Sapienza
Università di Roma

Harmonizing Deep Learning: A Journey through the Innovations in Signal Processing, Source Separation and Music Generation

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Dottorato di Ricerca in Informatica – XXXV Ciclo

Candidate
Michele Mancusi
ID number 1603546

Thesis Advisor
Prof. Emanuele Rodolà

2023/2024

Thesis defended on 26 January 2024
in front of a Board of Examiners composed by:

Prof. Lamberto Ballan (chairman)
Prof. Giovanni Petri
Prof. Alessandro Raganato

Harmonizing Deep Learning: A Journey through the Innovations in Signal Processing, Source Separation and Music Generation
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: mancusi@di.uniroma1.it

To my Father

Abstract

In this thesis, the profound intersection of deep learning and audio processing is explored, highlighting the transformative potential of these techniques in deciphering and manipulating audio signals. From the intricacies of marine ecosystems to the nuances of music, the application of deep learning has shown considerable promise in reshaping our understanding of sound. We commence by delving into deep extractors for audio source separation, showcasing their potency in tasks ranging from isolating marine sounds to identifying singing voices in music tracks. This journey emphasizes the role of neural networks in extracting and interpreting sounds from a complex mixture of signals, taking us to the world of autoregressive models, where we investigate their principles and applications in source separation, emphasizing unsupervised methods. Much of the research dwells on the innovative Bayesian approach with autoregressive models for signal source separation, demonstrating its efficiency across auditory and visual domains and the intriguing application of diffusion models for music generation and separation, accentuating their versatility in audio tasks. While the technical profundities form the core of the research, its broader implications shed light on the transformative potential of deep learning in myriad domains. From music production and music information retrieval to environmental surveillance, the adaptability of deep learning techniques promises a future replete with sophisticated audio processing tools. Conclusively, this thesis stands as a testament to the power of deep learning in enhancing, understanding, and enriching the world of sound, paving the way for further advancements in this captivating realm.

List of Publications

Mancusi, M., Postolache, E., Mariani, G., Santilli, A., Cosmo, L., & Rodolà, E. (2023). Latent Autoregressive Source Separation. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(8), 9444-9452.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodola. 2023. Accelerating Transformer Inference for Translation via Parallel Decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12336–12355, Toronto, Canada. Association for Computational Linguistics.

M. Mancusi, N. Zonca, E. Rodolà and S. Zuffi, "Towards the evaluation of marine acoustic biodiversity through data-driven audio source separation, "*2023 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, Bologna, Italy, 2023, pp. 1-10, doi: 10.1109/I3DA57090.2023.10289193.

M. Mancusi et al., "Exploiting Music Source Separation For Singing Voice Detection, "*2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, Rome, Italy, 2023, pp. 1-6, doi: 10.1109/MLSP55844.2023.10285863.

Mancusi, M., Mariani, G., Tallini, I., Postolache, E., Cosmo, L., & Rodolà, E. (2023). Multi-source diffusion models for simultaneous music generation and separation. *arXiv preprint arXiv:2302.02257.*

Mancusi, M., Postolache, E., Mariani, G., Fumero, M., Santilli, A., Cosmo, L., & Rodolà, E. (2021). Unsupervised source separation via Bayesian inference in the latent domain. *arXiv preprint arXiv:2110.05313.*

Contents

1 Introduction **1**

1.1 The Analog and Digital Representation of the Sound 2

1.2 Signal Processing 3

1.3 Contributions 4

2 Deep Extractors for Audio Sources **7**

2.1 Towards the evaluation of marine acoustic biodiversity through data-driven audio source separation 8

2.1.1 Introduction 9

2.1.2 Related Works 11

2.1.3 Method 12

2.1.4 Experiments 16

2.1.5 Conclusion 22

2.2 Exploiting music source separation for singing voice detection 23

2.2.1 Introduction 23

2.2.2 Related Works 23

2.2.3 Method 24

2.2.4 Experiments 26

2.2.5 Conclusion 29

2.3 Chapter Conclusions 30

3 Autoregressive Models for Source Separation **31**

3.1 Unsupervised source separation via bayesian inference in the latent domain 32

3.1.1 Introduction 33

3.1.2 Related Works 33

3.1.3 Method 34

3.1.4 Experiments 40

3.1.5 Conclusion 41

3.2 Latent Autoregressive Source Separation 42

3.2.1 Introduction 42

3.2.2 Related Works 43

3.2.3 Method 44

3.2.4 Experiments 49

3.2.5 Conclusion 54

3.3 Chapter Conclusions 54

4 Diffusion Models for Music Sources **55**
4.1 Multi-Source Diffusion Models for Simultaneous Music Generation and Separation 56
4.1.1 Introduction 56
4.1.2 Related Works 58
4.1.3 Method 59
4.1.4 Experiments 63
4.1.5 Conclusion 67
4.2 Chapter Conclusions 68

5 Conclusions **69**

Bibliography **71**

Chapter 1

Introduction

Since the dawn of civilization, sound and music have been a fundamental component of the human condition, contributing significantly to the definition of our species and society's shaping. The ability to perceive and interpret sounds has played a crucial role in survival and has provided humans with a unique means of expression, communication, and building the social fabric.

The primordial importance of sound lies primarily in its function as a means of perceiving the world around us. In a natural, ancestral context, the ability to interpret the sounds of one's surroundings could determine the difference between life and death: the rustling of leaves caused by an approaching predator, the gurgling of water from a nearby stream, or the alarm cry of a conspecific were all vital pieces of information conveyed through audio.

However, audio also took on a profoundly emotional and social dimension in addition to its pragmatic function. With its infinite nuances and tones, the human voice became the main instrument of communication between individuals, enabling the exchange of information and the transmission of emotions, intentions, and desires.

Parallel to this, music emerged as one of the earliest and most profound forms of human cultural expression. Although the precise origins of music are shrouded in mystery, ancient musical instruments and archaeological findings indicate that the production of melodies and rhythms has accompanied humankind for thousands of years. With its ability to evoke emotions, tell stories, and build communities, music has permeated every culture and civilization, testifying to its universality and fundamental relevance.

Music has always occupied a special place in the realm of human experience, serving not only as a means of artistic expression but also as a means of connecting with the divine and the unknowable. Throughout the millennia, music has functioned as a bridge between the physical and spiritual worlds, a means to achieve elevated states of consciousness and to bring the individual closer to the sacred.

Music is recognized as a vehicle for enlightenment and a means to approach the divine in many cultures and spiritual traditions. Mantras in Buddhism and Hinduism, sacred chants in Christian and Islamic traditions, and ceremonial rhythms in indigenous cultures are manifestations of the profound interaction between music and spirituality.

One of the most remarkable qualities of music is its universality. While words can divide and create barriers due to language differences, music transcends these limitations. It speaks directly to the soul, evoking emotions and states of mind that are fundamentally human. This universality has made music a powerful tool for mystical experiences, as it can connect individuals from different cultures and traditions in spiritual communion.

In many traditions, music is seen as a representation of the cosmos: 'musica mundana' or 'music of the spheres' suggests that the entire universe is ordered according to harmonic principles, with each planet and star emitting its own 'melody' in a grand cosmic concert. This view recognizes music as a manifestation of the divine, a reflection of the inherent order and beauty in the cosmos. Music, similar to language, manifests our need to find order in chaos, to give meaning and structure to the sound world around us. Moreover, the role of music as a catalyst for collective identities cannot be denied. From the ritual dances of tribal societies to the great mass concerts of the 21st century, music could always unite people under a single melody, creating a sense of belonging and community.

Music, in its many forms and functions, is an essential component of our biology and survival and a profound manifestation of our need for connection, expression, and understanding. It sheds light on the richness and complexity of the human condition, offering a window into the past, present, and perhaps even the future of humanity.

1.1 The Analog and Digital Representation of the Sound

Sound, a phenomenon that has captivated and intrigued humans for millennia, is a change or fluctuation in air pressure over time. Delving deeper into its properties, sound frequencies audible to the human ear typically lie within 20 Hz to 20 kHz. This range is quite remarkable, given the vast spectrum of frequencies in the natural world, and our ears have evolved to capture those particular sounds that are most pertinent for our survival and communication. However, to harness and manipulate this auditory medium, technology had to develop ways to capture, store, and reproduce these air pressure variations. At the heart of many traditional recording devices is a fascinating mechanism that can transform the kinetic energy of sound waves (pressure changes) into electrical energy. One such mechanism involves using an induction coil positioned within a magnetic field. As sound waves interact with a diaphragm, the induction coil moves within the magnetic field, producing a corresponding electrical signal. Conversely, the process is reversed when the goal is to reproduce the captured sound. Loudspeakers, a common endpoint in our audio reproduction chain, convert electrical signals back into air pressure fluctuations, thus re-creating the sound that was initially captured. These speakers rely on electrical currents to move a diaphragm, which creates pressure waves in the surrounding air – the sound we hear. Sound is often represented mathematically as a function $y(t)$. Here, t denotes time, which provides a reference frame for these pressure variations. Meanwhile, y could represent many measurable quantities, such as pressure or the electrical tension produced in the abovementioned conversion process in recording devices. Computers have revolutionized how we process and interact with sound in today's digitized era. Unlike analog systems, which continuously represent sound waves, computers rely on discrete data points. This means that the continuous function y(t) has to be converted, or 'digitized,' into a series of distinct values. These values can be stored using various encoding mechanisms. Some prevalent encodings in the audio industry are the 32-bit IEEE 754 floating-point representation and the 16-bit linear coding, especially notable in the widespread WAV file format. An essential concept for digitizing sound is the Nyquist-Shannon sampling theorem, a pivotal piece of understanding proposed by Claude Shannon in 1949 [158]. According to this theorem, to reproduce a sound faithfully without losing any of its frequency content, one needs to sample it at a rate at least twice its highest frequency. Given the human auditory range, a sampling rate of 40 kHz would technically suffice. However, to allow a margin for error and to accommodate filters, the industry settled on a slightly higher rate of 44.1 kHz. This rate has become the standard for audio CDs and other digital audio platforms. Sound, in all its magnificent intricacy, serves as a bridge between our physical reality and our perception. From its physical properties to the myriad ways we have developed to capture, analyze, and reproduce it, sound remains a testament to nature's complexity and human ingenuity.

1.2 Signal Processing

As we have seen, in its very essence, the audio signal captures variations in air pressure over time. Analyzing such signals has always been a fascinating realm in signal processing, and several transformation techniques have been developed over the years to study these signals in both time and frequency domains. One of the cornerstones in this domain is the Discrete Fourier Transform (DFT). The DFT provides a mechanism to analyze finite segments or "small chunks" of an audio signal, allowing for a representation in the frequency domain [165]. Digital audio signals are inherently discrete, representing amplitude values at regular intervals of time. These values are sampled from continuous audio signals. The DFT transforms this time-domain representation (waveform) into a frequency-domain representation (spectrum). For a discrete sequence $x[n]$, where $n = 0, 1, ..., N - 1$ is the time index and N is the number of samples, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j(2\pi/N)\cdot k\cdot n}$$

where $X[k]$ is the DFT result at frequency index k and k ranges from 0 to $N - 1$, representing the frequency bins. DFT provides insights into the frequency components of audio signals, allowing for various applications like filtering, compression, and feature extraction. When these segments overlap, a more comprehensive representation emerges, known as the Short-Time Fourier Transform (STFT).
The STFT is pivotal as it captures how frequency components evolve over time. This is particularly significant because most real-world signals, including audio, are non-stationary, implying that their frequency pattern changes as time progresses [2]. The primary motivation behind the STFT is to obtain a time-frequency representation of a signal. This means we can view both how the frequencies in a signal change over time and which frequencies are dominant at a given time.

$$X(t, \omega) = \int_{-\infty}^{\infty} x(\tau)w(t - \tau)e^{-j\omega\tau}d\tau \tag{1.1}$$

Where $X(t, \omega)$ is the STFT of $x(t)$, $w(t-\tau)$ is a window function which is nonzero for only a short period of time. ω is the angular frequency (in rad/s). For reconstruction purposes, the Inverse Short-Time Fourier Transform (ISTFT) can be employed, allowing the audio signal to be reconstructed from its time-frequency representation. While the STFT offers a robust method for time-frequency analysis, other transforms have been developed. The wavelet transform emerged as an alternative, introducing a novel approach that provides varying time resolutions depending on the frequency. This method is particularly beneficial for capturing rapid changes at higher frequencies and slower variations at lower frequencies [107]. Building on the principles of the wavelet transform, the scattering transform was introduced to address some of the challenges that traditional transforms could not effectively handle [4].
In the realm of speech recognition, accuracy and precision are crucial. The mel-spectrogram, a specialized form of the STFT, focuses on the perceptual properties

of human hearing. Warping the frequency scale emphasizes the regions most critical to human auditory perception [113, 20]. Inverting this representation back to the audio domain is inherently complex. However, advancements in deep learning have offered methodologies to achieve this inversion with considerable accuracy [160, 133]. Rainbowgrams, a relatively recent addition to the array of time-frequency representations, present a dual depiction of magnitude and phase information, offering a richer insight into the audio signal's characteristics [37].
Underpinning the development and adoption of the STFT is a foundational belief: audio signals, particularly musical ones, are fundamentally composed of stationary, periodic functions. However, the world of sound is vast, and not all audio can be perfectly encapsulated by periodic functions. Stochastic textures in sound, like the white noise, defy this assumption. Addressing these textures requires more sophisticated systems, leading researchers to explore and develop more intricate signal processing mechanisms [183].

1.3 Contributions

In Chapter 2, we present the paradigm of deep extractors in the realm of audio signal processing. Section 2.1 dives into the application of audio source separation in marine biodiversity assessment. The significance of monitoring underwater ecosystems and the challenges therein are addressed. We delve into the potential of Passive Acoustics Monitoring (PAM) as a reliable tool for such tasks. The outcomes reveal the efficacy of this methodology and its potential implications for a more profound understanding of aquatic ecosystems. Section 2.2 focuses on a specific sub-discipline within music information retrieval: Singing Voice Detection (SVD). The narrative introduces a ground-breaking system that harnesses the prowess of Demucs, a cutting-edge music source separator, alongside two sophisticated neural architectures – LRCN and Transformer network. These combinations shed light on the underlying potential of deep learning to enhance SVD tasks.

In Chapter 3, we will delve deep into the principles and applications of autoregressive models in the domain of source separation. Firstly, in Section 3.1, the challenges of audio source separation in an unsupervised setting will be addressed. By relying on deep Bayesian priors, our proposed method will be demonstrated to offer competitive performance when juxtaposed against supervised techniques while being more resource-efficient compared to other unsupervised counterparts. Subsequently, Section 3.2 will explore the Latent Autoregressive Source Separation (LASS) concept. This innovative approach, rooted in the principles of vector quantization, seeks to offer a solution to the complexities of adapting pre-trained models for novel tasks. LASS emphasizes a Bayesian model with autoregressive priors to execute source separation efficiently. The effectiveness of this method will be evaluated across both auditory and visual domains, highlighting its adaptability and efficiency.

In Chapter 4, we delved into the intricate relationship between diffusion models and deep learning and their influence on music. We explored the essence of diffusion models that operate on the deteriorating and restoring data principle, leveraging denoising concepts. The versatility of diffusion models in deep learning applications was showcased across disciplines ranging from computer vision and natural language processing to bioinformatics. A particular focus was cast on the pioneering application of multi-source diffusion models in simultaneous music generation and separation.

This novel approach stands out for its dual capacity: music generation and separation. We introduced innovative inference strategies highlighting the model's generative prowess and ability to isolate sources.

Chapter 2

Deep Extractors for Audio Sources

In this chapter, we will discuss the use of particular neural networks called deep extractors, particularly in two tasks: the isolation of sounds produced by fish from the marine background and the identification of the singing voice in an audio signal, particularly in music tracks. The works we will discuss use networks created to isolate and extract sounds from a mixture of signals.

Music source separation, a sub-discipline within the broader field of audio processing, focuses on separating the individual components or sources within a mixed musical signal. This separation is crucial for various applications, from remixing songs to enhancing the clarity of individual instruments. To achieve this, the separation methods primarily focus on the time-frequency representations of music, wherein both the temporal and frequency components of the audio signals are considered.

The core idea behind these methods is to predict a power spectrogram for each source, representing the intensity of frequencies over time. Once these predictions are made, the phase from the original or input mixture is reused to reconstruct the separated sources. The phase carries the essential information about the temporal structure of the signal, which, when combined with the predicted power spectrogram, provides a holistic view of the separated source.

Among traditional methods for the separation of music sources, the non-negative matrix factorization (NMF) method was among the earliest and most influential techniques. Introduced by [164], NMF breaks down the spectrogram of the input mixture into a set of base spectra and their corresponding activations, thereby facilitating separation. Following closely was the independent component analysis (ICA) by [63], a statistical method designed to transform the observed mixed signals into a set of statistically independent components. Then, there is the HMM-based prediction over power spectrograms presented by Roweis et al. [147]. This technique applies Hidden Markov Models to predict the progression of the sources over time-based on the power spectrograms. Last in this lineage of traditional methods are the segmentation techniques introduced by Bach and Jordan et al. [5]. These focus on segmenting the audio into distinct chunks, each dominated by a particular source.

However, the landscape of music source separation underwent a seismic shift with the advent of deep learning. Initially, the supervised methods were heavily influenced by their applications in speech source separation, as evidenced by the work of Grais [47]. However, soon after, the focus shifted towards music, leading to the development of various architectures. For instance, [185], pioneered the use of simple neural networks for this task. They later extended their work in 2017 [187] to incorporate Long Short-Term Memory networks (LSTMs), which are adept at handling sequences, making them especially relevant for time series data like audio. Furthermore, multi-scale networks became a topic of interest, with significant contributions from [96] and [180].

An exciting development in this deep learning era was using Wiener filtering as a post-processing step. Introduced by [120], this method helps refine the deep learning models' output, enhancing the separated sources' clarity and fidelity. In 2018, the MMDenseLSTM model by Takahashi et al. in 2018 [178], set new performance records. Stöter et al. [173] also introduced a baseline model named Open Unmix, which has been instrumental in standardizing evaluations in this domain.

While deep learning brought forth many new models, it is imperative to note that not all showed superior performance. For instance, models operating directly in the waveform domain, such as Wave-U-Net [67] and a model inspired by Wavenet [142], had performance metrics that were less than ideal when compared to other approaches.

In monophonic speech source separation, spectrogram masking methods have shown remarkable efficacy, as highlighted by the works of Kolbæk et al. in 2017 [79] and Isik et al. in 2016 [66]. Nevertheless, the landscape evolved further when Luo and Mesgarani [104] made significant strides in improving waveform domain methods: ConvTasnet emerged as a dominant force among the newer models, albeit with some artefacts. However, the Demucs architecture [34] has been then introduced trying to address the artefact issue, setting the stage for the next wave of advancements in the field.

In the following sections, we will also speak about Demucs and Conv-Tasnet, two top-tier models for source separation that showed incredible performance in the domain of speech and music source separation, and we will see how these models can be applied to other domains or be useful for different tasks.

2.1 Towards the evaluation of marine acoustic biodiversity through data-driven audio source separation

The marine ecosystem faces alarming changes, including biodiversity loss and the migration of tropical species to temperate regions. Monitoring underwater environments and their inhabitants is crucial, but challenging in vast and uncontrolled areas like oceans. Passive acoustics monitoring (PAM) has emerged as an effective method, using hydrophones to capture underwater sound. Soundscapes with rich sound spectra indicate high biodiversity, soniferous fish vocalizations can be detected to identify specific species. Our focus is on sound separation within underwater soundscapes, isolating fish vocalizations from background noise for accurate biodiversity assessment. To address the lack of suitable datasets, we collected fish vocalizations from online repositories and captured sea soundscapes at various locations. We propose an online generation of synthetic soundscapes to train two popular sound separation networks. Our study includes comprehensive evaluations on a synthetic test set, showing that these separation models can be effectively applied in our settings, yielding encouraging results. Qualitative results on real data showcase the model's generalization ability. Utilizing sound separation networks enables automatic extraction of fish vocalizations from PAM recordings, enhancing biodiversity monitoring and capturing animal sounds in their natural habitats.

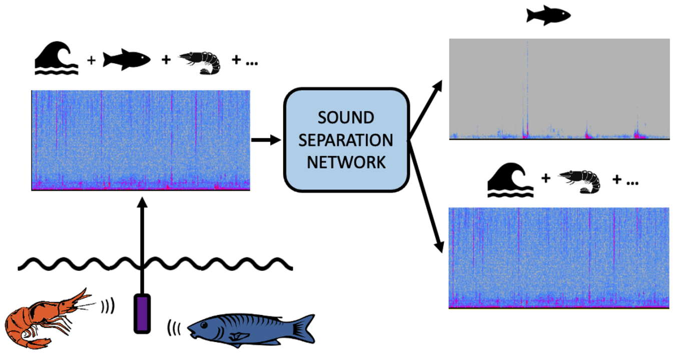


Figure 2.1. We consider the problem of separating the sound produced by fishes from the background sound of the sea.

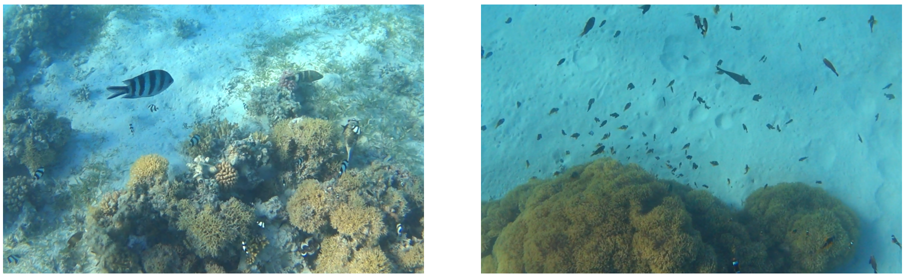


Figure 2.2. Frames from the video captured at Marsa Alam showing the presence of soniferous fish.

2.1.1 Introduction

The oceans cover 71% of the Earth's surface and represent the natural habitat of numerous marine species. The biodiversity present in these environments is impressive, and tracking the activity and quantity of all existing species is essential for monitoring the whole marine ecosystem. In fact, today more than ever, the environmental issue is of crucial importance, and the oceans, like the whole planet Earth, are facing drastic and dramatic changes due to human activity, among which overfishing and ocean warming [39]. These changes, in addition to damaging the marine ecosystem, mainly affect the species that inhabit the sea: monitoring biodiversity is of vital importance, to understand the trend of the abundance of marine fauna, identify the most vulnerable areas, and take action to safeguard endangered species [122]. However, monitoring marine animals is challenging because many of the methods used on the Earth's surface for tracking, such as photos and videos, are often ineffective in the marine environment, due to the limited accessibility to many areas and poor light and visibility conditions. Furthermore, a significant amount of data that can be collected on physical quantities, such as temperature, salinity and pressure, may not reflect the biodiversity in a specific location. Therefore, there is a need for tools capable of overcoming these challenges and providing an accurate assessment of the marine habitat biodiversity. Underwater, instead of relying on optical signals, sound can be used to monitor biodiversity [6]. Indeed, the acoustic environment faithfully reflects the traits of the fauna present in a specific location and its behavior [106][117]. One of the most popular and effective methods for monitoring marine biodiversity is passive acoustics monitoring (PAM), which employs hydrophones to capture underwater sound. Many aquatic organisms produce species-specific sounds, and modern technologies are becoming more and more convenient and precise, allowing for very accurate and careful data acquisition. Acoustic indices were initially used to assess biodiversity from PAM recordings [176][117]. These indices are used to estimate richness, amplitude, heterogeneity and evenness of an acoustic environment. Some of them are: the acoustic entropy index (H), which indicates how much the amplitude of a signal is uniform in time and frequency; the acoustic complexity index (ACI) which takes into account the variation of a signal in different frequency bins over time and then averages over the entire frequency range [127]. While easy to apply, a drawback of acoustic indices is that they are not learned from data, and they are not, therefore, discriminative for animal sounds with respect to sounds with similar patterns, but of a different origin. Therefore, in order for these techniques to be applied, only the soundscape produced by natural sources should be considered. When the objective is to detect fish vocalizations, the PAM audio signal is usually visualized as a spectrogram and visually examined by an expert. This approach exploits the fact that fish vocalize within a relatively narrow range of low frequencies and often produce repetitive sounds. In this chapter, our aim is to present a solution for distinguishing the sound produced by fish from the background noise, with the goal of automating fish sound detection process and facilitating the use of soundscape analysis for biodiversity assessment. We employ recent advances in sound separation for human speech and music to the problem of separating fish vocalizations in PAM recordings. Machine learning techniques, and deep learning, in particular, require a large amount of data. In supervised learning, data must be annotated to provide ground truth information for training neural networks. Data annotation typically involves the manual identification of the attributes one wants to automatically recover. Obtaining annotated data for the task of sound separation, given a mixed signal, is clearly challenging. A preferable strategy is to generate training data by combining individual audio sources. This approach has been largely exploited for speech, music and anthropic sound. But while for human speech and music there is an abundance of data, this is not the case for fish vocalizations. At present, to the best of our knowledge, no datasets exist that include many examples of fish vocalization examples. Nonetheless, it is widely recognized that this is required for future progress [125]. Therefore, we collected a dataset of fish vocalizations from the internet. It is worth noting that the website https://fishsounds.net/ did not exist when we started this project. In most cases, we obtained a single sound example for each species. This limits the possibility of applying AI techniques to automatically classify the fish species from sound, as several recordings for each species would be necessary. In addition, with about 35000 known species of fish, the number of known soniferous species is quite limited, and while some sources have been identified, the majority of fish sounds remain unidentified [125]. Nevertheless, vocalizations from different fish species share similarities and posses distinctive characteristics that enable training a network that can separate fish-produced sounds from the background noise, typically consisting from the sound of waves and snapping shrimps (members of the *Alpheidae* family). We created a sound separation dataset by randomly overlapping fish vocalizations with sea backgrounds recorded at various locations on the Greek island of Nisyros. We use this dataset creation process to train two recent and popular architectures for sound separation: Conv-TasNet [104] (which we will call TasNet) and Demucs (version 2) [34]. We quantitatively evaluate the performance of these networks on two synthetic test sets, one obtained with backgrounds recorded in Nisyros, and one generated with backgrounds recorded in Favignana (Italy). We qualitatively show performance on a few examples of recordings performed in Marsa Alam, Egypt

(Fig. 2.2). Our quantitative evaluation shows that the sound emitted from fishes can be successfully recovered from recordings with noisy background. The network performance visually slightly decreases in the in-the-wild setting, but still it is possible to recover the predominant fish sounds even if the networks have been trained with data captured with different devices in different locations. To our knowledge, this is the first work that applies modern sound separation techniques to PAM.

2.1.2 Related Works

The assessment of marine biodiversity through acoustic techniques is evolving rapidly and although several methods are used, at the moment none of these is considered the ideal tool for investigating the marine environment and its diversity. Some existing methods are discussed below.

Classical approaches

Spectrograms are valuable tools to analyze the temporal variation of an audio signal amplitude at different frequencies. They represent, through the Fourier transform, the 1D audio signal with a 2D image, with time and frequency as axes and pixel intensity as amplitude. By analyzing spectrograms, it is possible to identify the presence of some marine species through the identification of image patterns that are indicators of audio features in the species-specific vocalizations. For example, it is possible to observe whether the sound emitted is rhythmic or more smooth and harmonious. These patterns can be associated to known soniferous species or unidentified fish sources. While the examination of spectrograms as images enables the approximate detection of spectral patterns, it is insufficient for precise identification of the intricate modulation characteristics of underwater animal sounds. Automatic systems based on pattern recognition in images are often not sufficient for detecting fish vocalizations, and spectrograms needs to be inspected manually. However, studies focused on comprehending the drivers of marine biodiversity changes typically rely on prolonged audio recordings, spanning months or years. Conducting a manual analysis of such data is impractical. Unsupervised modeling techniques have been used to analyze spectrograms, with clustering being among the most widely adopted [194]. Assuming that the data has underlying patterns, clustering allows grouping elements with similar characteristics. Hence, large amounts of audio data can be modeled as a few audio clusters and these can be exploited to assess biodiversity by measuring per-cluster acoustic metrics. Unfortunately, this type of analysis can easily fail when non-biological sources contaminate the collected data [94]. In addition, it is still necessary to individually analyze the sources that are part of each cluster to understand the key elements that contribute to marine biodiversity.

Data-driven approaches

Several studies [65, 93, 44, 217, 204, 71, 103, 174, 92, 17] have been carried out to trace, recognize, and isolate the biological sound sources present in nature. The use of machine learning has been fundamental to obtaining significant results. In particular, in [103, 174], the authors propose using deep learning to detect *odontocete echolocation* and bird sounds, respectively. In [17], Clink et al. introduce a workflow for the automated detection and classification of female gibbon calls, testing supervised and unsupervised approaches. In [92], Li et al. propose to use generative adversarial networks (GANs) to generate training data for learning to extract of toothed whales' whistles from time-frequency spectrograms. Recently, Sun et al. [177] have introduced a toolbox for soundscape information retrieval based on non-negative matrix factorization.

Source separation

The work that has led to significant progress in this field is mainly in music and speech. In these contexts, the separation task is particularly challenging due to the inherent complexity of overlapping harmonics, temporal and spectral variability, and unpredictable background noise. Deep learning has brought significant advances to source separation by leveraging the ability of neural networks to model complex, non-linear relationships and learn high-level abstract features from data. This paradigm has provided a robust, data-driven approach to the source separation problem, outperforming traditional signal processing methods. One of the seminal works in speech separation is [104], where they propose an end-to-end, fully-convolutional time-domain audio separation network that significantly outperformed traditional frequency-domain methods. While, for music source separation, in [34], a model is proposed that relies on depthwise separable convolutions and bidirectional LSTMs (Long Short-Time Memory), leading to improved performance over previous state-of-the-art methods. Further advances for music source separation have been made in [130], where a novel Bayesian method for unsupervised source separation is introduced.

