Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Hierarchical Music Source Separation Using Mixing Secret Multi-track Dataset

Huicheng Zhang

Supervisor: Marius Miron

Co-Supervisor: Ethan Manilow

August 2022

Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Hierarchical Music Source Separation Using Mixing Secret Multi-track Dataset

Huicheng Zhang

Supervisor: Marius Miron

Co-Supervisor: Ethan Manilow

August 2022

Contents

Dedication

I would like to dedicate this work to the 16-year-old me,
who was always curious about how music actually works.

Acknowledgement

I would like to express my sincere gratitude to: my supervisor Marius Miron and my co-supervisor Ethan Manilow, who are always supportive and have enlightened me a lot on MIR research. Thanks to MTG, especially for Xavier's and Rafael's genuine interest in Jingju Music. Thank you Barcelona: I will always remember the taste of the Mediterranean sea and the greetings *Hola - ¿Cómo estás?* one after another every morning. Thanks to my guy Sponge; hope our friendship never die. My biggest thanks go to my parents, for giving me birth, for educating me and supporting me unconditionally. Special appreciation to my sister Kylie Zhang, for funding a large part of my master's education. To all of you, God bless.

Abstract

Music Source Separation (MSS) aims to separate the instrument sources from the mixture. Since MUSDB18 is the standard dataset for MSS, most MSS systems are developed on MUSDB18 and they can only focused on four stems, namely *vocals*, *drums*, *bass*, *other*. It is still impossible to separate more fine-grained level of sources, e.g., lead vocal, backing vocal and guitar. The limitation of MUSDB18 dataset and the lack of large multi-track datasets are possible obstacles.

In this thesis, we first focus on building **a new multi-track dataset called *MS21***, which can be viewed as an expanded and MSS oriented version of the Mixing Secrets dataset. With a total mixture duration of 34.2 hours, MS21 features multi-track content in an unprecedented size and industrial standard. Statistics analysis shows that MS21 contains more averaged number of multi-track files (20±13), a higher percentage of *vocal* stem (84%) among multi-track dataset and a more diverse genre distribution among MedleyDB, MUSDB18 and Slakh2100. MS21 provides MedleyDB like metadata files which capture the specially designed three-level hierarchy annotation for all 500 songs, in order to bridge between different source separation research scenarios.

Hierarchical Music Source Separation (HMSS) is a new research scenario, which aims to separate the sources that share hierarchical relation, for example *accompaniment - guitar - electric guitar*. Multi-level HMSS system showed better performance when separating sources at more fine-grained level, compared to the single-level MSS system. In this thesis, to further explore the research possibility in HMSS, we proposed a theory called **Mask Relation** and a new mask-based method called **Sequential Mask**. Our HMSS experiments showed that the sequential mask-based approach provides worse results for higher-level sources, but better results for lower-level sources (+2dB SI-SDR) compared to a single-level MSS system, which we explain as the leaf-oriented feature of the sequential mask.

Keywords: Hierarchical Music Source Separation; Deep Learning; Multi-track Music Dataset Construction; Mask Relation

Chapter 1

Introduction

1.1 Background

Listeners can mentally isolate an instrument of interest while will not being disturbed by other simultaneously sounding sources. This phenomenon is commonly known as **cocktail party** problem[1]. Neural biologists have pointed out that **auditory attention** plays an important role in our "mentally isolation" of each instrument source. According to Fritz et al. [2], "... a richly interconnected network for auditory attention assists in the identification and recognition of salient acoustic objects, enhancement of signal processing for the attended features or objects ...". In other words, the flexible and dynamic auditory attention has been proven to be crucial to how we perceive each source within music. To find out the focused auditory object, auditory attention decoding (AAD) was studied by analysing the listener's brain response such as electroencephalographic (EEG)[3] and continuous electrocorticography (ECoG)[4].

Researchers firstly study the cocktail party problem in the context of speaker separation, where each speaker was viewed as individual source. The large amount of paper published from [1] shows that much effort has been put into the problem of multi-speaker separation or speech enhancement in the signal processing domain. In

[1] International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

contrast, the study of source separation in the music context is relatively less popular until the establishment of International Society of Music Information Retrieval (ISMIR) in 2000. From then on, evaluation campaigns were carried out, such as singing vocal voice separation in Music Information Retrieval Evaluation eXchange (MIREX)[5] and professionally produced music source separation of *vocals*, *drums*, *bass*, and *other* in the Signal Separation Evaluation Campaign (SISEC)[6, 7]. Following the tradition of SISEC, the recent evaluation campaign is the Music Demixing Challenge[8] organized by Sony in 2021, which proposed a new professionally produced four-stem music recording dataset for testing the algorithms developed on MUSDB18 dataset[9].

In this thesis, we focus on Music Source Separation (MSS). We should note that the component sources of music mixture can vary significantly between different music traditions, which lead to totally different research scenarios. The common objective of MSS is to design a system that can produce high-quality signals of *vocals*, *drums*, *bass* and *other* from a given music mixture. The **four-stem** instrument arrangement assumes that the research object is mainly **Popular Music**, which contains the genres of Pop/Rock, Singer/Songwriter, and others. However, for **Western Classical Music**, those algorithms may not be successful. For example, in **orchestra music** only contains instruments from strings, brass, woodwind and percussion family, which mainly falls into the category of *other* stem. This complex auditory scene comprising a large number of sources of similar timbres and a large number of instruments per source makes it a difficult scenario for source separation. Readers can refer to[10, 11] for further information. Similarly, **choral music** only includes voices of *soprano*, *alto*, *tenor* and *bass* (SATB), which is often regarded as one single *vocals* stem. Choir singing separation was often addressed as monotimbral ensemble separation[12, 13]. For other less well known music traditions like **Chinese Traditional Music** and **Indian Art Music**, there are less research on source separation possibly due to the lack of available dataset. In a word, **the set up of dataset can affect the possible research scenario.**

Apart from the restricted music genre, the **flat hierarchical structure** of target

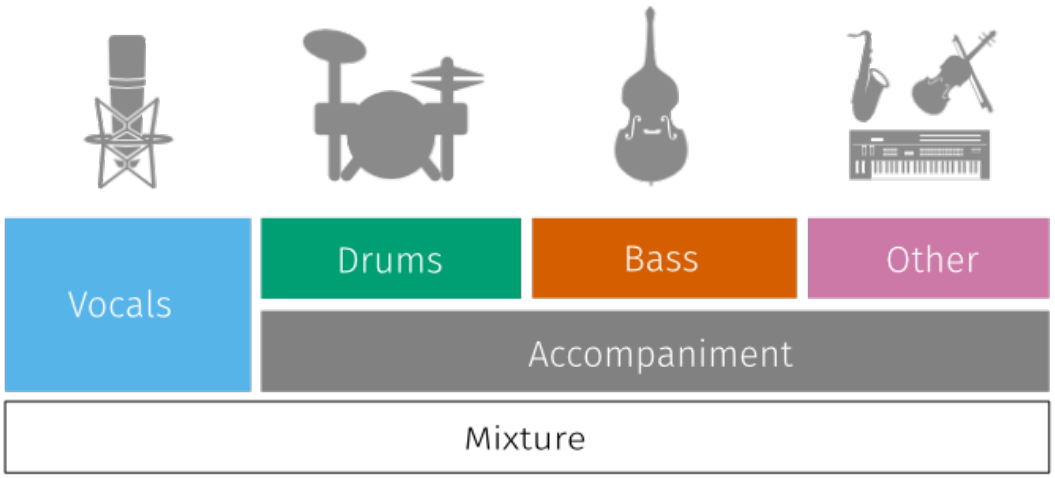

Figure 1: Flat Hierarchical Structure shown in MUSDB18 Dataset

music sources was also a hidden yet strong assumption for the literature based on the four-stem music recording dataset. The mixture M is produced by linearly mixing all four stems $M = \sum S_i$, $i \in \{1, 2, 3, 4\}$ together. This assumes that there is no relation between each stem in the signal perspective, meaning $S_i \cap S_j = \emptyset$, for i, j in $C\binom{4}{2}$. As illustrated in Figure 1 , in conventional MSS, the music mixture has one-level hierarchical structure: a parent source M and four child sources S_i , $i \in \{1, 2, 3, 4\}$. In practice, the *accompaniment* stem can be generated by mixing all stems except *vocals*, which indicates an extra hidden level for that three stems. This leads to an unbalanced hierarchical structure, which assumes that vocals has more important status compared to other three stems. To the best of our knowledge, there is no literature presenting a method to simultaneously generate *accompaniment* and its component sources, namely *bass*, *drums*, and *other*, which means that the hierarchical structure of music mixture was neglected.

The one and only work to tackle the Music Source Separation problem with the hierarchical structure information of music so far, was the work by Manilow et al. [14], based on a synthesized multi-track music dataset called Slakh2100. This new research scenario was called **Hierarchical Music Instrument Separation (HMIS)** according to the paper. Inspired by the music instrument hierarchical

ontology proposed by musicologists and the track-stem(submix)-master workflow of mixing engineers, they built a system that can generate "submixes of instruments corresponding to multiple levels of an instrument label hierarchy". The low-level submix contains instruments that are more similar and the higher-level submix contains different classes of instruments. In other words, estimated sources share **inclusion relation**. This is contrast to the conventional MSS algorithms, which assumes the intersection between any two estimated sources is empty set and should be regarded as independent sources in terms of timbre. To compare HMSS with conventional MSS using an example of *Electronic Guitar* separation, a HMSS system can generate a set of sources related to *Electronic Guitar* in a hierarchical manner: *Level3 string+guitar+keyboards - Level2 Guitar - Level1 Electronic Guitar*, while a MSS system would only be expected to generate *Electronic Guitar* at the single level. Experiments in [14] show that multilevel hierarchical networks improve over single-level models, with the largest gains occurring at lower hierarchy levels.

In this thesis, we follow the idea presented in the work of Manilow et al. [14] and seek a better understanding of this research scenario. Since Slakh2100 [15] dataset only contains synthesized instruments, we want to include **vocals** as the target source types. Therefore, we rephrase the research scenario as **Hierarchical Music Source Separation (HMSS)**, to deal with a real-world scenario. What's more, we also want to leverage the legacy in the literature of conventional MSS on MUSDB such as algorithms and understandings. We believe that HMSS is a promising research direction in the field of MSS.

To sum up, we formulate the research question of our work: *i*) **Can we build a multi-track music recording dataset to train, validate and test a data-driven source separation model?** *ii*) **What is the relation between conventional Music Source Separation (MSS) and Hierarchical Music Source Separation (HMSS)? Can we bridge the gap between MSS and HMSS?** *iii*) **What can we do to improve state-of-the-art algorithms?**

1.2 Motivation

Our work was first motivated by the idea of building a new multi-track music recording dataset to cater to the industrial demand. As we know, MSS is restricted by the dataset setup and for industrial applications, the setup is sometimes not cater to demand. For example, in automatic karaoke backing track generation, backing vocal along with lead vocal is eliminated by the algorithms developed on MUSDB18, leading to an unsatisfying user experience. This is due to the fact that the stem definition contradicts to the general belief that backing vocal is a part of the accompaniment.

A hierarchy-aware music source separation system would be extremely beneficial for other downstream MIR tasks such as automatic music transcription [16], fundamental frequency estimation [17], music instrument detection [18], etc. **A data set that represents the real-world auditory scene is the key to MSS system. The set up of dataset can highly affect the possible MSS research scenario.** As a new direction, HMSS asks for new requirements for dataset. MedleyDB[19, 20] is a multi-track music recording dataset mainly built for the task of fundamental frequency estimation. One drawback of MedleyDB is the missing of lead vocal and backing vocal annotation for the tracks. Mixing Secrets dataset[21] was a multi-track music recording dataset built in 2017, derived from the website *The 'Mixing Secrets' Free Multitrack Download Library* [2]. Up to now the available multi-track projects on the website has increased to more than 500 songs, which is significantly larger than the old Mixing Secret dataset. Besides, Mixing Secrets dataset was originally built for instrument detection. Manual annotations for music source separation are not available. Thus, **a better dataset for music source separation is needed to be created.**

By critically listening to the MUSDB18 dataset, we found that there is a hidden instrument hierarchy when creating this dataset. For example, *vocals* stem contains lead vocal and backing vocal, *bass* stem contains electric bass, synthesized bass

[2] `https://cambridge-mt.com/ms/mtk/`

and acoustic bass, *drums* stem contains all non-tonal percussive instruments such as drum set and percussion. *Other* stem contains any instruments not belonging to the previous mentioned category, e.g. *acoustic guitars, electric guitars, pianos, synthesizers, tonal percussion, Special effects (SFX)*. This mixing workflow indicates that music has hierarchy. **What should be the optimal hierarchy for music? Does an HMSS system have any distinguishing feature or more advanced approach compared to an MSS system?** This question intrigues us to explore the HMSS research scenario. "**Audio does contain a hierarchy**" serves as a strong assumption for our research, as we regard hierarchy design as an important part of our work.

1.3 Objectives

To properly answer the research questions, here we define the objectives of this thesis: *i*) **Construct a multi-track music recording dataset with sufficient annotations for research in MSS and HMSS;** *ii*) **Systematically compare the implementation detail between MSS and HMSS to seek the possible direction for improvement;** *iii*) **Train, validate and test a data-driven source separation model with these data in order to improve the state-of-the-art algorithm.**

For step (*i*), we first conduct a literature review on the datasets used in MSS. We present a lineage evolution of the standard dataset used in SiSEC. By comparing across different datasets, we try to construct a dataset with clean folder structure like the MedleyDB dataset, a larger size with high-quality annotation for MSS. We hope that this dataset will also be useful to other researchers in the field of MIR.

Through step (*ii*), a systematic comparison was made between research in MSS and HMSS. By categorizing the literature in a well-organized way, we can see what has been studied intensely and what is missing here, which means a new research idea can emerge. This naturally leads to step (*iii*) for potential scientific contribution. We closely inspect the research problem framework and software implementation of

the work of Manilow et al. in [14]. New theory and Deep Neural Network (DNN)-based experiments will be carried out to further explore HMSS.

1.4 Structure of the Thesis

Chapter 2 will present the state of the art of MSS and HMSS research, highlighting the relation between dataset and the research scenario and offering critical thoughts when defining a HMSS system. Chapter 3 will focus on the construction process of Mixing Secrets version 2 dataset, or **MS21**. Statistics analysis will also be performed among several datasets. Then, a theory called **Mask Relation** was proposed for the first time in Chapter 4, which describes two source estimation processes based on two semantically different masks, namely **Direct Mask** and **Sequential Mask**. Several experiments will be conducted. Evaluation and results will be presented in Chapter 5. Finally, we present the conclusions and future works in Chapter 6.

Chapter 2

State of the Art

In this section we present the relevant work in four sections. In the first section, we will present a extensive overview of the datasets used in the task of Music Source Separation (MSS) and Hierarchical Music Source Separation (HMSS) respectively. Since the dataset setup can hugely affect the research scenario, we classify the literature into four types according to different research scenarios. We will present an overview of the related work, namely Conventional MSS research (on four-stem or vocal-accompaniment dataset)(Section 2.3), Conventional MSS research on multi-track dataset (SubSection 2.3.8), HMSS research on four-stem dataset (Section 2.4) and HMSS research on multi-track dataset (Section 2.5). By doing this classification of literature, we can see the landscape of state of the art and some possible future research directions. We found that there are more research in Conventional MSS than in HMSS so we will introduce the typical building blocks in a MSS system in the subsections of Section 2.3.

2.1 Datasets in Music Source Separation

Over the years, datasets for MSS research has been published and recently served as an important role in data-driven deep learning approach. As pointed out in Section 1.1, research assumption lies in the dataset we work on.

We first focused on the standard datasets used in some active evaluation campaigns. An active community can shorten the development period of each algorithm design iteration. For example, Music Information Retrieval Evaluation eXchange (MIREX)[5] is a driving force for early stage research on singing voice separation by presenting MIR-1k and iKala for competition. Signal Separation Evaluation Campaign (SiSEC)[22, 23, 6, 24, 7, 25] was held regularly as a community-based scientific evaluation from 2008, based on the successful experience of Stereo Audio Source Separataion Evaluation Campaign (SASSEC) in 2007. In this thesis, we view SiSEC as an important information source in this chapter. We only focused on the task of Source signal estimation with professionally produced music recordings: " to evaluate source separation algorithms for estimating one or more sources from a set of mixtures in the context of professionally-produced music recordings."

Here we present the historical development of dataset used in SiSEC:

- In 2008, test data and development data is both from Musical Audio Signal Separation (MASS) [26]. The official development data and test data share the same two songs, but different excepts. Participants can use other songs in MASS and they were asked to separated vocals, guitar, piano, bass from the mixture snips.[1]

- In 2010 and 2011, test data and development data includes two songs from MASS [26] and six newly proposed songs done by Michel Desnoues from Telecom ParisTech. The test and development data was equally split and they still shared the same two songs, but different excepts. For the first time, a clear definition for sources such as vocals, bass and drums was proposed[2]:

  - "vocals" = "a sum of any singing including main vocal, back vocals and singing in the reverb"

  - "drums" = "a sum of any drums including bass drum, hi-hat, snare etc."

[1] http://sisec2008.wiki.irisa.fr/tiki-index165d.html?page=Professionally+produced+music+recordings

[2] http://sisec2010.wiki.irisa.fr/tiki-index165d.html?page=Professionally+produced+music+recordings

– "bass" = "bass guitar only (i.e., not bass drum)"

Participants were asked to separate vocals, guitar, piano, bass and drums from the 20-second mixture. Extra data in MASS was allowed to be utilized.

- In 2013 [3], development data repeat the one used in 2010 while new data from MASS and QUASI [22, 27] were added to test data. Participants were asked to separated sources from snips and from full-length mixtures, but no full-length songs with ground truth were provided.

- In 2015 [4], the Mixing Secret Dataset 100 (MSD100) was proposed to evaluate the separation of four predefined stems namely bass, drums, vocals and other (i.e the other instruments). All songs from various genre and they are of full length, same sampling frequency (44.1kHz). The dataset was derived from The 'Mixing Secrets' Free Multitrack Download Library [5]. It provides four stems along with mixture audio files for each track. However, all stems were not manually mixed using real professional Digital Audio Workstations. This lead to the fact that some stems could be mono. Besides, the split of test and development set contains songs from the same artist, which would lead to overfitting issues.

- In 2016 [6], MSD100 was heavily remastered by Antoine Liutkus and Stylianos Ioannis Mimilakis so that each track features semi-professionally engineered stereo source images. This new dataset was called Demixing Secrets Dataset 100 (DSD100) [7]. An open source software was developed for data processing and evaluation. Multichannel modelling, data augmentation, a fusion of different systems show promising result.

- In 2018 [7], MUSDB18 was created from DSD100, additional data taken from MedleyDB, data provided by Native Instruments and the personal donation of

[3] http://sisec.wiki.irisa.fr/tiki-index165d.html?page=Professionally+produced+music+recordings

[4] https://sisec.inria.fr/sisec-2015/2015-professionally-produced-music-recordings/

[5] https://cambridge-mt.com/ms/mtk/

[6] https://sisec.inria.fr/sisec-2016/2016-professionally-produced-music-recordings/

[7] https://sisec.inria.fr/2018-professionally-produced-music-recordings/

Canadian rock band The Easton Ellises. It followed the four-stem definition of stems and it provided a different test-train split to address the issue of overfitting on the same artist in DSD100. In terms of audio format, MUSDB18 was originally encoded in a lossy stem format proposed by Native Instruments. Later in 2019, MUSDB18-hq[28] was published featuring an industrial standard audio format, 44.1kHz and 16bits.

