Interference Reduction in Microphone Recordings for Music Source Separation

Submitted in partial fulfillment of the requirements
for the degree of

Master of Science
(by Research)

by

Rajesh R
(S21005)

Under the supervision of

Dr. Padmanabhan Rajan

School of Computing and Electrical Engineering
Indian Institute of Technology Mandi
Himachal Pradesh, India - 175005
April 2024

dedicated to my grandmother

DECLARATION BY THE SCHOLAR

I hereby declare that this work incorporated in this thesis is the outcome of the studies accomplished by me in the **School of Computing and Electrical Engineering, Indian Institute of Technology Mandi**, under the supervision of **Dr. Padmanabhan Rajan**. This work has not been submitted elsewhere for any degree or diploma. In keeping with the general practice, due acknowledgements have been made wherever the work described is based on the findings of other investigators. In addition, I certify that no part of this work will, in future, be used for submission in my name, for the award of any other degree at any university.

Rajesh R

CERTIFICATE

This is to certify that the thesis titled **“Interference Reduction in Microphone Recordings for Music Source Separation”**, submitted by **Rajesh R**, at Indian Institute of Technology Mandi for the award of Master of Science (by research) is a bonafide record of the research work carried out by him under my supervision. The content of this thesis, in full or in parts, has not been submitted to any other institute or university for the award of any degree or diploma.

Dr. Padmanabhan Rajan
Supervisor
School of Computing and Electrical Engineering
Indian Institute of Technology Mandi

Place: Indian Institute of Technology Mandi
Date: July 1, 2024

Acknowledgement

I would like to express my deep and heartfelt gratitude to my respected guide and mentor, Dr. Padmanabhan Rajan, for his unwavering blessings, valuable advice, support, good wishes, and encouragement throughout this journey. Without his steadfast support, this work would not have been possible. His thoughts and unwavering support have been unmatched, and I will always cherish our discussions and his guidance.

My sincere thanks also go to Dr. Dileep A.D., whose deep learning courses were unparalleled and laid a strong foundation for me. Serving as his Teaching Assistant helped me gain a deeper understanding of the subject. Additionally, I am grateful to Dr. Siddhartha Sarma for his assistance at various stages of my research, especially in the field of optimization. Without their guidance, this work would not have been possible.

I am also grateful to the members of my Academic Progress Committee, Dr. Sreelakshmi Manjunath, Dr. Aditya Nigam, and Dr. Gargee Sharma, for their invaluable feedback and reviews of my work.

I extend my thanks to my fellow labmates in my group: Akansha Tyagi, who has been both a senior and a sister to me, for her care, support, and blessings throughout my master's journey. I also appreciate the guidance and support from Shilpa Chandra not only in the lab but outside and Shubham Sarma, whose stimulating discussions and assistance during deadlines have been invaluable. Additionally, I thank my friends at IIT Mandi: Ritwik Ghosal, Akumalla Ravi Kiran, Gopal Sai Krishna, Surya Prakash, Akalya S, Akash K, and Vikram Kumar. In particular, I am grateful to Naisarg Pandya, Manoj Chowdary, and Utkarsh Yadav for making my days beautiful during my studies.

Special thanks are extended to my favorite people, Anand Babu and Nandini Priya, for their constant support, boundless love, and unwavering belief in me.

Last but not least, I would like to thank my beloved brother, Bhuvan Kumar, without whom this journey would not have been possible. I am thankful for all the sacrifices he made to support my research endeavors. I am also indebted to my family: my parents and my sister, Nathiya Deepak, for their unwavering spiritual support throughout my life.

Rajesh R

Abstract

Music Source Separation (MSS) is fundamental to various music information retrieval tasks, including pitch estimation, genre detection, and instrument classification. Its primary function is to extract distinct instrument sounds, or sources, from musical audio. However, the MSS problem remains less explored in the context of various other genres, such as Indian classical music, compared to the detailed studies in Western pop music. The objective here is to construct a dedicated model proficient at effectively isolating specific musical sources within a given composition, irrespective of the genre.

To achieve this, obtaining a high-quality dataset is essential but challenging. For example, as a test case, live concert recordings from Indian classical performances form a valuable data source but frequently suffer from issues of acoustic bleeding and interference due to a lack of acoustic shielding. Thus, the primary objective of this thesis is to develop an interference reduction system for microphone recordings.

To address the interference reduction issue, we introduce several interference reduction techniques to enhance the dataset's suitability for training MSS models. These techniques include a learning-free optimization approach and learning-based convolutional autoencoders (CAEs), truncated Unet (t-UNet), and graph-based interference reduction network (GIRNet). A dedicated CAE was used for each source, treating interference as noise. However, CAEs work in the time-frequency domain, accepting short-time Fourier transform (STFT) magnitude input and outputting an estimated clean STFT magnitude source. The t-UNet and the GIRNet, in turn, work with the raw waveform, learning the relationships among the various sources and using that information to reduce interference.

The proposed techniques have reduced interference and improved source-to-distortion ratios. Subsequently, we utilized the Wave-U-Net MSS model to effectively separate the stems in a Carnatic music dataset as a test case.

Keywords: Music Source Separation, Interference Reduction, Wave-U-Net, Graph Attention Networks

List of Tables

List of Figures

List of Notations and Operations

n	Number of sources
k	Number of microphones
Λ	Interference Matrix
$x(t)$	Microphone signal with length l
$s(t)$	True source signal with length l
$m(t)$	Mixture signal with length l
$\hat{s}(t)$	Estimated source signal from microphone recording $x(t)$
X	The matrix is of size $k \times l$, with microphone signals in each row.
S	The matrix is of size $n \times l$, with source signals in each row.
$X(f,t)$	STFT of the signal $x(t)$
$S(f,t)$	STFT of the signal $s(t)$

Abbreviations

A	
AI	Artificial Intelligence
AE	Autoencoder
AQ	Audio Quality
ADMM	Alternating Direction Method of Multipliers
B	
BSS	Blind Source Separation
C	
COBE	Common Orthogonal Basis Extraction
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
D	
DL	Deep Learning
DNN	Deep Neural Network
DAW	Digital Audio Workstation
DSP	Digital Signal Processing
G	
GNN	Graph Neural Network
GCN	Graph Convolutional Neural Network
GAT	Graph Attention Neural Network
GIRNet	Graph-based Interference Reduction Network
I	
ICA	Independent Component Analysis
IVA	Independent Vector Analysis
IR	Interference Reduction
IRQ	Interference Reduction Quality
ISS	Informed Source Separation
ICM	Indian Classical Music
K	
KAMIR	Kernel Additive Modelling for Interference Reduction
KAM	Kernel Additive Modelling
L	
LSTM	Long Short Term Memory
LGM	Local Gaussian Model
M	
MIRA	Multitrack Interference Reduction Algorithm
MSS	Music Source Separation

MIR	Music Information Retrieval
MMSE	Minimum Mean Squared Error
N	
NMF	Non-Negative Matrix Factorisation
P	
PSD	Power Spectral Density
R	
RPCA	Robust Principal Component Analysis
RNN	Recurrent Neural Network
S	
STFT	Short Time Fourier Transform
SS	Source Separation
SDR	Source to Distortion Ratio
SI-SDR	Scale Invariant Source to Distortion Ratio
SIR	Source to Interference Ratio
SAR	Source to Artifacts Ratio
T	
TF	Time-Frequency
U	
USS	Universal Source Separation
V	
VAE	Variational Autoencoder

Contents

CHAPTER 1

Introduction

Chapter 1 delves into the problem of music source separation and interference reduction, emphasizing its far-reaching impact on various music information retrieval tasks. This initial chapter establishes the foundation of music source separation and interference reduction, while also illuminating the key distinctions between them. Furthermore, it delves into the inherent challenges associated with building robust music source separation and interference reduction systems. Finally, to ensure clarity and coherence, the chapter outlines the specific objectives, scope, and contributions of the thesis, along with its organizational structure.

1.1 Music Source Separation

As we engage with this thesis, our surroundings are filled with a myriad of sounds, each distinct and recognizable to the human ear. Consider, for instance, the ambient noise in a computer lab: the whirring of fans, the hum of inverters, and the murmur of multiple conversations, both within and outside the room. Amidst this auditory tapestry, our brains effortlessly discern each sound. But have you ever paused to ponder how the brain accomplishes this remarkable feat? How does it unravel the amalgamation of sounds, separating them into distinct sources? And more intriguingly, can we replicate this process in machines?

Imagine ourselves amidst a lively party, where multiple conversations intermingle with background music. Yet, amidst this auditory complexity, individuals effortlessly tune in to specific voices, enabling seamless communication. This phenomenon, known as the cocktail party problem [7], is a classical example of Blind Source Separation (BSS) [8]. Blind source separation is a universal concept that involves the decomposition of multiple signals, whether they originate from distinct speakers or instruments.

Blind Source Separation has various flavours. It could involve isolating individual speakers in a conversation, extracting singing voices from background music, or segregating the distinct sounds of musical instruments such as vocals, bass, drums, and more.

Unlike natural speech, creation of music offers endless possibilities for composition, with different genres characterized by unique blends of instrument sounds. For instance, in popular Western pop music, vocals, bass, drums, guitar, and keyboards are commonly prominent elements, referred to as *sources* or *stems*. Similarly, in South Indian classical Carnatic music, compositions typically feature vocals, mridangam, and violin as primary sources. The process of extracting these sources from a composite audio mixture is termed music source separation (MSS) [3].

In the literature on music source separation, we encounter different types of MSS that are closely related:

Informed Source Separation [9] involves extracting sources armed with prior knowledge, such as insights into the mixing process, spatial information, and awareness of the number and types of sources present. This method empowers us to disentangle audio mixtures with a deeper understanding of their composition.

Music Source Separation, on the other hand, is tasked with extracting various sources when a predetermined number is already known. Armed with this knowledge of the number of sources and their instrument types, this approach dissects audio mixtures with precision, isolating each component with finesse. Here the mixing process and spatial information are not known.

Universal Source Separation [10] pushes boundaries by endeavouring to extract all available sources within a given context. Whether it's discerning every speaker in a conversation or uncovering all instrument types within a musical piece, this approach seeks comprehensive extraction, leaving no sonic stone unturned.

In contrast, **Blind Source Separation** operates in the absence of prior information. Without knowledge of the number of sources, instrument types, or the mixing process, this method embarks on the formidable task of extracting various sources purely from the raw audio mixture, relying solely on signal patterns and characteristics.

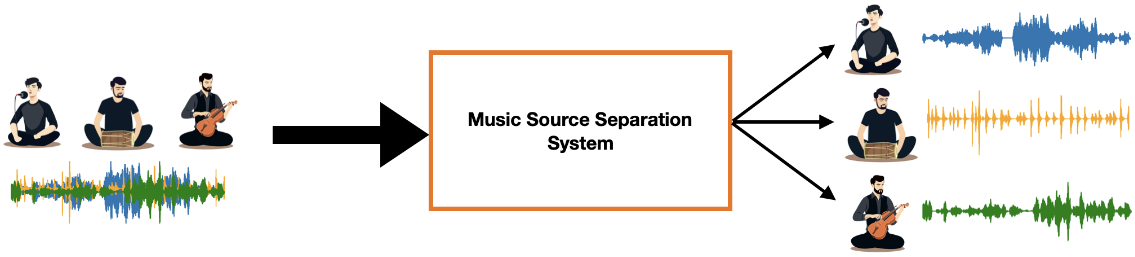


Figure 1.1: Music source separation System

The objective of this thesis is to develop a music source separation (MSS) system specifically tailored to effectively extract various sources present in Indian classical music as shown in Figure 1.1, with a particular focus on Carnatic compositions. While considerable research contributions have been made in the field of Music Information Retrieval (MIR) tasks within Western music, the domain of Carnatic Indian classical music remains relatively unexplored, highlighting the need for targeted advancements in music source separation solutions for this genre.

1.1.1 Why is Source Separation Needed?

We can try to motivate why we need music source separation systems in the first place and why it's crucial to incorporate this intelligence into machines. MSS serves a variety of applications across diverse domains [3, 11]. Primarily, it empowers music production by allowing remixing, remastering, and refining audio tracks through the isolation of vocals, instruments, or specific sounds, offering better control over the mix. Moreover, it enhances the listening experience by enabling users to customize their auditory engagement, isolating or amplifying particular instruments or vocals within songs. This technology aids in transcription and analysis, facilitating the study and transcription of individual instrument parts, contributing to music analysis and academic research. Additionally, it plays a pivotal role in audio restoration, salvaging old or impaired audio recordings by segregating unwanted noise or interference from the desired audio. Further applications span live performances, movies & entertainment industry, karaoke, MIR, education, audio coding, sound design, and even speech enhancement, showcasing its versatility and significance across entertainment, education, research, and technological advancements.

1.2 Interference Reduction

Significant research contributions in Music Information Retrieval (MIR) tasks within Western music have been made, largely due to the surge in music source separation (MSS) solutions tailored for this genre. This progress can be attributed to the availability of clean datasets. Typically, the sources are recorded in a studio environment, often in acoustically shielded rooms, and mixed offline using Digital Audio Workstations (DAWs) [3].

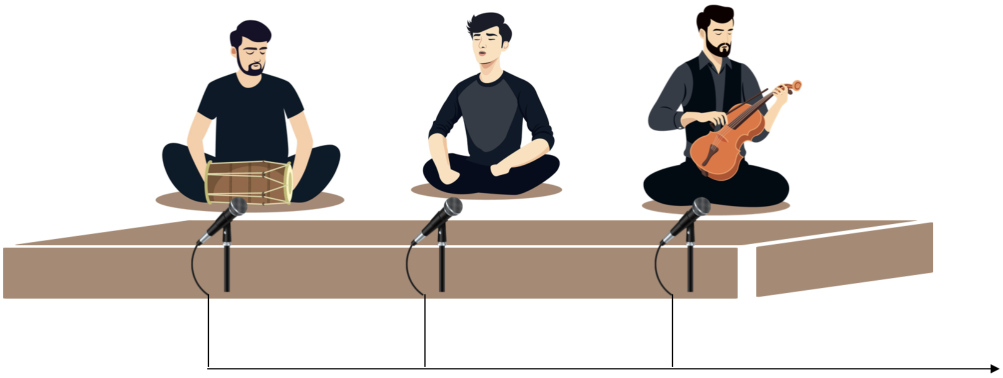


Figure 1.2: Typical setup in Carnatic concerts

However, the advancement of music source separation (MSS) in Western pop music does not extend to Carnatic Indian classical music. Despite the rich potential for exploration within this genre, the scarcity of available datasets poses a significant challenge to research and model training. Unlike Western pop, Carnatic Indian classical music is predominantly performed live rather than being recorded in studios. Performances often occur in non-acoustically treated rooms filled with audiences. Dedicated microphones are placed strategically to capture each source, as depicted in an example setup of a Carnatic concert in Figure 1.2. However, due to the lack of acoustic shielding and the close proximity of sources, the microphones intended for specific sources inadvertently capture other sources as well, as illustrated in Figure 1.3. These phenomena are commonly referred to as *interference, bleeding, leakage, crosstalk*, or *spill.* If these recordings can be cleaned to reduce interference, they could provide valuable data for training supervised source separation models.

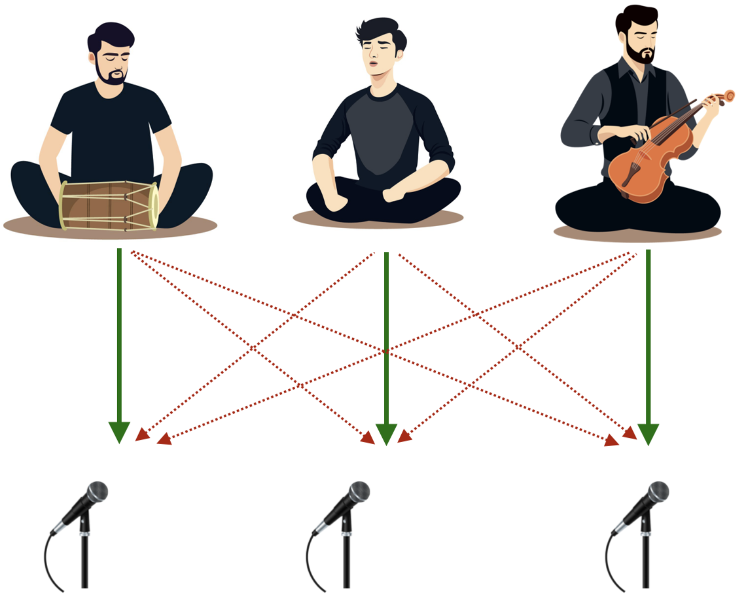


Figure 1.3: Interference effects in a live Carnatic concert

The primary objective of the thesis is to propose techniques for interference reduction in each source recorded in live settings. The availability of clean, interference-free multi-tracks would facilitate the training of large, complex, dedicated music source separation (MSS) models. Therefore, the system's goal, when given microphone recordings as input, is to minimize interference among the live microphone recordings and produce clean, interference-free versions of them.

The primary focus of this thesis is to develop interference reduction systems for live recordings

1.2.1 Interference Reduction Vs Music Source Separation

The objective of the interference reduction system is to produce clean, interference-reduced recordings. One application is to use these recordings to train source separation models. However, isn't the interference reduction problem itself a source separation problem? Indeed, the interference reduction problem can be viewed as a special case of a source separation problem.

To build supervised deep learning-based music source separation or interference reduction models, we require individual sources and their mixtures. These individual sources are live recordings in interference reduction systems.

In general, music source separation (MSS) aims to extract various sources from a given mixture. In interference reduction, on the other hand, the goal is to retain the dominant source from its corresponding microphone recording. Interference reduction is different and perceived as less complex than MSS due to the availability of information (each source's information is available in each microphone recording as interference). We have all the microphone recordings, which can be utilized to build the model. This interference reduction model can also be directly applied to source separation if the aim is to extract only the dominant source in the corresponding microphone recording.

As illustrated in Figure 1.4, we can distinguish between the interference reduction and source separation systems.

1.2.2 Challenges

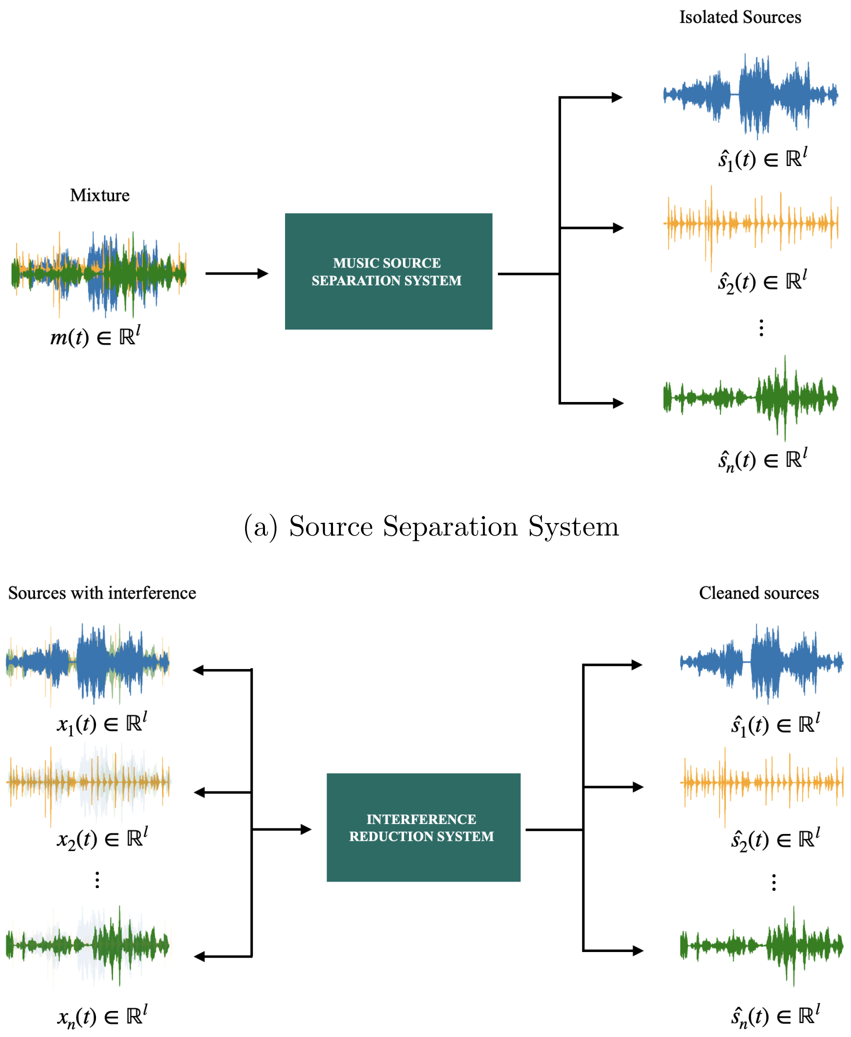
There are several challenges involved in building an interference reduction system, whether using learning-free models or neural network approaches.

- If iterative signal processing-based algorithms are constructed, they should be designed to be faster, memory-constrained, and more effective, given that iterative algorithms are generally slower and unsuitable for lengthy real-world recordings.

- Generally, neural networks outperform traditional signal processing algorithms, but again, the challenge is to obtain data for training. Thus, obtaining a dataset with a mixture and its constituent source recordings is challenging.

- In the case of a supervised model, the system should be capable of handling any type of input source. It should be instrument or source-independent and work perfectly for domain invariances.

These challenges are universal in building interference reduction systems. This thesis addresses the problem by developing optimization learning-free algorithms, convolutional autoencoder (CAEs) models, truncated UNet (tUNet), and Graph-based interference reduction network (GIRNet).


Isolated Sources
Mixture
$m(t) \in \mathbb{R}^l$
MUSIC SOURCE SEPARATION SYSTEM
$\hat{s}_1(t) \in \mathbb{R}^l$
$\hat{s}_2(t) \in \mathbb{R}^l$
$\hat{s}_n(t) \in \mathbb{R}^l$
Sources with interference
Cleaned sources
$x_1(t) \in \mathbb{R}^l$
$x_2(t) \in \mathbb{R}^l$
$x_n(t) \in \mathbb{R}^l$
INTERFERENCE REDUCTION SYSTEM
$\hat{s}_1(t) \in \mathbb{R}^l$
$\hat{s}_2(t) \in \mathbb{R}^l$
$\hat{s}_n(t) \in \mathbb{R}^l$

(a) Source Separation System

(b) Interference Reduction System

Figure 1.4: The key difference between source separation and interference reduction systems.

1.3 Objectives and Scope of the Thesis

As described, there is a notable research gap in various genres except for Western pop music due to its wide range of available datasets. This thesis focuses on such domains, for example, Indian classical music, specifically Carnatic music, to address the limitations inherent in the available datasets. Music source separation is a crucial task for enabling further advancements in information retrieval within this context. Unfortunately, existing datasets are plagued by significant interference among their sources, rendering them unsuitable for constructing supervised systems effectively.

To address this challenge, the thesis focuses on developing models capable of reducing interference among the stems. By doing so, we aim to produce cleaned, interference-reduced stems that can serve as valuable training data for the development of more complex music source separation models.

In the case of Indian classical music, particularly Carnatic music, the MSS system should effectively isolate different instruments such as vocals, mridangam, and violin. To achieve this, the interference reduction system must minimize the bleed in these sources using live recordings of Carnatic music, allowing them to be used for training MSS models.

1.4 Thesis Contributions

The contributions of this thesis are summarized as follows:

- A dedicated Convolutional Autoencoder-based interference reduction model for each source [Chapter 3]. In the literature on interference reduction, only DSP-based algorithms have been proposed, and to our knowledge, no neural network methods were available. Therefore, building the CAE will help accelerate the interference reduction process compared to those iterative DSP-based algorithms.
- Learning-free optimization algorithm for interference reduction, which accepts multiple microphone recording inputs and iteratively estimates the cleaned interference-free recording and its associated strength in the form of an interference matrix, unlike the proposed CAE [Chapter 3].
- Extending the idea of the optimization algorithm with a neural network called the truncated-UNet (t-UNet) for interference reduction. [Chapter 4]
- Extending t-UNet for nonlinear mixtures and live recordings by using GIRNet (graph attention-based full WaveUNet) for interference reduction. [Chapter 4]

1.5 Organisation of the Thesis

- **Chapter 1: Introduction** introduces the music source separation problem and interference reduction in live recordings.
- **Chapter 2: Background** briefly introduces the music representation and properties, mixing process, datasets, and the evaluation metrics used. Also, the chapter presents state-of-the-art techniques relevant to music source separation for MUSDB18HQ dataset and interference reduction in live recordings.