2.1.3 Method

The problem of separating audio sources consists of breaking down a mixture of signals $y(t) \in \mathbb{R}^T$ into its n components $c_1(t), \ldots, c_n(t) \in \mathbb{R}^T$, where,

$$y(t) = \sum_{i=1}^{n} c_i(t). \tag{2.1}$$

The mixture is represented as a vector in the waveform domain. In our case, we consider $n = 2$ sources: fish and background. In order to perform sound separation, we employ the two aforementioned networks, TasNet and Demucs. These two types of networks are trained in a supervised manner, and while both have an encoder-decoder structure and act directly on the audio waveform, they are fundamentally different: TasNet learns a mask to be applied to the mixture to filter the desired source signal, whereas Demucs learns to directly synthesize the required signals without using any filtering. We train both network with supervised training, on the same dataset. Critical for the success of the separation networks is the availability of a large training dataset with overlapped and separated sources. The availability of such dataset for the specific case of fish vocalizations poses several challenges. Here, we contribute with a novel synthetic dataset that we define as follows. We collected a large set of recorded vocalizations from online sources, these will be the basis of the foreground fish source. At the same time, we recorded a set of diverse sea recordings that constitute the data to represent background sound. Details on the collected data are reported in the Experiments section. During training, at each epoch we create random combinations, with randomized amplitude, of fish and background audio data. In this way, despite the limited number of sound sources, in particular for the fish data, we prevent the networks from overfitting on a fixed training dataset. Audio data is loaded from the network as a set of audio chunks of length 44160, obtained by splitting the audio data with an overlap fraction of 0.25. The foreground fish vocalization dataset is also loaded as a set of samples with 0.25 overlap, where each sample is a chunk of size 44160. The synthetic data for training is created as follows. At each epoch, for each foreground sample i, we define the two audio sources s_0 (foreground) and s_1 (background) as follows:

$$\begin{aligned} s_0 &= k_f \alpha_f x_f \\ s_1 &= (1 + k_b) x_b, \end{aligned}$$

where x_f is the sample with index i, and x_b is a random background chunk; k_f and k_b are two random coefficients sampled from a uniform distribution, while α_f is a fixed attenuation factor for the fish audio, required to model relative amplitude in real conditions. In this way, at each epoch, every fish sample is combined differently with a random background. We set $\alpha_f = 0.1$.

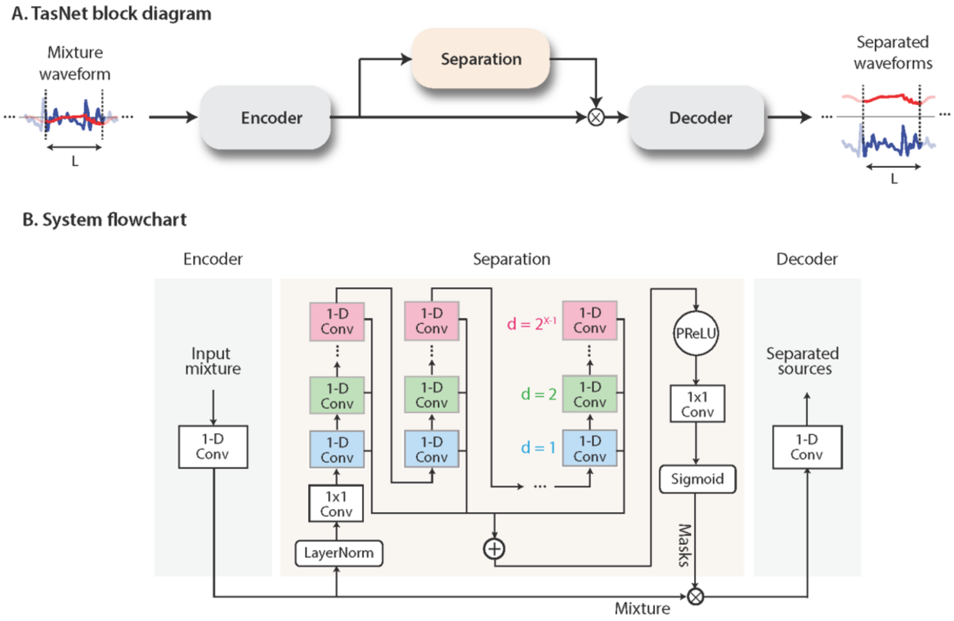


Figure 2.3. (A): TasNet block diagram. A piece of the input signal is projected into a multidimensional hidden space through the encoder. Then, a separation module calculates an estimated mask for each individual source. Ultimately, a decoder converts these masked encoded features back into waveform domain signals. (B): System flowchart. The encoder consists of a 1D convolutional module that maps the mixture into the features space. A temporal convolutional network (TCN) calculates the mask vectors, and the decoder reconstructs the separated signals by a 1D transposed convolution operation. In the separation module, different dilation factors in each 1D Conv block are highlighted with different colors. This figure is taken from [104].

TasNet

TasNet is a convolutional audio separation model in the time domain, composed by an encoder, a separation module and a decoder, as shown in figure 2.3 (A). The encoder transforms small overlapping fragments of the mixture into feature vectors in an intermediate latent space. Using this representation, the separation module calculates a mask for each source. Each mask, multiplied by the respective intermediate representation of the mixture, generates the latent features of the relative source. Finally, the decoder converts each latent representation into a time-domain waveform, thus obtaining the desired separated signals. In figure 2.3 (B) we report the entire system flowchart from [104].

Encoder Initially, the input mixture is divided into N overlapping parts $\mathbf{x}_i \in \mathbb{R}^L$, where $i = 1, \ldots, N$, each of length L. Each $\mathbf{x}_i$ is transformed by the encoder into the corresponding vector in the latent domain $\mathbf{z}_i \in \mathbb{R}^M$ through a 1D convolution operation (formally expressed by a matrix multiplication) followed by a ReLU activation function $\mathcal{G}(\cdot)$:

$$\boldsymbol{z}_i = \mathcal{G}(\boldsymbol{x}_i \boldsymbol{S}) \,, \tag{2.2}$$

where $\boldsymbol{S}$ is a $L \times M$ matrix of convolution coefficients.

Separation module The actual separation of each fragment of the mixture occurs in the separation module, in which n mask vectors $\mathbf{m}_i \in \mathbb{R}^M$ are estimated, where $i = 1, \ldots, n$ and n is the number of signals to be separated. Each of these vectors, being masks, must necessarily be $\mathbf{m}_i \in [0, 1]$. The vector representation in the latent space $\mathbf{b}_i \in \mathbb{R}^M$ of each signal is calculated by multiplying the relative mask $\mathbf{m}_i$ by the mixture $\mathbf{z}_i$,

$$\mathbf{b}_i = \mathbf{z}_i \odot \mathbf{m}_i \tag{2.3}$$

where $\odot$ denotes element-wise multiplication. This module is a temporal convolutional network (TCN) [86], which is fully convolutional and consists of stacked 1D dilated convolutional blocks with increasing dilation factors. These factors make it possible to gradually capture increasingly broad contexts, thus exploiting long-range dependencies within the signal. Here, with respect to the architecture described in [104], we do not make use of the skip connections in the 1D convolutional blocks.

Decoder The reconstruction of each source is computed by the decoder. The latter takes as input $\mathbf{z}_i$ and returns a vector $\hat{\mathbf{x}}_i$ in the waveform domain by applying a 1D transposed convolution operation,

$$\hat{\mathbf{x}}_i = \mathbf{z}_i \mathbf{T} \tag{2.4}$$

where $\hat{\mathbf{x}}_i \in \mathbb{R}^L$ is the reconstruction of $\mathbf{x}_i$ and $\mathbf{T}$ is a $M \times L$ matrix of convolution weights.

Demucs

Demucs is an autoencoder model made of a convolutional encoder and a convolutional decoder linked with skip U-Net connections and a 2-layers bidirectional LSTM. The size of the latent space is $C_B = 6$.

Encoder As illustrated in figure 2.4, the encoder consists of $B = 6$ stacked convolutional layers, and the number of output channels C_i in each layer equals the number of input channels C_{i+1} in the next layer. From the second layer onwards, the output channels are twice the number of input channels. All these stacked layers have the task of compressing the information in order to obtain a compact representation of the training data. The input channels in the first layer are $C_0 = 2$ and the output channels are $C_0 = 100$. The output channels in the last layer are $C_B = 3200$, which is the hidden size of the LSTM.

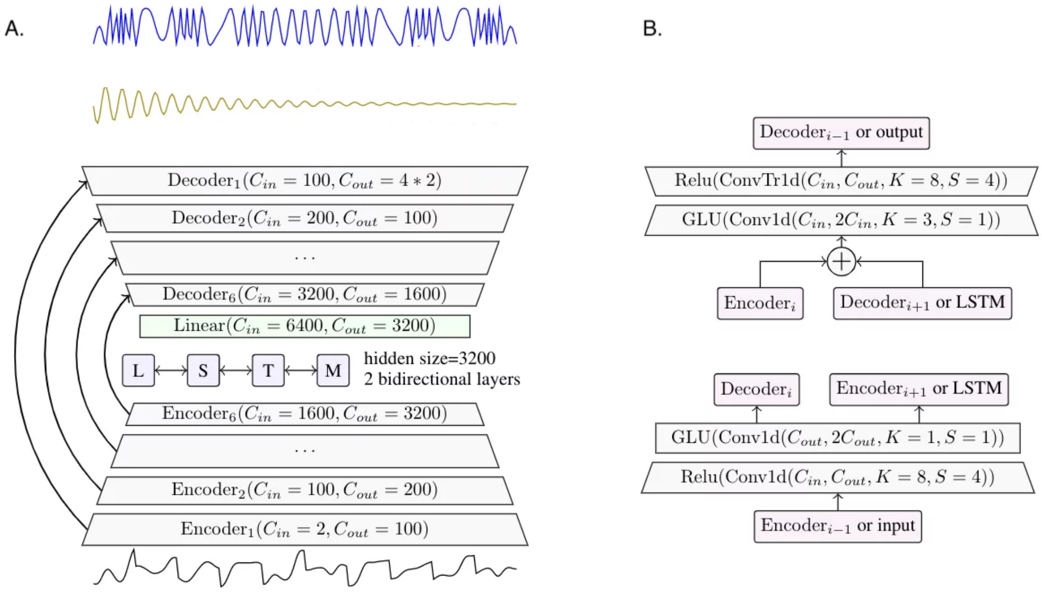

A.
Decoder$_1$($C_{in} = 100, C_{out} = 4 * 2$)
Decoder$_2$($C_{in} = 200, C_{out} = 100$)
. . .
Decoder$_6$($C_{in} = 3200, C_{out} = 1600$)
Linear($C_{in} = 6400, C_{out} = 3200$)
L
S
T
M
hidden size=3200
2 bidirectional layers
Encoder$_6$($C_{in} = 1600, C_{out} = 3200$)
. . .
Encoder$_2$($C_{in} = 100, C_{out} = 200$)
Encoder$_1$($C_{in} = 2, C_{out} = 100$)
B.
Decoder$_{i-1}$ or output
Relu(ConvTr1d($C_{in}, C_{out}, K = 8, S = 4$))
GLU(Conv1d($C_{in}, 2C_{in}, K = 3, S = 1$))
Encoder$_i$
Decoder$_{i+1}$ or LSTM
Decoder$_i$
Encoder$_{i+1}$ or LSTM
GLU(Conv1d($C_{out}, 2C_{out}, K = 1, S = 1$))
Relu(Conv1d($C_{in}, C_{out}, K = 8, S = 4$))
Encoder$_{i-1}$ or input

Figure 2.4. (A): Demucs model with the input mixture and the two output sources, all in the waveform domain. (B): Encoder/decoder block architecture. In each encoder block, there is a convolution with kernel size $K = 8$ (to have dependencies with adjacent time steps) and stride $S = 4$ followed by a ReLU activation function. The result is given as input to another convolution with kernel size $K = 1$ and stride $S = 1$, in order to increase the expressivity of the network with little additional computation. In the end, a gated linear unit (GLU) activation function [19] is applied. The decoder block is constructed in reverse order with respect to the encoder, and it consists of a convolution with kernel size $K = 3$ and stride $S = 1$, followed by a GLU and then a transposed convolution with kernel size $K = 8$ and stride $S = 4$, followed by a ReLU. This figure is taken from [34].

Decoder Since LSTM outputs a tensor with $2C_B$ channels, a linear layer is needed to reduce the number of channels to C_B. The decoder is built essentially like the encoder, but with the convolutional layers put in reverse order and transposed convolutions instead of the regular convolutions. The decoder has the task of expanding the dimensions of the compressed vectors in the latent space to regain vectors with sizes equal to those of the input space. The last layer returns tensors with $N \cdot C_0$ channels, synthesizing the N sources present, initially, in the input mixture.

U-network In this architecture, the encoder layers are connected to the decoder layers with the same index through skip connections, as happens in the Wave-U-Net [67]. The objective of these connections is to connect the various decoder layers with those of the encoders to transfer information directly from ones to the others in such a way as to facilitate reconstruction. Compared to Wave-U-Net, Demucs skip connections use transposed convolutions instead of linear interpolations, since they require less memory and computational time.

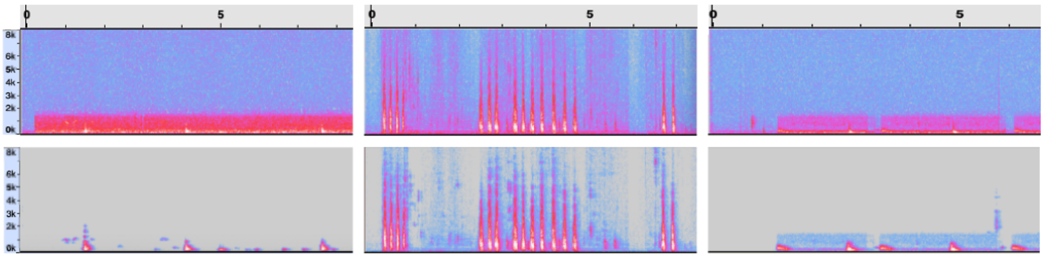


Figure 2.5. Fish vocalization before (top) and after (bottom) noise removal and normalization. Time in seconds.

2.1.4 Experiments

Fish Vocalization Data

We collected 191 audio files corresponding to the vocalization of 143 different species. Most of the recordings were downloaded from FishBase[1]. The collected data often exhibit unnatural noise, since in many cases the recordings are performed in fish tanks. In order to create a dataset that can be used to synthesize realistic audio data, we preprocessed for noise removal, and normalized for a peak amplitude of -1 dB. For this purpose, we used the open source software Audacity. Figure 2.5 illustrates examples before and after preprocessing. We employ the fish vocalizations for the online creation of training samples by combining them with recorded sea backgrounds ad described previously, and for creating a synthetic testset with ground-truth separated signals.

Sea Recordings

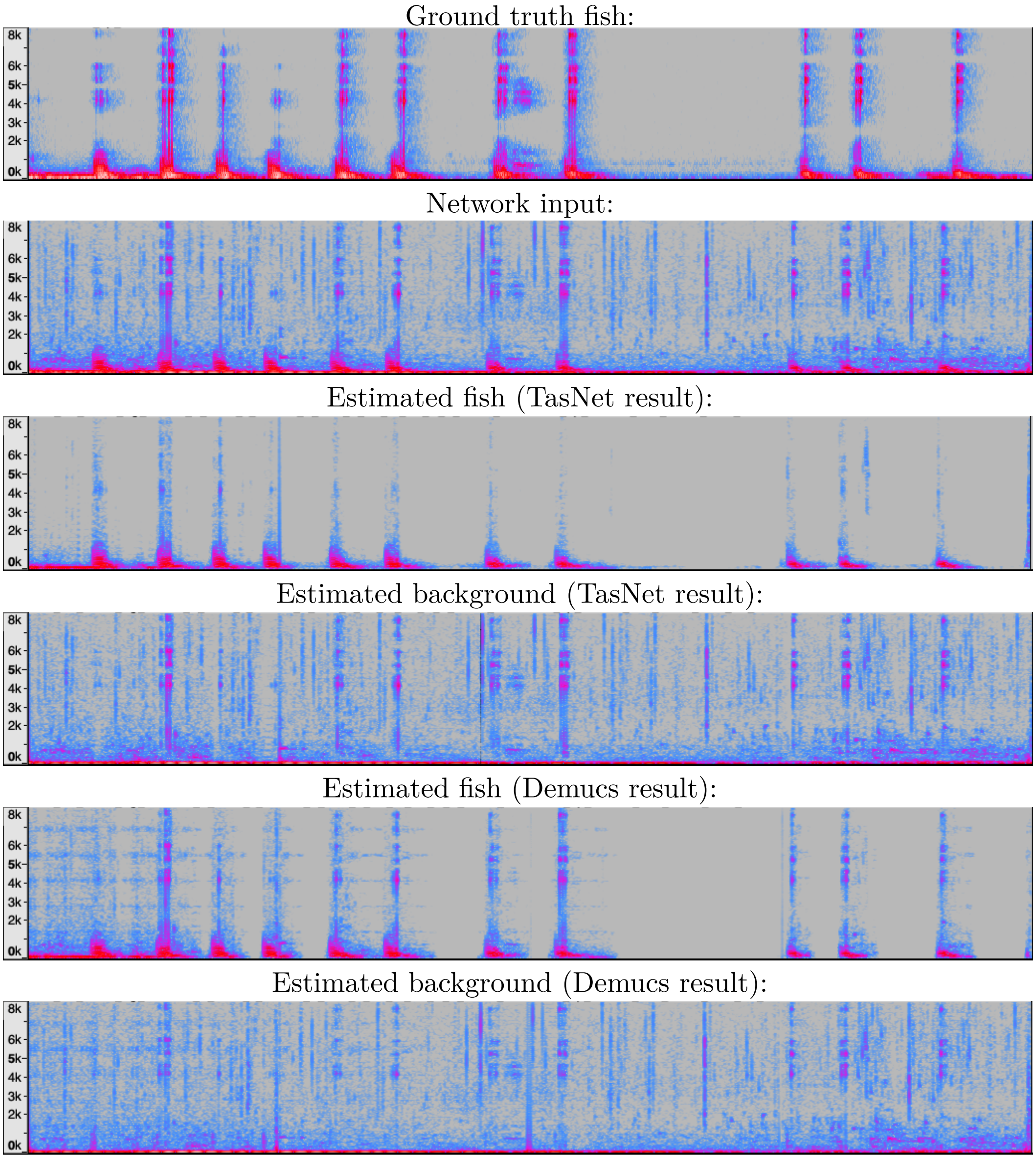
We performed sea recordings at the Greek island of Nisyros and at the Italian island of Favignana. Greek recordings were performed in October 2019, April 2021, August 2021, and October 2021 at different sites around the island, both near the coast and in the open sea; whether the Italian ones in June 2023. Data were captured with an Aquarian Scientific AS-1 hydrophone (linear range 1Hz to 100kHz ±2dB, operating depth 200 mt). Sea recordings in Nisyros are used as backgrounds for training and test. In addition, we collected a sound video dataset at Marsa Alam (Egypt) using an action camera Sony HDR-AS50 Full HD. The audio channel from this data is used for a qualitative evaluation.

Networks Training

During training, we considered different 11 sea recordings for creating backgrounds, captured in different locations and different times, and of various duration. The first 5 files were captured with sample rate of $192K$ and were converted to $44K$. Recordings with length greater than 3 minutes were divided in multiple files of smaller duration $(1-2$ minutes) and constitute a dataset of background chunks. We have a total of 133 files for background. A couple of recordings that we use for representing backgrounds were manually filtered for removing fish sounds. This was not necessary for most of the recordings, where fish sounds were harder to find.

We use the 80% of the fish vocalization data for training. We fed both the networks with samples of size 44160 and trained both the networks with a learning rate of 0.0001 and a number of epochs equal to 200. TasNet employs an autoencoder

[1] www.fishbase.org


Ground truth fish:

Network input:

Estimated fish (TasNet result):

Estimated background (TasNet result):

Estimated fish (Demucs result):

Estimated background (Demucs result):

Figure 2.6. Synthetic testset example. From top: fish vocalization (*Prionotus*) (4 seconds); overlap with sea background; TasNet fish and background separation; Demucs fish and background separation. Vertical axis is frequency, horizontal axis is time.

with 512 filters (N), each of length 256 (L). The bottleneck has 256 channels (B), and the convolutional blocks contain 512 channels (H). For convolutional operations, we use a kernel size of 3 (P) across 8 blocks (X) that are repeated 4 times (R). The network is designed to separate inputs into 2 distinct "speakers" (C).

Evaluation

For the quantitative evaluation, we generated a set of synthetic inputs using the 20% fish vocalizations that were not used for training, combining these sounds at random with background chunks also not used for training. For the qualitative evaluation of the data recorded in Egypt, we trained the network using the whole vocalization dataset. We apply the trained TasNet and Demucs networks to the synthetic testset and quantify the sound separation performance, computing an SDR (Source to Distortion Ratio) score [197], is considered an excellent metric to assess sound quality, between recovered and ground truth fish and background audio sources. In order to compute the SDR score, the reconstruction $\hat{s}_i$ of a source s_{target} is assumed of consisting of four components:

$$\hat{s}_i = s_{target} + e_{interf} + e_{artif} + e_{noise}$$

where e_{interf}, e_{artif} and e_{noise} are respectively error terms for interference, artifacts and noise [197]. Using these terms, the SDR is expressed as:

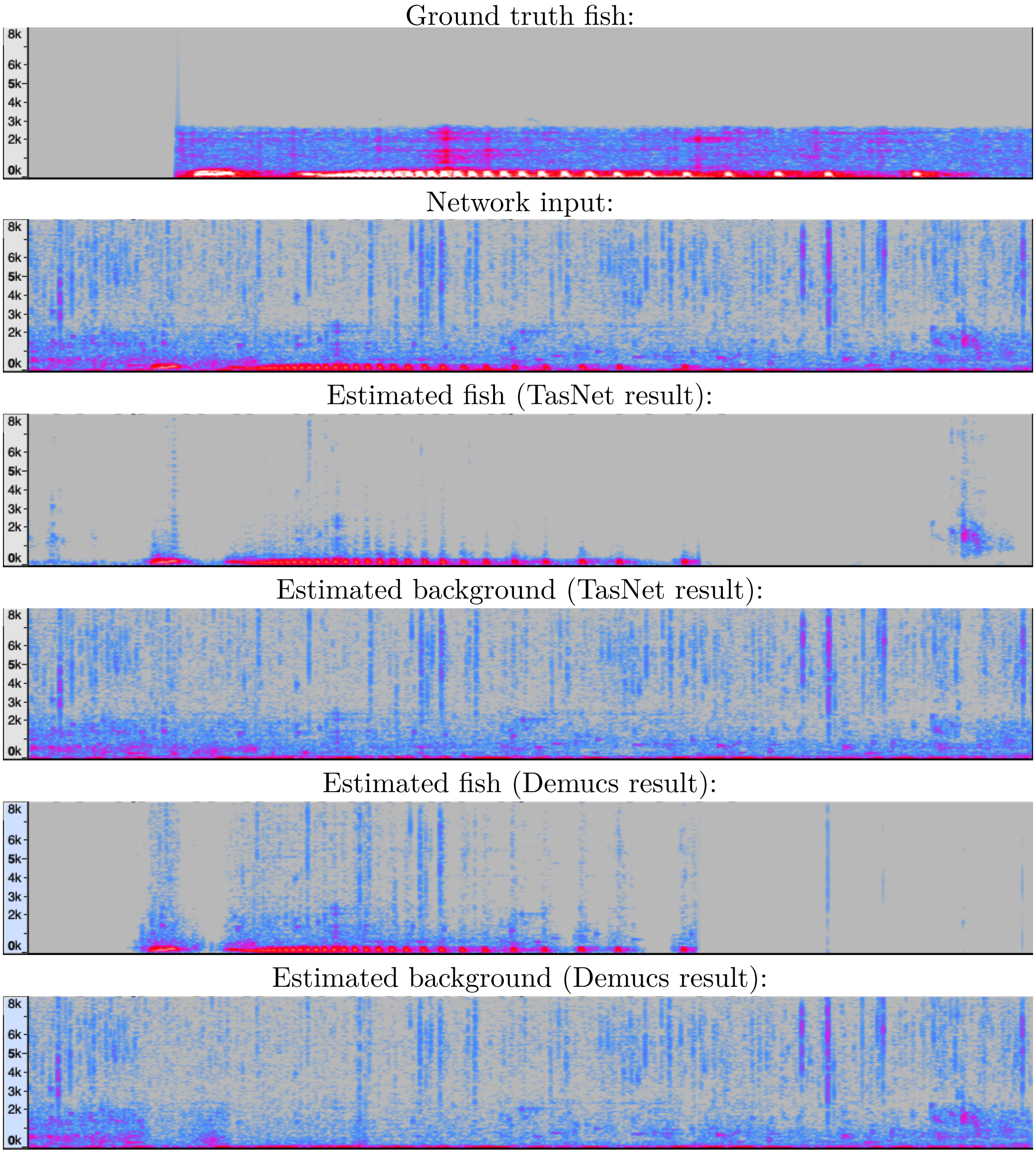
$$\text{SDR} := 10 \log_{10} \left(\frac{\|s_{target}\|^2}{\|e_{interf} + e_{artif} + e_{noise}\|^2} \right) .$$

Table 2.1 reports our results (the higher, the better).

Table 2.1. Quantitative evaluation on the synthetic testset using Nisyros backgrounds.

Metric	**TasNet**	
	Channel	*Value*
SDR	Fish	**10.60 ± 9.00**
SDR	Background	**17.60 ± 7.04**
Metric	**Demucs**	
	Channel	*Value*
SDR	Fish	−3.71 ± 2.03
SDR	Background	2.65 ± 4.05

We note from Table 2.1 that the Tasnet network performs significantly better than Demucs. The former reaches an SDR score equal to 10.59 on the separation of the sound of the fish and 17.60 on the background, while the latter obtains just 2.65 of SDR on the background and even a negative score on fish, equal to -5.96 of SDR. Figure 2.6 and 2.7 show two randomly selected examples of separation. Although the separations produced by Demucs appear to be perceptibly better, it can be seen how they show artifacts; in particular, vertical lines are introduced that are repeated periodically, while in Tasnet, this behavior is not present. Furthermore, it is possible to notice how, on the synthetic data, in correspondence with the sounds of the fish, Demucs generates fictitious frequencies that are not present in the TasNet separations. This is probably due to the fact that Demucs is a network that does



Ground truth fish:

Network input:

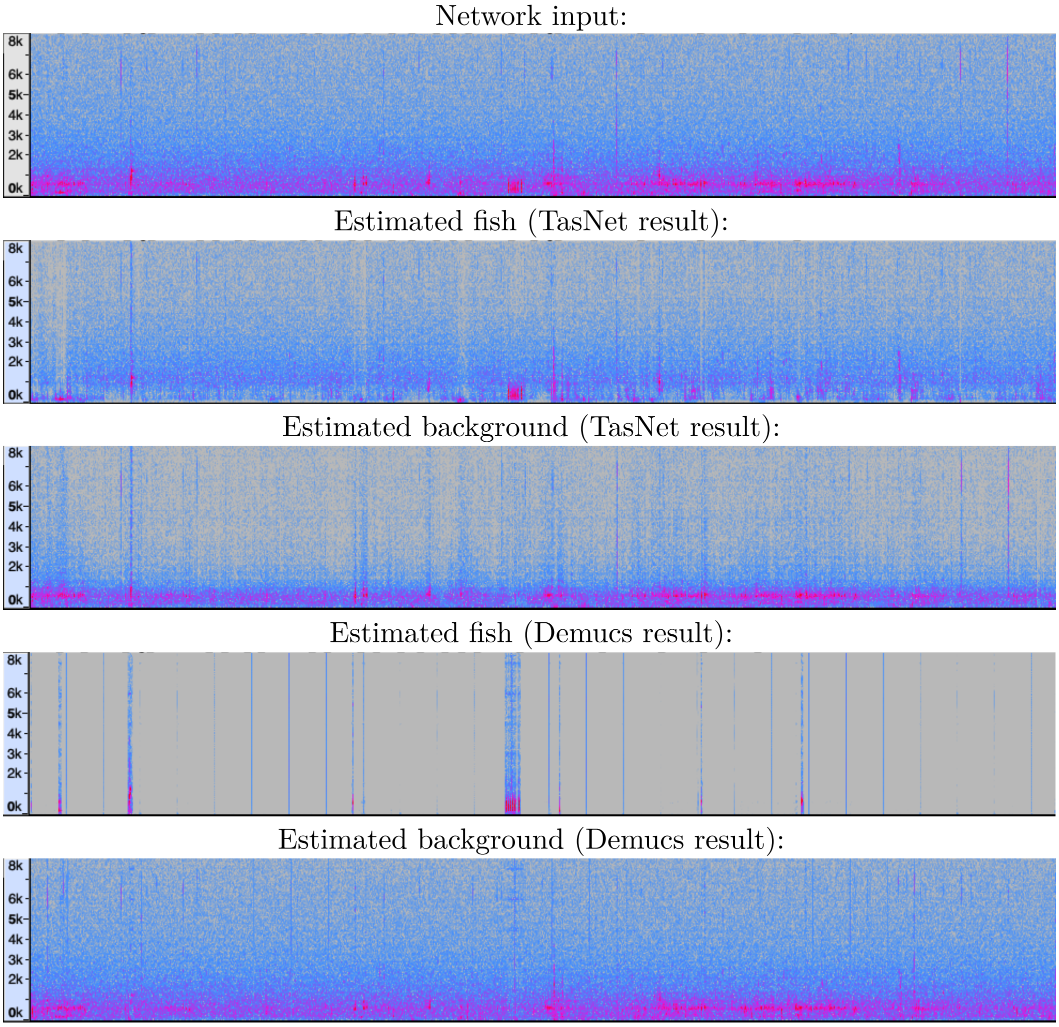
Estimated fish (TasNet result):

Estimated background (TasNet result):

Estimated fish (Demucs result):

Estimated background (Demucs result):

Figure 2.7. Synthetic testset example. From top: fish vocalization (*Epinephelus guttatus*) (2 seconds); overlap with sea background; TasNet fish and background separation; Demucs fish and background separation. Vertical axis is frequency, horizontal axis is time.