- In 2021 [8], following the long tradition of SiSEC, Music Demixing (MDX) [8] Challenge was held as a satellite event of ISMIR 2021. MDX2021 was carried out through an open crowd-based machine learning competition website, AIcrowd, and mainly featured a brand new hidden test set. Fairness was put in a high priority as the challenge was divided in three rounds, in the first two rounds only reveal a subset of the newly created test set MDXDB21 for evaluation, and final scores were computed on the complete test set after the end of submission, which prevent users from adapting their model on the test data. What's more, the test set contained songs with a wider range of genres and involved a greater number of mixing engineers. This aimed to address the bias within MUSDB18: mainly songs of Pop/Rock genre and homogeneous mixing characteristics.

Apart from the evolution of datasets used in SiSEC, we compared several datasets following some general criterion such as Number of tracks, Duration of tracks, Audio format, Instrument sources and Publish license, as shown in Table 1 . The detailed comparison between selected multi-track datasets will be presented in Section 3.2.2 when we introduced our custom dataset.

- MASS and QUASI dataset share the same tracks. Both were mainly proposed to studied the impact of different mixing scenarios on the separation quality so they contain multiple instruments. One obvious drawback is the small number of tracks, which is unsuitable for developing data-driven algorithms.

[8] `https://mdx-workshop.github.io/`

Table 1: Summary of datasets for Music Source Separation[29]

Dataset	Year	Ref.	Tracks	Dur(s)	Full/Stereo?	Format	Sources	License
MASS	2008	[26]	9	16±7	no / yes	44.1kHz 24bits	Multi-track	CC BY-NC-ND
MIR-1K	2010	[30]	1000	8±8	no / no	16kHz 16bits	Vocal-Accomp.	CC BY
QUASI	2011	[22, 27]	11	206±77	yes / yes	NAN	Multi-track	Mix of license
ccMixter	2014	[31]	50	231±77	yes / yes	NAN	Vocal-Accomp.	NAN
iKala	2015	[32]	206+100	30	no /no	44.1kHz 16bits	Vocal-Accomp.	NAN
DSD100	2015	[?]	100	251±60	yes / yes	44.1kHz 16bits	Four-stem	CC BY-NC-SA
MedleyDB	2016	[19, 20]	254	206±121	yes / yes	44.1kHz 16bits	Multi-track	CC BY-NC-SA 4.0
MUSDB18	2017	[9]	150	236±95	yes / yes	44.1kHz 256kbps	Four-stem	Mix of license
Slakh2100	2019	[15]	2100	240±95	yes / no	44.1kHz 16bits	Multi-track	CC BY 4.0

As mentioned before, they were served as the data source in the early stage of SiSEC.

- MIR-1K contains 1000 song clips extracted from 110 karaoke songs which contain a mixed track and a music accompaniment track. These songs were selected (from 5000 Chinese pop songs) and sung by the colleagues in the author's lab in [30], consisting of 8 females and 11 males. Most of the singers are amateurs with no professional training. The music accompaniment and the singing voice were recorded at the left and right channels, respectively. Besides, manual annotations (such as lyrics, pitch contours, unvoiced types, and so on) for each clip should be as sufficient as possible for all kinds of possible evaluations for singing voice separation.

- iKala is also a dataset of Chinese pop songs. Together along with MIR-1K, it served as a dataset for MIREX campaign. Both contains mono vocal and accompaniment. Compared to MIR-1K, iKala was performed and recorded professionally.

- MedleyDB is proposed mainly focus on melody detection, as the dataset contains pitch annotation. Multi-tracks are provided along with metadata file which includes song-level information such as artist, title, composer, genre and so on. Genre of *Classical*, *Jazz* and *World/Folk* held predominant status in MedleyDB as illustrated in Picture 2, which indicates a large number of songs that do not contain vocals. The bleeding label implied that potential leakage between different instrument exists. Further information about the

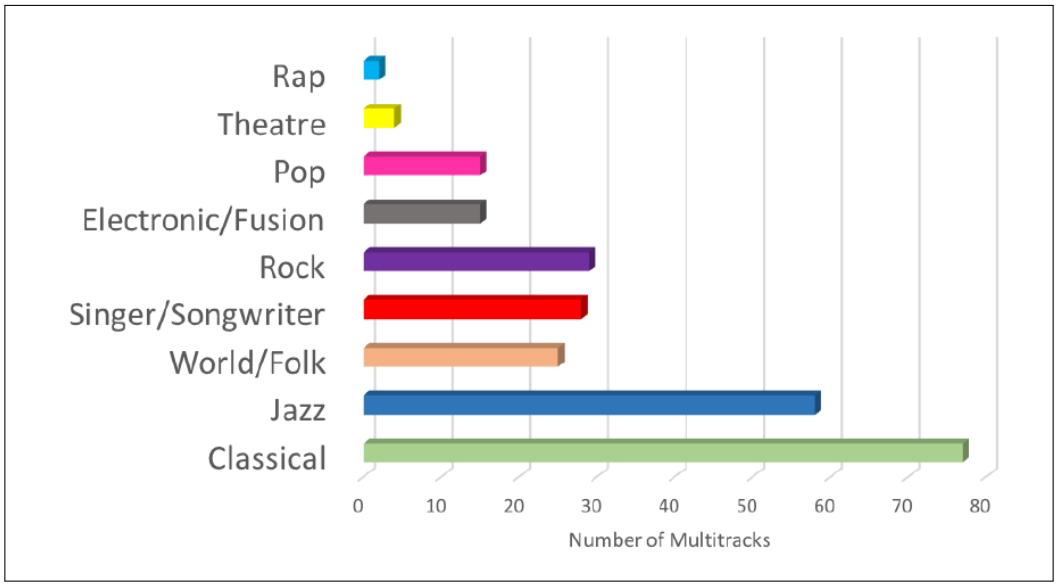

Figure 2: Genre distributions in MedleyDB 2.0

statistics of MedleyDB can be found in Section 3.2.2. Note that we found inconsistency in the number of publicly available multi-tracks and the number of metadata found in the Github repository. In this thesis, we only consider the 196 publicly available songs and their metadata files when doing quantitative analysis in Section 3.2.2.

- MUSDB18: As mentioned before, MUSDB18 contains professionally mixed stems and the well defined four-stem structure, which certainly helps to shorten the development period of MSS algorithms. However, on the other hand, this fixed four-stem scenario can be a hidden constraint for our future research on MSS. As indicated in the Section 1.2 algorithms developed on this dataset can be problematic for some application, more low level source separation can not be done; As pointed out in

- Slakh2100 is a synthesized version of Lakh MIDI dataset. It contains a large number of instrument tracks where guitar holds a predominant status. The drawback is obvious: its lack of Vocal and real life music performance made it hard to generalize well on music data from real life. However, it is good to do cross dataset validation when building a system to separate the sources that both exist in MUSDB18 and Slakh2100.

Table 2: Summary of Datasets for Culture Related Music Source Separation

Dataset	Year	Ref.	Songs	Dur(hh:mm:ss)	Stereo	Format	Sources	License
PHENICX-Anechoic	2017	[33]	4	00:10:37	yes / no	44.1kHz,16bits	Orchestra music	CC BY-NC-SA 3.0
Dagstuhl ChoirSet	2020	[34, 35]	2+exercise	00:55:30	yes / no	22.05kHz	SATB	CC BY 4.0 Inter.
Saraga	2021	[36]	168	NAN	yes / yes	44.1kHz,128kbps	Vocal-Accomp.	CC BY-NC 4.0

- The first version of Mixing Secrets [21] Dataset was published in 2017, only with annotations for the task of music instrument recognition. The instrument label is generated based on the file name of each track. Stems are not professionally produced like MUSDB18. One of the advantage of the dataset is that it includes human performed singing voice and professional recording quality. Thanks to the continuous good maintenance of Cambridge Music Technology, nowadays the number of available tracks on the website is more than 500 and it almost doubles the size of the first version of Mixing Secret dataset and MedleyDB, making it potentially a good data source to build the largest multi-track music recording dataset with human performance.

Dataset has been a crucial part in the evolution of MSS algorithms. In early stage, partly due to the issue of music copyright, dataset were created by receiving kind donation from artist (MASS, QUASI) or by recording unprofessional performance of themselves (MIR-1K). These dataset tend to be small, without professional quality. Dataset created by amateur did not follow industrial audio production standard.

Entering the deep learning era, data driven algorithms call for more data, better professionally produced. As indicated in the evolution of the datasets used in SiSEC, the size of the dataset used in the evaluation campaign was gradually increasing. From small open datasets such as MASS and QUASI, to DSD100 from The 'Mixing Secrets' Free Multitrack Download Library [9]. On the other hand, the definition of the target sources was clearer or more strict. The target stems and mixture files were provided according to the four stem definition. These datasets contain full-length tracks, offers good documentation and software tools to support the research community.

[9] `https://cambridge-mt.com/ms/mtk/`

However, there are several drawbacks in the current available dataset. When MUSDB18 becomes the standard dataset, researcher would take the hidden assumption of **four-stem flat hierarchy** for granted, which was pointed in Section 1.1. The publish of multi-track datasets such as Slakh2100 provides a possibility to challenge the assumption thanks to their rich category of instrument types. It opens a door for research on hierarchical music source separation, which is the main focus of this thesis. Lastly, there are MSS research with a special focus on less studied music tradition such as orchestral music, choral singing, Indian art music, which is out of the scope of our work. Readers can refer to the references shown in Table 2 for further information.

2.2 Historical View on Music Source Separation

From the previous section, we know that in the pre deep learning era, researcher mainly focus on singing voice verse accompaniment separation because of the limited availability of large music datasets. When there is only two sources, it is impossible to think about the inner hierarchical structure within the mixture. In other words, it is hard to formulate the problem as HMSS. Thus, main attention were attached to the **signal modeling** of vocal and accompaniment. For example, they managed to model the vocal signal with its **Harmonicity** nature and to construct the accompaniment signal with its **Redundancy** feature. Note that in the overview paper[29] that we mainly refer to in this section, the problem was named as lead and accompaniment separation, since in popular music, vocal remains a frequent component of lead signal. In this section, the historical approaches on vocal and accompaniment were introduced.

2.2.1 Modeling the Lead Signal: Harmonicity

Assuming that the lead signal is harmonic, one type of method focus on fundamental frequency tracking or melody extraction. Upon that, **Analysis-Synthesis** approach and **Comb Filtering** approach were developed. For Analysis-Synthesis approach, a sinusoidal model which decomposes the sound with a set of harmonically related

sine waves of varying frequency and amplitude was applied to construct the lead signal. This approach can date back to 1973, when Miller proposed in [37] to use the homomorphic vocoder to separate the excitation function and impulse response of the vocal tract. However, using such sinusoidal synthesis to generate the lead signal suffer from a typical metallic sound quality, which is due to the discrepancies between the estimated signals compared to the ground truth. Comb-Filtering approach was proposed to filter out everything from the mixture that is not located close to the detected harmonics [29]. This idea of generating the source by applying filtering to the mixture in time-frequency domain, was adapted as **masking** in the later literature.

There are shortcomings of the above mentioned harmonic modelling approach. Firstly, lead vocals are not always purely harmonic as they contain unvoiced phonemes. Consonant sounds sometimes were viewed as sharing a percussive characteristic, and this phenomenon was well utilized in music industry. Vocal production varies a lot across different music genre. However, for BACKING VOCAL, this harmonic assumption can be true for the most of the time, as backing vocal often contains long slowly evolved notes and not necessarily to be linguistically meaningful. This idea can be helpful for lead vocal verse backing vocal separation. Secondly, harmonic model based methods depend on the quality of the pitch detection method. Predominant pitch detection may fail when the lead vocal is not the loudest harmonic sound in the mixture.

2.2.2 Modeling the Accompaniment: Redundancy

With the assumption that the spectrogram can be well represented by only a few components, techniques like **Non-negative Matrix Factorization (NMF)** was applied to source separation. Because of the **low-rank** nature of music signal, mixture can be factorized into several templates and basses. "learn the non-negative reconstruction bases and weights of different sources and use them to factorize time-frequency spectral representations." NMF would decompose a mixture matrix into two matrices: Template and Activation. Since it is a linear model, the ability to

represent the complex musical sources is limited. What's more, the computational cost makes NMF system hard to be deployed in real-time application. Because of its non-negative fashion, it can not be applied to waveform based system in the fact that waveform ranges from -1 to 1. In the approach of **Robust Principal Component Analysis (RPCA)**, low-ranked assumption are made only in the accompaniment whereas vocals are assumed to be sparse and not structured. While the idea that "musical background should be redundant, adopting a low-rank model(NMF, RPCA) is not the only way to do it. Another way is to exploit the musical structure of songs, to find repetitions that can be utilized to perform separation." [29] Repeating Pattern Extraction Technique (REPET) algorithm [38] was designed based on this assumption. First, a repeating period is extracted by a music information retrieval system, such as a beat spectrum. Then, this extracted information is used to estimated the spectrogram of the accompaniment through an averaging of the identified repetitions. From this, a filter is derived [29]. One of the shortcoming for this method is that the estimated vocals suffer from the interference from unpredictable parts from the accompaniment, in that there is not an adequate model for lead signal.

Entering the era of Deep Learning, Data-Driven approaches out-perform the model-based approaches with significant improvement. One reason is that the large amount of data and the other is the great ability of neural network to simulate the target sources. In the following section, we will present the state-of-the-art deep learning methods and categorize them into four types according to their research topics and dataset setup.

2.3 Conventional Deep Learning based MSS Research

Deep Neural Network (DNN) was generally first used in Music Source Separation in [39]. From then on, many neural network architectures and training techniques emerged. In the following sub-sections, we will present the SOTA approaches in building blocks of a basic MSS system, namely Input, Output, Model Architecture, Loss Function, Evaluation Metric, Data Augmentation Techniques, Skip Connection

Style, Bridging & Blending. [10]

2.3.1 Input: Spectrogram vs Waveform

We first present the audio input presented in the literature. Huang et al. [39] concatenated **neighbouring STFT magnitude spectra** (1024 samples with 50% overlap) together as input features to the model in order to capture the contextual information. 3 frames, adding one neighbouring frame in both direction was found to perform the best for Deep Recurrent Neural Network. Huang also tried using log-mel filterbank spectrum and log power spectrum as the input but they both showed worse performance. Uhlich et al. [40] followed this idea in to feed neighbouring frames to the DNN with ReLU layers, aiming to provide the DNN with temporal context.

Using Long Short Term Memory (LSTM) in the model can capture the context information without the need to concatenate neighbouring frames as input. For example, in [41], Uhlich et al. utilized Bidirectional LSTM and they only feed a frame size of 1024 samples to it. For those models that utilized 2D Convolutional Neural Network (CNN), the input to the model could change from magnitude spectra to **STFT magnitude spectrogram** with certain segment length, for example, in [42, 43, 44, 45, 46].

Another important method was using **waveform** as the input to the model, in that waveform contains phase and magnitude information of the audio signal. This end-to-end style can be seen in [47, 48, 49, 50]. Apart from choosing waveform as the input feature, Complex as Channel (CaC) serves an alternative way to inject the phase information into the model. As introduced in [51], the **real and imaginary part of the complex-valued spectrogram** was concatenated along channel dimension and then fed to the model. Later in [52], CaC was used as the input of the model. Defossez et al. [53] pointed out that "theoretically there is no difference between spectrogram and waveform models especially when considering CaC, which

[10] Detailed information can be seen in `https://github.com/felixCheungcheung/MusicSourceSeparation_Literature`

is only a linear change of base for the input and output space.". However, for a small constrained dataset, spectrogram methods varies more between their input space and output space.

2.3.2 Output: One vs All

As indicated in [54], deep learning-based models for MSS can be categorized into three types according to their input/output setup, (1) **dedicated models**, (2)multi-head models which we will call it **multi-target models** from now on), (3)**conditioned models**. The first two categories are based on the output setup while the third category focus on whether non audio feature was added to the model.

For **dedicated models**, each of them will optimize for a single source and source-specific data pre-processing can be implemented to enhance the result, as shown in [55]. In the literature, dedicated models are **predominant**. *Bridging* is an important idea to compensate for the isolation between each source in this architecture, which we will discussed in more detail in later Section Bridging.

For **multi-target model**, multiple outputs are generated at a single inference. multi-target models can generate good results by leveraging the inter-class correlation of each source learnt by the model. Huang et al. [39] discovered that modeling two target sources *vocal* and *accompaniment* simultaneously provides better performance. Defossez et al. [50, 53], implemented an end-to-end multi-target model to estimate four stems in MUSDB18. However, as the number of sources increase, performance would be degraded due to the shared bottleneck and the inefficient memory usage[54].

Conditioned models take additional information of the target source. Music score informed approach were popular in MSS for orchestral music as shown in [56] using PHENICX-Anechoic dataset [33]. For research using MUSDB18, there are several types of additional information such as 1) Instrument activity, in [57]; 2)Audio query, in [58, 14]; 3) Class conditioned vector (which instrument to be separated), in [59]. Interestingly, Giorgia Cantisani et al. [60] utilize EEG as side information to a MSS

algorithm named Contrastive Non-negative Matrix Factorization (C-NMF), with the assumption that the neural activity is able to track musically relevant features. They worked on the MAD-EEG[61] dataset, which is a 20-channel EEG signals recorded from 8 subjects while they were attending to a particular instrument in polyphonic music. Their work shows promising results for the attended sources.

Mask based approach was vastly applied in machine learning methods. For deep learning based method, researcher has different preferences on whether estimating a mask or not. According to Rafii et al. [29], the reason why it is still desirable to have the output of the network to be a mask, is that the entire dynamic range of the data does not have to be covered by the output of the network. It is convenient in this context to truncate a to between 0 and 1, which fits the range of a sigmoid unit. For those who believe in the representation ability of DNN, models are set to estimate the source signal directly.

Choi [54] pointed out that four types of spectrogram-based source estimation setup: 1) Direct Magnitude Estimation; 2) Magnitude Mask Estimation; 3) Direct Complex-valued Spectrogram Estimation; 4) Complex-valued Spectrogram Mask Estimation. Within them, Direct Complex-valued Spectrogram Estimation works the best. This requires using Complex as Channel (CaC) as the input. For magnitude spectrogram estimation, there are different kinds of masks: Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM). IBM and IRM has an assumption that sources are subsets of mixture in terms of energy, the network only need to do energy allocation. Ideal Complex Ratio Masks (iCRM) was proposed in [62], aiming at faithfully recovering the complex spectrogram, where one source may exceed the mixture in terms of energy. Inspired by [63], the model learns to decouple the magnitude and **phase** information of the mask, contrary to other models which only use the phase of the mixture during inference. The oracle of iCRM is much higher than the upper bound calculated by IRM an IBM. Another way to include phase information is to use phase sensitive metric, as in [14], truncated phase sensitive approximation (tPSA) target function was used during training.

