


Master ATIAM - 2022/23

Internship at Deezer - 5th of March to 25th of August, 2023

Music Source Separation: A Generative Approach

Ivan Meresman Higgs

Supervisors

Gabriel Meseguer Brocal

Romain Hennequin

Masters Thesis Internship Report - 25/08/2023



Abstract

Music source separation is the decomposition of an audio recording into the recordings of its individual sources. It plays an integral role in applications ranging from musicological tasks such as transcription to practical industrial applications such as karaoke. In this dissertation, we explore the complex task of music source separation, examining the nuances of current challenges in the field and presenting novel methods to overcome them. Although state-of-the-art deep learning approaches have made significant progress, they often introduce distortions, artefacts and other perceptual inaccuracies. We propose a paradigm shift towards prioritising auditory experience over high precision in the reconstruction of a reference target. Our research presents a generative approach to the task, using models currently employed for automatic music generation to approach source separation in a less conditioned way. The primary goal is to improve sound quality, even if this means deviating from the original target, thus examining the essence of what makes audio "good" or "authentic" without being tied to direct waveform or spectrogram comparisons. To this end, we use token-based audio generation and exploit neural codec architectures to achieve a balance between fidelity of reconstruction of the original source and perceived audio quality. While our research has had its challenges, partly due to the rapidly evolving field of automatic music generation and the early-stage nature of some of the proposed methods, our work serves as a fundamental step in understanding the potential of token-based music generation in highly conditioned tasks. The work concludes by highlighting the need for a holistic approach to music source separation, one that aligns with the human auditory experience and expands the horizons of musicians and listeners alike.

Keywords: Source separation, music generation, enhancement, neural codecs, token-based generation

Résumé

La séparation de sources musicales est la décomposition d'un enregistrement audio en plusieurs pistes correspondant à ses sources individuelles. Elle joue un rôle important dans des applications allant des tâches musicologiques telles que la transcription automatique sur partition à des applications industrielles comme le karaoké. Dans ce mémoire, nous explorons la tâche complexe de la séparation de sources musicales, en examinant les défis actuels dans le domaine et en présentant de nouvelles méthodes pour les surmonter. Bien que les approches d'apprentissage profond de l'état de l'art aient fait des progrès significatifs, elles introduisent souvent des distorsions, des artefacts et autres imperfections audibles. Nous proposons un changement de paradigme en donnant la priorité à l'expérience auditive plutôt qu'à une grande précision dans la reconstruction exacte d'une source de référence. Notre recherche présente une approche générative, en utilisant des modèles actuellement employés pour la génération automatique de musique afin d'aborder la séparation de sources d'une manière moins conditionnée. L'objectif principal est d'améliorer la qualité du son, même si

cela implique de s'écarter de la référence originale, ce qui permet d'examiner l'essence même de ce qui rend un son "bon" ou "authentique" sans être lié à des comparaisons directes de formes d'ondes ou de spectrogrammes. À cette fin, nous utilisons la génération audio basée sur les *tokens* et exploitons les architectures de codecs neuronaux pour parvenir à un équilibre entre la fidélité de reconstruction de la source originale et la qualité audio perçue. Bien que notre recherche ait connu des difficultés, en partie dues à l'évolution rapide du domaine de la génération automatique de musique et au manque de maturité de certaines des méthodes utilisées, notre travail constitue une étape fondamentale dans la compréhension du potentiel de la génération de musique basée sur des *tokens* dans des tâches fortement conditionnées. Ce travail conclut en soulignant la nécessité d'une approche holistique de la séparation de sources musicales, qui s'aligne sur l'expérience auditive humaine et élargit les horizons des musiciens et des auditeurs.

Resumen

La separación de fuentes musicales es la descomposición de una grabación de audio en las grabaciones de sus fuentes sonoras individuales. Desempeña un papel integral en aplicaciones que van desde tareas musicológicas como la transcripción hasta usos industriales prácticos como el karaoke. En esta tesis, exploramos la compleja tarea de la separación de fuentes musicales, examinando los retos actuales en este campo y presentando métodos novedosos para superarlos. Aunque los enfoques del estado del arte basados en aprendizaje profundo han hecho progresos significativos, a menudo introducen distorsiones, artefactos y otras imperfecciones audibles. Proponemos un cambio de paradigma para dar prioridad a la experiencia auditiva, mas que a la alta precisión en la reconstrucción de un objetivo de referencia. Nuestra investigación presenta un enfoque generativo de la tarea, utilizando modelos empleados actualmente para la generación automática de música para abordar la separación de fuentes de una forma menos condicionada. El objetivo principal es mejorar la calidad del sonido, incluso si esto significa desviarse del target original, examinando así la esencia de lo que hace que un audio sea "bueno" o "auténtico" sin estar atado a comparaciones directas de forma de onda o espectrograma. Para ello, utilizamos la generación de audio basada en *tokens* y explotamos arquitecturas de códecs neuronales para lograr un equilibrio entre la fidelidad de la reconstrucción del objetivo original y la calidad de audio percibida. Aunque nuestra investigación ha tenido sus retos, en parte debido a la rápida evolución del campo de la generación automática de música y a la naturaleza incipiente de algunos de los métodos propuestos, nuestro trabajo sirve como paso fundamental para comprender el potencial de la generación de música basada en tokens en tareas altamente condicionadas. El trabajo concluye destacando la necesidad de un enfoque holístico de la separación de fuentes musicales, que se alinee con la experiencia auditiva humana y amplíe los horizontes tanto para músicxs, como para oyentes.

Acknowledgements

This year of ATIAM Masters has been made possible in part by the support of the Uruguayan National Agency for Research and Innovation, through its "International Graduate Studies" grant programme.

I'd like to thank the Deezer research team for welcoming me so warmly, and for sharing so much knowledge, music, and good times in general. I'd especially like to thank Gabi and Romain for their guidance, patience and great advice, always in the kindest possible way.

I'd also like to thank my promo ATIAM for sharing really great times, really interesting conversations, and for their patience with my crappy French.

This is the first time I've written an Acknowledgements section, so I'm going to thank my parents for their endless love and support, always.

Finally, thanks to Fran, for being the most loving, caring, and fun partner in the world, as well as the best proofreader in the galaxy.

Acronyms

SOTA state-of-the-art

MIR Music Information Retrieval

NMF Non-negative Matrix Factorization

SDR Source-to-Distortion Ratio

SIR Source-to-Interference Ratio

SAR Source-to-Artifact Ratio

SI-SDR Scale-Invariant Source-to-Distortion Ratio

FAD Fréchet Audio Distance

MUSHRA Multiple Stimuli with Hidden Reference and Ancho

MOS Mean Opinion Score

GAN Generative Adversarial Network

MFCC Mel-Frequency Cepstral Coefficients

MSG Make-it-Sound-Good

VAE Variational AutoEncoder

VQ Vector Quantization

RVQ Residual Vector Quantization

STFT Short Time Fourier Transform

DAC Descript Audio Codec

MDS MultiDimensional Scaling

PCA Principal Component Analysis

GT ground truth

MSE Mean Square Error

PQMF Pseudo Quadrature Mirror Filters

Contents

1 Introduction

Music has always played a prominent role in human culture, and with the advent of digital streaming platforms, it has witnessed an unparalleled rise in its ubiquity in our daily lives. The field of audio signal processing now offers limitless possibilities for these platforms to introduce innovative avenues of engagement. Currently, these modes of interaction are in a phase of rapid evolution, constantly shifting. Whether it's activities such as karaoke for casual enthusiasts, or delving into intricate endeavours such as music remixing and instrument identification, the demand from music providers for tools that deconstruct and manipulate musical compositions is steadily increasing. At the core of many of these applications lies the fundamental task of music source separation.

1.1 Context and Motivation

Music source separation is fundamentally the decomposition of a given audio scene, typically a music recording, into individual sound sources, be they vocals or various instruments. Interest in this task has increased since the beginning of the twenty-first century, and it has become a central problem in machine listening. Its applications are numerous, ranging from music remixing, or Music Information Retrieval (MIR) tasks such as instrument labelling or transcription, to practical industrial applications such as karaoke and play-along applications.

One of the standard approaches to this task, popularised in part by easily accessible datasets and competitions (MUSBD18[1] and MusicDMX challenge[2] respectively), is to separate commercial-quality stereo recordings into four stems: vocals, bass, drums and others.

Current state-of-the-art (SOTA) source separation systems [3][4][5], which use data-driven deep learning approaches, achieve great results, producing separated outputs that are often of very good quality and intelligibility. On the other hand, they can produce source estimates that contain perceptible distortions and artefacts, such as high-frequency noise, source leakage interference, unnatural transients, loudness variations and inconsistencies, or missing harmonics; especially with percussive instruments or in instrumental sections. Frequential masking between instruments, or spatial components of the recording process, as well as modern trends in mixing and post-processing, contribute to these artefacts, making separation (increasingly) difficult. Furthermore, the modifications introduced in these processes alter the inherent properties of individual sound sources, making it also challenging to extract and separate them, not only in their original form but also in a minimally intelligible form.

These artefacts are a major hindrance for many artistic or MIR applications, and although the current focus of the source separation community is to continue to modify architectures to obtain small increases in performance as measured by the traditional metrics, this may not be the best approach, as the systems continue to acheive small gains in the metrics, but the perceived quality is the same or worse. There are already many works[6][7][8] looking at the effectiveness of these metrics and their relation to human perception, and our work is motivated by the idea of focusing

on improving sound quality, and not so much on reconstructing the target.

In addition to distortion and artefacts, the task of source separation in music presents numerous additional challenges, such as generalising from the 4-stem task to a wide variety of instruments. Because each instrument has a unique timbre and acoustic properties that define its sound, training a model that works equally well for different instruments would require large datasets with multi-track recordings that include the instruments in different contexts. Consequently, we're also motivated to look at approaches that can be generalised or adapted without large amounts of multi-track data.

There are several advantages to a generative approach: it can provide better-sounding karaoke or play-along applications, improving the user experience for budding and experienced musicians alike; and it can offer new applications in education, allowing students studying specific instruments within a composition to enjoy good-quality sounds while gaining insight into musical structures, styles and techniques. In addition, the tools developed can open up new possibilities for artists when mixing live, or even for listeners who may be exposed to new ways of engaging with music.

1.2 Objectives

Inspired by the current predicament, where the models that attempt to improve objective metrics scores manage to do so without improving our perceived quality or removing artefacts, we suggest addressing the source separation task with a generative approach. By applying models typically used for generative tasks (automatic generation of new music in varied genres and artist styles[9][10][11][12]) we can approach the problem of source separation in a "less deterministic" manner. This method can result in a more pleasant-sounding isolated source, even when objectively further away from the ground truth. As depicted in Figure 7*, this approach, which involves a second processing step, enhances the separation and removes artefacts, even if the guitar amplifier changes in the process.