Network input:

Estimated fish (TasNet result):

Estimated background (TasNet result):

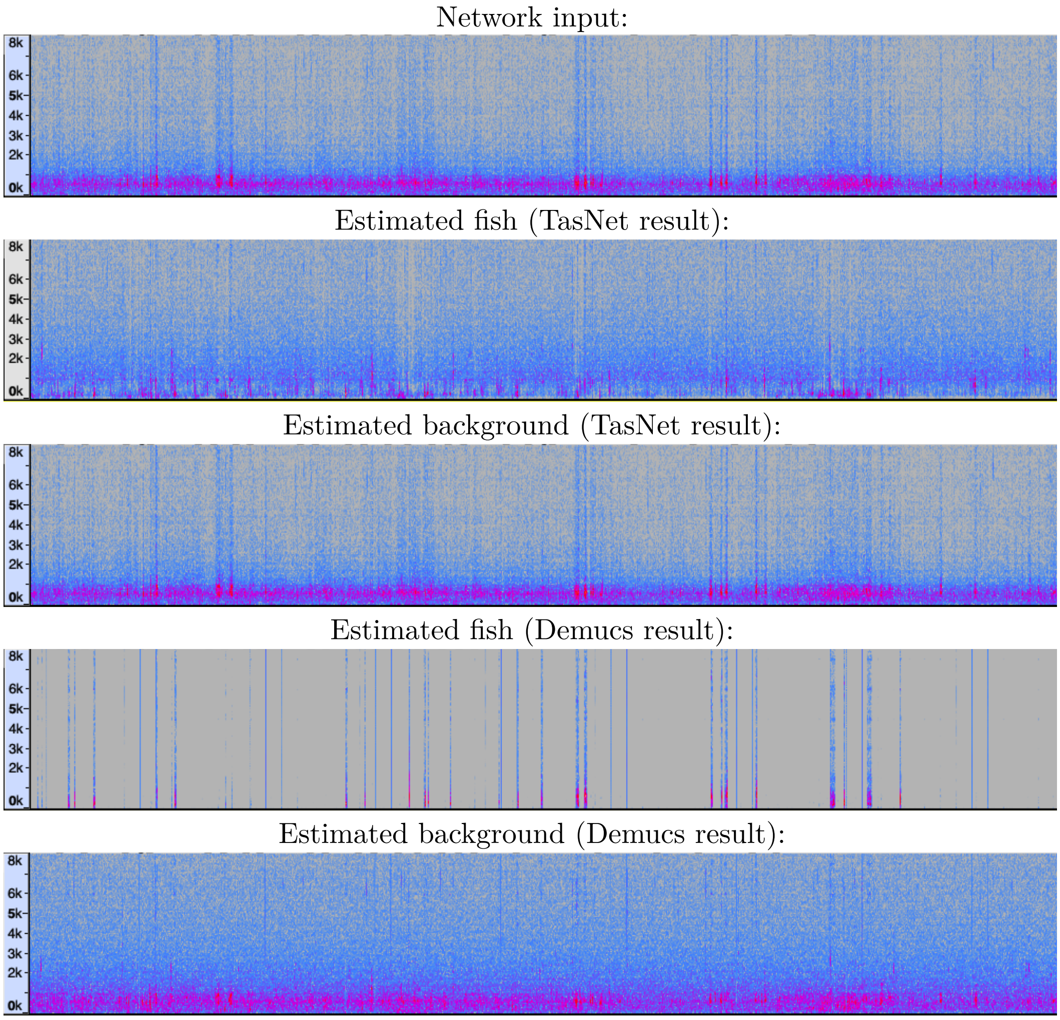
Estimated fish (Demucs result):

Estimated background (Demucs result):

Figure 2.8. In-the-wild experiment at Marsa Alam (20 seconds). From top: sea recording; TasNet fish and background separation; Demucs fish and background separation. Vertical axis is frequency, horizontal axis is time.

Table 2.2. Quantitative evaluation on the synthetic testset using Favignana backgrounds.

Metric	**TasNet**	
	Channel	*Value*
SDR	Fish	**8.11 ± 15.48**
SDR	Background	**6.27 ± 5.14**
Metric	**Demucs**	
	Channel	*Value*
SDR	Fish	−5.27 ± 7.92
SDR	Background	−2.81 ± 2.14



Network input:

Estimated fish (TasNet result):

Estimated background (TasNet result):

Estimated fish (Demucs result):

Estimated background (Demucs result):

Figure 2.9. In-the-wild experiment at Marsa Alam (60 seconds). From top: sea recording; TasNet fish and background separation; Demucs fish and background separation. Vertical axis is frequency, horizontal axis is time.

not separate the signal by filtering it, but by directly synthesizing the requested source, not performing well with the data of our dataset. Instead, Tasnet, a more classical network that filters the desired signal from the mixture, appears to be more robust and performs better with the data in our possession. Results in Table 2.2, obtained with a set of backgrounds captured at different locations in relation to the data used for training, further confirm the above discussion. Figure 2.8 and 2.9 show two examples of separation applied to the data recorded in Marsa Alam. Note that the performance of the networks are here qualitatively lower than on the synthetic dataset, and this can be due in particular to the distribution shift between the background data: while in the Aegean Sea we noticed a consistent presence of clicks sounds emitted by shrimps, these are not present in the Marsa Alam dataset. Moreover, the latter data includes a significant sensor noise. Despite these differences, both networks are able to identify sounds that we can attribute to the fish species observed in the video channel of the captured data.

2.1.5 Conclusion

With this study, we demonstrate the effective application of deep learning techniques for source separation of marine data. We achieve this by applying the most effective source separation architectures to the problem of isolating fish vocalizations from sea background, obtaining competitive signal-to-distortion ratio (SDR) scores on a synthetic test set generated composing real animal and background sources. Notably, as observed by experts, our trained networks also perform qualitatively well on in-the-wild data, captured with a different device in a different environment. We attribute this generalization ability to our online training strategy, where a new synthetic training set is generated at each epoch. We hope these results will pave the way for new methods of studying the marine environment and contribute to developing new automatic PAM techniques for monitoring marine biodiversity and, possibly, accurately tracking fauna in the oceans.

2.2 Exploiting music source separation for singing voice detection

Singing voice detection (SVD) is essential in many music information retrieval (MIR) applications. Deep learning methods have shown promising results for SVD, but further performance improvements are desirable since it underlies many other tasks. This work proposes a novel SVD system combining a state-of-the-art music source separator (Demucs) with two downstream models: a Long-term Recurrent Convolutional Network (LRCN) and a Transformer network. Our work highlights two main aspects: the impact of a music source separation model, such as Demucs, its zero-shot capabilities for the SVD task, and the potential for deep learning to improve the system's performance further. We evaluate our approach on three datasets (Jamendo Corpus, MedleyDB, and MIR-1K) and compare the performance of the two models to a baseline root mean square (RMS) algorithm and the current state-of-the-art for the Jamendo Corpus dataset.

2.2.1 Introduction

Singing voice detection (SVD) is a classification task determining whether a singing voice exists in a given audio segment. It is crucial in many Music Information Retrieval (MIR) applications, such as lyrics alignment [43], singer identification [214, 215], and lyrics transcription [112]. Traditional approaches to SVD focused on analyzing the audio mixture directly, extracting features from the raw waveform, and employing various machine-learning techniques to classify the singing voice.

In recent years, there has been a shift in the SVD research landscape due to the growing interest in utilizing music source separation (MSS) techniques. Two years ago, [215] proposed a state-of-the-art (SOTA) SVD method that leverages MSS to preprocess the input audio signal and subsequently classify the singing voice. This approach has significantly improved SVD performance, indicating the potential benefits of incorporating MSS techniques in SVD systems.

In this chapter, we follow this research direction and build upon the SOTA MSS method, Demucs, by integrating it with two downstream models: LRCN and a Transformer network. Our study aims to assess the effectiveness of MSS methods on SVD tasks and determine whether the two downstream models can further enhance performance.

2.2.2 Related Works

In recent years, there has been a growing interest in using deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), such as long short-term memory (LSTM) networks, for solving the task of singing voice detection. Lee et al. [89] proposed end-to-end approaches using CNNs and LSTMs, respectively, to process the audio mixture and classify the singing voice directly.

In addition to the choice of neural network architecture, researchers have also explored different feature extraction techniques to represent the audio mixture. In 2015, Schlüter et al. [154], Lehner et al. [91], and Leglaive et al. [90] proposed three different methods based on CNNs, LSTM-RNNs, and bidirectional LSTM (Bi-LSTM) networks, respectively. These methods extract high-level audio features such as spectrograms, mel-spectrograms, MFCCs, and singing voice and percussive components from the harmonic-percussive source separation (HPSS) algorithm. Schlüter et al. [154] investigated the effectiveness of data augmentation methods on spectrograms and mel-spectrograms, while Lehner et al. [91] used 30 MFCCs and Leglaive et al. [90] combined the singing voice and percussive components from the HPSS algorithm. The best model achieved an accuracy of 0.923 [154], while the best f1-score was 0.910 [90].

In 2016 and 2017, Choi et al. [13, 14] studied the effectiveness of using pre-trained convnet features for singing voice detection and achieved similar results compared to previous works.

Recently, Zhang et al. [216] proposed a new approach based on a CNN combined with an LSTM network called the long-term recurrent convolutional network (LRCN). Unlike previous methods, the LRCN approach used the singing voice-separated signal as input rather than the audio mixture. This approach achieved an accuracy of 0.924 and an f1-score of 0.927, outperforming previous methods. In addition to the differences in network architecture and feature extraction techniques, the LRCN approach differs from previous end-to-end approaches in using a pre-trained vocal separation algorithm to extract the singing voice signal. This approach could improve the classification quality and reduce interference from other audio sources, but it is sensitive to the separation quality.

2.2.3 Method

In 2021, a new open-source SOTA music source separator was proposed in [21]. It was developed by Facebook AI Research [2] and introduced a number of architectural changes to the previous Demucs architecture [34], considerably improving the quality of source separation for music. We used this state-of-the-art (SOTA) pre-trained model to address SVD, stacking Demucs to a downstream system, as proposed in [216]. To verify the importance of a music source separation model, we trained two networks: an LRCN and a Transformer, directly on the mixtures and the separated vocal tracks produced by Demucs on Jamendo Corpus. In addition, to assess the zero-shot capabilities of Demucs, we stack a classical signal processing root mean square (RMS) algorithm after it. Moreover, we tested the combination of Demucs and a neural network (the LRCN and the Transformer) on other datasets, such as MIR1k and MedleyDB, to assess the potential for deep learning to improve performance further.

To the best of our knowledge, we are the first to use the Transformer network for the task of SVD, motivated by the successes of this architecture in other classification tasks [32, 124].

The subsequent sections will explain these three systems in the following sequence: RMS, LRCN, and Transformer.

Voice Activity Detection through RMS

The root mean square (RMS) of an audio signal is defined as the square root of the mean of the squared values of the signal samples. To compute the RMS, the audio signal is first squared at each sample point, then the squared values are averaged over the entire signal, and finally, the square root of the average is taken. Mathematically, this can be expressed as:

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x[n])^2} \tag{2.5}$$

[2] https://github.com/facebookresearch/demucs

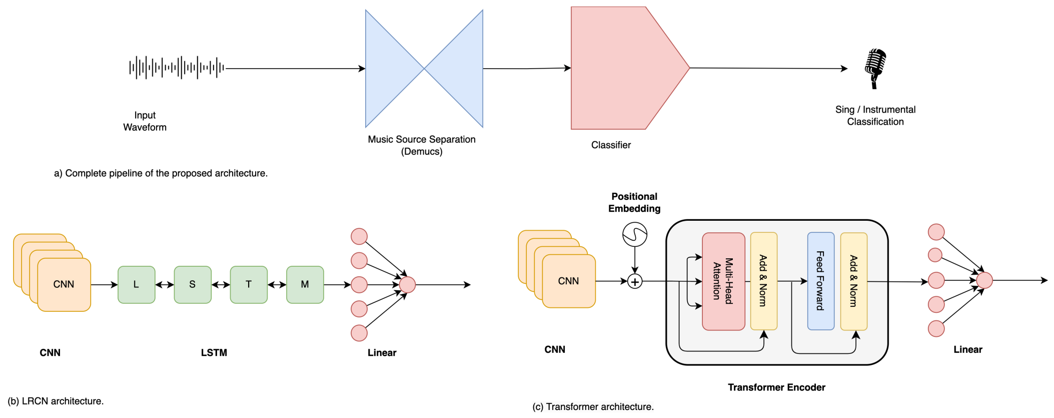


Figure 2.10. Panel (a) shows the general pipeline, consisting of Demucs followed by a generic Classifier. Panels (b) and (c) detail the two neural Classifiers we tested that follow Demucs.

where $x[n]$ is the audio sample at time n, and N is the total number of samples in the signal.

The RMS metric is a reliable method for measuring the overall amplitude of an audio signal, as it captures both the strength and the duration of the signal. This metric is well-suited for identifying sections of an audio signal in which a person is singing, even in the presence of residual noise and artifacts. Applying the RMS metric to the isolated vocal source obtained from a music source separation system makes it possible to obtain a time-varying measure of the vocal amplitude throughout the song. We used the default parameters of the librosa RMS algorithm, while the threshold used to discriminate between vocals and non-vocals was determined by a grid search in the validation set of the Jamendo Corpus dataset and was set to 0.015. This method can identify the sections in which the singer is present.

LRCN Network

The Long-term Recurrent Convolutional Network (LRCN) architecture, first proposed in [161], is a powerful deep learning model designed for a wide range of applications, including video classification, image captioning, image classification, activity recognition, image labeling, video captioning, and singing voice detection [216]. The network has two main components: a spatial Convolutional Neural Network (CNN) and a temporal Long Short-Term Memory (LSTM) network. The CNN component of the LRCN network is responsible for extracting spatial features from the raw waveform signal. Specifically, the CNN applies a sliding window approach to convolve over the waveform, generating feature maps that capture different aspects of the signal's spatial structure. The LSTM component then processes these feature maps. It is responsible for capturing the temporal dynamics of the audio signal. It processes the feature maps the CNN component generates sequentially over time, using a set of memory cells to capture long-term dependencies between different time steps. The output of the LSTM component is a sequence of high-level features that encode the temporal dynamics of the audio signal. By combining the spatial and temporal features extracted by the CNN and LSTM components, the LRCN network can learn a rich representation of the input audio signal that is well-suited for a wide range of audio processing tasks.

Transformer Network

The Transformer network architecture is a highly effective deep-learning model for processing sequences data [196]. The Transformer is a type of neural network that uses self-attention mechanisms to capture the long-term dependencies in the input sequence. Specifically, the Transformer is designed to model the relationships between different feature maps by considering all pairs of feature maps simultaneously and then computing a weighted sum of the feature maps based on their relative importance. As in the previous architecture, a Convolutional Neural Network (CNN) precedes the Transformer and extracts spatial features from the raw waveform input. The resulting feature maps are then passed to the Transformer component, responsible for processing the sequence of feature maps over time to classify the singing and instrumental sections. By combining the spatial features extracted by the CNN with the self-attention mechanisms of the Transformer, the resulting model is able to learn a highly effective representation of the input audio signal that is well-suited for classification tasks.
These two approaches (LRCN and Transformer) enable the model to automatically learn to recognize singing and instrumental sections directly from the raw waveform input without needing hand-crafted feature engineering, as done in [13, 14].

2.2.4 Experiments

The experiments were designed to investigate two aspects: *(i)* the impact of a music source separation model, such as Demucs, and its zero-shot capabilities for the SVD task; *(ii)* the potential for deep learning to improve performance further.

To address the first aspect, we train the LRCN and the Transformer directly on mixtures on the Jamendo Corpus dataset [139] (without separating them with Demucs). Later, we kept the weights of Demucs frozen and trained the LRCN and Transformer networks on the Jamendo Corpus dataset, feeding these two models with the separated singing voice signal. Furthermore, to verify the zero-shot capabilities of Demucs, we also measure the performance of a standard signal processing metric (RMS) directly on the separated singing voice signal by Demucs, as described previously.

To address the second aspect, we further investigate the performance of the deep learning models and verify whether training on a specific dataset would enable them to outperform the RMS metric consistently. We expanded our experiments by training the LRCN and Transformer networks (keeping the weights of Demucs frozen) on the separated vocal tracks of two additional datasets: MedleyDB and MIR-1K.

Also inspired by [154], we noticed that our three augmentation techniques improved the neural networks' performance. Augmentations were applied to the separate vocal tracks. The first technique is pitch shifting, which shifts sounds up or down in the frequency spectrum without changing the tempo. The second technique is a gain adjustment, which involves multiplying the audio by a random amplitude factor to reduce or increase the volume, helping the model invariant to the input audio's overall gain. The third technique involves the addition of background noise, while the last approach is polarity inversion, which reverses the audio waveform, effectively inverting the signal phase. All these techniques aim to increase the diversity of the training data and assist the models in learning invariant features for the given task.

In the following subsection, we will describe in detail the three datasets we used to perform the experiments.

Datasets

The Jamendo Corpus includes 93 songs with Creative Commons licenses from Jamendo's free music-sharing website, constituting approximately 7 hours of music in total. Each file is a stereo track with a sampling rate of 44.1 kHz and is manually annotated with singing and no-singing parts by the same person to provide ground truth data. The Jamendo Corpus is publicly available [3]. The official split provides 61 songs for training, 16 for validation, and 16 for testing; the total singing and no-singing frames are about 50% of the whole set for each label, so the dataset is well-balanced.

The MIR1k dataset comprises 1000 singing voice tracks with a musical background. Clips vary in duration between 4s and 13s, their sampling rate is 16kHz, and the dataset has a cumulative duration of 133 minutes. The selected snippets come from 110 karaoke tracks, and they are chosen from a pool of 5,000 Chinese pop songs and performed by MIR lab researchers (consisting of 8 women and 11 men). Annotations of pitch contours in semitone, indices, and types for unvoiced frames, lyrics, and vocal/non-vocal segments were made manually. The MIR1k dataset is publicly available [4].

The MedleyDB dataset comprises 61 audio tracks in WAV format (44.1 kHz, 16-bit) featuring vocal signals accompanied by melody annotations. Each track includes melody annotations and instrument activation data for assessing automatic instrument identification. Labels for vocal and non-vocal segments are determined by pitch values, with nonzero pitch categorized as vocal and zero as non-vocal. A semi-automated process employing monophonic pitch tracking was used for melody annotation. The dataset showcases various music genres, such as Singer/Songwriter, Classical, Rock, World/Folk, Fusion, Jazz, Pop, Musical Theatre, and Rap. The MedleyDB dataset is publicly available [5].

Implementation Details

After the separation step, all datasets are downsampled to 16kHz and transformed into mono samples. Furthermore, we split the MIR1k and the MedleyDB dataset into nonoverlapping training, testing, and validation sets using an 8:1:1 ratio (since the Jamendo Corpus is already split).

In order to perform augmentation, we used the open-source PyTorch-augmentations library [6]. All the training experiments were conducted on the AWS Sagemaker platform [7], with an ml.g4dn.xlarge machine equipped with 1 GPU Nvidia T4. It took around 5 hours to train a model on the original Jamendo training set for 300 epochs. We save network parameters only when the F1-score validation metric exceeds the previous score.

Metrics

To provide a comprehensive view of the results, as proposed in [114], model predictions were compared with the ground truth to obtain the number of false negative (FN), true negative (TN), false positive (FP), and true positive (TP). The frame-wise recall, accuracy, precision, and f1-score were computed to summarize the results.

[3] https://zenodo.org/record/2585988#.YoTKaZNBxhE

[4] https://zenodo.org/record/3532216#.ZFpj8y9Bxf0

[5] https://zenodo.org/record/1715175#.XAzIzxNKjyw

[6] https://pytorch.org/audio/main/tutorials/audio_data_augmentation_tutorial

[7] https://aws.amazon.com/pm/sagemaker

Input Audio	Dataset	Model	Accuracy	Precision	Recall	F1-score
Mixture	Jamendo	RMS	.563	.531	.999	.679
		LRCN	**.868**	**.856**	**.893**	**.864**
		Transformer	.848	.804	.910	.845
Vocals	Jamendo	RMS	.949	.937	.964	.949
		LRCN	**.960**	.945	**.974**	.958
		Transformer	.959	**.953**	.968	**.960**
Vocals	MedleyDB	RMS	.777	.688	.957	.793
		LRCN	**.854**	**.795**	.916	**.849**
		Transformer	.833	.757	**.936**	.833
Vocals	MIR-1K	RMS	.908	.935	.949	.941
		LRCN	.921	**.946**	.955	.949
		Transformer	**.926**	.945	**.960**	**.952**

Table 2.3. Results of the proposed singing voice detection systems trained and tested on the same datasets.

Author	Input Audio	Accuracy	Precision	Recall	F1-score
Schlüter et al. [154]	Mixture	.923	-	.903	-
Lehner et al. [91]	Mixture	.894	.895	.906	.902
Leglaive et al. [90]	Mixture	**.915**	**.895**	**.926**	**.910**
Ours [LRCN]	Mixture	.868	.856	.893	.864
Zhang et al. [216]	Vocals	.924	.926	.924	.927
Ours [LRCN]	Vocals	**.960**	.945	**.974**	.958
Ours [Transformer]	Vocals	.959	**.953**	.968	**.960**

Table 2.4. Results of the proposed singing voice detection system compared with existing methods on the Jamendo Corpus test set.

Results and Discussion

The effects of music source separation and the Demucs zero-shot capabilities in singing voice detection Regarding the effects of music source separation, our experiments provided strong evidence for incorporating music source separation, such as Demucs, in the singing voice detection task. As shown in Table 2.3, when the LRCN and Transformer models were provided with the audio mixture as input, their performance was notably inferior compared to when given the separated vocal signals as input. For example, on the Jamendo dataset, the LRCN model's accuracy increased from 0.868 when using the mixture to 0.960 when using the separated vocals, while the Transformer model's accuracy improved from 0.848 to 0.959 under the same conditions.

These significant performance improvements demonstrate that music source separation is a crucial preprocessing step for enhancing singing voice detection, as asserted by [215]. The models can concentrate on the relevant vocal information by employing Demucs to separate the vocal signals from the audio mixture, while the influence of other audio components, such as background instruments, is minimized. This enables the LRCN and Transformer models to identify and classify the singing voice more accurately and effectively, leading to state-of-the-art results.

Moreover, our investigation of zero-shot capabilities provided valuable insights into the versatility of the Demucs model in the context of singing voice detection.

When used in conjunction with a standard signal processing algorithm, Demucs demonstrated competitive performance in the SVD task, as evidenced by the results presented in Table 2.3. For instance, on the Jamendo dataset, the combination of Demucs and the RMS yielded an accuracy of 0.949, close to the performance of the LRCN (0.960) and Transformer (0.959) models.

These promising results underscore the potential of leveraging pre-trained models, such as Demucs, for tasks beyond their original scope, such as singing voice detection. The ability of Demucs to perform well in the SVD task without any specific training or fine-tuning suggests that its inherent capacity to separate vocals from complex audio mixtures can be effectively utilized across different tasks and applications.

This finding opens up new avenues for future research. It highlights the possibility of harnessing the power of pre-trained models to achieve high-quality performance in various tasks with minimal additional training. Furthermore, it encourages the exploration of transfer learning and multi-task learning techniques to enhance further the adaptability and efficiency of models like Demucs in various audio processing tasks, including singing voice detection.

The potential for deep learning to further improve performance The results presented in Table 2.3 showcase the potential of deep learning models, such as LRCN and Transformer, to significantly improve performance in the singing voice detection task when compared to the baseline RMS.

In our study, we observed that the performance of our models on Jamendo and MIR1k is significantly different from the performance on MedleyDB, even if all three datasets present songs belonging to the same musical genre. This can be attributed to the fact that these datasets have been annotated differently, with the Jamendo and MIR-1K datasets having been annotated manually while MedleyDB has been annotated automatically. This discrepancy in annotation methods may have led to inconsistencies in the data, which could have, in turn, affected the overall learning process of the models. Moreover, when comparing our deep learning models with existing methods on the Jamendo Corpus test set, it becomes evident that the LRCN and Transformer models offer substantial improvements over the current state-of-the-art, as shown in Table 2.4. Specifically, when provided with separated vocal signals as input, our LRCN model achieves an accuracy of 0.960, while the Transformer model reaches 0.959. These results significantly overcome the previous best performance reported by Zhang et al. [216], who achieved an accuracy of 0.924. These performance improvements highlight the effectiveness of Demucs and our deep learning models in the singing voice detection task and their potential applicability to a wide range of music genres and recording conditions. In conclusion, given the results we obtained, we note that the performance of the Transformer is in line with that of the LRCN, so we believe that the Transformer has the potential to perform very well in this task and, therefore, there is a need for a more in-depth study, also supported by testing on other data.

2.2.5 Conclusion

In this chapter, we presented a comprehensive study on singing voice detection, focusing on the impact of music source separation and the potential of deep learning models for improving performance in this task. Our experiments were designed to investigate two main aspects: *(i)* the impact of a music source separation model, such as Demucs, and its zero-shot capabilities for the SVD task; *(ii)* the potential for deep learning to improve performance further. Our results demonstrated that incorporating music source separation with Demucs significantly improved the performance of the LRCN and Transformer models compared to using the audio mixture directly. This finding established the importance of music source separation as a crucial preprocessing step for enhancing singing voice detection. Moreover, our investigation of Demucs' zero-shot capabilities revealed its potential for leveraging pre-trained models in tasks beyond their original scope, such as singing voice detection. Lastly, our deep learning models, LRCN and Transformer, outperformed the baseline RMS and the state-of-the-art methods on the Jamendo Corpus. Based on these findings, future research efforts should address the challenges of diverse dataset annotations, refine data preprocessing techniques, and explore alternative annotation methods to improve further the models' ability to generalize across various musical contexts. Further investigation into the potential of zero-shot and transfer learning for singing voice detection could lead to more accurate and robust models.

2.3 Chapter Conclusions

This chapter delved deep into deep learning, focusing on its applications in audio processing, especially the separation of sound sources. Starting from the history of traditional methods in music source separation, we journeyed through the evolution brought about by introducing deep learning techniques. Key architectures such as Demucs and Conv-Tasnet were explored in detail, highlighting their remarkable potential in audio source separation tasks.

Our first study provided a novel approach to separating fish vocalizations from underwater soundscapes using synthetic soundscapes and two aforementioned sound separation networks. The positive results suggest the practicality of applying deep learning models for enhancing passive acoustic monitoring, thus augmenting the efforts towards effective biodiversity monitoring in vast marine environments. This has significant implications for conservationists, marine biologists, and environmentalists aiming to understand better and protect marine ecosystems.

Our subsequent venture into singing voice detection (SVD) demonstrated the synergy of combining deep learning models to achieve improved performance in this task. By integrating a music source separator, Demucs, with downstream models such as LRCN and Transformer networks, we showcased the power and flexibility of deep learning in tackling complex MIR tasks. The comparative results on multiple datasets reaffirmed the promise of our proposed approach in comparison with traditional methods and current state-of-the-art models. To wrap up, the contributions of this chapter underscore the transformative capabilities of deep learning in audio tasks. As we progress, it will be intriguing to witness how further advancements in deep learning can shape the future of audio processing and analysis.

Chapter 3

Autoregressive Models for Source Separation

Autoregressive models have emerged as a potent mechanism for sequence prediction in various fields, particularly in natural language processing and time series forecasting. These models predict a given output sequence based on its preceding values, incorporating a sequential dependency that leverages historical information to make more accurate forecasts [7]. One of the significant advancements in autoregressive modelling within deep learning is the advent of the Transformer architecture [195].

The Transformer model, proposed by Vaswani et al. (2017) [195] in the seminal paper *Attention is All You Need*, departs from the recurrent and convolutional architectures traditionally used in sequence modelling and introduces a novel self-attention mechanism. The self-attention mechanism enables the model to weigh the importance of different parts of the input sequence when making a prediction, allowing long-range dependencies within the data to be captured more effectively. The Transformer architecture has since become a cornerstone for many state-of-the-art models in natural language processing tasks such as language translation, summarization, and text generation [159, 36, 213].

The architecture is comprised of two primary components: the encoder and the decoder. Each component consists of multiple layers of self-attention and feed-forward neural networks. The encoder processes the input sequence and creates a high-dimensional representation, which the decoder then uses to generate the output sequence autoregressively. This mechanism allows the Transformer to handle sequences of varying lengths and types, making it a flexible and powerful tool for many sequence modelling tasks.

