the STFT?

(3) How does a change of input representation from the STFT to L-MCFT affect the network architecture design and number of training epochs?

4.3.1.1. Dataset. The dataset used in this set of experiments is created in a similar way to the dataset used in Section 3.5. The audio data is extracted from *UrbanSound8K*[1] [89], a dataset containing 8732 sound excerpts of length ≤ 4 s, taken from field recordings. From 10 sound classes included in the dataset three classes with distinct spectro-temporal characteristics are chosen: car horn (harmonic with no frequency modulation), dog bark (wideband almost impulsive with sharp frequency modulation patterns), and siren (harmonic with pronounced frequency modulation). Figure 4.2 illustrates the magnitude CQT of examples from the three selected sound classes.

Audio mixtures in the generated mixture dataset are 2 seconds long and sampled at 16 kHz. The choice of 2-second audio mixtures (versus 4-second mixtures in Section 3.5) was made to keep running experiments with different combinations of parameters for the two representations manageable. Each mixture is composed of at least one sound event from one of the three selected classes. Sound events are of arbitrary length, ranging from 0.5 s to 2 s, with a start time sampled uniformly at random under the constraint that the event fits entirely in the 2 s mixture. The level of each sound event is randomly sampled from a uniform distribution of -30 to -25 loudness units full-scale (LUFS) [27]. I use folds 1-6 of *UrbanSound8K* for creating the training set, folds 7-8 for the validation set, and folds 9-10 for the test set. The training, validation, and test sets include 10K, 2K, and 2K mixtures, respectively.

The total number of sound events per mixture is sampled from a zero-truncated Poisson distribution with an expected value of λ. In this set of experiments the value of λ is set to 5. To demonstrate the level of activity of the sound sources and the amount of overlap between them, the frame-level prior probabilities of activity for the three sound classes and the distribution of frames containing different numbers of sources in the entire training set are presented in Tables 4.5 and 4.6, respectively.

[1] https://urbansounddataset.weebly.com/urbansound8k.html

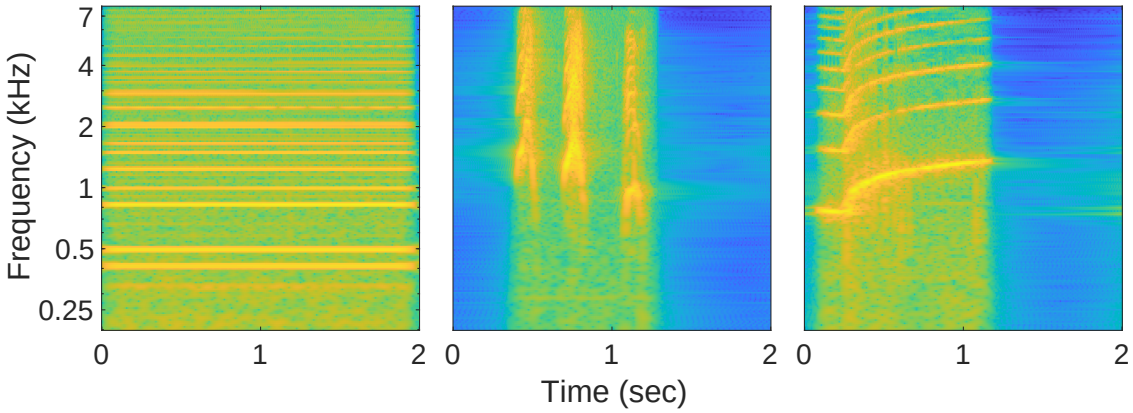

Figure 4.2. Examples of sounds from the three classes included in the dataset: car horn (left), dog bark (middle) and siren (right). Each panel shows the magnitude of the CQT of an audio signal.

Table 4.5. Frame-level prior probabilities of activity for the three selected sound classes. The probabilities are computed for the training dataset with $\lambda = 5$.

Sound class	Car horn	Dog bark	Siren
Prior probabiltiy of activity	0.47	0.57	0.62

Table 4.6. Distribution of frames containing different numbers of sources in the training datasets with $\lambda = 5$ values.

Number of sources per frame	0	1	2	3
Probability	0.13	0.28	0.38	0.21

4.3.1.2. Audio representations. Two audio representations are considered as input for separation training: the L-MCFT and the STFT, as the baseline time-frequency representation. The STFT was selected because: i) it is one of the most commonly used representations in deep-learning-based source separation (much more commonly than the CQT), ii) the joint separation-classification system proposed in Chapter 3 was tested in detail using the STFT as input, thus choosing the STFT makes the network design for the current set of experiments and the comparison between the separation results more straightforward, and iii) neither STFT nor CQT captures the spectro-temporal modulation patterns as explicit dimensions, thus experimenting with one of them is sufficient to demonstrate the fact that learning complex modulation patterns from a time-frequency representation can be difficult for a deep neural network.

The STFT is computed using the square root of a Hann window. I try three different conditions for the STFT experiments, each with a different window size. The window sizes are: 256 (16 ms), 512 (32 ms) and 1024 (64 ms). The hop size between adjacent windows is 25% of the window length in all cases.

In all experiments with the L-MCFT as input representation, it is computed with the critical subsampling method since it generates the smallest (the most memory efficient) version of the representation, which compared to the larger versions imposes the least amount of computational complexity on the separation training process. The minimum and maximum frequencies of the CQT used as the intermediary representation are 200 Hz and 8 kHz, respectively. Since the energy of all sound classes over low frequencies is negligible, components below 200 Hz are included in the lowpass part of the CQT.

One important point to take into account is that vertical patterns in the STFT domain (e.g., dog bark in this dataset) look somehow smeared over low frequencies in the CQT domain due to the low sampling rate of low filters. One observation in separation training was that in practice these partially smeared shapes, which are certainly more complex than straight lines are not easy for deep neural networks to learn. To get simple vertical lines in the CQT domain which give rise to cleaner patterns in the L-MCFT domain, the lower filters must be slightly upsampled. This can be simply done by adding a small offset to the filter bandwidths (see [91]). In these experiments the offset was set visually to 30 Hz (to make impulsive sounds look as close as possible to vertical lines in the CQT domain). To study the effect of the frequency and time resolutions on the performance of the L-MCFT-based system, two values for the frequency resolution of the CQT are considered: 24 (bins/oct) and 48 (bins/oct).

The scale filterbank includes a lowpass filter centered at 2^{-4} (cyc/oct), a bandpass filter centered at 2^{0} (cyc/oct), and a highpass filters covering the remaining range centered at the Nyquist rate for the scale axis, which is determined by the frequency resolution of the CQT. The rate filterbank similarly includes a lowpass filter centered at 2^{-2} (cyc/sec), a bandpass filters at 2^{0} (cyc/sec) and a highpass filters covering the remaining range centered at the Nyquist rate for the rate axis, determined by the time resolution of the CQT. Given the lack of symmetry of the signal representation in the scale-rate domain due to the inclusion of the CQT phase, the mirror bandpass filters and each half of highpass filters should be separately included in the analysis. This gives rise to a total 25 two-dimensional filters in the spectro-temporal filterbank.