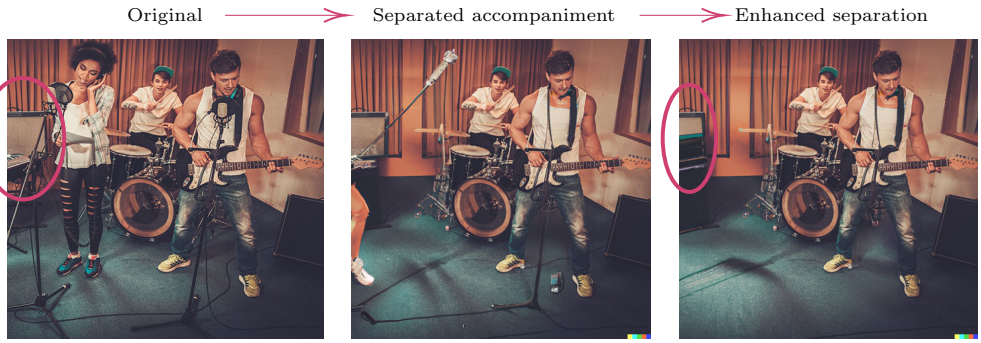


Figure 1: Music source separation enhancement illustration

*Image separation performed with DALL-E

Generative models have already been used for tasks such as music enhancement [13], compression [14][15][16], denoising[17][18][19], end-to-end separation [20][21], and separation enhancement[22]. The latter task is the most similar to our work. Our aim is to explore and evaluate the capabilities of SOTA generative models for separation, for enhancement of the output of existing source separation methods, and for other forms of strongly conditioned music generation.

Our primary focus in this research is on the 4-stem task, i.e. vocals, bass, drums and others. However, we remain attentive to the wider implications of our methods. In particular, we are interested in analysing how the techniques we develop can be applied to a broader range of instruments.

Throughout our work, we will give significant importance to our own auditory perception of the results. We believe that the sonic quality of the output is the ultimate measure of any source separation system's effectiveness, therefore, rather than relying solely on conventional metrics, we give priority to the experiential, auditory quality of the separated tracks. We have not been able to carry out a standardised subjective evaluation, as would have been desirable.

We propose a paradigm shift in the way we approach music source separation: a generative approach to the task would involve the introduction of novel, more perceptually focused metrics, as has occurred in the field of image generation, and this in turn would stimulate the development of separation systems that prioritise human auditory experience over mathematical accuracy.

Our work intersects with the burgeoning field of audio generation, and we posit that while current research trends exhibit significant interest in the unconditioned and loosely conditioned generation, the real frontier and interest of innovation should be in highly conditioned generation. Such techniques can provide controllable tools that open up new avenues of expression and creativity for musicians, rather than attempting (and failing) to automate the creative processes that make people's life worthwhile and an artist's expertise valuable.

1.3 Deezer

The internship for which this work was carried out was hosted by Deezer's research team at the company's headquarters in Paris. Founded in 2007, Deezer is an international music streaming company that currently serves millions of users in 180 countries. Its catalogue contains over 90 million audio tracks, including music of all genres and geographical origins, as well as audiobooks and podcasts.

Deezer Research, which states its objective as "expanding knowledge about music and how people interact with it", is one of the most active private players at the forefront of audio and music data research. The team's primary mission is "to help make Deezer a cutting-edge audio streaming service, tailored to its customers' needs". Their work focuses on challenges related to the organisation of audio catalogues and human-music interaction, but is not limited to improving the streaming experience. The team also plays an active role in the global scientific community, consistently

contributing their work to international scientific conferences in areas ranging from audio signal processing to natural language understanding and recommender systems. The team prioritises the reproducibility of their work, which they have achieved through extensive sharing of their code and data. They also frequently collaborate with academic institutions by hosting PhD and Masters students.

Deezer Research is well known for its work in music source separation. In 2019, they launched Spleeter, which at the time was the most advanced solution for the task of 4-stem music separation. Since its inception, Spleeter has cemented its place as a foundational tool in the music demixing landscape and currently serves as the baseline in the ongoing Music Demixing Challenge. Its performance and ease of use have led to its adoption in over 500 projects on GitHub, further cementing Deezer Research's reputation for creating impactful and widely applicable technological solutions in the realm of music and audio.

During the internship, the research team consisted of Manuel Moussalam (Director of Research), Romain Hennequin (Head of Research), Marion Baranes, Elena Epure, Benjamin Martin, Bruno Massoni Sguerra, Gabriel Meseguer Brocal, Guillaume Salha Galvan and Viet Anh Tran (Research Scientists), Rodolfo Ripado, Kamil Akesbi, Darick Lean, Dorian Desblancs and Karl Hayek (Research Engineers), Darius Afchar, Yuexuan Kong, Kristina Matrosova and Gaspard Michel (PhD students), Harin Lee (PhD intern), and Paul Chauvin, Rayane Donni, Lilian Marey and myself (Masters interns). The internship was characterised as a research internship on music source separation with a generative head, and the work was carried out under the supervision of Romain Hennequin and Gabriel Meseguer Brocal, trying to extend and continue the legacy of the Deezer's research team's influential work on source separation.

1.4 Report Organisation

This report is organised into nine main sections to ensure a logical flow of the information and to facilitate the reader's understanding.

Following the "Introduction" presented above, we continue with the "State of the Art" section, where we delve into the existing literature and critically analyse both the current methods of source separation and the generative architectures that underpin our approach. This includes a look at diffusion models, generative adversarial networks, and recent innovations in neural codecs and codec-based transformers.

The third section takes the reader through our research design, from the problem formulation to a discussion of the datasets used, the preprocessing separation methods, the current limitations of these separation models, and an overarching view of our experimental design.

Section 4 is dedicated to establishing the baselines. We discuss two previous work approaches, detailing their methodologies and the results of their application. Following this, Section 5 focuses on

our experience in training neural codecs. The methodology behind training the codec is dissected, followed by an in-depth look at the corresponding experiments.

We then move on to one of our main contributions, our research into codec fine-tuning (Section 6). Here we explore aspects such as the codecs' compression interpretability and experiments on fine-tuning the EnCodec codec for enhancement and separation.

In section 7 we present our work leveraging the VampNet transformer for enhancement. This is followed by a final "Discussion" section where we reflect on our findings and the challenges we faced throughout our work.

We finally provide a "Conclusions" section, which aims to summarise our main contributions to the field, highlight the key findings and provide insights into potential future research directions.

2 State of the Art

The work presented in this report lies at the intersection of traditional source separation, and recent generative models based on deep learning architectures. Throughout our research, we draw from the recent advances in the latter to explore innovative approaches to the complex task of source separation. Consequently, this review of the SOTA will predominantly cover these two main axes.

We will first delve into the landscape of modern source separation approaches, highlighting the latest methodologies and benchmarks that are driving progress in the field, which are typically supervised, data-driven methods, built on deep learning architectures. A special emphasis will be placed on the typical objective evaluation paradigm, a common cornerstone in the development of these systems. We will examine its advantages and shortcomings, providing a balanced perspective on its usefulness and its limitations in capturing the nuances of source separation.

We will then consider the field of generative music models, which has seen tremendous growth in the last five years, through various types of architectures and (big) data-driven methods. We will review the comparatively negligible impact these advances have had on the domain of source separation up until the moment. Our focus will be on Generative Adversarial Networks, Diffusion models, and especially token-based transformer methods, based on neural codecs, which will be discussed in detail as they underlie a great number of generative approaches that have surfaced over the last twelve months.

We hope that this review of technologies that underpin novel generative approaches can provide a tantalising glimpse into a possible future direction for work on source separation.

2.1 Source Separation

A classic way to formulate the source separation task[23][24] is the instantaneous additive mixture model. It goes as follows: let I be the number of channels of a recording and J the number of present sources, $s_j(t) \in \mathbf{R}^{I\times 1}$ is the I-channel spatial image of source j, and $x(t) \in \mathbf{R}^{I\times 1}$ the observed I-channel mixture signal. The source separation aims to recover independent sources $s_j(t)$ from the mixture signal $x(t)$, where signals $s_j(t)$ and $x(t)$ are in the time domain and related by

$$x(t) = \sum_{j=1}^{J} s_j(t) \tag{1}$$

It must be said that though the instantaneous additive mixture model is largely dominant, it is not the only model for source separation, and this simple model could indeed be a limitation of current approaches.

As mentioned above, deep learning data-driven supervised models are the current state-of-the-art approaches for 4-stem music source separation($J = 4$ and $I = [1, 2]$ - with the stems being vocal, bass, drums and other); for example, all the baselines and leading entries in the 2021 and 2023 Music Demixing Challenge[2] were deep learning architectures.

It's important to note that our exploration will primarily revolve around these deep learning models, setting aside more classical signal processing unsupervised approaches that were the focus of much research in the first decade of the 21st century, such as methods based on Maximum Likelihood Blind Source Separation[25] or Non-negative Matrix Factorization (NMF)[26][27]. These non-data-driven methods can be more flexible with different instrumentations, recording settings (domain adaptation), or less conventional musical practices, and also easier to interpret, but they have non the less been relegated by the data-driven highest performance separators. These separators can normally be classified in terms of their input/output, as *spectrogram models*[3][28], *waveform models* [29], and *hybrid models* [5], as can be seen in the overview presented in Figure 2.

Spectrogram models utilize a time-frequency representation of the signal $x(f, n) \in \mathbf{C}^{I\times 1}$ as their input, either the amplitude of the spectrogram or the complex-valued spectrogram[4][30]. The output $\hat{s_j}(f, n) \in \mathbf{C}^{I\times 1}$ can be a mask on the spectrogram (Weiner filtering) that uses the phase of the mixture for the sources[3], a complex modulation of the input [31], or the concatenated real and imaginary parts.

Waveform-based models[29][32][33] are directly fed with the raw waveform of the audio, and output an audio waveform directly for each of the sources. In reality, these methods normally perform a learned time-frequency analysis in their first layers through convolutions[29][32], though some works rely on Recurrent Neural Networks at the input and thus do not count with this analysis[33].

It is worth noting that from a theoretical standpoint, there should be no discernible difference between full-spectrogram and waveform models. This would be the result in the case of having infinite

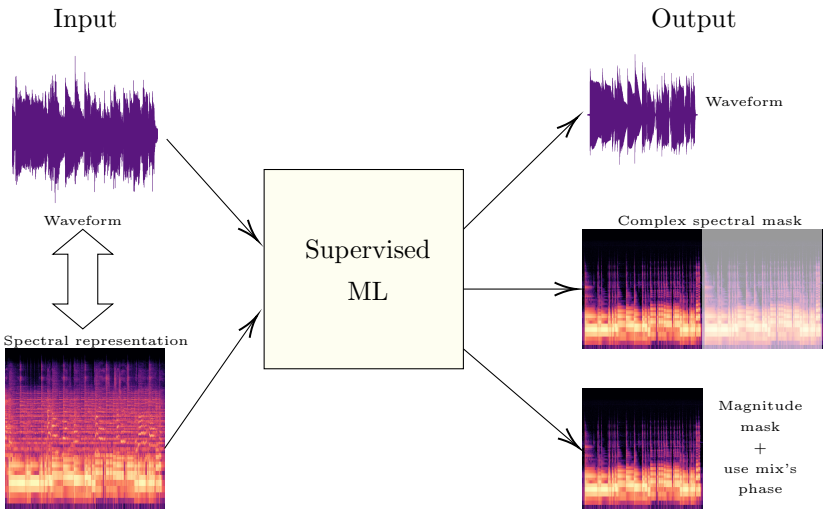


Figure 2: Basic pipeline with input and output options

data to train the models, but given the limited dataset constraints (eg. the 100 songs of MusDB) inductive bias can significantly impact the results. It has been observed that spectrogram and waveform models tend to generate different artifacts[5][22], these artefacts will also vary depending on the source, ranging from muffled attack sounds in drums and bass due to phase inconsistency in spectrogram models to crunchy static noise in vocals separated by waveform models. Trying to compensate for these shortcomings, the current SOTA models take a hybrid approach, where the input is processed simultaneously by two parallel branches, one receiving the raw waveform, the other the spectrogram[5].