The Transformer's scalability and ability to model long-range dependencies have led to its adoption and extension in various domains. Notable variants and extensions of the Transformer architecture include the BERT [75], GPT-3 [9], and T5 [134], which have pushed the boundaries of what exists possible in natural language processing and other fields. The Transformer's influence extends beyond text, finding applications in image and video processing, where models like Vision Transformer [31] and Video Transformer [101] have showcased their versatility. Furthermore, the Transformer architecture has facilitated research into parallel processing of sequences, significantly reducing training times and enabling the training of larger models on larger datasets. Its modular design has also encouraged a vibrant research community to explore various modifications and improvements, continually expanding the boundaries of what autoregressive models can achieve in deep learning.

The autoregressive nature of the Transformer, where each element of the output sequence is generated one at a time conditioned on both the input sequence and the previously generated elements, has proven to be a powerful paradigm. It effectively captures sequential dependencies, which is crucial in many real-world tasks. Through this, the Transformer architecture has significantly advanced the field of deep learning, offering a robust and efficient framework for sequence modelling and prediction.

Initially developed for text-based tasks, the Transformer architecture has exhibited remarkable adaptability, finding utility in various domains, including audio processing. One notable application of Transformers in the audio domain is OpenAI's Jukebox, a generative model capable of creating music, including melody, rhythm, and even vocals [25]. Jukebox utilizes a hierarchy of Transformers to generate audio in a coarse-to-fine manner. At the highest level, a Transformer model outlines the macro-structure of a piece, deciding the general style and thematic elements. Subsequent, finer-grained Transformers fill in the details, adding layers of complexity to the audio. This hierarchical approach allows Jukebox to handle the high-dimensional audio data space efficiently and generate novel and coherent musical pieces spanning minutes. The autoregressive nature of the Transformer, which inherently operates sequentially, aligns well with the temporal structure of audio data. Each sample in the generated audio sequence is conditioned on the preceding samples, allowing the model to capture the temporal dependencies crucial for producing coherent musical or spoken sequences. Furthermore, the Transformer's self-attention mechanism is particularly useful in audio generation tasks, as it enables the model to weigh the relevance of different parts of the audio sequence when generating each new sample. This ability to account for long-range dependencies is crucial for modelling the structure and coherence of musical pieces.

Autoregressive models, particularly the Transformers, have substantially contributed to the advancements in deep learning and, with their novel self-attention mechanism and scalable design, have not only achieved state-of-the-art performance across a variety of tasks but also fostered a rich vein of research exploring further innovations in autoregressive modeling.

In the following two sections, we shall delve into two endeavors focused on signal separation, explicitly about audio and imagery. These works employ autoregressive models, notably the Transformer architecture, to navigate this task adeptly.

3.1 Unsupervised source separation via bayesian inference in the latent domain

State-of-the-art audio source separation models rely on supervised data-driven approaches, which can be expensive in terms of labeling resources. On the other hand, approaches for training these models without any direct supervision are typically high-demanding in terms of memory and time requirements, and remain impractical to be used at inference time. We aim to tackle these limitations by proposing a simple yet effective unsupervised separation algorithm, which operates directly on a latent representation of time-domain signals. Our algorithm relies on deep Bayesian priors in the form of pre-trained autoregressive networks to model the probability distributions of each source. We leverage the low cardinality of the discrete latent space, trained with a novel loss term imposing a precise arithmetic structure on it, to perform exact Bayesian inference without relying on an approximation strategy. We validate our approach on the Slakh dataset [110], demonstrating results in line with state of the art supervised approaches while requiring fewer resources with respect to other unsupervised methods.

3.1.1 Introduction

Generative models have reached promising results in a wide range of domains, including audio, and can be used to solve different tasks in unsupervised learning. A relevant problem in the musical domain is the task of source separation of different instruments. Given the sequential nature of music and the high variability of rhythm, timbre and melody, autoregressive models [85] represent a popular and effective choice to process data on such domain, showcasing high multi-modality in the modeled probability distributions. The widely adopted WaveNet autoregressive architecture [189] works in the temporal domain. Given that audio signals are typically sampled at high frequencies (e.g. 44 kHz) for music, the choice of modeling the data distribution directly in the time domain leads to short contexts being captured by neural computations and quick saturation of memory. Nevertheless, existing unsupervised approaches for source separation operate in the time domain [69]. In order to capture longer contexts and to reduce memory burden, different quantization schemes have been introduced for autoregressive models [192, 140], where chunks in time are mapped to sequences of latent tokens belonging to a small vocabulary. OpenAI's Jukebox [26] follows this approach and excels as an architecture that can capture very long contexts, generating highly consistent tracks. Leveraging the useful properties of this architecture, we propose a novel approach to unsupervised source separation that works directly on quantized latent domains.

Our contributions can be summarized as follows:

1. We perform source separation applying exact Bayesian inference directly in the latent domain, exploiting the relative small size of the latent dictionary. We do not rely on any approximation strategy, such as variational inference or Langevin dynamics.

2. We introduce LQ-VAE: a quantized autoencoder trained with a novel loss that imposes an algebraic structure on the discrete latent space. This allows us to alleviate noisy and distorted samples which arise from a vanilla quantization approach.

3.1.2 Related Works

The problem of source separation has been classically tackled in an unsupervised fashion [18], where the sources to be separated from a mixture signal are unknown [163]. With the advent of deep learning, most source separation tasks applied to musical data started relying on supervised learning, training models on data with known correspondence between sources. Recently, following the success of deep generative models, there has been a renewed interest in unsupervised methods.

Supervised source separation

Supervised source separation aims to map high dimensional observations of audio mixtures to a smaller dimensional space and apply, explicitly or implicitly, a mask to filter out the sources from the latent representation of the mixtures in a supervised way. Most of these works can be divided into *frequency-domain* or *waveform-domain* approaches. The former [148] operate on the spectral representation of the input mixtures. This line of works has highly benefited from the incoming of deep learning techniques from simple fully connected networks [186], LSTM [188], and CNN coupled with recurrent approaches [97, 3]. Recent approaches such as [15] and [181] hold the state of the art in music source separation over the dataset MUSDB18 [135], by respectively extending the conditional U-net architecture of [115] to multi-source separation, and by exploiting multi-dilated convolution that applies different dilation factors in each layer to model different resolutions simultaneously. In contrast, waveform domain approaches process the mixtures directly in the time domain to overcome phase estimation, which is necessary when converting the signal from the frequency domain. The method of [34] performs in line with the state of the art by extending a WaveNet-like architecture, coupled with an LSTM in the latent space.

The main limitation of these state-of-the-art methods for audio source separation is that they require large amounts of fully separated, labeled data to perform the training.

Unsupervised source separation

Recent approaches in unsupervised source separation leverage self-supervised learning. A prominent baseline is MixIt [201], which trains a model by trying to separate sources from a mixture of mixtures. Although promising, such model suffers from the *over-separation* problem, where at test time a number of sources that is greater than those present in the mixture are estimated. As such, stems can be split across different output tracks. Generative approaches instead overcome this problem by imposing that a model should output an individual stem.

Closer to our work, [118] proposes to leverage generative priors in the form of GANs trained on individual sources. They use projected gradient descent optimization to search in the source-specific latent spaces and effectively recover the constituent sources in the time domain. Although promising, GANs suffer from modal collapse, so their performance is limited in the musical domain, where variability is abundant. [69] proposes to use Langevin dynamics on the global log-likelihood of the audio sequences to parallelize the sampling procedure of autoregressive models used as Bayesian priors. This approach produces good results but with a high computational cost due to the need of training distinct models for each noise level, and due to the costly optimization procedure in the time domain.

Differently, our inference procedure has much lower computational and memory requirements, allowing us to efficiently run the model on a single GPU. In addition, we can perform exact Bayesian inference without relying on an approximation scheme of the posterior (e.g., its score).

3.1.3 Method

In this section we briefly introduce the background concepts necessary to understand our architecture, which builds upon [26]. The overall architecture can be split into two parts: (i) a quantization module mapping the input sequences to a discrete latent space, and (ii) an autoregressive prior (one per source) which models the distribution of a given source in the discrete latent space. We point the reader to [26] for a deeper understanding.

Quantization module

Let us consider an input sequence $\mathbf{x} = x_1, \dots, x_T \in [-1, 1]^T$ of length T, which represents a normalized waveform in the time domain. In order to be representative of an expressive portion of the audio sequence, T should be large. However, due to the complexity of modern neural architectures, choosing a large enough value of T is not always feasible. To reduce the dimensionality of the space one can leverage the VQ-VAE architecture [192] to map large continuous sequences in the time domain to smaller sequences in a discrete latent domain. A VQ-VAE is composed of three blocks:

- A convolutional encoder $E : [-1,1]^T \to \mathbb{R}^{S \times D}$, with $S \ll T$, where S is the length of the latent sequence and D denotes the number of channels;
- A bottleneck block $B = B_I \circ B_Q$ where $B_Q : \mathbb{R}^{S \times D} \to \mathcal{C}^S \subseteq \mathbb{R}^{S \times D}$ is a vector quantizer, mapping the sequence of latent vectors $\mathbf{h} = \mathbf{h}_1, \dots, \mathbf{h}_S = E(\mathbf{x})$ into the sequence of nearest neighbors contained in a codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^K$ of learned latent codes, and $B_I : \mathcal{C}^S \to [K]^S$ is an indexer mapping the codes $\mathbf{e}_{k_1}, \dots, \mathbf{e}_{k_S}$ into the associated codebook indices $z_1 = k_1, \dots, z_S = k_S$. Note that since B_I is bijective, the codes $\mathbf{e}_k$ and their indices k are semantically equivalent, but we shall use the term 'codes' for the vectors in $\mathcal{C}$ and 'latent indices' for the associated integers;
- A decoder $D : [K]^S \to [-1,1]^T$ mapping the discrete sequence back into the time domain.

The VQ-VAE is trained by minimizing the composite loss:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{codebook}} + \beta \mathcal{L}_{\text{commit}} \,, \tag{3.1}$$

where:

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_t \| x_t - D(z_t) \|_2^2 \tag{3.2}$$

$$\mathcal{L}_{\text{codebook}} = \frac{1}{S} \sum_s \| \text{sg}[\mathbf{h}_s] - \mathbf{e}_{z_s} \|_2^2 \tag{3.3}$$

$$\mathcal{L}_{\text{commit}} = \frac{1}{S} \sum_s \| \mathbf{h}_s - \text{sg}[\mathbf{e}_{z_s}] \|_2^2 \,, \tag{3.4}$$

where sg is the stop-gradient operator and β is the commitment loss weight. The losses $\mathcal{L}_{\text{codebook}}$ and $\mathcal{L}_{\text{commit}}$ update the entries of the codebook $\mathcal{C}$ during the training procedure. In addition, we introduce a novel loss term $\mathcal{L}_{\text{lin}}$, described in Section 3.1.3, which imposes a precise algebraic structure on the latent space, facilitating the task of source separation.

Latent autoregressive priors

Once the VQ-VAE is trained, time domain data $\mathbf{x} \sim p^{\text{data}}$ can be mapped to latent sequences $\mathbf{z}$. Autoregressive priors $p(\mathbf{z}) = p(z_1)p(z_2|z_1)\dots p(z_S|z_{S-1}, \dots, z_1)$ can then be learned over the discrete domain. In this work, the autoregressive models are based on a deep scalable Transformer architecture as in [26]. In order to generate new time-domain examples, sequences of latent indices are sampled from $p(\mathbf{z})$ via ancestral sampling and then mapped back to the time domain via the decoder of the VQ-VAE.

The proposed algorithm is composed of two parts. A first *separation phase* in the latent domain, in which we sequentially sample from an exact posterior on discrete indices. A following *rejection sampling procedure* based on a (scaled) global posterior conditioned on the separation results, which we use to sort the proposed solutions and select the most promising one.

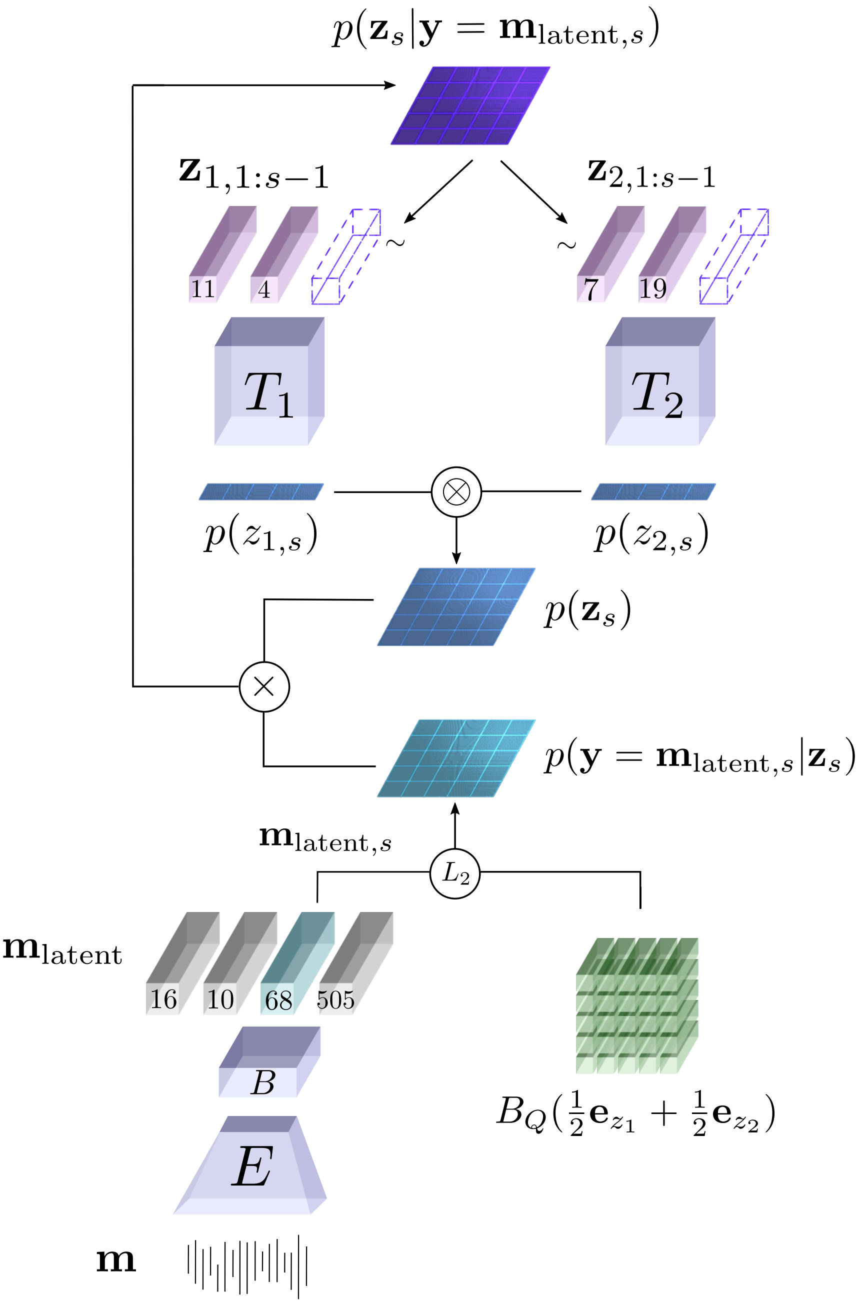

$p(\mathbf{z}_s|\mathbf{y} = \mathbf{m}_{\text{latent},s})$
$\mathbf{z}_{1,1:s-1}$
$\mathbf{z}_{2,1:s-1}$
$\sim$
$\sim$
11
4
7
19
T_1
T_2
$p(z_{1,s})$
$p(z_{2,s})$
$p(\mathbf{z}_s)$
$p(\mathbf{y} = \mathbf{m}_{\text{latent},s}|\mathbf{z}_s)$
$\mathbf{m}_{\text{latent},s}$
L_2
$\mathbf{m}_{\text{latent}}$
16
10
68
505
B
E
$B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2})$
$\mathbf{m}$

Figure 3.1. In our method, two autoregressive priors T_1 and T_2 are trained on different instrument sources in the latent domain. At each step s they provide the joint prior $p(\mathbf{z}_s)$. The prior is combined with a σ-isotropic Gaussian likelihood $p(y = \mathbf{m}_{\text{latent},s}|\mathbf{z}_s) = \mathcal{N}\left(\mathbf{m}_{\text{latent},s}|B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2}), \sigma^2\mathbf{I}\right)$ in order to compute the posterior $p(\mathbf{z}_s|y = \mathbf{m}_{\text{latent},s})$ from which new samples are drawn.

Latent Bayesian source separation

Our task is to separate a mixture signal $\mathbf{m} = \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2$ into $\mathbf{x}_1 \sim p_1^{\text{data}}$ and $\mathbf{x}_2 \sim p_2^{\text{data}}$, where p_1^{data} and p_2^{data} represent the distributions of each instrument class

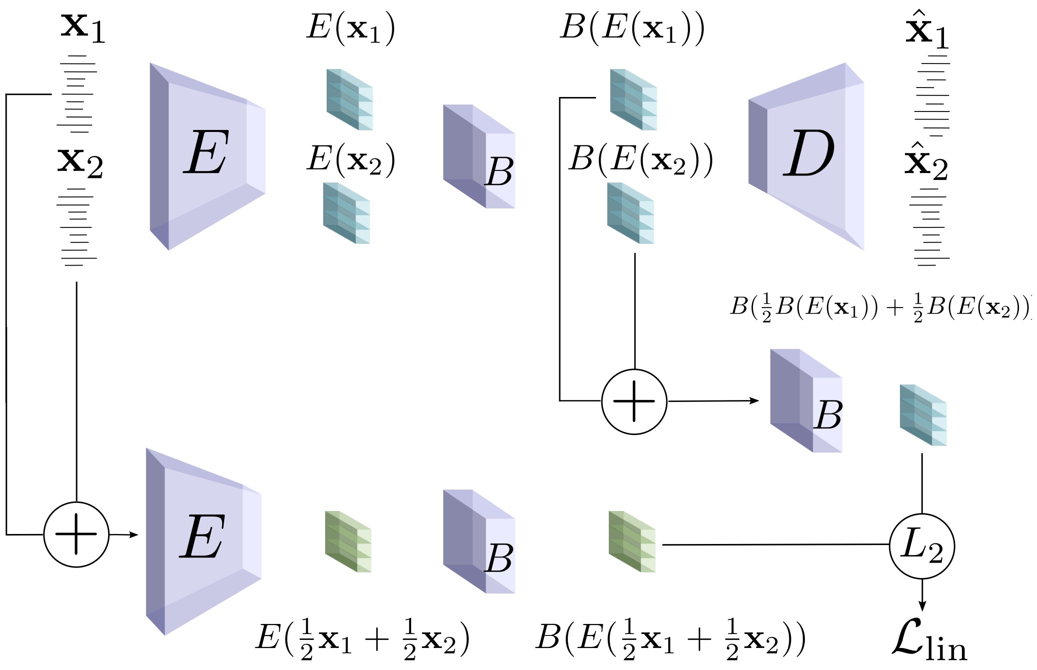

$\mathbf{x}_1$
$E(\mathbf{x}_1)$
$B(E(\mathbf{x}_1))$
$\hat{\mathbf{x}}_1$
$\mathbf{x}_2$
E
$E(\mathbf{x}_2)$
B
$B(E(\mathbf{x}_2))$
D
$\hat{\mathbf{x}}_2$
$B(\frac{1}{2}B(E(\mathbf{x}_1)) + \frac{1}{2}B(E(\mathbf{x}_2))$
B
E
B
L_2
$E(\frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2)$
$B(E(\frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2))$
$\mathcal{L}_{\text{lin}}$

Figure 3.2. Training scheme of the LQ-VAE: reconstructions $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$ are obtained from input pairs $\mathbf{x}_1, \mathbf{x}_2$ as in the VQ-VAE, leading to the loss $\mathcal{L}_{\text{VQ-VAE}}$ (Eq. (3.1)). To this loss we add the post-quantization linearization loss $\mathcal{L}_{\text{lin}}$ (Eq. (3.8)), that is computed by matching time-domain sums with latent vector sums.

in the time domain. In a Bayesian framework, a candidate solution $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2$ is distributed according to the posterior $p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{m}) \propto p_1^{\text{model}}(\mathbf{x}_1) p_2^{\text{model}}(\mathbf{x}_2) p(\mathbf{m} | \mathbf{x}_1, \mathbf{x}_2)$, where the priors p_1^{model}, p_2^{model} are typically deep generative models and the likelihood $p(\mathbf{m} | \mathbf{x}_1, \mathbf{x}_2)$ is parameterized as $p(\mathbf{m} | \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2)$.

In this chapter, we follow the Bayesian approach but we work in the latent domain. After training the VQ-VAE on an *arbitrary* audio dataset (with samples lying also outside p_1^{data} and p_2^{data}), we learn two latent autoregressive priors $p_1(\mathbf{z}_1)$ and $p_2(\mathbf{z}_2)$ over the two instrument classes. The priors do not require any correspondence between the sources, being trained in a completely unsupervised setting. We assume the two priors to be independent, i.e. $p(\mathbf{z}) = p(\mathbf{z}_1, \mathbf{z}_2) = p_1(\mathbf{z}_1) p_2(\mathbf{z}_2)$. Therefore, for each step $s \in [S]$, we can compute the posterior distribution $p(z_{1,s}, z_{2,s} | \mathbf{z}_{1:s-1}, \mathbf{y}) \propto p_1(z_{1,s} | \mathbf{z}_{1,1:s-1}) p_2(z_{2,s} | \mathbf{z}_{2,1:s-1}) p(\mathbf{y} | z_{1,s}, z_{2,s}, \mathbf{z}_{1:s-1})$.

The random variable $\mathbf{y} = f(\mathbf{m})$ is a function of the mixture $\mathbf{m}$. One can choose to model $\mathbf{y}$ in multiple ways; a naive approach is to choose f as the identity and set $\mathbf{y} = \mathbf{m}$, thus computing the likelihood function directly in the time domain. This approach, however, requires the decoding of at least $2K$ possible latent indices in order to locally compare the mixture $\mathbf{m}$ with the hypotheses $z_{1,s}$ and $z_{2,s}$. Note that this corresponds to a lower bound, given that the convolutional nature of the decoder requires a larger past context to produce meaningful results. Differently, we propose to define $\mathbf{y}$ in the latent domain, setting $\mathbf{y} = B_Q(E(\mathbf{m})) := \mathbf{m}_{\text{latent}}$. This approach is preferable since it does not require decoding the hypotheses at each step s, resulting in lower memory usage and computation time. Our method benefits from the choice of operating in the latent space, thanks to the relatively small size of the priors and the likelihood function domain (we choose $K = 2048$, as in [26]). In addition, by exploiting the Transformer architecture, the prior distributions can be computed in parallel. For these reasons, evaluating and sampling from $p(z_{1,s}, z_{2,s}|\mathbf{z}_{1:s-1}, \mathbf{y})$ at each s is computationally feasible and has $O(K^2)$ memory complexity. See Figure 3.1 for a visual description of the inference algorithm.

Latent likelihood via LQ-VAE

In this section we describe how we model the likelihood function and introduce the LQ-VAE model. Following [68] we chose a σ-isotropic Gaussian likelihood, setting:

$$\begin{aligned} p\left(\mathbf{m}_{\text{latent}}|z_{1,s}, z_{2,s}, \mathbf{z}_{1:s-1}\right) &= \\ &= p\left(\mathbf{m}_{\text{latent},s}|z_1, z_2\right) \\ &= \mathcal{N}\left(\mathbf{m}_{\text{latent},s}|B_Q(\tfrac{1}{2}\mathbf{e}_{z_1} + \tfrac{1}{2}\mathbf{e}_{z_2}), \sigma^2\mathbf{I}\right) . \end{aligned} \tag{3.5}$$

The hyper-parameter σ balances the trade-off between the likelihood and the priors. Lower values promote the likelihood: the separated tracks combine perfectly with $\mathbf{m}$, but may not sound like the instrument of the class they belong to. Instead, higher values of σ give importance to the priors: the separated tracks contain only sounds from the corresponding source distribution, but may not mix back to $\mathbf{m}$ (not resembling the sources). The logarithm of the likelihood is:

$$-\frac{1}{2\sigma^2}\left\|\mathbf{m}_{\text{latent},s} - B_Q\left(\tfrac{1}{2}\mathbf{e}_{z_1} + \tfrac{1}{2}\mathbf{e}_{z_2}\right)\right\|_2^2 . \tag{3.6}$$

At each step s, we compare a variable term $\mathbf{m}_{\text{latent},s}$ with a constant matrix $B_Q\left(\tfrac{1}{2}\mathbf{e}_{z_1} + \tfrac{1}{2}\mathbf{e}_{z_2}\right)$ representing all possible (scaled) sums over all codes in $\mathcal{C}$. This term can be precomputed once and then reused during inference, saving additional computational resources.

We observed that performing separation with the likelihood in Eq. (3.5) using a VQ-VAE trained with the loss in Eq. (3.1), results in disturbed and noisy outcomes. Such behavior is expected because the standard VQ-VAE does not impose any algebraic structure on the discrete domain; therefore, summing codes as in Eq. (3.5) does not lead to meaningful results. This problem can be lifted by enforcing a post-quantization linearization loss on the VQ-VAE:

$$\mathcal{L} = \mathcal{L}_{\text{VQ-VAE}} + \mathcal{L}_{\text{lin}} , \tag{3.7}$$

where $\mathcal{L}_{\text{VQ-VAE}}$ is defined as in Eq. (3.1) and

$$\mathcal{L}_{\text{lin}} = \frac{1}{T}\sum_t \|LQ_t - QL_t\|_2^2 \tag{3.8}$$

$$QL_t = B_Q\left(\tfrac{1}{2}B_Q\left(E\left(\mathbf{x}_{1,t}\right)\right) + \tfrac{1}{2}B_Q\left(E\left(\mathbf{x}_{2,t}\right)\right)\right) \tag{3.9}$$

$$LQ_t = B_Q\left(E\left(\tfrac{1}{2}\mathbf{x}_{1,t} + \tfrac{1}{2}\mathbf{x}_{2,t}\right)\right) . \tag{3.10}$$

Minimizing this loss pushes the quantized latent code representing a mixture of two arbitrary source signals (LQ_t term) to be equal to the sum of the quantized latent codes, corresponding to the single sources (QL_t term), therefore enforcing the discrete codes to behave in an approximately linear way. We shall refer to the VQ-VAE trained as above, as a *Linearly Quantized Variational Autoencoder* (LQ-VAE). See Figure 3.2 for a visual illustration of the LQ-VAE training procedure.

Method	Drums	Bass	Drums	Guitar	Guitar	Bass
Ours (best)	**5.83**	**7.42**	**8.33**	3.80	3.75	**8.65**
Ours (rej)	4.08	5.31	6.93	2.48	1.95	6.35
Demucs†	5.42	5.36	5.80	5.36	6.42	7.68
TasNet†	5.51	5.43	5.87	**5.47**	**7.80**	8.46
rPCA[59]	0.60	1.05	2.27	-0.42	0.52	-1.12
ICA[64]	-0.99	-1.53	-0.53	-3.23	-0.73	-2.79
HPSS [41]	-0.56	-0.33	0.31	-2.72	0.15	-0.38
REPET[136]	0.53	1.54	2.91	0.11	0.40	-1.09
FT2D [156]	0.59	1.31	2.63	-0.15	0.65	-1.02

Table 3.1. SDR scores evaluated on Slakh2100 test set. All methods are unsupervised except those marked with †. The rej attribute indicates that the solutions were obtained by the rejection sampling procedure with $\alpha = 0$. The scores are computed according to the implementation in [172]

.

Rejection α	Drums	Bass	Drums	Guitar	Guitar	Bass
0	**4.08**	**5.31**	**6.93**	**2.48**	**1.95**	**6.35**
0.5	3.61	4.78	6.69	2.17	1.68	6.00
1	2.94	4.03	6.44	1.95	1.15	5.35

Table 3.2. Ablation study for rejection parameter α.

Method	Drums	Piano
Ours	**0.68**	**3.66**
Ours (rejection $\alpha = 0$)	0.08	2.75
GAN [20]	-3.16	-2.26

Table 3.3. SDR table evaluated on the test set of [118].