The number of filters in the scale and rate filterbanks here is smaller than the number used in the analysis of the unison mixture dataset in Section 4.2.2. This is because the level of source overlap in the current

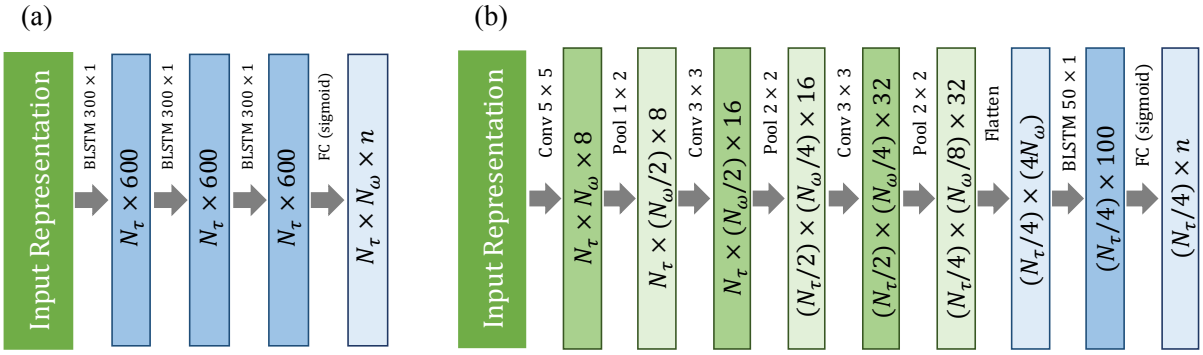

Figure 4.3. Architectures of (a) the separator and (b) the 2D-CRNN frame-level classifier. N_τ and N_ω denote the total number of time frames and frequency bins in the stacked input representation, respectively. n is the total number of sound classes.

dataset is not quite as extreme as in the previous case. Therefore, using a large number of bandpass filters does not improve the performance significantly, however, it does add to the complexity of the representation. The spectral and temporal filters used in all experiments are both Gabor-like (second derivative of a Gaussian pdf) due to its desirable symmetric shape, which allows capturing maximum signal energy in each band while using the critical subsampling method.

4.3.1.3. Network architecture. Bidirectional long short-term memory (BLSTM) networks are commonly used in the context of audio source separation due to their successful performance in capturing the dynamic behavior of audio signals [32][60][79], thus, they are selected for the architecture of the separator block in this set of experiments. The separator block in the joint separation classification system receiving the L-MCFT as input is a 3-layer BLSTM network, with each layer including 300 nodes in each direction. A fully connected layer maps the output of the BLSTM network to n masks with the same size as the input mixture. Activation functions of all BLSTM units are *tanh*. The fully connected layer outputs go through *sigmoid* functions, so that the mask values are always in [0,1]. This network is illustrated in Figure 4.3 (a). To feed the L-MCFT to such a network the filtered slices, which are of different sizes due to critical sampling, are stacked in both vertical and horizontal directions to form a two dimensional shape with stacked frequency values along one axis and stacked time values along the other.

Convolutional recurrent neural networks (CRNNs) have become very popular in the context of audio event detection and classification [2][116]. It was also demonstrated in Chapter 3 that the 2D-CRNN classifier yields the best performance in separation training using weak labels. Thus, this architecture is employed in this set of experiments as well. The frame-level SED classifier is a 2D-CRNN architecture composed

of a 3-layer 2D convolutional network including max pooling after each layer, followed by a BLSTM layer and a fully connected layer that maps the BLSTM output to class probabilities. Activation functions of convolutional, BLSTM, and fully connected layers are *relu*, *tanh*, and *sigmoid*, respectively. The output of each convolutional layer is batch normalized prior to the application of the activation function. Figure 4.3 (b) shows the architecture of the frame-level classifier.

For the separation system receiving the STFT as input, two architectures are considered: i) the same architecture as the one used with the L-MCFT input, and ii) the architecture used in Section 3.4.6, which is a similar but larger BLSTM network with 600 nodes in each direction. The separation performance for a system receiving the STFT as input and trained using different types of labels was discussed in detail in Chapter 3 and it was demonstrated that training on strong labels yields the upper bound on the system performance. Thus, for the sake of brevity, here the STFT-based system is only trained using strong labels (presumabely the best case scenario). The L-MCFT-based system is trained using strong label as well as frame-level labels to investigate the effect of label strength on the separation performance in this new representation domain.

4.3.2. Training setup

The mask inference objective function presented in Section 3.4.2 is used in the strong-label case and the joint separation-classification objective proposed in Section 3.4.4.2 for training on the frame-level labels. In the frame-level case, since all sound classes are active around half of the time, no class weighting is used for the objective function terms (see Table 4.5). In all training sessions, the ADAM optimizer was used, with a learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The batch size was set to 10 in all experiments with the STFT as input and to 5 for all experiments with the L-MCFT as input. All networks are trained until the loss on the validation set stops improving for five consecutive epochs, with a maximum of 100 epochs.

4.3.3. Results

In this section, I present the separation results for L-MCFT-based separation systems trained on strong labels and frame-level labels. I compare the performance of the former separation system to a system receiving the STFT as input representation and trained on strong labels. The comparison is made based on: i) separation quality, ii) number of learnable parameters, and iii) the speed of convergence (number of training epochs).

Table 4.7. Mean/Median SI-SDR improvement (dB) for separators of different sizes receiving the STFT with different window sizes as input, over all classes. BLSTM-300 means a 3-layer BLSTM network with 300 nodes in each direction per layer. In all cases the overlap between windows is 75%, the separator is trained on strong labels, and the mean and median of the input SI-SDR are both −2.4 dB. Higher values are better.

		Sound class			
Network	Window Size	Car horn	Dog bark	Siren	Overall
BLSTM-300	16 ms	3.9 / 2.2	3.4 / 2.3	1.7 / 1.5	3.0 / 2.0
	32 ms	4.1 / 2.7	3.6 / 2.5	1.6 / 1.5	3.1 / 2.2
	64 ms	3.8 / 2.7	3.1 / 2.1	0.9 / 0.9	2.6 / 1.8
BLSTM-600	16 ms	4.1 / 2.3	3.7 / 2.5	2.0 / 1.9	3.3 / 2.3
	32 ms	3.9 / 2.5	3.6 / 2.5	1.7 / 1.6	3.1 / 2.2
	64 ms	3.6 / 2.6	2.6 / 1.6	1.1 / 0.9	2.4 / 1.6

Table 4.8. Mean/Median SI-SDR improvement (dB) for separators receiving the L-MCFT with different frequency resolutions as input and trained on different labels types, over all classes. In all cases, the separator is a 3-layer BLSTM network with 300 nodes in each direction per layer and the mean and median of the input SI-SDR are both −2.4 dB. In all frame-level cases, $\alpha = 1$. Higher values are better.