The machine learning architecture that underpins many of these SOTA systems, regardless of the type of input, is called the U-net. Originally developed for image segmentation, it has been adapted to a variety of tasks beyond its original domain, including audio processing[20]. Fundamentally, the U-net can be characterised by its symmetrical structure, consisting of a contracting path to capture context and an expansive path to provide localisation. This configuration facilitates the capture of patterns at multiple scales, making it particularly suited to a task such as source separation, where different frequential components may require different treatments. Typically, each convolutional layer in the contracting path is followed by a max-pooling operation that reduces spatial dimensions and progressively increases depth, while the expansive path uses transposed convolutions to upscale features. Skip links bridge the layers from the contracting path to the expansive one, helping to preserve high-resolution features (fine detail).

Recent research by Guso et al.[8] on which objective evaluation metrics should be used as possible loss functions for training, recommends training with the spectrogram-based losses: L2-freq, SI-SDR-freq, LogL2-freq or LogL1-freq with potentially phase-sensitive targets and adversarial regularisers. They also recommend LogL1-time, which was found to be able to deliver competent

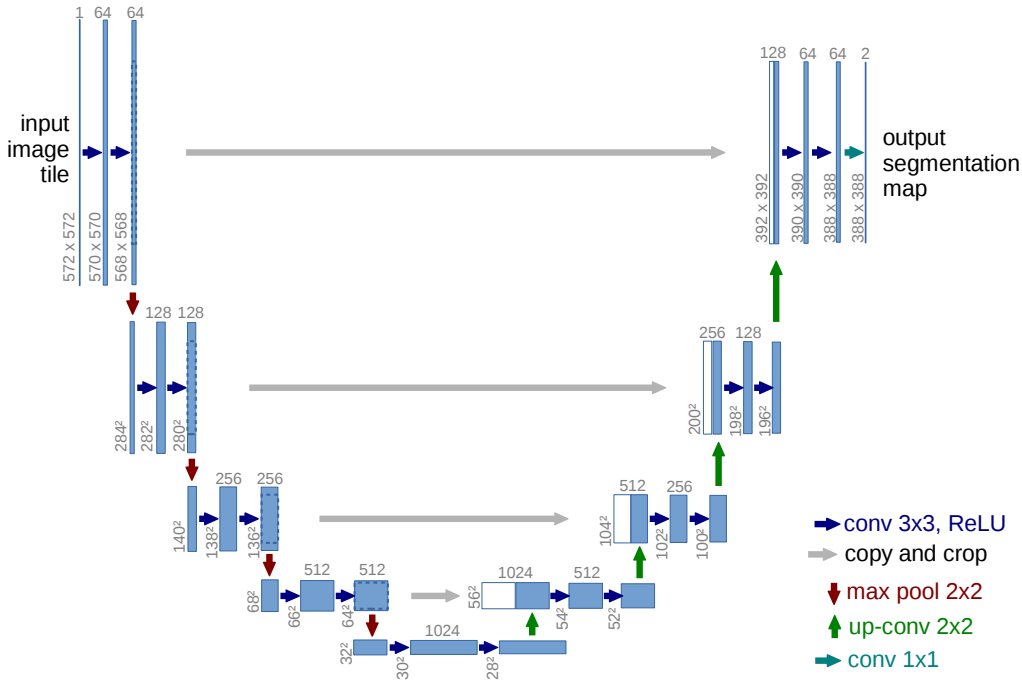


Figure 3: Classic U-Net architecture presented in the original paper by Ronneberger et al.[34]

results, and do not recommend the use of mask-based losses.

Source Separation Evaluation

Models of all these types perform very well on standard evaluation metrics, but still present audible artefacts, which highlights a key element in source separation: the evaluation of the separation. This task is extremely complex as the human ear is very finely tuned to unexpected or unusual sounds and noises, and it is difficult to find an evaluation criterion that can correctly detect what our ear perceives. In general, there are two main ways of evaluating the results of a source separation approach: objective[6][35] and subjective[36][37], both of which have their advantages and disadvantages[8][38][39][40].

Objective measures evaluate the quality of source separation by performing a series of calculations that compare the output signals of a source separation system with the "ground truth" isolated sources. The most commonly used are the Source-to-Distortion Ratio (SDR), the Source-to-Interference Ratio (SIR) and the Source-to-Artifact Ratio (SAR) as defined by [35], and the Scale-Invariant Source-to-Distortion Ratio (SI-SDR) as defined by [6]. For these metrics, estimates of a source $\hat{s}_i$ are assumed to consist of four separate components, the target source s_{target} (a version of s_i modified by an allowed distortion), the interference error e_{interf}, the noise error e_{noise}, and the artefacts e_{artif}.

Another interesting approach to evaluation is proposed in [22], which suggests proxies for evaluating perceptually relevant features: the spectral roll-off distance at 98% to analyse whether an estimate has too much high-frequency content or lacks desirable high-frequency content, and an F1 between the binary onset strength threshold on the ground-truth source and a source estimate to analyse how well an estimate preserves transients.

Generative works in audio have also incorporated the fairly recent Fréchet Audio Distance (FAD) by Kilgour et al.[7]. FAD is a relatively new metric, inspired by those used to evaluate image generation with Generative Adversarial Network (GAN)s, that uses deep learning representations (embeddings) to evaluate how the generated signal compares to studio-recorded audio and, unlike other metrics, provides a reference-free evaluation. This perspective on audio evaluation, which attempts to re-examine the essence of what makes audio "good" or "authentic" without being tied to direct waveform or spectrogram comparisons, is also what inspires our work.

Subjective evaluation methods involve human evaluators assigning scores to the output of the source separation system. The SOTA for subjective evaluation standard are the Mean Opinion Score (MOS), and Multiple Stimuli with Hidden Reference and Ancho (MUSHRA), which was originally developed to evaluate the quality of audio reconstruction by codecs, but has since been adapted for evaluation of various tasks. These tests should be developed according to the recommendations of the International Telecommunications Union[36], and are ideally carried out by well-trained sound engineers in a sound-treated room, comparing the qualities of various conditioned sound files with a reference. In the MUSHRA evaluation, files to be compared should include a hidden reference and at least one hidden anchor, an intentionally bad variant of the audio traditionally obtained by low-pass filtering. These are intended to provide boundaries between evaluators, and to alert when an evaluation may have been performed incorrectly.

This kind of evaluation is considered to be a better manner of evaluation if carried out correctly, but it can be quite costly and difficult to implement. Crowd-sourced variants have been implemented[37][41], which greatly facilitate the possibility of performing subjective evaluation, especially in terms of cost and time. They can be performed by any fit person with a pair of headphones, and have been shown to be an effective alternative.

2.2 Generative Architectures

Research into automatic music generation remains one of the most challenging tasks in the audio domain and has received a great deal of attention in the last year, with models such as the ones proposed by Agostinelli et al.[12], Liu et al.[11], Donahue et al.[42], and Copet et al.[43] revolutionising the field. At their core, music generation systems strive to create cohesive and sonically pleasing audio sequences, often emulating the complexity and expressiveness of human-performed music. This section provides an overview of the SOTA in generative models applied to music, summarising the different approaches and architectures that have been proposed in recent

years.

In terms of audio generation, most of the current SOTA work for audio generation are models adapted from image generation, such as the GAN[44] or diffusion models [45], both of which have demonstrated their ability to produce high-quality music snippets, capturing intricate structures and dynamics. The recent surge in transformer-based architectures[46], particularly those that harness the potential of neural codec tokens[47], has set new benchmarks in the music generation task. These models combine the ability to process and generate long sequences of tokens from transformers with the versatility and high sonic quality of neural codecs trained for sound compression and reconstruction, to produce compositions with overarching thematic coherence and great sonic detail at the beat level.

We will now review the most relevant work in this area and conclude with an analysis of whether the advances and breakthroughs in music generation have been, or can be, applied to the task of source separation.

2.2.1 Generative Adversarial Networks

GANs are a class of models developed for unsupervised machine learning. They are characterised by the interplay between two sub-architectures: a generator and a discriminator. The generator, as the name suggests, generates new data that aims to be indistinguishable from real data, while the discriminator evaluates the authenticity of the data it receives, distinguishing between real data and those generated by the generator. The two networks are trained simultaneously in what's known as "adversarial training", where the generator constantly refines its outputs to fool the discriminator, while the discriminator also tries to get better at its task. If trained correctly, over time this "duel" should result in the generator producing high-quality synthetic data that matches the real thing.

Specifically for automatic music generation, GANs have experienced an upsurge in adoption, and many of the current state-of-the-art approaches implement at least part of their model and training scheme with adversarial training. GANs have been used for purely generative audio tasks[48][49][50], but also for tasks such as artifact correction and audio enhancement[17][18], and crucially for our work, source separation enhancement[22].

The standout research on enhancement, by Su et al.[17][18], proposes a recursive network that predicts clean Mel-Frequency Cepstral Coefficients (MFCC)s, which are then used to condition a WaveNet with noisy input to predict the output clean signal at 16kHz, which is then extended to 48kHz with a separate bandwidth extension network. Theirs is a medium sized architecture (approximately 34 million parameters) and their training on around 12 hours of audio gives excellent results. The caveat is that they only train on speech and do not provide a public implementation of their system.

The work of Schaffer et al.[22] proposes a source separation post-processor, termed Make-it-Sound-

Good (MSG), which both denoises and imputes, aiming to bolster the outputs of various source separation models, whether they operate in the waveform or spectrogram domain. They train and evaluate their model on the MUSDB18 dataset to improve the output of four popular separators (DemucsV2, Wavenet, Spleeter and OpenUnmix) at a sample rate of 16kHz. Although their model showed promising results, especially for bass and drums, the overall performance left a lot of room for improvement.

Some of the challenges associated with GANs relate to their training complexity: training can be unstable due to the balance between generator and discriminator, and the model can be very sensitive to hyperparameters, requiring careful tuning. In addition, the computational power required to train these models can be extremely resource intensive, and is greatly improved by large amounts of data, which is often not the case for audio tasks. Particular complexities in audio generation arise from the need for temporal coherence over a long track, as well as phase coherence, which is fundamental to sound quality.

2.2.2 Diffusion Models

Diffusion models are a relatively new paradigm in the generative deep learning landscape. Based on the study of stochastic transitions, these models involve a process in which noise is progressively added to an input signal, and the model attempts to learn to reverse this noise-induced alteration. When introduced[45] for image generation, they claimed the potential to surpass the capabilities of GANs[51].

Diffusion models have since found applicability in the realm of audio generation: some approaches to this task have taken models designed for image generation and applied them to audio by working with spectrograms[13][52]. On the other hand, some research initiatives have opted for a more direct method, adapting the models to work directly with waveforms[11][53].

A first application of diffusion models to music source separation is manifested in the work of Plaja-Roglans et al.[21], where they take inspiration from the diffusion models paradigm and introduce a deterministic diffusion perturbation using the music mixtures. This perturbation is used to progressively morph the isolated singing voice into its corresponding mixture, while the model learns to reverse this process, attempting to extract the singing voice from the "noise" (mixtures). By operating in the waveform domain, the method avoids the problems associated with phase estimation, and another highlight of their work is the efficiency of the model, which uses fewer parameters than its state-of-the-art counterparts, demonstrating the potential for lightweight yet effective applications. A notable drawback is the lack of audio results or pre-trained weights, although the full model architecture and training code are available.

The diffusion model-based approaches to audio generation are still in their early stages, so the quality of the generated audio is currently struggling to compete with some of the more established methods, although it's undeniably an exciting avenue of research that promises to soon be up to

par in terms of sound quality [54].