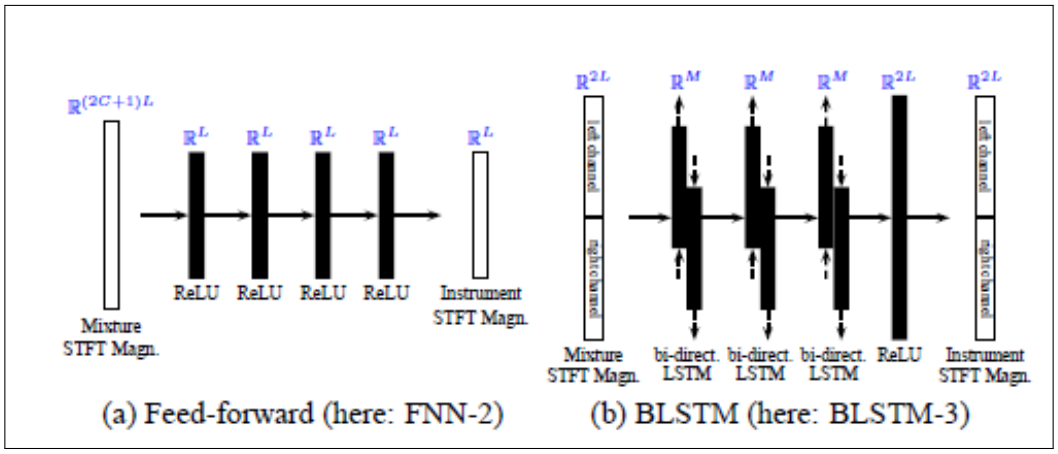

Figure 3: DNN structure in [41], Left: *UTL2*, Right: *UTL3*

2.3.3 Model Architecture

Feedforward Neural Network (FNN, or DNN) and Recurrent Neural Network (RNN) was first adapted inn MSS by Huang et al. [39]. They compared FNN and different types of Deep Recurrent Neural Network (DRNN) such as recurrent connection at the k-th hidden layer (denoted as DRNN-k) and recurrent connection as all layers (denoted as sRNN). All architectures are of three layers plus two dense layers as time-frequency masking layers and Rectified Linear Unit (ReLU) was the non-linear function. Results show that DRNN is better than FNN. DRNN-*2* provides the best results among all DRNN. Lastly, compared to previously proposed algorithms such as RPCA [64] and RNMF [65], DNN based methods show significant improvement over machine learning based algorithms.

Uhlich et al. [41] investigated **Model Blending** between FNN and BLSTM. There are three models in [41], a FNN with three Fully Connected layers and ReLU as non-linear function, which was denoted as ***UTL1*** in [7]. Note that compared to [39], *UTL1* has no time-frequency masking layers, which means the network estimates the magnitude STFT of the target sources directly. A four-layer version of *UTL1* can be seen in Figure 3. ***UTL2*** was trained with more data samples compared to *UTL1* by feeding longer context frames (C = 8) and more unseen tracks from MedleyDB. ***UTL3*** consists of three BLSTM layers where each layer contains 250 forward and 250 backward LSTM cells. Compared to RNN, LSTM networks have the advantage

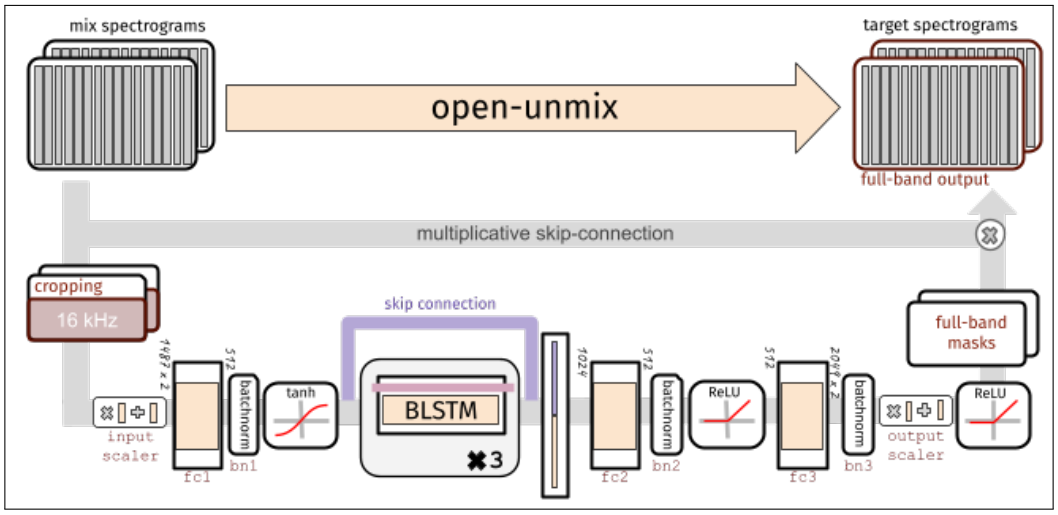

Figure 4: Open-Unmix architecture

that they do not suffer from the vanishing/exploding gradient problem. *UTL3* shows more consistent improvement compared to *UTL2*. By Blending the raw outputs of *UTL2* and *UTL3* and then perform a Multichannel Wiener Filtering (MWF) as post-processing, better results were achieved (Averaged 0.2 SDR gain and being the SOTA in SiSEC 2016).

Deep Clustering [66, 67] and **Cerberus** [16] also use BLSTM layers as building blocks. The body of the network consists of 4 BLSTM layers and each layer has N hidden units. Then, a fully connected layer outputs a $F \times D$ representation served as a latent vector Z shared by the subsequent network modules. For Deep Clustering head, Z then pass through a *tanh* non-linearity and unit-length normalization independently form each $T - F$ bin, yielding a $TF \times D$ embedding matrix V for clustering. For Mask-Inference head, a fully connected layer with softmax as the non-linear function takes Z as input and it outputs C masks, one for each source. The transcription head in [16] is a fully connected layer with sigmoid as the non-linear function, which outputs transcription for each source.

Open-Unmix (UMX) in [68] combines FC layers and three layers of BLSTM. The model takes two spectrogram as the input. After cropping anything content above 16kHz and normalizing the input by the mean and standard deviation, the two channel input of was fed to a FC layer, followed by a batch normalization layer and

tanh non-linear activation. Three BLSTM layers of 512 hidden units takes the input and generates a deeper representation, which was concatenated with the residual skip connection output [69]. Another FC layer + batch normalization and ReLU module halves the input size. Then a FC layer upsamples the latent vector to the original input size, followed by batch normalization, an output scalar and a ReLU non-linear activation for generating two non-zero full band masks, each corresponds to left and right channel. Finally, target source spectrograms were calculated by multiplication between the original mixture input spectrogram and the estimated mask. Compared to FNN & BLSTM model output blending presented in [41], Open-Unmix puts the BLSTM as the bottleneck, which already receives the processed information from the previous FC layer. The distilled representation will be decoded to full-band ideal ratio masks. The implementation of residual skip connection can let the network choose between different levels of representation. Open-Unmix serves an important bassline model for MSS.

Convolutional Neural Network (CNN) [70] has been successful in image related tasks. In [43], MSS problem was formulated as the translation of a mixed spectrogram into target source spectrograms. CNN requires less computational resources and memory than FNN thus it allows faster and more efficient deployment. By stacking many 2D convolutional layers, the receptive field of Neural Network gets larger, resulting in longer context and wider frequency range.

2D CNN captures local spatial correlations in 2-dimensional vectors. Thus it is usually fed with **Spectrogram** as input. There are several architectures mainly utilizing 2D convolutional layers:

- Multi-scale multi-band Densenet: **MMDenseNet** [45] consists of dense block module, down-sampling layers and up-sampling layers, where *inter-block skip connection* was directly added between two dense blocks of the same scale. The down-sampling layer was defined as 1×1 convolution followed by a 2×2 average pooling layer. The up-sampling layer contains transposed convolutional layer with kernel size of 2×2. Within the dense block, features from all pre-

ceding layers are concatenated along with the output of the current layer. All dense blocks are equipped with 3×3 kernels with L layers and k growth rate. MMDenseNet has a multi-band structure, which can better capture different frequency bands by using different parameters. Followed by the idea of *Blending* in [41], the same research team proposed **MMDenseLSTM** [44] which combines LSTM block and dense block in a unified architecture, in contrast to an output blending manner in [41]. Takahashi et al. showed that inserting only a small number of LSTM blocks in the up-sampling path for low scales (Bottleneck) is more effective in that it helps the network to capture global structure of the input. MMDenseLSTM was denoted as ***TAK1*** in SiSEC2018. This idea was later adapted in [68, 50]. In 2021, **D3Net** was proposed as another variation based on MMDenseNet, emphasizing multi-resolution modeling by the multi-dialated convolution within the dense block. In the dense block, between each 2D convolutional layers, a multi-dilated convolutional layer was inserted to apply different dilation factors according to which layer the channel come from. This helps to integrate the information from different receptive field and avoid aliasing.

- **U-net** [60] was named after its hourglass like Encoder-Decoder architecture, where the input was compressed through a bottleneck block and then was expanded to the original scale. U-net[71] was first deployed in the domain of Biomedical Image Segmentation. Later, Jansson et al. [43] applied this architecture to the task of singing voice separation. A stack of convolutional layer serves as encoder, where each down-sampling layer halves the size of the image while doubles the size of channel, yielding a deeper representation of the input. The deep latent vector is then decoded back to the original size of the image by a stack of up-sampling layers. Similar to the *inter-block skip connection*, U-net adds skip connections between the layers at the same scale level, for constructing better output with fine details and "raw" information. Later in [50, 51] also follow the U-net architecture, with specific design in encoder layer or intermediate block and bottleneck layer. Last but not least, a popular open source pre-trained model, **Spleeter** is also a U-net CNN model.

However, spectrogram is not as symmetry as 2D images since time axis and frequency axis contain different information of sound sources. Many researchers had tackled this issue by designing dedicated network structure. For example, Chandna et al. [42] proposed the first CNN based approach to separate MUSDB18-ish stems from mixture. Their network structure **DeepConvSep** utilized vertical and horizontal filters in order to capture timbre features and temporal evolution of different instruments. Inspired by Frequency Transformation (FT) blocks [63], Choi et al. [51] implemented (1) Time-Distributed Fully-connected network (TDF) and Time-Distributed 1D Convolutions (TDC) to extract time-independent features that help source separation without using inter-frame operations. (2) Time-Frequency Convolutions (TFC) which is a dense block of 2D CNNs, and TFC-TDF which contains a TFC block and a TDF block, and TDC-RNN block. These intermediate blocks are implemented within U-net down-sampling and up-sampling layer to capture dynamic global harmonic frequency relationship of sources. An intermediate block transform an input spectrogram-like tensor into an equally-sized tensor (possibly with a different number of channels). A down sampling/up sampling layer halves/doubles the scale of an input tensor. They found out that Time-Frequency Convolutions followed by a Time-Distributed Fully connected layers (**TFC-TDF-U-Net**) performs better compared to other fundamental building blocks. Following this architecture and with some modifications [11], Choi's subsequent work **KUIElab-MDX-Net** in [55] achieved the SOTA performance in *Vocal* and *Other* sources, in MDX2021.

Another approach is to build the model using 1D convolutional layers. The input tend to be **Waveform** so the training system is in an end-to-end fashion.

- **Conv-TasNet** [47] consists of Encoder, Separation and Decoder. Encoder and decoder used 1D Convolutional block, served as a learn-able transformation which can be jointly optimized during training. 1D Convolutional block consists of 1×1 convolution layer, parametric rectified linear unit (PReLU), global layer normalization (gLN) and depth-wise separable convolution. Depthwise

[11] In version 2, the number of intermediate block channels increased/decreased when down-sampling/up-sampling. U-net concatenation was changed to multiplication. Source specific pre-processing was applied

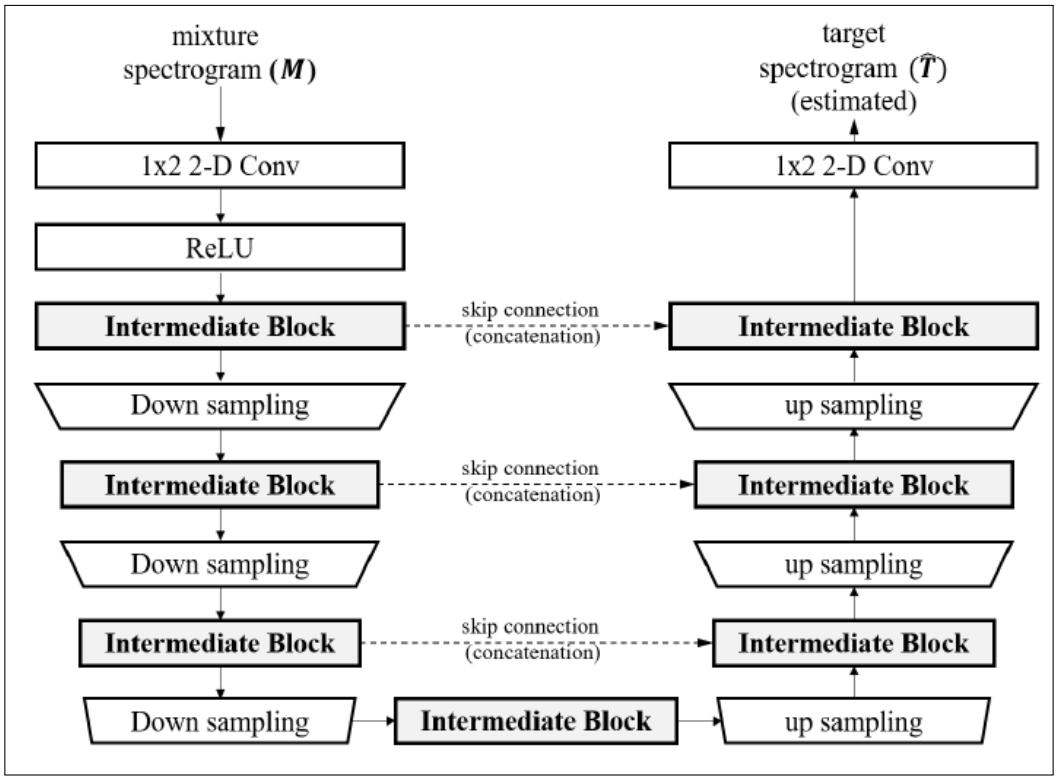

Figure 5: Architecture of TFC-TDF-U-Net

separable convolution makes the model size to be smaller, while maintaining high capacity.

The original TasNet architecture use LSTM as the separation module, which significantly limited its applicability. To avoid the decoupling of the phase and magnitude of the signal, the suboptimality of time-frequency representation and the long latency in calculating the spectrogram, a fully-convolutional time-domain audio separation network was proposed. Separator consists of several Temporal Convolutional Network (TCN), a replacement for RNNs in various sequence modeling tasks. Each layer in TCN consists of 1-D convolutional blocks with exponentially increasing dilation factors, which allows the network to model the long-term dependencies of the source signal while maintaining a small model size and shorter minimum latency.

However, in terms of generalization ability, Conv-Tasnet can't perform well when encountering new data, as reported in Speech Separation. For MSS, please note that in the implementation of Conv-Tasnet in [50], skip connection

branch was not implemented. A up-to-date implementation can be found in this repository [12]

- **Demucs** [50] is the first **waveform domain** approach to outshine spectrogram based approach. Proposed be Defossez et al., Demucs perform extraordinary well phase construction, which leads to excellent results in *Bass* and *Drum* separation.

  The original Demucs architecture is a U-Net encoder/decoder structure. A BLSTM block is applied between the encoder and decoder to provide long range context. Each encoder layer is composed of a convolution with a kernel size of 8, stride of 4 and doubling the number of channels (except for the first layer, which sets it to a fix value, typically 48 or 64). It is followed by a ReLU, and a 1×1 convolution with Gated Linear Unit (GLU) as activation function. The 1×1 convolution double the channels and the GLU halves them, keeping them constant overall.

  Two layers of BLSTM with 2048 hidden units serve as the bottleneck and are followed by a linear layer to halves the output from BLSTM layers.

  For Decoder, transposed convolutional layers are used to perform the up-sampling function rather than using an up-sampling layer, in that it requires 4 times less operations and memory than linear interpolation followed by a convolution with a stride of 1, as implemented in Wave-U-Net [48]. Each decoder layer is composed of a 1D convolutional layer with a kernel size of 3, stride of 1 with GLU, and then a transposed convolution with kernel size of 8 and stride of 4 and ReLU as activation function.

  In MDX2021, Defossez proposed a hybrid version of Demucs (**Hybrid Demucs**). It features multi-domain analysis: a dual U-net structure with temporal and spectral branches. These two branches are mainly made of 1D convolutional layers but each one performs along different directions. For spectra branch, convolution was performed along the frequency dimension for every

[12] `https://github.com/tky823/DNN-based_source_separation/blob/main/src/models/tdcn.py`

frame in STFT. After 5 encoding layers, the frequency axis has only 1 dimension left and the information was stored in lager channel numbers. For temporal branch, convolution was performed along the time dimension and after 5 layers of condensing, the output has $T/1024$ time steps, which equals to the number of frames in STFT [13]. By this sophisticated alignment design, temporal (waveform) and spectra (time-frequency) domain were able to be **merge** together. The output from two branches were summed together before going to a shared encoder/decoder layer. Since the bottle neck of BLSTM was removed, these two shared layers serves as a bottleneck. For each encoder layer, two compressed residual branches were added between the original 1D convolutional layers. Inside a residual branch, there are dilated convolutions performed regarding to time dimension only. a two-layer BLSTM block with limited span of 200 time steps and Local Attention [72] were implemented in the inner most encoder layers (5,6). For detailed information of Hybrid Demucs, please refer to [53].

2.3.4 Connection between Modules

There are many types of skip connection: Residual style skip connection, DenseNet style connection, U-net style skip connection.

- Residual style connection, which uses skip connections between consecutive modules to allow for deeper propagation of information and to deal with the degradation problem when increasing the depth of the architecture. Element-wise addition is usually applied, for example in [47, 53, 62]. Note that ResUNetDecouple has 143 layers, which is the deepest model by now. Concatenation was applied in [50, 68]

- DenseNet style connection. Within dense block, every layer is densely connected, which means the current layer receives the output from all preceding layers. Concatenation was used in MMDenseNet [45] and MMDenseLSTM [44]

[13] Calculation of number of frames in STFT: $floor(T - 4096)/1024 + 1$, but padding may have applied.

to ensure feature reusability thus no need to learn redundant feature maps. D3Net [73] involves a novel multi-dilated convolution layer that has different dilation factors in a single layer to model different resolutions simultaneously. Addition was applied in that layer to integrate information from very local to exponentially large receptive field.

- U-net style skip connection is a rather restricted type of skip connection, which only links the same level representation together rather than applying addition to all the skip connection output from all previous modules like Conv-TasNet [47]. Concatenation was used in original U-net [43], ResUNetDecouple [62] and TFC-TDF-U-Net in [51]. Element-wise addition was implemented in Demucs [50, 53]. Element-wise multiplication was applied in KUILAB-MDX-NET [55].