- **Chapter 3: Linear Mixing Models for Interference Reduction** discusses various methods proposed for interference reduction. It explores both the learning-free optimization approach and learning-based models, such as convolutional autoencoders (CAEs), complex CAEs, and truncated UNets (tUNet). Additionally, the chapter discusses how these methods are applied to enhance music source separation.
- **Chapter 4: Non-Linear Mixing Models for Interference Reduction** extends the discussion from Chapter 3 to live recordings, introducing the dilated full wave-u-net and the dilated wave-u-net with graph attention. The chapter also includes experiments on source separation models.
- Finally, **Chapter 5: Conclusion** concludes the thesis work and suggests future directions for the proposed methodologies.

CHAPTER 2

Background

This chapter provides a brief introduction to the background for the thesis. Topics covered include the structure of music signals, music properties and representations, datasets, mixing processes, and evaluation metrics. It then delves into the literature on music source separation models and interference reduction models for Western pop music. Detailed explanations of several state-of-the-art models are provided, along with brief overviews of others. Additionally, insights into the networks proposed specifically for source separation to date are offered, along with a discussion on the direction in which the literature is progressing. Furthermore, the chapter touches briefly upon the latest generative AI models and introduces the new paradigm of Language-Queried Audio Source Separation (LQASS).

The fundamental question that arises from the outset is why there is a need for different source separation systems for speaker source separation, singing voice and background music separation, and music source separation. This arises due to the distinct features and properties of each audio type. Audios can be commonly classified into three categories: speech, where humans speak; music, where there is no speech involved and only singing voice and compositions of various musical instruments are present; and sound, which comprises all other environmental sounds such as rain, claps, and noise.

To build music source separation model, one needs to understand the basic differences between music and speech signals, as well as properties of music and their representations.

2.1 Speech Vs Music

Speech and music signals exhibit distinct characteristics stemming from their intended communication purposes and the types of sound sources involved. The difference in the spectrogram of music and speech is shown in Figure 2.1.

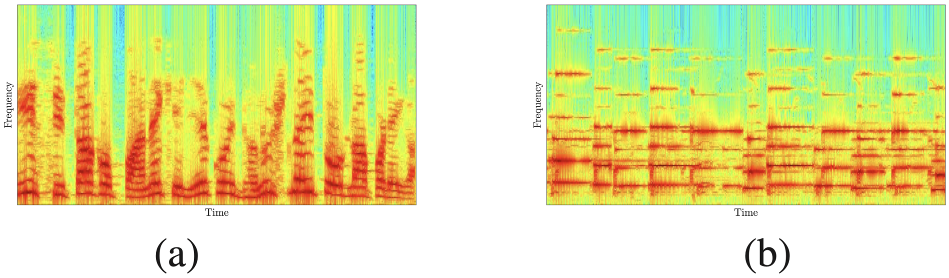


Figure 2.1: A spectrogram example of (a) speech and (b) music.

The music spectrogram appears to have different structure than the speech spectrogram, as observed in Figure 2.1. Additionally, it exhibits certain patterns that are distinct from speech spectrograms due to the involvement of musical instruments. To provide an understanding of the fundamental differences, consider the following points:

Speech Signals:

- Speech signals primarily convey linguistic information through the modulation of vocal tract resonances and articulatory movements.
- Spectrally, speech signals often contain formant structures representing the resonant frequencies of the vocal tract, which contribute to vowel sounds' perceptual qualities.
- Temporally, speech signals exhibit rapid variations corresponding to phoneme transitions, pauses, and prosodic features such as intonation and stress.

- Speech signals are characterized by relatively predictable patterns of spectral and temporal evolution due to the regularities in language structure and articulation.

Music Signals:

- Music signals encompass a wide variety of sounds produced by musical instruments, each with unique timbral qualities and harmonic content.
- Spectrally, music signals often exhibit complex harmonic structures resulting from the simultaneous vibration of multiple overtones and harmonics produced by musical instruments.
- Temporally, music signals can vary widely in rhythmic complexity, tempo, and dynamics, reflecting the expressive intentions of the performers and composers.
- Music signals may contain percussive elements, sustained notes, melodic lines, harmonic progressions, and singing voice, leading to rich and diverse spectro-temporal patterns.

These differences necessitate the development of distinct source separation systems for speech and music. In music, instruments often occupy the same time-frequency (TF) bands, leading to more overlap compared to speech separation [3]. This increased overlap poses greater difficulty for music source separation. The next section outlines several fundamental distinctions between speech and music source separation systems.

2.1.1 Music Vs Speech Source Separation:

Given the different properties and characteristics of music and speech signals, their source separation systems also involve some key differences. They are based on the following:

Complex Spectral and Temporal Interactions:

Music source separation faces challenges due to the complex interactions between harmonic, percussive, and melodic components present in music signals. Harmonic overlap between different instruments and harmonic-rich sounds can make it challenging to separate individual sources accurately, particularly in polyphonic music recordings.

Temporal and Frequency Masking:

Temporal and frequency masking phenomena occur when the presence of one sound source obscures or masks the perception of another source in the mixture. Percussive sounds and transient events can mask harmonic components or melodic lines, complicating the separation process, especially when multiple sources overlap in time and frequency.

Reverberation and Spatial Effects:

The presence of reverberation and spatial effects in music recordings introduces additional challenges, as they create acoustic reflections and spatial colouration in the signals. Spatial effects such as stereo panning and spatial positioning can affect the perceived localization and separation of sound sources in the mixture.

Speech Overlapping:

Multiple speech source separation involves dealing with overlapping speech segments, where different speakers' utterances occur simultaneously in the mixture. Overlapping speech segments can result in spectral and temporal interference, making it challenging to isolate individual speakers' contributions accurately.

Speaker Variability:

Variability in speakers' vocal characteristics, including pitch, timbre, and speaking rate, poses challenges for separating multiple speech sources, particularly when speakers exhibit similar acoustic profiles.

Background Noise and Reverberation:

Multiple speech separation tasks often involve dealing with background noise and reverberation, which can degrade speech intelligibility and complicate source separation. Noise and reverberation effects introduce additional interference and mask speech signals' spectral and temporal features, making it challenging to extract clean speech sources.

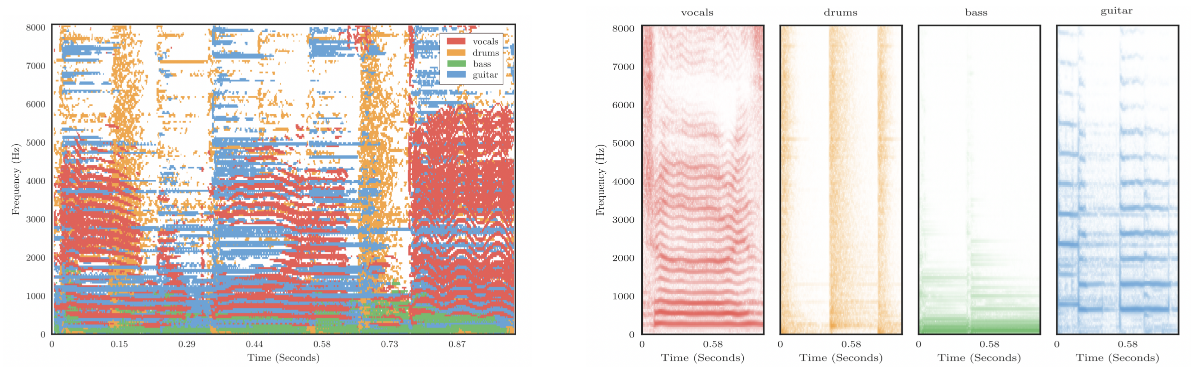
In summary, while both music and speech source separation tasks involve separating multiple sound sources from a mixture, they face distinct challenges stemming from the unique characteristics of speech and music signals. Music source separation encounters complexities related to harmonic structure, temporal interactions, and spatial effects, whereas multiple speech separation must contend with speech overlapping, speaker variability, and background noise.

To visualize the complexity of the MSS problem, a mixture spectrogram of an example is shown in Figure 2.2 (a), taken from the article [3]. Each source is represented with a unique colour. There are four sources: vocals, bass, drums, and others. Their respective spectrograms are shown in Figure 2.2 (b). At certain TF bins, a high overlap of instrument sounds is visible.

Additionally, the harmonic structure of the vocals and violin is visible from the figure, while the percussive instruments exhibit vertical line patterns in the spectrogram, as seen in the drums.

2.1.2 Music Properties and Representations

Understanding why audio types such as speech, sound, and music behave differently is crucial for comprehending the underlying problem. This understanding allows us to discern musical patterns and properties, which are essential for effectively addressing the source separation challenge.



(a) Mixture

(b) Its individual sources

Figure 2.2: Example of a mixture spectrogram and its sources, taken from [3]

Music Properties

The properties of the music are widely classified into spatial, temporal, and spectral properties [12, 13].

Spatial Properties: Music exhibits a diverse range of spatial characteristics that significantly influence the perceived spatial distribution and localization of sound sources within an auditory scene. Stereo and multichannel recordings capture spatial cues such as stereo panning, spatial imaging, and spatialization effects, which contribute to the perceived width, depth, and envelopment of the soundstage. During the production and mixing stages, spatial effects in music recordings can be manipulated to create a sense of spaciousness, immersion, and spatial separation between different instruments and sound sources, enhancing the overall spatial experience for listeners.

Temporal Properties: Temporal characteristics play a crucial role in shaping the rhythmic patterns, tempo variations, and timing nuances inherent in musical performances. Rhythmic elements such as beat, meter, and rhythmic accents establish the underlying pulse and rhythmic feel of a musical piece, influencing listeners' perception of timing and groove. Variations in tempo, articulation, and phrasing introduce dynamic changes in musical dynamics, expressive timing, and rhythmic complexity, adding depth and dimension to the temporal dynamics and emotional impact of the music.

Spectral Properties: Spectral characteristics define the frequency content, harmonic structure, and timbral qualities of individual musical sounds and instruments. Harmonic components in music signals arise from the fundamental frequencies and harmonic overtones generated by musical instruments, contributing to the richness and complexity of the sound spectrum. The timbral qualities of musical sounds, including brightness, warmth, and richness, are shaped by the spectral shape, frequency distribution, and amplitude envelope, reflecting the distinctive acoustic properties of different instruments and sound sources.

Now that we have a basic understanding of music properties, it's essential to consider which representation would be most beneficial for capturing relevant information. Particularly for MSS, the choice of representation for a mixture audio signal is crucial in decomposing it into various sources. To gain insight into this aspect, let's explore some frequently used music representations.

Music Representations

Music representations are fundamental for analyzing, processing, and synthesizing music signals in various audio applications [13, 14, 15, 16, 17]. These representations capture different aspects of musical content, including its temporal, spectral, and timbral characteristics. Some common music representations include:

Time-domain Representation: Time-domain representations encode music signals as amplitude variations over time. This raw waveform representation captures the instantaneous amplitude of the audio signal at each time sample. While time-domain representations provide direct access to the raw audio data, they are often less informative for analyzing frequency content and spectral characteristics.

Frequency-domain Representation: Frequency-domain representations analyze the frequency content of music signals by decomposing them into their constituent frequency components. Techniques such as Fourier analysis, Short-Time Fourier Transform (STFT), and Spectrogram convert time-domain signals into frequency-domain representations. Spectrograms provide a visual depiction of the frequency content of music signals over time, highlighting variations in spectral energy and harmonic structure.

Pitch-based Representation: Pitch-based representations focus on extracting pitch-related information from music signals, including fundamental frequencies (F0) and pitch contours. Pitch detection algorithms analyze the periodicity and pitch periodicity of audio signals to estimate the dominant pitch and pitch variations over time. Pitch-based representations are essential for tasks such as melody extraction, pitch tracking, and harmonic analysis in music signals.

Timbral Representation: Timbral representations capture the timbral qualities and spectral characteristics of musical sounds, including brightness, warmth, and richness. Techniques such as cepstral analysis, Mel-frequency cepstral coefficients (MFCCs), and perceptual audio features extract timbral descriptors from music signals. Timbral representations are valuable for tasks such as sound classification, instrument recognition, and timbre-based synthesis.

Symbolic Representation: Symbolic representations encode musical content in symbolic or discrete form, such as musical notation, MIDI (Musical Instrument Digital Interface), or symbolic music formats. These representations represent musical elements as symbolic events, including notes, chords, rhythms, and musical structures. Symbolic representations facilitate music composition, arrangement, and analysis tasks, enabling interoperability between different music software and systems.

Hybrid Representations: Hybrid representations combine multiple modalities or levels of representation to capture diverse aspects of musical content comprehensively. For example, audio features extracted from spectrograms or time-frequency representations can be combined with symbolic music data to enrich music analysis and understanding. Hybrid representations leverage the complementary strengths of different representation modalities for more robust and versatile music-processing tasks.

2.2 Datasets

Now that we've covered some basics of music properties and representation, let's delve into datasets for MSS tasks. As mentioned earlier, MSS for Western pop music has shown remarkable performance, largely due to the availability of clean and extensive datasets. There are numerous datasets tailored for training MSS models specifically for Western pop music in a supervised fashion. These datasets typically comprise mixture signals along with individual clean sources present in the mixture. Some widely used datasets include MASS [18], MIR-1K [19], QUASI [20], MelodyDB [21], iKala [22], DSD100 [23], Slakh100 [24], and the renowned MUSDB18 dataset [4]. An overview of these datasets has been described in Table 2.1. Among these datasets, the MUSDB18 dataset holds particular popularity within the research community.

Dataset	Year	Instrument	Tracks	Avgerage duration (s)	Full songs	Stereo
MASS [18]	2008	N/A	9	16 ± 7	×	✓
MIR-1K [19]	2010	N/A	1,000	8 ± 8	×	×
QUASI [20]	2011	N/A	5	206 ± 21	✓	✓
ccMixter	2014	N/A	50	231 ± 77	✓	✓
MedleyDB [21]	2014	82	63	206 ± 121	✓	✓
iKala [22]	2015	2	206	30	×	×
DSD100 [23]	2015	4	100	251 ± 60	✓	✓
MUSDB18 [4]	2017	4	150	236 ± 95	✓	✓
Slakh2100 [24]	2019	34	2100	249	✓	×

Table 2.1: Overview of the MSS datasets, taken from [1].

2.2.1 MUSDB18HQ

The MUSDB18 dataset comprises 150 full-length music tracks spanning approximately 10 hours in total, encompassing Western pop genres. These tracks are accompanied by isolated stems, including drums, bass, vocals, and other components, as shown in Figure. The dataset is organized into two main folders: “train,” containing 100 songs for training, and “test,” comprising 50 songs for evaluation. All audio signals are in stereo and sampled at 44.1kHz. MUSDB18 is intended for academic use only, with many of its tracks licensed under a Creative Commons Non-Commercial Share Alike license (BY-NC-SA).

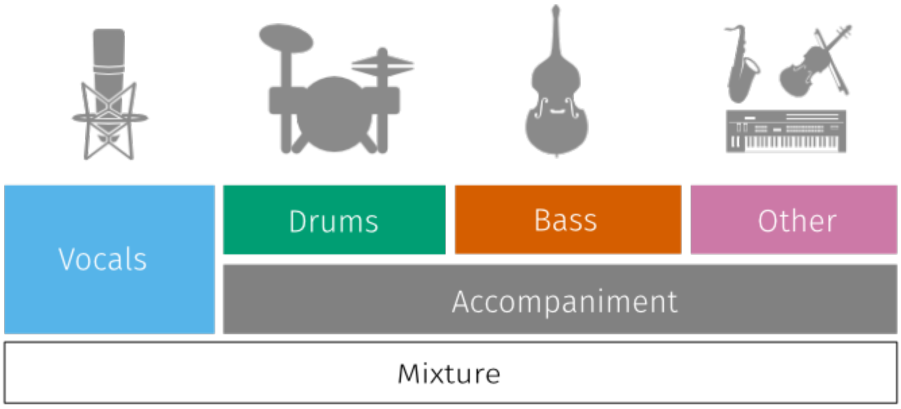


Figure 2.3: MUSDB18 Dataset. Figure adapted from [4].

Each song in the MUSDB18 dataset is encoded in the Native Instruments stems format (.mp4), which consists of five stereo streams. These streams correspond to different components of the music, namely the mixture, drums, bass, accompaniment, and vocals. The mixture signal is the summation of all the individual stems.

Additionally, uncompressed WAV files are provided as an alternative for models aiming to predict high bandwidths of up to 22 kHz. These WAV files are available in a separate version of the dataset known as MUSDB18-HQ [25], which is otherwise identical to the standard MUSDB18 dataset.

Given the availability of complete tracks within the MUSDB18HQ dataset, artificial mixing can be achieved in numerous ways, rather than simple summation. In this thesis, we have employed various artificial mixtures for our experiments. Details are provided in their respective chapters and sections.

2.2.2 Saraga

The Saraga dataset [26] is a valuable resource in the field of music information retrieval (MIR) and audio signal processing, particularly for tasks related to Indian classical music (ICM). It contains genres from both Hindustani and Carnatic music traditions. Serving as a comprehensive collection of audio recordings and associated annotations, it offers researchers and practitioners a rich dataset for various applications in the field.

Comprising recordings of ICM performances, the Saraga dataset covers a wide range of musical styles, artists, and compositions, providing a diverse and representative sample of ICM traditions. The dataset includes recordings of vocal, instrumental, and ensemble performances across different genres and regions of Indian classical music.

For source separation purposes, the dataset comprises 164 songs, each containing a mixture and its corresponding sources. The multitrack includes primary vocals, secondary vocals, left and right channels of the mridangam[1], violin, and ghatam[2]. However, only four examples feature the ghatam. These examples vary in length, ranging from one minute to two hours.

Limitations: While the dataset provides multitrack recordings alongside the mixture, it may not be directly suitable for training source separation models. The recordings were captured during live sabhas, which introduce challenges like reverberation, interference, etc. Due to the lack of acoustic shielding and the non-studio recording environment, the recordings suffer from significant bleeding effects or interference among the sources, compounded by other non-linear effects.

2.3 Mixing Process Used In The Thesis

The datasets themselves can be classified into various types. One important classification in source separation is whether the recordings are live microphone recordings or artificial mixtures. The MUSDB18HQ dataset consists of artificial mixtures, while the Saraga dataset comprises live microphone recordings. In this section, we will study the differences between live microphone recordings and artificial mixtures.

2.3.1 Live Microphone Recordings Vs Artificial Mixtures

According to [27], live microphone recordings are the audios captured when multiple sources, which are simultaneously active, are captured by a microphone array. Some examples include

[1] Mridangam is the percussive instrument largely used in Carnatic music.

[2] A round pot struck with the hands, used as a percussion instrument in Carnatic music.

recordings of live concerts, audio conferencing, hearing aids, and hands-free phones. On the other hand, artificial mixtures are generated by mixing individually recorded sound sources using appropriate hardware or software. Examples include audio media such as cinema, music, and television.

Ideally, different rooms have different acoustic impulse responses, which contribute to time delays, reverberations, and other acoustic characteristics. The reverberation time (RT60) varies depending on the environment. For instance, the RT60 will differ in environments such as a car, office, meeting room, living room, kitchen, bedroom, classroom, restaurant, and marriage hall. An acoustically treated room will be less affected by these effects, reducing nonlinearities when mixed using DAW or any other mixing process, which produces artificial mixtures.

2.3.2 Artificial Mixtures

Artificial mixtures are created using hardware or software after individual sources are recorded. According to [2], the process of creating artificial mixtures typically involves four steps. Firstly, the sound engineer applies a series of effects to each source independently. Secondly, the sources are transformed into multichannel spatial images. In the third step, the spatial images of all the sources are combined or summed to obtain what is known as the mixture. Finally, additional effects may be added to the resulting mixture.

In this thesis, an artificial mixing we assume it adheres to the convolutive signal model (see Figure 2.4) to match real-world characteristics, such as room impulse response, reverberation, time delays, and non-linear mixes.

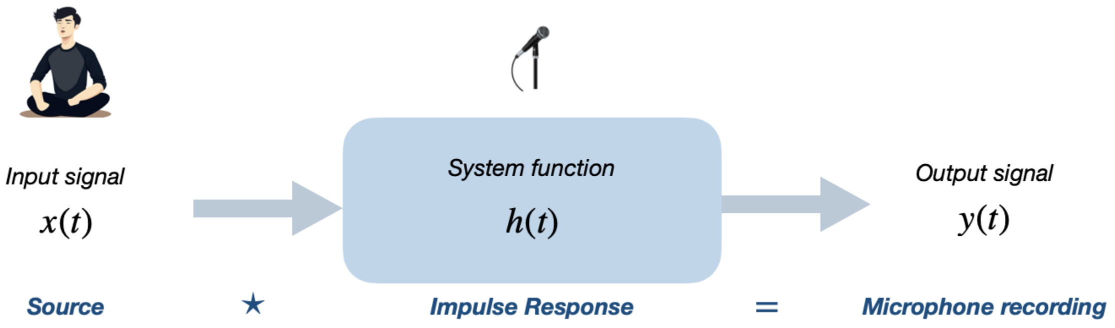


Figure 2.4: Signal Model

Some examples of artificial mixing effects has been shown in Table 2.2:

Linear instantaneous effects	Gain
	Panning (Instantaneous mixing)
Linear convolute effects	Equalization
	Reverberation
	Delay
Nonlinear effects	Dynamic Compression
	Chorus
	Distortion

Table 2.2: Example artificial mixtures taken from [2]

2.4 Evaluation Metrics

We have established a background in music, datasets, and mixing processes for producing music. Once the source separation models are constructed, an important consideration is how to measure their performance. What are the performance metrics that can be used to evaluate source separation model performance? Evaluation metrics for source separation can be categorized into quantitative and qualitative metrics, each with its own advantages and disadvantages. These are discussed below.

2.4.1 Quantitative Metrics: SDR, SIR, SAR, and SI-SDR

Evaluation metrics play a crucial role in assessing the performance of music source separation (MSS) systems by quantifying their ability to accurately separate audio signals. Among the various evaluation metrics used in MSS, Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), Source-to-Artifact Ratio (SAR), and Scale-Invariant Source-to-Distortion Ratio (SI-SDR) are some of the most commonly employed ones [28].

Let $s_j(t) \in \mathbb{R}^l$ be the jth true source of length l, $m(t) \in \mathbb{R}^l$ be the mixture signal, and $\hat{s}_j(t) \in \mathbb{R}^l$ be the estimated sources from the MSS system. The estimated sources $\hat{s}_j(t)$ can be further decomposed into,

$$\hat{s}_j = s_j + e_{interference} + e_{artifact} + e_{noise} \tag{2.1}$$

where s_j is the clean target source that needs to be estimated, but is corrupted by error terms $e_{interference}, e_{artifact}, e_{noise}$, representing interference, artifacts, and noise respectively.The terms $e_{interference}$, $e_{artifact}$, and e_{noise} are unknown. The precise computations involved in determining these terms are rather complex, thus interested readers are directed to consult the original paper [28] for detailed calculations.