2.2.3 Neural Codecs and Codec-based Transformers

Finally, there is a new approach to generative audio modelling that uses Transformers[46] and pre-trained language models for generation, and exploits the quantized intermediate representations of neural codecs as compact and meaningful tokens from which audio can be reconstructed. This includes models such as MusicLM[12], SingSong[42], MusicGen[43], AudioLM[47][55], and VampNet[56].

These works are based on research into neural codecs, a recent development, with architectures such as SoundStream[15], EnCodec[16], the discrete version of RAVE[57], or the Descript Audio Codec (DAC)[58], which achieves 90x compression while maintaining exceptional fidelity and minimal artefacts. These codecs are a special case of GAN, where the generator is a Variational AutoEncoder (VAE) with a residual quantization scheme for the latent embeddings. The models present small differences in the architectures and training schemes, but the main factors are as follows: they present an encoder-decoder architecture with a Residual Vector Quantization (RVQ) scheme for quantizing the encoded embeddings, trained with adversarial schemes on a variety of datasets ranging from 40k to 280k hours.

RVQ is a technique used to compress the encoder output to a given bit rate. It adapts Vector Quantization (VQ)[59][60], a technique that converts the encoder output into a bitrate-specific format using a codebook of vectors, as illustrated in Figure 4. These codebooks can be learnt, and included in training with the encoder and decoder, allowing integrated end-to-end training through backpropagation. The encoded audio is then transformed into a sequence of one-hot vectors, providing a compact representation that efficiently reduces the bit rate while retaining the necessary information.

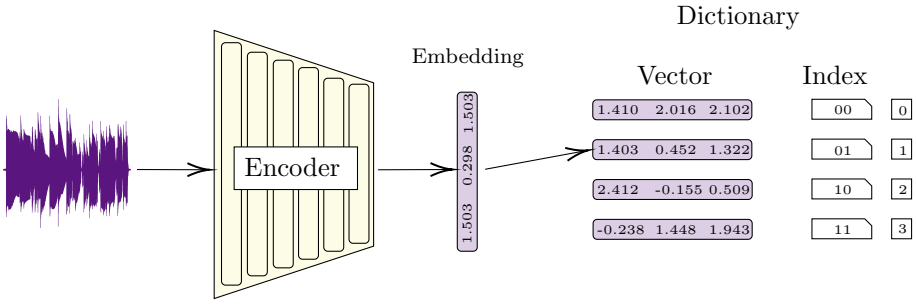


Figure 4: Vector Quantization (VQ) example

However, traditional VQ has its limitations: quantizing the information of a multidimensional embedding with a finite dictionary of embeddings and at a relatively low bit rate would force the quantizer to lose either temporal definition or the resolution of the embedding. The iterative approach of the RVQ method, uses multiple quantizers to progressively approximate the residuals,

ensuring a more accurate quantized output. The process is illustrated in Figure 5: the initial audio vector is quantized, residuals are computed, and these residuals are then quantized through a series of additional stages. This provides a nice balance between computational requirements and coding efficiency, with the rate divided equally between the VQ stages, each of which uses its own dictionary, which is refined using exponential moving average updates.

Algorithm 1: Residual Vector Quantization as introduced by Zeghidour et al.[15]

Input: $y = \text{enc}(x)$ the output of the encoder, vector quantizers Q_i for $i = 1..N_q$

Output: the quantized $\hat{y}$

$\hat{y} \leftarrow 0.0$

residual $\leftarrow y$

for $i = 1$ *to* N_q **do**

  $\hat{y} += Q_i(\text{residual})$

  residual$-= Q_i(\text{residual})$

return $\hat{y}$

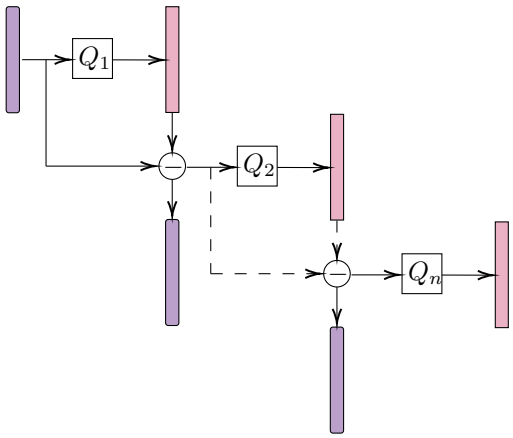


Figure 5: RVQ overview

Training the RVQ has a few tricks up its sleeve: The k-means algorithm is often used to optimise the initialisation based on the training data, under-used codebook vectors can be replaced with current batch samples to try to improve codebook efficiency, and the number of layers is changed during training (this is easy with the hierarchical structure of the RVQ) to avoid locking the bitrate to specific bitrates, but also so that the learning of the codebooks follows the hierarchical design, encoding basic information in the first layers and fine details in the latter.

In the case of EnCodec, which we will use extensively in our work, the RVQ layer consists of a maximum of 32 cascaded vector quantization layers, each with a dictionary of 1024 codebooks that quantize the generated 128-dimensional embeddings. The encoder consists of four blocks of residual convolutions, which are mirrored by the decoder. The codec training uses at least six different losses, which will be detailed in our section 5.

DAC improves on EnCodec mainly through two advances: they identify a problem because of which EnCodec does not use the full bandwidth it has available due to codebook collapse (where a fraction of the codes are unused), and address it using improved codebook learning techniques introduced in the Improved VQ-GAN Image Model[61]: factorisation and L2-normalisation. Factorisation decouples code lookup and code embedding by performing the code lookup in a low-dimensional space (8 dimensions) while the code embedding resides in a high, 1024-dimensional space, which is essentially a code lookup using only the principal components of the input vector that maximally explain the variance in the data. The L2 normalisation of the encoded and codebook vectors

converts Euclidean distance to cosine similarity, which is helpful for stability and quality. They also address a side-effect of the quantizer dropout proposed in SoundStream, which actually degrades full-bandwidth audio quality, and propose a low (but not zero) probability of codebook dropout as a compromise to achieve better reconstruction.

Central to the usefulness of these models is their ability to produce meaningful tokens. Transformer-based architectures have been successfully used to model natural signals such as images, speech and music, and the key component of these works is the high-quality neural compression models that compresses the high-dimensional natural signals into lower-dimensional discrete tokens.

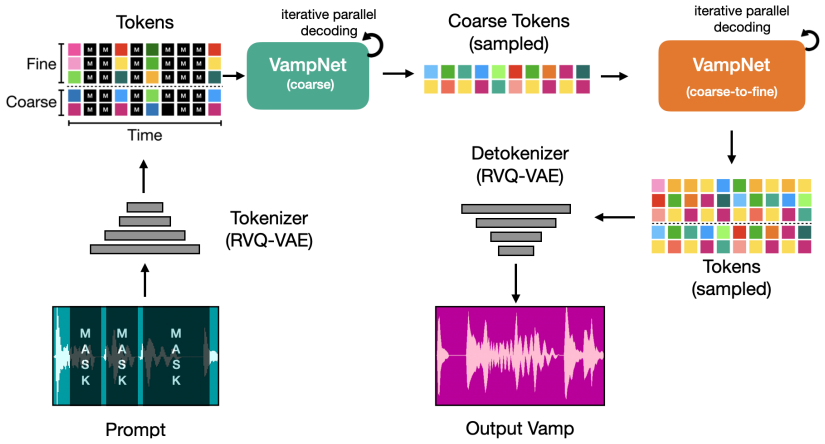


Figure 6: VampNet by Garcia et al.[56] as an example of codec token leverage with transformers

In this way, the intermediate quantized representations of these models have been used for generative applications (see the example in Figure 6) by combining them with language models[47][55] and as mentioned before, have proven to be very flexible and capable of generating new high-quality musical content[12][42][43][56]. The field of generative audio modelling was given a major boost by the MusicLM architecture by Agostinelli et al. which was trained on 280k hours of music, the resulting audio is of high quality and demonstrates the generative potential of this approach, where they present three hierarchical transformers that go from "semantic" audio tokens to codec tokens they can reconstruct into audio.

More specifically, MusicLM[12] follows the paradigm proposed in AudioLM[47], where a T5 transformer[62], pre-trained on a variety of language tasks, was first fine-tuned to predict tokens from low-level residuals, called "coarse" tokens, and a second T5 was fine-tuned to predict the posterior residual ("fine") tokens from the coarse ones (coarse-to-fine prediction). The more recent MusicGen[43] architecture consists of a single language model that operates over multiple streams of tokens but consists of a single-stage transformer LM together with efficient token interleaving patterns, eliminating the need to cascade multiple models hierarchically.

The recently published VampNet paper[56] also proposes to work in a herarchical fashion, but

focuses on the relevant task of conditioning generation with music (rather than text). To do this, they implement a variable masking schedule during training, which allows them to sample coherent music from the model by applying different masking approaches during inference. The VampNet transformer they propose is non-autoregressive and uses a bidirectional transformer architecture that considers all tokens in a forward pass, allowing the model to "generate coherent, high-fidelity musical waveforms with only 36 sampling passes".

In summary, these models demonstrate that the use of neural codecs can lead to high-quality generative audio modelling, opening up new possibilities for a wide range of applications. The codec is reinvented to have generative capabilities, even though it is designed to reconstruct a perfect audio signal, and a mixture of reconstruction and generative capabilities is exactly what we want for our task of reconstructing sources from degraded estimates produced by a source separation system.

3 Generative Approach to Source Separation

3.1 Problem Formulation

As we explained in the introduction, we're interested in refining or post-processing the estimated source $\hat{s}_i$ to belong to a class of "natural instrument signals", this is, "natural bass signals" for the bass approximation, "natural drums signals" for the drums, etc. Let's call the set of natural instrument signals S_I. Our objective is to find an approximation $\bar{s}_i$ such that $\bar{s}_i$ is close to s_i in terms of some metric or criteria and $\bar{s}_i \in S_I$.

This problem could be defined through a projection, where given a well-defined space S_I of natural instrument signals, we wish to find the projection of $\hat{s}_i$ onto S_I:

$$\bar{s}_i = \text{Proj}_{S_I}(\hat{s}_i)$$

Ensuring that $\bar{s}$ is the closest (in terms of Euclidean distance, or another appropriate metric) signal in S_I to $\hat{s}_i$.

Another way of viewing the same formulation would be optimization-based, where we aim to minimise the distance between $\hat{s}_i$ and $\bar{s}_i$ with the constraint that $\bar{s}_i \in S_I$.

$$\begin{aligned} \bar{s}_i &= \underset{s'_i}{\text{argmin}} \|s'_i - \hat{s}_i\| \\ &\text{subject to } s'_i \in S_I \end{aligned} \tag{2}$$

Of course, the formulation's effectiveness will largely depend on how well we can define the set S_I and the criteria for "closeness" between signals. In our case, signals from S_I are obtained through

a network trained to generate "natural instrument signals", and our pipeline could be trained to receive $\hat{s}_i$ or even x and process it to produce $\bar{s}_i$.

One last observation about this formulation is that evidently, in an ideal case, $\bar{s}_i = s_i$ as $s_i \in S_I$, and this should be the case when the approximation $\hat{s}_i$ is extremely close to s_i. But it must be taken into account that there are many "natural instrument signals" that are *musically* similar even though their sound is different (in attack duration or timbre for example). Thus, with the minimisation function not considering s_i at all, it is likely we will end up with a signal $\bar{s}_i$ that is further away from s_i than $\hat{s}_i$. $\bar{s}_i$ may consequently perform worse in objective metrics than the separated input to the system $\hat{s}_i$, but the goal is that it will sound "better" (more natural) than $\hat{s}_i$.