2.3.5 Bridging & Blending & Hybrid

Bridging means sharing weights among different sources within the model. This is often seen in dedicated models to compensate for the individual extraction networks. For example, in **X-UMX** [74], Sawata et al' proposed a bridging operation to utilize the relationship among instruments by adding averaging operators across different single head models, resulting a *CrossNet* structure. Besides, a *Combination Loss* (CL) was proposed to consider the loss of all possible sources combination. Another example is the single-head model **LaSAFT** [52] proposed by Choi et al', in which a weight modulation mechanism across different heads was used to generate a specific source.

Blending means combining the outputs from independent models to estimate sources. Firstly proposed by Uhlich [41], who was blending the raw neural network output from **UTL2** and **UTL3** before applying multichannel Wiener filtering (MWF). In MDX2021, the model **Danna-Sep** [75] performs a weighted average of individual outputs from three models, namely U-net, X-UMX and Demucs v2. One iteration of WMF was applied to X-UMX and U-net before blending. Another model **KUIELab-MDX Net** [55] by the same author features an additional CNN called *Mixer*, which applies 1D convolution to four estimated sources together with mix-

ture in the waveform domain. Four refined stems are generated using this U-net blending module. Finally, blending was also applied between the output of *Mixer* and a pre-trained Demucs v2.

Hybrid means the model can process the information from waveform domain and time-frequency domain. **Hybrid Demucs** [53] is the first model which explores this direction and it achieved the first place in leader board A of MDX2021. A brief introduction of the model structure can be seen above.

2.3.6 Conventional MSS Research on Multi-track Dataset

Multi-track dataset was often served as additional four-stem dataset to MUSDB18 dataset. For example, [45] use MedleyDB [19, 20] to train the model. Uhlich [41] use a subset of MedleyDB to train UTL2 to study the effect of more training data.

For the sources outside from the four-stem scenario, multi-track dataset serves as an important dataset. For example, a five-stem version of Spleeter [76] was trained on a large internal dataset called BEAN dataset [77], which contains 24097 songs and total 79 hours of audio stems. Hung et al. [78] trained models for Electric Guitar, Acoustic Guitar and Piano based on MM dataset, which combines MedleyDB and Mixing Secrets v1 [21]. As a synthesized dataset, Slakh2100 [15] is less used but it is also an important dataset for *guitar, piano*.

2.4 Hierarchical Music Source Separation Research on Four-stem Dataset

Up to now, there is no reported research in this research scenario.

As mentioned in previous Section 1.1, there is a hidden hierarchical structure within the four-stem dataset, which is $mixture - accompaniment - \{bass, drums, other\}$. However, to the best of our knowledge, there is no literature reporting generating *accompaniment* and *{bass, drums, other}* from the mixture at the same time, in order to improve the separation performance of either *accompaniment* or any of

{bass, drums, other}. In Music Demixing Challenge 2021, Team ByteMSS presented several tricks to improve the performance on *Other* stem. One of them is to feed the estimated *accompaniment* stem to the model as input and do separation again, and the SDR result got improved by 0.105dB (from 4.635dB to 4.74dB). This information could be a motivation for us, to follow a sequential way to generate *other* stem from *accompaniment* rather than *Mixture*. This is a naive way of conducting HMSS on four-stem dataset.

2.5 Hierarchical Music Source Separation Research on Multi-track Dataset

Here in this section, we present the definition of HMSS in [14]: A HMSS system that can generate a set of sources that share hierarchical relation. In other words, they are different levels of submixes in the same instrument family, such as *Level3 Pluck Strings - Level2 Guitar - Level1 Electric Guitar*. We then try to find out the necessary technical conditions for a MSS system to become a HMSS system. Differences and similarities between the building blocks of the state-of-the-art implementation of HMSS [14] and those in conventional MSS are highlighted, which ultimately leads to our proposed methods in Section 4.1.

2.5.1 Defining HMSS

What makes a hierarchical music source separation system different from a conventional music source separation system? After closely inspecting the implementation details of the baseline proposed by Manilow et al. [14], here we present several distinct components of HMSS: **(1) A multi-track dataset with more than three tracks; (2) A hierarchy more than two levels designed for the multi-track dataset; (3) A data loading scheme which prepares a set of hierarchical related sources as ground truth sources (more than two sources), corresponding to a manually defined hierarchy**.

As for (1), we regard HMSS can work on any dataset with more than three sources

of each song. Because the submix of only two sources can only be the mixture. However, a multi-track dataset which contain more diverse categories of instruments is necessary to generate a semantically meaningful hierarchy. In [14], Slakh2100 dataset with up to 34 categories of instruments was used. One should note that a multi-track dataset is a necessary component for HMSS but not a distinct feature because there were MSS systems trained on multi-track dataset such as MedleyDB and Mixing Secret v1 [78].

For (2), a hierarchy should contain more than two levels. It declares the grouping rules of different levels of sub-mixes. In our baseline system, hierarchy was documented in a .json file and an example can be found in here. In this hierarchy file, the top level contains four categories: mid-range strings and keys (guitars, keyboards, and orchestral strings), bass instruments (acoustic and electric basses), winds (flutes, reeds, and brass), and percussion (drum sets and chromatic percussion). The middle level has seven categories (e.g., from mid-range strings: orchestral strings, guitars, keyboards, and electric keyboards), and the lowest level has eighteen categories (e.g., from guitars: clean guitars, and effected guitars). As you can see, compared to the four-stem dataset in MSS, the instrument hierarchy is deeper. However, due to the fact that Slakh2100 does not contain *vocals* and most of the instruments are from *other* stem.

Secondly, please note that the three-level hierarchy is not optimal for every instrument family in that not every instrument family is diverse enough to be grouped in a three-level hierarchy in a meaningful way. For example, according to the hierarchy defined in the baseline, the top level of *percussion level3* only contain *percussion level2* as a source for lower level, and then it was divided into *drums* and *chromatic percussion*. The same situation happened for *bass level3*. This indicated that, for this dataset, a three-level hierarchy is **redundant** for certain instrument family which does not contain enough number of different sub-class of instruments. This unbalanced instrument distribution within the dataset can lead the identical output in the lower level outputs, as pointed out in [14].

Based on the manually defined hierarchy, a data loading scheme (3) should be imple-

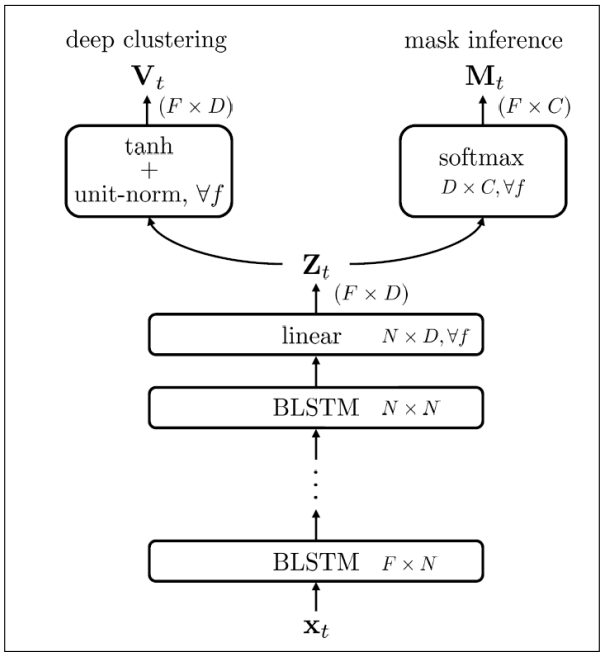

Figure 6: Structure of the Chimera Network

mented to feed the model with a set of target sources that are hierarchical related. In the baseline, for each mixture in the dataset, the saliency of each hierarchical submix was first calculated in advance, to save memory when generating submixes on the fly.

In terms of the model architecture, Manilow et al. [14] studied two multi-target models namely **Mask-Inference** and **Query-By-Example (QBE)**. As illustrated in Figure 6, Mask-Inference is one head of the chimera network proposed in [66]. The Mask-Inference network's body is comprised of 4 bidirectional long short-term memory (BLSTM) layers with 600 hidden units in each direction and dropout of 0.3, followed by a fully connected layer with sigmoid activation function that outputs masks. As shown in Figure 7, the QBE model are composed of two sub-networks, a query net and a masking net. The query net is composed of 2 BLSTM layers with 600 nodes in each direction and dropout of 0.3, followed by a fully-connected layer that maps each time-frequency bin to an embedding space of 20 dimensions. The masking net is the same as the Mask-Inference model, with a larger input feature vector size to accommodate the concatenated query anchor with mixture vector.

In the original paper, Manilow et al. proposed four types of HMSS systems, along

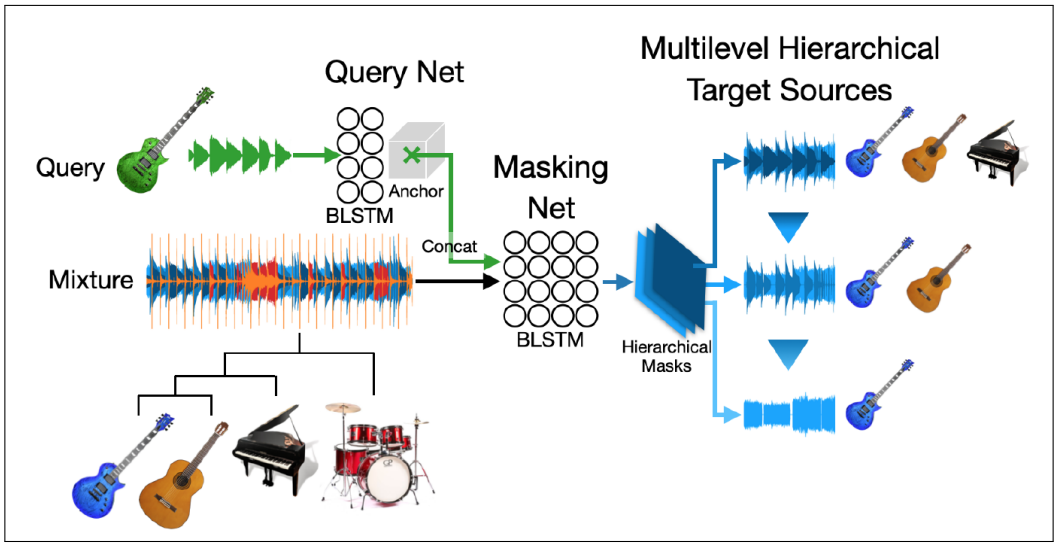

Figure 7: Signal Flow of the Query By Example System in [14]

two dimensions: whether they are single-instrument (i.e., Source-Specific Separation, denoted as SSS) or multi-instrument (i.e., query by example, denoted as QBE), and whether they output one source from a specific hierarchical level or multiple sources of multiple hierarchical levels. In our thesis, Mask-Inference model for Multilevel SSS serves as our baseline, in that it has a better SI-SDR performance than QBE in separating single instrument such as *clean guitar* according to [14]. The detailed information of experiments can be seen in Chapter 4

Chapter 3

Dataset Construction

In this chapter, we first present the detailed construction process of MS21: Data Gathering 3.1.1, Semi-automatic Annotation Generation 3.1.2. We also introduce the designed hierarchical structure of MS21 in Section 3.1.3, which was utilized to implement Automatic Stem Generation 3.1.4. In Section 3.2, We perform quantitative analysis among a collection of dataset used in Music Source Separation (MSS).

3.1 Dataset Construction

3.1.1 Data Gathering from *The 'Mixing Secrets' Free Multitrack Download Library*

The 'Mixing Secrets' Free Multitrack Download Library [1] is an additional source for the book *Mixing Secrets For The Small Studio*, written by Mike Senior. There are more than 500 multi-track projects which can be freely downloaded for mixing practice purposes. All projects contain uncompressed WAV files (24-bit or 16-bit and 44.1kHz or 48kHz sampling rate) and mp3 mixture for reference. A subset of projects contain un-mastered wav for mastering practise. To maximize the mixing flexibility, the contributors follow a guideline to provide audio with well-defined filename and aligned.

[1] `https://cambridge-mt.com/ms/mtk/`

A python script was used to gather the necessary information from the website, such as *Track Name, Artist Name, Broad Genre, Sub Genre, Archive URL, Preview MP3*. Then, a unzipping process was carried out. After the elimination of broken archives, we have more than 500 songs in total.

3.1.2 Semi-automatic Annotation Generation

An automatically instrument classification at multi-track level was conducted utilizing the naming convention of multi-tracks. Specifically, we created text filtering conditions to group the instruments at the same level together. For example, we group *Rhode*, *Wurlizer* and *Keys* to *Electric Piano*. Results were documented in a CSV (Comma Separated Value) file. Then, manually correction was carried out to correct the automatically generated labels, based on the actually recorded music content within the multi-track. Besides, with special attention on the bleeding of vocal tracks, a label called *Vocal Quality* was annotated. It should be noted that, because of live recording setup, multi-tracks of certain instruments (each instrument of drum set) may contain leakage. Such contaminated tracks makes it impossible to separate the individual instrument (e.g. Separating *Tom-Tom* and *Hi-Hat* from *Drum Set*).

While we sill labelled those drum instrument as long as their leakage is from other drum instrument, we created a category named *Ambient Microphone* to get rid of those contaminated tracks containing tracks from totally different instrument family members such as *Room, Mainpair, Ruffmix*. For the "outsider" tracks of the current instrument ontology, such as *Rain, Kick Trigger*, we discarded them by labelling them as *Unused*. Note that we still keep those contaminated vocal track in use with additional *Vocal Quality* label, waiting for future solution of dealing with leakage data (Refer to Future Works Section 6.2.1). *Lead Back both* and *Guitar Both* are generated automatically by checking the corresponding instrument columns. *Broad Genre, Sub Genre* metadata was aggregated from the previous data gathering process. Finally, over 500 songs were annotated using more than 70 instrument labels, 2 genre labels and 1 leakage label.

3.1.3 Hierarchical Structure of Multi-track Dataset

As briefly mentioned in [19], there is a hierarchy of the audio files for each song, namely *Raw*, *Stems* and *Mix*. This hierarchy leads to the folder structure of MedleyDB, where RAW folder contains unprocessed multi-tracks and STEMS folder contains professionally mixed stems, each stem corresponding a specific sound source. Since stems are stereo audio components of the final mix and include all effects processing, gain control, and panning, researchers in MSS tend to view stems as the smallest units to work on. The mixing or grouping standard were not documented in [19] and no analysis was done to present the track-to-stem process.

After closely inspecting the relation between the instrument label of raw audio files and the one of stem audio files in MedleyDB, we found some possible errors: 1) Contradicted labels of raw tracks exist in one stem, e.g. raw tracks labelled as *clean electric guitar* are grouped in a stem labelled as *distorted electric guitar* or vice versa; 2) Different levels of instrument label exists: Some labels overlap with other in terms of ontology, e.g. *auxiliary percussion* and *drum set* share many common instruments, while some label is a subset of another, e.g. *drum set* and *drum machine*[2].

One possible explanation for (2) is that, when creating labels for stems, annotators did not follow a global instrument ontology. Raw multi-tracks were mixed to create stems mostly because they should be viewed as an independent source according to the situation of that specific song. For example, for music genre of electronic, *drum machine* is an independent source while in the genre of pop, *drum machine* is a complementary instrument along with *drum set*. For world/folk music, stems which includes *cymbal* and *tom-toms* are labelled as *auxiliary percussion* rather than *drum set* because the use of *drum set* is rare in that genre. As a result, there are 82 instrument labels for stems, but there are not 82 distinct instruments.

In order to compare the instrument distribution between MedleyDB and MS21, we created a hierarchy to define the relations of different level instrument labels,

[2] The detailed documentation can be seen in Appendix

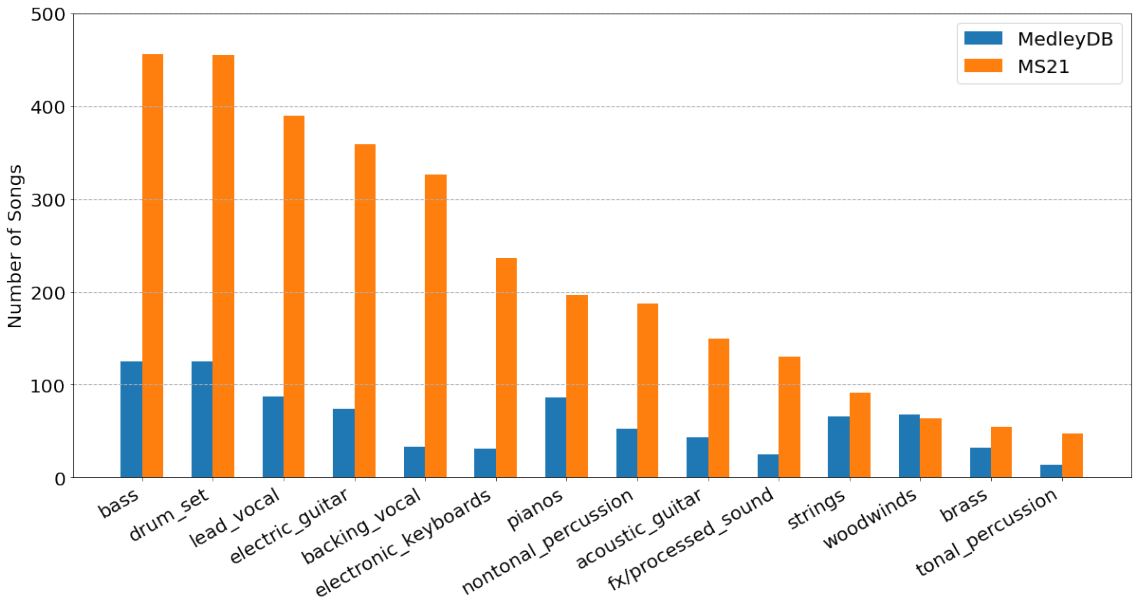

Figure 8: Instrument Distribution of MS21 and MedleyDB

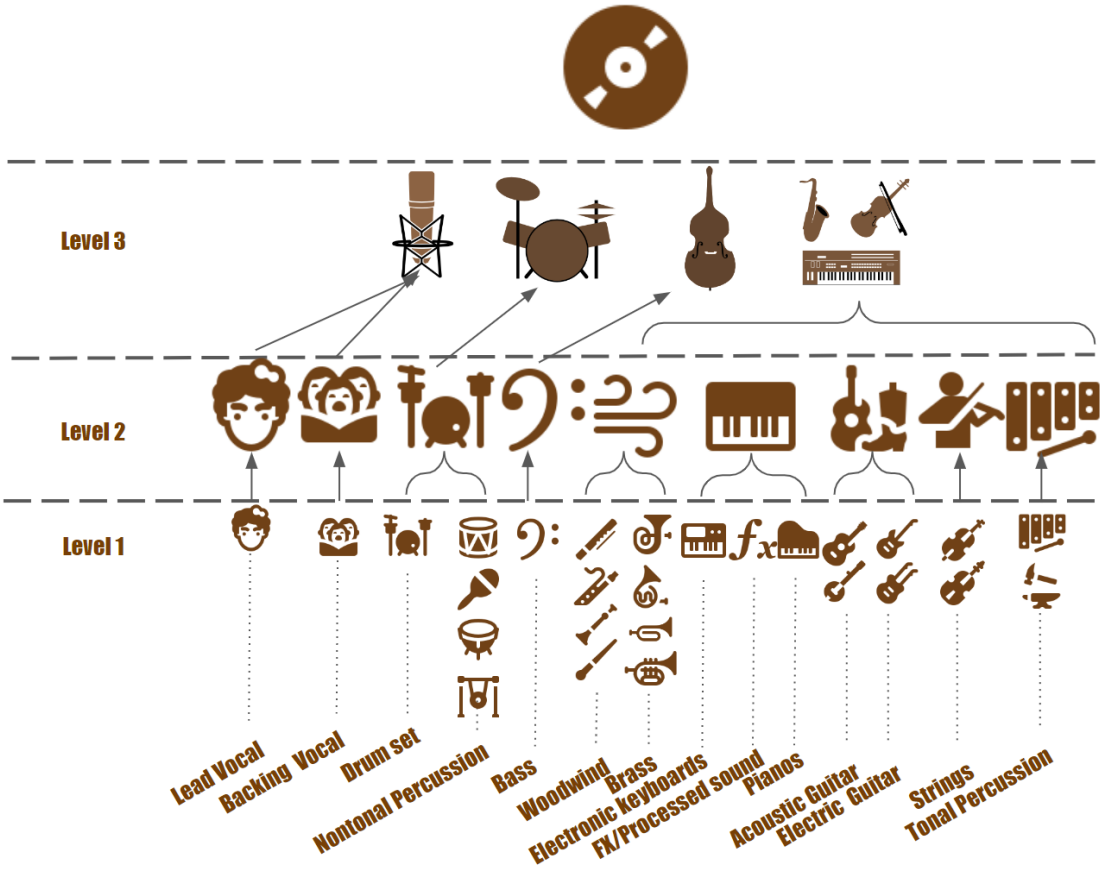

Figure 9: The Hierarchy of MS21, an Extension from MUSDB18

grouping the lower level labels to higher-level labels. The audio structure is a three-level hierarchy, namely *track-to-inst.*, *inst.-to-stem*, *stem-to-stem*. In *track-to-inst.*,

we group the most fine-grained instrument labels into 14 instruments[3], as shown in Figure 8 and in the Level 1 of Figure 9. In *inst.-to-stem* stage, we follow the four-stem tradition, and group the 14 instruments into *vocal*, *bass*, *drums* and *other*, in the hope that conventional MSS research legacy can be leveraged. In *stem-to-stem* phase, we further create *accompaniment* by grouping *bass*, *drums* and *other* together, leaving *vocal* as another source in this level[4]. This hierarchy serves as the hierarchy we used when conducting HMSS experiment on MS21 in Section 4.4.2, as shown in Figure 9.