Source-to-Distortion Ratio (SDR) is a widely used metric in MSS that measures the quality of the separated sources by comparing them to the true sources. It quantifies the ratio of the power of the desired source to the power of the interference and artifacts introduced during the separation process. A higher SDR value indicates better separation performance, as it reflects a higher proportion of the desired signal relative to the distortion.

$$\text{SDR} = 10\text{log}_{10}\frac{\|s_j\|^2}{\|e_{interference} + e_{artifact} + e_{noise}\|^2} = 10\text{log}_{10}\frac{\|s_j\|^2}{\|\hat{s}_j - s_j\|^2} \tag{2.2}$$

Source-to-Interference Ratio (SIR) evaluates the quality of the separation by assessing the ratio of the power of the desired source to the power of the interference from other sources. It measures how well the system suppresses the interference from the undesired sources, providing insights into the system's ability to isolate the desired signal from the background noise. Similar to SDR, a higher SIR value signifies superior separation performance.

$$\text{SIR} = 10\text{log}_{10}\frac{\|s_j\|^2}{\|e_{interference}\|^2} \tag{2.3}$$

Source-to-Artifact Ratio (SAR) focuses on the presence of residual artifacts or distortions in the separated signals. It quantifies the ratio of the power of the desired source to the power of any remaining artifacts introduced during the separation process. SAR helps assess the clarity and fidelity of the separated signals, with lower values indicating a higher level of residual artifacts and poorer separation quality.

$$\text{SAR} = 10\text{log}_{10}\frac{\|s_j + e_{interference} + e_{noise}\|^2}{\|e_{artifact}\|^2} \tag{2.4}$$

Scale Invariant - Source-to-Distortion Ratio (SI-SDR) [29] is a relatively newer metric that addresses some limitations of SDR by providing a scale-invariant measure of the separation quality. Unlike SDR, which is sensitive to the scaling of the separated sources, SI-SDR accounts for variations in the source scales and provides a more robust evaluation of the separation performance. It computes the ratio of the energy of the estimated source to the energy of the distortion, considering the scale-invariance property.

$$\text{SI-SDR} = 10\log_{10}\frac{\|\alpha s_j\|^2}{\|\hat{s}_j - \alpha s_j\|^2} \tag{2.5}$$

where $\alpha = \frac{\hat{s}_j^T s_j}{\|s_j\|^2}$ is the scaling invariance factor.

These evaluation metrics are typically used together to comprehensively assess the performance of MSS systems, as each metric offers unique insights into different aspects of the separation quality. For instance, while SDR and SIR focus on the suppression of interference and artifacts, SAR provides additional information about the presence of residual distortions, and SI-SDR offers a scale-invariant measure of separation performance.

Limitations: Several limitations exist in the evaluation metrics despite their widespread use in the research community [1].

1. **Single SDR:** Many researchers often report a single SDR value by calculating the mean or median across examples and sources. While this is a common practice, it may not be the most accurate representation of system performance. A single number can be misleading and may not capture the nuances of source separation quality across different scenarios.

2. **Bias:** It has been observed that systems operating on waveform data tend to achieve higher SDR values compared to those operating on time-frequency (TF) domain representations. However, the human perception of audio quality is not correlated with SDR scores.

3. **Better SDR:** A higher SDR score does not always indicate superior source separation performance. For example, ConvTasnet and OpenUnMix achieve the same SDR score on the MUSDB18HQ dataset, yet ConvTasnet perceived as poorer quality when listened to compared to OpenUnMix.

Implementation: The widely used `bss_eval` [30] Python package was utilized in this thesis to compute these metrics.

2.4.2 Qualitative Metrics: AQ and IRQ

Listening tests are an essential component of evaluating MSS systems because they provide valuable insights into the perceptual quality of the separated audio signals. While quantitative metrics such as Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), and Signal-to-Artifact Ratio (SAR) offer objective measures of separation performance, they do not fully capture the subjective experience of human listeners. Therefore, listening tests complement quantitative evaluations by assessing how well the separated audio signals align with human perception.

One primary need for listening tests in MSS is to validate the effectiveness of separation algorithms in real-world scenarios. While quantitative metrics can provide useful benchmarks for comparing different algorithms, they may not accurately reflect the user experience in practical applications. Listening tests allow researchers to evaluate MSS systems in terms of their usability, naturalness, and overall audio quality, which are crucial factors for end-users such as musicians, audio engineers, and consumers.

Additionally, listening tests help identify potential limitations or artifacts introduced by MSS algorithms that may not be captured by quantitative metrics alone. For example, while a system may achieve high SDR scores, listeners may still perceive artifacts such as distortion, reverberation, or unnatural timbre in the separated audio signals. By conducting listening tests, researchers can gather qualitative feedback from listeners to identify areas for improvement and refine their algorithms accordingly.

Moreover, listening tests enable researchers to assess the robustness of MSS systems across different musical genres, recording conditions, and input configurations. Since music encompasses a wide range of styles and sonic characteristics, it is essential to evaluate MSS algorithms using diverse audio materials representative of real-world scenarios. Listening tests allow researchers to gauge how well their algorithms generalize to unseen data and whether they exhibit consistent performance across various contexts.

This thesis uses two metrics **Audio Quality (AQ)** and **Interference Reduction Quality (IRQ)**. Audio quality is rated on a scale of 0 to 5, indicating how well the audio is reconstructed without artifacts or distortion. Interference reduction quality is also rated on a scale of 0 to 5, assessing how effectively the sources are separated and evaluating the presence of other instrument sounds in the estimated source.

Limitations: Human evaluations are prone to errors, and individuals may have their own marking criteria. To improve reliability, a higher population is needed; the more people participate, more reliable. Also, conducting human evaluations is time-consuming and costly.

2.5 Literature Survey

Given the background on music signals, datasets, mixing processes, and their evaluation metrics, let's now delve into the literature on previously proposed source separation models, particularly focusing on those developed for Western pop music datasets.

The music source separation (MSS) problem has gained significant attention in recent years, particularly with the advent of deep learning. This problem is fundamental to various other music information retrieval (MIR) tasks. Several state-of-the-art MSS models, including Hybrid Transformer Demucs, BandsplitRNN, Wave-U-Net, Spleeter, and OpenUnMix, among others, have been developed. About a decade ago, traditional iterative algorithm-based solutions such as Independent Component Analysis (ICA), Non-Negative Matrix Factorisation (NMF), and Robust Principal Component Analysis (PRCA) were also commonly used.

In the remaining part of this chapter, a brief overview of these models and the trends in MSS are provided. At the end of the chapter, existing models specifically designed for interference reduction among microphone recordings, such as Kernel Additive Modelling for Interference Reduction (KAMIR), Multitrack Interference Reduction (MIRA), and FastMIRA, are discussed.

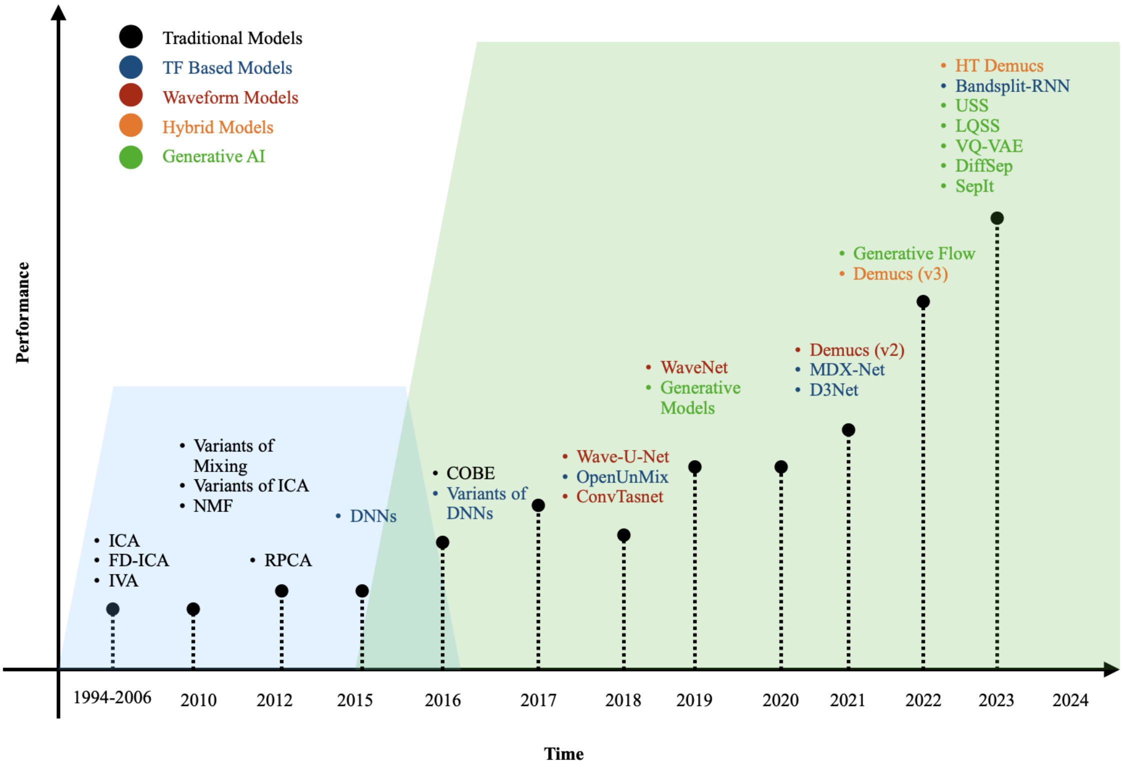


Figure 2.5: Timeline of improvements in music source separation models.

Figure 2.5 depicts the timeline of proposed music source separation models. The blue region represents the era of traditional models, while the green area denotes the emergence of neural network methods. The transition to neural networks occurred between 2015 and 2016. Each model is represented by a different colour code, reflecting its characteristics: dark red for raw waveform end-to-end models, dark blue for time-frequency (TF) based models (spectrograms), saffron for hybrid models that utilize both raw waveform and spectrograms, and green for Generative AI models.

2.6 Traditional Methods for Music Source Separation

The traditional models included Independent Component Analysis (ICA) and its variants, Non-Negative Matrices (NMF), Robust Principal Component Analysis (RPCA), and Common Orthogonal Basis Extraction (COBE). ICA serves as a fundamental solution to the blind source separation problem and was proposed in 1994. Since then, various variants of ICA and IVA (Independent Vector Analysis) have been introduced. From around 2010 to 2016, techniques based on NMF and RPCA were commonly used for source separation.

2.6.1 Independent Component Analysis

The Independent Component Analysis (ICA) is a powerful technique used to separate linearly mixed signals without prior knowledge of the mixing process [5]. Figure 2.6 illustrates the basic form of the blind source separation problem.

ICA relies on several assumptions to separate mixed signals into their constituent sources. Firstly, it assumes that the observed signals $x(t)$ are generated by a linear combination of statistically independent source signals $s(t)$. This means that the sources $s(t)$ are not correlated with each other and exhibit independent behaviours. Additionally, ICA assumes that the source signals are non-Gaussian, implying that their probability distributions are not Gaussian. This non-Gaussian assumption is essential for ICA algorithms to effectively distinguish between the different sources. Finally, ICA assumes that the mixing process itself is linear, indicating that the observed signals $x(t)$ are linear combinations of the source signals $s(t)$. These assumptions collectively form the foundation of ICA algorithms, guiding their approach to decomposing mixed signals into their original sources.

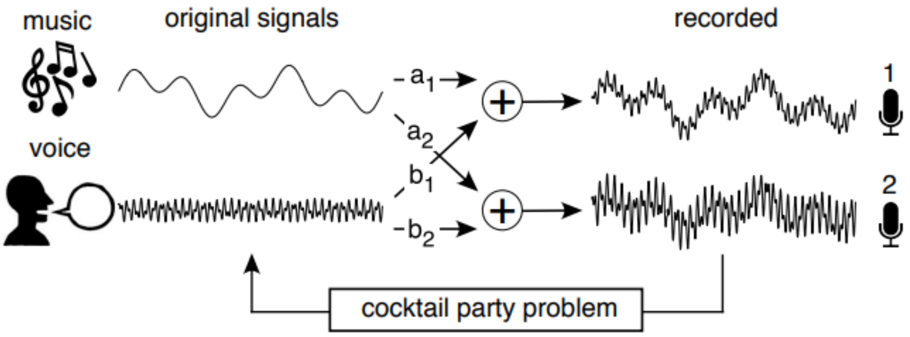


Figure 2.6: Simplest example of source separation problem where voice and music mixed together linearly. Figure taken from [5].

The concept behind ICA is to place multiple microphones at different locations to capture the signals around them. The number of microphones should match the number of sources that need to be recovered. The mixing process is described by equation 2.6:

$$X = \Lambda S \tag{2.6}$$

Here, X represents the observed microphone recordings, Λ is the mixing matrix, and S is the matrix of true sources. The goal of ICA is to estimate the original sources S from the observed mixture X by finding the inverse of the mixing matrix Λ. Most importantly, Λ is assumed to be invertible and square matrix.

The ICA estimates a new matrix W, such that the linear transformation of the data provides an estimate of the underlying sources,

$$\hat{S} = WX \tag{2.7}$$

The goal of the algorithm is to estimate the unmixing matrix W, such that when $W \to \Lambda^{-1}$ we get, $\hat{S} \approx S$.

Solution: The steps involved in deriving ICA can be summarized as follows:

1. Preprocessing: Normalize the data to zero mean and unit variance.

2. Whitening: Decorrelate the data by transforming it into a new space where the covariance matrix is the identity matrix. This is achieved through Principal Component Analysis (PCA) or Singular Value Decomposition (SVD).

3. ICA Model: Assume that the observed data X is a linear combination of independent components S through a mixing matrix Λ, such that $X = \Lambda S$.

4. Statistical Independence: Maximize the statistical independence of the estimated sources $\hat{S}$. This is typically done by maximizing non-Gaussianity measures of X such as negentropy, kurtosis, or mutual information.

5. Estimation: Use optimization algorithms such as gradient descent or fixed-point iteration to estimate the unmixing matrix W, which is the inverse of the mixing matrix A.

6. Back-projection: Retrieve the estimated independent components by applying the unmixing matrix to the observed data: $\hat{S} = WX$.

Various solutions and algorithms exist to solve ICA. Among the most recent and fastest ones is FastICA, which is considered as an efficient algorithm [31]. Recognizing the potential of utilizing time-frequency information for effectively extracting sources compared to using raw waveforms, ICA in the frequency domain was introduced [32]. Additionally, to address the linearity assumptions, efforts towards building ICA for convolutive [33, 34] and nonlinear [35] mixtures gained traction.

Limitations:

1. **Statistical Independence Assumption:** ICA assumes that the sources are statistically independent of each other. However, this assumption may not hold true in all scenarios, leading to inaccuracies in source separation. For example, music sources usually are highly correlated.

2. **Non-Gaussianity Requirement:** ICA typically requires the sources to have non-Gaussian distributions. If the sources are Gaussian or have similar distributions, ICA may struggle to separate them effectively.

3. **Sensitivity to Mixing Model:** ICA is sensitive to the mixing model used to create the observed signals. If the mixing process deviates significantly from the assumptions made by ICA, the separation performance may degrade.

4. **Computational Complexity:** Depending on the size of the dataset and the number of sources, ICA can be computationally intensive, making it less practical for real-time applications or large-scale problems.

5. **Ambiguity in Source Ordering:** ICA cannot determine the order of the separated sources, meaning that the order of the sources in the output may not match the original order, leading to ambiguity in interpretation.

2.6.2 Non-Negative Matrix Factorisation

Non-negative Matrix Factorization (NMF) served as a powerful technique in the field of music source separation (MSS) [36, 37], offering a robust approach to decompose mixed audio signals into their constituent sources. Unlike methods like Independent Component Analysis (ICA) that rely on statistical assumptions, NMF operates on the premise that both the observed spectrogram and the underlying sources can be represented as non-negative matrices. This characteristic makes NMF particularly suitable for MSS tasks, where audio signals are inherently non-negative and sparse.

At its core, NMF seeks to factorize a given non-negative matrix V, typically representing the magnitude spectrogram of a mixed audio signal, into two non-negative matrices W and H. The matrix W represents the spectral templates or basis functions of the sources, while the matrix H contains the corresponding activation coefficients or temporal dynamics associated with each basis function. Mathematically, the factorization can be expressed as:

$$V \approx WH \tag{2.8}$$

Where: V is an $m \times n$ matrix representing the magnitude spectrogram of the mixed audio signal, where m is the number of frequency bins and n is the number of time frames. W is an $m \times r$ matrix containing the spectral templates of r sources. H is an $r \times n$ matrix containing the activation coefficients corresponding to each source over time.

The goal of NMF is to find the optimal factorization W and H such that their product closely approximates the original spectrogram V. This is typically achieved by minimizing a cost function such as the Euclidean distance or the Kullback-Leibler divergence between V and WH, subject to the constraint that all elements of W and H are non-negative.

One of the key advantages of NMF is its interpretability, as the basis functions in W often correspond to meaningful spectral features of the sources. For example, in music source separation, the basis functions may resemble spectral profiles of musical instruments or vocal characteristics. This interpretability facilitates the identification and extraction of specific sources from the mixed audio signal.

Numerous variations and extensions of NMF have been proposed in the context of MSS, each tailored to address specific challenges or enhance performance in certain scenarios [38, 39, 40, 41, 42]. For instance, sparse NMF imposes sparsity constraints on the activation coefficients H [43], promoting a more parsimonious representation of the sources. Convolutional NMF (CNMF) [44] extends the basic NMF framework to capture temporal dependencies and structural patterns in the spectrogram, while robust NMF variants [45] aim to mitigate the influence of outliers or noise in the input data.

2.6.3 Robust Principal Component Analysis

Robust Principal Component Analysis (RPCA) [46] technique has been used for singing voice source separation from monaural recordings [6]. In music source separation tasks, RPCA aims to decompose the observed spectrogram into two components: a low-rank component representing the background music and a sparse component capturing the singing voice. This separation is achieved by exploiting the inherent low-rank structure of the background music and the sparsity of the singing voice in the spectrogram domain, as shown in Figure 2.7.

RPCA is a method used for decomposing a given data matrix X into two components: a low-rank matrix L and a sparse matrix S. Mathematically, this decomposition can be represented as:

$$X = L + S \tag{2.9}$$

where L represents the low-rank component containing the background music, and S represents the sparse component containing the singing voice. The RPCA algorithm aims to minimize the rank of L and the sparsity of S, subject to certain constraints.

The optimization problem associated with RPCA can be formulated as follows:

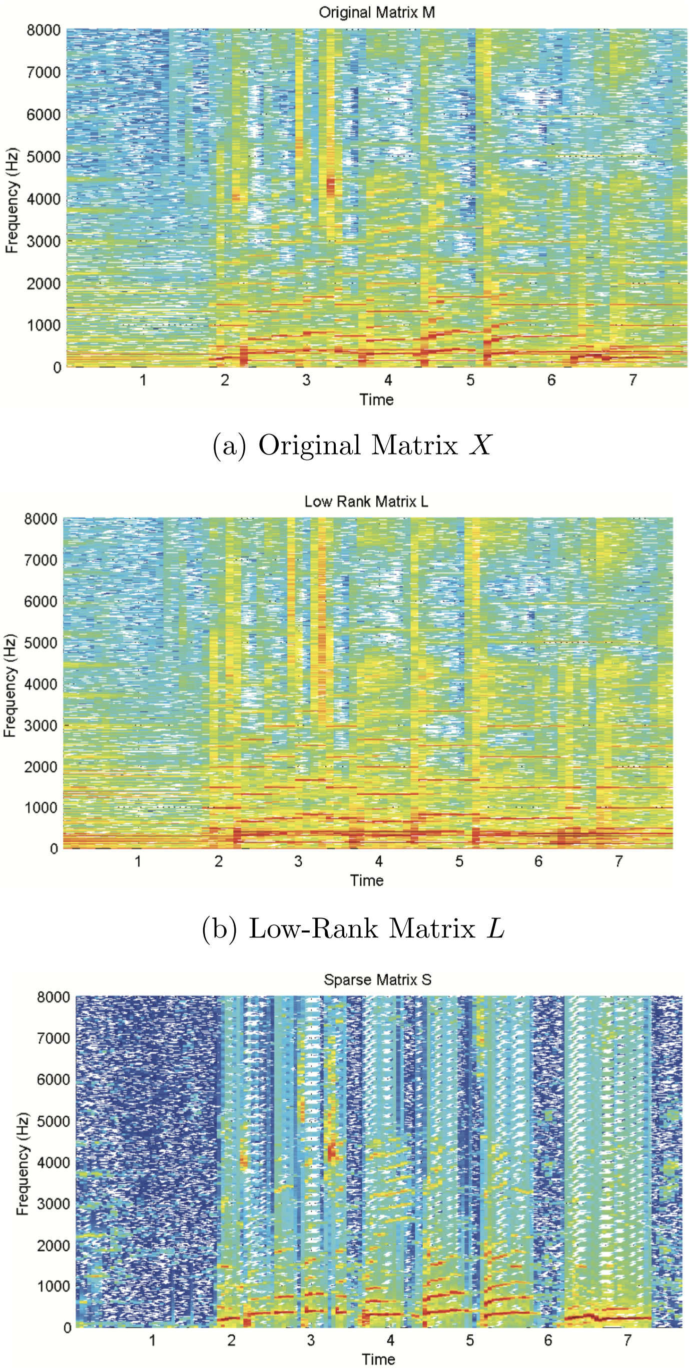
$$\min_{L,S} \operatorname{rank}(L) + \lambda \|S\|_0 \quad \text{subject to} \quad X = L + S \tag{2.10}$$

where $\operatorname{rank}(L)$ denotes the rank of the matrix L, $\|S\|_0$ represents the sparsity of the matrix S, and λ is a regularization parameter controlling the trade-off between the low-rank and sparse components.

The optimization problem is typically solved using iterative algorithms such as the Alternating Direction Method of Multipliers (ADMM) or the Principal Component Pursuit (PCP) algorithm. These algorithms iteratively update the low-rank and sparse components until convergence is reached.

One of the key advantages of RPCA is its ability to handle highly corrupted or noisy recordings, making it particularly suitable for real-world scenarios where recordings may contain various forms of interference or background noise. By decomposing the spectrogram into low-rank and sparse components, RPCA effectively isolates the singing voice from the background music, even in the presence of significant interference.

In summary, RPCA offers a powerful solution for singing voice source separation from monaural recordings. By decomposing the input spectrogram into low-rank and sparse components, RPCA can effectively isolate the singing voice from the background music, making it a valuable tool for various audio processing applications. RPCA is widely used only for singing voice separation from the background music.



(a) Original Matrix X

(b) Low-Rank Matrix L

(c) Sparse Matrix S

Figure 2.7: Figure adapted from [6], shows an example RPCA results at SNR=5 for (a) the original matrix X, (b) the low-rank matrix L, and (c) the sparse matrix S.

2.7 Deep Neural Networks

The main disadvantage of traditional methods for source separation (SS) lies in their linear formulation and the simplified assumptions they make, which are often invalid in realistic scenarios. Given the surge in neural networks in the recent decade, various deep neural network (DNN)-based architectures have been employed to address the SS problem.

2.7.1 Overview

Early deep neural network (DNN) methods [47, 48, 49, 50, 51] used for music source separation (MSS) laid the foundation for more advanced techniques by exploring the capabilities of neural networks in extracting source signals from mixed audio. These methods primarily focused on employing basic architectures like feedforward neural networks (FNNs) and fully connected layers to address the MSS problem. While these approaches may seem rudimentary compared to contemporary methods, they played a crucial role in demonstrating the potential of neural networks for MSS and paving the way for subsequent research.

One of the pioneering works in this area is the study by [52] in the year 2014, where a simple FNN architecture was employed for monaural source separation. The authors proposed a novel approach that utilized the non-negative matrix factorization (NMF) technique to preprocess the input spectrogram before feeding it into the neural network. The FNN, consisting of multiple hidden layers with ReLU activation functions, was trained to learn the mapping from the preprocessed spectrogram to individual source signals. Despite its simplicity, the model showed promising results in separating vocals from background music in monaural recordings.

Building upon this foundation, [53] introduced a more sophisticated DNN architecture for MSS, termed Deep Unmixing Network (DUNet). The DUNet comprised multiple layers of convolutional and recurrent neural networks, enabling it to capture both temporal and spectral dependencies in the audio signals. The model was trained using a large dataset of mixed music recordings with ground truth source signals. By leveraging the power of deep learning, DUNet achieved remarkable performance improvements over traditional methods, demonstrating its efficacy in tackling the MSS challenge.

Another notable contribution to early DNN methods for MSS is the work by [54], where a convolutional neural network (CNN) architecture was proposed for singing voice separation. In their approach, the CNN was designed to learn the spectral patterns associated with vocal and non-vocal regions in the audio spectrogram. By leveraging the inherent structure of the spectrogram, the model effectively discriminated between singing voice and background music components, leading to accurate source separation results.

Model	Year	Type	SDR (dB)
Wave-U-Net [57]	2018	Waveform	3.2
OpenUnMix [60]	2018	Spectrogram	5.3
ConvTasnet [56]	2018	Waveform	5.7
Wavenet [58]	2019	Waveform	3.0
D3Net [61]	2021	Spectrogram	6.0
Demucs (v2)	2021	Waveform	6.3
KUIELAB-MDX-NET [62]	2021	Hybrid	7.5
Hybrid Demucs (v3)	2022	Hybrid	7.7
Hybrid Transformer Demucs (v4) [63]	2023	Hybrid	9.0
Band-Split RNN [64]	2023	Spectrogram	9.0

Table 2.3: Performances of the recent MSS models on MUSDB18HQ dataset.