Rejection sampling

Given the low memory requirements of our method, at inference time we can sample in parallel multiple solutions $\{\mathbf{z}^{(b)}\}_{b=1}^{B}$ in the same batch. Autoregressive models tend to accumulate errors over the course of ancestral sampling, therefore the quality of the solutions varies across the batch. In order to select a solution, we look at the posterior $p_{\text{rej}}(\mathbf{z}|\mathbf{m}) \propto p_{\text{rej},1}(\mathbf{z}_1)p_{\text{rej},2}(\mathbf{z}_2)p_{\text{rej}}(\mathbf{m}|\mathbf{z})$, conditioned by the sampling event. We obtain the priors $p_{\text{rej},1}$ and $p_{\text{rej},2}$ by normalizing p_1 and p_2 over the batch (computed by integrating over s during the inference). For numerical stability, we scale their logits by the length of the latent sequences S. The likelihood function $p_{\text{rej}}(\mathbf{z}|\mathbf{m}) = \mathcal{N}\left(\mathbf{m}|\frac{1}{2}D(\mathbf{z}_1) + \frac{1}{2}D(\mathbf{z}_2), \sigma_{\text{rej}}^2\mathbf{I}\right)$ is computed directly in the time domain, with the decoding pass being executed only once at the end of the sampling procedure. The hyper-parameter σ_{rej} plays a similar role to the σ used in Eq. (3.5). We can balance the likelihood and the priors by setting:

$$\mathbb{E}_b\Big[\log p_{\text{rej}}(\mathbf{z}^{(b)})\Big] = -\frac{1}{2\sigma_{\text{rej}}^2}\mathbb{E}_b\Big[\Big\|\mathbf{m} - \tfrac{1}{2}(D(\mathbf{z}_1^{(b)}) + D(\mathbf{z}_2^{(b)}))\Big\|_2^2\Big]$$

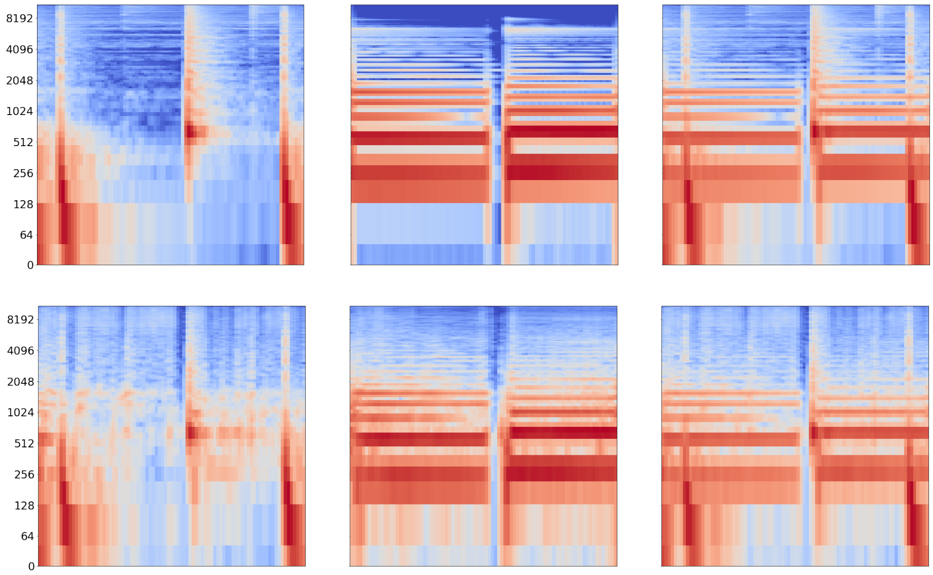


Figure 3.3. Mel spectrograms in log scale of a separation result (bottom row) and the corresponding ground truth signals (top row). Left: drums source. Center: bass source. Right: mixture.

and solving for σ_{rej}. Albeit natural, this framework does not lead to the best selection. We performed an ablation study by weighting the contribution of the global likelihood with a scalar $\alpha \in [0, 1]$ (using $\sigma'^2_{\text{rej}} = \alpha\sigma^2_{\text{rej}}$) and the best empirical results are obtained when the global likelihood is not taken into account ($\alpha = 0$), see Table 3.2. We call this selection criterion *prior-based rejection sampling*.

3.1.4 Experiments

We validate our approach on *Slakh2100* [110]: a large musical source dataset containing mixed tracks separated into 34 instrument categories. We select tracks from the classes 'drum', 'bass' and 'guitar' coming from the training and test splits, sub-sampled at a frequency of 22kHz. We train the convolutional LQ-VAE over mixtures obtained by randomly mixing sources from the individual tracks of the training set. The LQ-VAE has a downsampling factor of $\frac{T}{S} = 64$ and uses a dictionary of $K = 2048$ latent codes. After training the LQ-VAE, we train two autoregressive models, one per source, on latent codes extracted from ~ 1200 tracks each. In all our separation experiments we fixed $\sigma = 0.1$ in Eq. (3.6).

In Table 3.1 we compare our method with two state-of-the-art supervised approaches and different non-learning based unsupervised methods. To this end, we iterate on the test split of [110] made up of about 150 different songs, and for each we extract 450 random chunks each of 3 seconds. In Figure 3.3 we show a qualitative result of our algorithm.

In order to strengthen our empirical evaluation, we show in Table 3.3 results of our model applied to a different validation data set in order to perform a comparison with the GAN model of [118]. We evaluate both methods over the test dataset proposed in [118], consisting of 1000 mixtures of 1 second each. Each mixture combines a drum sample with a piano track randomly, thus independence in the test data is assumed, resulting in a more artificial setting with respect to the one present in Slakh2100. For [118] we use the pre-trained model given by the authors while for our method we use the “drums” and “piano” priors trained on Slakh2100 thus showing the cross-dataset generalization capability of our model.

All our experiments are performed on a Nvidia RTX 3080 GPU with 16 GB of VRAM. With this GPU our method can sample a batch of 200 candidate solutions (100 for each instrument) simultaneously. The code to reproduce our experiments is available at https://github.com/michelemancusi/LQVAE-separation. Interestingly, even if solutions selected by the rejection sampling algorithm have slightly lower metrics than supervised approaches, by individually selecting the best solution for each instrument we achieve performance in line with the state of the art (especially on ‘bass‘ and ‘drum‘ stems). This testifies the quality of our separation. Remarkably, our method employs 3 minutes on average for sampling a track of 3 seconds, compared to the more than 100 minutes of [69].

3.1.5 Conclusion

In this chapter, we introduced a simple algorithm to perform exact Bayesian inference in the discrete latent domain. Our method allows us to achieve good separation results while being much faster than other likelihood-based unsupervised approaches.

The main bottleneck of our method lies in the rejection sampling strategy. Future work will attempt to improve this aspect by investigating the design of more accurate learning-based rejection samplers. Other benefits could come from the adoption of multi-level VQ-VAEs [26] or by leveraging deeper autoregressive priors.

3.2 Latent Autoregressive Source Separation

Autoregressive models have consistently demonstrated high-quality generation capabilities across various fields. One significant factor that enables this is the deployment of quantized latent spaces, like those derived from VQ-VAE autoencoders. These spaces pave the way for reduced dimensionality and enhanced inference speeds, making them useful in the continuous realm. However, adapting pre-existing trained models for novel complex tasks is often challenging. It may require further fine-tuning or intensive training processes.

This chapter presents LASS, an innovative vector-quantized Latent Autoregressive Source Separation approach. This approach involves decomposing an input into its fundamental sources, bypassing the need for further fine-tuning or model adjustments. Our separation approach leans into a Bayesian model, wherein the autoregressive models act as priors. Furthermore, frequency evaluations of latent addend token aggregates establish a discrete likelihood function.

We tested our method extensively on both images and audio using diverse sampling methodologies, including ancestral and beam search techniques. Our method matches the efficacy of existing approaches in separation. It outshines them with faster inference speeds and adaptability to more complex data dimensions.

3.2.1 Introduction

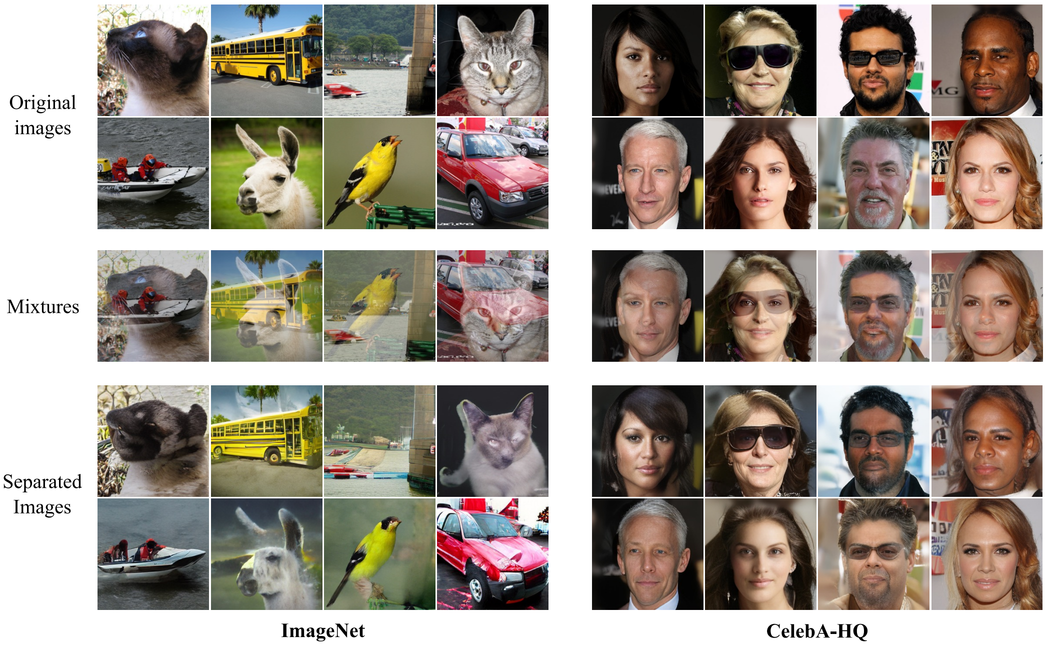


Figure 3.4. 256x256 separations obtained with *LASS* using pre-trained autoregressive models. Left: class-conditional ImageNet. Right: unconditional CelebA-HQ.

Autoregressive models have made notable strides across diverse fields such as natural language processing [9], audio [26], vision [140, 38], and multimodal interfaces [138, 211]. When dealing with dense domains, training these models on discrete latent configurations derived by quantizing continuous data is commonplace, often utilizing VQ-VAE autoencoders [192]. This technique supports the generation of finer samples and expedites inference. The latent configurations crafted in this manner also hold value for subsequent tasks [10]. However, for more sophisticated tasks, the prevalent strategy involves refining the model or using prompt-based scaled training [199, 151]. While the former is frequently adopted, it requires further optimization or model amendments. The latter, especially outside the natural language spectrum, is more intricate [208, 53].

This research delves into the intricate task of source separation, harnessing pre-trained quantized autoregressive models without needing gradient-dependent optimization or structural changes. The realm of audio, especially areas like speech [33], music [22], and universal source separation [200, 131], has seen a surge in the exploration of extracting multiple sources from a composite signal with the advent of deep learning. Though image source separation has yet to garner as much attention as its audio counterpart, it has been documented [51]. Dominant methods in the field either rely on direct supervision for achieving commendable results [104, 34] or tap into large-scale unsupervised regression [202].

Our proposed methodology seeks to employ generative tactics for source separation, anchored on autoregressive prior distributions developed within a latent VQ-VAE environment. Depending on the inclusion of class information, our approach either veers towards weak supervision or fully unsupervised. A distinctive non-parametric sparse likelihood function is crafted by monitoring the frequency of mixed latent tokens compared to source-specific tokens. Tokens are computed by projecting the mixture signals from the data domain and their associated components through the VQ-VAE. This function does not alter the VQ-VAE or the autoregressive priors since the VQ-VAE's latent space remains constant. Combining this likelihood function with autoregressive prior predictions, using Bayes' theorem, results in a posterior distribution. Discrete sampling techniques like ancestral or beam search are deployed to achieve separations. We have labeled this innovative framework as Latent Autoregressive Source Separation (LASS).

To encapsulate, our contributions include:

- The unveiling of LASS, a novel Bayesian framework for source separation, capitalizing on previously trained autoregressive models in quantized latent spaces.
- Empirical validation of LASS in image processing, revealing efficient outcomes on platforms like MNIST and CelebA (32x32), with qualitative insights into ImageNet (256x256) and CelebA-HQ (256x256) datasets, underscoring LASS's scalability to pre-trained models.
- A demonstration of LASS's prowess in the domain of music source separation using the Slakh2100 dataset, where it showcases results rivaling top-tier supervised techniques but with reduced computational demands.

3.2.2 Related Works

In this section, we will briefly recall the concepts seen in the introduction of the second chapter. Traditionally, blind source separation tackled the issue of source separation, employing unsupervised strategies without utilizing any knowledge about the sources embedded within a mixed signal, instead leaning on mathematical principles such as source independence [64] or recurrence [136] for carrying out the separation. With the emergence of deep learning, contemporary strategies for source separation have been mainly segregated into two primary categories: regression-based and generative-based methodologies.

Regression-Based Source Separation

This paradigm involves inputting a mixed signal into a parametric model, commonly a neural network, producing separated sources. The prevalent mode of training involves a supervised methodology, wherein the separated sources produced by the model are compared with actual sources using a regression loss, such as $\mathcal{L}_1$ or $\mathcal{L}_2$ loss [50]. Although it has seen application in image source separation [51], the methodology has predominantly been researched within the audio domain. Here, two main strategies are prevalent:

- Mask-Based Strategy: The model separates by applying computed masks to mixtures, commonly in the STFT domain [148, 186, 60, 119, 97, 3].
- Waveform Strategy: The model directly produces the sources in the time domain, mitigating the need for phase estimation in conversions from the STFT domain to the waveform domain [99, 104, 34].

Generative Source Separation

With the advancements made by deep generative models [46, 77, 55, 170], there is a budding interest in new techniques related to generative source separation. Here, emphasis is laid on harnessing generative models, particularly pre-trained ones, to execute the separation task without needing a specialized architecture. In the initial phases, deep generative separation relied significantly on GANs [175, 80, 118]. After that, Jayaram and Thickstun [68] introduced the generative separation technique BASIS, utilizing score-based models [168] in an image setting (BASIS-NCSN) and a variant of flow-based models with noise annealing (BASIS-Glow). The inference process is carried out in the image domain via Langevin dynamics [123], showing encouraging quantitative and qualitative outcomes. Building on the principles of Langevin dynamics for inference in autoregressive models, the authors introduced a noise scheduling technique, the Parallel and Flexible (PnF) method [70]. While inventive, especially for tasks like inpainting, it does not utilize pre-trained autoregressive models directly and demands fine-tuning under various noise levels. Additionally, due to its operation directly in the data domain, it experiences extensive inference times and struggles to scale to higher resolutions. Continuing this research trajectory, the work presented in this chapter introduces a novel separation method for latent autoregressive models. This method forgoes the need for re-training, offers scalability to pre-trained models of any stage, and seamlessly aligns with traditional discrete samplers.

3.2.3 Method

This section briefly recalls the concepts already seen 3.1.

VQ-VAE Overview

The VQ-VAE [192] offers a mechanism to transform a data sample, denoted as $\mathbf{x} \in \mathbb{R}^N$ (where N represents the overall data sample size, such as an audio sequence's length or a picture's pixel channels count), into a discrete latent space. Utilizing an encoder $E_\theta : \mathbb{R}^N \to \mathbb{R}^{S \times C}$, the data point $\mathbf{x}$ is converted to $E_\theta(\mathbf{x}) = (\mathbf{h}_1, \ldots, \mathbf{h}_S)$. Here, C indicates latent channels, while S signifies the latent sequence's length. A constraining block, denoted by $B : \mathbb{R}^{S \times C} \to [K]^S$, translates this encoding into a discrete sequence, $\mathbf{z} = (z_1, \ldots, z_S)$. It does so by associating each $\mathbf{h}_s$ with the nearest neighboring vector index, termed token, $z_s = B(\mathbf{h}_s)$ present in a learned vector set $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$ in $\mathbb{R}^C$, referred to as codes. Lastly, a decoder $D\psi : [K]^S \to \mathbb{R}^N$ projects the latent sequence back to the original data space, resulting in a reconstruction denoted as $\hat{\mathbf{x}} = D_\psi(\mathbf{z})$. The VQ-GAN [38] is an advanced VQ-VAE variant incorporating a discriminator and a perceptual loss into the training process, facilitating better data reconstruction and increased compression capability. For a comprehensive understanding of VQ-VAE and VQ-GAN, readers can consult [192] and [38]. In subsequent sections, both models will be addressed as VQ-VAE, with distinctions made when appropriate.

Introduction to Autoregressive Models

Autoregressive models provide a means to specify a probability distribution over a discrete space, specifically $[K]^S$, in the context of the VQ-VAE's latent domain. For any sequence $\mathbf{z} = (z_1, \ldots, z_S)$, its joint probability is expressed using the chain rule as:

$$p_\phi(\mathbf{z}) = \prod_{s=1}^{S} p_\phi(z_s | \mathbf{z}_{<s}),$$

In this formulation, $p_\phi(\cdot)$ is a learned model, typically a neural architecture like CNNs [190, 150] or Transformers [195]. During the inference phase, different sampling methods can be employed. A common approach is ancestral sampling. Here, each token z_s is sampled probabilistically from the conditional distribution $p_\phi(z_s|\mathbf{z}_{<s})$. Techniques like top-$k$ [83] might enhance the diversity of the resultant data [57]. For objectives that optimize the entire sequence's probability, heuristic methods, for instance, beam search, are favored [141]. During the inference process, beam search simultaneously evaluates B potential sequence hypotheses $\mathbf{z}^1, \ldots, \mathbf{z}^B$. For every step s, the method calculates the conditional probabilities for each beam and updates the B hypotheses to maximize the joint probabilities $p_\phi(\mathbf{z}^b_{<s})p_\phi(z_s|\mathbf{z}^b_{<s})$.

Consider two sources, denoted as $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2) \in \mathbb{R}^{2\times N}$, which adhere to the distribution $p_{\text{data}} = (p^1_{\text{data}}, p^2_{\text{data}})$. We define an observable combination,

$$\mathbf{y} = \frac{\mathbf{x}^1 + \mathbf{x}^2}{2} \tag{3.11}$$

In the realm of generative source separation, our objective is to deduce the sources $\mathbf{x}$ from the composite $\mathbf{y}$, employing the Bayesian posterior (under the presumption of independent sources):

$$p(\mathbf{x}^1, \mathbf{x}^2|\mathbf{y}) \propto p^1_{\text{data}}(\mathbf{x}^1)p^2_{\text{data}}(\mathbf{x}^2)p(\mathbf{y}|\mathbf{x}^1, \mathbf{x}^2). \tag{3.12}$$

The direct manipulation of Eq. (3.12) within the continuous data sphere poses challenges. We initially approximate p_{data} via autoregressive models in a VQ-VAE's latent dimension to navigate this. Transitioning domains lets us newly define the likelihood function $p(\mathbf{y}|\mathbf{x}^1, \mathbf{x}^2)$, eliminating the need for gradient-based refinement or additional training. The forthcoming subsections address these concerns. We will first delve into the latent autoregressive source separation and subsequently discuss inference methods using *LASS* and a refinement strategy post-inference.

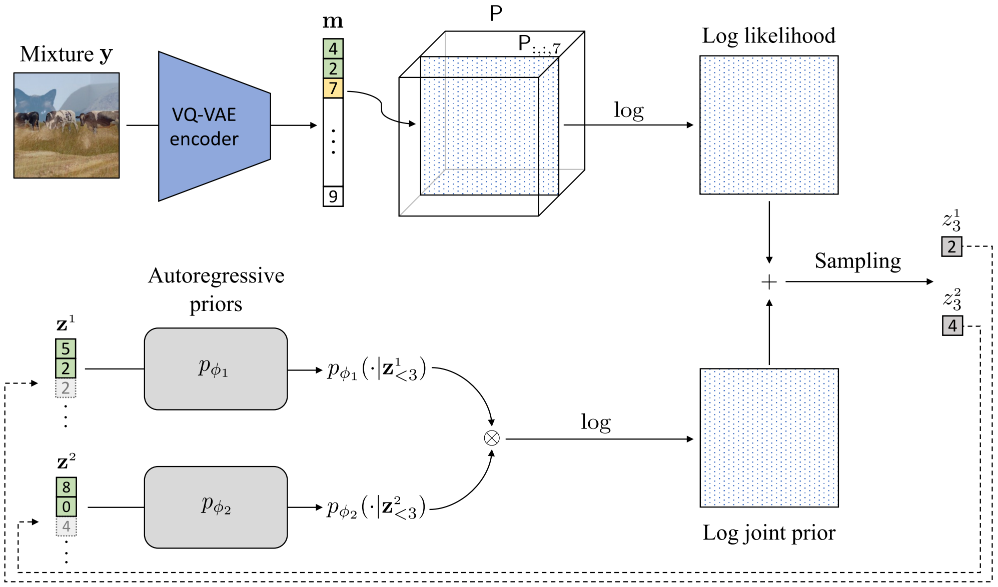


Figure 3.5. Schematic of the *LASS* separation procedure. The picture shows the separation procedure at $s = 3$ and is repeated until $s = S$. At the end of inference, we obtain $\mathbf{x}^1$ and $\mathbf{x}^2$ decoding $\mathbf{z}^1$ and $\mathbf{z}^2$ via the VQ-VAE decoder (not depicted in the picture). We refer the reader to Algorithm 1 for more details.

Latent Autoregressive Source Separation Framework

Our study evaluates situations where p_{data} is approximated using a singular autoregressive model, labeled p_ϕ (for every source, in an unsupervised manner[1]), and cases with two separate models, $p_\phi = (p_{\phi_1}, p_{\phi_2})$. The emphasis will be on the latter scenario since the former can be extrapolated by equating p_{ϕ_1} and p_{ϕ_2}. Let's signify the latent sources and composites as $\mathbf{z} = (\mathbf{z}^1, \mathbf{z}^2) = B(E_\theta(\mathbf{x}))$ and $\mathbf{m} = B(E_\theta(\mathbf{y}))$, respectively. Applying Eq. (3.12), we can express the posterior distribution in the latent space as:

$$p(\mathbf{z}_s|\mathbf{z}_{<s}, \mathbf{m}_{\leq s}) \propto p_\phi(\mathbf{z}_s|\mathbf{z}_{<s})p(\mathbf{m}_{\leq s}|\mathbf{z}_{\leq s}), \tag{3.13}$$

for every $s = 1, \ldots, S$. The initial factor symbolizes the (collective) Bayesian prior, depicted with autoregressive distributions. The latter factor represents the likelihood function. Given that each code in the convolutional VQ-VAE encapsulates a specific data segment and that mixing is data-point specific, the connection between latent codes is equally localized. Consequently, we can simplify the likelihood function in Eq. (3.13) as:

$$p(\mathbf{m}_{\leq s}|\mathbf{z}_{\leq s}) \approx p(m_s|\mathbf{z}_s). \tag{3.14}$$

Being position-independent, we can omit the positional index s:

$$p(m_s|\mathbf{z}_s) = p(m_s|z_s^1, z_s^2) = p(m|z^1, z^2). \tag{3.15}$$

The subsequent subsection will elucidate how *LASS* depicts the likelihood function.

[1] This should not be mixed up with the unsupervised blind scenario. In our version of unsupervised, we access the sources but lack class labels.

Discrete Likelihoods for Source Separation

Prior research in generative source separation [68, 70] has typically focused on modeling likelihood functions within the data domain. This often involves the use of a σ-isotropic Gaussian expression:

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|(\mathbf{x}^1 + \mathbf{x}^2)/2, \sigma^2\mathbf{I}).$$

In the approach we have adopted, it is unfeasible to merge z^1_s with z^2_s (or their associated dense codes) using the standard sum operation. VQ-VAE does not set a clear arithmetic structure for the latent space. We leverage the likelihood function defined in Eq. (3.15) to address this challenge. This is realized through discrete conditionals characterized by rank-3 tensors[2], denoted as $\mathsf{P} \in \mathbb{R}^{K\times K\times K}$:

$$p(\cdot\,|z^1, z^2) = \mathsf{P}_{z^1,z^2,:}.$$

For acquiring P, we employ frequency counting on latent mixed tokens based on latent source tokens. This is done by looping through a dataset X. We begin by initializing a null integer tensor, denoted as $\mathsf{F}^0 \in \mathbb{N}^{K\times K\times K}$. As we progress through $\mathbf{x}^1, \mathbf{x}^2 \in X$, we compute: $\mathbf{y} = (\mathbf{x}^1 + \mathbf{x}^2)/2$, followed by determining the latent sequences and so forth:

$$\begin{aligned}\mathbf{z}^1 &= B(E_\theta(\mathbf{x}^1)),\\ \mathbf{z}^2 &= B(E_\theta(\mathbf{x}^2)),\\ \mathbf{m} &= B(E_\theta(\mathbf{y})).\end{aligned}$$

For every element (z^1_s, z^2_s, m_s) within the sets $\mathbf{z}^1$, $\mathbf{z}^2$, and $\mathbf{m}$ at iteration t, we just augment the prior count by a unit:

$$\begin{aligned}\mathsf{F}^t_{z^1_s,z^2_s,m_s} &= \mathsf{F}^{t-1}_{z^1_s,z^2_s,m_s} + 1\,,\\ \mathsf{F}^t_{z^2_s,z^1_s,m_s} &= \mathsf{F}^{t-1}_{z^2_s,z^1_s,m_s} + 1\,.\end{aligned}$$

To guarantee the sum's commutative property, we interchange the order of the addends. After these computations, P is defined as:

$$\mathsf{P}_{z^1,z^2,:} = \frac{1}{\sum_{k=1}^K \mathsf{F}_{z^1,z^2,k}} \mathsf{F}_{z^1,z^2,:}.$$

At the time of inference, the likelihood function is achieved by segmenting the tensor using m. On the surface, parameter-free conditional distributions might appear inefficient regarding memory usage, given its $O(K^3)$ complexity. However, the tensor P is predominantly sparse in real-world scenarios, as evidenced in Table 3.7.

Utilizing discrete likelihood functions within the VQ-VAE's latent domain presents several advantages: the representation remains unchanged, the learning process is free from hyperparameters, and there is no need for retraining autoregressive priors.

[2] Our tensor notation is consistent with that of Goodfellow, Bengio, and Courville [45].

Dataset	K	Density (%)
MNIST	256	1.49×10^{0}
CelebA	512	6.06×10^{0}
CelebA-HQ	1024	3.80×10^{-1}
ImageNet	16384	3.90×10^{-3}
Slakh (Drum + Bass)	2048	7.60×10^{-2}

Table 3.4. Statistics on likelihood functions over different datasets. K is the number of VQ-VAE (or VQ-GAN) latent codes. Density is the percentage of nonzero elements in the likelihood function.

Algorithm 1 *LASS* inference

```
Input: $\mathbf{y}$
Output: $\mathbf{x}^1, \mathbf{x}^2$
 1: $\mathbf{m} \leftarrow B(E_\theta(\mathbf{y}))$
 2: $\mathbf{z}^1 \leftarrow []$
 3: $\mathbf{z}^2 \leftarrow []$ $s = 1$ to $S$
 4: prior $\leftarrow \log(p_{\phi_1}(\cdot\,|\mathbf{z}^1) \otimes p_{\phi_2}(\cdot\,|\mathbf{z}^2))$
 5: likelihood $\leftarrow \log(\mathsf{P}_{:,:,m_s})$
 6: posterior $\leftarrow$ prior $+\ \lambda$ likelihood
 7: $(z^1_s, z^2_s) \leftarrow$ Sampler(posterior)
 8: $\mathbf{z}^1 \leftarrow$ concat($\mathbf{z}^1, z^1_s$)
 9: $\mathbf{z}^2 \leftarrow$ concat($\mathbf{z}^2, z^2_s$)
10: $\mathbf{x}^1 \leftarrow D_\psi(\mathbf{z}^1)$
11: $\mathbf{x}^2 \leftarrow D_\psi(\mathbf{z}^2)$
12: $\mathbf{x}^1$, $\mathbf{x}^2$
```

Inference Technique

Consider a mixture $\mathbf{y}$, the autoregressive models p_{ϕ_1}, p_{ϕ_2}, and the trained probability tensor P. This setup allows us to deduce and derive $\mathbf{x}^1, \mathbf{x}^2$ as elaborated in Algorithm 1 and illustrated in Figure 3.5.

We begin by translating $\mathbf{y}$ into the latent space to get $\mathbf{m} = B(E_\theta(\mathbf{y}))$. We then set the initial estimates $\mathbf{z}^1, \mathbf{z}^2$ to empty sequences and proceed with the iterations over $s = 1, \ldots, S$.

During each iteration, the joint prior (a $K \times K$ matrix) is determined (Line 5) using the outer product of the two distributions, predicted from the autoregressive models based on the previous context. For the sake of numerical consistency, we employ logarithms of these distributions. Subsequently, the log-likelihood function is determined (Line 6) by taking the logarithm of $\mathsf{P}_{:,:,m_s}$. In our studies, varying scaling factors λ can be utilized on the log-likelihood to equate it with the priors. These two matrices are merged to establish the posterior in Line 7.

In the concluding steps (Lines 8-10), diverse strategies can be utilized to select the optimal candidate tokens (z^1_s, z^2_s) from the resulting posterior. For our studies, we employed ancestral selection (with or without top-k filters) and beam search. Once the deductive cycle concludes, the derived sequences are translated back into the data domain using the VQ-VAE's decoder (Lines 12-13), resulting in $\mathbf{x}^1$ and $\mathbf{x}^2$.

Refinement After Inference The resolution of the distinguished images is determined by the clarity of images attained through the VQ-VAE decoding process.

To augment these distinctions, an extra refinement process can be employed. This involves the iterative enhancement of the VQ-VAE's latent descriptions of the samples:

$$\mathbf{e}^1_{t+1} = \mathbf{e}^1_t + \alpha \nabla_{\mathbf{e}^1_t} \| D_\psi(\mathbf{e}^1_t) + D_\psi(\mathbf{e}^2_t) - 2\mathbf{y} \|_2 \tag{3.16}$$

$$\mathbf{e}^2_{t+1} = \mathbf{e}^2_t + \alpha \nabla_{\mathbf{e}^2_t} \| D_\psi(\mathbf{e}^1_t) + D_\psi(\mathbf{e}^2_t) - 2\mathbf{y} \|_2 \tag{3.17}$$

for $t = 1, \ldots, T-1$ and with $\mathbf{e}^1_1 = E_\theta(\mathbf{x}^1)$, $\mathbf{e}^2_1 = E_\theta(\mathbf{x}^2)$. This step refines dense latent vectors to ensure their decodings aptly sum up the mix, using the results from Algorithm 1 as an initial point. This approach showed considerable improvements on the MNIST dataset, where we gauge the distinction's accuracy with a pixel-specific metric (PSNR), especially as the VQ-VAE tends to generate more refined images.

3.2.4 Experiments

The effectiveness and scalability of *LASS* is demonstrated through qualitative and quantitative examination across various datasets. Within the domain of imagery, evaluations are performed on MNIST [87] and CelebA (32×32) [98], with qualitative results shared for higher resolution databases like CelebA-HQ (256×256) [72] and ImageNet (256×256) [24].