3.1.4 Automatic Stem Generation

Considering memory overflow issues when mixing the instrument stems from the raw multi-tracks, we provide three levels of pre-computed stems: Instrument stems, MUSDB-ish stems and the mixture file, following the hierarchy mentioned in the previous subsection. In the first stage *track-to-inst.*, multi-tracks that belong to the same instrument class are mixed into one instrument stem. Loudness normalization of ITU-R BS.1770-4 standard implemented in pyloudnorm repository was applied when creating stems. According to [79], a relatively simple ITU-R BS.1770 integrated loudness was preferred over other complex psycho-acoustic model. Integrated Loudness was used for non-percussive instrument stem and target loudness was set to -25 LUFS. Whereas peak normalization was applied to percussive instrument stem such as drum set and non-tonal percussion. The target loudness level for peak normalization is set to -1 LUFS. In the second stage *inst.-to-stem*, MUSDB-ish stems were also generated specially for conventional MSS research. The same loudness normalization process was carried out. By directly loading the MUSDB-ish stems rather than mixing the relavant multi-tracks on the fly, computational resources can be significantly reduced and the training process can speed up. Finally, the mixture file is created by mixing the MUSDB-ish stems. We neglect stem generation for the third stage *stem-to-stem* because memory cannot not be a big

[3] Note that for MedleyDB, we regard the vocal tracks with "melody" label as "lead vocal" and those vocal track without "melody" label as "backing vocal"

[4] Note that we leave the hierarchy where *backing vocal* is grouped to *accompaniment* for future work.

problem at this stage.

Note that only linear mixing was applied after loudness normalization, which means no equalization, dynamic range compression and distortion was applied during mixing. As to our best knowledge, a baseline of auto-mixing system does not exist. However, if we only focus on the separation of top level sources such MUSDB-ish stems, **automixing algorithms can be applied during Data Augmentation process**. We leave this idea for future works as discuss in Section 6.2.2.

Following the standard multi-track dataset folder structure proposed in MedleyDB, we put multi-tracks in RAW folder, the automatically generated stems in STEM folder, the metadata file and mixture audio file were left in the root directory of each song.

3.2 Dataset Statistics

3.2.1 Split of the Dataset

We did the split of the dataset following several principles:

- Same artists would not be in both train and test sets, otherwise the system would overfit on the artist, as mentioned in [80]. The detailed split of the dataset can be found in Appendix.

- Different sub sets should share similar content distribution, e.g. instrument. We here present the genre distribution and the instrument distribution of each subset of MS21, as illustrated in Figure 10 and Figure 11. Note that for some artists, their music genre are hard to define so in The 'Mixing Secrets' Free Multitrack Download Library [5], their Sub Genre label is "Various Styles" while sharing different Broad Genre labels. The detailed report of those songs can be seen in Table 10 in Appendix. Besides, as shown in Table 9, the raw annotations of Sub Genre is too detailed, in order to compare the genre

[5] urlhttps://cambridge-mt.com/ms/mtk/

distribution across different datasets, we validated the sub genre label with a taxonomy of music genres. Finally, distributions over 16 genres was computed, for those genres that do not fall into the category of the 16 genres, we labeled them as "Other" in the final column of Figure 10 and Figure 13.

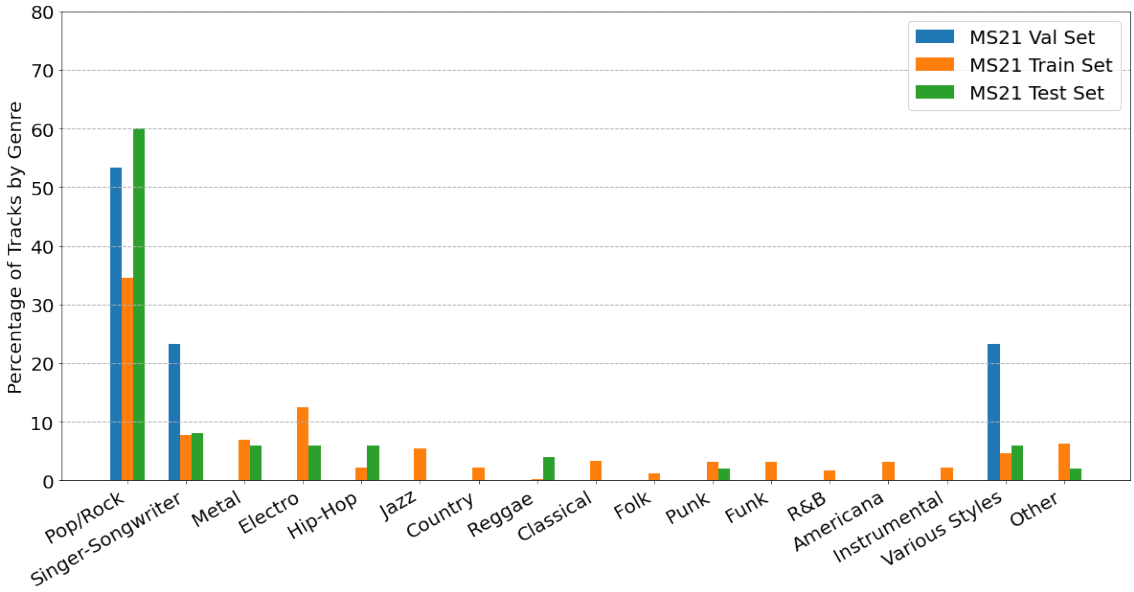

Figure 10: Genre distribution of each sub set of MS21

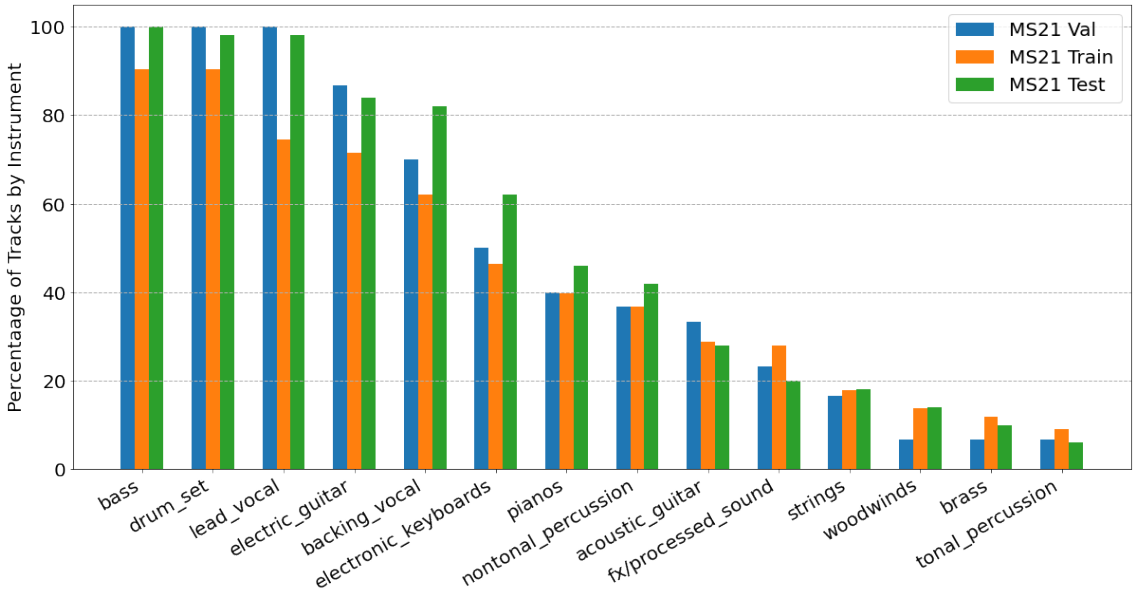

Figure 11: Instrument distribution of each sub set of MS21

- When creating test set of MS21, we tried to include all the songs in the test set of MUSDB18, since these two datasets share one data source and the unified

test set can be useful when comparing results. In the end, only 7 of 50 from the test set of MUSDB are missing from MS21 dataset.

3.2.2 Summary: MS21 in Cross Datasets Comparison

Comparison across different dataset should follow specific criteria. In [19], criteria such as *size*, *duration*, *quality*, *content*, *annotation*, *audio* are addressed when creating MedleyDB multi-track dataset. Besides,according to [81], in addition to total training duration, *separation quality* and *diversity* were mentioned as important criteria of datasets in MSS. We here follow this framework when evaluating our proposed dataset, also with our further development of the theory framework. Here we present our criteria when building our new dataset along with further discussion.

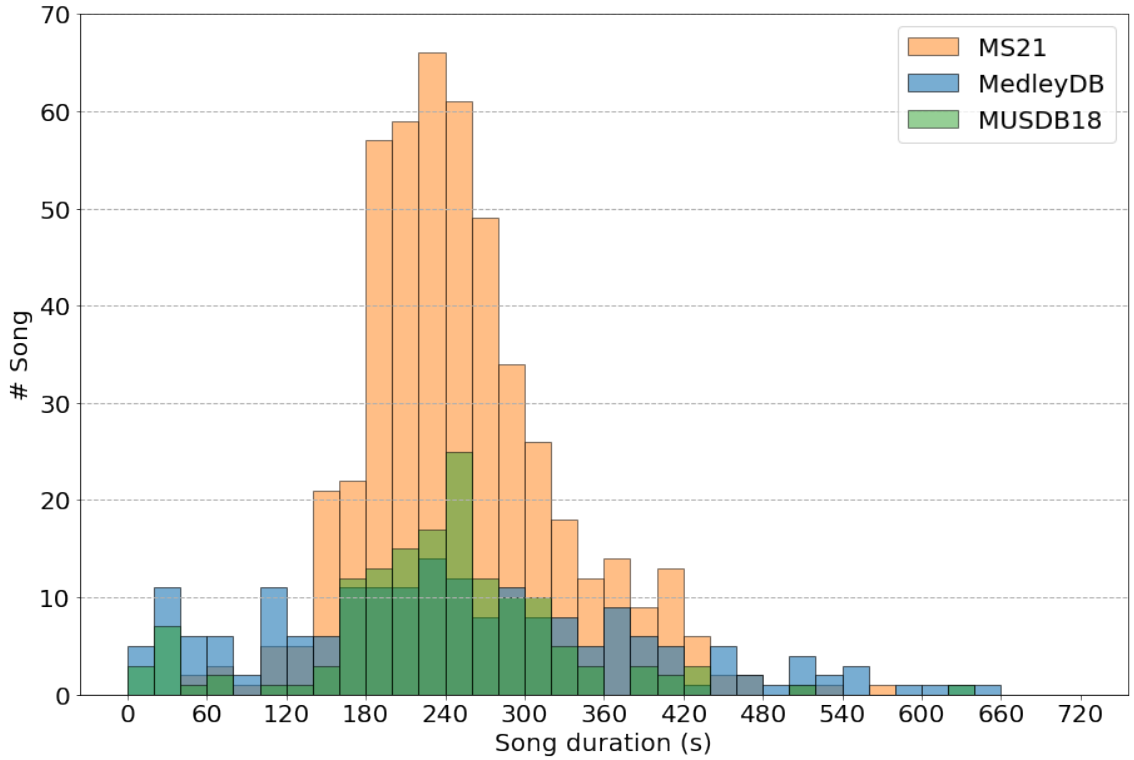

Figure 12: Length Distribution of the dataset

- **Size**: the dataset should be at least one order of magnitude greater than previous dataset.
- **Duration**: the dataset should primarily consist of full length-songs. Besides, total training duration matters.

Dataset	Size	Duration(h)	Num. Tracks	Instr. Cat.	Vocal %	Leakage %	Stereo/Mono
MS21	500	34.2	20±13	70	84	20	Stereo
MedleyDB	196	12.7	18±17	82	44	41	Stereo
MUSDB18	150	9.8	4	4	100	0	Stereo
Slakh2100	2100	145.0	11±3	34	0	0	Mono

Table 3: Statistics across multi-track dataset in Music Source Separation

- **Quality**: the audio should be of professional or near-professional quality. Quality can be divided into **Performance Quality**, **Recording Quality** and **Production Quality**. For performance quality, dataset should be consisted of music composed and performed by professional or near professional artists. For recording quality, the recording session should be carried out in a recording studio with proper acoustic treatment and adequate audio engineering devices. The audio format of the dataset should be at least 44.1kHz, 16bits and stereo, which is the industrial standard for commercial music recording. Note that, musical leakage or **bleeding** can be found in multi-track due to live-recording schemes. This phenomenon should be carefully addressed especially for multi-track dataset. For production quality, the stems and mixture should be produced by professional or near professional sound engineers in a recording studio.

- **Content Diversity**: the dataset should consist of songs from a variety of genres. A diverse genre distribution of the dataset can lead to a relatively balanced instrument distribution, which is of great concerned in the task of source separation.

- **Annotation**: the annotations must be accurate, well-documented and available publicly.

- **Audio**: each song and corresponding multi-track session should be available for research purposes.

We here further present the distinct features of MS21 following the criteria proposed above. Statistics of MS21, MedleyDB, MUSDB18 and Slakh2100 were shown in Table 3. In terms of **total mixture duration** of the dataset, MS21 is the largest one compared to MedleyDB and MUSDB18. However, if we consider the number

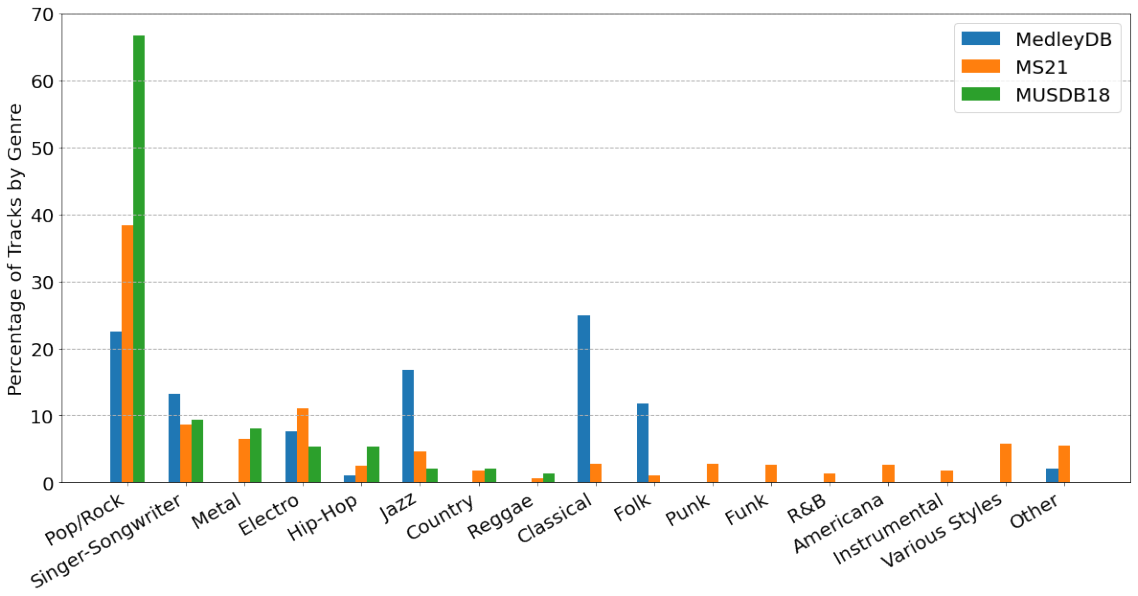

Figure 13: Genre distribution comparison across dataset

of multi-tracks per song, the size of MS21 is significantly larger than MUSDB18. As shown in Figure 12, the **length distribution** of MS21 and MUSDB18 is more clustered to the mean value (240 seconds) while MedleyDB shows a flat and scattered distribution. What's more, MedleyDB and MUSDB18 even contains songs under 40 seconds. For **content diversity**, as seen in Figure 13, although *Pop/Rock* genre remains a predominant status (38%), MS21 contains a wide variety of genre and some genres such as *Punk, Funk, RB, Americana* exist exclusively in MS21. In contrast, MUSDB18 shows huge bias in *Pop/Rock* genre (over 65%). MedleyDB shows a unique distribution in the genre of *Folk* (11%), *Jazz* (17%) and *Classical* (25%), while other two dataset contains very few songs in these three genres. Genre distribution affects Instrument distribution. As shown in Figure 8, compared with MedleyDB, MS21 contains more number of instrument in every category except for woodwinds, which is a distinctive instrument family in *Classical* and *Folk* genre.