Despite their success, early DNN methods for MSS faced several challenges and limitations. One common issue was the lack of robustness to variations in audio recordings, such as differences in recording conditions, music genres, and instrumentation. Additionally, these methods often required large amounts of labelled training data to achieve satisfactory performance, which limited their scalability and applicability in real-world scenarios.

After the emergence of DNN, multiple methods were introduced in the time-frequency (TF) domain. The first waveform speech separation model, Tasnet [55], was introduced in 2018. Subsequently, ConvTasnet [56] and its variants were proposed, demonstrating comparable performance to TF-based methods. These methods were introduced in speech separation and following their success, adapted to MSS. This led to a surge in end-to-end systems such as Wave-U-Net [57] and WaveNet [58] for MSS. Facebook introduced their Demucs source separation model, which is built on Wave-U-Net's principles. Hybrid models, capable of processing both TF and waveform data simultaneously, also emerged around 2018.

By 2023, the Hybrid Transformer Decumucs model stood out as the best-performing model, achieving an SDR of 9.0. Additionally, in the same year, Band-Split RNNs were introduced, delivering performance on par with Facebook's Demucs, also with an SDR of 9.0. These models were trained and tested on MUSDB18HQ datasets. Table 2.3 lists various models proposed, along with their respective years of introduction and achieved Signal-to-Distortion Ratio trained and evaluated using the MUSDB18HQ dataset.

In 2023, Carnatic singing voice separation using cold diffusion on training data with bleeding was proposed [59]. This method skips the interference reduction problem by directly training the source separation model with the interference data.

2.7.2 Wave-U-Net

The Wave-U-Net [57] presents a novel neural network architecture designed specifically for audio source separation tasks in the waveform domain, developed in the year 2018. The authors motivate the need for a neural network architecture that can perform end-to-end audio source separation, bypassing the need for handcrafted features or manual intervention. They propose the Wave-U-Net architecture as a solution to this problem, aiming to leverage the representational power of deep neural networks to learn complex mappings from mixed audio signals to their constituent sources.

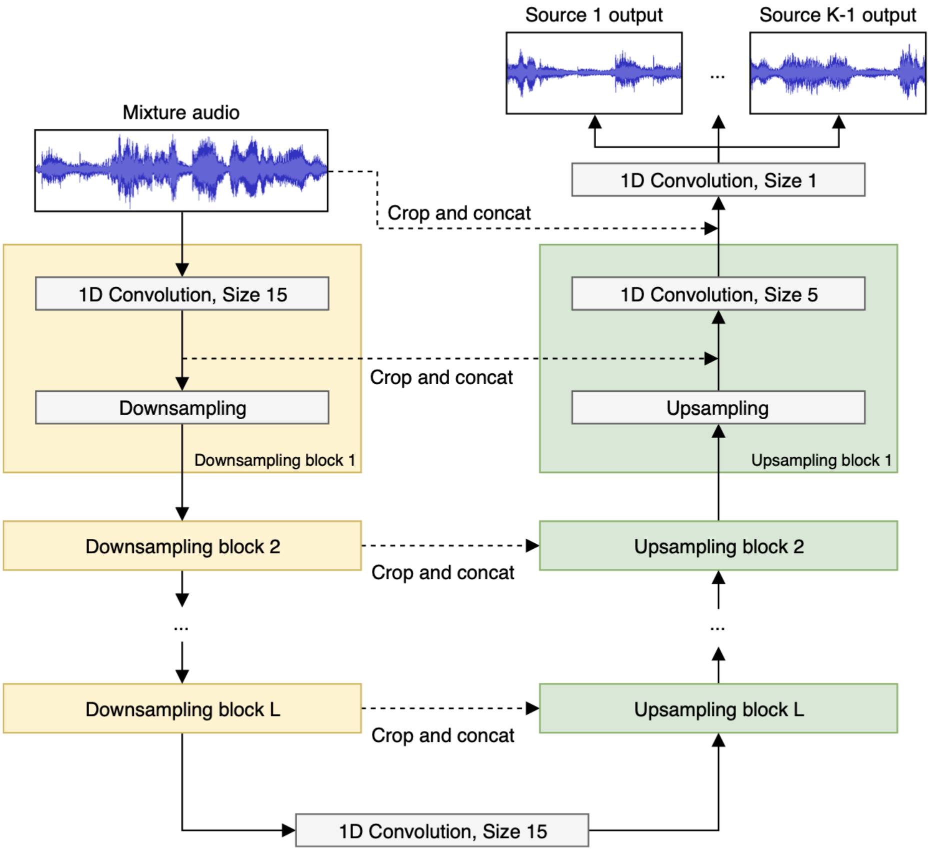


Figure 2.8: Wave-U-Net Architecture.

Wave-U-Net Architecture: The Wave-U-Net architecture consists of a U-Net-like structure combined with dilated convolutions and skip connections to capture both local and global contextual information in the audio signals. The architecture operates at multiple scales, allowing it to effectively model long-range dependencies while preserving fine-grained details. The architecture is shown in Figure 2.8.

Model Components:

1. **Downsampling Path:** The downsampling path of the Wave-U-Net architecture consists of convolutional layers followed by max-pooling operations, progressively reducing the temporal resolution of the input audio waveform while increasing its receptive field. This allows the network to capture high-level features and context from the input signal.

2. **Upsampling Path:** Conversely, the upsampling path consists of convolutional layers followed by upsampling operations, gradually increasing the temporal resolution of the features learned by the network. This path facilitates the reconstruction of source signals from the learned representations.

3. **Skip Connections:** Skip connections are incorporated between corresponding layers in the downsampling and upsampling paths, enabling the network to combine both local and global contextual information effectively. These connections help mitigate the vanishing gradient problem and allow the network to retain fine-grained details during the upsampling process.

4. **Dilated Convolutions:** Dilated convolutions are employed within the convolutional layers to increase the receptive field without sacrificing spatial resolution. By incorporating dilations, the network can capture long-range dependencies in the audio signals while maintaining computational efficiency.

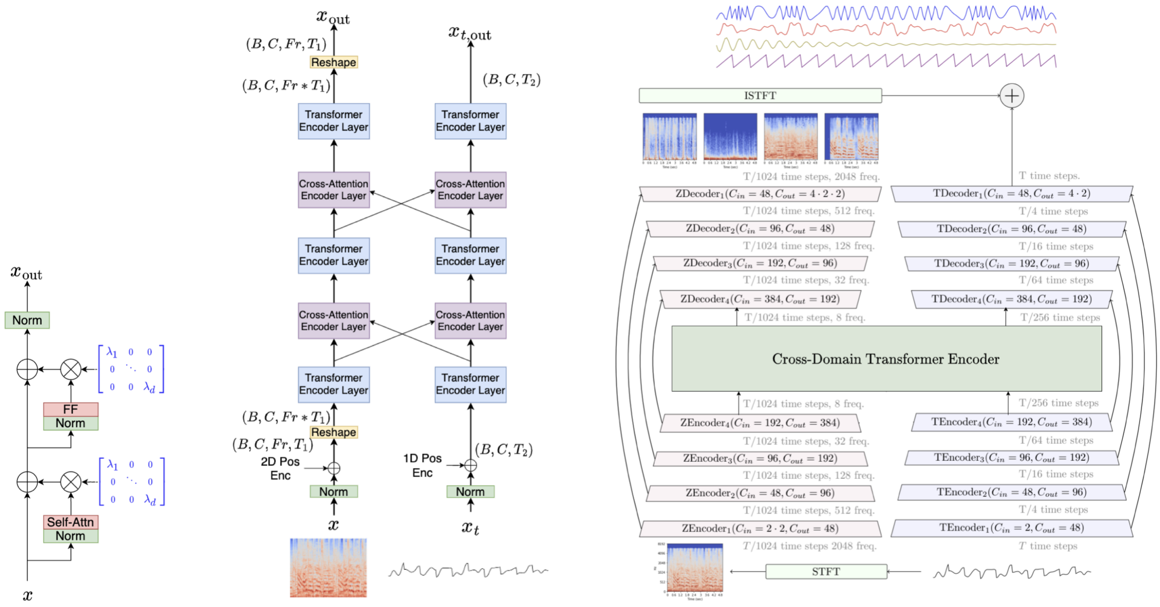
Training Procedure: The Wave-U-Net is trained using a supervised learning approach, where pairs of mixed audio signals and their corresponding clean source signals are used as input-output pairs. The network is trained to minimize a suitable loss function, such as the mean squared error or the source-to-distortion ratio (SDR), which measures the quality of the separated source signals compared to the ground truth.

Evaluation and Results: The paper evaluates the performance of the Wave-U-Net architecture on various audio source separation tasks, including music source separation and speech enhancement. The results demonstrate the effectiveness of the proposed architecture in separating complex audio mixtures into their constituent sources, achieving state-of-the-art performance on benchmark datasets.

2.7.3 Hybrid Transformer Demucs

Facebook's Hybrid Transformer Demucs model [63] represents a significant advancement in the field of audio source separation with exceptional accuracy. Developed by researchers at Facebook AI in the year 2023, this model combines elements of transformer-based architectures with the principles of Demucs (Deep Extractor for Music Sources), offering a potent solution for audio source separation tasks. The Demucs is an UNet architecture, with deep connections which works in raw waveform.

The architecture of the Hybrid Transformer Demucs model as shown in Figure 2.9 builds upon the success of the original Demucs model, which itself was a groundbreaking development in audio source separation. Demucs leveraged a novel approach to source separation by combining deep neural networks with the Unet architecture, which enabled it to achieve state-of-the-art performance in isolating musical sources from mixed audio recordings. However, Demucs primarily operated in the time domain and had limitations in handling complex mixtures with reverberations and overlapping sources.



x_{out}
Norm
λ_1 0 0 / 0 ⋱ 0 / 0 0 λ_d
FF
Norm
Self-Attn
Norm
x
x_{out}
(B, C, Fr, T_1)
Reshape
$(B, C, Fr * T_1)$
$x_{t,\text{out}}$
(B, C, T_2)
Transformer Encoder Layer
Cross-Attention Encoder Layer
2D Pos Enc
1D Pos Enc
Norm
x
x_t
ISTFT
T/1024 time steps, 2048 freq.
T time steps.
ZDecoder$_1(C_{in} = 48, C_{out} = 4 \cdot 2 \cdot 2)$
TDecoder$_1(C_{in} = 48, C_{out} = 4 \cdot 2)$
T/1024 time steps, 512 freq.
T/4 time steps
ZDecoder$_2(C_{in} = 96, C_{out} = 48)$
TDecoder$_2(C_{in} = 96, C_{out} = 48)$
T/1024 time steps, 128 freq.
T/16 time steps
ZDecoder$_3(C_{in} = 192, C_{out} = 96)$
TDecoder$_3(C_{in} = 192, C_{out} = 96)$
T/1024 time steps, 32 freq.
T/64 time steps
ZDecoder$_4(C_{in} = 384, C_{out} = 192)$
TDecoder$_4(C_{in} = 384, C_{out} = 192)$
T/1024 time steps, 8 freq.
T/256 time steps
Cross-Domain Transformer Encoder
T/1024 time steps, 8 freq.
T/256 time steps
ZEncoder$_4(C_{in} = 192, C_{out} = 384)$
TEncoder$_4(C_{in} = 192, C_{out} = 384)$
T/1024 time steps, 32 freq.
T/64 time steps
ZEncoder$_3(C_{in} = 96, C_{out} = 192)$
TEncoder$_3(C_{in} = 96, C_{out} = 192)$
T/1024 time steps, 128 freq.
T/16 time steps
ZEncoder$_2(C_{in} = 48, C_{out} = 96)$
TEncoder$_2(C_{in} = 48, C_{out} = 96)$
T/1024 time steps, 512 freq.
T/4 time steps
ZEncoder$_1(C_{in} = 2 \cdot 2, C_{out} = 48)$
TEncoder$_1(C_{in} = 2, C_{out} = 48)$
T/1024 time steps 2048 freq.
T time steps
STFT

(a) Transformer Encoder Layer

(b) Cross-domain Transformer Encoder of depth 5

(c) Hybrid Transformer Demucs

Figure 2.9: Hybrid Transformer Demucs Architecture.

To address these challenges, the Hybrid Transformer Demucs model introduces several key innovations. First and foremost, it incorporates transformer-based components, known for their effectiveness in capturing long-range dependencies in sequential data, such as language modelling and machine translation. By integrating transformers into the architecture, the model gains the ability to effectively process both time-domain and frequency-domain representations of audio signals, thereby enhancing its capacity to handle complex mixtures with varying temporal and spectral characteristics.

One of the defining features of the Hybrid Transformer Demucs model is its hybrid architecture, which seamlessly combines convolutional and transformer layers. This hybrid approach allows the model to leverage the strengths of both types of layers: the convolutional layers excel at capturing local patterns and short-term dependencies in the audio signal, while the transformer layers excel at capturing global structures and long-range dependencies. This synergistic combination enables the model to achieve superior performance in separating overlapping sound sources while preserving the temporal coherence and spectral details of the individual sources.

Furthermore, the model incorporates innovative training techniques and optimization strategies to enhance its performance and generalization capabilities. It utilizes self-supervised learning objectives, such as time contrastive learning, to learn robust representations of audio signals without the need for manually annotated labels. Additionally, the model leverages advanced optimization algorithms, such as AdamW, to stabilize training and prevent overfitting, thereby improving its ability to generalize to unseen data.

In terms of performance, the Hybrid Transformer Demucs model has demonstrated remarkable results across a range of audio source separation tasks. It consistently outperforms previous state-of-the-art models in terms of signal-to-distortion ratio. By achieving higher SDR scores, the model produces cleaner and more accurate separations, making it highly effective for applications such as music remixing, speech enhancement, and audio denoising.

2.8 Generative AI

Since 2019, there have been notable advancements in generative-based source separation techniques. These include variational autoencoder (VAE) models, vector quantized variational autoencoder (VQ-VAE) architectures [65, 66], as well as novel methods such as Sepformer [67], Diffusion-based source separation techniques like diffsep [68, 69], and SepIt algorithms [70]. These are predominantly used as both speech and music separation.

In the past two years, Large Language Models (LLMs) have emerged as a dominant force, not only in natural language processing but also in other domains such as computer vision and audio processing (LAMs: Large Audio Models [71]). The proliferation of LLMs and LAMs has catalyzed advancements in source separation, leading to the emergence of a new paradigm known as Language Queried Audio Source Separation (LQSS). LQSS models, such as AudioSep [72], represent a universal source separation framework capable of accepting mixed audio inputs and generating individual sources based on text prompts. This novel approach holds significant promise for the future of source separation, offering a versatile and intuitive means of isolating audio components from complex mixtures.

2.9 Interference Reduction Systems

The last section provided an overview of recent literature on MSS, outlining its progress. Now, let's shift the focus to interference reduction techniques. As outlined in Chapter 1, interference reduction differs significantly from source separation systems, necessitating distinct algorithms or models to address its challenges.

While significant advancements have been made in source separation literature, particularly in the Western domain, interference reduction systems have not received as much attention. Until 2023, only DSP-based iterative algorithms had been introduced to tackle this issue.

In the literature, most of the proposed works in interference reduction make use of the assumption that the microphone(s) physically nearest to a source maximally captures the source, and captures other sources to a lesser extent. Also, the time-frequency domain techniques [73, 74, 75] have been the main focus of interference reduction algorithms, which have produced good results.

The Kernel Additive Modelling for Interference Reduction (KAMIR) algorithm, as presented in [76], emerged as a considerable advancement by delivering state-of-the-art outcomes. This iterative technique employs a stepwise approach to ascertain the clean sources, systematically estimating the power spectral density and its corresponding magnitude via generalized Wiener filtering. KAMIR effectively processes the interference spectrogram, generating both the interference-reduced spectrogram and its corresponding strength by using non-negative matrix factorisation coupled with the β divergence, as outlined in [77].

Later, the Multi-track Interference Reduction Algorithm (MIRA) was introduced, which optimises KAMIR by replacing the power spectral density with the fractional power spectral density inspired by [78]. It simplifies the algorithm by dropping the kernel filtering and the frequency dependence of the strength of the sources.

Both the KAMIR and MIRA algorithms shared the common drawback of being time-intensive due to their iterative nature. Consequently, the applicability of these algorithms to real-world, full-length live recordings was constrained. Addressing these time limitations, the FastMIRA [79] algorithm was introduced as a remedy. FastMIRA introduces a novel approach by leveraging random projections, effectively bypassing the need to process the entire full-length input. This adaptation yields outcomes of sufficient quality that are akin to the results generated by its predecessor algorithm, MIRA.

In earlier techniques, time-domain-based echo cancellation, IIR filters [80], and adaptive filtering [81] were explored. [82] proposes an algorithm that estimates the interference spectra through gradient descent by minimising the distance between the actual and estimated interference spectra, and then obtains the clean source by performing spectral subtraction. The disadvantage of this method is that it assumes that the sources present in the track are known beforehand.

When compared to time-domain filtering in the aforementioned methods, Wiener filtering-based time-frequency approaches produced good results. However, the sound quality, the blind source separation (BSS) evaluation metrics like source-to-distortion ratio (SDR), and the time complexity of these models are poor, making them unsuitable for use in many real-world scenarios.

2.9.1 KAMIR

The “Kernel Additive Modeling for Interference Reduction in Multi-channel Music Recordings” [76] presents a method for reducing interference in multi-channel music recordings using Kernel Additive Modeling (KAM). This serves as the current state-of-the-art in the interference reduction literature. The technique aims to separate individual sources from the recorded signals by modelling their interdependencies and employing short-term Fourier transform (STFT) representations. The KAMIR has been used for removing interference from live recordings.

Notations and Model: The model considers I microphones capturing J sources, where each source may be present in all channels to varying degrees. The observed signals from each microphone are represented as a sum of contributions from the individual sources. STFTs are computed for each microphone channel, resulting in matrices of TF bins. The model assumes independence across frequency bins and independence between the STFTs of different sources. Each STFT is modelled using a Local Gaussian Model (LGM), where each TF bin is assumed to follow a complex isotropic Gaussian distribution.

Separation Method: Given the independence assumption, the resulting sum of STFTs for each microphone channel follows a complex isotropic Gaussian distribution. By assuming known power spectral densities (PSDs) for each source and frequency-dependent interference matrices, the Minimum Mean-Squared Error (MMSE) estimate of each source’s STFT can be obtained using Wiener filtering. The algorithm aims to estimate the clean signals for each source by iteratively updating parameters such as PSDs and interference matrices. The initialization involves setting interference matrices with minimal interference assumptions and initializing estimates of the source’s STFTs with observed signals. Parameters such as PSDs are updated based on estimates of the STFTs, while interference matrices are updated using Non-negative Matrix Factorization (NMF) to enforce non-negativity and adjust for interference.

Parameter Estimation Algorithm: The KAMIR algorithm iteratively updates parameters through a separation step and a parameter fitting step. In the separation step, MMSE estimates of each source's STFT are obtained using Wiener filtering. In the parameter fitting step, PSDs and interference matrices are updated based on the estimated STFTs. The algorithm alternates between these steps until convergence.

The initialization involves setting minimal interference assumptions for interference matrices and initializing estimates of the source's STFTs with observed signals. Parameters such as PSDs are updated based on estimates of the STFTs, while interference matrices are updated using Non-negative Matrix Factorization (NMF) with β divergence to enforce non-negativity and adjust for interference.

The KAMIR algorithm iteratively updates parameters through a separation step and a parameter fitting step. In the separation step, MMSE estimates of each source's STFT are obtained using Wiener filtering. In the parameter fitting step, PSDs and interference matrices are updated based on the estimated STFTs. The algorithm alternates between these steps until convergence.

The input to the algorithm consists of the STFT of interfered sources x_i. The algorithm iteratively estimates the interference matrix Λ and reduces the interference in the PSD of the sources. Subsequently, the audio is reconstructed using Wiener Filtering. The KAMIR algorithm is outlined below. Overall, the KAM approach offers a robust method for reducing interference in multi-channel music recordings, with applications in various audio processing tasks.

Algorithm 1 KAMIR: Kernel Additive Modelling for Interference Reduction Algorithm

```
1. Input:
    • $X_i(\omega, t)$ for each channel $x_i$
    • Channel selection function $\phi(j)$ for each source $j$
    • Minimal interference $\rho \in [0, 1]$
    • Optional: Kernels $k_j$ for each source $j$
2. Initialization:
    • For each $\omega$, initialize $\Lambda(\omega)$ as in $\forall(i, j, w), \Lambda(\omega) = 1$ if $i \in \phi(j)$ and $\rho$ otherwise
    • For each $j$, for each $i \in \phi(j)$, $Y_{ij} \leftarrow X_i$
3. Parameter fitting step:
    • For each $j$: update $P_j$ as in,
          $P_j(\omega, t) \leftarrow \frac{1}{|\phi(j)|} \sum_{i \in \phi(j)} \frac{1}{\Lambda_{ij}(\omega)} |\hat{Y}_{ij}(\omega, t)|^2$
    • Optional: For each $j$: apply median filter on $P_j$ with kernel $k_j$
    • For each $\omega$: update $\Lambda(\omega)$ using,
          $\lambda_{ij}(\omega) \leftarrow \lambda_{ij} \frac{\sum_{t=1}^{N_t} \hat{V}_i(\omega, t)^{\beta-2} V_i(\omega, t) P_j(\omega, t)}{\sum_{t=1}^{N_t} \hat{V}_i(\omega, t)^{\beta-1} P_j(\omega, t)},$
      where $V_i(\omega, t) \triangleq |X_i(\omega, t)|^2$ and $\hat{V}_i(\omega, t) \triangleq \sum_j \lambda_{ij}(\omega) P_j(\omega, t)$
    • Re-scale $P_j$ with $P_j(\omega, t) \leftarrow P_j(\omega, t) \sum_{i=1}^{I} \lambda_{ij}(w)$ and normalize $\Lambda$ with $\lambda_{ij}(\omega) \leftarrow max \left[ \rho, \frac{\lambda ij(\omega)}{\sum_{i'=1}^{I} \lambda_{i' j(\omega)}} \right]$
4. Separation step:
    • For each $j$, for each $i \in \phi(j)$: update $Y_{ij}$ as in,
          $\hat{Y}_{ij}(\omega, t) = \frac{\lambda_{ij} P_j(\omega, t)}{\sum_{j'=1}^{J} \lambda_{ij'}(\omega) P_{j'}(\omega)} X_i(\omega, t)$
          $= W_{ij}(\omega, t) X_i(\omega, t),$
      where $W_{ij}(\omega, t)$ is the Wiener gain
    • For another iteration, return to step (3)
5. Output: $\hat{Y}_{ij}(\omega, t)$ for each $j$, for each $i \in \phi(j)$
```

CHAPTER 3

Linear Mixing Models for Interference Reduction

Chapter 3 introduces linear mixing models proposed for interference reduction. It begins by presenting straightforward convolutional autoencoder models for each source. Next, the chapter delves into the linear mathematical formulation of the problem, addressing it through an optimization approach. Lastly, a neural network architecture termed tUNet is described. Finally, all the models are examined in the context of improving the music source separation model.

3.1 Convolutional Autoencoders

One straightforward approach to solving the interference reduction problem is by treating interference as noise. This can be modelled using a denoising autoencoder, trained in a supervised fashion with clean-interference paired data. For this purpose, we utilize a convolutional autoencoder.

Convolutional autoencoders (CAEs) have been used for processing audio inputs in both the time domain and the time-frequency domain. CAEs have been used to separate music of various genres from speech in [83]. They have also been used for single [84] and multichannel speech enhancement [85].