Slakh2100 [110], a vast music source separation dataset ideal for generative models, serves as our testing ground for the auditory domain. All our tests were executed on a single Nvidia RTX 3090 GPU boasting 24 GB of VRAM. Implementation details are shown in Table 3.5 and 3.6

Image Source Separation

The Transformer framework [195] is selected as the autoregressive for all image source separation experiments foundation. Employing MNIST and CelebA, we initially train a VQ-VAE and subsequently train the autoregressive Transformer in its latent realm. For MNIST, we apply $K = 256$ codes, and due to the increased variability in CelebA, we utilize $K = 512$ codes. When it comes to CelebA-HQ and ImageNet, pre-trained VQ-GANs [38] are combined with the already available Transformers from the authors[3] (`celebahq_transformer` checkpoint for CelebA-HQ and `cin_transformer` for ImageNet). Given *LASS* 's adaptability, it is incorporated into the separation algorithm without alterations. For CelebA-HQ, the VQ-GAN is designed with $K = 1024$ codes, while ImageNet uses $K = 16384$ codes. Regarding the experiments about the images, we initially derive the P tensor based on the methodology outlined in the "Method" section. Table 3.7 indicates that CelebA has the least sparsity (maximum density) and ImageNet the most. In every scenario, the density remains under 7%, ensuring the inference process remains unaffected by memory constraints.

Quantitative Results In order to determine the quality of image separations by *LASS*, we juxtapose our approach against various benchmarks on MNIST and CelebA. For MNIST, we set *LASS* against results noted for generative separation techniques like "BASIS NCSN" (score-based) and "BASIS Glow" (noise-annealed flow-based) from [68], the GAN-oriented "S-D" strategy [80], the fully supervised version of Neural Egg "NES", and the "Average" baseline where separations are

[3] `github.com/CompVis/taming-transformers`

Hyperparameter	MNIST	SLAKH	CelebA
Transformer Priors			
Number of Layers	3	48	12
Hidden Size	128	1024	832
Embedding Size	128	1024	832
Vocabulary Dim (K)	256	2048	512
Attention Heads	2	1	8
Context Tokens (S)	49	8192	64
Learning Rate	2e-4	3e-4	3e-4
Adam ϵ	1e-6	1e-6	1e-6
Adam β_1	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.999
VQ-VAE			
Vocabulary Dim (K)	256	2048	512
Embedding size	128	64	64
Learning Rate	1e-4	1e-4	1e-4
Adam ϵ	1e-6	1e-6	1e-6
Adam β_1	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.999

Table 3.5. Hyperparameters used to train transformer priors and the VQ-VAE encoder. The architecture of the priors is based on the scalable transformer proposed by Dhariwal et al. (2020). For the experiments on CelebA-HQ and ImageNet we used pretrained priors from Esser, Rombach, and Ommer (2021) (`celebahq_transformer` checkpoint for CelebA-HQ and `cin_transformer` for ImageNet). We refer the reader to their implementations for additional details on the hyperparameters used.

Dataset	Likelihood Steps	λ
MNIST	30000K	1.0
SLAKH	2584K	1.0
CelebA	32554K	3.0
CelebA-HQ	32554K	2.0,3.0,4.0
ImageNet	10000K	2.0,3.0,4.0

Table 3.6. *LASS* Hyperparameters. The number of single entry update steps used to construct the tensor P and the hyperparameter λ used at inference.

deduced from the compound $\mathbf{x}^1 = \mathbf{x}^2 = \mathbf{y}/2$. PSNR (Peak Signal to Noise Ratio) [58] is the chosen evaluation metric in all these instances. We emulate the testing approach of [68] on MNIST, separating a collection of 6,000 compounds formed by blending 12,000 test sources. To determine the optimal sampler for this dataset, samplers from Table 3.9 are verified on 1,000 compounds formed from the test segment. Stochastic samplers emerge as the most effective (PSNR $>$ 20 dB), while MAP strategies fall short. Due to MNIST's sparse nature, we theorize that early inference beam search can land on non-ideal solutions. Top-k sampling with $k = 32$ outperforms others, making it the chosen evaluation method (a visual comparison is depicted in Figure 3.6). For each compound in the test database, we draft a candidate batch of 512 separations, select the separation that most closely aligns with the compound (based on the $\mathcal{L}_2$ distance), and conclude with the refinement process using Eqs. (3.16), (3.17) with parameters $T = 500$ and $\alpha = 0.1$. Results of

Dataset	K	Density (%)
MNIST	256	1.49×10^{0}
CelebA	512	6.06×10^{0}
CelebA-HQ	1024	3.80×10^{-1}
ImageNet	16384	3.90×10^{-3}
Slakh (Drum + Bass)	2048	7.60×10^{-2}

Table 3.7. Statistics on likelihood functions over different datasets. K is the number of VQ-VAE (or VQ-GAN) latent codes. Density is the percentage of nonzero elements in the likelihood function.

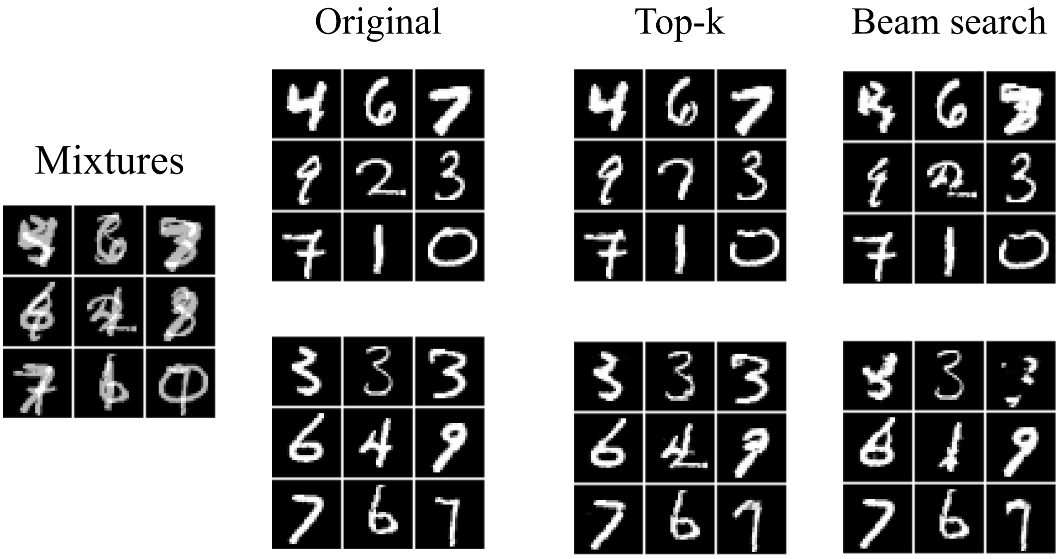


Figure 3.6. Results on MNIST with top-k sampling ($k = 32$) over a random batch of examples. Top-k sampling produces more defined digits, in agreement with the results in Table 3.9.

this examination are shown in Table 3.8 and inference durations in Table 3.10. Our technique surpasses "NMF", "S-D", and "BASIS Glow" in performance metrics and is swifter than "BASIS NCSN" due to latent quantization. The superior PSNR by the latter is likely because their generative models sample directly in the image domain; in our scenario, the compression through VQ-VAE might impact metrics.

On CelebA, we contrast our approach with "BASIS NCSN", deploying the pre-established NCSN model [168]. Here, we opt for the FID metric [54] over PSNR since datasets with a broader range than MNIST can render source separation as an ambiguous task [68]. The FID metric aptly determines if separations align with the source's distribution. We evaluate 10,000 compounds created from image pairs in the validation section using a top-k sampler set at $k = 32$. The likelihood component is amplified by a factor of $\lambda = 3$. While the literature has established that score-centric models typically outdo autoregressive ones on FID metrics [28], our method, coupled with an autoregressive model, exhibits promising outcomes against the score-based "BASIS NCSN".

Qualitative Results To illustrate the adaptability of *LASS* in incorporating pre-existing models without adjustments, we use pre-trained checkpoints from both

Separation Method	MNIST (PSNR)	CelebA (FID)
Average	14.9	15.19
NMF	9.4	-
S-D	18.5	-
BASIS Glow	22.7	-
BASIS NCSN	29.3	7.55
***LASS* (Ours)**	24.2	8.96

Table 3.8. Comparison with other methods on MNIST and CelebA test set. Results are reported in PSNR (higher is better) and FID (lower is better).

Sampling Method	MNIST (PSNR)	Slakh (SDR)
Greedy	17.36 ± 5.90	1.23 ± 2.33
Beam Search	16.96 ± 5.78	5.01 ± 2.39
Ancestral Sampl.	24.03 ± 6.37	4.23 ± 2.29
Top-k ($k = 16$)	23.74 ± 6.55	3.13 ± 2.53
Top-k ($k = 32$)	24.23 ± 6.23	2.93 ± 2.20
Top-k ($k = 64$)	23.85 ± 6.13	3.24 ± 3.29

Table 3.9. Performance of *LASS* with different sampling methods. On MNIST, the reported score is PSNR (dB) (higher is better), while on Slakh is SDR (dB) (higher is better). When stochastic samplers are used (ancestral or top-k), the selected solution in the batch is the one whose sum minimizes the $\mathcal{L}_2$ distance to the input mixture.

CelebA-HQ and ImageNet. Here, only the likelihood tensor P undergoes learning. We present a selection of results in Figure 3.4, with an in-depth collection accessible on our supplementary website. As per our research, this is the pioneering approach to achieve resolutions of 256×256, and it allows integration with more robust latent autoregressive models without the need for retraining (a process that's challenging for huge models). Consequently, users can execute generative separation without the necessity for vast computational training resources.

Music Source Separation

Our experiments for separating music sources utilize the Slakh2100 dataset [110]. This dataset comprises 2100 songs, each with distinct sources spanning 34 instrument classifications and 145 hours of mixed tracks. We emphasize the "Drums" and "Bass" classifications, with tracks at a 22kHz sampling rate. The public checkpoint from [26] for the VQ-VAE model is employed, capitalizing on its capacity to model audio over a quantized domain. Since this model trains at 44kHz, we linearly upsample the input and later reduce the output's sample rate to 22kHz. We introduce two Transformer models for autoregressive priors, one dedicated to "Drums" and the other to "Bass," and derive the likelihood function for the VQ-VAE (details found in Table 3.7). We position *LASS* alongside unsupervised blind source separation techniques like "rPCA" [59], "ICA" [64], "HPSS" [136], "FT2D" [156], and supervised benchmarks Demucs [34] and Conv-Tasnet [104]. The SDR (dB) metric, calculated via the `museval` tool [172], serves as our evaluation criterion. To assess the methodologies, 900 music fragments, each 3 seconds long, are chosen from the test portions of the "Drums" and "Bass" categories and are merged to create 450 composite tracks. The validation set follows a similar structure but uses distinct music segments. Our

	Method	Time
MNIST	***LASS* (Ours)**	4.49 s ± 0.27 s
	BASIS NCSN	53.34 s ± 0.51 s
Slakh	***LASS* (Ours)**	1.33 min ± 0.87 s
	PnF	42.29 min ± 1.08 s

Table 3.10. Inference speed comparisons for computing one separation. To estimate variance, we repeat inference 10 times on MINST and 3 times on Slakh. We consider 3-second-long mixtures on Slakh.

Separation Method	Avg	Drums	Bass
rPCA	0.82	0.60	1.05
ICA	-1.26	-0.99	-1.53
HPSS	-0.45	-0.56	-0.33
REPET	1.04	0.53	1.54
FT2D	0.95	0.59	1.31
***LASS* (Ours)**	4.86	4.73	4.98
Demucs	5.39	5.42	5.36
Conv-Tasnet	5.47	5.51	5.43

Table 3.11. Comparison with other source separation methods on Slakh ("Drums" and "Bass" classes). Results are reported in SDR (dB) (higher is better). Lower part of the table shows supervised methods. With "Avg" we refer to the mean between the results over the two classes.

chosen sampling methodology is beam search, as it demonstrated superior outcomes in a validation involving 50 mixtures (refer to Table 3.9), deploying $B = 100$ beams. The evaluation outcomes are in Table 3.11, where *LASS* visibly outperforms all unsupervised benchmarks and aligns closely with supervised methods. Moreover, we juxtapose the time efficiency of *LASS* with the generative source separation approach "PnF" [70] by gauging the duration needed to separate a 3-second, 22kHz sample (piano vs. voice for "PnF"). The data in Table 3.10 reveals that *LASS* operates notably swifter, making it apt for more practical inference environments.

Limitations

In this study, our focus centers on separating two distinct sources. This configuration, prevalent mainly in the realm of image separation [70, 51], could be expanded to accommodate more sources as a prospective avenue for research. Operationalizing this within our proposed framework would necessitate the enlargement of the dimensions related to the discrete distributions, encompassing both the prior knowledge and the likelihood measure. To circumvent the challenges associated with this enlargement, one might consider utilizing methods akin to recursive separation [182]. A further constraint of the method we have delineated is the localized assumption manifested in Eq. (3.14). Alternative undertakings like enhancing resolution or color enhancement demand an expansive conditioning environment. Moreover, introducing cutting-edge quantization strategies is imperative to consolidate latent code in broader contexts, notably by leveraging self-attention mechanisms within both the encoder and decoder phases of the VQ-VAE [210]. An ideal recourse might be the integration of a VQ-VAE, quantified relative to latent channels [206], amalgamated with a parametric likelihood measure. This would address the abovementioned constraint and simultaneously uphold the flexible separation between VQ-VAE, established priors, and likelihoods, as elucidated in this study.

3.2.5 Conclusion

In this chapter, we introduced *LASS* as a technique for source separation in latent autoregressive models without altering the underlying prior structures. Our approach has been evaluated across various datasets, demonstrating performance on par with contemporary leading techniques yet offering enhanced scalability and swifter inference capabilities. Our methodology further distinguishes itself by producing superior-resolution qualitative outcomes compared to rival methods. The advancement in autoregressive models will further enhance the efficacy of our approach, refining both the objective benchmarks and the perceptual outcomes.

3.3 Chapter Conclusions

The exploration in this chapter has elucidated the compelling potential and adaptability of autoregressive models, particularly the Transformer architecture, in handling complex sequential data. Through a meticulous journey from the model's theoretical underpinnings to its pragmatic applications, we have spotlighted its prowess in sequence prediction and its cascading benefits to natural language processing, time series forecasting, and audio processing. The innovative application of the unsupervised separation algorithm outlined in the first section not only showcases a promising avenue to alleviate the taxing demands of supervised data-driven approaches but also underscores the potency of deep Bayesian priors in harnessing the latent representations of time-domain signals. Through a novel loss term, the proposed mechanism has exhibited a significant stride towards exact Bayesian inference, which is instrumental in economizing computational resources while ensuring a robust separation performance. The validation of the Slakh dataset has further cemented the algorithm's merit in juxtaposition with state-of-the-art supervised methodologies, heralding a promising direction towards more resource-efficient and practical unsupervised separation approaches. In the subsequent section, the advent of LASS (Latent Autoregressive Source Separation) is a testament to the transformative potential of autoregressive models in seamlessly adapting to novel complex tasks without the necessity of exhaustive fine-tuning or model modifications. Introducing vector-quantized latent spaces has unveiled a pathway towards reduced dimensionality and expedited inference, essential in the real-time processing of high-dimensional data such as images and audio. The performance of LASS in our extensive evaluations, its superior inference speeds, and its adaptability echo the substantial advancements that autoregressive models encapsulate. Furthermore, the insights gleaned from the frequency evaluations of latent addend token aggregates have enriched our understanding of the discrete likelihood function, paving the way for more nuanced and effective separation methodologies in the future. Through a prism of rigorous experiments and evaluations, this chapter has underscored the transformative potential of autoregressive models and kindled a vista of exploration that could further stretch the boundaries of what's achievable in sequence modeling and source separation domains. The symbiotic relationship between autoregressive models and the burgeoning realm of deep learning is poised to continue blossoming, fostering a fertile ground for novel research endeavors and real-world applications.

Chapter 4

Diffusion Models for Music Sources

The infusion of diffusion paradigm within deep learning augments the comprehension and application of both fields, creating a fertile ground for innovation. The intersection of diffusion processes and deep learning techniques has engendered a novel paradigm instrumental in solving myriad problems, particularly in data-driven domains. This introduction elucidates the theoretical underpinnings, methodologies, and applications of diffusion models in deep learning, drawing from a wealth of scholarly contributions.

The conceptual scaffolding of diffusion models traces back to the early 20th century, with the foundational work of Fick (1855) [40] on diffusion processes, which was later enriched by the contributions of Einstein (1905) [35]. The mathematical characterization of diffusion processes has evolved over the years, culminating in a robust theoretical framework that underlies modern diffusion models. The synergy between diffusion processes and deep learning is rooted in the stochastic nature of diffusion, which integrates with the probabilistic frameworks often employed in deep learning.

The confluence of diffusion models and deep learning has given rise to innovative algorithms that excel in capturing complex data distributions and dependencies. Diffusion models operate by deteriorating and restoring the input data. In [166], Sohl-Dickstein et al. show how to revert a diffusion process creating new samples through the mechanics of denoising, deriving this idea from the principles of non-equilibrium thermodynamics.

These models are a subset of Markov random fields (MRFs). In this context, a sequence of diffusion steps incrementally introduces noise to the original data. The goal of the model then becomes to invert this diffusion, producing new and unique data samples from the altered data. In [168], they explored score-based generative modeling, which disrupts data using varying noise intensities, similar to diffusion models. In [171], Song argued that score-based generative and diffusion probabilistic models can be interpreted as *"approximations to stochastic differential equations steered by score functions."* The applications of diffusion models in deep learning are manifold, spanning across various disciplines, including but not limited to computer vision, natural language processing, and bioinformatics. In computer vision, diffusion models have been crucial in image generation, reconstruction, and denoising [205]. In natural language processing, diffusion models have shown promise in natural language generation, sentiment analysis, topic modeling, and machine translation [219]. Furthermore, bioinformatics has employed diffusion models for protein design and generation, drug and small molecule design, and protein-ligand interaction [48] The deployment of diffusion models in deep learning has indubitably broadened the horizons of what can be achieved in artificial intelligence. The cross-fertilization of ideas between these fields continues to propel the frontiers of knowledge forward, with the promise of addressing some of the most pressing challenges in data analytics and artificial intelligence.

In the following section, we will delve into a novel application of diffusion models, venturing into a domain that, while recently explored, remains burgeoning and rich with potential: the realm of audio and music.

4.1 Multi-Source Diffusion Models for Simultaneous Music Generation and Separation

In this chapter, we present a generative paradigm rooted in diffusion principles tailored for both the synthesis of music and the disentanglement of its sources, achieved by comprehending the shared contextual score of the probability density of these sources. Beyond the conventional inference operations, such as producing a combined output or isolating individual sources, we also explore the nuanced task of source imputation. In this task, given a set of musical sources, we produce complementary sources, for instance, synthesizing a piano sequence that harmoniously accompanies a drum track. We further unveil an innovative inference strategy for the source disentanglement problem, employing the Dirac likelihood functions. Leveraging the widely acknowledged Slakh2100 dataset, geared towards musical source disentanglement, we offer insights into the generative capabilities of our model and present competitive metrics on the source isolation front. This approach pioneers the unification of generative and separative capabilities within a singular audio modeling framework, pushing the frontier of holistic audio models.

4.1.1 Introduction

Recent advancements in generative models have showcased their potential across multiple domains, including NLP [121, 184], image creation [137, 143], and protein engineering [162]. The arena of audio has not remained untouched by these developments [1, 95]. An intrinsic characteristic of the audio domain is the perception of an audio instance $\mathbf{y}$ as an aggregate of several discrete sources $\mathbf{x}_1, \dots, \mathbf{x}_N$, leading to the resultant mixture:

$$\mathbf{y} = \sum_{n=1}^{N} \mathbf{x}_n .$$

In contrast to some audio sub-domains, notably speech, musical elements (stems) inherently possess a shared *context* due to their intertwined nature. Consider, for instance, how a song's bass sequence aligns with the drumbeat and resonates with the guitar tune. From a mathematical perspective, this implies the joint distribution of sources $p(\mathbf{x}_1, \dots, \mathbf{x}_N)$ does not decompose into the multiplication of individual source distributions $\{p_n(\mathbf{x}_n)\}_{n=1,\dots,N}$. Grasping the joint $p(\mathbf{x}_1, \dots, \mathbf{x}_N)$ leads to an understanding of the mixture's distribution $p(\mathbf{y})$, derived from their summation. However, the reverse process poses intricate mathematical challenges, representing an inverse problem.

Intriguingly, humans have honed the skill to concurrently handle multiple sound channels in synthesis (e.g., musical composition) and analysis (e.g., segregating sources). To elaborate, musicians can conjure multiple channels $\mathbf{x}_1, \dots, \mathbf{x}_N$ that

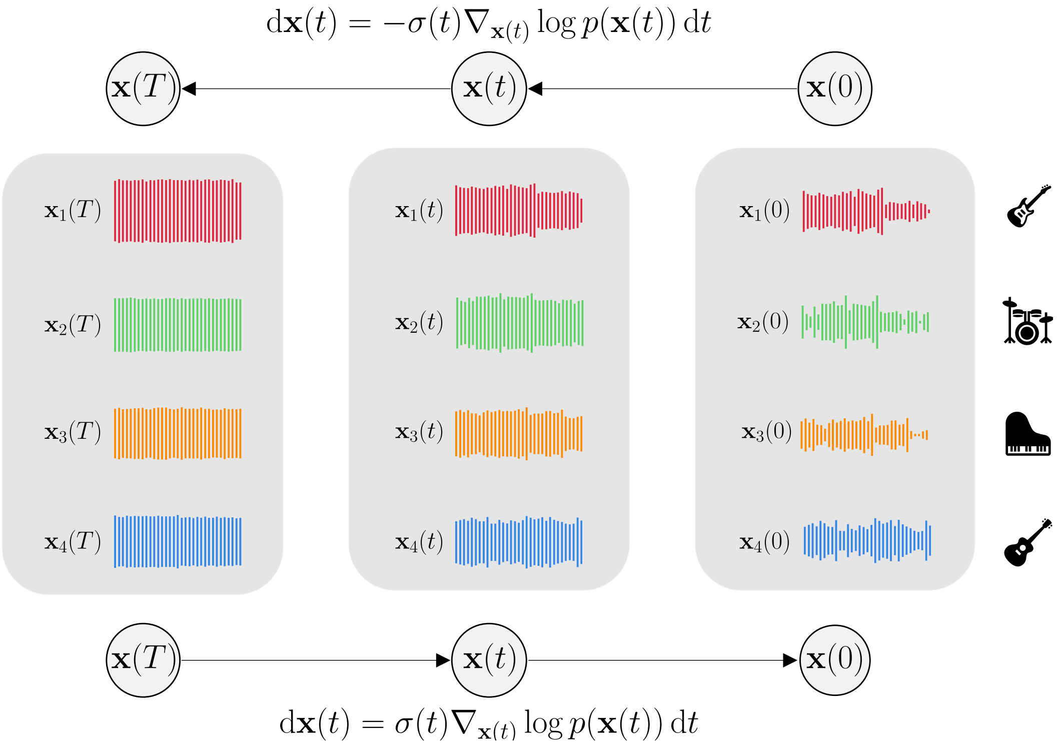


Figure 4.1. Diagram illustrating our proposed method. We leverage a forward Gaussian process (right-to-left) to learn the score over contextual sets (indicated by large rectangles) of instrumental sources (represented by waveforms) across different time steps t. During inference, the process is reversed (left-to-right), enabling us to perform tasks such as total generation, partial generation, or source separation (detailed in Figure 4.2).

coalesce into a harmonized mixture $\mathbf{y}$, and simultaneously discern particulars of the individual sources $\mathbf{x}_1, \ldots, \mathbf{x}_N$ within a given mixture $\mathbf{y}$. This dual proficiency in composing and deconstructing sounds is pivotal for generative music modeling. A quintessential music composition assistant model should exhibit the finesse to discern unique channels within a mix and facilitate standalone alterations on each. Such adeptness offers composers unparalleled autonomy in sculpting their musical pieces. Hence, we posit that generative music creation is intrinsically linked to music source separation. To our discernment, existing deep learning models only partially excel in both domains. Generative models focus on discerning the distribution $p(\mathbf{y})$ of mixtures, thereby overshadowing the nuances required for separation. Such models might excel in capturing the essence of mixtures but overlook individual channels' specifics. Notably, methods that condition mixture distributions on textual datasets [155, 1] encounter similar constraints. Conversely, models tailored for source disentanglement [23] either prioritize $p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \mathbf{y})$, contingent on the mixture, or design singular models $p_n(\mathbf{x}_n)$ for each channel, referring to the mix during application [68, 129]. In both scenarios, the holistic creation of mixtures becomes unfeasible.

Contribution. Our work stands on three pillars. **(i)** We pioneer the fusion of source separation and music synthesis by comprehending $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, representing the unified distribution of contextual channels (those from identical songs). We employ the denoising score-matching paradigm for this objective to train a *Multi-Source Diffusion Model (MSDM)*. This singular model adepts source separation and musical creation at inference time. Explicitly, while synthesis is realized by sampling the prior, separation is accomplished by making the prior conditioned on the mixture, followed by sampling from the ensuing posterior distribution. **(ii)** This avant-garde approach unlocks unprecedented ventures in the generative realm, like *source imputation*, where complementary channels are envisioned by generating certain sources in alignment with others (e.g., formulating a piano segment congruent with drum beats). **(iii)** In our pursuit of rivaling the prowess of contemporary discriminative models [109] on the Slakh2100 dataset [111], we introduce a refined technique to calculate the posterior score, leveraging *Dirac delta functions*, to capitalize on the intrinsic relationship between channels and their mix.

4.1.2 Related Works

Audio Generative Models

Generative models tailored for audio leverage deep learning techniques to either directly or indirectly capture the distribution of mixtures denoted as $p(\mathbf{y})$, which might also be influenced by auxiliary information such as text. Several widely used generative architectures like autoregressive models, GANs [30], and diffusion models have been repurposed for applications in audio.

Autoregressive models boast a robust legacy in audio modeling [193]. The Jukebox proposal [25] aims to represent musical compositions employing Scalable Transformers [195] on hierarchical discrete approximations obtained via VQ-VAEs [191]. Further enhancing its capabilities, it has a lyric-conditioning feature that generates vocal tracks in alignment with textual input. A notable challenge faced by Jukebox, though, was the emergence of quantization artifacts in audio outputs. By introducing residual quantization [212], more recent latent autoregressive models [8, 84] have managed to encompass broader contexts, leading to more refined and authentic audio outputs. Top-tier latent autoregressive models tailored for music, for instance, MusicLM [1], can modulate generation using textual embeddings garnered from expansive contrastive pre-training [108, 61]. MusicLM also possesses the capability to take a melody as input and adjust its style based on textual cues. Alternatively, SingSong [29] has generated accompaniment from vocals. The distinctiveness of our method lies in our generation mechanism at the stem level, which is modular, unlike SingSong, which produces a singular accompaniment mix.

Initiating the foray into diffusion (score) based audio generative models with a focus on speech synthesis were DiffWave [82] and WaveGrad [11]. This pioneering effort paved the way for numerous models, each catering to specialized objectives, be it speech enhancement [102, 157, 152, 149], audio upsampling [88, 209], conversion from MIDI-to-waveform [116, 52], or the transformation from spectrogram to MIDI [12]. The trailblazing effort in domain-specific generative outputs using diffusion models is attributed to CRASH [145]. Proposals from [207, 126, 95] have emphasized text-conditioned diffusion models aimed at creating diverse sounds, stepping beyond bounded categories such as speech or melodies. Parallel to our research, Riffusion [42] and Moûsai [155] focus on diffusion models for musical contexts. While Riffusion refines Stable Diffusion [143], a comprehensive pre-trained text-conditioned vision diffusion model, over STFT magnitude spectrograms, Moûsai's generation is set in a latent space, spanning contexts that can exceed a minute in duration. In our work, we have borrowed inspiration from Moûsai's U-Net design but with a twist, as we employ the waveform data format.

Audio Source Separation

Models specializing in audio source separation can be predominantly categorized into two primary classes: discriminative and generative. Models of a discriminative nature operate deterministically with parametric designs. They aim to receive audio mixtures and consistently extract individual or collective sources, emphasizing the likelihood of the inherent conditional distribution $p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \mathbf{y})$. Commonly, these models undergo training using a regression loss function [49]. The output signal is generally represented as a waveform [100, 104, 23], STFT [179, 16], or occasionally both [22].

Conversely, generative source separation models primarily aim to understand a prior model pertinent to each source, focusing on the distributions $\{p_n(\mathbf{x}_n)\}_{n=1,\ldots,N}$. It is noteworthy that mixtures are predominantly observable during the inference phase. A likelihood function aligns the mix with its foundational sources during this phase. The academic discourse has delved into various priors, including GAN-based techniques [175, 81, 118], methodologies reliant on normalizing flows [68, 218], and those leveraging autoregressive models [70, 129]. The NCSN-BASIS method [68], an approach focusing on source separation in image contexts, aligns closely with our methodological approach. This method employs Langevin Dynamics, using the NCSN score-oriented model to disentangle mixtures. An intriguing aspect of this method is its application of a Gaussian likelihood function during the inference phase. Our experimental results, however, indicate that our newly introduced Dirac-based likelihood function exhibits superior performance. Compared to other generative source separation techniques (NCSN-BASIS included), a distinctive feature of our approach is its ability to model the comprehensive joint distribution. Consequently, using a singular model, our approach is equipped to separate sources and craft mixtures or partial stems.