Derived from Mixing Secrets Website, one characteristics of MS21 is that it contains songs with **human performance** and **professional recording quality**. Besides, for different recording setup, the raw multi-tracks may contain different levels of leakage or *Bleeding*. In MedleyDB, a song-level label of bleeding was provided and it shows 41% of songs contain contaminated stems. One possible reason is due to the

live-recording setup, especially MedleyDB contain many songs in genre of *Classical* and *Jazz*. In MS21, leakage situation is currently only annotated for vocal source, not for all multi-tracks of a song. The annotation was done manually by a single annotator (the author), who listened to all the vocal multi-tracks of all the songs. If any bleeding was heard, then this song would be labeled as "0" in *Vocal Quality*. As shown in Table 3, it means for those songs that contain vocal tracks, 20% of them contain leakage from other instruments in MS21.

For **annotation**, we provide detailed annotation as mentioned in subsection 3.1.2. This annotation was kept in the similar style of MedleyDB. Compared to MUSDB18, which does not provide more information other than the four well labelled stems, one advantage of MS21 is that instrument information about each stem was provided. With these information, a researcher can know what instrument each stem contains, which would be helpful when dealing with *Other* stem in conventional MSS.

For **audio**, the audio files and annotations of MS21 is freely available for research purpose. No commercial purpose was allowed for using this dataset. For other purpose, one should contact the artist directly.

Chapter 4

Methods

In this chapter, we will first provide the mathematical definition of the mask based method used in Music Source Separation (MSS)in Section 4.1. We then propose a new theory called **Mask Relation** in Section 4.2, which described the **Direct Mask** and **Sequential Mask** based method when estimating one particular source. This theory was based on the hierarchical relations observed in the multilevel target sources in the context of Hierarchical Music Source Separation (HMSS).

Then, several experiments would be presented. The implementation parameters and training schemes would be presented. Experiment 1: A multilevel single-instrument experiment was designed to explore the features of implementing the **Sequential Mask** in HMSS. Then, Experiment 2: Multilevel single-instrument separation would be carried out on MS21, to compare with Slakh2100 [15].

4.1 Mask based approach in Music Source Separation

We first present a mathematical definition of the Music Source Separation problem. We assume that the time-domain mixture signal x consists of J sources, i.e.,

$$x = \sum_{j=1}^{J} s_j; \tag{4.1}$$

where s_j denotes the time-domain signal of the j-th source. In this general formulation, the source s_j have no relation to source s_k, for $j \neq k$. Furthermore, for mask based approach, we assume that the output of the DNN is a mask M_j which can extract the j-th desired source from the mixture spectrogram $X = \mathcal{S}\{x\}$:

$$\hat{s}_j = \mathcal{S}^{-1}\{\hat{S}_j\} = \mathcal{S}^{-1}\{\hat{M}_j \cdot X\}; \tag{4.2}$$

where $\mathcal{S}$ and $\mathcal{S}^{-1}$ denotes the STFT and inverse STFT. Furthermore, $\hat{s}_j$ and $\hat{S}_j$ are the predicted results of time and frequency domain ground truths s_j and S_j.

We hereby introduce the hierarchy related declaration when defining the HMSS problem. Considering a hierarchy with L levels, we denote by $S_{l,j}$ the j-th source type related node at hierarchy level l, for $l = 1, \ldots, L$, where we assume that the set of leaf source types $S_{1,j}$ cannot be decomposed into further source types, and $S_{L,1}$ is the mixture X at the top of the hierarchy and includes all source types. We further denote $\mathcal{C}_{l,c}$ as the set of indices of the child sources at level $l-1$ of $\mathcal{S}_{l,c}$. The hierarchical relation of multilevel submixes can be denoted as:

$$S_{1,j} \subseteq S_{2,j} \subseteq \cdots \subseteq S_{L-1,j} \subset X; \tag{4.3}$$

where

$$S_{l,c} = \sum_{c' \in \mathcal{C}_{l,c}} S_{l-1,c'}; \tag{4.4}$$

for $l = 2, \ldots, L$. A clear hierarchical path is shown due to the **Inclusion Relation**.

4.2 From MSS to HMSS: From Direct Mask to Sequential Mask

Generating estimated sources from mixture is a heuristic approach in MSS, because mixture contains all the necessary information for source reconstruction. To achieve this end, we can train a model to generate the source directly. An estimating process is usually done by multiplying the estimated masks with the mixture. Considering the HMSS problem set up, we rewrite the definition in Equation 4.2 as:

$$\hat{S}_{l,j} = \hat{M'}_{l,j} \cdot X; \tag{4.5}$$

This is the **Direct Estimation Process**, where we denote $\hat{M'}_{l,j}$ as the **Direct Mask** of source type $S_{l,j}$ at hierarchical level l. Note that, **up to now, all mask based approach in the literature only use Direct Mask in the estimation process.**

However, in HMSS, we have different ways to generate the desired sources. Because of the inclusion relation shown in Equation 4.3, we can get the estimated source type from its hierarchical related source which is at higher level of the hierarchy. We first denote $M_{l,j}$ as the **Sequential Mask** of source type $S_{l,j}$. The definition can be shown as follows:

$$\hat{M}_{l,j} = \frac{\hat{S}_{l,j}}{\hat{S}_{l+1,j}} \tag{4.6}$$

We then formulate the **Sequential Estimation Process** as:

$$\hat{S}_{l,j} = \hat{M}_{l,j} \cdot \hat{S}_{l+1,j} = \hat{M}_{l,j} \cdot \hat{M}_{l+1,j} \cdot \hat{S}_{l+2,j} = \hat{M}_{l,j} \cdot \cdots \cdot \hat{M}_{L-1,j} \cdot X; \tag{4.7}$$

Apart from the mathematical definition, we here provide Figure 14 to illustrate the different source estimation process based on the combination of these two semantically different masks. The association between different masks is called **Mask Relation**.

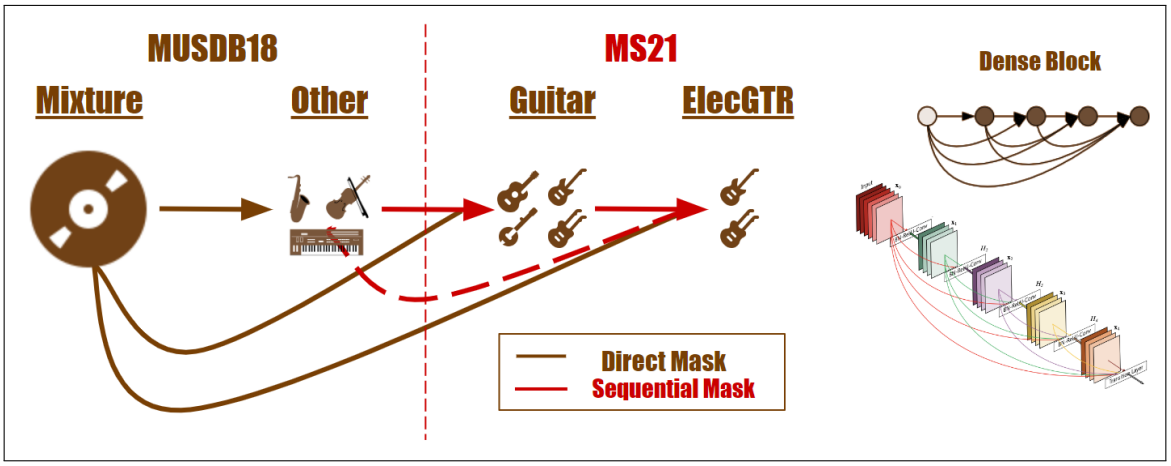

Figure 14: Mask Relation: Brown lines symbolize the Direct Estimation Process 4.5 using Direct Mask and red arrows represent the Sequential Estimation Process 4.7 using Sequential Mask.

Based on mask relation, we can discover alternative source estimation process. For example, apart from using direct mask or purely sequential mask to estimate *Electric Guitar*, we can first use direct mask to estimate *Guitar* and then use sequential mask to get *Electric Guitar*. In this thesis, we will only focus the sequential masked based method (shown in straight arrows) and direct mask based method (shown in curve arrows). Note that since there is only direct mask for all the highest level sources, e.g. *other*, we don't distinguish it and view it as sequential mask when we refer to sequential estimation process. On the right hand side of Figure 14, we point out the similarity between the proposed Mask Relation and DenseNet [82] architecture, which can be an inspiration for a potential neural network architecture.

Ideally speaking, the multiplication of sequential masks of all higher level sources equals to the direct mask of the current level source, formulated as:

$$\hat{M}'_{l,j} = \prod_{i=l}^{L-1} \hat{M}_{i,j}; \tag{4.8}$$

But in practice, this equation might not hold true due to the limited mask inference capability of poorly performed model.

Each sequential mask represents the different hierarchical relation between the source from neighbouring level, which is actually a form of **Hierarchical Constraint**. In

[14], hierarchical constraint was firstly proposed as masks at higher levels in the hierarchy must apportion at least the same amount of energy as the masks at lower level, as:

$$\hat{M}_{l,j} = max(\hat{M'}_{l,j}, \hat{M}_{l-1,j}); \tag{4.9}$$

Ideal Ratio Mask (IRM) is generally used as the target mask. Because of the fact that IRM ranges from 0 to 1, sequential mask naturally functions as the energy-wise hierarchical constraint, which can be regarded as a source energy allocation for every time-frequency frame of mixture magnitude spectrogram.

4.3 Loss Function

The first important feature of loss function is measuring the difference between estimated and target content. In early stage of deep learning applied in MSS, Square Error or Mean Square Error (MSE) were widely used in the literature [39, 40, 41, 42, 45, 44, 46, 68, 51]. MSE on magnitude spectrograms can be formulated as:

$$MSE_{freq} = \frac{1}{N}\sum_{j=0}^{N}(|S_j| - |\hat{S}_j|)^2 \tag{4.10}$$

In [43, 47, 50, 53, 55], L1 loss was used, either on spectrogram and waveform signal. L1 Loss Function is used to minimize the error which is the sum of the all the absolute differences between the target source and the estimated source. L1 Loss on magnitude spectrograms can be formulated as:

$$L1_{freq} = \frac{1}{N}\sum_{j=0}^{N}||S_j| - |\hat{S}_j|| \tag{4.11}$$

L1 Loss on magnitude spectrogram masks can be formulated as:

$$L1_{mask} = \frac{1}{N}\sum_{j=0}^{N}|M_j - \hat{M}_j| \tag{4.12}$$

$$\mathcal{L}_{tpsa} = \|\hat{M}_j \odot |X| - T_0^{|X|}(|S_c| \odot cos(\angle S_c - \angle X))\|_1; \tag{4.13}$$

Enric et al. [83] conducted experiments on loss function in the context of MSS and they recommended using loss on signal rather than on mask.[1]

In [14], truncated phase sensitive approximation (tPSA) was used as the objective function, as shown in Equation 4.13. However we discover that in the implementation, loss function is calculated as follows:

$$\mathcal{L}'_{tpsa} = \|\hat{M}_j - \frac{T_0^{|X|}(|S_c| \odot cos(\angle S_c - \angle X))}{|X|}\|_1; \tag{4.14}$$

which means L1 loss was applied between the mask tensor $\hat{M}_j$ and the target mask. Based on the findings from Enric et al. in [83], in order to improve the state-of-the-art HMSS performance, we re-implemented the original loss function from being applied on mask to magnitude STFT using Equation 4.13, results will be shown later in Chapter Evaluation.

Another feature of loss function is that it can serve as a delicate regularization of the model. Huang et al. [39] proposed MSE along with Kullback Leibler (KL) divergence criteria to decrease the similarity between the prediction and the targets of other sources. In [74], Multi-domain loss was proposed, which was calculated before and after the STFT transformation of the estimated signal. In our opinion, there is no actual time representation of the signal within the model so this loss may not guide the model to learn from both domains. For models like Deep Clustering [66, 67] and Cerberus [16], each head of the model use a task-specific loss function such as clustering loss, transcription loss and source reconstruction loss. This setup can train the model in a multi-task manner.

4.4 Experiments

As introduced in the beginning of this Chapter 4, three different experiments were conducted. The first two are Hierarchical Music Source Separation related, either on Slakh2100 or on MS21. The third one is pre-trained four-stem MSS model evaluation on the auto-generated MUSDB18-ish stems on MS21, to show that this dataset

[1] Recommend loss function: $L2_{freq}$, $SI-SDR_{freq}$, $LogL2_{freq}$, $LogL1_{freq}$

is qualified to working on. Specially, in the first experiment, further exploration on **Sequential Mask** was studied, compared to Direct Mask used in the baseline implementation in [14]. In the following sections, we will present the detailed implementation of the experiments.

4.4.1 Hierarchical Music Source Separation using Sequential Mask

To carry out this experiment, we first do the pre-processing step: Calculating the saliency information for all level submixes. The saliency information was computed for every 10-second long frame, with a hop size of 2.5 seconds. Energy (Root Mean Square) level of -30dB is set to be the saliency threshold. Any child source of the submix higher than this threshold will be considered to be salient. This pre-processing step helps to consume less memory space when generating different levels of submix on the fly during training.

During training, different level of submixes of one leaf node target source are computed according to the saliency information for each audio chunk. Then the log magnitude spectrogram would be the input feature of the Mask-Inference model. Then, tPSA 4.13 was used as the objective function during training. A Mask-Inference model is trained to estimate three masks of different levels. This model mainly consists of 4 layers of BLSTM with 600 hidden units for each direction and dropout of 0.3. Subsequently, a fully connected layer was used to generate the masks for different levels of submixes. We train this model for 100 epochs. The detailed parameters is presented in Table 4.

We provide two kinds of mask based method when estimating sources, namely Direct Mask and Sequential Mask. The difference between this two approaches lies in different learning target. For **Direct Mask** training, loss function was calculated between the output of the Mask-Inference model and the divisions between magnitude spectrograms from different level submixes (target source) $|\hat{S}_{l,j}|, l = 1, \ldots, L$

and the magnitude mixture spectrogram $|X|$, which can be formulated as:

$$L1_{DR} = \frac{1}{N}\sum_{j=0}^{N} |\hat{M}_{l,j} - \frac{|S_{l,j}|}{|X|}| \tag{4.15}$$

During evaluation process, sources are calculated by doing multiplication between the estimated masks and mixture magnitude spectrogram directly, as defined in Equation 4.5.

For **Sequential Mask** based approach, loss function was calculated between the output of the Mask-Inference model and the divisions between magnitude spectrograms from multi-level submixes (target source) $|S_{l-1,j}|, l = 2, \ldots, L$ and the magnitude spectrogram of neighbouring higher level $|S_{l,j}|$, which can be formulated as:

$$L1_{SQ} = \frac{1}{N}\sum_{j=0}^{N} |\hat{M}_{l-1,j} - \frac{|S_{l-1,j}|}{|S_{l,j}|}| \tag{4.16}$$

Finally, to construct the estimated source signal in time domain, iSTFT was carried out using the phase of the mixture and the magnitude spectrograms of the estimated sources.

When a model is taught to generate direct masks, it would learn the difference and similarity across all the output sources, which is the **inter-class relation**. However, for a model that can generate a sequential mask, theoretically, it would be oriented to the leaf node source type with **intra-class relation**. **By carrying out this experiment, we aim to uncover the characteristics of Sequential Mask for the first time.**

4.4.2 Hierarchical Music Source Separation on MS21

The second experiment was carried out using direct mask based method on our newly constructed dataset MS21. As mentioned in Section: Hierarchical Structure of Multi-track Dataset, in order to perform hierarchical music source separation in MS21, we created a hierarchy to define the mixing rules of different level submixes.

Parameters	Value
Target Source	Electric Guitar
Sequential Masks	Yes & No
Hidden Unit	600
Drop out	0.3
Epochs	100
Sampling Rate	16000
Audio Channel	Mono
Chunk Length	10s
Chunk Hop Ratio	0.25
Saliency Threshold	-30.0
Window Size	1024
Hop Size	512
Input Feature	Log Magnitude STFT
Batch Size	25
Optimizer	Adam
Learning Rate	0.0001

Table 4: Training parameters for experiment 4.4.1

We follow the same pre-processing procedure using the same parameters. Note that this saliency computation is done before entering the training phase. Besides, it is closedly related to specific dataset split and manually designed hierarchy , which means any modification in the dataset and hierarchy would lead to a re-computation of saliency information. For training, we use the same network structure in the previous experiment without fine-tuning. The detailed information can be found in Table 5. **We conducted this experiment to test the performance of the baseline Mask-Inference model on a different dataset, which contains a real-life auditory scene.**

4.4.3 Pre-trained X-UMX Model Evaluation on Different Test Set

We conducted this experiment to test the compatibility of MS21 to conventional MSS research scenario. We first carried out a series of evaluation experiment of using a pre-trained model X-UMX [73], which was only trained on the training set of MUSDB18 and was capable to estimate 44.1kHz stereo stems in conventional MSS research scenario (*bass*, *drums*, *vocals*, *other*).

Parameters	Value
Target Source	Electric Guitar
Sequential Masks	No
Epochs	100
Sampling Rate	16000
Audio Channel	Mono
Chunk Length	10s
Chunk Hop Ratio	0.25
Saliency Threshold	-30.0
Window Size	1024
Hop Size	512
Batch Size	25
Optimizer	Adam
Learning Rate	0.0001

Table 5: Training parameters for experiment 4.4.2

In terms of the test set, we chose all the available test set in the conventional four-stem setup, namely MS21 (our proposed one), MUSDB18 [9, 28] and MDXDB21 [8]. MS21 test set has a large overlap with MUSDB18 test set, thus **the difference only lies in whether the stems are professional mixed or not**. MDXDB21 contains brand new songs, created by Sony Music Entertainment (Japan) Inc. (SMEJ). Three songs are excluded for demo purposes. Thus, for evaluation experiment, the results of 27 songs were reported in MDX2021 challenge AI crowd platform. Evaluation metric would be introduced in the Chapter 5.

Chapter 5

Evaluation

In the previous chapters we have discuss the method and the configuration of each experiment. In Section 5.1, we would first present the conventional evaluation methods in MSS, highlighting the existing **misunderstanding** of the two definitions of Source-to-Distortion Ratio (SDR). Based on the metric, we then evaluate the performance of sequential masks 4.4.1 in Section 5.2. We also present the results of the HMSS experiment on MS21 4.4.2 in Section 5.3.

5.1 Evaluation Metric

Objective measures in MSS mainly follow the metrics proposed in Blind Source Separation Evaluation (**BSS Eval**) toolkit [84], which decomposes the estimated source $\hat{s_j}$ by orthogonal projections. Let's denote $\langle a, b\rangle$ the inner product between two complex-valued signals a and b. Besides, $\|a\|^2 := \langle a, a\rangle$. We also declare $P_a b := \frac{\langle a,b\rangle}{\langle b,b\rangle} b$ as the orthogonal projection of a onto the subspace spanned by the vector b. Finally we denote $\mathbf{s} := \sum_{j'=1}^{n} s_{j'}$ as the mixture vector consists of all n sources $s_{j'}$ [1], and $\mathbf{n} := \sum_{i=1}^{m} n_i$ as stochastic noise.