3.2 Datasets and Pre-processing

The domain of music source separation has a variety of datasets, but MUSDB18[1] stands out as a key benchmark. Serving as the backbone for numerous community initiatives (including the Music Demixing Challenge[2]), the MUSDB18 dataset offers 150 full-length music tracks spanning approximately 10 hours. These tracks represent a spectrum of Western music genres, predominantly pop/rock, but also genres such as hip-hop, rap and metal. For each track, MUSDB18 provides isolated stems for drums, bass, vocals, an "other" category and the mix, which can also be recreated by summing the individual stems, providing a comprehensive resource for separation tasks.

In addition to the well-established MUSDB18, our research also uses the Bean dataset, a proprietary multi-track collection presented in the work by Pretet et al.[63]. Bean presents excerpts of 25,938 songs, all approximately 30 seconds long. The genre analysis presented by Pretet et al.[63] revealed that, similarly to MUSDB18, Bean's genre distribution is heavily skewed towards pop and rock, reinforcing its relevance and compatibility.

In most of our experiments, we will use pre-trained source separation systems to pre-process our data before input. We have chosen to use two widely recognised benchmarks in the field: Spleeter and Demucs (the latest -4th- hybrid version). Both tools have been previously introduced and discussed in the SOTA, section 2.1, and given their performance and adoption in the community, they provide a good foundation for our experiments, ensuring that they are based on recognised baselines.

3.3 Current Limitations

Procuring a strong foundation, we started our work by doing a thorough analysis of the quality of the pre-trained source separation methods we will use. Our first step was to focus on identifying and analysing the errors present in the output of Spleeter-Pro, a non-public version of the source separation system developed by Deezer[3] to separate into two stems: vocals and accompaniment. The main objective was to subjectively assess a limited number of Spleeter-Pro outputs in terms

of equalisation, timbre, volume, artefacts and bleeding, and for this analysis to serve as a starting point for understanding the limitations and potential of Spleeter-Pro outputs, and to inform subsequent decisions on the direction of our work.

This analysis identified several problems, such as loss of high frequencies, occasional artefacts and bleeding in quiet sections, difficulties in handling more than one voice, and bleeding problems with choral-sounding synthesizers. We also found that loudness was affected and that there were abrupt changes in the Spleeter-Pro output, which could be related to the presence of a significant amount of energy in the vocal section that was not compensated for when the accompaniment was extracted from the compressed mix. An interesting observation was that the Spleeter-Pro output often had parts that sounded perfect, while other parts sounded imperfect. The information from the perfectly separated sounds could potentially be used to set a baseline for improving the quality of the other parts.

The performance of Spleeter's and Demucs' 4-stem systems was also considered, but these systems are already present in the community and their shortcomings have already been addressed in the literature, as reported in the SOTA section 2.1 of this document. Suffice it to say that there is definitely room for improvement, with the "other" track showing bleeding from the other stems, some artificial artefacts and apparent filtering at some frequencies, possibly caused by masking by other instruments; the drums sounding frequentially unbalanced, with not much mid or low frequencies, a kind of underwater sound, and with noticeable volume changes; and the bass lacking high frequencies, which greatly muffles the attacks and also gives the kind of submerged sound.

3.4 Experimental overview

Having arrived at a formulation of the problem, chosen the data and pre-processing tools, and identified the limitations of these separation systems' outputs, the next step was to explore the performance of previous works in generative audio modelling, to improve source separation. Several types of models were considered, with the main choice being between diffusion-based models and GAN-based models.

In the following sections, we delve into the experiments conducted and analyse their results. An integral part of our work relies on subjective auditory perception. Whilst we present graphs and figures that offer quantitative insights, many of our decisions and insights are based on the qualities heard in the produced audio. For this reason, we have created a dedicated website with audio examples that correspond to our experiments. Throughout this document, when we refer to specific audio examples, we strongly encourage the reader to visit the website to experience the audio first-hand. We hope this will add invaluable context and depth to our written analysis.

Website with examples: ivanlmh.github.io/DeezerATIAM

It's worth noting that all of the experiments discussed in the following sections were conducted

using a single GPU, whether explicitly stated or not.

4 Baselines

To determine which type of model to focus on, we listened to the reported audio results from various SOTA models[18][13][50]. We also deployed two models from previous literature in search of a baseline for our work. These were the diffusion-based separation model by Plaja-Rogalns et al.[21] and the Make-it-Sound-Good post-processor by Schaffer et al.[22].

4.1 Diffusion-based Separation

The diffusion-based model for music source separation provided by Plaja-Roglans et al.[21] was the main contender in its class. As stated in the SOTA section 2.2.2, instead of performing the diffusion perturbation with noise, they introduce a deterministic diffusion using the mixture. In this way, the model learns to reverse the process and extract the singing voice from the mixtures.

One issue with their work is that they did not report audio results, nor did they provide any pre-trained weights. The model is a relatively small one, and training code was available, but with some bugs and no inference function. We implemented a fix, and trained the model so as to be able to hear what the results where like.

We performed only one experiment, which was to train the model for separation with $J = 2$, the two stems being vocal and accompaniment. We trained on MUSDB18 HQ until convergence at 99750 steps, which took about 48 hours on a GPU. Example 1 on the accompanying website is the separation at this final checkpoint, on a song from the MUSDB18 test set, which was our validation during training. As can be heard, the separation is not good, the voice is still present in the accompaniment and vice versa, frequency filtering artefacts are very present, and decent separation is only achieved when there is no voice singing (the vocal track goes silent and the accompaniment is undistorted). These results, together with the reported audio results from other diffusion model-based audio generation work, gave us the impression that the approach was not mature enough and was not generating audio of sufficient quality for us to lean towards it. We therefore stopped working with diffusion models at that point.

After considering the results reported in various works, it was decided that GAN-based models would be our starting point. This decision was based on their superior sonic quality in comparison to the results from SOTA diffusion models that are currently still grainy and noisy, and not as high quality as some GAN-based generations.

4.2 Make-it-Sound-Good

One of the first steps in working with GAN-based models was to train the MSG[22] model by Schaffer et al. on the output of just Spleeter, rather than on four different models as originally

proposed by the authors. This would allow us to focus on improving the errors of just one source separation system, rather than the heterogeneous errors of many. We trained it for the enhancement of bass and of the whole accompaniment (a 2-stem separation variant).

4.2.1 Methodology

MSG proposes an architecture based on a U-net with 1-D convolutions, similar to DemucsV2[64], and use adversarial discriminators similar to those proposed in HiFi-GAN[18], that adopts both multi-period and multi-resolution tactics, and multiresolution spectrograms to condition their generator, which has about 30 million parameters. The overview of their system can be seen in Figure 7.

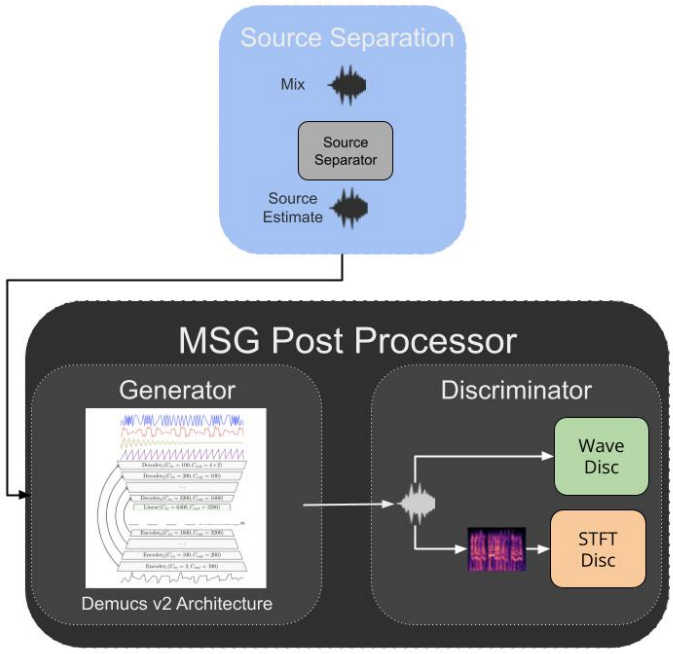


Figure 7: MSG paper architecture overview from Schaffer et al.[22]

Training the generator, they use three loss functions: first, there's the loss of the Least Squares GAN generator, a stable variant of the GAN loss which helps the generator to produce outputs that the discriminator finds indistinguishable from the real data:

$$L_G = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left(D_k(G(\hat{s}))-1\right)^2\right] \tag{3}$$

where $\hat{s}$ is the raw source estimate from the separator, D_k, is the k-th discriminator, K is the total number of discriminators, and G is the generator.

Then there is the deep feature matching loss, which is computed as the L1 distance between intermediate activations of the discriminators when evaluating both real and generated data, ensuring that not just the final outputs but also intermediate representations of generated data are similar to real data.

And finally, the generator uses a multi-scale Mel-spectrogram reconstruction loss. This loss helps to refine the spectral content of the generated audio, ensuring that it closely matches the desired spectral structure:

$$L_{\text{MS}-\text{Recon}} = \sum_{i=1}^{M} L_{\text{stft}}^{(i)}(y, \hat{y}) \tag{4}$$

$$\begin{aligned}
L_{\text{stft}}(y, \hat{y}) &= L_{sc}(y, \hat{y}) + L_{\text{mag}}(y, \hat{y}) \\
L_{sc}(y, \hat{y}) &= \frac{\| |\operatorname{STFT}(y)| - |\operatorname{STFT}(\hat{y})| \|_F}{\| |\operatorname{STFT}(y)| \|_F} \\
L_{\text{mag}}\ (y, \hat{y}) &= \frac{1}{T} \| \log |STFT(y)| - \log |STFT(\hat{y})| \|_1
\end{aligned}$$

where M is the number of Short Time Fourier Transform (STFT) losses, and each $L_{\text{sft}}^{(i)}$ applies the STFT loss at different resolution with number of FFT bins $\in \{512, 1024, 2048\}$, hop sizes $\in \{50, 120, 240\}$, and window lengths $\in \{240, 600, 1200\}$.

Regarding the discriminators, there are two main types, resulting in a total of eight different discriminators: the multi-period waveform discriminators, of which there are five, are convolutional networks acting on undersampled waves, so as to examine the waveform structure at different temporal resolutions. The other three are multi-resolution spectrogram discriminators that focus on different spectral resolutions of the audio content:

$$L_D = \mathbb{E}\left[(D(s) - 1)^2 + (D(G(\bar{s})))^2\right] \tag{5}$$

where $\bar{s}$ is the enhanced source estimate from the MSG generator and s is the ground truth source audio.

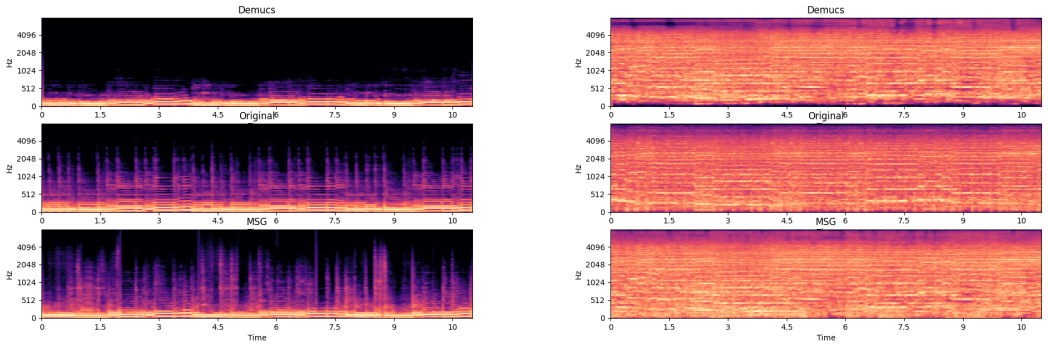
Though their approach uses adversarial losses intrinsic to GANs, they abstain from conditioning on a random input vector, we are thus considering a deterministic model (or at least a densely-conditioned, not traditionally generative model).