		Sound class			
Label Type	Freq. Resolution	Car horn	Dog bark	Siren	Overall
Strong	24 bins/oct	4.7 / 4.1	4.0 / 3.8	1.7 / 2.3	3.5 / 3.5
	48 bins/oct	6.3 / 5.7	3.4 / 3.3	1.6 / 2.4	3.8 / 3.9
Frame-level	24 bins/oct	2.4 / 1.9	3.5 / 3.5	1.2 / 1.6	2.4 / 2.5
	48 bins/oct	4.2 / 3.7	2.9 / 2.9	1.1 / 1.5	2.7 / 2.9

Similar to the performance evaluation in Section 3.5, where a realistic separation algorithm is being assessed (as opposed to ideal binary masks), the separation quality is measured by Scale-independent Source to Distortion Ration (SI-SDR) in these experiments. The results for STFT-based systems trained on strong labels and three different window sizes are presented in Table 4.7. As it can be seen, the separation performances of the systems using shorter window sizes (16 ms and 32 ms) are very close while increasing the window size to 64 ms (decreasing the time resolution) results in a decline in the overall separation quality. The best performing system, BLSTM-600 with an STFT window size of 16 ms is selected to be compared to the L-MCFT-system in terms of the three aforementioned criteria.

Table 4.8 presents the results of experiments with L-MCFT-based systems using different CQT frequency resolutions and trained on strong and frame-level labels. Increasing the frequency resolution seems to have an overall positive effect on the separation results for both label types. However, this performance gain

is mainly due to the car horn class (harmonic with no modulation), the dog bark class seems to have a better separation quality with a lower frequency resolution and the quality of the siren class does not show a considerable difference.

A noticeable difference between the results of the L-MCFT-based and STFT-based systems is that the mean and median of the distributions seem to be much closer in the L-MCFT case, implying the concentration of a larger number of points in the distribution over high SI-SDR values (the mean value is not affected much by the outliers). It can be observed that both L-MCFT-based systems trained on strong labels provide overall performance improvement compared to the best performing STFT-based system. The most improvement in terms of median of SI-SDR values is achieved for the car horn class (up to 3.4 dB) and the least for the siren class (up to 0.5 dB). As a reminder, the siren class is very diverse and thus difficult to learn regardless of the representation or the label type.

In the experiments with the L-MCFT-based system and frame-level labels a range of values were tried for the α parameter, i.e., the weight of the mixture loss term in the joint separation-classification objective function (see Section 3.4.4). The best performance was achieved for $\alpha = 1$, which is the value used for the reported results. It is interesting to note that the L-MCFT-based system trained with frame-level labels provides an overall performance comparable (median of the SI-SDR values) to the performance of the STFT-based system.

Figure 4.4 illustrates the distribution of output SI-SDR values for the L-MCFT-based system with a frequency resolution of 48 (bins/oct) versus the output SI-SDR values for the best performing STFT-based network with a BLSTM-600 separation network and a window size of 16 ms. Both systems are trained on strong labels. The identity line is displayed with a red dashed line and identity $\pm$ 5 dB with black dashed lines. It can be observed that in all cases the high-density part of the distribution is mostly located between the identity line and identity + 5 dB, indicating that the output of the L-MCFT-based system for the majority of the testing examples is equal to or up to 5 dB higher than the output of the **best-performing** STFT-based system. In the case of the car horn class, the high-density part is almost entirely located above the identity line demonstrating that the performance of the L-MCFT system is significantly better than the STFT-based system.

The L-MCFT-based and STFT-based systems are further compared in Table 4.9. From the set of STFT window sizes used with each separator network the one resulting in the best SI-SDR median is included in the

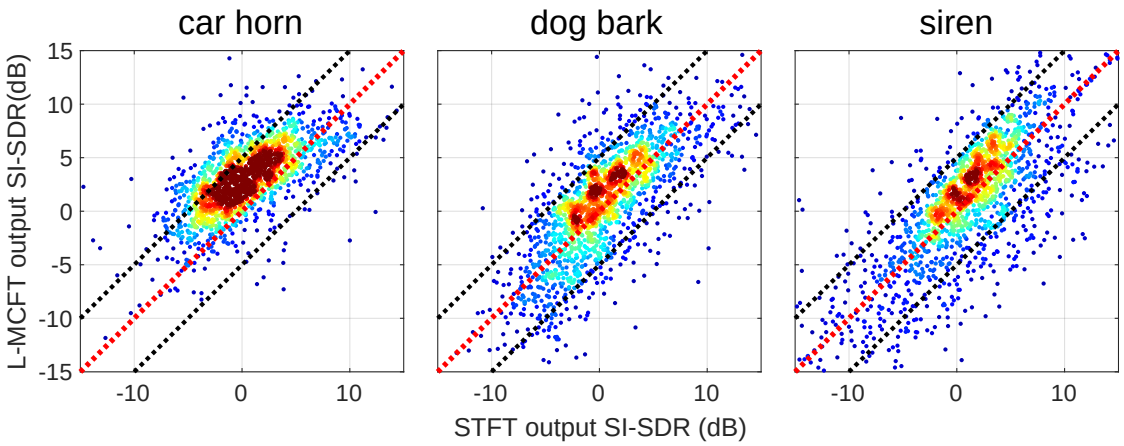

Figure 4.4. Separation results for all sound classes when the separator is trained on the strong labels. Each panel shows the output SI-SDR of the L-MCFT-based separator with a frequency resolution of 48 (bins/oct) versus the output SI-SDR of the STFT-based BLSTM-600 network with a window size of 16 ms. The identity line is displayed by a red dashed line. The black dashed lines have a slope of one and offsets of ±5 dB. There are over 1500 datapoints in each plot (between 1553 for siren and 1583 for car horn). Warmer colors mean higher densities of data points. Higher values on x-axis and y-axis are better.

table. The L-MCFT-based systems clearly outperform the STFT-based systems regardless of the network capacity or the STFT window size. The number of learnable parameters of the smaller STFT-based network is close to the number of parameters of the L-MCFT-based networks, however, its convergence speed is almost 5 times slower. Increasing the number of learnable parameters in the case of the larger STFT-based system (around 3.5 times as many parameters) speeds up the convergence, without providing any meaningful improvement in the separation quality. This observation clearly goes against the argument usually made in the context of deep-learning-based audio processing that a network with a high enough capacity can easily learn from the data all the necessary information that is traditionally provided by analytical signal processing methods and representations. All criteria considered, both L-MCFT-based systems appear to perform more successfully in these set of experiments than the STFT-based systems.

4.4. Conclusion

In this chapter, I presented a subsampling method for the MCFT which significantly reduces the complexity of the representation and thus makes it easier to use in realistic scenarios. Similar to the original MCFT, the subsampled version, termed L-MCFT is perfectly invertible and captures spectro-temporal modulation patterns as explicit dimensions. Benchmarking experiments with the unison mixtures of musical

Table 4.9. Comparison between L-MCFT-based and STFT-based networks trained on strong labels, in terms of separation performance, number of learnable parameters in the separator network and the speed of convergence (total number of training epochs before reaching a local minimum).