Here we present the definition of the source decomposition by orthogonal projections:

[1] Here we assume that sources are mutually orthogonal and the mixture model is additive. Please refer to [84] for the detailed math definition

$$\hat{s_j} := s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{5.1}$$

where

$$s_{target} := P_{s_j}\hat{s_j} = \frac{\langle \hat{s_j}, s_j \rangle}{\langle s_j, s_j \rangle} s_j \tag{5.2}$$

$$e_{interf} := P_{\mathbf{s}}\hat{s_j} - P_{s_j}\hat{s_j} \tag{5.3}$$

$$e_{noise} := P_{\mathbf{n}}\hat{s_j} \tag{5.4}$$

$$e_{artif} := \hat{s_j} - s_{target} - e_{interf} - e_{noise} = \hat{s_j} - P_{\mathbf{s}}\hat{s_j} - P_{\mathbf{n}}\hat{s_j} \tag{5.5}$$

The computation of s_{target} shows that it is the **projection** of the estimate onto the target signal. e_{interf} is the projection of the estimate onto all other sources. e_{noise} is the projection of the estimate onto stochastic noise signal. Based on Equation 5.1 to Equation 5.5, we now present the global performance metrics by computing energy ratios expressed in decibels in [84]. Source-to-Distortion Ratio (SDR) was defined as:

$$SDR := 10\log_{10}\frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} = 10\log_{10}\frac{\|s_{target}\|^2}{\|\hat{s_j} - s_{target}\|^2}; \tag{5.6}$$

and Source-to-Noise Ratio (SNR):

$$SNR := 10\log_{10}\frac{\|s_{target} + e_{interf}\|^2}{\|e_{noise}\|^2}; \tag{5.7}$$

and Source-to-Interference Ratio (SIR):

$$SIR := 10\log_{10}\frac{\|s_{target}\|^2}{\|e_{interf}\|^2}; \tag{5.8}$$

and Source-to-Artifacts Ratio (SAR):

$$SAR := 10\log_{10}\frac{\|s_{target} + e_{interf} + e_{noise}\|^2}{\|e_{artif}\|^2}; \tag{5.9}$$

Although [84] is the most cited paper when it comes to SDR, the most used SDR implemention does not follow the Equation 5.6 in [84]. As pointed out by Rachael Bittner [2], there are two different definitions of SDR in the literature, namely the original definition in [84], and the image version in [85]:

$$SDR(image) := 10\log_{10}\frac{\|s_j\|^2}{\|e_{spat}+e_{interf}+e'_{artif}\|^2} = 10\log_{10}\frac{\|s_j\|^2}{\|\hat{s_j}-s_j\|^2}; \quad (5.10)$$

where e_{spat} is defined as the error between s_{target} (the projection of estimate onto the target) and true target signal s_j:

$$e_{spat} := s_{target} - s_j = P_{s_j}\hat{s_j} - s_j; \quad (5.11)$$

and e'_{artif} is defined as:

$$e'_{artif} := \hat{s_j} - e_{spat} - e_{interf}; \quad (5.12)$$

The common used implementations of BSSEval toolkit in Music Source Separation all implemented the image version in [85] by default, such as the Matlab toolbox `bsseval`[3], the python implementation `museval`[4] and `mir-eval`[5],. Most researcher cited the original version [84] instead, rather than citing the image version [85]. Scale Invariant SDR (SI-SDR) was proposed in [86], addressing the scale sensitive issue of the image version SDR. However, SI-SDR shares the same definition with the original SDR. In this thesis, we will refer SDR only to the Equation 5.10and SI-SDR to the Equation 5.6.

Another issue related to evaluation metric is the **statistics method**. Even though the different paper claim to use the same evaluation metric and the same software package, the reported results may not be comparable. For example, **framewise**

[2] Keynote speeches in Music Information Processing Frontier Final Workshop 2021 and MDX workshop 2021, both were titled as "Source Separation Metrics: What are they measuring?" `https://mip-frontiers.eu/2021/10/11/final-workshop.html#rachel`

[3] `https://gitlab.inria.fr/bass-db/bss_eval`

[4] `https://github.com/sigsep/sigsep-mus-eval`

[5] `https://github.com/craffel/mir_eval`

calculation using different statistics such as **mean** or **median** can cause a noticeable difference. In MDX2021 [8], **Global** SDR was used as the objective metric. It was computed over the full song, which is simple to compute. Besides, it was shown that global SDR correlates with **framewise** median/mean multichannel SDR in BSS Eval v3 and v4 as the Pearson and Spearman correlations are on average larger than 0.9 . **In our thesis, framewise mean SI-SDR was used as the objective metric of HMSS experiment.** For evaluation experiment, global SDR and frame-wise SDR was chose to be the metric.

5.2 Hierarchical Music Source Separation using Sequential Mask

Table 6: Results of Experiment 4.4.1

		All Levels			Level 3			Level 2			Level 1		
Model Type	lvls	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ
SSS(Guitar)	1	-4.6	-2.8	1.8	0.6	3.8	3.2	-6.7	-4.0	2.7	-7.6	-8.3	-0.7
SSS(Guitar)	3	-4.6	-0.8	3.8	0.6	4.0	3.4	-6.7	-2.7	4.0	-7.6	-3.5	4.1
SQ_SSS(Guitar)	3	-4.6	-4.3	0.3	0.6	-2.6	-2.0	-6.7	-4.6	2.1	-7.6	-5.9	**1.7**
SSS(Guitar)$_{stft}$	3	-4.6	0.5	**5.1**	0.6	5.8	**5.2**	-6.7	-1.7	**5.0**	-7.6	-2.7	**5.0**

Here we present the results of the Experiment 4.4.1. This experiment was conducted on Slakh2100 for comparing the performance of Sequential Mask and Direct Mask. As we can see in Table 6, the first row and second row is the reproduction experiment of Single-level Guitar (denoted as SSS(Guitar)-1) and Three-level Guitar (denoted as SSS(Guitar)-3) in [14]. Note that the actual value of noisy Si-SDR is different from the one in the paper but the $\delta SI - SDR$ value is similar so it is regarded as reproducible.

Unlike the above Direct Mask based approach, in the fourth row, we present the results of Sequential Mask based approach (denoted as SQ_SSS(Guitar)). Note that this experiment was using the loss function on mask.

Compared to SSS(Guitar)-3 experiment, the naive implementation of sequential masks cannot outscore multi-level direct mask. The worst performance is in the

highest level, which shows negative improvement (-2dB SI-SDR) over the noisy SI-SDR. Compared to SSS(Guitar)-1 experiment, sequential masks show the better performance in the lowest level (1.7dB) and comparable performance in level 2 (2.1dB).

We provide possible explanations of this result. Sequential masks capture the inner hierarchical information within the multilevel submixes. It cannot utilize the information from other sources in the same level, while multi-level direct masks can do so. Both mask based approach show positive improvement in the lowest level compared to single-level experiment. This indicates that both mask based approaches are **leaf oriented**.

As shown in , different level of Another characteristic of the sequential masks is the **Subtraction Nature** that we discover through visualization and listening. As shown in the upper row of Figure 15, the red rectangle shows that level 3 estimation result from experiment SQ_(Guitar)-3 does not capture the bass line from the Guitar source, while direct mask is capable to. As for level 1, some components are filtered out in sequential masks while direct mask contain more correct information. In other words, the negative performance of sequential masks is not due to the addition of extra artifacts or interference, is due to the fact that the quality of upper level masks can largely affect the quality of lower level masks. Right now sequential mask can still achieve positive improvement on the lowest level source despite of the negative SI-SDR performance on the highest level source. We hypothesize that sequential masks can perform better when integrated with direct masks.

In the last row, we present the improvement experiment (denoted as SSS(Guitar)-3-STFT) by changing the loss function 4.14 to 4.13. We discovered averaged **1.3dB** SI-SDR improvement compared to SSS(Guitar)-3. The biggest boost occurred in the highest level, with 1.8dB SI-SDR increase. This experiment confirm the findings in Enric's work in [83], that loss on signal is better than on mask. However, this finding is specifically for direct mask based approach. As pointed out in the previous paragraph, the lower level estimated signal is highly dependent on the higher level estimated signal, which means **if we only focus on the quality of the recon-**

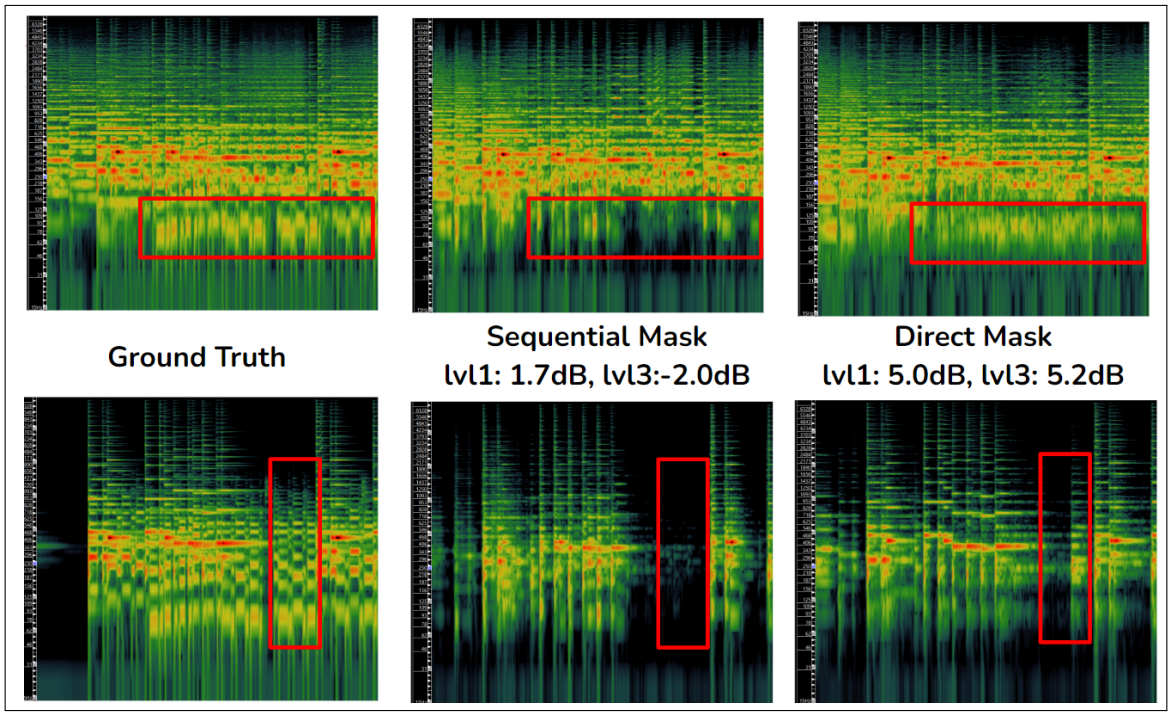

Figure 15: Comparison between Direct Mask and Sequential Mask based approach. The first row shows level 3 source and the second row show level 1 source.

structed signal, lower level sequential masks can be wrongly penalized due to other bad performed upper level masks, even though it perfectly model the neighbouring hierarchical relation. In terms of how to evaluate sequential mask properly while generating high quality estimated results, we leave it as an open question.

5.3 Hierarchical Music Source Separation on MS21

Here we show the result of direct masks based approach experiment on MS21 shown in Table 7, which is denoted as SSS(ElecGTR). The target source is Electric Guitar and it is different from the target source of SSS(Guitar), which is clean guitar. This is due to different hierarchy and instrument category of the dataset. However, we keep the same levels in the hierarchy in both experiments.

Compared to the experiment results on slakh2100, for all levels, SSS(ElecGTR) shows worse performance. One of the reason is the difference of dataset: Slakh2100 does not contain *vocal* source and it is synthesized, while MS21 contains *vocal* source and other professionally performed and recorded content, which represents the com-

plex auditory scene in real life. We hope to point out the challenging situation right now, in applying HMSS on MS21.

Table 7: Results of conducted experiments

		All Levels			Level 3			Level 2			Level 1		
Model Type	lvls	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ
SSS(ElecGTR)	3	-5.7	-4.0	1.7	-1.9	0.8	2.7	-6.7	-5.7	1.0	-8.3	-7.2	1.1
SSS(Guitar)	3	-4.6	-0.8	3.8	0.6	4.0	3.4	-6.7	-2.7	4.0	-7.6	-3.5	4.1

5.4 Pre-trained X-UMX Model Evaluation on Different Test Set

In Table 8, we can compare the evaluation results of a pre-trained X-UMX model [74] (trained on MUSDB18-hq [28]) on different evaluation sets (test set), namely MS21, the newly proposed MDXDB21 [8] and MUSDB18-hq [28]. For each song, global SDR [8] and median frame-wise SDR [6] was calculated. Finally, Median was chose to be the statistics to represent the result of the whole test set.

Table 8: Pre-trained X-UMX Model Evaluation

Test set	Global-SDR					SDR				
	Avg.	Bass	Drums	Vocals	Other	Avg.	Bass	Drums	Vocals	Other
MS21	5.10	**6.36**	5.14	5.46	3.42	5.23	**6.62**	5.22	5.65	3.40
MDXDB21	5.37	5.62	5.81	6.34	3.72	-	-	-	-	-
MUSDB18	**5.85**	4.86	6.70	7.06	4.77	**5.91**	5.12	6.72	7.08	4.73

As we can see, *bass* stem of MS21 test set has the highest performance. We assume the reason is that the automatically generated MUSDB-ish stems of MS21 has a different balance compared to MDXDB2021 and MUSDB18. Loudness normalization algorithm before linear mixing emphasizes *bass* over *drums*, *vocals* and *other*. However, this deviation of performance can be explained as **the pre-trained X-UMX model is not robust to mixing coefficients**. Finally, in terms of averaged performance of the four stems, MS21 is -0.27dB and -0.75dB worse than MUSDB18 and

[6] Window length=30s, hop length=15s.

MDXDB2021 respectively. The evaluation result of MS21 is close to MDXDB21, perhaps it is because of the lack of mastering in MDXDB21. According to [8], "the loudness and tone across different songs were not normalized to any reference level, since these songs are not meant to be distributed on commercial platforms."

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we are first motivated by the fact that the dataset highly restricts the research scenario in Music Source Separation. We then focuses on building **a new multi-track dataset called *MS21***, which can be viewed as an expanded and MSS oriented version of the Mixing Secrets dataset. With total mixture duration of 34.2 hours, MS21 features multi-track content in an unprecedented size and industrial standard. Statistics analysis shows that MS21 contains more averaged number of multi-track files (20±13), a higher percentage of *vocal* stem (84%) compared to MedleyDB and Slakh2100. Besides, MS21 presents a more diverse genre distribution compared to MedleyDB and MUSDB18. MS21 provides MedleyDB like metadata files which capture the specially designed three-level hierarchy annotation for all 500 songs, in order to bridge between different source separation research scenarios. Experiment 4.4.2 indicates that it is challenging to apply HMSS on MS21 as it contains a more complex auditory scene.

To further explore the research possibility in HMSS, we proposed a theory called **Mask Relation** and a new mask based method called **Sequential Mask**. Mathematical definitions were provided. HMSS Experiments showed two features of sequential mask based approach: one is its **Leaf Oriented** feature and the other is

Subtraction Nature. Evaluation experiment 5.4 shows that the averaged SDR performance among the test set of MS21, MDXDB21 and MUSDB18 is within 1dB, which means they share similar auditory scene. The performance degradation of the model on MS21 and MDXDB21 can be explained as that the pre-trained X-UMX model is not robust to mixing coefficient.

For reproducible research, we provide several Github repository for interested readers. For State-of-the-Art, we provide a detailed summary in here. For Dataset Construction, we provide several scripts and jupyter notebook for data downloading, metadata generation and statistics analysis in here. For HMSS experiment using Sequential Mask 4.6 and Direct Mask 4.5, our work is largely based on the open source implementation found in here. For evaluation experiment, our work is based on the X-UMX implementation found in Asteroid, in a branch called ms21.

6.2 Future Works

With MS21, we do think of lots of possible research directions. However, due to limited time span of the master thesis project, we cannot test all these ideas. We introduce them in the following sections.

6.2.1 Bleeding Phenomenon in Music Dataset

During the annotation process of MS21, we found that the widespread existence of bleeding in the dataset. For example, for each single drum instrument such as Hi-Hat and Snare Drum, it is hard to get clean isolated track in live recording set up since each closely located drum instrument tend to be easily recorded by other microphones that are not for it. Semi-automatic method to detect bleeding multi-tracks is desired. However, there is no baseline algorithm for automatic bleeding detection in the literature.

For Music Source Separation, clean data with strong label means clean instrument track with no bleeding from other instrument source. Multi-tracks with bleeding are viewed as weak label data. As pointed out in [8], in professional music recording

practise, stems with bleeding is common. How to deal with large amount of bleeding data and train a bleeding robust MSS system should be a research topic in the future.

6.2.2 Automatic Mixing as a Data Augmentation Method

Data augmentation is an important method to train a DNN based MSS system that can achieve novel performance. In [41], several data augmentation techniques were introduced, such as Random Channel Swapping, Random Gain Scaling by a uniform factor between 0.25 and 1.25, Random Chunking into sequences for each instrument and Random Mixing instruments from different songs. Models trained using data augmentation performed significantly better on test set. In [50], Pitch/Tempo Shift were applied 20% of the time, they randomly change the pitch by -2, -1, 0, , +1, or +2 semitones, and the tempo by a factor taken uniformly between 0.88 and 1.12, using the Soundstretch. Kong et al' [62] applied the *mix-audio* data augmentation used in [87] to augment vocals, accompaniment, drums, and other instruments which randomly mix two 3-second segments from a same source as a new 3-second segment for training. In [53] improve the data augmentation techniques in Demucs v2 by generating 30 seconds long realistic remix from the tracks that are compatible in terms of certain beat and pitch range. 3 semi-tones of shift and 15% of tempo change was set as the boundary.

Given the fact that MS21 contains large amount of multi-tracks per song. We can utilize automatic mixing to generate real remix for a specific source. For example, *backing vocal* stem tend to contain multiple singers singing different harmony content. By changing the mixing scheme of the multi-tracks, we can provide **infinite** *backing vocal* content. Most importantly, they are still **musically related** to the other stems because the multi-track materials are synchronised.

Another motivation is that, for MSS, it is hard to get the professionally mixed and mastered music data. Besides, a fixed mixing coefficients of the sources within the dataset can be a potential bias. Using automatic mixing and mastering to generate different style of mixes during training is one solution to developing **a robust MSS system** that can perform well regardless of the mixing style of the mixture. This

idea can be viewed as an extension of Random Gain Scaling proposed in [41].

6.2.3 Hierarchical Music Source Separation on MUSDB18

As indicated in 2.4, up to now, there is no literature applying HMSS on MUSDB18. The only way to do so is generating accompaniment and its child source together, either using direct masks or sequential mask based approach. We imagine that applying HMSS on MUSDB18 can boost the performance especially for *other* stem, which shows bigger room for improvement.