A deliberate effort is to model the interrelation between sources in specific studies such as [109] and [132]. Given the foundational mixture and residual sources, the former approach utilizes an orderless NADE estimator to predict a subset of the sources, considering the input mixture and other remaining sources. The latter pushes the envelope with universal source separation [74, 203], leveraging adversarial training and a context-based discriminator to discern the interplay between sources. These techniques are intrinsically discriminative due to their mixture-dependent architectural design. A similar constraint is observable in discriminative strategies for source separation that adopt diffusion-centric [153, 105] or diffusion-motivated [128] methodologies. A distinctive trait of our approach is the architectural freedom from a mixture conditioner, allowing us to carry out unconditional generation.

4.1.3 Method

Our model's core principle is anchored in gauging the joint distribution of the sources, represented by $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. This model is generative, as we construct an unconditional distribution, which we refer to as the prior. Subsequent tasks leverage this prior during the inference phase. We adopt a diffusion-based generative paradigm as cited in [167, 55] and train it using denoising score-matching techniques [169]. The framework and conventions we use draw from the foundational work described in [73]. A pivotal concept behind score-matching, as referenced in [62, 78, 198], centers on approximating the "score" function of the sought-after distribution $p(\mathbf{x})$, specifically $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, instead of the distribution per se. To efficaciously hone the

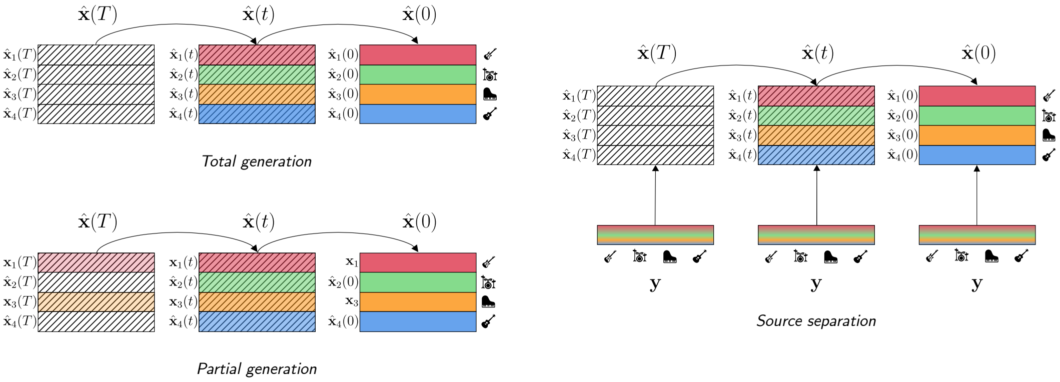


Figure 4.2. Inference tasks with MSDM. Oblique lines represent the presence of noise in the signal, decreasing from left to right, with the highest noise level at time T when we start the sampling procedure. *Top-left:* We generate all stems in a mixture, obtaining a total generation. *Bottom-left:* We perform partial generation (source imputation) by fixing the sources $\mathbf{x}_1$ (Bass) and $\mathbf{x}_3$ (Piano) and generating the other two sources $\hat{\mathbf{x}}_2(0)$ (Drums) and $\hat{\mathbf{x}}_4(0)$ (Guitar). We denote with $\mathbf{x}_1(t)$ and $\mathbf{x}_3(t)$, the noisy stems obtained from $\mathbf{x}_1$ and $\mathbf{x}_3$ via the perturbation kernel in Eq. (4.1). *Right:* We perform source separation by conditioning the prior with a mixture $\mathbf{y}$, following Algorithm 2.

score in regions with sparse data, denoising diffusion techniques infuse the data with controlled noise and subsequently learn its removal. Formally, the data's distribution undergoes a perturbation via a Gaussian kernel as expressed:

$$p(\mathbf{x}(t) \mid \mathbf{x}(0)) = \mathcal{N}(\mathbf{x}(t); \mathbf{x}(0), \sigma^2(t)\mathbf{I}), , \tag{4.1}$$

The variable $\sigma(t)$ dictates the noise intensity infused into the data. In alignment with [73], our choice for $\sigma(t)$ is a progressive schedule, defined by $\sigma(t) = t$. Consequently, the data point's progressive trajectory, denoted as $\mathbf{x}(t)$, over time can be depicted by a probabilistic flow as defined by the ODE in [170]:

$$\mathrm{d}\mathbf{x}(t) = -\sigma(t)\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)), \mathrm{d}t, . \tag{4.2}$$

Given a sufficiently large $t = T$, the data point $\mathbf{x}(T)$ approximates the Gaussian distribution $\mathcal{N}(\mathbf{x}(t); \mathbf{0}, \sigma^2(T)\mathbf{I})$, facilitating simplified sampling. The equation (4.2) can undergo a time inversion, yielding the subsequent backward ODE that elucidates the denoising mechanism:

$$\mathrm{d}\mathbf{x}(t) = \sigma(t)\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)), \mathrm{d}t, . \tag{4.3}$$

Sampling from the data distribution can be realized by integrating Eq. (4.3) using a conventional ODE solver, initiating from a noisy sample sourced from $\mathcal{N}(\mathbf{x}(t); \mathbf{0}, \sigma^2(T)\mathbf{I})$. The score function, epitomized by a neural network $S^\theta(\mathbf{x}(t), \sigma(t))$, is honed by minimizing the ensuing score-matching loss:

$$\mathbb{E}t \sim \mathcal{U}([0,T])\mathbb{E}\mathbf{x}(0) \sim p(\mathbf{x}(0))\mathbb{E}\mathbf{x}(t) \sim p(\mathbf{x}(t) \mid \mathbf{x}(0)) \left\| S^\theta(\mathbf{x}(t), \sigma(t)) - \nabla\mathbf{x}(t) \log p\left(\mathbf{x}(t) \mid \mathbf{x}(0)\right) \right\|_2^2 .$$

By extrapolating $p(\mathbf{x}(t) \mid \mathbf{x}(0))$ using Eq. (4.1), the score-matching loss can be streamlined to:

$$\mathbb{E}t \sim \mathcal{U}([0,T])\mathbb{E}\mathbf{x}(0) \sim p(\mathbf{x}(0))\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2(t)\mathbf{I})} \left\| D^\theta(\mathbf{x}(0) + \epsilon, \sigma(t)) - \mathbf{x}(0) \right\|_2^2 ,$$

With this framework, we articulate $S^\theta(\mathbf{x}(t), \sigma(t))$ as $(D^\theta(\mathbf{x}(t), \sigma(t)) - \mathbf{x}(t))/\sigma^2(t)$.

Multi-Source Audio Diffusion Models

In our configuration, we consider N unique source waveforms represented as $\mathbf{x}_1, \ldots, \mathbf{x}_N$, with every $\mathbf{x}_n$ belonging to $\mathbb{R}^D$. These sources are coherently combined to produce the mixed waveform $\mathbf{y} = \sum_{n=1}^{N} \mathbf{x}_n$. Occasionally, we represent this collection in the concatenated form: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \subset \mathbb{R}^{N \times D}$. Several operations can be carried out within this framework: it is possible to construct a cohesive mixture, $\mathbf{y}$, or separate the individual components, $\mathbf{x}$, from a provided mixture. Creating the mixture is termed as *generation* while extracting the components is termed *source separation*. In *generation*, a specific subset of sources can be held constant, and the remaining sources generated in a compatible manner. This is termed as *partial generation* or *source imputation*. A salient feature of our approach is the capability to execute all these operations concurrently by employing a singular multi-source diffusion model (MSDM) that encapsulates the prior $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. The design of this model is depicted in Figure 4.1 and is engineered to approximate the noisy score function:

$$\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)) = \nabla_{(\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t))} \log p(\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t)),$$

leveraging a neural network framework:

$$S^\theta(\mathbf{x}(t), \sigma(t)) : \mathbb{R}^{N \times D} \times \mathbb{R} \to \mathbb{R}^{N \times D}, \tag{4.4}$$

where $\mathbf{x}(t) = (\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t))$ represents the sources after being influenced by the Gaussian kernel in Eq. (4.1). The operations, as mentioned earlier, are elucidated (and visualized in Figure 4.2) through the prior distribution:

- *Total Generation.* In this operation, the aim is to generate a coherent mixture, $\mathbf{y}$. This is realized by drawing the sources $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ from the prior distribution and subsequently integrating them to form the mixture $\mathbf{y}$.

- *Partial Generation.* In scenarios where only a subset of sources is available, this operation generates a harmonious accompaniment. The given sources are represented as $\mathbf{x}_\mathcal{I}$, and the absent sources, $\mathbf{x}_{\overline{\mathcal{I}}}$, are generated by drawing from the conditional probability distribution $p(\mathbf{x}_{\overline{\mathcal{I}}} \mid \mathbf{x}_\mathcal{I})$.

- *Source Separation.* Upon being provided with a mixture $\mathbf{y}$, this operation aims to extract the individual contributing sources. This is realized by drawing from the posterior distribution $p(\mathbf{x}|\mathbf{y})$.

Inference Our methodology addresses its three core tasks at inference time by discretizing the backward Eq. (4.3). Even though every task necessitates a unique score function, they are fundamentally derived from the initial score function provided in Eq. (4.4). We delve deeper into the nuances of each of these score functions.

Total Generation The complete synthesis task is accomplished by drawing samples from Eq. (4.3) exploiting the score function detailed in Eq. (4.4). Subsequently, the mixture is achieved by aggregating all the individual sources.

Partial Generation For the partial generation task, a particular subset of source indices, denoted by $\mathcal{I}$ which is a subset of $\{1, \ldots, N\}$, is set alongside the associated sources represented as $\mathbf{x}_\mathcal{I} := \{\mathbf{x}_n\}_{n \in \mathcal{I}}$. The objective here is to synthesize the residual sources, represented as $\mathbf{x}_{\overline{\mathcal{I}}} := \{\mathbf{x}_n\}_{n \in \overline{\mathcal{I}}}$ in a harmonized manner, where

Algorithm 2 ‘MSDM Dirac’ sampler for source separation.

```
Require: $I$ number of discretization steps for the ODE, $R$ number of corrector steps, $\{\sigma_i\}_{i \in \{0,\dots,I\}}$ noise schedule, $S_{\text{churn}}$
Initialize $\hat{\mathbf{x}} \sim \mathcal{N}(0, \sigma_I^2 \mathbf{I})$
$\alpha \leftarrow \min(S_{\text{churn}}/I, \sqrt{2} - 1)$
for $i \leftarrow I$ to 1 do
    for $r \leftarrow R$ to 0 do
        $\hat{\sigma} \leftarrow \sigma_i \cdot (\alpha + 1)$
        $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
        $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \sqrt{\hat{\sigma}^2 - \sigma_i^2}\epsilon$
        $\mathbf{z} \leftarrow [\hat{\mathbf{x}}_{1:N-1}, \mathbf{y} - \sum_{n=1}^{N-1} \hat{\mathbf{x}}_n]$
        for $n \leftarrow 1$ to $N - 1$ do
            $\mathbf{g}_n \leftarrow S_n^\theta(\mathbf{z}, \hat{\sigma}) - S_N^\theta(\mathbf{z}, \hat{\sigma})$
        end for
        $\mathbf{g} \leftarrow [\mathbf{g}_1, \dots, \mathbf{g}_{N-1}]$
        $\hat{\mathbf{x}}_{1:N-1} \leftarrow \hat{\mathbf{x}}_{1:N-1} + (\sigma_{i-1} - \hat{\sigma})\mathbf{g}$
        $\hat{\mathbf{x}} \leftarrow [\hat{\mathbf{x}}_{1:N-1}, \mathbf{y} - \sum_{n=1}^{N-1} \hat{\mathbf{x}}_n]$
        if $r > 0$ then
            $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
            $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon$
        end if
    end for
end for
return $\hat{\mathbf{x}}$
```

$\overline{\mathcal{I}} = \{1, \dots, N\} - \mathcal{I}$. This leads us to compute the gradient of the conditional distribution as:

$$\nabla_{\mathbf{x}_{\overline{\mathcal{I}}}(t)} \log p(\mathbf{x}_{\overline{\mathcal{I}}}(t) \mid \mathbf{x}_{\mathcal{I}}(t)). \tag{4.5}$$

This task is analogous to the process of data imputation, or, as it is prevalently termed in the realm of image processing, inpainting. We adopt a strategy for imputation based on the study by [170]. The gradient described in Eq. (4.5) can be deduced as:

$$\nabla_{\mathbf{x}_{\overline{\mathcal{I}}}(t)} \log p([\mathbf{x}_{\overline{\mathcal{I}}}(t), \hat{\mathbf{x}}_{\mathcal{I}}(t)])\,,$$

with $\hat{\mathbf{x}}_{\mathcal{I}}$ being a sample derived from the forward process: $\hat{\mathbf{x}}_{\mathcal{I}}(t) \sim \mathcal{N}(\mathbf{x}_{\mathcal{I}}(t); \mathbf{x}_{\mathcal{I}}(0), \sigma(t)^2 \mathbf{I})$. While estimating the score function, we elucidate:

$$\nabla_{\mathbf{x}_{\overline{\mathcal{I}}}(t)} \log p(\mathbf{x}_{\overline{\mathcal{I}}}(t) \mid \mathbf{x}_{\mathcal{I}}(t)) \approx S_{\overline{\mathcal{I}}}^\theta([\mathbf{x}_{\overline{\mathcal{I}}}(t), \hat{\mathbf{x}}_{\mathcal{I}}(t)], \sigma(t))\,,$$

in which $S_{\overline{\mathcal{I}}}^\theta$ signifies the segments of the score network that pertain to the sources indexed by $\overline{\mathcal{I}}$.

Source Separation We interpret source separation within the context of conditional generation, conditioning our generation on a given mixture represented as $\mathbf{y} = \mathbf{y}(0)$. The crucial component is determining the gradient of the log posterior distribution:

$$\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t) \mid \mathbf{y}(0))\,. \tag{4.6}$$

Conditional generation in diffusion models often incorporates either a direct computation of the posterior gradient during training, as seen in Eq. (4.6) and articulated by [56], or an estimation of the likelihood function $p(\mathbf{y}(0) \mid \mathbf{x}(t))$, subsequently using Bayes' theorem to obtain the posterior. Notably, the latter approach frequently necessitates an independent model, typically a classifier, to ascertain the likelihood function's gradient, as depicted in Classifier Guided conditioning documented by [27]. In diffusion-based separation, learning a likelihood model becomes redundant. This is primarily because the connection between $\mathbf{x}(t)$ and $\mathbf{y}(t)$ is encapsulated through a straightforward function, the summation in this case. Therefore, basing the likelihood function on this functional relationship emerges as a logical strategy. This methodology is embraced by [68], who employ a Gaussian likelihood function:

$$p(\mathbf{y}(t) \mid \mathbf{x}(t)) = \mathcal{N}(\mathbf{y}(t) \mid \sum_{n=1}^{N} \mathbf{x}_n(t), \gamma^2(t)\mathbf{I}), \tag{4.7}$$

with $\gamma(t)$ being a hyperparameter dictating the standard deviation. They emphasize that calibrating the value of $\gamma(t)$ in line with $\sigma(t)$ enhances the efficacy of the NCSN-BASIS separator. We introduce an innovative estimation for the gradient of the posterior in Eq. (4.6), representing $p(\mathbf{y}(t) \mid \mathbf{x}(t))$ via a Dirac delta function centered around $\sum_{n=1}^{N} \mathbf{x}_n(t)$:

$$p(\mathbf{y}(t) \mid \mathbf{x}(t)) = \mathbb{1}_{\mathbf{y}(t)=\sum_{n=1}^{N} \mathbf{x}_n(t)} \,. \tag{4.8}$$

The intricate derivation of this function is elaborated later, but we primarily focus on the final construct, termed as 'MSDM Dirac.' In this method, a particular source, represented as $\mathbf{x}_N$, is restricted by the equation $\mathbf{x}_N(t) = \mathbf{y}(0) - \sum_{n=1}^{N-1} \mathbf{x}_n(t)$, which then estimates:

$$\begin{aligned}\nabla_{\mathbf{x}_m(t)} \log p(\mathbf{x}(t) \mid \mathbf{y}(0)) \approx S_m^\theta((\mathbf{x}_1(t), \dots, \mathbf{x}_{N-1}(t), \mathbf{y}(0) - \sum_{n=1}^{N-1} \mathbf{x}_n(t)), \sigma(t)) \\ - S_N^\theta((\mathbf{x}_1(t), \dots, \mathbf{x}_{N-1}(t), \mathbf{y}(0) - \sum_{n=1}^{N-1} \mathbf{x}_n(t)), \sigma(t)) \,,\end{aligned}$$

Here, $1 \leq m \leq N-1$, with S_m^θ and S_N^θ signifying score network entries pertaining to the m-th and N-th sources respectively. This method captures the extreme scenario where $\gamma(t) \to 0$ within the Gaussian likelihood function. It characterizes a situation where the interaction between $\mathbf{x}(t)$ and $\mathbf{y}(t)$ is tightly bound, hence refining the generation's conditional aspect based on the provided mixture. The 'MSDM Dirac' source separation sampler's pseudo-code, leveraging the Euler ODE integrator documented by [73], is detailed in Algorithm 2. The Euler ODE discretization methodology capitalizes on the S_{churn} technique by [73], complemented by optional correction steps as mentioned by [170]. Lastly, this separation protocol can be further applied in weakly supervised source separation scenarios, a common challenge in generative source separation as reported by [68, 218, 129]. In these situations, while there is knowledge about specific audio data's affiliation to a musical instrument category, contextual source sets are absent. To navigate this, we hypothesize a source independence denoted as $p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{n=1}^{N} p_n(\mathbf{x}_n)$ and train an individual model for every source category. This different model is labeled 'Independent Source Diffusion Model with Dirac Likelihood' or 'ISDM Dirac.'

4.1.4 Experiments

Experiments are executed on Slakh2100 [111], which stands out as a prime dataset for music source separation. The decision to use Slakh2100 is influenced by

Table 4.1. Comparison between total generation capabilities of MSDM (Slakh2100) and an equivalent architecture trained on Slakh2100 mixtures. Both subjective (quality and coherence) and objective (FAD) evaluation is shown. Subjective evaluation is performed through listening tests, where subjects are asked to evaluate songs from 1 to 10 with respect to overall quality of the chunk and to coherence (i.e. how the instruments sound plausible together). Results show a very small difference between the model trained on mixtures and MSDM. *This suggests that, given the same dataset and architecture, the generative power of MSDM is the same as the model trained on mixtures*, while being able to perform separation and partial generation.

Model	FAD ↑	Quality ↑	Coherence ↑
MSDM	6.55	6.44 ± 2.12	6.34 ± 2.37
Mixture Model	6.67	6.04 ± 2.48	5.63 ± 2.65

its extensive data volume (145h), surpassing other multi-source waveform datasets like MusDB [135] that offer only 10h. The volume of data significantly impacts the quality of a generative model, positioning Slakh2100 as an optimal selection.

Dataset

Our experimental analysis utilizes the Slakh2100 dataset [111], a benchmark dataset for music source separation. Originating from MIDI files, the Slakh2100 encompasses synthesized multi-track waveform music data created using top-tier virtual instruments. The dataset offers 2100 tracks, broken down into 1500 training tracks, 375 for validation, and 225 designated for testing. Each track in the dataset has associated stems spanning 31 different instrument categories. In line with [109] and to maintain consistency in comparison, we restricted our analysis to the four predominant classes: Bass, Drums, Guitar, and Piano. These instruments are evident in a vast majority of the tracks: 94.7% (Bass), 99.3% (Drums), 100.0% (Guitar), and 99.3% (Piano).

Architecture and Training

The design foundation of our score network is rooted in the non-latent, time-domain variant of Moûsai [155]. For the implementation, we turned to the open-source codebase `audio-diffusion-pytorch/v0.0.432`[1]. Central to our model is the U-Net structure [144], which encompasses an encoder, a central bottleneck, and a decoder, with the inclusion of skip connections bridging the encoder and decoder. The encoder boasts six layers, with the first half housing two convolutional ResNet segments, while multi-head attention is present only in the latter half. Sequentially, the signal undergoes a 4-fold downsampling in each layer. The channel configuration across encoder layers is [256, 512, 1024, 1024, 1024, 1024]. The intermediary bottleneck holds a ResNet block and self-attention sequence, followed by another ResNet block (each bearing 1024 channels). The decoder, in essence, mirrors the encoder in its architecture. We employed the `audio-diffusion-pytorch-trainer`[2] during our training phase. We adjusted the data to 22kHz and trained the score-network with four unified mono channels for MSDM (a single channel per stem) and one dedicated mono channel for every model in ISDM. The context length was maintained at roughly 12 seconds. The training was executed on an NVIDIA RTX

[1] https://github.com/archinetai/audio-diffusion-pytorch/tree/v0.0.43

[2] https://github.com/archinetai/audio-diffusion-pytorch-trainer/tree/79229912

Table 4.2. Hyperparameter search for source separation using ‘MSDM Dirac’ (top-left), ‘ISDM Dirac’ (bottom-left), ’MSDM Gaussian’ (top-right) and ’ISDM Gaussian’ (bottom-right) posteriors. We report the SI-SDR$_\mathrm{i}$ values in dB (higher is better) averaged over all instruments (Bass, Drums, Piano, Guitar).

		Dirac Likelihood				**Gaussian Likelihood**						
		Constrained Source				$\gamma(t)$						
	S_{churn}	Bass	Drums	Guitar	Piano	$0.25\sigma(t)$	$0.5\sigma(t)$	$0.75\sigma(t)$	$1\sigma(t)$	$1.25\sigma(t)$	$1.5\sigma(t)$	$2\sigma(t)$
MSDM	0	4.41	5.05	3.28	2.87	-41.54	6.37	6.05	5.67	5.729	5.13	4.33
	1	7.90	8.18	7.03	7.05	-47.24	6.79	6.51	6.15	6.19	5.66	4.45
	20	**14.29**	12.99	12.19	11.69	-47.17	11.07	10.51	9.43	10.19	9.18	7.58
	40	14.28	13.02	5.51	4.78	-47.17	-36.92	**12.48**	11.25	11.87	10.80	9.03
ISDM	0	5.05	3.69	-2.50	6.93	-45.46	7.12	6.50	5.78	5.02	4.49	3.69
	1	9.23	8.57	7.28	9.20	-47.54	7.57	7.20	6.32	5.35	4.82	3.83
	20	15.35	15.08	13.20	15.36	-46.86	12.89	12.21	10.87	9.32	8.32	6.47
	40	**17.26**	15.77	15.30	14.98	-46.86	-35.97	**14.09**	12.82	10.85	10.02	8.26
	60	16.21	15.57	15.51	14.20	-46.80	-46.85	14.06	12.57	11.83	10.81	9.24

A6000 GPU with 24 GB of VRAM. The Adam optimizer [76] was adopted with a learning rate set to 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a batch size of 16.

Sampling Methodology

Our sampling methodology incorporates a first-order ODE integrator that relies on the Euler method, and we infuse stochasticity as delineated by [73]. The degree of stochasticity is governed by the variable S_{churn}. As corroborated in the next section and elaborated in [73], introducing stochastic elements substantially elevates sample fidelity. We introduced a correction approach [170, 68] that undergoes R iterative steps post each prediction step i, infusing more noise and re-tuning with the score network anchored at σ_i. In accord with [73], our time discretization adopts a non-linear trajectory, accentuating lower noise levels. This can be defined as:

$$t_i = \sigma_i = \sigma_{\text{max}}^{\frac{1}{\rho}} + \frac{i}{I-1}(\sigma_{\text{min}}^{\frac{1}{\rho}} - \sigma_{\text{max}}^{\frac{1}{\rho}})^{\rho} ,$$

where the range of i is $0 \le i < I$, and I signifies the count of discretization iterations. We designated values as $\sigma_{\text{min}} = 10^{-4}$, $\sigma_{\text{max}} = 1, and \rho = 7$.

Tuning Hyperparameters for Music Source Separation

To discern the impact of stochasticity on music source separation, we undertook a hyperparameter sweep over S_{churn}. This assessment was conducted over a selectively chosen subset of the Slakh2100 test set, encompassing 100 segments, each lasting 12 seconds. Ensuring a balanced comparison between Dirac (’MSDM Dirac’, ’ISDM Dirac’) and Gaussian (’MSDM Gaussian’, ’ISDM Gaussian’) posterior metrics, we carried out a granular search over their intrinsic hyperparameters, namely the restricted source for Dirac separators and the coefficient $\gamma(t)$ for the Gaussian variants. The outcomes are encapsulated in Table 4.2. Our observations highlight that: **(i)** all separators benefit from stochasticity, as highest SI-SDR$_\mathrm{i}$ values are noted at $S_{\text{churn}} = 20$ and $S_{\text{churn}} = 40$, **(ii)** Dirac likelihoods tend to offer superior SI-SDR$_\mathrm{i}$ values compared to Gaussian counterparts for both the MSDM and ISDM separators, and **(iii)** ISDM separators outperform the context-driven MSDM separators, albeit at the trade-off of not enabling complete and partial generation.

Table 4.3. Quantitative results for source separation on the Slakh2100 test set. We use the SI-SDR$_\text{i}$ as our evaluation metric (dB – higher is better). We present both the supervised ('MSDM Dirac', 'MSDM Gaussian') and weakly-supervised ('ISDM Dirac', 'ISDM Gaussian') separators and specify if a correction step is used. 'All' reports the average over the four stems. The results show that: (i) Dirac likelihood improves overall results, even *outperforming the state of the art when applied to ISDM* (ii) adding a correction step is beneficial (iii) *MSDM with Dirac likelihood and one step of correction gives results comparable with the state of the art and superior to standard Demucs overall.* We stress again that, while the baselines are trained on the separation task alone, MSDM is able to perform also generative tasks.

Model	Bass	Drums	Guitar	Piano	All
Demucs [23, 109]	15.77	19.44	15.30	13.92	16.11
Demucs + Gibbs (512 steps) [109]	17.16	19.61	**17.82**	**16.32**	**17.73**
Dirac Likelihood					
ISDM	18.44	20.19	13.34	13.25	16.30
ISDM (correction)	**19.36**	**20.90**	14.70	14.13	17.27
MSDM	16.21	17.47	12.71	13.29	14.92
MSDM (correction)	17.12	18.68	15.38	14.73	16.48
Gaussian Likelihood [68]					
ISDM	13.48	18.09	11.93	11.17	13.67
ISDM (correction)	14.27	19.10	12.74	12.20	14.58
MSDM	12.53	16.82	12.98	9.29	12.90
MSDM (correction)	13.93	17.92	14.19	12.11	14.54

The subsequent sections present findings on Music Generation and then delve into Source Separation outcomes.

Music Generation

MSDM's efficacy in generative assignments is gauged by subjective and objective metrics. The subjective assessment involves auditory tests. Two distinct forms were developed; the former, showcased in Table 4.1, prompts participants to evaluate the quality and instrument consistency of 30 generated segments, with half stemming from the mixture model and the other half from MSDM. The latter form necessitates participants to evaluate, keeping the instrument set constant, the quality, and the richness (density) of the generated accompaniment. From an objective standpoint for generative evaluations, we adapt the FAD protocol as described in [29] to our total generation assignment and partial generation with multiple sources. Consider D_real as a dataset encompassing authentic mixture samples and $\mathcal{I}$ as a set denoting conditioning sources (with $\emptyset$ signifying total generation). We then create a dataset D_gen wherein components constitute the summation of conditioning sources (indexed via $\mathcal{I}$) and their corresponding generated sources. The *sub-FAD* is defined as $FAD(D_\text{real}, D_\text{gen})$. Innovatively, our technique pioneers the generation of diverse partial source combinations, and in its absence of a direct comparative benchmark, we introduce the sub-FAD outcomes of our approach as foundational metrics for ensuing studies combined with auditory test outcomes. The results for total and partial generations are summarized in Tables 4.1 and 4.4. Table 4.1 indicates that MSDM's generative prowess mirrors a model with analogous architecture trained on similar dataset mixtures. Conversely, Table 4.1 underscores that the partial generation challenge is accomplished with discernible finesse, establishing a precedent for subsequent endeavors in universal accompaniment generation.

Table 4.4. Quantitative and qualitative results for the partial generation task on Slakh2100. We use the FAD as our objective evaluation metric (lower is better). In bold (**B**: Bass, **D**: Drums, **G**: Guitar, **P**: Piano) the combinations of generated sources. The 'quality' and 'density' columns refer to the average scores of the listening tests, with respective variances. Namely, 'quality' tests how the full chunk sounds plausible, with respect to the dataset, and 'density' tests how present in the chunk are the instruments that the model has to generate. They both are given on a scale from 1 to 10. No baseline is reported since our work is the first able to generate any combination of accompaniments; the results thus pose a baseline for future works on general accompaniment generation.

Slakh2100	**B**	**D**	**G**	**P**	**BD**	**BG**	**BP**	**DG**	**DP**	**GP**	**BDG**	**BDP**	**BGP**	**DGP**	**Quality**	**Density**
MSDM	0.45	1.09	0.11	0.76	2.09	1.00	2.32	1.45	1.82	1.65	2.93	3.30	4.90	3.10	6.2 ± 2.6	6.1 ± 2.6

Source Separation

For the assessment of source separation, we employ the scale-invariant SDR improvement metric, denoted as SI-SDR$_\text{i}$ [146]. The computation of SI-SDR for a source $\mathbf{x}_n$ in juxtaposition with its estimate $\hat{\mathbf{x}}_n$ is calculated as:

$$\text{SI-SDR}(\mathbf{x}_n, \hat{\mathbf{x}}_n) = 10 \log_{10} \frac{\|\alpha \mathbf{x}_n\|^2 + \epsilon}{\|\alpha \mathbf{x}_n - \hat{\mathbf{x}}_n\|^2 + \epsilon},$$

with $\alpha = \frac{\mathbf{x}_n^\top \hat{\mathbf{x}}_n + \epsilon}{\|\mathbf{x}_n\|^2 + \epsilon}$ and $\epsilon = 10^{-8}$. The enhancement over the mixture baseline is represented as $\text{SI-SDR}_\text{i} = \text{SI-SDR}(\mathbf{x}_n, \hat{\mathbf{x}}_n) - \text{SI-SDR}(\mathbf{x}_n, \mathbf{y})$.