Representation	Network	Median SI-SDR (dB)	No. Parameters	No. Epochs
STFT (16 ms)	BLSTM-600	2.3	29.9×10^6	51
STFT (32 ms)	BLSTM-300	2.2	8.3×10^6	91
L-MCFT (24 bis/oct)	BLSTM-300	3.5	8.6×10^6	19
L-MCFT (48 bis/oct)	BLSTM-300	3.9	9.7×10^6	18

notes played with different techniques showed that with minimal parameter tuning, the L-MCFT provides a significantly higher level of separability compared to the baseline time-frequency representation, i.e., the CQT, while being more than five times smaller than the original MCFT with the same parameter values.

It was further demonstrated that the L-MCFT can replace a commonly used representation domain such as the STFT in deep-learning-based audio source separation. In a set of experiments on mixtures of highly-overlapped urban sounds with distinct spectro-temporal modulations, the L-MCFT-based separation systems trained using strong labels outperformed separation systems of different sizes receiving the STFT with a range of window sizes as input. Moreover, the L-MCFT-based systems were trained over a lower number of epochs compared to the STFT-based systems to reach the reported separation performance.

Finally, the findings of Chapter 3 and Section 4.2 were combined by training joint separation-classification systems on the L-MCFT of audio mixtures as input and frame-level sound class labels as target. Similar to the results of STFT-based separation systems in Chapter 3, L-MCFT-based systems trained on frame-level labels yield a lower separation quality than the systems trained on strong labels. However, in this set of experiments, the overall separation quality provided by L-MCFT-based systems trained on frame-level labels appears to be comparable to the separation quality of STFT-based trained on strong labels.

CHAPTER 5

Conclusion

In this dissertation, I have proposed methods for audio signal processing and for training deep models that are inspired by biological auditory systems, addressing two major challenges in the field of audio source separation, an important task for machine hearing:

- Creating a biologically-inspired audio representation that disentangles audio sources that overlap in time and frequency in a way that is practically useable for source separation and sound object recognition.
- Learning to segregate sound sources from training data using only the presence or absences of the sources in the scene (weak labels) as training targets.

This work paves the way for combining sophisticated, interpretable audio representations with modern deep learning networks for a variety of tasks (e.g., audio source separation, even detection, and classification). The use of such representations increases the interpretability of learned systems and allows smaller networks to achieve higher performance than is possible with lower-level audio representations currently used in deep-learning-based audio processing. Moreover, the approach to training models on weak labels can be applied more broadly than the specific task illustrating its effectiveness. Adopting such training approaches allows the use of larger and more realistic training datasets (which are currently difficult or impossible to use) and thus promises the development of models that perform more successfully in realistic situations.

In Chapter 2, I presented my work on the development of the MCFT [77][81], a biologically-inspired audio representation that disentangles two audio sources that overlap in time and frequency and discussed its properties with the aid of illustrative examples. Objective measures for two desirable representation properties: separability of source signals and clusterability of components of each signal in the representation domain were used to compare the MCFT to commonly used time-frequency representations, the STFT and CQT, as well as the CFT [103], another common-fate-based audio representation. I showed that the MCFT is perfectly invertible (i.e., a time-domain audio signal can be perfectly reconstructed given its representation in the MCFT domain). I further demonstrated that the MCFT increases the separability of mixtures of

multiple audio signals with a high level of overlap in the time-frequency domain, and that the multi-resolution character of the MCFT circumvents the resolution issue of the CFT, whose separability and clusterability are significantly affected by the choice of the time-frequency window size used in its computation. Figure 2.18 summarizes the results of separability and clusterability experiments on a dataset containing unison (same-pitch) mixtures of musical notes played by different instruments and different techniques.

In Chapter 3, I presented a new algorithm for supervised audio source separation training in scenarios where ground truth isolated sources are not available to be used as training targets [79][80], e.g., recording a bird song in a forest or recording the sound of a machine part that only occurs when a machine is running. This method only relies on weak labels indicating the activity of different sound types over time, which can be provided by human listeners (e.g. mechanical turk workers). This method bridges the gap between strong (time-frequency-bin-level) and weak (frame- or clip-level) labels by using the output of a a sound identification system, trained on audio mixtures, as a metric for assessing the separation performance. The core component of this new approach is a multi-task optimization framework combining an audio event classification objective with a separation-specific objective enforcing the separated sources to sum up to the mixture. It was shown that although there is still a gap between the separation performance of systems trained on strong and weak labels, the proposed method is capable of isolating the sources to a great extent. Moreover, it was demonstrated that the proposed framework is flexible with respect to network architectures used for the classification and separation tasks. Figure 3.5 summarizes the most important experimental results of Chapter 3, for separation systems trained on a dataset containing mixtures of urban sounds.

The findings of Chapter 2 and Chapter 3 were brought together in Chapter 4, where I first presented a subsampling method that significantly reduces the size of the original MCFT to make it more practical for scenarios where low computational complexity is desirable. Through a series of a benchmarking experiments, I showed that the subsampled version, termed L-MCFT, is perfectly invertible and by capturing spectro-temporal modulations as explicit dimensions offers a higher level of separability compared to the baseline time-frequency representation, i.e., the CQT. Next, I demonstrated the utility of the L-MCFT in realistic scenarios by training deep neural networks receiving the L-MCFT as input to separate mixtures of highly overlapped audio sources with distinct spectro-temporal modulation patterns. Both strong and frame-level labels were successfully used as targets in L-MCFT-based separation training. The performances of the L-MCFT-based systems trained on both label types were compared to the performance of a system receiving

a time-frequency representation (i.e., the STFT) as input and strong labels as training target. It was shown that when trained on strong labels, the L-MCFT-based system outperforms the STFT-based system (trained on strong labels) and when trained on frame-level labels it yields separation results comparable to the results of the STFT-based system (trained on strong labels).

5.1. Limitations

The principle limitation of the current version of the MCFT (and by extension the L-MCFT) is that it does not explicitly capture spatial cues, e.g., Interaural Level Difference, (ILD) and Interaural Time Difference (ITD), and thus it is better suited to tasks dealing with single-channel mixture recordings. Although a similar approach to methods that compute the representation for each recording channel separately and then extract the spatial cues from them (e.g., the DUET source separation algorithms [87]) can be taken with the MCFT as well, it would be more convenient if the representation inherently makes such information salient, and hence easier to extract for audio processing algorithms dealing with multi-channel audio recordings.

Although the L-MCFT makes the modulation information more salient and thus helps the separation performance as well as the convergence speed when training a deep neural network, it is still heavier than time-frequency representations, and thus introduces some extra latency in the training/testing pipeline. Techniques available in deep learning toolboxes such as parallel example loading and processing alleviate this problem to some extent. However, a probably more elegant solution would be to develop an intelligent mechanism (similar to the auditory attention mechanism in the human auditory system) that based on the type of a target sound source determines the parts of the representation (e.g., certain spectro-temporal filters) that are required so that the computation of the unnecessary information can be bypassed entirely.