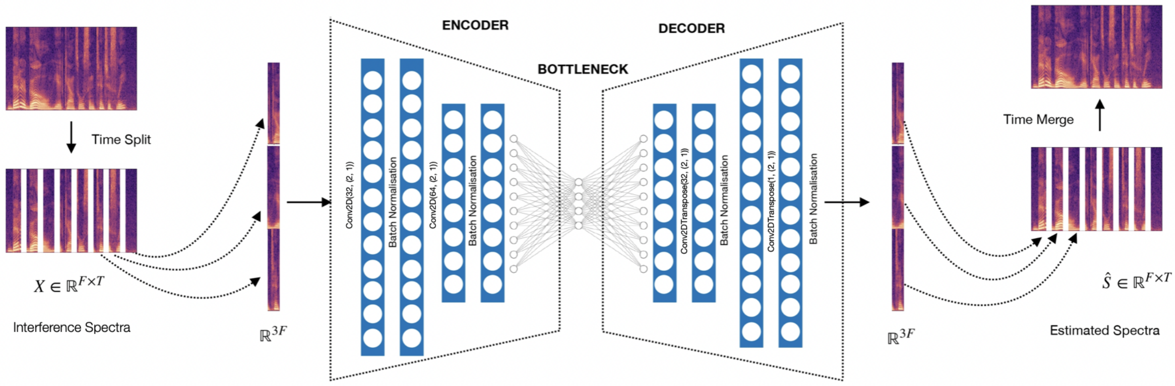


Figure 3.1: Convolutional autoencoder (CAE) for interference reduction.

The proposed CAE model for interference reduction is depicted in Figure 3.1. In this approach, we treat the interference in each stem as noise. If $x(t)$ represents the signal in a particular track, then we have

$$x(t) = s(t) + n(t), \tag{3.1}$$

where $s(t)$ represents the source(s) associated with the track, and $n(t)$ is the unwanted signals from all other sources. Let $X(f,t) \in \mathbb{R}^{F\times T}$ be the short-time magnitude spectrum of the input $x(t)$. Using paired training data, the CAE learns the relationship between the short-time magnitude spectra of $x(t)$ and $s(t)$. During evaluation, given the input $X(f,t)$ containing interference, the CAE estimates the spectrum of $s(t)$, denoted in the figure by $\hat{S}(f,t)$. To give temporal context, the input is provided by stacking three frames of the magnitude spectra, and is reshaped at the output.

3.1.1 Network architecture

The network consists of an encoder, decoder, and dense layers as shown in Figure 3.1. The encoder contains two sets of 2D convolution layers, each followed by batch normalisation layers. The convolution layer has 32 (2×1) and 64 (2×1) kernel filters. The encoder outputs latent features of size (e_1, e_2). We flatten the feature and pass it to a dense layer (100 neurons with RELU activation), then another dense layer ($e_1 e_2$ neurons with RELU activation) and reshape it to pass as input to the decoder. The decoder has a symmetric structure, the last layer of which outputs the estimated source spectrum $\hat{S}(f, t)$. The output signal is reconstructed by taking the inverse short-time Fourier transform, using the input phase.

3.1.2 Experiments and Results

Datasets: We primarily utilize the standard MUSDB18HQ dataset [25] to train and evaluate the proposed CAE network. Since the MUSDB18HQ dataset contains clean, interference-free stems along with their respective mixtures, we need to artificially stimulate interference effects.

As discussed in Section 2.2.1 in Chapter 2, the stems in MUSDB18HQ are used to create artificial interference effects. We then use these artificial mixes of the standard MUSDB18HQ dataset to train and evaluate the CAE model.

For training the CAE, data were generated by interfering with a given stem with the other three stems, each reduced by 20 dB. The stems were chosen from the same track, and this process was repeated for all the tracks in the training subset of MUSDB18HQ, resulting in interfered versions of each stem. Having access to isolated sources allows us to use BSS evaluation metrics to report the effectiveness of various techniques.

CAE training: Spectrograms were computed with a window size of 93 msec, with a hop of 25% and a 2048 point FFT. A temporal context of three frames is used as shown in Figure 3.1. The CAE was trained using pairs of clean and interfered spectrograms, using mean-square error (MSE) loss function, Adam optimizer with a learning rate of 0.001, and a batch size of 16. Separate CAEs are trained for vocals, bass, drums and other stems.

Results: Individual dedicated CAEs have been trained for each source, and the results are included in Section 3.4 along with the proposed tUNet. The results show that the CAE models were able to reduce interference in the microphone recordings and produced a higher Source-to-Distortion Ratio (SDR) when compared with the KAMIR algorithm.

3.1.3 Limitations

Generalization and Domain Shift Issues:

1. The primary limitation of the CAE approach lies in its inability to generalize across different stems. For instance, a CAE model trained specifically for the vocal stem may not effectively separate other types of stems such as drums or bass. Consequently, constructing dedicated CAE models for each stem becomes necessary, which can be a laborious and time-consuming process.

2. Additionally, CAE models are susceptible to domain shift issues, where the performance of the model may degrade when applied to data from a different domain than the one it was trained on. This further exacerbates the generalization problem, making it challenging to deploy CAE models in real-world scenarios where the input data may vary considerably.

Phase Issues

1. Another limitation of the CAE approach is related to phase information. CAE models typically operate on the magnitude of the Short-Time Fourier Transform (STFT), discarding valuable phase information present in the audio signals. This loss of phase information can affect the accuracy of the separation process, particularly in scenarios where phase coherence plays a crucial role in distinguishing between different audio components.

2. The reliance on magnitude-only representations can lead to phase inconsistencies in the separated signals, potentially impacting the perceptual quality of the separated stems. As a result, despite achieving satisfactory results in terms of interference reduction, CAE-based approaches may fall short in preserving the phase information of the separated audio components, thereby limiting their effectiveness in certain applications.

3.1.4 Complex CAEs

One of the limitations of the CAE model is that it takes the STFT spectrum as input. It neglects the phase information. One way to overcome this issue is to process both magnitude spectrum and phase in CAE. Given the signal $x(t)$, the spectrum $X(f,t)$ has been obtained by short-time Fourier transform. The magnitude spectrum $|X(f,t)|$ and phase $\angle X(f,t)$ are stacked together and passed to the new complex CAE. The only difference in the architecture is all the 2D convolutions are replaced with 3D convolutions in Figure 3.1.

Though the model includes phase information, its major drawback on generalisation is not addressed. Additionally, the SDR obtained by this model was lower than that achieved by the CAE models.

In conclusion, modelling interference as noise offers a partial solution but does not fully resolve the problem. It's important to recognize that this approach has significant limitations that render it unsuitable for real-world scenarios.

3.2 Linear Mathematical Formulation

The CAE encountered several limitations by treating interference as noise. Instead of modeling interference as noise, we can address the problem using a linear mixing model. This approach offers several advantages over the CAE's simplistic noise model.

Let $x(t) \in \mathbb{R}^l$ be the signal of length l that is picked up by the microphone and $s(t) \in \mathbb{R}^l$ be the true clean sources. Assuming there are n sources and k microphones and for each source, there is at least one microphone capturing the dominant source ($k \geq n$). Then for any source i, $x(t)$ is given by,

$$x_i(t) = \lambda_{ii} s_i(t) + \sum_{j=0, j \neq i}^{n} \lambda_{ij} s_j(t) \tag{3.2}$$

where λ_{ij} is the amount of bleed observed in $x(t)$. Generalizing the equation 4.1 for n sources and k microphones,

$$\begin{aligned}
x_1(t) &= \lambda_{11} s_1(t) + \lambda_{12} s_2(t) + \ldots + \lambda_{1n} s_n(t) \\
x_2(t) &= \lambda_{21} s_1(t) + \lambda_{22} s_2(t) + \ldots + \lambda_{2n} s_n(t) \\
&\vdots \\
x_k(t) &= \lambda_{k1} s_1(t) + \lambda_{k2} s_2(t) + \ldots + \lambda_{kn} s_n(t)
\end{aligned}$$

$$X = \Lambda S \tag{3.3}$$

Where Λ is the interference or bleed matrix, $X \in R^{k \times l}$ be the signals picked up by the k microphones and $S \in R^{n \times l}$ be the sources which need to be recovered. This equation 3.3 can be treated as a special case of source separation problem. The interference matrix Λ, X and S in the equation 3.3is given by,

$$\Lambda_{k \times n} = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \ldots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \ldots & \lambda_{2n} \\ \vdots & & & \vdots \\ \lambda_{k1} & \lambda_{k2} & \ldots & \lambda_{kn} \end{pmatrix},$$

$X_{k\times l} = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_k(t) \end{bmatrix}^T$ and $S_{n\times l} = \begin{bmatrix} s_1(t) & s_2(t) & \dots & s_n(t) \end{bmatrix}^T$ respectively. The mixture signal, denoted as $m(t) \in \mathbb{R}^l$ represents the composite audio signal resulting from the combination of all the sources in a manner specific to our experimental setup.

$$m(t) = \sum_{i=0}^{n} \beta_i s_i(t) = b^T S, \tag{3.4}$$

where b is the column vector of mixing coefficients.

In a typical concert setting, it is common practice to place a single microphone to capture each individual source, resulting in $k = n$, where k represents the number of microphones and n denotes the number of sources. This assumption implies that each microphone predominantly captures its dedicated source, as indicated by the presence of a larger λ_{ii} (self-influence) compared to λ_{ij} (cross-influence) in the bleed matrix Λ. However, it is worth noting that in some scenarios, multiple microphones may be used to capture the single source. For example, two microphones can be used to capture left and right side of mridangam, and in such cases, we can average the signals captured by these microphones. Given X and $m(t)$, our objective is to estimate S, Λ, and b which are the bleed reduced sources, bleed matrix and mixing coefficients.

3.2.1 Hidden Information

It is important to note that all microphone recordings are captured simultaneously, with each microphone picking up interference from the others. This implies that the same source is present in all microphone recordings but at varying levels of perception. Each source will dominate its dedicated microphone, while appearing in the background of other microphone recordings. This hidden information forms the basis for all the approaches proposed below.

3.3 Learning-free Optimisation Algorithm

Indeed, Equation 3.3 can be formulated as a constrained optimization problem to estimate the source signals S. It's important to acknowledge that Equation 3.3 represents an underdetermined problem where the number of unknowns exceeds the number of equations, leading to the possibility of multiple solutions. In light of this, we introduce various valid assumptions and constraints to effectively delimit the solution space. These constraints serve to guide the optimization process and ensure that any local minima encountered during optimization guarantees bleed reduction.

The most important assumption is that there exists at least one microphone where only one source is dominant. For simplicity, we assume that the number of sources n and the number of microphones k are equal ($k = n$). In this case, the diagonals of the interference matrix Λ should be greater than its off-diagonals due to this assumption.

By incorporating these assumptions and constraints into the optimization framework, we enhance the robustness of our approach in mitigating bleed effects and ultimately lead to a more reliable and effective solution for estimating the source signals S.

Problem statement: *minimize* $||X - \Lambda S||_F^2 + ||m - b^T S||_F^2$ *with respect to* Λ*,* S *and* b *subject to constraints:*

1. $\Lambda \neq I$ ▷ I is the identity matrix
2. $\lambda_{ii} > \lambda_{ij}$
3. $\gamma_1 \leq \lambda_{ij} \leq \gamma_2 \ \forall i \neq j$

Where $\|.\|_F^2$ is the frobenius norm. Since the mixture $m(t)$ contains the information about the sources, it is utilised and serves as the condition to converge at a suitable solution. The Λ should not be an Identity matrix I. If $\Lambda = I$, then $X = S$ which means that the system will not remove any sort of bleed. Another assumption is that for every microphone its corresponding source is dominant. Finally, to restrict the solution within the solution space, we project all the other mixing coefficients in Λ to be within some range (γ_1, γ_2).

This problem can be solved using the alternating minimisation approach by deriving the update rules for Λ, S and b.

3.3.1 Gradient Calculations

To calculate the gradients ∇_Λ, ∇_S, and ∇_b the objective function is rewritten using the trace properties.

$$\begin{aligned} ||X - \Lambda S||_F^2 &= tr\left((X - \Lambda S)^T (X - \Lambda S)\right) \\ &= tr\left(X^T X - X^T \Lambda S - S^T \Lambda^T X + S^T \Lambda^T \Lambda S\right) \\ &= tr(X^T X) - 2tr(S^T \Lambda^T X) + tr(S^T \Lambda^T \Lambda S) \end{aligned}$$

Similarly,

$$||m - b^T S||_F^2 = tr(m^T m) - 2tr(S^T bm) + tr(S^T bb^T S)$$

Using appropriate trace properties and matrix calculus following update rules are derived.

Update rule for Λ:

Taking the gradient of the objective function with respect to Λ and equating it to zero, we get

$$
\begin{aligned}
\nabla_\Lambda = \frac{\partial}{\partial \Lambda}(||X - \Lambda S||_F^2 + ||m - b^T S||_F^2) &= 0 \\
0 - 2XS^T + 2\Lambda SS^T + 0 &= 0 \\
-2(X - \Lambda S)S^T = -2S(X - \Lambda S)^T &= 0 \\
SX^T - SS^T\Lambda^T &= 0 \\
(SS^T)\Lambda^T &= SX^T \\
\Lambda(SS^T) &= XS^T \\
\Lambda &= (XS^T)(SS^T)^{-1}
\end{aligned}
$$

To avoid singularity issues while updating the parameter Λ, a small number η is added as,

$$\Lambda = (XS^T)(SS^T + \eta I)^{-1} \tag{3.5}$$

Update rule for S:

Similar to the update rule for Λ, we can derive the update rule for S as follows,

$$
\begin{aligned}
\nabla_S = \frac{\partial}{\partial S}(||X - \Lambda S||_F^2 + ||m - b^T S||_F^2) &= 0 \\
0 - 2\Lambda^T X + 2\Lambda^T \Lambda S + 2(bb^T S - bm) &= 0 \\
-2\Lambda^T(X - \Lambda S) + 2(bb^T S - bm) &= 0 \\
-\Lambda^T X + \Lambda^T \Lambda S + bb^T S - bm &= 0 \\
(\Lambda^T \Lambda + bb^T)S &= bm + \Lambda^T X
\end{aligned}
$$

re-arranging we get,

$$S = (\Lambda^T \Lambda + bb^T)^{-1}(bm + \Lambda^T X) \tag{3.6}$$

Update rule for b:

Following the similar procedure as above,

$$\begin{aligned}
\nabla_b = \frac{\partial}{\partial b}(||X - \Lambda S||_F^2 + ||m - b^T S||_F^2) &= 0 \\
0 - 2Sm^T + 2S(S^T b) &= 0 \\
-2S(m^T - S^T b) &= 0 \\
-Sm^T + SS^T b &= 0 \\
SS^T b &= Sm^T \\
b &= (SS^T)^{-1}(Sm^T)
\end{aligned}$$

To avoid singularity issues, the parameter η is added similar to the update rule of Λ,

$$b = \left(SS^T + \eta I\right)^{-1}(Sm^T) \tag{3.7}$$

3.3.2 Algorithm

The algorithm takes the bleed sources X and the mixture $m(t)$ as the input and outputs the bleed reduced source $\hat{S}$, interference or bleed matrix Λ and the mixing coefficients b. The complete workflow is shown in Figure 3.2.

Algorithm 2 Time-domain Optimization Algorithm for Bleed Reduction

```
Inputs: $X \in \mathbb{R}^{k \times l}$ and $m \in \mathbb{R}^l$
Initialize: $\Lambda \leftarrow I$
Initialize: $S \leftarrow X$
Initialize: $b \leftarrow [1, 1, ...1]^T \in \mathbb{R}^l$
while $||X - \Lambda S||_F^2 + ||m - b^T S||_F^2 \geq \epsilon$ do
    $\Lambda \leftarrow (XSS^T)(SS^T + \eta I)^{-1}$            ▷ A update rule
    $\Lambda \leftarrow projection(\Lambda)$
    $S \leftarrow (\Lambda^T \Lambda + bb^T)^{-1}(bm + \Lambda^T X)$            ▷ S update rule
    $b \leftarrow (SS^T + \eta I)^{-1}(Sm^T)$            ▷ b update rule
end while
```

The constraints have been implemented as projection on Λ where the values of λ_{ij} are projected in the range of (γ_1, γ_2) for off-diagonals and $\lambda_{ii} > \lambda_{ij}$. Projection involves finding the point in a feasible set that is closest to a given point. Mathematically, the projection of a point

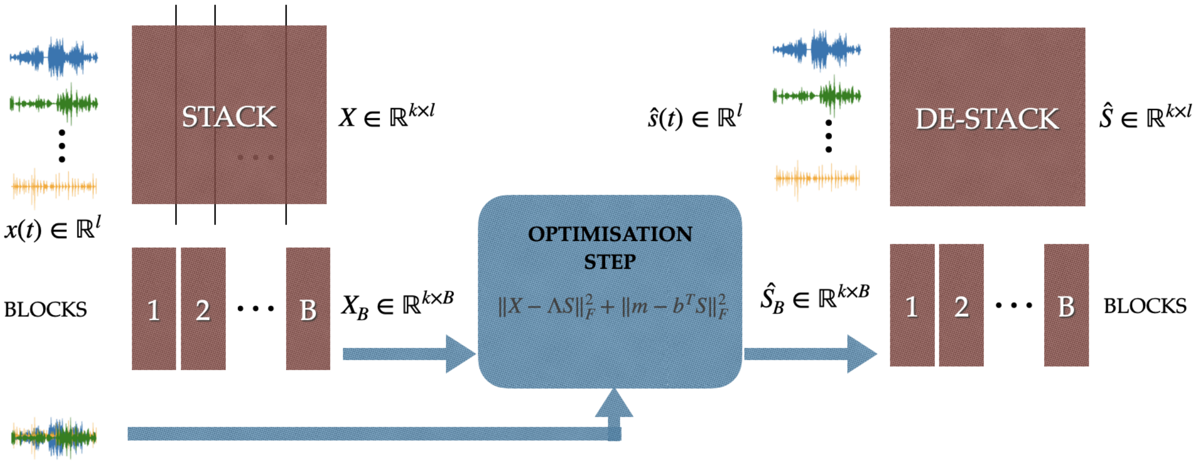

STACK
$X \in \mathbb{R}^{k\times l}$
$x(t) \in \mathbb{R}^l$
BLOCKS
1
2
···
B
$X_B \in \mathbb{R}^{k\times B}$
OPTIMISATION STEP
$\|X - \Lambda S\|_F^2 + \|m - b^T S\|_F^2$
$\hat{s}(t) \in \mathbb{R}^l$
DE-STACK
$\hat{S} \in \mathbb{R}^{k\times l}$
$\hat{S}_B \in \mathbb{R}^{k\times B}$
1
2
···
B
BLOCKS

Figure 3.2: Working procedure of the proposed optimization algorithm

x onto a set C is defined as:

$$projection_C(x) = \arg\min_{y\in C} \|x - y\|$$

where $\|x - y\|$ is the Euclidean distance between x and y.

3.3.3 Experiments

Datasets: We utilize the same dataset setup as that used for the CAE in Section 3.1.2. The artificially stimulated interference of MUSDB18HQ is utilized to evaluate the optimization algorithm.

As discussed in Section 3.1.2, data were generated by interfering a given stem with the other three stems, each reduced by 20 dB, similar to the approach used for the CAE model. The stems were chosen from the same track, and this process was repeated for all the tracks in the training subset of MUSDB18HQ, resulting in interfered versions of each stem. Unlike the CAE, where each stem was treated separately, all four stems have been combined into a single input for the optimization algorithm.

Hyperparameters: In our experimental setup, a single audio track containing all four stems is input into the algorithm. These stems are stacked together and represented as $X \in \mathbb{R}^{k\times l}$, where k represents the number of microphones and l denotes the length of the audio track. To facilitate processing, X is divided into multiple blocks, with each block containing 250 samples. The choice of block size can vary based on the available computational resources and preferences. In our experiments, we opted for sequential processing, where these blocks are passed one after the other through the algorithm.

Following the minimization process, the predicted source signals (S) are reconstructed and de-stacked to obtain the various bleed-reduced stems. During our experiments, we employed fixed values for certain hyperparameters: ϵ was set to 10^{-8} as the stopping criterion, η was set to 10^{-6}, γ_1 was set to 0.0001, and γ_2 was set to 0.4.

It's important to note that the choice of the (γ_1, γ_2) hyperparameters is a critical consideration. These values need to be selected carefully to ensure that the algorithm converges within the desired solution space. The tuning of (γ_1, γ_2) may require iterative experimentation to achieve optimal results for a given dataset and problem scenario.

3.3.4 Results

In our comparative analysis, we evaluated the performance of the proposed algorithm against two benchmark algorithms: Independent Component Analysis (ICA) and the state-of-the-art Kernel Additive Modelling for Interference Reduction (KAMIR) algorithm. For ICA, we implemented the efficient algorithm called FastICA [31] using python's sklearn library. Similar to the CAE model, obtaining BSS evaluation metrics is possible due to the availability of the clean sources.

The SDR metric serves as a crucial indicator of the effectiveness of our proposed model. Notably, our results demonstrated that the proposed algorithm significantly outperforms both ICA and KAMIR. In fact, our algorithm achieved an SDR improvement of nearly four times compared to the KAMIR algorithm, as depicted in Figure 3.3. This substantial enhancement underscores the superiority of our approach in mitigating bleed effects and successfully isolating source signals.

To further gauge the proximity of our solution to the ground truth, we computed the difference in l2 norm between the true bleed matrix and the predicted bleed matrix as described by equation,

$$closeness = ||\Lambda_{true} - \Lambda_{predicted}||^2 \tag{3.8}$$

This measure provides insights into the accuracy and fidelity of our algorithm's bleed matrix estimation, affirming the robustness and reliability of our proposed methodology. We see that the solutions converge to the optimal point in Figure 3.4.

Further, it is important to note that this optimization algorithm operates effectively only under restricted conditions, wherein the interference matrix Λ adheres to a pattern of dominant diagonal values and low off-diagonal values. However, this pattern is often not observed in

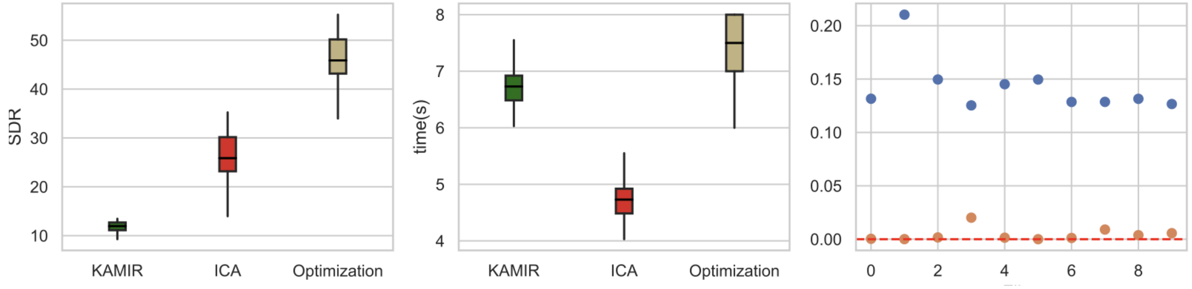
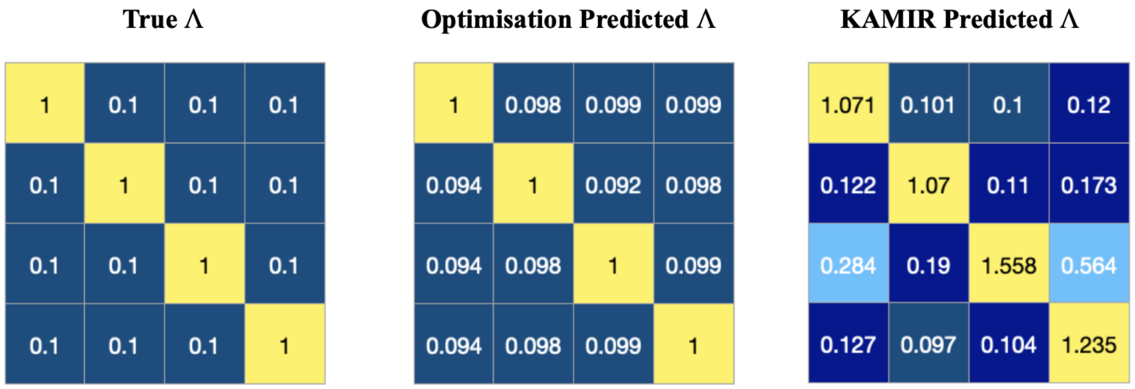


Figure 3.3: SDR, time taken and the difference l2 norm is compared with ICA and KAMIR



True Λ

1	0.1	0.1	0.1
0.1	1	0.1	0.1
0.1	0.1	1	0.1
0.1	0.1	0.1	1

Optimisation Predicted Λ

1	0.098	0.099	0.099
0.094	1	0.092	0.098
0.094	0.098	1	0.099
0.094	0.098	0.099	1

KAMIR Predicted Λ

1.071	0.101	0.1	0.12
0.122	1.07	0.11	0.173
0.284	0.19	1.558	0.564
0.127	0.097	0.104	1.235

Figure 3.4: Obtained interference matrix compared with KAMIR

practice. Hence, the high SDR obtained in this model is largely attributable to its specific setup.