For a deeper dive into the specifics of these loss functions and their underlying methodologies, one can refer to the foundational works cited by Schaffer et al. in section 3 of their paper[22].

4.2.2 Experiments

We trained the model with the small variance that we only used the output of Spleeter's separation, so that the model could focus on improving only one type of sound error, and we also wanted to

take advantage of the fact that the MSG architecture is based on Demucs V2, so we would be combining two different architectures that might complement each other. We trained the model for two tasks: bass enhancement and the much more complex accompaniment enhancement. Both tasks were trained for 50 epochs (120k steps) on the MUSDB18 train set, which took just over a day on a GPU, and at which point the losses flattened out for the validation set.



(a) Bass enhancement

(b) Accompaniment enhancement

Figure 8: Spectrograms of audio examples 2 and 3, the input, target and output of MSG

The mean SDR went from 7.4 to 12.5 and from 4.4 to 8.7 for bass and accompaniment respectively. These are good results, especially for bass, which was to be expected as their work reported bass as one of the strongest improvements along with drums. Examples 2 and 3 on the accompanying website, and Figure 8, show the input (Spleeter's output), the target (ground truth) and the output of the MSG system after 50 epochs. We can see and hear that, in the case of the bass, attacks and high frequencies are recovered at the cost of adding new artefacts that appear as high-frequency and unnatural noise. In the case of accompaniment, it's even worse: although the SDR goes up, we don't hear any noticeable difference in the output, apart from a few high-frequency artefacts.

These first experiments illustrate very well one of the main problems we are trying to address in this work: the improvement of SDR metrics or distance metrics to the ground truth signal reaches a point where they do not correlate with what we hear.

5 Codec Training

After establishing the baseline, our work's focus was solely on methods using the outputs of the 4-stem systems (not 2-stem as was tested with the baseline). The rationale for this was, firstly, that the baseline established for this was robust and provided a solid foundation for further exploration and, secondly, that the 4-stem separation inherently presented a simpler starting point, reducing the complexity that might arise from attempting to enhance more complex signals (such as the whole accompaniment).

Our desire was to generate more pleasing sounds, so we wanted to distance ourselves from the task of approximating a direct representation of the target signal, such as its waveform and spectrogram. While these representations are perfectly informative about the target signal, our goal was to capture the intrinsic musical content without being bound by the limitations of traditional signal approximations. The increasing capabilities of token-based music generation systems, which have recently demonstrated exemplary auditory results, offered a good perspective for achieving our goal by leveraging the codec's latent space and exploring previously proposed generation schemes conditioned in a variety of ways.

The idea of the "coarse-to-fine" model, which predicts the fine-detailed tokens from the coarse ones, intrigued us greatly. We reasoned that if we had the coarse-level details of the target, predicting its finer nuances might become more feasible, but in addition, the coarse representation of the target might bear a resemblance to the approximate outputs we get from source separation tools like Demucs and Spleeter. This hypothesis, if proven, could significantly bridge the gap between the approximate and the desired output, allowing us to have natural-sounding music, even if it is slightly different from the ground truth.

Furthermore, we considered what would happen if instead of relying on generic codecs, we trained our own codecs specifically to reconstruct a particular instrument. This dedicated system would be optimised, not for a wide range of sounds, but for a very specific timbral and spectral space, which could be the "natural instrument sounds" set S_I for this instrument. Such a targeted approach would intuitively allow for higher fidelity and precision in the reconstruction of that specific instrument, reduced artefacts related to unwanted sounds, and possibly also make the reconstruction more robust to noise.

Part of our experiments were dedicated to try and train one of these codecs. We were particularly interested in the SoundStream[15] and EnCodec[16] codec architectures, which has shown extremely good results in audio compression and synthesis tasks, but training them proved to be a complex undertaking as there was no public training code for either, which severely limited our options. In addition, the training process for such codecs is far from trivial, requiring large amounts of data, computational resources and intricate fine-tuning of a variety of hyper-parameters. There was also ambiguity about certain loss functions, as they were not all clearly detailed in the available literature, so the practical difficulties of implementation were significant.

5.1 Methodology

All neural codecs have their own original ideas and novel proposals for the training target, which is usually cleverly constructed to provide a combination of reconstruction and perceptual losses and optimal training of the RVQ. Regardless of the specification, there are always reconstruction losses to train the generator and a discriminator architecture. We will now introduce these main losses as implemented for the EnCodec paper and shown in figure 9, trying to keep them as general

as possible.

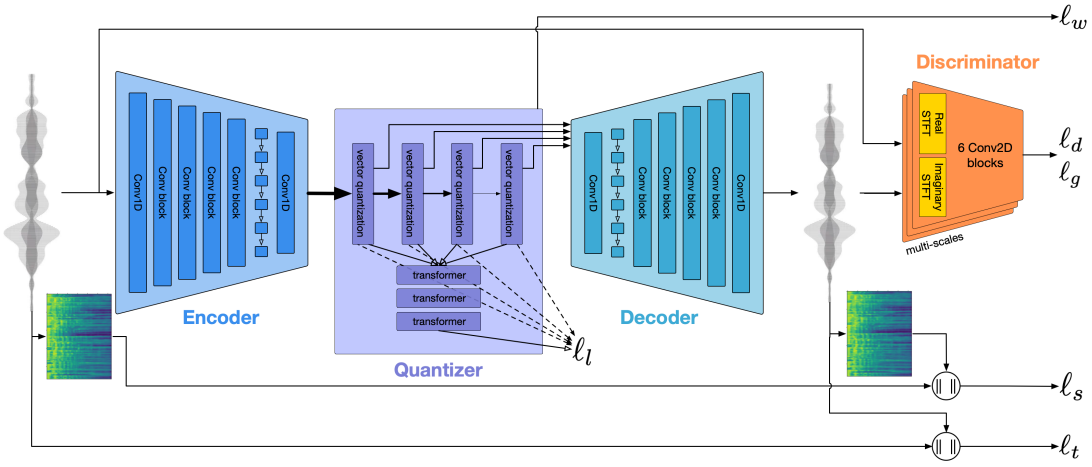


Figure 9: Neural codec architecture with losses, presented in EnCodec by Defossez et al.[16]

In the generator, the reconstruction loss operates on the signal in both the time and frequency domains. While in the time domain (l_t) the focus is on minimising the L1 distance between the target and the compressed audio, in the frequency domain a combined L1 and L2 loss (l_s) is applied to the mel spectrogram over multiple time scales. This combination is designed to achieve a balance between precision and stability.

A discriminator is added to attempt to improve the quality of the resulting samples, with a discriminator perceptual loss term using a multi-scale STFT-based discriminator to capture structure in the audio at different resolutions.

An adversarial loss l_g (hinge loss on the logits of the discriminators) is used to penalise the generator when the discriminator identifies real or generated samples. At the same time, a feature matching loss compares the outputs of the discriminator's internal layers for the target audio with the generated audio, promoting better alignment and similarity between them.

The VQ commitment loss, as proposed by van den Oord et al.[59], acts as a bridge between the encoder output and its quantized value. It encourages the encoder to produce outputs that closely resemble values from a given codebook, similar to the idea behind k-means clustering.

The losses are mixed with constant weights or more clever balancing methods, depending on the architecture.

In our work, we looked at the open source initiative led by LucidRains[65], which aimed to develop a community-driven version of SoundStream. Unfortunately, at the time of our experiments, it hadn't reached a stage where it could deliver satisfactory results.

Our attention was drawn to the RAVE VAE's, which has a "discrete" version supposedly analogous to EnCodec's, although its focus is not on error-free compression and reconstruction, but on

generative exploration. Caillon et al.[48] published a working training pipeline, which gave us a more direct route to exploration.

RAVE's discrete model training follows a two-phase methodology (shown in Figure 10), designed to address problems in reconstruction caused by phase variations when computing losses on the waveforms. The first phase, called the representation learning phase, focuses on training the encoder, quantizer, and decoder for accurate reconstruction using spectral distance as the primary loss. The next phase, called adversarial fine-tuning, freezes the encoder and quantization while the decoder is trained using a GAN discriminator. The transition point between these two training methods is determined by a hyperparameter.

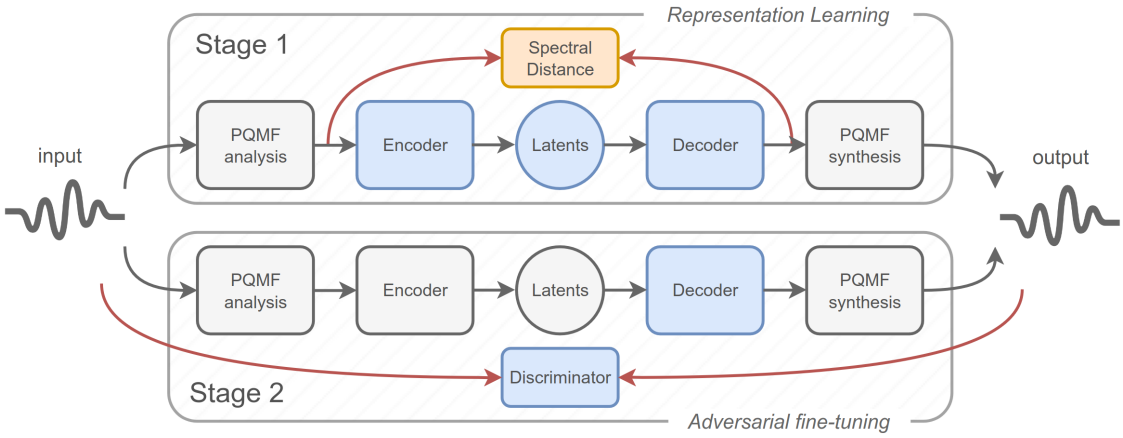


Figure 10: RAVE VAE architecture with losses throughout both training phases, grey components are frozen and blue ones are training, presented by Caillon[57]

Another characteristic of the discrete RAVE model is its strategy of appending noise to the 128-dimensional latent space, effectively doubling the dimensions per frame. This inclusion of noise acts as a form of Wasserstein regularisation (see page 67 in [57]), as such an addition affects the training since the decoder has a 1D convolutional layer at its input. We ran training runs both with and without this noise to measure its effect.

5.2 Experiments

SoundStream Training

When focusing on training a SoundStream-like codec, we first looked at the open source initiative led by LucidRains[65], which aimed to develop a community-driven version of SoundStream.

Our first attempt was to train this implementation of the codec on a small amount of data (the four stems of the MUSDB18 train set, representing about 40 hours of music); this proved to be a difficult task, with little data but much diversity (sound types varied too much) the model was unable to get more than noise, even after more than three days of training. A second attempt

was made to train the architecture on 10 hours of drum-only audio from MUSDB18, in the hope that this less sonically diverse dataset would allow the model to produce realistic drum samples. This converged after 36 hours, but even though we were practically overfitting on simple data, it reached a point where the reconstruction was far from decent: although we were able to produce drum-like sounds, the quality was poor and the samples sounded mechanical.