The weakly supervised separation training method proposed in this work is an important step towards the development of models trained on large amounts of data collected in realistic setups. However, there is still a gap between the separation performance of systems trained on strong labels and those trained on weak labels. It was shown in Chapter 4 that using a representation, which makes certain cues that are important to the separation task salient results in some performance boost. Exploring other ways to make the necessary information easier to access while training the network (or at the inference time), for instance augmenting the joint separation-classification system by an auditory-inspired attention mechanism is left to future work.

5.2. Future work

The audio representation method and the joint separation-classification framework proposed in this dissertation can be used towards improving the performance of assistive hearing devices (e.g., hearing aids). Although current assistive hearing devices can effectively suppress some background noise, they cannot enhance a target source effectively when there are other sources in the auditory scene that have spectro-temporal characteristics similar to the target source (e.g., a multi-talker scenario) or if it is not clear what source is "the target" (e.g., speech and music are both present in a sound mixture). To improve the performance of assistive hearing devices in such scenarios, the devices must be able to 1) properly segregate the sources in the sound mixture, 2) determine what sound source a listener is trying to attend, and then 3) recombine the sounds to make it easier to understand the attended signal (e.g., making it louder).

Recent studies on the human auditory system have demonstrated that an attended speech stream can be identified by non-invasively measuring the electrical activity from the brain of a listener using electroencephalography (EEG) and then comparing the measured activity to the energy envelopes of the sound sources in the environment, a technique termed Auditory Attention Decoding (AAD) [73] [112].

An AAD system receives as input a mixture of sounds recorded with one or more microphones. To compare each sound source to neural activity of the listener and decode the attended source, the AAD system needs to first separate the sources. Beamforming is a possible solution if the hearing device is equipped with multiple microphones and if the target and interfering sources are spatially well separated [83][84][3]. However, in some scenarios spatial cues are absent or unreliable, including situations where multiple speech streams come from the same direction (e.g., cocktail parties, Zoom meetings, etc.).

The methods proposed in this dissertation will allow the development of more powerful AAD systems that can detect and amplify an attended speech stream based on a combination of auditory cues (e.g., spectro-temporal structure in naturally occurring signals representing the sound's timbre, pitch, and other attributes). An AAD system can be designed such that it is composed of a separation block followed by a detection block (e.g., a SED classifier), both receiving as input a representation that explicitly encodes spectro-temporal modulations. The separation block receives a single-channel mixture and produces estimates of the speech streams present in the mixture. The detection block then compares each estimated source to EEG recordings to decode which is the target speech stream. The AAD separator can be trained

using the proposed joint separation-classification approach and only on weakly labeled training data [80], which allows the use of larger, more diverse, and more realistic datasets.

More specifically, two separation training methods within the joint separation-classification framework can be tried and compared: 1) Training the separator using the mixtures and the classifier as critic to separate all speech streams. The spectrogram of each speech stream is compared to the reconstructed spectrogram of the attention signal to detect the attended source. 2) Training the separator using the mixtures as well as attention signals (class-conditional approach) to decompose the mixture into a foreground (the attended source) and a background containing all the unattended sources. The classifier as critic should label both correctly. The latter approach reduces the computational burden at the inference time.

Development of algorithms capable of exploiting the auditory attention information represents a foundational step towards building brain-controlled assistive hearing devices, which can improve communication in social situations not only for listeners with hearing impairment and non-native speakers, but also for anyone conversing while wearing a mask in a post COVID-19 world.

APPENDIX A

Spectral and temporal filters

In this appendix, a list of seed functions, which can be used as spectral or temporal filters is presented. In each case, f and F denote the function in the original domain and the Fourier transform domain, respectively. C is the dilation factor in all the equations. σ denotes the spread parameter for the Gabor function, and β denotes the time constant of the exponential term for the Gammatone and damped sinusoid functions.

I) Gabor-like function (second derivative of a Gaussian pdf):

$$f(x;C) = C(1-2(\pi Cx)^2)e^{-(\pi Cx)^2} \tag{A.1}$$

$$F(y;C) = (\frac{y}{C})^2 e^{(1-(\frac{y}{C})^2)} \tag{A.2}$$

II) Gabor function:

$$f(x;C,\sigma) = Ccos(2\pi Cx)e^{-\frac{(Cx)^2}{2\sigma^2}} \tag{A.3}$$

$$F(y;C,\sigma) = e^{-2\pi^2\sigma^2(\frac{y}{C}-1)^2} + e^{-2\pi^2\sigma^2(\frac{y}{C}+1)^2} \tag{A.4}$$

III) Gammatone function:

$$f(x;C,\beta) = C(Cx)e^{-(\beta Cx)}sin(2\pi Cx) \tag{A.5}$$

$$F(y;C,\beta) = j\left(\frac{\left(\beta + j2\pi(\frac{y}{C}-1)\right)^3 - \left(\beta + j2\pi(\frac{y}{C}+1)\right)^3}{\left(\beta + j2\pi(\frac{y}{C}-1)\right)^3\left(\beta + j2\pi(\frac{y}{C}+1)\right)^3}\right) \tag{A.6}$$

IV) Damped sinusoid function:

$$f(x; C, \beta) = Ce^{-\beta(Cx)}sin(2\pi Cx) \tag{A.7}$$

$$F(y; C, \beta) = \frac{2\pi}{\beta^2 - 4\pi^2((\frac{y}{C})^2 - 1) + j4\pi\beta(\frac{y}{C})} \tag{A.8}$$

References

[1] Mototsugu Abe and Shigeru Ando. "Auditory scene analysis based on time-frequency integration of shared FM and AM". in: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. vol. 4. IEEE. 1998, pp. 2421–2424.

[2] Sharath Adavanne and Tuomas Virtanen. "Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network". In: *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. Nov. 2017.

[3] Ali Aroudi, Marc Delcroix, Tomohiro Nakatani, Keisuke Kinoshita, Shoko Araki, and Simon Doclo. "Cognitive-driven convolutional beamforming using EEG-based auditory attention decoding". In: *arXiv preprint arXiv:2005.04669* (2020).

[4] Francis R Bach and Michael I Jordan. "Learning spectral clustering, with application to speech separation". In: *Journal of Machine Learning Research* 7.Oct (2006), pp. 1963–2001.

[5] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[6] Juan José Burred, Axel Robel, and Thomas Sikora. "Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.3 (2010), pp. 663–674.

[7] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. "Convolutional recurrent neural networks for polyphonic sound event detection". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1291–1303.

[8] Mark Cartwright, Graham Dove, Ana Elisa Méndez Méndez, Juan P Bello, and Oded Nov. "Crowdsourcing multi-label audio annotation tasks with citizen scientists". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–11.