3.3.5 Discussions, Limitations, and Directions

As illustrated in Figure 3.2, processing the entire input at once poses challenges due to the high sampling rate and lengthy duration, resulting in resource constraints. To address this issue, we partition the audio into blocks and process them individually. Since these blocks are independent of each other, they can be processed in parallel, leading to significant time savings.

Each block generates an interference matrix Λ. We conducted experiments on the Λ's of each block and observed that minor changes occur. Hence, after computing all the Λ's, we estimated their mean, median, minimum, and maximum interference matrices to be used for source estimation. However, this approach resulted in interference reduction with unwanted artifacts in the audio. Superior performance was achieved when using the dedicated Λ for each block.

Equations 3.3 and 3.4 are linear and do not consider non-linearity. Live recordings often contain more complex mixtures, including room impulse responses and time delays, which our approach does not address. Consequently, its performance in live recording conditions is not as expected. The following chapters will focus on extending the work to handle non-linear mixtures in live recordings.

3.4 Truncated UNet

So far, we have proposed two models for interference reduction, namely CAEs and optimization algorithms. As discussed in Section 3.1.3, it is worth reminding that, despite producing better results, the CAE model may have difficulty generalizing to new types of sources. For instance, the CAE for reducing interference in the vocal track may not work effectively for reducing interference in the drum track. Thus, a separate CAE has to be created to handle each track. Also, the CAE works on the magnitude spectrogram, which discards phase information. Recent source separation models, such as those proposed in [86, 55], reveal that time-domain approaches performs as same as spectrogram-based techniques. Also, as discussed in Section 3.3.5, the optimisation approach takes the linear form and works only for linear mixtures. In an attempt to learn nonlinear relationships, we replace the optimisation model with the neural network. Also, to better generalise the interference reduction for various sources and to avoid artifacts due to short-term processing, we propose our second learning framework called t-UNet.

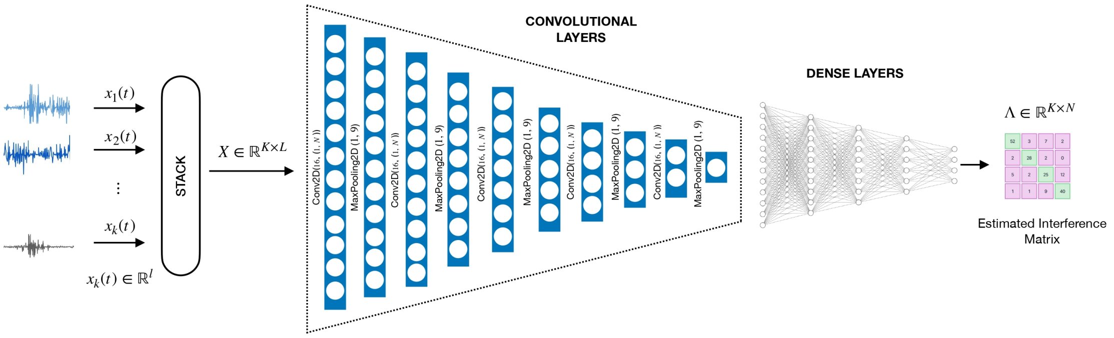


Figure 3.5: t-UNet for interference reduction.

The t-UNet also utilizes the same linear mixing mathematical model as described in Section 3.2. Thus, we consider the formulation with k microphones capturing n sources, where $k \geq n$, and the assumption that each source has at least one dedicated microphone. These dedicated microphone(s) for a source can be thought of as predominantly capturing the signal from that source, with signals from other sources contributing to a lesser extent. Therefore, the signal received at the kth microphone can be represented as:

$$x_k(t) = \lambda_{k1}s_1(t) + \lambda_{k2}s_2(t) + \ldots + \lambda_{kn}s_n(t), \tag{3.9}$$

where λ_{kn} represents the gain of the acoustic path from the nth source to the kth mic, and $s_n(t)$ represents the nth true source.

As denoted in Section 3.2, let $X \in \mathbb{R}^{k \times l}$ represent the time-aligned signal received by k microphones corresponding to an audio signal of l samples, and let $S \in \mathbb{R}^{n \times l}$ represent the true sources. Then the relationship between X and S is captured by the $\mathbb{R}^{k \times n}$ interference matrix Λ such that,

$$X = \Lambda S, \tag{3.10}$$

where,

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \ldots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \ldots & \lambda_{2n} \\ \vdots & & & \vdots \\ \lambda_{k1} & \lambda_{k2} & \ldots & \lambda_{kn} \end{pmatrix},$$

$X = \begin{bmatrix} x_1(t) & x_2(t) & \ldots & x_k(t) \end{bmatrix}^T$, and $S = \begin{bmatrix} s_1(t) & s_2(t) & \ldots & s_n(t) \end{bmatrix}^T$.

Instead of directly solving this under-determined problem, the approach is to learn the relationships among the interfered sources, which are captured by the microphone recordings. As discussed in Section 3.2.1, all microphone recordings share some common information regarding the sources. The goal is to learn these relationships to reduce interference among the stems. Leveraging the observation that all rows in X are interrelated, in t-UNet, the interference matrix Λ is learned in the time domain. By utilising relevant training data consisting of pairs of X and the corresponding Λ (equation 3.10), the relationship between the interfered signal and the individual sources is inferred by the network. Once Λ is inferred, the interference reduction is achieved by approximating the true sources S as

$$\hat{S} = \Lambda^{\dagger} X, \tag{3.11}$$

where † represents the pseudoinverse.

3.4.1 Network architecture

The network consists of an encoder and dense layers as shown in Figure 3.5. The encoder has five levels of convolution layers with size $(1 \times k)$ and max-pooling with kernel size (1×9). This ensures k-dimensional input throughout the network. The encoder is adapted from Wave-U-Net [86] without concat connections. The encoder encodes the relation between the sources and gives a meaningful feature representation. Five fully connected layers of size 512, 128, 64, 32 and kn are used. The output of the last layer is reshaped to give the interference matrix Λ.

3.4.2 Experimental evaluation

Datasets: Unlike the CAE and optimization algorithm, t-UNet utilizes a distinct artificially simulated MUSDB18HQ dataset. One limitation of the previously used dataset for the CAE and optimization (Section 3.1.2) is that the artificial mixtures are created by adding the dominant stem with other stems, each reduced by 20 dB. This could potentially result in same off-diagonals in the interference matrix Λ. For instance, the dataset with a 20 dB reduction used in the optimization algorithm is depicted as an interference matrix in Figure 3.4. This visualization illustrates that the diagonals are set to 1, while the off-diagonals exhibit the same value of 0.1, reflecting the impact of the 20 dB reduction in the artificial mixtures. Using such data for the proposed t-UNet may penalize the method. Hence, we utilized two types of datasets, each employing a different mixing strategy. The training and testing datasets are mixed in the same fashion, as described below.

Dataset I - MUSDB: We generate Λ such that the diagonal elements are dominant, falling within the range of $[0.6, 1]$, while off-diagonal elements are randomly selected from the range $[0.01, 0.5]$. This choice is informed by the characteristics of the MUSDB18HQ dataset, where the number of sources equals the number of microphones ($k = n$), as discussed in Chapter 2, Section 2.2.1. 10-second segments of stems from the same track are interfered according to the generated Λ. Audio segments having only zeros were not included, resulting in 2450 10-second segments for each stem.

Dataset II - MUSDBR: We also employ another type of mixing to simulate real-world effects such as time delays and room impulse responses on the above MUSDB dataset. This is achieved by utilizing `pyroomacoustics` [87]. The resulting dataset is termed MUSDBR. In this dataset, we utilize the same 2450 10-second segments for each stem and stimulate room impulse responses individually by simulating an artificial room. Then, the resultant stems are mixed using the same Λ matrix as the one used in the above MUSDB dataset.

Evaluation data: Similar to the training dataset setup, 10-second segments of stems from

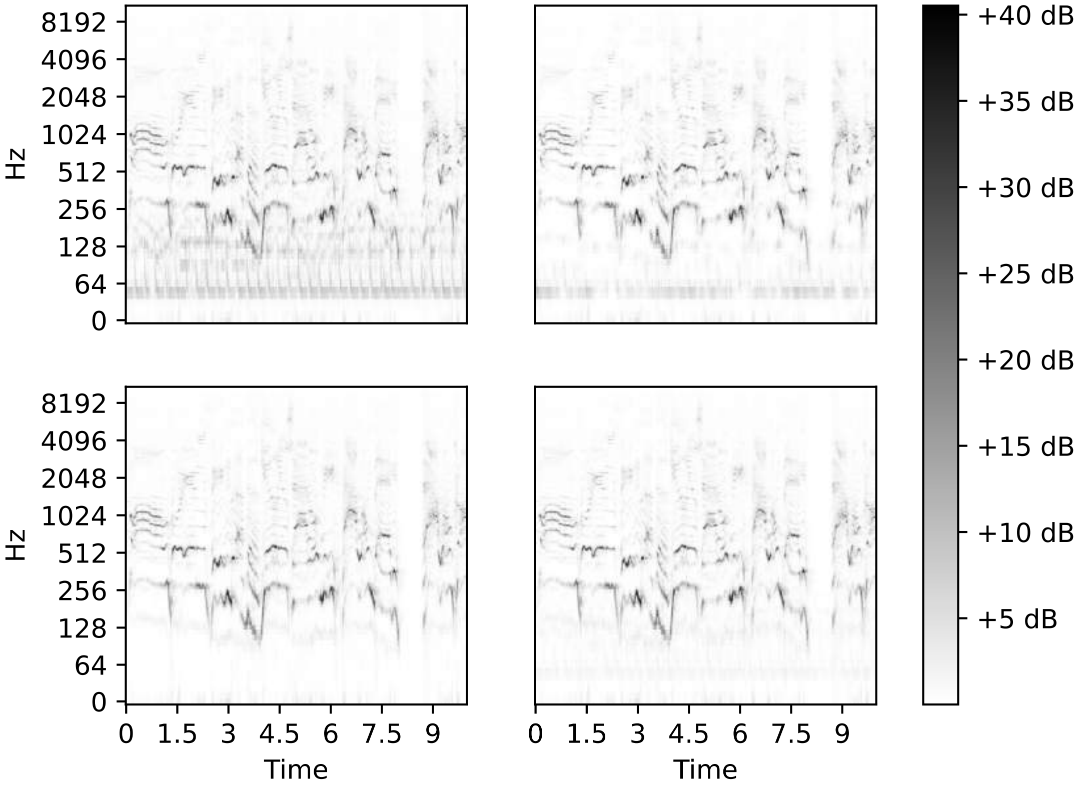


Figure 3.6: Spectrogram of a specific vocal example is shown. From top left clockwise: vocal with interference from bass, drums and others; KAMIR prediction; CAE prediction; and t-UNet prediction.

the same track in the MUSDB18HQ test data are interfered with using a random Λ. Two conditions are employed: high and low interference. Off-diagonal values between $[0.01, 0.2]$ are considered low interference, while those between $[0.2, 0.4]$ are regarded as high interference. This process is repeated to create evaluation data with both high and low interference condition. There are a total of 100 test tracks, with each track containing two 10-second segments: one with low interference and one with high interference. Similarly, MUSDBR is obtained by incorporating time delays and room impulse responses into the MUSDB18HQ test data. Thus a total of 200 test examples were utilised to evaluate the t-UNet model.

Hyperparameters: Mean square error loss function, Adam optimiser with a learning rate of 0.01, and batch size of 64 is used to train the proposed t-UNet model.

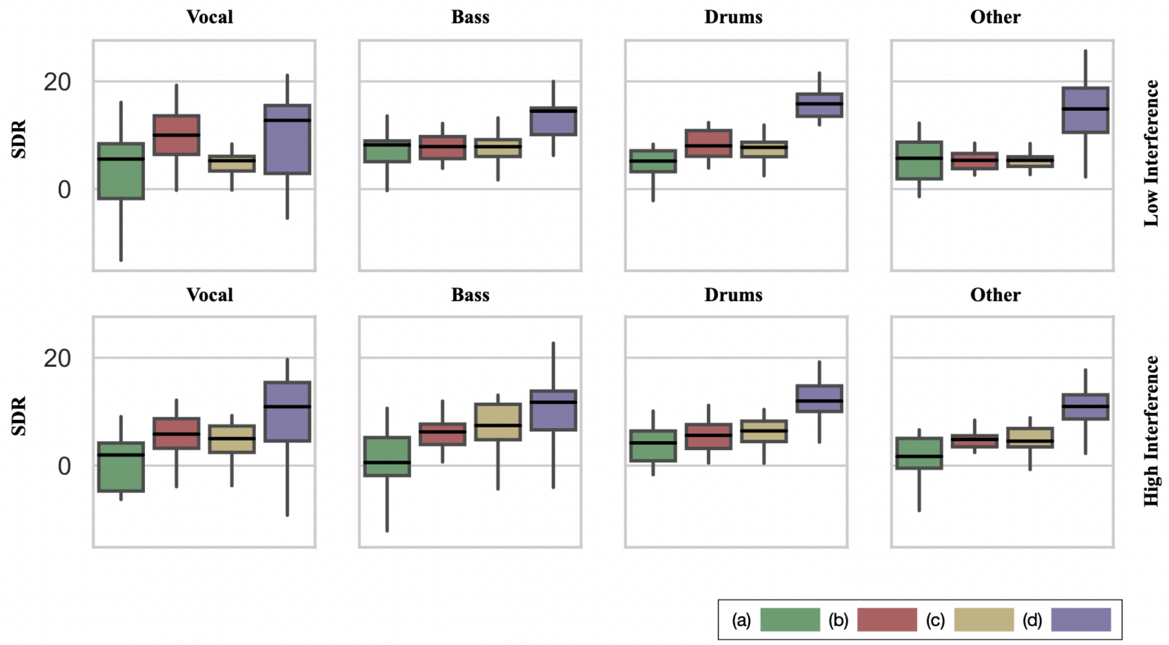


Figure 3.7: MUSDB: SDR comparison of two settings, low interference and high interference for different algorithms. (a) Reference SDR, (b) KAMIR, (c) CAE, and (d) t-UNet.

3.4.3 Results

Results on MUSDB dataset:

We compare the performance of the proposed models t-UNet and CAE with the state-of-the-art KAMIR algorithm [88]. Additionally, the reference SDR of the true source $s(t)$ and the interfered input $x(t)$ is also estimated. These results are summarized in Figure 3.7. It is apparent that any form of interference reduction improves the SDR. All algorithms, t-UNet and CAE are more successful in removing low interference than high interference. The CAE model performs at par or better than the KAMIR algorithm, except for vocals under low interference conditions. The t-UNet consistently performs better than both KAMIR and CAE in all evaluation conditions. Moreover, both CAE and t-UNet are much faster than the KAMIR, which is an iterative algorithms. It took on average 660.4s for KAMIR, 2.4s for CAE and 2.19s for t-UNet, for evaluating 100 test tracks, each of 10 seconds, on a 12GB GPU under Keras environment.

An example of the spectrograms of the resulting vocal outputs are shown in Figure 3.6. It can be seen that interference components are present in KAMIR, but are completely removed by both CAE and t-UNet. Figure 3.8 shows boxplots corresponding to the difference of Frobenius norms of the actual Λ and the Λ predicted by the methods compared. It can be seen that the t-UNet model estimates the interference matrix with high accuracy.

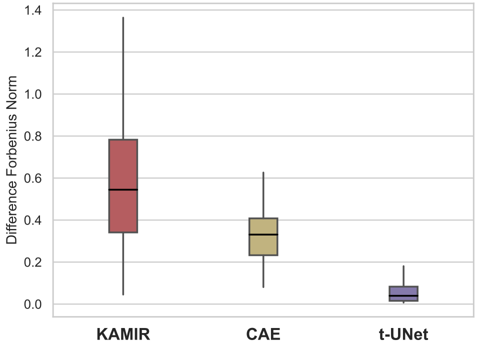


Figure 3.8: Difference of Frobenius norm of the true Λ with the predicted $\hat{\Lambda}$.

Results on MUSDBR dataset:

In terms of SDR, all models perform poorer on MUSDBR data. The proposed t-UNet model handles mixtures with room responses and time delays reasonably well. Models trained with MUSDB were fine-tuned with MUSDBR and showed improved performance. On the other hand, the proposed CAE model and KAMIR does not perform as well as the t-UNet. Fig 3.9 gives the average SDR across the four stems.

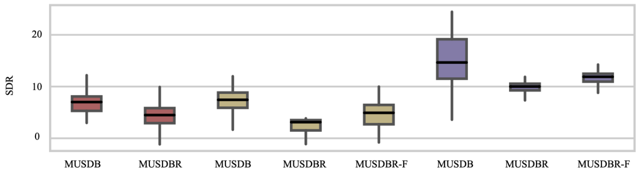


Figure 3.9: MUSDBR: SDR for different experiments for KAMIR, CAE, and t-UNet represented in Red, Yellow, and Magenta respectively. Suffix F represents models fine-tuned with MUSDBR.

3.5 Evaluation of music source separation performance

As stated earlier in Chapter 1, the interference reduction systems can be used as a pre-processing step before building music source separation (MSS) systems. Effective removal of interfering sources creates tracks of higher quality, resulting in better supervised MSS models. We evaluate the MSS performance of the recently proposed Wave-U-Net [86] using interference reduction of the two methods proposed in this thesis. Evaluation is performed on MUSDB, with MSS models trained separately on clean data, data with interference, and data pre-processed with the proposed methods.

Table 3.1 summarizes the results. It can be seen that using training data having interference brings down the MSS performance. Pre-processing with CAE or t-UNet before building MSS models improves performance.

	Clean	Interference	CAE Cleaned	t-UNet cleaned
SDR	2.32	0.96	1.72	2.03

Table 3.1: Music source separation performance.

Results show that the interference reduction methods help in improving the music source separation performances.

3.6 Discussion and Shortcomings

The results presented in the CAE and t-UNet result section (Section 3.4.3) are based on data in which interference, time delays, and room responses have been artificially created. It is known that in real-world live recordings, the interference could be more complex. CAE, Optimisation, and t-UNet are also evaluated on a few recordings from the Saraga dataset, which was discussed in Chapter 2, Section 2.2.2. Since these are live recordings, the clean sources are not available, and hence BSS metrics like SDR cannot be estimated. Preliminary listening tests after interference reduction seem to indicate that the interference is not completely removed. Factors such as domain mismatch (trained on MUSDB18HQ, evaluated on classical music) can be possible shortcomings. Although the optimization approach is replaced with t-UNet to capture non-linearity, its base mathematical formulation is linear 3.10. Thus, to make the model capture non-linearity, a simple linear mathematical formulation is not valid.

CHAPTER 4

Non-Linear Mixing Models for Interference Reduction

Chapter 4 presents the proposed neural network GIRNet, which aims to address the limitations of the models introduced in Chapter 3. It delves into the motivation behind GIRNet, its mathematical formulation, architectural details, graph representations, graph attention networks, and the experiments conducted to evaluate its performance. Additionally, the chapter conducts ablation studies to further enhance the model's effectiveness and presents insightful results.

Following a thorough exploration of GIRNet, the chapter employs it to enhance the source separation performance, particularly in the context of Indian classical Carnatic music. This application demonstrates the versatility and efficacy of GIRNet in real-world scenarios.

In the previous chapter, three models for interference reduction were proposed: Convolutional Autoencoder (CAEs), optimization algorithm, and truncated UNet (t-UNet). Their performance was evaluated on various datasets derived from the MUSDB18HQ dataset using artificial mixing. However, as discussed in Chapter 3, these models struggled to perform well on real live recordings. Although the truncated UNet (t-UNet) model showed improvement in realistic mixtures, its mixing process remained linear and failed to perform adequately in live recordings. This chapter extends the concept of t-UNet to nonlinear mixtures, which can be effectively applied to live recordings.

4.1 Mathematical Formulation

The earlier linear mixing model, introduced in Chapter 3, was proven to be less effective in real-world live recordings. The simple linear equation used in the proposed models could not capture the actual non-linearities present in the live recordings. Therefore, the linear mixing model is extended to a non-linear mixing model.

Let $x(t) \in \mathbb{R}^l$ be the interfered signal of length l that is picked up by the microphone placed to capture a single source. Assume that each source has atleast one microphone. Let $s(t) \in \mathbb{R}^l$ be the true sources that need to be recovered from the signal $x(t)$. For k microphones and n sources, we formulate the problem as: for $k \geq n$,

$$\begin{aligned} x_1(t) &= f(\mathbf{s_1(t)}, s_2(t), \ldots, s_n(t)) \\ x_2(t) &= g(s_1(t), \mathbf{s_2(t)}, \ldots, s_n(t)) \\ &\vdots \\ x_k(t) &= h(s_1(t), s_2(t), \ldots, \mathbf{s_n(t)}) \end{aligned} \tag{4.1}$$

where $f(.)$, $g(.)$, and $h(.)$ are unknown functions. It is also assumed that for each source, there is at least one microphone and for that microphone, its corresponding source is dominant, shown as bold in equation 4.1. In this section, for simplicity, we assume that the number of microphones is equal to the number of sources ($k = n$).

4.2 Using Graph Attentions

Designing a neural network for interference reduction presents unique challenges heavily reliant on the specific dataset. For instance, a model trained on vocal sources might excel with vocals but struggle with other instruments, rendering multiple models impractical. Furthermore, the data exhibits significant distribution shifts even within the same musical sources, such as vocals across various classical genres. These intricacies complicate the creation of a learning-based interference reduction framework. The objective becomes the development of a model capable of delivering satisfactory results across diverse sources. Therefore, inspired by the methodology in [89], we construct a network engineered to be dataset-agnostic while achieving substantial interference reduction performance.

Without losing context, the linear mixing model provided a strong foundation for addressing the interference reduction problem. In Section 4.1, we attempted to extend the linear mixing model to a non-linear mixing model. The primary objective was to capture the non-linearities by learning the relationships among the microphone recordings. As a result, we decided to explore Graph Neural Networks for this purpose.

Intuition: The motivation behind using Graph Neural Networks (GNNs) is that each microphone recording can be modeled as a node in a graph. By representing the microphone recordings as nodes in a graph, GNNs can effectively learn the non-linearities among them. This graph-based representation allows for a more effective capture of the relationships between the recordings compared to a Euclidean space. Once these non-linearities are learned as a graph, the information can be utilized to reduce interference.

In simpler terms, consider the linear mixing formulation in Equation 3.3 and Figure 1.3. Here, the values of λ_{ij}, where $i \neq j$, should ideally be kept as low as possible to prevent interference among the sources. However, in reality, there may still be some level of interference caused by non-zero λ_{ij} values, which need to be minimized for effective interference reduction.

This valuable insight can be leveraged to design a neural network by transforming the interference matrix Λ into a weighted adjacency matrix. This transformation effectively represents the problem as a graph, illustrated in Figure 4.1, and serves as the primary motivation for utilizing graph neural networks. In essence, we can integrate the linear mixing model into a non-linear mixing framework by employing graph neural networks.

In this context, we frame the problem as described in Equation 4.1, aiming to discern and learn the complex relationships among the inputs, ultimately providing reasonable estimates in the time domain. To capture the nonlinear relationships inherent among the sources, we propose an architecture called Graph-based Interference Reduction Network (GIRNet).