In the Slakh context, our supervised MSDM and its weakly-supervised variant are juxtaposed against the 'Demucs' [23] as well as the 'Demucs + Gibbs (512 steps)' baselines from [109]. These benchmarks in supervised music source separation for Slakh2100, are consistent with the evaluation procedure of [109]. Our evaluation is carried out on the Slakh2100 test set, extracting 4-second segments (overlapped by two seconds) and excluding segments that are either silent or possess only a singular source, given the sub-optimal SI-SDR$_\text{i}$ performance for such intervals. We then compare our Dirac score posterior against the Gaussian score posterior by [68] involving 150 inference iterations.

The outcomes are delineated in Table 4.3. Succinctly, MSDM demonstrates a performance that nearly aligns with the SOTA. Furthermore, in its weakly supervised version, the introduced sampling approach occasionally surpasses the prevailing benchmarks for certain stems.

4.1.5 Conclusion

This chapter introduced a versatile technique anchored in denoising score-matching for source separation, total generation, and music accompaniment production. The novelty lies in deploying a singular neural network, which undergoes a single training phase, distinguishing tasks during the inference process. Furthermore, we proposed an innovative sample technique specifically for source separation. Our evaluations on source separation proved the model's ability, showcasing performance metrics similar to the leading regressor models. We subjected the model to qualitative and quantitative assessments for total and partial generation tasks. In the holistic evaluation, the model mirrored the generative capabilities of its counterpart trained on blends. The results confirmed the generation of credible and sophisticated accompaniments in the segmental context. Our model stands out due to its adept handling of holistic, segmental generation, and source separation, making it a pivotal advancement for holistic audio models. Such adaptability heralds the emergence of sophisticated musical composition utilities, allowing users to interact with and adjust individual elements in a blend seamlessly. Our model's efficacy is tied to the volume of contextual data it can access. A potential enhancement could involve pre-differentiating blends and orienting the training using these differentiations, as evidenced in [29]. It is also worth probing into adapting our technique for scenarios where the sub-signals do not interact additively but possibly through another distinct function. An avenue for further exploration is the model's transition to the discrete MIDI realm, aiming for more nuanced source imputation, especially given the rich data environment of this domain.

4.2 Chapter Conclusions

The dawn of deep learning has indubitably been conducted in an era of monumental advances in artificial intelligence as these evolutions continue to shape various facets of computational sciences, the fusion of diffusion setting with deep learning witnesses the innovative spirit that thrives at the intersection of apparently disparate concepts.

This chapter delved deep into the intersection of the diffusion paradigm and deep learning in audio, specifically music. By building upon the foundational ideas of diffusion, a process that has been with us since the early works of Fick and later enriched by the likes of Einstein, we introduced an innovative paradigm that tailors these principles for music synthesis and source separation.

Our work is innovative because it shows how a single model can tackle two seemingly unrelated tasks. Usually, models designed for source separation are not trained even for music generation. Our proposed Multi-Source Diffusion Model (MSDM) solution addresses this lacuna.

Our work also highlighted our model's capability for source imputation, for instance, presents a promising direction where models can generate complementary channels in alignment with existing ones. Our innovative technique of leveraging the Dirac delta functions further aids in capturing the intrinsic relationships between channels and their mixtures.

Source imputation is the ability to conceive complementary sources based on existing ones that mirror the very essence of artistic creation. Imagine a maestro, with a symphony in mind, composing pieces for individual instruments, ensuring each stands out in its melody and aligns with the collective harmony. Our model aims to echo this idea. The theoretical strength of our approach found its practical manifestation as we pitted MSDM against contemporary models. Employing the Slakh2100 dataset for musical source separation and generation, our pursuit was not merely to compete but to accentuate the latent potentials of our model. The novel technique, harnessing Dirac delta functions, accentuated our model's prowess, reinforcing our conviction in its capabilities.

In retrospect, this chapter is not just a culmination of rigorous research, innovative techniques, and empirical evaluations. It is a narrative that reinforces the belief that when foundational principles, such as those of the diffusion framework, are married with cutting-edge methodologies of deep learning, the results can transcend traditional boundaries. As deep learning and audio processing evolve, we hope our contributions serve as a cornerstone, inspiring further innovations and fostering a deeper understanding of the intricate dance between sound sources. It is our ardent wish that this work, while a significant stride, is but the first step in a journey where the horizons of audio modeling are continually expanded and redefined.

Chapter 5

Conclusions

The journey of this thesis, which began at the intersection of deep learning and audio processing, has traversed a broad spectrum of research, innovation, and application. This expedition into deep learning, with a keen focus on audio, has revealed the transformative potential of these techniques, their adaptability, and their capacity for innovation. This trip has been enriching and enlightening, showing how deep learning can enhance our understanding and manipulation of audio signals. The sound and music, particularly in human civilization, have been of primordial importance. The ability to perceive, interpret, and manipulate sounds has been a cornerstone of human evolution, playing a crucial role in communication, expression, and social cohesion. With the advent of digital technology, sound manipulation has evolved from a purely physical process to a complex interplay of mathematical and computational techniques. The introduction of deep learning into this domain has further revolutionized our ability to analyze, understand, and generate sounds, opening up new possibilities for artistic expression, scientific exploration, and technological innovation.

The research presented in this thesis has covered a broad range of topics, from the application of deep extractors to the use of autoregressive models for audio source separation. Furthermore, we delved into the innovative domain of diffusion models, leveraging them for music generation and source separation.

Throughout this journey, we have consistently demonstrated the power and versatility of deep learning techniques in handling complex audio data. Our research has shown that these techniques can be effectively applied to a wide range of audio tasks, from the separation of marine sounds to the detection of singing voices in music tracks. We have dug into unsupervised learning, exploring the potential of Bayesian inference in the latent domain for source separation. We have also investigated using latent autoregressive models for source separation, demonstrating their ability to adapt to complex tasks without requiring exhaustive fine-tuning. Furthermore, we have even ventured into the world of diffusion models, showcasing their potential for simultaneous music generation and separation.

In addition to these technical contributions, this thesis has shed light on the broader implications of deep learning for audio processing. The ability to separate and manipulate individual sound sources has far-reaching implications for music production, music information retrieval, and environmental monitoring. Moreover, the capacity to generate and modify music using deep learning models opens significant strides made in this thesis. The voyage of deep learning in audio processing still needs to be completed, and the field is still ripe with challenges and opportunities for further research and innovation. As deep learning techniques evolve and improve, we expect to see even more sophisticated and powerful audio processing tools.

In conclusion, this thesis has been a testament to the transformative potential of deep learning in audio processing. Through a combination of theoretical exploration, innovative algorithm development, and empirical evaluation, we have demonstrated the power of deep learning to enhance our understanding and manipulation of sound. As we continue to push the boundaries of what is possible in this exciting field, we look forward to a future where deep learning is an integral part of our auditory experience, enriching our lives with sound in ways we can only begin to imagine.

Bibliography

[1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[2] Jonathan Allen. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238, 1977.

[3] Naoya Takahashi an. Mmdenselstm: An efficient combination of convolutional and recurren. *ArXiv preprint*, 2018.

[4] Joakim Andén and Stéphane Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

[5] Francis Bach and Michael Jordan. Blind one-microphone speech separation: A spectral learning approach. *Advances in neural information processing systems*, 17, 2004.

[6] Pijanowski BC, Villanueva-Rivera LJ, Dumyahn SL, Farina A, Krause BL, Napoletano BM, Gage SH, and Pieretti. Soundscape ecology: the science of sound in the landscape. *Bioscience*, page 203–216, 2006.

[7] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[8] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation. *arXiv preprint arXiv:2209.03143*, 2022.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Proc. NeurIPS*, 33:1877–1901, 2020.

[10] Rodrigo Castellon, Chris Donahue, and Percy Liang. Codified audio language modeling learns useful representations for music information retrieval. *arXiv preprint arXiv:2107.05677*, 2021.

[11] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

[12] Kin Wai Cheuk, Ryosuke Sawata, Toshimitsu Uesaka, Naoki Murata, Naoya Takahashi, Shusuke Takahashi, Dorien Herremans, and Yuki Mitsufuji. Diffroll: Diffusion-based generative music transcription with unsupervised pretraining capability. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[13] Keunwoo Choi, György Fazekas, and Mark B. Sandler. Automatic tagging using deep convolutional neural networks. *CoRR*, abs/1606.00298, 2016.

[14] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *CoRR*, abs/1703.09179, 2017.

[15] Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung. Lasaft: Latent source attentive frequency transformation for conditioned source separation, 2020.

[16] Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung. Lasaft: Latent source attentive frequency transformation for conditioned source separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175. IEEE, 2021.

[17] Dena J Clink, Isabel Kier, Abdul Hamid Ahmad, and Holger Klinck. A workflow for the automated detection and classification of female gibbon calls from long-term acoustic recordings. *Frontiers in Ecology and Evolution*, 11:28, 2023.

[18] Pierre Comon. Independent Component Analysis, a new concept? *Signal Processing*, 1994.

[19] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

[20] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[21] Alexandre Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[22] Alexandre Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[23] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*, 2019.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009.

[25] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[26] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020.

[27] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[28] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *ArXiv*, abs/2112.07068, 2021.

[29] Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, et al. Singsong: Generating musical accompaniments from singing. *arXiv preprint arXiv:2301.12662*, 2023.

[30] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2019.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[33] Shaked Dovrat, Eliya Nachmani, and Lior Wolf. Many-speakers single channel speech separation with optimal permutation training. In *Interspeech*, 2021.

[34] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain, 2019.

[35] Albert Einstein et al. On the motion of small particles suspended in liquids at rest required by the molecular-kinetic theory of heat. *Annalen der physik*, 17(549-560):208, 1905.

[36] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165:113679, 2021.

[37] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, pages 1068–1077. PMLR, 2017.

[38] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proc. CVPR*, pages 12873–12883, 2021.

[39] Harvell CD et al. Emerging marine diseases—climate links and anthropogenic factors. *Science*, page 1505–1510, 1999.

[40] Adolph Fick. On liquid diffusion. *Journal of Membrane Science*, 100(1):33–38, 1995.

[41] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *Proc. of DAFX*, volume 10, 2010.

[42] Seth* Forsgren and Hayk* Martiros. Riffusion - Stable diffusion for real-time music generation, 2022.

[43] Hiromasa Fujihara and Masataka Goto. Lyrics-to-audio alignment and its application. In *Multimodal Music Processing*, 2012.

[44] Douglas Gillespie, David K Mellinger, Jonathan Gordon, David McLaren, Paul Redmond, Ronald McHugh, Philip Trinder, Xiao-Yan Deng, and Aaron Thode. Pamguard: Semiautomated, open source software for real-time acoustic detection and localization of cetaceans. *The Journal of the Acoustical Society of America*, 125(4):2547–2547, 2009.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Proc. NIPS*, 27, 2014.

[47] Emad M Grais, Mehmet Umut Sen, and Hakan Erdogan. Deep neural networks for single channel source separation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3734–3738. IEEE, 2014.

[48] Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen, Duolin Wang, Dong Xu, and Jianlin Cheng. Diffusion models in bioinformatics: A new wave of deep learning revolution in action. *arXiv preprint arXiv:2302.10907*, 2023.

[49] Enric Gusó, Jordi Pons, Santiago Pascual, and Joan Serrà. On loss functions and evaluation metrics for music source separation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 306–310. IEEE, 2022.

[50] Enric Gusó, Jordi Pons, Santiago Pascual, and Joan Serrà. On loss functions and evaluation metrics for music source separation. In *Proc. ICASSP*, pages 306–310, 2022.

[51] T. Halperin, A. Ephrat, and Y. Hoshen. Neural separation of observed and unobserved distributions. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4548–4557, 2019.

[52] Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. In *International Society for Music Information Retrieval Conference*, 2022.

[53] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[54] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. NeurIPS*, volume 30, 2017.

[55] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proc. NeurIPS*, 33:6840–6851, 2020.

[56] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[57] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proc. ICLR*, 2020.

[58] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *Proc. ICPR*, pages 2366–2369, 2010.

[59] Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60. IEEE, 2012.

[60] Po-Sen Huang, Minje Kim, Mark A. Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *Proc. ISMIR*, 2014.

[61] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio and natural language. In *International Society for Music Information Retrieval Conference*, 2022.

[62] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[63] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[64] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[65] Satoshi Innami and Hiroyuki Kasai. Nmf-based environmental sound source separation using time-variant gain features. *Computers & Mathematics with Applications*, 64(5):1333–1342, 2012.

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

[67] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *ISMIR*, 2017.

[68] Vivek Jayaram and John Thickstun. Source separation with deep generative priors. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, 2020.

[69] Vivek Jayaram and John Thickstun. Parallel and flexible sampling from autoregressive models via langevin dynamics, 2021.

[70] Vivek Jayaram and John Thickstun. Parallel and flexible sampling from autoregressive models via langevin dynamics. In *Proc. ICML*, pages 4807–4818. PMLR, 2021.

[71] Jia-jia Jiang, Ling-ran Bu, Fa-jie Duan, Xian-quan Wang, Wei Liu, Zhong-bo Sun, and Chun-yue Li. Whistle detection and classification for whales based on convolutional neural networks. *Applied Acoustics*, 150:169–178, 2019.

[72] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

[73] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

[74] Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey. Universal sound separation. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 175–179. IEEE, 2019.

[75] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[76] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[77] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014.

[78] Durk P Kingma and Yann LeCun. Regularized estimation of image statistics by score matching. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[79] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1901–1913, 2017.

[80] Qiuqiang Kong, Yong Xu, Wenwu Wang, Philip J. B. Jackson, and Mark D. Plumbley. Single-channel signal separation and deconvolution with generative adversarial networks. In *Proc. IJCAI*, page 2747–2753. AAAI Press, 2019.

[81] Qiuqiang Kong, Yong Xu, Wenwu Wang, Philip J. B. Jackson, and Mark D. Plumbley. Single-channel signal separation and deconvolution with generative adversarial networks. In *Proc. IJCAI*, page 2747–2753. AAAI Press, 2019.

[82] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[83] Wouter Kool, Herke van Hoof, and Max Welling. Ancestral gumbel-top-k sampling for sampling without replacement. *Journal of Machine Learning Research*, 21(47):1–36, 2020.

[84] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

[85] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 2011.

[86] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54. Springer, 2016.

[87] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[88] Junhyeok Lee and Seungu Han. NU-Wave: A Diffusion Probabilistic Model for Neural Audio Upsampling. In *Proc. Interspeech 2021*, pages 1634–1638, 2021.

[89] Kyungyun Lee, Keunwoo Choi, and Juhan Nam. Revisiting singing voice detection: a quantitative review and the future outlook. *CoRR*, abs/1806.01180, 2018.

[90] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In IEEE, editor, *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, Brisbane, Australia, April 2015.

[91] Bernhard Lehner, Gerhard Widmer, and Sebastian Bock. A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 21–25, 2015.

[92] Pu Li, Marie A Roch, Holger Klinck, Erica Fleishman, Douglas Gillespie, Eva-Marie Nosal, Yu Shiu, and Xiaobai Liu. Learning stage-wise gans for whistle extraction in time-frequency spectrograms. *IEEE Transactions on Multimedia*, 2023.

[93] Tzu-Hao Lin and Yu Tsao. Listening to the deep: Exploring marine soundscape variability by information retrieval techniques. In *2018 OCEANS-Mts/IEEE Kobe Techno-Oceans (OTO)*, pages 1–6. IEEE, 2018.

[94] Tzu-Hao Lin, Yu Tsao, and Tomonari Akamatsu. Comparison of passive acoustic soniferous fish monitoring with supervised and unsupervised approaches. *The Journal of the Acoustical Society of America*, 143(4):EL278–EL284, 2018.

[95] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

[96] Jen-Yu Liu and Yi-Hsuan Yang. Denoising auto-encoder with recurrent skip connections and residual regression for music source separation. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 773–778. IEEE, 2018.

[97] Jen-Yu Liu and Yi-Hsuan Yang. Denoising auto-encoder with recurrent skip connections and residual regression for music source separation, 2018.

[98] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, December 2015.

[99] Francesc Lluís, Jordi Pons, and Xavier Serra. End-to-end music source separation: Is it possible in the waveform domain? In *INTERSPEECH*, pages 4619–4623, 2019.

[100] Francesc Lluís, Jordi Pons, and Xavier Serra. End-to-end music source separation: Is it possible in the waveform domain? In *INTERSPEECH*, pages 4619–4623, 2019.

[101] Suhas Lohit, Qiao Wang, and Pavan Turaga. Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12426–12435, 2019.

[102] Yen-Ju Lu, Yu Tsao, and Shinji Watanabe. A study on speech enhancement based on diffusion probabilistic model. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 659–666. IEEE, 2021.

[103] Wenyu Luo, Wuyi Yang, and Yu Zhang. Convolutional neural network for detecting odontocete echolocation clicks. *The Journal of the Acoustical Society of America*, 145(1):EL7–EL12, 2019.

[104] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

[105] Shahar Lutati, Eliya Nachmani, and Lior Wolf. Separate and diffuse: Using a pretrained diffusion model for improving source separation. *arXiv preprint arXiv:2301.10752*, 2023.

[106] Southworth M. The sounds: our sonic environment and the tuning of the world. *Behav.*, page 49–70, 1969.

[107] Stéphane Mallat. *A wavelet tour of signal processing.* Elsevier, 1999.

[108] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Learning music audio representations via weak language supervision. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 456–460. IEEE, 2022.

[109] Ethan Manilow, Curtis Hawthorne, Cheng-Zhi Anna Huang, Bryan Pardo, and Jesse Engel. Improving source separation by explicitly modeling dependencies between sources. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 291–295. IEEE, 2022.

[110] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[111] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[112] Matt McVicar, Daniel P. W. Ellis, and Masataka Goto. Leveraging repetition for improved automatic lyric transcription in popular music. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3117–3121, 2014.

[113] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

[114] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6), 2016.

[115] Gabriel Meseguer-Brocal and Geoffroy Peeters. Conditioned-U-Net: Introducing a Control Mechanism in the U-Net for Multiple Source Separations. *arXiv:1907.01277 [cs, eess]*, 2019. arXiv: 1907.01277.

[116] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[117] T. Aran Mooney, Lucia Di Iorio, Marc Lammers, Tzu-Hao Lin, Sophie L. Nedelec, Miles Parsons, Craig Radford, Ed Urban, and Jenni Stanley. Listening forward: approaching marine biodiversity assessments using acoustic methods. *Royal Society Open Science*, 7(8):201287, 2020.

[118] Vivek Narayanaswamy, Jayaraman J. Thiagarajan, Rushil Anirudh, and Andreas Spanias. Unsupervised audio source separation using generative priors, 2020.

[119] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. Multichannel audio source separation with deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1652–1664, 2016.

[120] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. Multichannel music separation with deep neural networks. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1748–1752. IEEE, 2016.

[121] OpenAI. Gpt-4 technical report, 2023.

[122] Balvanera P, Pfisterer AB, Buchmann N, He JS, Nakashizuka T, Raffaelli D, and Schmid B. Qantifying the evidence for biodiversity effects on ecosystem functioning and services. *Ecol. Lett.*, page 1146–1156, 2006.

[123] G. Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

[124] Hyunji Hayley Park, Yogarshi Vyas, and Kashif Shah. Efficient classification of long documents using transformers, 2022.

[125] Miles J. G. Parsons, Tzu-Hao Lin, T. Aran Mooney, Christine Erbe, Francis Juanes, Marc Lammers, Songhai Li, Simon Linke, Audrey Looby, Sophie L. Nedelec, Ilse Van Opzeeland, Craig Radford, Aaron N. Rice, Laela Sayigh, Jenni Stanley, Edward Urban, and Lucia Di Iorio. Sounding the call for a global library of underwater biological sounds. *Frontiers in Ecology and Evolution*, 10, 2022.

[126] Santiago Pascual, Gautam Bhattacharya, Chunghsin Yeh, Jordi Pons, and Joan Serrà. Full-band general audio synthesis with score-based diffusion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[127] N. Pieretti, A. Farina, and D. Morri. A new methodology to infer the singing activity of an avian community: The acoustic complexity index (aci). *Ecological Indicators*, 11(3):868–873, 2011.

[128] Genís Plaja-Roglans, Miron Marius, and Xavier Serra. A diffusion-inspired training strategy for singing voice extraction in the waveform domain. In *Proc. of the 23rd Int. Society for Music Information Retrieval*, 2022.

[129] Emilian Postolache, Giorgio Mariani, Michele Mancusi, Andrea Santilli, Luca Cosmo, and Emanuele Rodolà. Latent autoregressive source separation. In *Proc. AAAI*, AAAI Press, 2023.

[130] Emilian Postolache, Giorgio Mariani, Michele Mancusi, Andrea Santilli, Luca Cosmo, and Emanuele Rodolà. Latent autoregressive source separation. In *Proc. AAAI*, AAAI Press, 2023.

[131] Emilian Postolache, Jordi Pons, Santiago Pascual, and Joan Serrà. Adversarial permutation invariant training for universal sound separation. *arXiv preprint arXiv:2210.12108*, 2022.

[132] Emilian Postolache, Jordi Pons, Santiago Pascual, and Joan Serrà. Adversarial permutation invariant training for universal sound separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[133] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[134] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[135] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017.

[136] Zafar Rafii and Bryan Pardo. Repeating pattern extraction technique (repet): A simple method for music/voice separation. *IEEE transactions on audio, speech, and language processing*, 21(1):73–84, 2012.

[137] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[138] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proc. ICML*, pages 8821–8831. PMLR, 2021.

[139] Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1885–1888, 2008.

[140] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[141] D Raj Reddy et al. Speech understanding systems: A summary of results of the five-year research effort. *Department of Computer Science. Camegie-Mell University, Pittsburgh, PA*, 17:138, 1977.

[142] Dario Rethage, Jordi Pons, and Xavier Serra. A wavenet for speech denoising. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5069–5073. IEEE, 2018.

[143] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[144] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[145] Simon Rouard and Gaëtan Hadjeres. CRASH: raw audio score-based generative modeling for controllable high-resolution drum sound synthesis. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, pages 579–585, 2021.

[146] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. Sdr – half-baked or well done? In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630, 2019.

[147] Sam Roweis. One microphone source separation. *Advances in neural information processing systems*, 13, 2000.

[148] Sam T. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, 2000.

[149] Koichi Saito, Naoki Murata, Toshimitsu Uesaka, Chieh-Hsin Lai, Yuhta Takida, Takao Fukui, and Yuki Mitsufuji. Unsupervised vocal dereverberation with diffusion-based generative models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[150] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[151] Victor Sanh, Albert Webson, Colin Raffel, et al. Multitask prompted training enables zero-shot task generalization. In *Proc. ICLR*, 2022.

[152] Ryosuke Sawata, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Takashi Shibuya, Shusuke Takahashi, and Yuki Mitsufuji. A versatile diffusion-based generative refiner for speech enhancement. *arXiv preprint arXiv:2210.17287*, 2022.

[153] Robin Scheibler, Youna Ji, Soo-Whan Chung, Jaeuk Byun, Soyeon Choe, and Min-Seok Choi. Diffusion-based generative speech source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[154] Jan Schlüter and Thomas Grill. Exploring data augmentation for improved singing voice detection with neural networks. In *ISMIR*, 2015.

[155] Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

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

[157] Joan Serrà, Santiago Pascual, Jordi Pons, R Oguz Araz, and Davide Scaini. Universal speech enhancement with score-based diffusion. *arXiv preprint arXiv:2206.03065*, 2022.

[158] Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[159] Sonali Sharma, Manoj Diwakar, Prabhishek Singh, Amrendra Tripathi, Chandrakala Arya, and Shilpi Singh. A review of neural machine translation based on deep learning techniques. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–5. IEEE, 2021.

[160] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[161] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[162] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403, 2021.

[163] P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, (3), 2014.

[164] Paris Smaragdis, Cedric Fevotte, Gautham J Mysore, Nasser Mohammadiha, and Matthew Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3):66–75, 2014.

[165] Julius O Smith III. Spectral audio signal processing. *(No Title)*, 2011.

[166] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[167] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015.

[168] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[169] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907, 2019.

[170] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. ICLR*, 2021.

[171] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[172] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. In *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK*, pages 293–305, 2018.

[173] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-unmix-a reference implementation for music source separation. *Journal of Open Source Software*, 4(41):1667, 2019.

[174] Dan Stowell, Michael D Wood, Hanna Pamuła, Yannis Stylianou, and Hervé Glotin. Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge. *Methods in Ecology and Evolution*, 10(3):368–380, 2019.

[175] Y Cem Subakan and Paris Smaragdis. Generative adversarial source separation. In *Proc. ICASSP*, pages 26–30. IEEE, 2018.

[176] Jérôme Sueur, Sandrine Pavoine, Olivier Hamerlynck, and Stéphanie Duvail. Rapid acoustic survey for biodiversity appraisal. *PloS one*, 3(12), 2008.

[177] Yi-Jen Sun, Shih-Ching Yen, and Tzu-Hao Lin. soundscape_ir: A source separation toolbox for exploring acoustic diversity in soundscapes. *Methods in Ecology and Evolution*, 13(11):2347–2355, 2022.

[178] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. In *2018 16th International workshop on acoustic signal enhancement (IWAENC)*, pages 106–110. IEEE, 2018.

[179] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. In *Proc. IWAENC*, pages 106–110, 2018.

[180] Naoya Takahashi and Yuki Mitsufuji. Multi-scale multi-band densenets for audio source separation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 21–25. IEEE, 2017.

[181] Naoya Takahashi and Yuki Mitsufuji. D3net: Densely connected multidilated densenet for music source separation, 2020.

[182] Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji. Recursive speech separation for unknown number of speakers. *arXiv preprint arXiv:1904.03065*, 2019.

[183] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

[184] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[185] Stefan Uhlich, Franck Giron, and Yuki Mitsufuji. Deep neural network based instrument extraction from music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2135–2139. IEEE, 2015.

[186] Stefan Uhlich, Franck Giron, and Yuki Mitsufuji. Deep neural network based instrument extraction from music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015.

[187] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 261–265. IEEE, 2017.

[188] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017.

[189] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[190] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Proc. NeurIPS*, 29, 2016.

[191] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[192] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017.

[193] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. In *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, page 125, 2016.

[194] CJ van Rijsbergen. *Information retrieval.* Butterworth, London, UK, 1979.

[195] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[196] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[197] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[198] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[199] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *CoRR*, abs/2109.01652, 2021.

[200] Scott Wisdom, Hakan Erdogan, Daniel P. W. Ellis, Romain Serizel, Nicolas Turpault, Eduardo Fonseca, Justin Salamon, Prem Seetharaman, and John R. Hershey. What's all the fuss about free universal sound separation data? In *Proc. ICASSP*, pages 186–190, 2021.

[201] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. Unsupervised sound separation using mixture invariant training. 33:3846–3857, 2020.

[202] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. Unsupervised sound separation using mixture invariant training. In *Proc. NeurIPS*, volume 33, pages 3846–3857, 2020.

[203] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. Unsupervised sound separation using mixture invariant training. *Advances in Neural Information Processing Systems*, 33:3846–3857, 2020.

[204] Jie Xie, Michael Towsey, Jinglan Zhang, and Paul Roe. Adaptive frequency scaled wavelet packet decomposition for frog call classification. *Ecological Informatics*, 32:134–144, 2016.

[205] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25, 2012.

[206] Yilun Xu, Yang Song, Sahaj Garg, Linyuan Gong, Rui Shu, Aditya Grover, and Stefano Ermon. Anytime sampling for autoregressive models via ordered autoencoding. *arXiv preprint arXiv:2102.11495*, 2021.

[207] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[208] Hao Yang, Junyang Lin, An Yang, Peng Wang, Chang Zhou, and Hongxia Yang. Prompt tuning for generative multimodal pretrained models. *arXiv preprint arXiv:2208.02532*, 2022.

[209] Chin-Yun Yu, Sung-Lin Yeh, György Fazekas, and Hao Tang. Conditioning and sampling in variational diffusion models for speech super-resolution. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[210] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[211] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[212] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[213] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 2022.

[214] Tong Zhang. Automatic singer identification. In *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, volume 1, pages I–33, 2003.

[215] Xulong Zhang, Jiale Qian, Yi Yu, Yifu Sun, and Wei Li. Singer identification using deep timbre feature learning with KNN-Net. *CoRR*, abs/2102.10236, 2021.

[216] Xulong Zhang, Yi Yu, Yongwei Gao, Xi Chen, and Wei Li. Research on singing voice detection based on a long-term recurrent convolutional network with vocal separation and temporal smoothing. *Electronics*, 9(9), 2020.

[217] Zhenbin Zhang and Paul R White. A blind source separation approach for humpback whale song separation. *The Journal of the Acoustical Society of America*, 141(4):2705–2714, 2017.

[218] Ge Zhu, Jordan Darefsky, Fei Jiang, Anton Selitskiy, and Zhiyao Duan. Music source separation with generative flow. *IEEE Signal Processing Letters*, 29:2288–2292, 2022.

[219] Hao Zou, Zae Myung Kim, and Dongyeop Kang. Diffusion models in nlp: A survey. *arXiv preprint arXiv:2305.14671*, 2023.