[9] Taishih Chi, Powen Ru, and Shihab A Shamma. "Multiresolution spectrotemporal analysis of complex sounds". In: *The Journal of the Acoustical Society of America* 118.2 (2005), pp. 887–906.

[10] Selina Chu, Shrikanth Narayanan, and C-C Jay Kuo. "Environmental sound recognition with time–frequency audio features". In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6 (2009), pp. 1142–1158.

[11] Martin Cooke, John R Hershey, and Steven J Rennie. "Monaural speech separation and recognition challenge". In: *Computer Speech & Language* 24.1 (2010), pp. 1–15.

[12] Elliot Creager, Noah Stein, Roland Badeau, and Philippe Depalle. "Nonnegative tensor factorization with frequency modulation cues for blind audio source separation". In: *17th International Society for Music Information Retrieval (ISMIR) Conference*. 2016.

[13] Didier A Depireux, Jonathan Z Simon, David J Klein, and Shihab A Shamma. "Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex". In: *Journal of neurophysiology* 85.3 (2001), pp. 1220–1234.

[14] Frank Ehlers and Heinz G Schuster. "Blind separation of convolutive mixtures and an application in automatic speech recognition in a noisy environment". In: *IEEE Transactions on Signal processing* 45.10 (1997), pp. 2608–2612.

[15] Mounya Elhilali and Shihab A Shamma. "A cocktail party with a cortical twist: how cortical mechanisms contribute to sound segregation". In: *The Journal of the Acoustical Society of America* 124.6 (2008), pp. 3751–3771.

[16] Scott Epter, Mukkai Krishnamoorthy, and M Zaki. "Clusterability detection and cluster initialization". In: *Proceedings of the Workshop on Clustering High Dimensional Data and its Applications at the 2nd SIAM International Conference on Data Mining*. 2002, pp. 47–58.

[17] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 708–712.

[18] Hakan Erdogan, John R. Hershey, Shinji Watanabe, and Jonathan Le Roux. "Phase-Sensitive and Recognition-Boosted Speech Separation using Deep Recurrent Neural Networks". In: Apr. 2015, pp. 708–712.

[19] Sebastian Ewert and Mark B Sandler. "Structured dropout for weak label and multi-instance learning and its application to score-informed source separation". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 2277–2281.

[20] Derry Fitzgerald. "Upmixing from mono-a source separation approach". In: *2011 17th International Conference on Digital Signal Processing (DSP)*. IEEE. 2011, pp. 1–7.

[21] Derry FitzGerald, Matt Cranitch, and Eugene Coyle. "Non-negative tensor factorisation for sound source separation". In: *IN: PROCEEDINGS OF IRISH SIGNALS AND SYSTEMS CONFERENCE*. Citeseer. 2005.

[22] Jonathan Fritz, Shihab Shamma, Mounya Elhilali, and David Klein. "Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex". In: *Nature neuroscience* 6.11 (2003), p. 1216.

[23] Jonathan B Fritz, Mounya Elhilali, Stephen V David, and Shihab A Shamma. "Auditory attention—focusing the searchlight on sound". In: *Current opinion in neurobiology* 17.4 (2007), pp. 437–455.

[24] Hiromasa Fujihara, Masataka Goto, Jun Ogata, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno. "Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals". In: *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*. IEEE. 2006, pp. 257–264.

[25] Ruohan Gao and Kristen Grauman. "Co-separating sounds of visual objects". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3879–3888.

[26] John N Gowdy and Zekeriya Tufekci. "Mel-scaled discrete wavelet coefficients for speech recognition". In: *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. Vol. 3. IEEE. 2000, pp. 1351–1354.

[27] EM Grimm, R Van Everdingen, and MJLC Schöpping. "Toward a recommendation for a European standard of peak and LKFS loudness levels". In: *SMPTE Motion Imaging Journal* 119.3 (2010), pp. 28–34.

[28] Ervin R Hafter, Anastasios Sarampalis, and Psyche Loui. "Auditory attention and filters". In: *Auditory perception of sound sources*. Springer, 2008, pp. 115–142.

[29] Simon Haykin and Zhe Chen. "The cocktail party problem". In: *Neural computation* 17.9 (2005), pp. 1875–1902.

[30] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. “Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation.” In: *International Society for Music Information Retrieval conference (ISMIR)*. 2009, pp. 327–332.

[31] Toni Heittola, Annamaria Mesaros, Tuomas Virtanen, and Antti Eronen. “Sound event detection in multisource environments using source separation”. In: *Machine Listening in Multisource Environments*. 2011.

[32] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. “Deep clustering: Discriminative embeddings for segmentation and separation”. In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE. 2016, pp. 31–35.

[33] Eric Humphrey, Simon Durand, and Brian McFee. “OpenMIC-2018: An Open Data-set for Multiple Instrument Recognition.” In: *ISMIR*. 2018, pp. 438–444.

[34] Satoshi Imai. “Cepstral analysis synthesis on the mel frequency scale”. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’83.* Vol. 8. IEEE. 1983, pp. 93–96.

[35] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. “Single-channel multi-speaker separation using deep clustering”. In: *arXiv preprint arXiv:1607.02173* (2016).

[36] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. “Single-Channel Multi-Speaker Separation using Deep Clustering”. In: Sept. 2016, pp. 545–549.

[37] Se-Woon Jeon, Young-Cheol Park, Seok-Pil Lee, and Dae-Hee Youn. “Robust representation of spatial sound in stereo-to-multichannel upmix”. In: *Audio Engineering Society Convention 128*. AES. 2010.

[38] Ertuğ Karamatlı, Ali Taylan Cemgil, and Serap Kırbız. “Audio source separation using variational autoencoders and weak class supervision”. In: *IEEE Signal Processing Letters* 26.9 (2019), pp. 1349–1353.

[39] Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey. “Universal Sound Separation”. In: Oct. 2019.

[40] Bongjun Kim and Bryan Pardo. “I-SED: an Interactive Sound Event Detector”. In: *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM. 2017, pp. 553–557.

[41] Bongjun Kim and Bryan Pardo. “Sound event detection using point-labeled data”. In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2019.

[42] Bongjun Kim and Bryan Pardo. “Sound event detection using point-labeled data”. In: Oct. 2019.

[43] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. "Multi-Talker Speech Separation with Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks". In: 25.10 (2017), pp. 1901–1913.

[44] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.10 (2017), pp. 1901–1913.

[45] Qiuqiang Kong, Yong Xu, Iwona Sobieraj, Wenwu Wang, and Mark D Plumbley. "Sound event detection and time–frequency segmentation from weakly labelled data". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.4 (2019), pp. 777–787.

[46] Lakshmi Krishnan, Mounya Elhilali, and Shihab Shamma. "Segregating complex sound sources through temporal coherence". In: *PLoS Comput Biol* 10.12 (2014), e1003985.

[47] R Kubichek. "Mel-cepstral distance measure for objective speech quality assessment". In: *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on.* Vol. 1. IEEE. 1993, pp. 125–128.