In an attempt to improve the sound quality of the reconstruction, and bearing in mind that the SoundStream architecture is based on large encoders that often perform well when presented with large amounts of data, we made a final attempt to train the architecture on 200 hours of drum-only audio from the Bean dataset in the hope of producing better quality drum samples. This took a long time to converge, and after 30k steps (four days on a GPU) the reconstruction was still quite poor. Example 4 shows the reconstruction of random samples from the training set, in the last training step we saved for both the model on MUSDB18 and the model on Bean.

At the moment of writing, some people in the community seem to have managed to train the codec to do decent audio reconstruction, but it is still not up to the level of the pre-trained codecs or the results reported by Soundstream. At the time of training, no one in the community had reported getting good quality reconstruction with this architecture and pipeline, and after our three attempts with different data, with the losses not giving us a clear idea of where the problem might be, we decided to move on.

RAVE Training

When training the discrete version of the RAVE VAE, we decided to focus on the bass sounds, as RAVE has been proven to work correctly on harmonic signals and in some cases present high-frequency artefacts on drum cymbals (which would defeat our enhancement purpose). Also, in our posterior work of fine-tuning the codecs, we would focus on enhancing bass signals and recuperating frequential information such as the bass's attacks.

We first did a test run of 36 hours on the MUSDB train set, using the default configuration of a 128-dimensional latent space with concatenated noise, 16 quantizers, a codebook size of 1024, a batch size of 8, and switching from the representation learning phase to the adversarial fine-tuning phase at step number 200k. This converged well, with a spike in total loss at the time of the change to the adversarial phase, but later converged to the value it was at before the switch. The audio results were good on a subset of audios from the train set that had been separated for validation, but we wondered how this would perform on the Bean dataset.

We trained the codec on the Bean dataset with the same hyperparameters, but found that the representation learning phase was far from converging at step 200k, so we ran another experiment with the phase change at 500k. This caused the model to converge in its representation learning phase, but the curious fact was that the total loss went up in the adversarial fine-tuning training phase and converged to the same point as the previous experiment. This shows that either the

representation learning was overfitting to the signal and the second phase is very necessary for good audio quality, or the second phase was not working very well. Audio example 5 on the website shows the reconstruction quality at the end of the representation learning phase, and at the end of the adversarial fine-tuning phase, and as can be heard, the reconstruction improved, supporting the decision to add an adversarial discriminator training phase. The question remains whether a discriminator layer *throughout* the training, as is traditional in GANs, would have improved the final result, but this option was not part of the RAVE pipeline, and since the codec is only a tool towards our goal of reconstruction with enhancement, we decided not to proceed in this direction.

We did perform a fourth experiment in which we removed the noise associated with the embedding and retained the long initial training phase. The result was that the learning of the representation converged slightly faster, but the second phase then converged to a slightly higher value as can be seen in Figure 11, perhaps illustrating the value of this method for regularisation. In any case, the audio results did not show any noticeable difference.

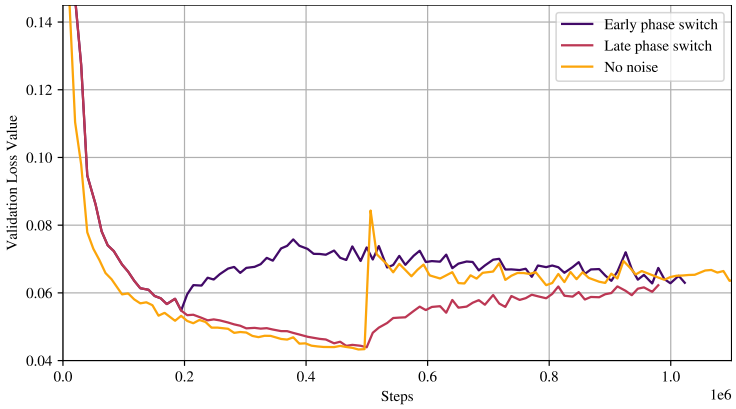


Figure 11: RAVE validation set losses

Finally, to test RAVE's adaptability, we trained the system to attempt to learn to reconstruct the ground truth bass directly from Demucs' bass output, and even from the unseparated mix itself. Both methods diverged during training, due to the fact that some of the losses are designed to enforce symmetry between the input to the encoder and the output of the decoder in the representation learning phase. It might have worked using a pre-trained model and training using only the second phase, but as we decided not to fixate on RAVE due to its inferior performance to EnCodec, this is left for future work.

6 Codec Fine-tuning

So far we have described and detailed the architectures and processes from previous work that we have implemented, adapted or deployed. This was a large part of our work as many of the systems were not in a mature state of development. The following sections describe the customisations,

architectures, training pipelines and training objectives that have been developed during our work and represent some of our most significant contributions to the field.

While the results from our RAVE experiments were decent, they still didn't surpass the reconstruction capabilities of EnCodec, and the latter model has been proven in generative applications with Transformers[43], making it an attractive option for our exploration. Although training pipelines for EnCodec were not available, there was access to both its architecture and a pre-trained model, so we decided to investigate how the codebooks were used and whether the quantized intermediate representations had any interesting musical signal representation properties. We used the causal EnCodec model, operating at 24kHz and compressing to 24kbps, which is the minimum compression, as the version that operates at 48kHz compresses to the same bit rate. The compression provides us with tokens at 75 frames per second, encoded with 32 residual layers with dictionaries of size 1024.

We started by analysing how the pre-trained model performed on encoding the stems of the MUSDB18 train set. Table 1 shows the calculated mean SDR of the EnCodec reconstruction over the whole dataset. We compute this source separation metric on the codec's reconstruction (no source separation involved), so as to obtain the ceiling we will have for our performance in this metric. This "ceiling" is quite high, especially for bass, and when the bass is mixed with drums the overall performance also goes up, which is interesting and reflects how compression scales with signal complexity.

	Bass	**Drums**	**Vocals**	**Drums & bass mix**	**Mixture**
SDR	24.7	11.7	11.1	15.1	11.6

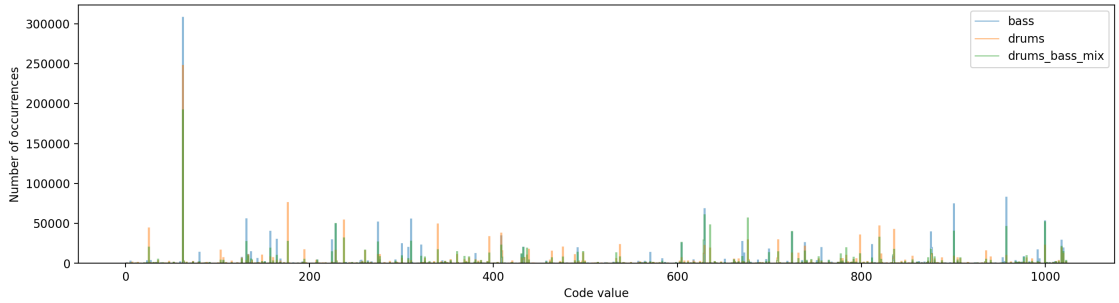
Table 1: Mean SDR of EnCodec over the 100 MUSDB18 train tracks

It should also be noted that the sound quality of the reconstructions was very good, with only a few minor artefacts appearing in a couple of cases.

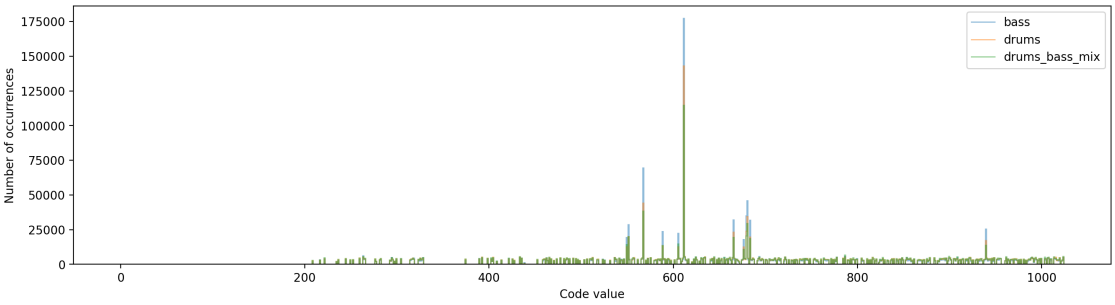
6.1 Compression Interpretability

To analyse the EnCodec model, we used the pre-trained model to encode drums, bass, a track with the drum-bass mix, and vocal audio files. We examined the occurrence of codebooks for each instrument by plotting a histogram (as shown in Figure 12). This facilitated the identification of frequently used codebooks for each VQ layer. The fact that the codebook usage is actually quite sparse, as reported in [56], is noticeable, and although for the posterior residuals the usage is more spread out, in the figure we can see a number of codebooks that are never selected. Of course, the fact that we only encode a few types of sounds in our experiments has to be taken into account, whereas the encoder is trained to encode a huge variety of sounds. We hypothesised that some of the most common codebooks, which differed greatly in frequency from the rest, might correspond

to periods of silence. To test this hypothesis, we created an artificial embedding from these most frequent codebooks and found that our reconstructions did indeed produce silence.



(a) 1st quantizing layer



(b) 32nd (last) residual quantizer

Figure 12: Codebook choice occurrence for bass, drums and bass+drums

Digging deeper into the generated latent space, we visualised the dimensionality reduction of the audio embeddings using techniques such as MultiDimensional Scaling (MDS) and Principal Component Analysis (PCA), as shown in Figure 13 and Figure 14a.

MDS and PCA are both popular dimensionality reduction techniques that transform high-dimensional data into a lower-dimensional space while preserving specific relationships within the data. MDS aims to preserve the pairwise distances between data points and is primarily used to visualise the similarity or dissimilarity between data sets. PCA works by identifying and ranking the axes (the principal components) that capture most of the variance in the data. By projecting the data onto these axes, PCA also provides a means of reducing dimensionality, while retaining as much of the original variance in the data as possible.

These visualisations allow us to appreciate the distinctions the model makes internally between different audio samples. One striking observation is the clear separation of bass and drums in this reduced dimensional space, while the mixed track logically overlapped with the individual instrument samples.