Network: Recent times have witnessed the emergence of various time-domain, end-to-end music source separation models [57, 90], showcasing performance comparable to spectrogram-based models [60, 56, 64]. To bypass the phase problem in spectrogram-based models, we adopt a time-domain U-Net architecture akin to the Wave-U-Net [57]. The Wave-U-Net model is a source

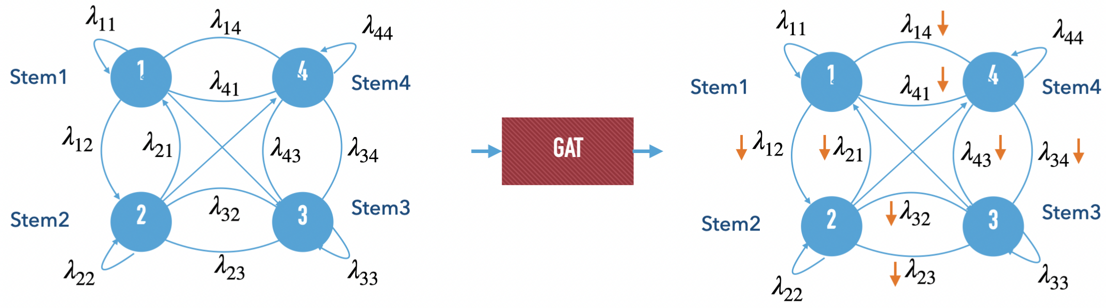


Figure 4.1: Problem modelled as a graph. Connections indicate the interference strength among the sources.

separation model, which accepts input mixtures and estimates their constituent sources. The principal difference between the Wave-U-Net and the proposed GIRNet resides in the latter's multi-channel input capability, capable of processing k microphone recordings simultaneously whereas, Wave-U-Net accepts a single mixture input. Wave-U-Net operates with 1D convolution, whereas GIRNet operates in 2D. Notably, GIRNet harnesses the power of dilated convolution and graph attention within its bottleneck, a feature absent in the Wave-U-Net architecture. While Wave-U-Net employs bilinear interpolation in its upsampling layer, GIRNet employs a general convolutional transpose layer for this purpose.

It has been substantiated in existing literature that dilated convolutions excel at capturing comprehensive global information due to their increased receptive field [91]. Furthermore, they have exhibited better performance when compared to RNN/LSTM models, particularly when utilizing an exponentially increasing dilated factor in speech enhancement contexts [92]. In a related study [58], dilated convolutions showcased enhanced performance in terms of source separation capabilities.

4.2.1 Architecture

The model architecture consists of four main components: the encoder, the Graph Attention Network (GAT), the bottleneck, and the decoder, all illustrated in Figure 4.2. The encoder incorporates four down-sampling blocks, each detailed in Table 4.1. Notably, the `dilation rate` exhibits an exponential growth pattern: 1, 2, 4, 16.

Complementing the encoder, the decoder mirrors its structure, featuring four up-sampling blocks. To facilitate information flow and preserve contextual features, the encoder and decoder are interlinked through skip connections. Beyond the encoder, the latent representation is converted into a graph-based representation and subsequently passed through a two-layer GAT

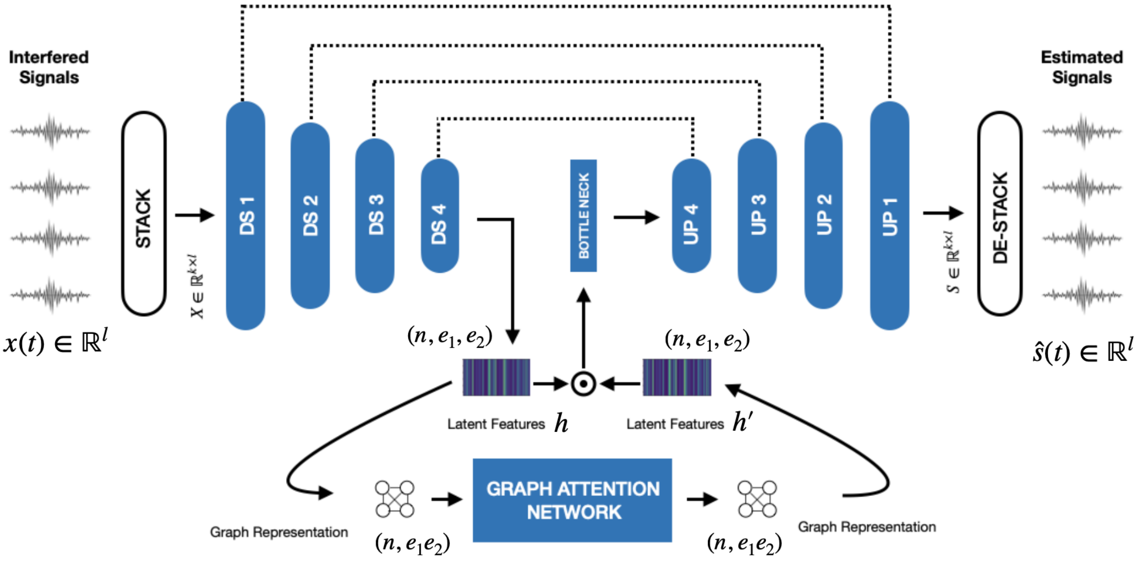


Figure 4.2: GIRNet: Graph-based Interference Reduction Network Architecture. The $\odot$ represents the Hadamadard product.

network. The GAT's output is reshaped to match the latent dimension.

Within the bottleneck section, two conv2d blocks are followed by batch normalization and leaky ReLU activations. Notably, the bottleneck receives the Hadamard product computed between the input and output of the GAT. This bottleneck output is then relayed to the decoder. Lastly, the decoder's output undergoes transformation through a single conv2dtranspose layer followed by a leaky ReLU activation, ultimately yielding the desired clean source.

4.2.2 Graph Attention Network

Convolutional graph embeddings have found application in speech enhancement, as evidenced by their usage in [93], where notable performance enhancements were demonstrated. Empirical evidence supports the notion that graph embeddings contribute to improved performance in audio-related tasks.

In our proposed model, we introduce a graph attention embedding technique, drawing inspiration from the architecture of the graph attention neural network [94]. Our rationale for adopting these attention layers is to learn the relationships among the given sources. We approach each source (in its latent state) as an individual node within the graph and strive to encapsulate the relationships between these nodes through the employment of attention mechanisms.

Downsampling Block	Upsampling Block
Conv2D(filters, (4, 4), padding="same", dilation rate)	Conv2DTranspose(filters, (4, 4), strides=(1, 1), padding="same")
Conv2D(filters, (4, 4), padding="same", dilation rate)	Batch Normalization()
Batch Normalization()	LeakyReLU(0.3)
LeakyReLU(0.3)	Concatenate()
MaxPooling2D((1, 2), strides=None, padding="valid")	Conv2D(filters, (4, 4), padding="same", dilation rate)
Dropout(0.1)	Conv2D(filters, (4, 4), padding="same", dilation rate)
	LeakyReLU(0.3)

Table 4.1: Details of Downsampling and Upsampling Blocks

Graph Representation

The output of the encoder block, characterized by dimensions (n, e_1, e_2), undergoes a transformation into dimensions (n, e_1e_2). These transformed features can be envisaged in the form of a graph denoted as $G = (v, \epsilon)$, where n nodes are represented as entities of dimension $\mathbb{R}^{e_1e_2}$ and n represents the number of sources.

The interference matrix Λ is introduced with the help of a $n \times n$ weighted adjacency matrix in the graph. A higher value in the adjacency matrix would indicate a stronger connection (more interference) between the corresponding audio signals, while a lower value would indicate a weaker connection (less interference). For interference reduction, the adjacency matrix would have unit weights on the diagonal (self-edges) and very low values (other edges) on the off-diagonal entries.

Attention Network

Consider the input to the graph network as $\vec{h} = \{\vec{h_1}, \vec{h_2}, ..., \vec{h_n}\}$, where each $\vec{h_i} \in \mathbb{R}^{e_1e_2}$. The number of nodes in the graph is equal to the number of sources. Through the operation of the network, it produces an output denoted as $\vec{h'} = \{\vec{h'_1}, \vec{h'_2}, ..., \vec{h'_n}\}$. The attention operation can be described by,

$$\vec{h'_i} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h_j}\right) \tag{4.2}$$

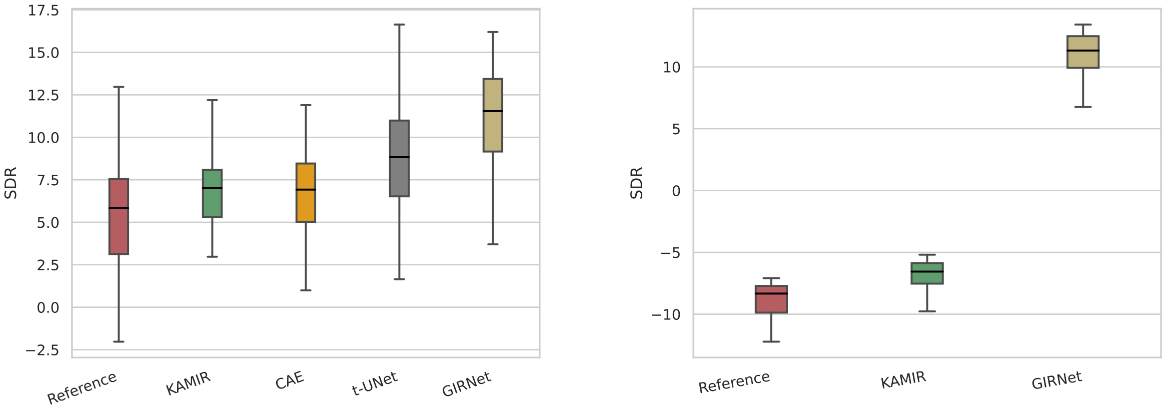
where $\sigma(.)$ is an activation function. $\mathcal{N}_i$ is some neighbourhood of node i in the graph. Let the attention mechanism, denoted as $a : \mathbb{R}^{e_1e_2} \times \mathbb{R}^{e_1e_2} \to \mathbb{R}$, take the form of a single-layer feed-forward neural network. This mechanism is parameterized by a weight vector $\mathbf{W}$ and employs the Leaky ReLU activation function with a slope of 0.2. The coefficients α computed by the attention mechanism are represented as:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU\left(\vec{a}^T\left[\mathbf{W}\vec{h_i}\|\mathbf{W}\vec{h_j}\right]\right)\right)}{\sum_{k\in\mathcal{N}_i} exp\left(LeakyReLU\left(\vec{a}^T\left[\mathbf{W}\vec{h_i}\|\mathbf{W}\vec{h_k}\right]\right)\right)} \tag{4.3}$$

where $\|$ is concatenate operator. $i, j \in \{1, 2, ..., n\}$. The R multi-head graph attention layer is computed as follows:

$$\vec{h_i'} = \sigma\left(\frac{1}{R}\sum_{r=1}^{R}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^r\mathbf{W}^r\vec{h_j}\right) \tag{4.4}$$

The graph data are propagated through the two-layer graph attention network.



(a) Linear mixtures

(b) Reverberant mixtures

Figure 4.3: The mean SDR of the models under linear mixture and reverberant mixture datasets.

4.3 Experiments and Results

The proposed models are evaluated in different scenarios. We begin by evaluating the interference reduction performance of simple (though non-realistic) linear mixtures. Then we evaluate the performance in more realistic (though simulated) reverberant environments. Finally, we evaluate the performance on live recordings from live concerts.

4.3.1 Datasets

For training and testing, three types of datasets are used, which are distinct from the datasets used to train and test the previously proposed models: CAE, optimization algorithm, and t-UNet.

Linear Mixtures (LM): The standard MUSDB18HQ dataset [25] train set has been utilized. Similar to the MUSDB dataset used for training t-UNet in Chapter 3, Section 3.4.2, the stems within tracks are segmented into 10-second chunks. If any stem within a chunk remains silent, that particular chunk is omitted, resulting in 1588 files. Interference within the same track is artificially created using the linear relationship between the sources as described in Equation 3.3. Random Λ matrices are generated and used to linearly mix the sources. This dataset is termed Linear Mixtures (LM).

Reverberant Mixtures (RM): The same 1588 files are taken, and instead of mixing artificially using Equation 3.3, pyroomacoustics [95] is used to simulate real-world interference effects. Artificial rooms of different sizes are created and audio is simulated to capture the non-linear mixing. This process incorporates time delays, reverberation, and room impulse responses, resulting in mixing that more closely resembles realistic data. This dataset is referred to as Reverberant Mixtures (RM).

Note that this dataset is distinct from the MUSDBR dataset introduced in Chapter 3, Section 3.4.2. In the MUSDBR dataset, stems were individually simulated with room impulse responses and mixed linearly using Λ. However, in this dataset, the mixing process itself is convolute and non-linear. All four stems are simulated together in the artificially created room, and the microphone outputs are directly captured, resulting in interfered sources.

Live Recordings: The Saraga Dataset which was collected from various live concerts has been utilised. As discussed in Chapter 2, Section 2.2.2, the dataset encompasses six microphone recordings that correspond to three sources: primary and secondary vocal, violin, mridangam left and right, and ghatam. Notably, the ghatam source is available only for select examples and has been consequently excluded. To consolidate the sources, the primary and secondary vocals are merged to form the vocals category, while the mridangam left and right are merged to constitute the mridangam source. Consequently, the final three distinct sources are vocals, mridangam, and violin.

We have trained the GIRNet with the RM dataset and tested it using LM, RM, and live recordings of the Saraga dataset. 200 files of 10 seconds each, two from each example of MUSDB18HQ, have been utilized for test data in both LM and RM.

4.3.2 Results

The GIRNet model is trained for 30 epochs with a Mean Square Error loss function, Adam optimizer with a learning rate of 0.01, and a batch size of 1 due to limited computational resources available.

Results on Linear Mixtures

The model is compared with KAMIR, CAE, and t-UNet. A summary of the results is presented in Figure 4.3. The average source-to-distortion ratio (SDR) across the stems is computed and shown. Additionally, the reference SDR, which is the SDR taken between the true and interference sources, is included. A value greater than the reference implies a reduction in interference. The results indicate the superiority of the proposed model. The GIRNet performs better than the previous proposed models CAE, t-UNet, and the state-of-the-art KAMIR.

Results on Reverberant Mixtures

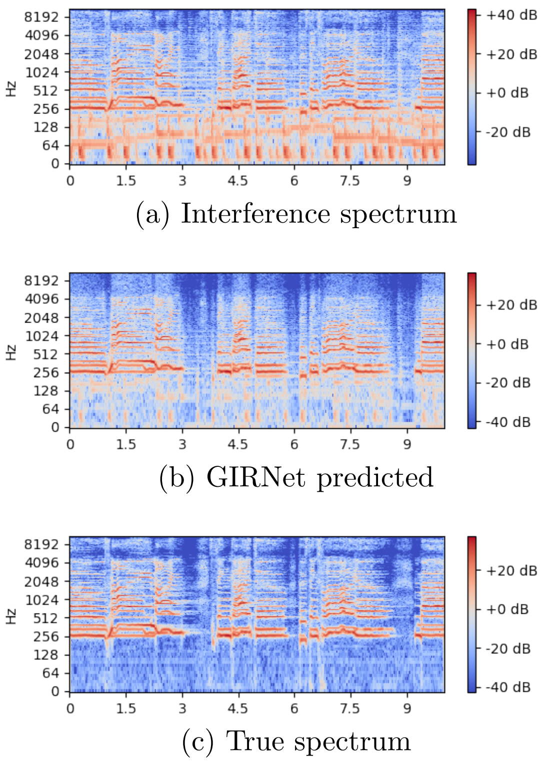
Since the CAE and t-UNet are designed only for linear mixtures, they have not been used in reverberant mixtures. The model is compared with KAMIR and the reference. Performance of both KAMIR and the reference has reduced considerably in this non-linear mixing. The proposed model works and attains a significant SDR, as shown in Figure 4.3. The GIRNet achieved 300% improvement with respect to KAMIR.

An example of the vocal source with interference, the GIRNet predicted source, and the true clean source without interference are shown in Figure 4.4. It is evident that the neural network effectively removes the other instrument sounds from the given vocal spectrum.

Results on Live Recordings

The same model, without any fine-tuning, was applied to the Saraga dataset to reduce interference. It's important to note that the model was trained with the RM dataset, which is a part of MUSDB18HQ—a dataset consisting of Western pop music. It was then tested on Indian classical music with three different stems. In other words, the trained model was tested on out-of-domain samples.

Since there is no ground truth for live recordings, obtaining SDR is not possible. Thus a listening test has been conducted. A preliminary listening test indicates that the neural network effectively suppresses the interference but not completely removing it. This is likely due to the domain shift and the out-of-domain inputs such as mridangam and violin, which have not been seen during training.



(a) Interference spectrum

(b) GIRNet predicted

(c) True spectrum

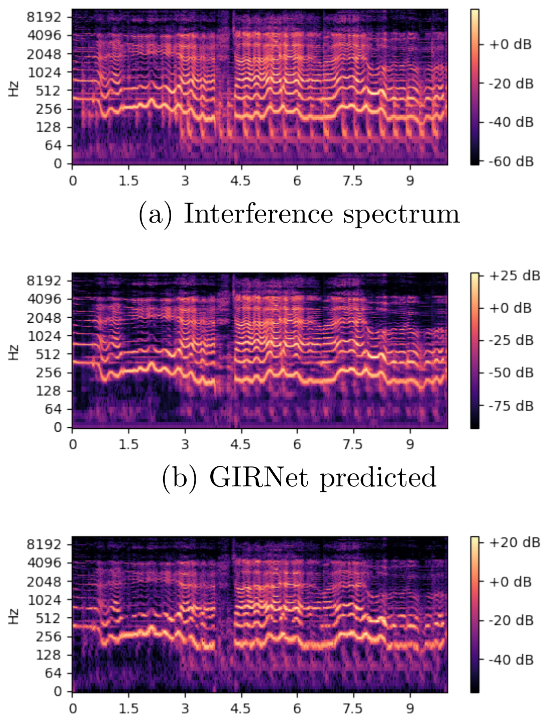
Figure 4.4: Example of vocal stem with interference, GIRNet predicted, and the true clean source without interference from the RM dataset.

The figures 4.4 and 4.5 represent the spectrum of the vocal source with and without interference from RM and the Saraga dataset, respectively. In Figure 4.5 (b), we can observe that the spectral components of other instruments are still present but comparatively reduced. Based on this result, to further remove the presence of other spectral components in live recordings, we introduce post-processing.

4.3.3 Post Processing

As detailed above, the proposed GIRNet model is trained with RM data from MUSDB18HQ, which consists of Western pop music, and tested with the Indian classical Saraga dataset. Consequently, the model is expected to perform less effectively due to the out-of-domain test samples. Initial experiments suggest that the network can perform adequately to some extent. To improve the model's performance on out-of-domain data, post-processing techniques are implemented.

Here, $\hat{s}_i(t)$, with $i = 0, 1, ..., n$, represents the sources predicted by the neural network, while $s_i'(t)$ stands for the sources following the post-processing stage. Notably, both the raw waveform input (with interference) and the resultant output undergo transformation into a suitable time-



(a) Interference spectrum

(b) GIRNet predicted

(c) After postprocessing

Figure 4.5: Example of vocal stem with interference, GIRNet predicted, and after postprocessing spectrum from the Saraga dataset.

frequency representation. The $X_i(f,t)$ and $\hat{S}_i(f,t)$ is the STFT of $x_i(t)$ and $\hat{s}_i(t)$ respectively, where $X_i(f,t), \hat{S}_i(f,t) \in \mathbb{C}^{T \times F}$.

The post-processing phase comprises two key steps: the generation of hard masks and spectral subtraction. The model's functionality extends to suppressing extraneous instrument sounds that are not dominant within a given context. This is achieved by detecting spectral variations and subsequently constructing a hard mask. The hard mask selectively retains solely the predominant instrument's sound while discarding the rest. Subsequently, this discerned information is harnessed to execute spectral subtraction, ultimately culminating in the recovery of clean, interference-reduced sources.

Hard Masks

The absolute difference of the magnitude spectrum is taken across the interfered and the predicted signal as,

Stems	IRQ				AQ			
	Mean		Median		Mean		Median	
	KAMIR	GIRNet	KAMIR	GIRNet	KAMIR	GIRNet	KAMIR	GIRNet
Vocal	3.71	3.41	4	3.5	3.71	3.25	4	3
Mridangam	3.73	3.53	4	4	3.45	3.28	3	3
Violin	3.68	3.45	4	3	3.86	3.08	4	3

Table 4.2: Listening Test: KAMIR vs GIRNet on Interference Reduction Quality (IRQ) and Audio Quality (AQ) Metrics

$$change_i(f,t) = |X_i(f,t)| - |\hat{S}_i(f,t)| \tag{4.5}$$

Then a threshold (tr) is chosen to compute a hard mask as follows:

$$mask_i = \begin{cases} 1, & \text{if } change_i(f,t) > tr \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

Spectral Subtraction

For every source n, the hard mask is computed. The individual source is recovered by equation 4.7.

$$S'_i(f,t) = |X_i(f,t)| - \sum_{j=0,j\neq i}^{n} |X_j(f,t)| \ mask_j \tag{4.7}$$

Then, the phase of $x_i(t)$ is used to convert back into the time-domain.

To visualise the effect of postprocessing, an example of the vocal source from the Saraga dataset with interference, the GIRNet predicted source, and the after-postprocessing spectrum are shown in Figure 4.5.

To measure the effectiveness of the model, a listening test was conducted with 44 participants, and the findings are outlined in Table 4.2. Each participant was tasked with providing ratings ranging from 1 to 5, based on two key metrics: audio quality and interference reduction quality.

For comparative assessment, our model was compared with the KAMIR algorithm. The results underscore that both models excel in terms of interference reduction. The obtained results are competitive, and KAMIR shows a slight edge over the GIRNet. This listening test provides valuable insights into the model's performance, highlighting its favourable attributes and comparative advantages over existing methods.

Without GIRNet: The post-processing cannot be applied without GIRNet due to Equation

No of files	KAMIR	GIRNet
200	1320.8	4.2

Table 4.3: Comparison of time taken for the proposed method in seconds (Inference time)

4.5. Nonetheless, we conducted tests on the live recordings using only the post-processing, by bypassing Equation 4.5 and directly computing the hard mask while applying an arbitrary threshold t_r. The resulting audio quality was poor, and it failed to eliminate any interference.

4.3.4 Time Complexity

The tabulation of the average time required for the GIRNet to evaluate 200 test tracks, each spanning 10 seconds while utilizing a GPU with 12GB of memory is presented in Table 4.3.4.

4.4 Discussions and Ablation Study

The ablation study is threefold: (1) To emphasize that multiple stem input aids in interference reduction more than single input dedicated models, (2) To study the GIRNet with and without the graph attention network as a bottleneck, and (3) By using various graph representations through different adjacency matrices.

The RM dataset is utilized for this study as it allows for quantitative comparison.

4.4.1 Single vs multiple stem Input

A specialized model utilizing single-channel input was trained and subjected to testing. The models are trained for vocal, bass, drums and other sources separately. The outcomes underscore the efficacy of the proposed GIRNet. The SDR of the models is tabulated in Table 4.4. Specifically, the GIRNet model surpasses the dedicated single-channel models in terms of performance, highlighting its superior capabilities. It shows that the idea of learning the relationship among the sources in interference reduction is helpful.