[48] Anurag Kumar and Bhiksha Raj. "Audio event detection using weakly labeled data". In: 2016, pp. 1038–1047.

[49] Rajath Kumar, Yi Luo, and Nima Mesgarani. "Music Source Activity Detection and Separation Using Deep Attractor Network." In: *Interspeech*. 2018, pp. 347–351.

[50] Jonathan Le Roux, Gordon Wichern, Shinji Watanabe, Andy Sarroff, and John R Hershey. "Phasebook and friends: Leveraging discrete representations for source separation". In: *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019), pp. 370–382.

[51] Jonathan Le Roux, Gordon Wichern, Shinji Watanabe, Andy Sarroff, and John R Hershey. "Phasebook and Friends: Leveraging discrete representations for source separation". In: *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019), pp. 370–382.

[52] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. "SDR–Half-baked or well done?" In: May 2019, pp. 626–630.

[53] Chang-Hsing Lee, Chin-Chuan Han, and Ching-Chien Chuang. "Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.8 (2008), pp. 1541–1550.

[54] David Little and Bryan Pardo. “Learning Musical Instruments from Mixtures of Audio with Weak Labels.” In: Sept. 2008, pp. 127–132.

[55] Antoine Liutkus, Derry Fitzgerald, Zafar Rafii, Bryan Pardo, and Laurent Daudet. “Kernel additive models for source separation”. In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4298–4310.

[56] Beth Logan et al. “Mel Frequency Cepstral Coefficients for Music Modeling.” In: *ISMIR*. vol. 270. 2000, pp. 1–11.

[57] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. “Speech enhancement based on deep denoising autoencoder.” In: *Interspeech*. 2013, pp. 436–440.

[58] Yi Luo and Nima Mesgarani. “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation”. In: *IEEE/ACM transactions on audio, speech, and language processing* 27.8 (2019), pp. 1256–1266.

[59] Yi Luo and Nima Mesgarani. “Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation”. In: 27.8 (2019), pp. 1256–1266.

[60] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani. “Deep clustering and conventional networks for music separation: Stronger together”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 61–65.

[61] Richard Lyon. “A computational model of binaural localization and separation”. In: *ICASSP’83. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 8. IEEE. 1983, pp. 1148–1151.

[62] Richard Lyon. “Computational models of neural auditory processing”. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’84*. Vol. 9. IEEE. 1984, pp. 41–44.

[63] Richard F Lyon, Andreas G Katsiamis, and Emmanuel M Drakakis. “History and future of auditory filter models”. In: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE. 2010, pp. 3809–3812.

[64] Shoji Makino, Shoko Araki, Ryo Mukai, and Hiroshi Sawada. “Audio source separation based on independent component analysis”. In: *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*. Vol. 5. IEEE. 2004, pp. V–V.

[65] Michael I Mandel and Daniel PW Ellis. "Multiple-instance learning for music information retrieval". In: Sept. 2008, pp. 577–582.

[66] Ethan Manilow, Prem Seetharaman, Fatemeh Pishdadian, and Bryan Pardo. "Predicting algorithm efficacy for adaptive multi-cue source separation". In: *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017 IEEE Workshop on*. IEEE. 2017, pp. 274–278.

[67] Brian McFee, Justin Salamon, and Juan Pablo Bello. "Adaptive pooling operators for weakly labeled sound event detection". In: 26.11 (2018), pp. 2180–2193.

[68] Annamaria Mesaros, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. "Sound event detection in the DCASE 2017 challenge". In: 27.6 (2019), pp. 992–1006.

[69] Nima Mesgarani and Shihab Shamma. "Denoising in the domain of spectrotemporal modulations". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2007.3 (2007), p. 3.

[70] Michael Michelashvili, Sagie Benaim, and Lior Wolf. "Semi-supervised monaural singing voice separation with a masking network trained on synthetic mixtures". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 291–295.

[71] John C Middlebrooks, Jonathan Z Simon, Arthur N Popper, and Richard R Fay. *The auditory system at the cocktail party*. Vol. 60. Springer, 2017.

[72] Arun Narayanan and DeLiang Wang. "Ideal ratio mask estimation using deep neural networks for robust speech recognition". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 7092–7096.

[73] James A O'Sullivan, Alan J Power, Nima Mesgarani, Siddharth Rajaram, John J Foxe, Barbara G Shinn-Cunningham, Malcolm Slaney, Shihab A Shamma, and Edmund C Lalor. "Attentional selection in a cocktail party environment can be decoded from single-trial EEG". in: *Cerebral cortex* 25.7 (2015), pp. 1697–1706.

[74] Lucas C Parra and Christopher V Alvino. "Geometric source separation: Merging convolutive source separation with geometric beamforming". In: *IEEE Transactions on Speech and Audio Processing* 10.6 (2002), pp. 352–362.

[75] Roy D Patterson, Mike H Allerhand, and Christian Giguere. "Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform". In: *The Journal of the Acoustical Society of America* 98.4 (1995), pp. 1890–1894.

[76] Michael Syskind Pedersen, DeLiang Wang, Jan Larsen, and Ulrik Kjems. "Separating underdetermined convolutive speech mixtures". In: *International Conference on Independent Component Analysis and Signal Separation*. Springer. 2006, pp. 674–681.

[77] Fatemeh Pishdadian, Bryan Pardo, and Antoine Liutkus. "A multi-resolution approach to common fate-based audio separation". In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 566–570.

[78] Fatemeh Pishdadian, Prem Seetharaman, Bongjun Kim, and Bryan Pardo. "Classifying Non-speech Vocals: Deep vs Signal Processing Representations". In: (2019).

[79] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux. "Finding strength in weakness: Learning to separate sounds with weak supervision". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2386–2399.

[80] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux. "Learning to Separate Sounds from Weakly Labeled Scenes". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 91–95.

[81] Fatemeh Pishdadian and Bryan Pardo. "Multi-Resolution Common Fate Transform". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.2 (2019), pp. 342–354.

[82] Mark D Plumbley, Samer A Abdallah, Juan Pablo Bello, Mike E Davies, Giuliano Monti, and Mark B Sandler. "Automatic music transcription and audio source separation". In: *Cybernetics & Systems* 33.6 (2002), pp. 603–627.

[83] Wenqiang Pu, Jinjun Xiao, Tao Zhang, and Zhi-Quan Luo. "A Joint Auditory Attention Decoding and Adaptive Binaural Beamforming Algorithm for Hearing Devices". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 311–315.

[84] Wenqiang Pu, Peng Zan, Jinjun Xiao, Tao Zhang, and Zhi-Quan Luo. "Evaluation of Joint Auditory Attention Decoding and Adaptive Binaural Beamforming Approach for Hearing Devices with Attention Switching". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 8728–8732.

[85] Zafar Rafii and Bryan Pardo. "Repeating pattern extraction technique (REPET): A simple method for music/voice separation". In: *IEEE transactions on audio, speech, and language processing* 21.1 (2013), pp. 73–84.