6.2.4 Complex Mask Relation: Beyond MSS and HMSS?

The key of HMSS lies in the definition of the estimated sources: we force them to be hierarchical related sources and each shares inclusion relation. However, what if the estimated sources do not only contain inclusion relation (parent-child), but also contain independent relation (child-sibling). This problem emerged when one design a system to estimate *accompaniment* and all its child sources all together, where the child sources are independent to each other. To put it another way, this problem is a mixture of conventional MSS and HMSS. This is just one example, many other scenario where the target sources share complex relation to each other may exist, especially outside of the four-stem set up.

As shown in [14], Query-By-Example model aims to separate any instrument according to the hierarchy. This multi-level multi-instrument model is capable to retrieve a hierarchy of the mixture, by stacking the estimated hierarchical sources up in the hierarchy based on the leaf level instrument examples that we feed to the network. This still involves human active listening and sample offering. It would be interesting to develop a hierarchy structure aware system, based on the mixture input and retrieve a whole hierarchy made of separated sources, where more complex relations are found.

List of Figures

List of Tables

Bibliography

[1] Bronkhorst, A. W. The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica* **86**, 117–128 (2000).

[2] Auditory attention—focusing the searchlight on sound. *Current Opinion in Neurobiology* **17**, 437–455 (2007). URL `https://www.sciencedirect.com/science/article/pii/S0959438807000943`. Sensory systems.

[3] O'Sullivan, J. A. *et al.* Attentional selection in a cocktail party environment can be decoded from single-trial eeg. *Cerebral cortex* **25**, 1697–1706 (2015).

[4] Mesgarani, N. & Chang, E. F. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature* **485**, 233–236 (2012).

[5] Downie, J. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology* **29**, 247–255 (2008).

[6] Ono, N., Koldovský, Z., Miyabe, S. & Ito, N. The 2013 signal separation evaluation campaign. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6 (2013).

[7] Liutkus, A. *et al.* The 2016 signal separation evaluation campaign. In *International Conference on Latent Variable Analysis and Signal Separation*, 323–332 (Springer, 2017).

[8] Mitsufuji, Y. *et al.* Music demixing challenge 2021. *Frontiers in Signal Processing* **1** (2022). URL `https://doi.org/10.3389%2Ffrsip.2021.808395`.

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

[10] Miron, M., Orti, J., Bosch, J., Gómez, E. & Janer, J. Score-informed source separation for multichannel orchestral recordings. *Journal of Electrical and Computer Engineering* **2016**, 1–19 (2016).

[11] Miron, M. *Source Separation Methods for Orchestral Music: Timbre-Informed and Score-Informed Strategies.* Ph.D. thesis, Pompeu Fabra University, Barcelona (2018). URL `https://doi.org/10.5281/zenodo.1163675`.

[12] Sarkar, S., Benetos, E. & Sandler, M. Vocal harmony separation using time-domain neural networks. 3515–3519 (2021).

[13] Sarkar, S., Benetos, E. & Sandler, M. Monotimbral ensemble separation using permutation invariant training .

[14] Manilow, E., Wichern, G. & Roux, J. L. Hierarchical musical instrument separation. In *ISMIR* (2020).

[15] Manilow, E., Wichern, G., Seetharaman, P. & Le Roux, J. 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).

[16] Manilow, E., Seetharaman, P. & Pardo, B. Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments (2019). URL `http://arxiv.org/abs/1910.12621`.

[17] Jansson, A., Bittner, R. M., Ewert, S. & Weyde, T. Joint singing voice separation and f0 estimation with deep u-net architectures. In *2019 27th European Signal Processing Conference (EUSIPCO)*, 1–5 (2019).

[18] Heittola, T., Klapuri, A. & Virtanen, T. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In Hirata, K., Tzanetakis, G. & Yoshii, K. (eds.) *ISMIR*, 327–332 (International Society for Music Information Retrieval, 2009). URL http://dblp.uni-trier.de/db/conf/ismir/ismir2009.html#HeittolaKV09.

[19] Bittner, R. *et al.* Medleydb: A multitrack dataset for annotation-intensive mir research. URL http://marl.smusic.nyu.edu/medleydb.

[20] Bittner, R. M., Wilkins, J., Yip, H. & Bello, J. P. Medleydb 2.0 : New data and a system for sustainable data collection. URL http://www.github.com/marl/medleydb.

[21] Gururani, S. & Lerch, A. Mixing secrets : A multi-track dataset for instrument recognition in polyphonic music (2017).

[22] Vincent, E. *et al.* The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges. *Signal Processing* **92**, 1928–1936 (2012). URL https://www.sciencedirect.com/science/article/pii/S0165168411003604. Latent Variable Analysis and Signal Separation.

[23] Araki, S. *et al.* The 2011 signal separation evaluation campaign (sisec2011): - audio source separation -. 414–422 (2012).

[24] Ono, N., Rafii, Z., Kitamura, D., Ito, N. & Liutkus, A. The 2015 signal separation evaluation campaign. In *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation - Volume 9237*, LVA/ICA 2015, 387–395 (Springer-Verlag, Berlin, Heidelberg, 2015). URL https://doi.org/10.1007/978-3-319-22482-4_45.

[25] Stöter, F.-R., Liutkus, A. & Ito, N. The 2018 Signal Separation Evaluation Campaign. In Y., D., S., G., R., M., M., P. & D., W. (eds.) *LVA/ICA: Latent Variable Analysis and Signal Separation*, vol. LNCS, 293–305 (Springer, Surrey, United Kingdom, 2018). URL https://hal-lirmm.ccsd.cnrs.fr/lirmm-01766791.

[26] Vinyes, M. MTG MASS database. `http://www.mtg.upf.edu/static/mass/resources` (2008).

[27] Liutkus, A., Badeau, R. & Richard, G. Gaussian processes for underdetermined source separation. *IEEE Transactions on Signal Processing* **59**, 3155–3167 (2011).

[28] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I. & Bittner, R. Musdb18-hq - an uncompressed version of musdb18 (2019). URL `https://doi.org/10.5281/zenodo.3338373`.

[29] Rafii, Z. *et al.* An overview of lead and accompaniment separation in music (2018).

[30] Hsu, C.-L. & Jang, J.-S. R. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE Transactions on Audio, Speech, and Language Processing* **18**, 310–319 (2010).

[31] Liutkus, A., Fitzgerald, D., Rafii, Z., Pardo, B. & Daudet, L. Kernel additive models for source separation. *IEEE Transactions on Signal Processing* **62**, 4298–4310 (2014).

[32] Chan, T.-S. *et al.* Vocal activity informed singing voice separation with the ikala dataset. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 718–722 (2015).

[33] Miron, M. & Bosch, J. J. PHENICX-Anechoic: note annotations for Aalto anechoic orchestral database (2017). URL `https://doi.org/10.5281/zenodo.840025`.

[34] Rosenzweig, S. *et al.* Dagstuhl choirset (2021). URL `https://doi.org/10.5281/zenodo.4618287`.

[35] Chandna, P., Cuesta, H., Petermann, D. & Gómez, E. A deep-learning based framework for source separation, analysis, and synthesis of choral ensembles. *Frontiers in Signal Processing* **2** (2022). URL `https://www.frontiersin.org/articles/10.3389/frsip.2022.808594`.

[36] Srinivasamurthy, A., Gulati, S., Caro Repetto, R. & Serra, X. Saraga: Open datasets for research on indian art music. *Empirical Musicology Review* **16**, 85–98 (2021).

[37] Miller, N. J. Removal of noise from a voice signal by synthesis (1973).

[38] Rafii, Z. & Pardo, B. A simple music/voice separation method based on the extraction of the repeating musical structure. 221 – 224 (2011).

[39] Huang, P.-S., Kim, M., Hasegawa-Johnson, M. A. & Smaragdis, P. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR* (2014).

[40] Uhlich, S., Giron, F. & Mitsufuji, Y. Deep neural network based instrument extraction from music. vol. 2015-August, 2135–2139 (Institute of Electrical and Electronics Engineers Inc., 2015).

[41] Uhlich, S. *et al.* Improving music source separation based on deep neural networks through data augmentation and network blending. 261–265 (Institute of Electrical and Electronics Engineers Inc., 2017).

[42] Chandna, P., Miron, M., Janer, J. & Gómez, E. Monoaural audio source separation using deep convolutional neural networks. vol. 10169, 258–266 (2017).

[43] Jansson, A. *et al.* Singing voice separation with deep u-net convolutional networks (2017).

[44] Takahashi, N., Goswami, N. & Mitsufuji, Y. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation (2018). URL `http://arxiv.org/abs/1805.02410`.

[45] Takahashi, N. & Mitsufuji, Y. Multi-scale multi-band densenets for audio source separation (2017). URL `http://arxiv.org/abs/1706.09588`.

[46] Takahashi, N. & Mitsufuji, Y. D3net: Densely connected multidilated densenet for music source separation. *ArXiv* **abs/2010.01733** (2020).

[47] Luo, Y., Chen, Z. & Mesgarani, N. Speaker-independent speech separation with deep attractor network. *IEEE/ACM Transactions on Audio Speech and Language Processing* **26**, 787–796 (2018).

[48] Stoller, D., Ewert, S. & Dixon, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *CoRR* **abs/1806.03185** (2018). URL `http://arxiv.org/abs/1806.03185`. `1806.03185`.

[49] Lluís, F., Pons, J. & Serra, X. End-to-end music source separation: Is it possible in the waveform domain? In *INTERSPEECH*, 4619–4623 (2019). URL `https://doi.org/10.21437/Interspeech.2019-1177`.

[50] Défossez, A., Usunier, N., Bottou, L. & Bach, F. Music source separation in the waveform domain (2019). URL `http://arxiv.org/abs/1911.13254`.

[51] Choi, W., Kim, M., Chung, J., Lee, D. & Jung, S. Investigating u-nets with various intermediate blocks for spectrogram-based singing voice separation (2019). URL `http://arxiv.org/abs/1912.02591`.

[52] Choi, W., Kim, M., Chung, J. & Jung, S. Lasaft: Latent source attentive frequency transformation for conditioned source separation (2020). URL `http://arxiv.org/abs/2010.11631`.

[53] Défossez, A. Hybrid spectrogram and waveform source separation (2021). URL `https://arxiv.org/abs/2111.03600`.

[54] Choi, W. *Deep Learning-based Latent Source Analysis for Source-aware Audio Manipulation.* Ph.D. thesis, PhD thesis]. Korea University (2021).

[55] Kim, M., Choi, W., Chung, J., Lee, D. & Jung, S. Kuielab-mdx-net: A two-stream neural network for music demixing.

[56] Miron, M., Janer, J. & Gómez, E. Monaural score-informed source separation for classical music using convolutional neural networks. In *ISMIR* (2017).

[57] Slizovskaia, O., Haro, G. & Gómez, E. Conditioned source separation for music instrument performances (2020). URL http://arxiv.org/abs/2004.03873http://dx.doi.org/10.1109/TASLP.2021.3082331.

[58] Lee, J. H., Choi, H.-S. & Lee, K. Audio query-based music source separation (2019). URL http://arxiv.org/abs/1908.06593.

[59] Meseguer-Brocal, G. & Peeters, G. CONDITIONED-U-NET: INTRODUCING A CONTROL MECHANISM IN THE U-NET FOR MULTIPLE SOURCE SEPARATIONS. In *Proceedings of the 20th International Society for Music Information Retrieval Conference* (Delft, Netherlands, 2019). URL https://hal.archives-ouvertes.fr/hal-02448917.

[60] Cantisani, G., Essid, S. & Richard, G. Neuro-steered music source separation with eeg-based auditory attention decoding and contrastive-nmf. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 36–40 (2021).

[61] Cantisani, G., Trégoat, G., Essid, S. & Richard, G. MAD-EEG: an EEG dataset for decoding auditory attention to a target instrument in polyphonic music. In *Speech, Music and Mind (SMM), Satellite Workshop of Interspeech 2019* (Vienna, Austria, 2019). URL https://hal.archives-ouvertes.fr/hal-02291882.

[62] Kong, Q., Cao, Y., Liu, H., Choi, K. & Wang, Y. Decoupling magnitude and phase estimation with deep resunet for music source separation (2021). URL http://arxiv.org/abs/2109.05418.

[63] Yin, D., Luo, C., Xiong, Z. & Zeng, W. Phasen: A phase-and-harmonics-aware speech enhancement network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 9458–9465 (2020).

[64] Yang, y.-h. On sparse and low-rank matrix decomposition for singing voice separation. 757–760 (2012).

[65] Sprechmann, P., Bronstein, A. & Sapiro, G. Real-time online singing voice separation from monaural recordings using robust low-rank modeling. *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012* (2012).

[66] Luo, Y., Chen, Z., Hershey, J., Le Roux, J. & Mesgarani, N. Deep clustering and conventional networks for music separation: Stronger together. vol. 2017, 61–65 (2017).

[67] Hershey, J. R., Chen, Z., Roux, J. L. & Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. vol. 2016-May, 31–35 (Institute of Electrical and Electronics Engineers Inc., 2016).

[68] St
"oter, F.-R., Uhlich, S., Liutkus, A. & Mitsufuji, Y. Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software* (2019). URL `https://doi.org/10.21105/joss.01667`.

[69] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *CoRR* **abs/1512.03385** (2015). URL `http://arxiv.org/abs/1512.03385`. `1512.03385`.

[70] Krizhevsky, A., Sutskever, I. & Hinton, G. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems* **25** (2012).

[71] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. *ArXiv* **abs/1505.04597** (2015).

[72] Vaswani, A. *et al.* Attention is all you need. *CoRR* **abs/1706.03762** (2017). URL `http://arxiv.org/abs/1706.03762`. `1706.03762`.

[73] Takahashi, N., Singh, M. K. & Mitsufuji, Y. Hierarchical disentangled representation learning for singing voice conversion (2021). URL `http://arxiv.org/abs/2101.06842`.

[74] Sawata, R., Uhlich, S., Takahashi, S. & Mitsufuji, Y. All for one and one for all: Improving music separation by bridging networks. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 51–55 (2021).

[75] Yu, C.-Y. & Cheuk, K. W. Danna-sep: Unite to separate them all. *ArXiv* **abs/2112.03752** (2021).

[76] Hennequin, R., Khlif, A., Voituret, F. & Moussallam, M. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software* **5**, 2154 (2020).

[77] Prétet, L., Hennequin, R., Royo-Letelier, J. & Vaglio, A. Singing voice separation: A study on training data. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 506–510 (2019).

[78] Hung, Y.-N. & Lerch, A. Multitask learning for instrument activation aware music source separation. *ArXiv* **abs/2008.00616** (2020).

[79] wichern, g., wishnick, a., lukin, a. & robertson, h. comparison of loudness features for automatic level adjustment in mixing. *journal of the audio engineering society* (2015).

[80] Flexer, A. A closer look on artist filters for musical genre classification. 341–344 (2007).

[81] Prétet, L., Hennequin, R., Royo-Letelier, J. & Vaglio, A. Singing voice separation: A study on training data. In *ICASSP 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, 506–510 (IEEE, 2019).

[82] Huang, G., Liu, Z. & Weinberger, K. Q. Densely connected convolutional networks. *CoRR* **abs/1608.06993** (2016). URL `http://arxiv.org/abs/1608.06993`. `1608.06993`.

[83] Gusó, E., Pons, J., Pascual, S. & Serrà, J. On loss functions and evaluation metrics for music source separation. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 306–310 (2022).

[84] Vincent, E., Gribonval, R. & Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing* **14**, 1462–1469 (2006).

[85] Vincent, E., Sawada, H., Bofill, P., Makino, S. & Rosca, J. P. First stereo audio source separation evaluation campaign: Data, algorithms and results. In *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation*, ICA'07, 552–559 (Springer-Verlag, Berlin, Heidelberg, 2007).

[86] Roux, J. L., Wisdom, S., Erdogan, H. & Hershey, J. R. SDR - half-baked or well done? *CoRR* **abs/1811.02508** (2018). URL `http://arxiv.org/abs/1811.02508`. `1811.02508`.

[87] Song, X., Kong, Q., Du, X. & Wang, Y. Catnet: music source separation system with mix-audio augmentation. *CoRR* **abs/2102.09966** (2021). URL `https://arxiv.org/abs/2102.09966`. `2102.09966`.

Appendix A

Tables of Dataset Statistics

Sub Genre	Other
'Bollywood', 'Musical Theatre', 'Intricate Solo Acoustic Guitar', 'Indie Soul', 'Downtempo Neo-Soul', 'Blues', 'Cinematic Soundtrack', 'Guitar/Sax Acoustic Duo', 'Atmospheric String Textures', 'Thai Headbanging', 'Leftfield Indie', 'Bollywood Hindi Song', 'Experimental Post-rock', 'Traditional Boogie Woogie', 'Atmospheric Alternative', 'Dark Break-based Dance', 'Dramatic Short Story', 'A Capella Duo', 'Afrobeat', 'Cumbia'	Other

Table 9: Sub Genres that are labeled as Other

Track Name	Broad Genre	Sub Genre
Andres Guazzelli - Attention	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	Various Styles
Andres Guazzelli - Flores De Abril	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	-
BaumXmedia - Dream State (feat. Flora Lin)	Pop / Singer-Songwriter	-
BaumXmedia - Koishii (feat. N.I.A.)	Pop / Singer-Songwriter	-
Digital Humans - Electrvm	Pop / Singer-Songwriter	-
Digital Humans - Relentlessly	Pop / Singer-Songwriter	-
Forkupines - Semantics	Rock / Punk / Metal	-
Forkupines - Sleep By The Fire Bloom In Water	Rock / Punk / Metal	-
Forkupines - Sugar - Faith	Rock / Punk / Metal	-
Hazael - A Mi Lado	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	-
Hazael - Resistencia Para Un Nuevo Comenzar	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	-
Spektakulatius - Christmas Blues	Acoustic / Jazz / Country / Orchestral	-
Spektakulatius - Is You Is Or Is You Ain't	Acoustic / Jazz / Country / Orchestral	-
Spektakulatius - Jeden Winter	Acoustic / Jazz / Country / Orchestral	-
Spektakulatius - Our Love Is Here To Stay	Acoustic / Jazz / Country / Orchestral	-
Spektakulatius - Wayfaring Stranger	Acoustic / Jazz / Country / Orchestral	-
Spektakulatius - What Child Is This	Acoustic / Jazz / Country / Orchestral	-
Street Noise - I'd Rather Be Drinkin	Rock / Punk / Metal	-
Street Noise - Revelations	Rock / Punk / Metal	-
Street Noise - You Are The One	Rock / Punk / Metal	-
Tommy Marcinek - Happy Blues	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	-
Tommy Marcinek - My Childhood Sweetheart	Alt Rock / Blues / Country Rock / Indie / Funk / Reggae	-
Triviul - Alright	Pop / Singer-Songwriter	-
Triviul - Angelsaint	Pop / Singer-Songwriter	-
Triviul - Better	Pop / Singer-Songwriter	-
Triviul - Dorothy	Pop / Singer-Songwriter	-
Triviul - Gimme	Pop / Singer-Songwriter	-
Triviul - To Sam Rawfers	Pop / Singer-Songwriter	-
Triviul - Widow (feat. The Fiend)	Pop / Singer-Songwriter	-

Table 10: Tracks with Sub Genres as "Various Styles"