	Vocal	Bass	Drums	Other	Overall
Single	10.11	9.52	9.01	11.7	9.25
Multi	12.53	11.97	11.77	13.0	12.31

Table 4.4: Median SDR of single channel dedicated models vs the multiple stem GIRNet network

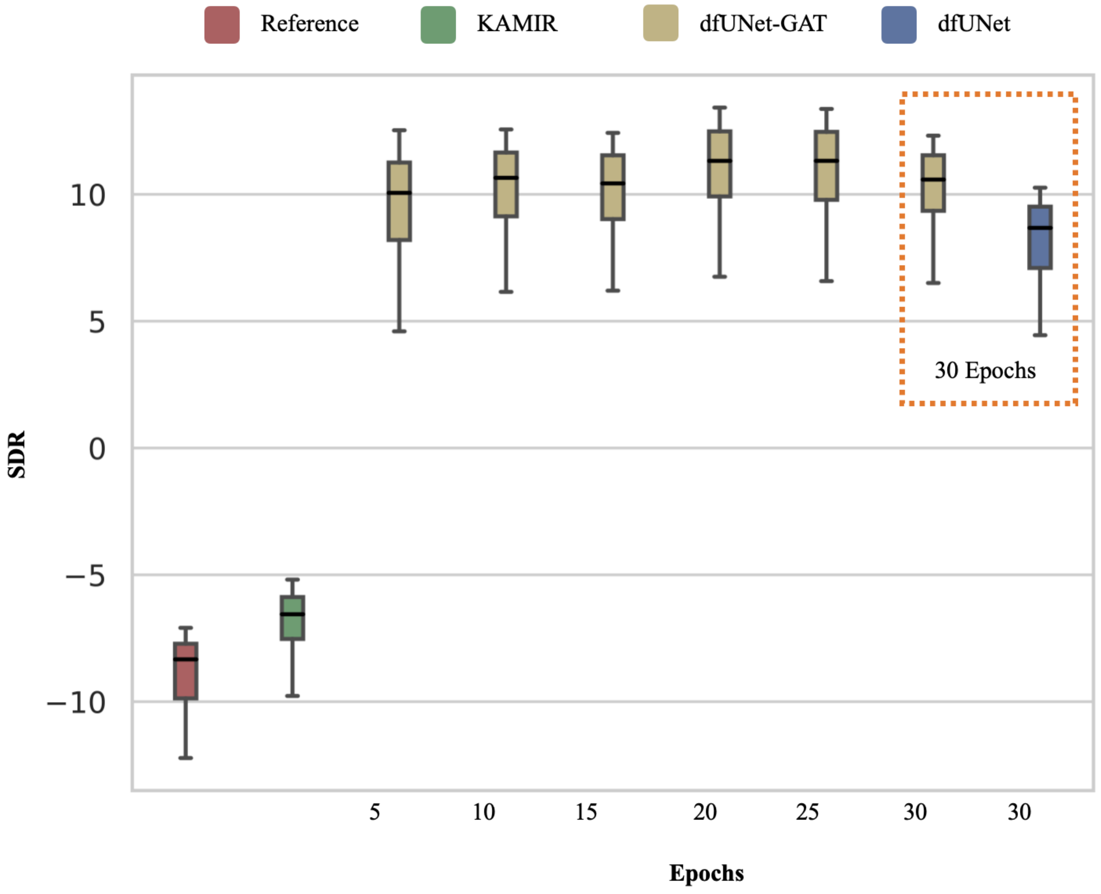


Figure 4.6: Performance trends of epoch vs SDR of GIRNet with and without graph attention

4.4.2 GIRNet with and without graph

Due to computational limitations, the GIRNet is trained for only 30 epochs. We utilised 12GB GPU memory and tensorflow-keras environment. To ensure fair comparisons and to study the convergence behaviour, the average SDR is plotted for every five epochs and compared with and without the graph attention network. The results are shown in Figure 4.6. The average SDR saturates after 20 epochs, and the GIRNet without graph performs slightly poorer than the GIRNet with a graph.

4.4.3 Different graph representations

To reiterate, the use of graphs in the interference reduction problem involves mapping each microphone recording as nodes of the graphs. This facilitates the effective incorporation of the Λ factor as a weighted adjacency matrix. However, different graph representations have been widely used in various other audio-related problems such as speech enhancement, etc. As part

Adjacency Matrix	A1	A2	A3
Average SDR	10.05	9.16	10.61

Table 4.5: Average of median SDR across all the stems with different adjacency matrices

of an ablation study, an alternative approach could involve constructing the graph using $e_1 \times e_2$ nodes with n dimensions, utilizing various adjacency matrices and variations in the $e_1 \times e_2 \times e_1 \times e_2$ order adjacency matrix. Here, n represents the number of sources, and e_1 and e_2 denote the dimensions of the latent space.

Adjacency Matrix:

Efforts to optimize the training of GIRNet, due to its notably time-consuming nature, led us to explore enhancements while maintaining network performance. A key focus was on the GAT, which emerged as the primary computational bottleneck, contributing significantly to prolonged training durations. To enhance the efficiency of the GAT component, we conducted an investigation into the initialization of the adjacency matrix, considering three distinct approaches.

- The first, denoted as *A1*, is characterized by a matrix of ones, and zeros for the diagonals.
- The second, *A2*, involves computing cross-similarities between latent features and utilizing a threshold value, t. If $t > 0.5$, the value is set to 1; otherwise, it is set to 0.
- The third approach, *A3*, leverages the assumption that musical activity at a specific time instant has limited direct influence on other time instants within our context. Hence, we introduced a hyperparameter, p, which designates the placement of ones. Specifically, the first row contains p ones and zeros elsewhere, and subsequent rows feature a rightward shift by one element. In our experimental setup, we selected $p = 100$.

The results were tabulated in Table 4.5. The models were trained for a limited number of epochs due to computational constraints. Initial findings suggest that the model utilizing $A3$ yields the best performance. However, further training for additional epochs is necessary to reach a conclusive outcome. Therefore, at this point of time the results remain inconclusive.

4.5 Conclusion

The Chapter 4 introduced a neural network approach - the multiple stem Graph-based Interference Reduction Network (GIRNet) which operates on raw time domain audio inputs. By learning relationships among the inputs, the model yields interference-reduced outputs. The proposed model not only performs reasonably well when compared with all existing interference

	4 Stems	3 Stems	3 Stems Cleaned*
SDR	-0.19	1.16	0.92

Table 4.6: MSS performance of Wave-U-Net on Saraga datasets and interference reduced dataset

reduction methods in terms of source-to-distortion-ratio (SDR), but also exhibits good performance after post-processing with out-of-domain data, showcasing its potential for generalization. Moreover, the model's effectiveness is corroborated through listening tests conducted on real live recordings.

A pivotal advantage of this approach lies in its computational efficiency during interference, notably outperforming the KAMIR algorithm in terms of processing speed. This attribute renders our model highly applicable to practical scenarios. Thus, this method offers a promising avenue for effective preprocessing in the music source separation application.

4.6 Music Source Separation for Carnatic Dataset

In the existing literature, a dedicated source separation model tailored specifically for Indian classical Carnatic music has yet to be developed. A typical music source separation system for Carnatic music should be capable of decomposing a music mixture into three distinct stems: primary and secondary vocals, mridangam (left and right), and violin.

To address this gap, we employed the Wave-U-Net source separation model and utilized the Saraga dataset. We trained two versions of the model for this task:

1. The first version was trained to separate the mixture into four stems: primary + secondary vocals, mridangam left, mridangam right, and violin.

2. The second version was trained with only three stems, by combining the left and right mridangam stems into a single stem. This approach aimed to explore the trade-off between stem granularity and computational efficiency.

The results are outlined in Table 4.6. However, the values in this table must be viewed in light of the fact that discrepancies persist when comparing these systems. The SDR is computed between the estimated audio and the interfered audio, as no ground truth is available. These numbers only indicate that there is some improvement in the source separation performance when the interference is clean. However, this performance is highly dependent on how well the interference reduction model works, as the ground truths were cleaned using the interference model and then used for MSS training. Given that the estimated, interfered, and cleaned audio differ across models, comparisons are to be taken with this in mind.

CHAPTER 5

Conclusion

This chapter concludes the thesis by summarizing the models introduced for interference reduction and assessing their performance in source separation. A brief overview of each model, including their advantages and limitations, is provided. Additionally, the chapter delves into potential future directions to further advance research in the field.

5.1 Summary and Conclusion

The thesis proposes four models for interference reduction: Convolutional Autoencoder (CAEs), an optimization algorithm, truncated-UNet (t-UNet), and Graph based Interference Reduction Network (GIRNet). Beginning with the CAE models, interference reduction was effective, particularly in the TF domain, treating interference as mere noise. However, a significant drawback of the CAE model is its requirement for dedicated models for each source, which is not practical. Additionally, CAE operates on the TF domain, utilizing STFT magnitude as input, leading to the loss of phase information. Attempts to address this issue, such as with complex CAE, still necessitate dedicated models for each source.

Shifting away from the conventional treatment of interference as mere noise, the thesis formulates a mathematical model for interference reduction akin to the ICA formulation. With valid assumptions, such as each source having at least one dedicated microphone capturing a single dominant source, a discernible pattern emerges in the mathematical formulation in terms of the interference matrix. This problem is then solved using an optimization approach, deriving appropriate update rules similar to NMF. While the results were impressive, this algorithm is limited to linear mixtures, unsuitable for real-world live recordings.

Subsequently, the optimization model is replaced with a neural network, accepting raw waveforms as input, learning the relationships among microphone recordings, and predicting the interference matrix. However, experiments with various artificially created datasets reveal a decline in results with reverberant mixtures due to the linear formulation's inability to capture nonlinearities present in reverberant mixtures. To address this, pyroomacoustics is utilized to create an artificial room, regenerate room impulse response, time delays, and reverberants, resulting in mixtures closer to live recordings. Consequently, the model is reproposed in a nonlinear formulation.

The tUNet was extended to a full wave-u-net, dubbed GIRNet, with differences in input and core working mechanisms. The aim was to enforce robustness and domain invariance within the neural network, ensuring it learned only the complex relationships among input microphone recordings to reduce interference. To achieve this, the interference matrix was incorporated into the network using a graph attention neural network. After the encoded representation, each stem was represented as a node in a graph, with the interference matrix serving as a weighted adjacency matrix. The output of the graph attention network was reshaped and passed to the decoder to reconstruct the sources. As anticipated, the network performed well on the MUSDB18HQ dataset, with GIRNet achieving the highest SDR performance compared to all models compared and the state-of-the-art KAMIR model. The models have been summarised in Table 5.1.

Subsequently, GIRNet was tested with live recordings from the Saraga dataset, which comprised vocals, mridangam, and violin as sources. Remarkably, even with out-of-domain samples, GIRNet performed reasonably well. Post-processing techniques were introduced to further enhance interference reduction performance. Listening tests were conducted on the Saraga dataset since obtaining SDR was not feasible due to the lack of ground truth. It was observed that SDR did not correlate with human perception, as models operating on the time domain consistently yielded higher SDR than those working on the TF domain, despite TF domain models being more perceptually accurate. The listening test results revealed that GIRNet performed similarly to KAMIR on out-of-domain samples, suggesting its usefulness. Additionally, it offers faster inference times, making it suitable for real-world environments.

Method	Highlights	Benefits	Disadvantages	SDR on MUSDB18HQ
Convolutional Autoencoder (CAEs)	Problem modelled as noise and works in the TF domain.	Faster training, low compute resources.	Needs dedicated CAEs for each source and loss of phase information.	Linear mixtures: 7.4
Optimisation Algorithm	Problem modelled as linear mixtures and works in the waveform domain. Learning-free optimisation technique.	Higher SDR and converges to near optimum solution.	Does not work for non-linear mixing data. Operates within specific constraints: dominant diagonals, low off-diagonals.	Linear mixtures: 41 (restricted setup)
Truncated UNet (t-UNet)	Problem modelled as linear mixtures and works in waveform domain. Replaces optimisation algorithm with neural networks.	Faster training, low computer resources, faster interference time, low artifacts.	Does not work for real-world live recordings.	Linear mixtures: 9.1
Graph-based Interference Reduction Network (GIRNet)	Problem modelled as non-linear mixtures and works in waveform domain. Estimates the interference-reduced sources directly and uses graph attention.	Works reasonably well for out-of-domain samples.	Highly complex network, high training time and requires lots of data. Still a lot of scope for improvement on out-of-domain data.	Linear mixtures: 12.0 Reverberant mixtures: 12.2

Table 5.1: Summary of the proposed models in the thesis.

5.2 Future work

There is ample scope for improvement and further research in the fields of interference reduction and source separation. Some potential directions for future work include:

- The thesis proposed interference reduction models separately and utilized different networks for source separation. Therefore, an ideal direction for future work would involve designing an end-to-end neural network that addresses both source separation and interference reduction simultaneously by integrating the concepts of both tasks. One potential approach is to leverage the GIRNet to train both source separation and interference reduction using two input branches and one decoder branch.

- While the GIRNet demonstrates reasonable performance, there is still room for improvement. Exploring multimodal systems or revisiting traditional models could be beneficial. Additionally, integrating appropriate beamformers and direction of arrival techniques, which have seen advancements in recent research, could further enhance the performance of interference reduction models.

- One promising avenue is to focus on generalizing the GIRNet. Since the GIRNet currently requires post-processing for out-of-domain data, a potential direction is to replace the post-processing with a neural network architecture, thus making it end-to-end trainable.

- Another potential direction could involve developing domain adaptation techniques to address the challenges posed by multiple datasets with high domain variance, rather than relying solely on post-processing methods.

- Finally, this problem can also be addressed using generative techniques. By employing the latest generative audio techniques, clean interference audio can be generated from the interference signal, akin to source separation methods.

APPENDIX A

Source Code

To reproduce the results, we have published our codes in the public GitHub repository, which includes all the proposed models and dataset creation codes.

A.1 Dataset Creation

We utilized two types of datasets in our experiments in the thesis: linear mixtures and reverberant mixtures. To generate the dataset, use the following link. Note that for both datasets, the interference matrix Λ and the room dimensions are chosen at random. Thus, the dataset might be slightly different from what we used, leading to marginal performance differences.

1. **LM and RM:** https://github.com/its-rajesh/IRMR

A.2 Models

For these models, the source code can be accessed via the public repository called IRMR.

1. **CAE and tUNet:** https://github.com/its-rajesh/IRMR
2. **GIRNet:** https://github.com/its-rajesh/GIRNet

References

[1] E. Manilow, P. Seetharman, and J. Salamon, *Open Source Tools & Data for Music Source Separation.* https://source-separation.github.io/tutorial, 2020. [Online]. Available: https://source-separation.github.io/tutorial

[2] N. Sturmel, A. Liutkus, J. Pinel, L. Girin, S. Marchand, G. Richard, R. Badeau, and L. Daudet, "Linear Mixing Models for Active Listening of Music Productions in Realistic Studio Conditions," in *AES 2012 - 132nd AES Convention*, Budapest, Hungary, Apr. 2012, p. Paper 8594. [Online]. Available: https://hal.science/hal-00790783

[3] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, "Musical source separation: An introduction," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, 2018.

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

[5] J. V. Stone, "Independent component analysis: a tutorial introduction," 2004.

[6] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2012, pp. 57–60.

[7] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[8] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Information Processing-Letters and Reviews*, vol. 6, no. 1, pp. 1–57, 2005.

[9] A. Liutkus, J.-L. Durrieu, L. Daudet, and G. Richard, "An overview of informed audio source separation," in *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS).* IEEE, 2013, pp. 1–4.

[10] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* IEEE, 2019, pp. 175–179.

[11] B. Pardo, A. Liutkus, Z. Duan, and G. Richard, "Applying source separation to music," *Audio Source Separation and Speech Enhancement*, pp. 345–376, 2018.

[12] D. Pressnitzer and S. McAdams, "Acoustics, psychoacoustics and spectral music," *Contemporary Music Review*, vol. 19, no. 2, pp. 33–59, 2000.

[13] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications.* Springer, 2015, vol. 5.

[14] R. B. Dannenberg, "Music representation issues, techniques, and systems," *Computer Music Journal*, vol. 17, no. 3, pp. 20–30, 1993.

[15] M. Muller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE Journal of selected topics in signal processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[16] F. R. Moore, *Elements of computer music.* Prentice-Hall, Inc., 1990.

[17] P. V. Bohlman, "Music as representation," *Journal of musicological research*, vol. 24, no. 3-4, pp. 205–226, 2005.

[18] M. Vinyes, "MTG MASS database,"
http://www.mtg.upf.edu/static/mass/resources, 2008.

[19] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the mir-1k dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.

[20] T. Fillon, "QUASI Database: a musical audio signal database for source separation," Retrieved from https://adasp.telecom-paris.fr/resources/2012-03-12-quasi-database/.

[21] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research." in *ISMIR*, vol. 14, 2014, pp. 155–160.

[22] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 718–722.

[23] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation - 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings*, P. Tichavský, M. Babaie-Zadeh, O. J. Michel, and N. Thirion-Moreau, Eds. Cham: Springer International Publishing, 2017, pp. 323–332.

[24] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[25] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-HQ - an uncompressed version of MUSDB18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[26] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: Open datasets for research on indian art music," *Empirical Musicology Review*, vol. 16, no. 1, pp. 85–98, 2021.

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

[28] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[29] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[30] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL toolbox user guide – revision 2.0," INRIA, Technical Report inria-00564760, 2005.

[31] E. Oja and Z. Yuan, "The FastICA algorithm revisited: Convergence analysis," *IEEE transactions on Neural Networks*, vol. 17, no. 6, pp. 1370–1381, 2006.

[32] J. Benesty, S. Makino, J. Chen, H. Sawada, R. Mukai, S. Araki, and S. Makino, "Frequency-domain blind source separation," *Speech enhancement*, pp. 299–327, 2005.

[33] R. Prasad, H. Saruwatari, and K. Shikano, "An ICA algorithm for separation of convolutive mixture of speech signals," *International Journal of Information Technology*, vol. 2, no. 4, pp. 273–283, 2004.

[34] A. Ciaramella, R. Tagliaferri, M. Funaro *et al.*, "Separation of convolved mixtures in frequency domain ICA," in *International Mathematical Forum*, vol. 16, 2006, pp. 769–795.

[35] Y. Zheng, I. Ng, and K. Zhang, "On the identifiability of nonlinear ICA: Sparsity and beyond," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 411–16 422, 2022.

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

[37] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization." in *Interspeech*, vol. 2. Citeseer, 2006, pp. 2–5.

[38] P. Magron and T. Virtanen, "Complex isnmf: A phase-aware model for monaural audio source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 20–31, 2019.

[39] S. Ewert and M. Müller, "Using score-informed constraints for nmf-based source separation," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 129–132.

[40] C. Févotte, E. Vincent, and A. Ozerov, "Single-channel audio source separation with nmf: divergences, constraints and algorithms," *Audio Source Separation*, pp. 1–24, 2018.

[41] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative nmf and its application to single-channel source separation." in *Interspeech*, 2014, pp. 865–869.

[42] K. Yoshii, R. Tomioka, D. Mochihashi, and M. Goto, "Beyond nmf: Time-domain audio source separation without phase reconstruction." in *ISMIR*. Citeseer, 2013, pp. 369–374.

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

[44] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. v–341, 2004.

[45] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 92, no. 3, pp. 708–721, 2009.

[46] M. Partridge and M. Jabri, "Robust principal component analysis," in *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501)*, vol. 1. IEEE, 2000, pp. 289–298.

[47] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep neural networks for music source separation," in *IEEE/ACM Transactions in Audio Signal and Language Processing ASLP*. IEEE, 2013, pp. 3734–3738.

[48] P.-S. Huang, M. Chen, and P. Smaragdis, "Music source separation with deep neural networks," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 3734–3738.

[49] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 6, pp. 965–978, 2016.

[50] S. Uhlich, J. Le Roux, and E. Vincent, "Improving music source separation based on deep neural networks through data augmentation and network blending," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1856–1867, 2017.

[51] D. Stoller, S. Ewert, and M. D. Plumbley, "Improving music separation with deep learning," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 64–68.

[52] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3734–3738.

[53] J. Li, J. Hu, and Z. Yi, "Music source separation based on deep neural networks," *IEEE Access*, vol. 7, pp. 82 756–82 764, 2019.

[54] K. Han, J. Zhu, and R. Ding, "Deep clustering and conventional clustering approaches for music source separation: A comparative study," *IEEE Access*, vol. 7, pp. 107 012–107 022, 2019.

[55] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[56] ——, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions in Audio Signal and Language Processing ASLP*, vol. 27, no. 8, pp. 1256–1266, 2019.

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

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

[59] G. Plaja-Roglans, M. Miron, A. Shankar, and X. Serra, "Carnatic singing voice separation using cold diffusion on training data with bleeding," 2023.

[60] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

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

[62] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "Kuielab-mdx-net: A two-stream neural network for music demixing," *arXiv preprint arXiv:2111.12203*, 2021.

[63] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

[64] Y. Luo and J. Yu, "Music source separation with band-split RNN," *arXiv preprint arXiv:2209.15174*, 2022.

[65] L. Pandey, A. Kumar, and V. P. Namboodiri, "Monoaural audio source separation using variational autoencoders." in *Interspeech*, 2018, pp. 3489–3493.

[66] J. Neri, R. Badeau, and P. Depalle, "Unsupervised blind source separation with variational auto-encoders," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 311–315.

[67] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[68] S. Lutati, E. Nachmani, and L. Wolf, "Separate and diffuse: Using a pretrained diffusion model for better source separation," in *The Twelfth International Conference on Learning Representations*, 2023.

[69] G. Plaja-Roglans, M. Miron, and X. Serra, "A diffusion-inspired training strategy for singing voice extraction in the waveform domain," in *Rao P, Murthy H, Srinivasamurthy A, Bittner R, Caro Repetto R, Goto M, Serra X, Miron M, editors. Proceedings of the 23nd International Society for Music Information Retrieval Conference (ISMIR 2022); 2022 Dec 4-8; Bengaluru, India.[Canada]: International Society for Music Information Retrieval; 2022. p. 685-93.* International Society for Music Information Retrieval (ISMIR), 2022.

[70] S. Lutati, E. Nachmani, and L. Wolf, "Sepit: Approaching a single channel speech separation bound," *arXiv preprint arXiv:2205.11801*, 2022.

[71] S. Latif, M. Shoukat, F. Shamshad, M. Usama, H. Cuayáhuitl, and B. W. Schuller, "Sparks of large audio models: A survey and outlook," *arXiv preprint arXiv:2308.12792*, 2023.

[72] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *arXiv preprint arXiv:2308.05037*, 2023.

[73] E. K. Kokkinis and J. Mourjopoulos, "Unmixing acoustic sources in real reverberant environments for close-microphone applications," *Journal of the Audio Engineering Society*, vol. 58, no. 11, pp. 907–922, 2010.

[74] E. K. Kokkinis, J. D. Reiss, and J. Mourjopoulos, "A wiener filter approach to microphone leakage reduction in close-microphone applications," *IEEE/ACM Transactions in Audio Signal and Language Processing ASLP*, vol. 20, no. 3, pp. 767–779, 2011.

[75] E. Kokkinis, A. Tsilfidis, T. Kostis, and K. Karamitas, "A new DSP tool for drum leakage suppression," in *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.

[76] T. Prätzlich, R. M. Bittner, A. Liutkus, and M. Müller, "Kernel additive modeling for interference reduction in multi-channel music recordings," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 584–588.

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

[78] A. Liutkus and R. Badeau, "Generalized wiener filtering with fractional power spectrograms," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 266–270.

[79] D. Di Carlo, A. Liutkus, and K. Déguemel, "Interference reduction on full-length live recordings," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 736–740.

[80] J. Reiss and C. Uhle, "Determined source separation for microphone recordings using iir filters," in *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

[81] A. Clifford, J. D. Reiss *et al.*, "Microphone interference reduction in live sound," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, 2011.

[82] F. Seipel and A. Lerch, "Multi-track crosstalk reduction using spectral subtraction," in *Audio Engineering Society Convention 144*. Audio Engineering Society, 2018.

[83] M. Zhao, D. Wang, Z. Zhang, and X. Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015, pp. 338–341.

[84] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech 2013*, 2013, pp. 436–440.

[85] H. Zhang, W. Li, J. Li, and F. Liu, "A novel speech enhancement method based on convolutional autoencoder and wavelet transform," *IEEE Access*, vol. 9, pp. 106 509–106 518, 2021.

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

[87] R. Scheibler, E. Bezzam, and I. Dokmanić, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 351–355.

[88] T. Prätzlich, R. M. Bittner, A. Liutkus, and M. Müller, “Kernel additive modeling for interference reduction in multi-channel music recordings,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2015, pp. 584–588.

[89] R. R and P. Rajan, “Neural networks for interference reduction in multi-track recordings,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) 2023*, 2023.

[90] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2023, pp. 1–5.

[91] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.

[92] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea, and S. Poria, “A review of deep learning techniques for speech processing,” *Information Fusion*, p. 101869, 2023.

[93] P. Tzirakis, A. Kumar, and J. Donley, “Multi-channel speech enhancement using graph neural networks,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 3415–3419.

[94] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, “Graph attention networks,” *ICLR*, vol. 1050, no. 20, pp. 10–48 550, 2018.

[95] R. Scheibler, E. Bezzam, and I. Dokmanić, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 351–355.

Publications

Journals

1. Rajesh R and Padmanabhan Rajan, "GIRNet: A Graph-Based Approach for Interference Reduction in Live Microphone Recordings" *(under review)*

Conferences

1. Rajesh R and Padmanabhan Rajan, "Neural Networks for Interference Reduction in Multi-Track Recordings," 2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 2023, pp. 1-5, doi: 10.1109/WASPAA58266.2023.10248133.