[86] Douglas A Reynolds. "Speaker identification and verification using Gaussian mixture speaker models". In: *Speech communication* 17.1-2 (1995), pp. 91–108.

[87] Scott Rickard. "The DUET blind source separation algorithm". In: *Blind Speech Separation* (2007), pp. 217–237.

[88] Scott Rickard and Ozgiir Yilmaz. "On the approximate W-disjoint orthogonality of speech". In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. vol. 1. IEEE. 2002, pp. I–529.

[89] J. Salamon, C. Jacoby, and J. P. Bello. "A Dataset and Taxonomy for Urban Sound Research". In: Nov. 2014, pp. 1041–1044.

[90] Jan Schlüter. "Learning to Pinpoint Singing Voice from Weakly Labeled Examples." In: *ISMIR*. 2016, pp. 44–50.

[91] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution". In: *53rd International Conference on Semantic Audio*. AES. 2014.

[92] Prem Seetharaman, Gordon Wichern, Shrikant Venkataramani, and Jonathan Le Roux. "Class-conditional embeddings for music source separation". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 301–305.

[93] Prem Seetharaman, Fatemeh Pishdadian, and Bryan Pardo. "Music/voice separation using the 2d fourier transform". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2017, pp. 36–40.

[94] Robert V Shannon, Fan-Gang Zeng, Vivek Kamath, John Wygonski, and Michael Ekelid. "Speech recognition with primarily temporal cues". In: *Science* 270.5234 (1995), pp. 303–304.

[95] Jianbo Shi and Jitendra Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[96] Andrew JR Simpson. "Probabilistic Binary-Mask Cocktail-Party Source Separation in a Convolutional Deep Neural Network". In: *arXiv preprint arXiv:1503.06962* (2015).

[97] Malcolm Slaney. "Pattern Playback in the '90s". In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. NIPS'94. Denver, Colorado: MIT Press, 1994, pp. 827–834.

[98] Malcolm Slaney, Daniel Naar, and RE Lyon. "Auditory model inversion for sound separation". In: *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 2. IEEE. 1994, pp. II–77.

[99] Olga Slizovskaia, Leo Kim, Gloria Haro, and Emilia Gomez. "End-to-end Sound Source Separation Conditioned on Instrument Labels". In: May 2019, pp. 306–310.

[100] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. "A scale for the measurement of the psychological magnitude pitch". In: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190.

[101] Daniel Stoller, Sebastian Ewert, and Simon Dixon. "Adversarial semi-supervised audio source separation applied to singing voice extraction". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 2391–2395.

[102] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji. "Open-Unmix - A Reference Implementation for Music Source Separation". In: *Journal of Open Source Software* (2019).

[103] Fabian-Robert Stöter, Antoine Liutkus, Roland Badeau, Bernd Edler, and Paul Magron. "Common Fate Model for Unison source Separation". In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE. 2016.

[104] Dan Stowell and Richard E Turner. "Denoising without access to clean data using a partitioned autoencoder". In: *arXiv preprint arXiv:1509.05982* (2015).

[105] Ting-Wei Su, Jen-Yu Liu, and Yi-Hsuan Yang. "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks". In: Mar. 2017, pp. 791–795.

[106] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation". In: *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE. 2018, pp. 106–110.

[107] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. "MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation". In: Sept. 2018.

[108] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. "Improving music source separation based on deep neural networks through data augmentation and network blending". In: Mar. 2017, pp. 261–265.

[109] Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. "Scream and gunshot detection and localization for audio-surveillance systems". In: *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE. 2007, pp. 21–26.

[110] Rivarol Vergin, Douglas O'shaughnessy, and Azarshid Farhat. "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition". In: *IEEE Transactions on speech and audio processing* 7.5 (1999), pp. 525–532.

[111] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. "Performance measurement in blind audio source separation". In: *IEEE transactions on audio, speech, and language processing* 14.4 (2006), pp. 1462–1469.

[112] Vibha Viswanathan, Hari M Bharadwaj, and Barbara G Shinn-Cunningham. "Electroencephalographic signatures of the neural representation of speech during selective attention". In: *eNeuro* 6.5 (2019).

[113] DeLiang Wang. "On ideal binary mask as the computational goal of auditory scene analysis". In: *Speech separation by humans and machines*. Springer, 2005, pp. 181–197.

[114] DeLiang Wang and Guy J Brown. *Computational Auditory Scene Analysis:Principles, Algorithms, and Applications*. Wiley-IEEE press, 2006.

[115] DeLiang Wang and Jitong Chen. "Supervised speech separation based on deep learning: An overview". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.10 (2018), pp. 1702–1726.

[116] Yun Wang, Juncheng Li, and Florian Metze. "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling". In: May 2019, pp. 31–35.

[117] Zhong-Qiu Wang, Jonathan Le Roux, and John R Hershey. "Alternative Objective Functions for Deep Clustering". In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018.

[118] Zhong-Qiu Wang, Jonathan Le Roux, and John R. Hershey. "Alternative Objective Functions for Deep Clustering". In: Apr. 2018.

[119] Mitchel Weintraub. "A theory and computational model of monaural auditory sound separation". PhD thesis. Ph. D. dissertation, Stanford Univ, 1985.

[120] Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller. "Discriminatively trained recurrent neural networks for single-channel speech separation". In: *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE. 2014, pp. 577–581.

[121] Felix Weninger, Jonathan Le Roux, John R. Hershey, and Björn Schuller. "Discriminatively Trained Recurrent Neural Networks for Single-Channel Speech Separation". In: *Proc. IEEE GlobalSIP Symposium on Machine Learning Applications in Speech Processing*. Dec. 2014.

[122] Felix Weninger, Florian Eyben, and Björn Schuller. "Single-channel speech separation with memory-enhanced recurrent neural networks". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 3709–3713.

[123] Gordon Wichern, Emmett McQuinn, Joe Antognini, Michael Flynn, Richard Zhu, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. "WHAM!: Extending Speech Separation to Noisy Environments". In: Sept. 2019.

[124] John F Woodruff, Bryan Pardo, and Roger B Dannenberg. "Remixing Stereo Music with Score-Informed Source Separation." In: *International Society for Music Information Retrieval conference (ISMIR)*. 2006, pp. 314–319.

[125] Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley. "Large-scale weakly supervised audio classification using gated convolutional neural network". In: Apr. 2018, pp. 121–125.

[126] Yang Yu, Wenwu Wang, and Peng Han. "Localization based stereo speech source separation using probabilistic time-frequency masking and deep neural networks". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2016.1 (2016), p. 1.

[127] Ning Zhang, Junchi Yan, and Yuchen Zhou. "Weakly supervised audio source separation via spectrum energy preserved wasserstein learning". In: *arXiv preprint arXiv:1711.04121* (2017).

[128] Zoran Zivkovic. “Improved adaptive Gaussian mixture model for background subtraction”. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on.* Vol. 2. IEEE. 2004, pp. 28–31.