We then examined the role of the different layers within EnCodec, in order to analyse how each

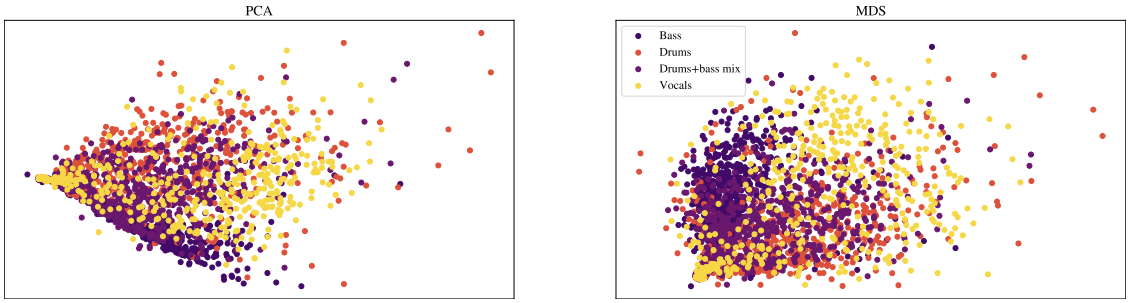
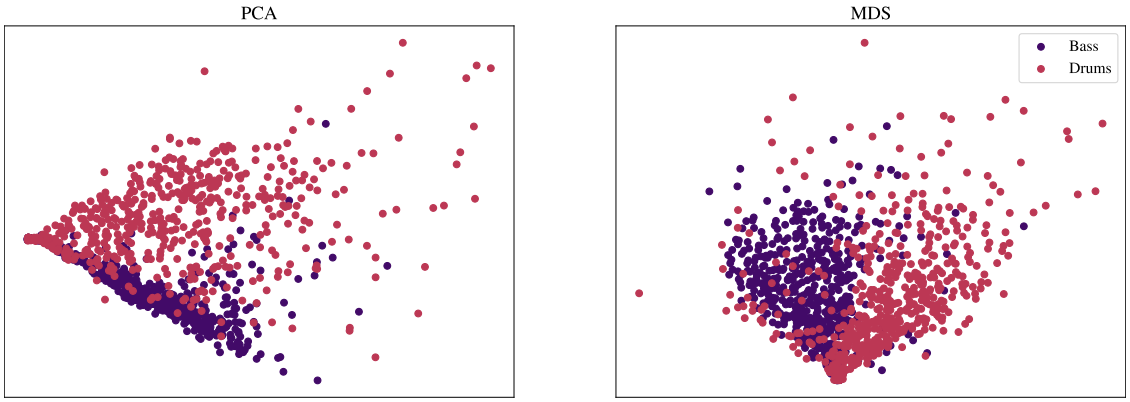
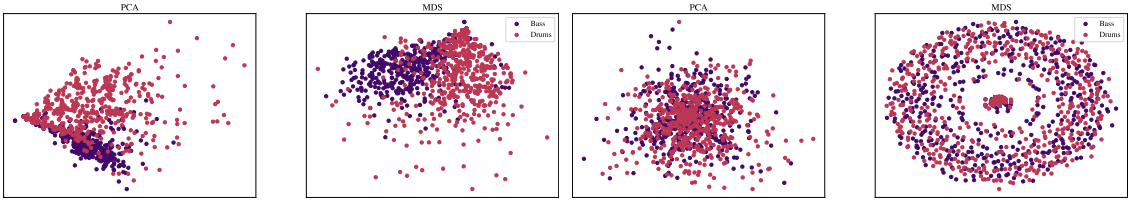


Figure 13: MDS and PCA on quantized embeddings for drums, bass, drums+bass, and voice samples

layer influences the separation we observed in the embeddings, and what their auditory effects were. Visual representations of the dimensionality reduction of these coarser embeddings showed us how the first two residuals had already achieved a good separation, i.e. were able to represent the signal, and also how the "fine information" of the last layers became random when separated from the previous residuals (see Figure 14c). As part of our experiments, we toyed with the idea of manipulating the latent space, specifically by shifting samples located in the bass latent region towards the silent region to achieve a rudimentary bass removal, but the results for this naive approach were less than stellar.



(a) 32 RVQ quantization



(b) 2 RVQ quantization

(c) Quantization of last 16 layers

Figure 14: MDS and PCA on quantized embeddings for drums and bass samples

6.2 Fine-tuning EnCodec for Enhancement

These analyses and modifications of the EnCodec model and its latent space can allow us to better understand its capabilities and limitations. We can also pose the question of whether it would be possible to predict the tokens for the ground truth instrument signal, from the output of Spleeter or Demucs (the separated signals), and from the ground-truth tokens a great reconstruction would be possible.

As we move away from traditional metrics such as the distance between spectrograms or the ubiquitous SDR, we try to avoid models that would impose a loss on the waveform and spectrogram representations. This approach is parallel to the idea behind FAD[7] and has also been present in recent generative approaches[66].

6.2.1 Methodology

To this end, we carried out experiments with variants of the codec architecture and with different losses. The architecture in Figure 15 shows an overview of all the components of our pipeline, and these component are combined in different ways which will be detailed in the Experiments section 6. We will now describe the architecture and losses' variants, so that in the experiments section we can focus on the practicalities of the implementations and tests.

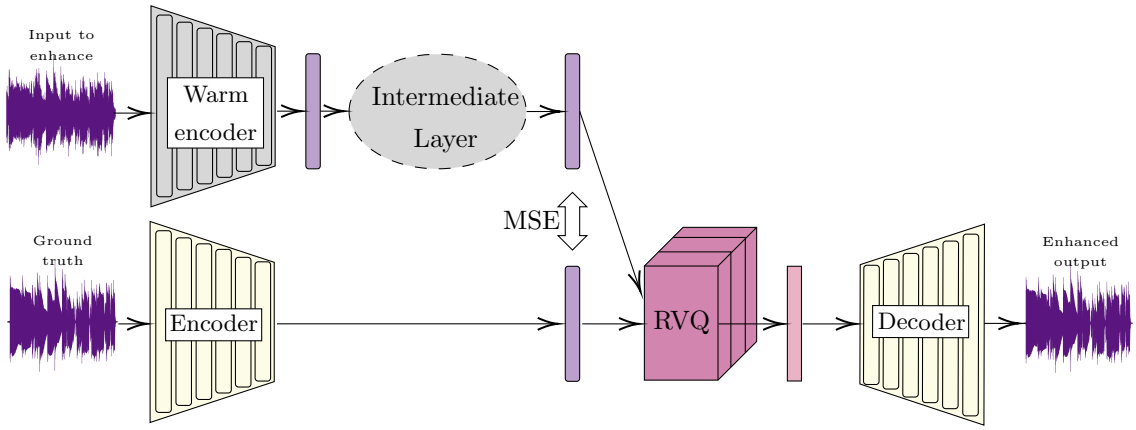


Figure 15: Codec architecture with the optional intermediate layer (grey elements are trained during finetune)

Variation in EnCodec's Architecture:

1) Frozen codec with warm encoder (FCWE): This consists of adding a second trainable (warm) encoder to the codec, exactly the same as the original, which will receive the separated input, while the original will receive the ground truth. It can be initialised with the pre-trained or random weights, and will be trained to obtain the same embeddings or tokens (depending on the loss) as those obtained from the ground truth.

Encodec's encoder is a convolutional structure that starts with a 1D convolution with 32 channels and a kernel size of 7. This initial convolution is followed by 4 convolution blocks. These blocks consist of a single residual unit that leads to a downsampling phase performed by stepwise convolutions (kernel size is twice the step size S, set to [8, 5, 4, 2]). This residual unit contains two convolutions with kernel size 3 and a skip connection, and the channel capacity doubles each time it is downsampled. Following these convolution blocks, EnCodec incorporates a two-layer LSTM "for target sequence modelling" and concludes with a 1D convolution layer with a kernel size of 7 and 128 output channels. Non-linearity is introduced using the ELU activation function, complemented by weight normalisation techniques.

2) FCWE + intermediate dense layer: Same as the previous approach, but we add a dense layer that receives the embedding (warm or not) from the separated input and attempts to predict the correct categorical token, at all the residuals or only at a few of the first ones.

From a high-level perspective, the dense module acts as a compact classifier, translating embeddings into categorical probabilities. It's tailored to handle the 128-dimensional embeddings and translate them into a sequence of 32 vectors, where each vector has 1024 elements, each of which represents the probability that the embedding belongs to a particular category. Internally, the module first passes the embeddings through a linear layer that compresses the 128-dimensional embedding down to an intermediate 256 dimensions, applies a ReLU activation function for non-linearity, and passes the data into another linear layer that expands the 256 dimensions into a composite tensor comprising the product of the number of residuals (32) and the number of categories (1024).

3) FCWE + intermediate transformer: In this case, the intermediate layer is a small transformer that receives the embedding from the separated input, and attempts to predict the embedding from the ground truth, which can be the quantized embedding or before quantization.

The transformer consists of a positional encoding layer and three transformer encoder layers, which are responsible for transforming the input embeddings. It uses positional encoding at the input to gain an understanding of the sequence order (relevant because of the sequential nature of our data). The positional encoding module achieves this by mathematically constructing a mixture of sine and cosine functions which are added to the embeddings to give the model a sense of order in the data without altering the original information.

Each transformer encoder layer is defined by a particular model dimension and the number of attention heads. In our configuration, we've chosen 128 dimensions for our embeddings, with 4 attention heads. One of the defining aspects of the model is its autoregressive decoding approach. As it processes the embeddings, it takes into account not only the current step, but also all previous steps. This autoregressive nature ensures that the predicted output for a given step is informed by all previous steps, thus capturing the essence of the entire sequence.

Loss Functions Employed:

a) Mean Square Error (MSE) loss: Mean squared error between the derived embeddings and the ground truth embeddings, both post and pre-quantization.

$$\text{MSE}(e, \bar{e}) = \frac{1}{n}\sum_{i=1}^{n}(e_i - \bar{e}_i)^2$$

Where e is the ground truth embedding and $\bar{e}$ is the predicted embedding.

b) Categorical CrossEntropy (softmax loss): Calculates the disparity between the codebook selection probabilities across layers.

$$\text{H}(c, p) = -\sum_{i} c_i \log(p_i)$$

Where c is the target (ground truth) vector, a one-hot vector with a positive class and 1023 negative classes, and p is the predicted distribution.

c) Residual loss:

The Residual Loss we propose, presents a scheme wherein the embedding and the residuals are progressively approximated using quantized codebooks, layer by layer. The residuals are approximated by the sum of the codebooks for the layer, weighted by the probability of that codebook. This provides a differentiable approach to residual layers, essential for backpropagation and optimisation during model training. We pose:

$$p_i^0 = \textbf{softmax}(\frac{||c_i^0 - v^0||_2^2}{\tau}) = \frac{e^{-\frac{||c_i^0 - v^0||_2^2}{\tau}}}{\sum_j e^{-\frac{||c_j^0 - v^0||_2^2}{\tau}}} \tag{6}$$

$$v^1 = v^0 - \sum_i p_i^0 . c_i^0 \tag{7}$$

$$p_i^k = \textbf{softmax}(\frac{||c_i^k - v^k||_2^2}{\tau}) = \frac{e^{-\frac{||c_i^k - v^k||_2^2}{\tau}}}{\sum_j e^{-\frac{||c_j^k - v^k||_2^2}{\tau}}} \tag{8}$$

$$v^{k+1} = v^k - \sum_i p_i^k . c_i^k \tag{9}$$

$$c^k = \textbf{argmin}_{i \in [1, N=1024]} ||c_i^k - v^k||_2^2 \tag{10}$$

- c^k the quantized embedding at layer K
- τ a temperature,

- p_i^0 the probability of codebook i in the residual 0,
- c_i^0 quantized codebook i for residual 0,
- $v^0 = e$ the embedding from the encoder,
- v^j the residual embedding at residual j

Here the softmax operation ensures that the probabilities sum to one, and the negative sign in the exponent encourages the selection of codebooks that are closer to the embedding. The temperature parameter controls the sharpness of the probability distribution; a lower τ makes the probabilities more peaked, tending towards a single codebook, while a higher τ makes the distribution smoother.

What we call the Residual Loss is thus calculated as the sum of the classification loss (CrossEntropy loss) calculated for each layer based on the vector p^k. Allowing us to calculate a token-oriented classification loss but directly using the encoder's embedding, without the need of an intermediate layer that receives an embedding and outputs the probabilities.

An inherent challenge, very present in this last loss, but also an issue with the other implementations, is the propagation of errors across layers. If the approximation is suboptimal in earlier layers, the compounded residual error could introduce challenges in later layers.

d) Adversarial (discriminator) loss:

We also try to improve the training with adversarial components, in particular by integrating a discriminator. The role of the discriminator is to differentiate between real and fabricated embeddings, and to guide the encoder to produce more authentic representations.

The discriminator architecture is relatively simple: It consists of two linear layers interleaved by a ReLU activation function. The input embeddings, each of size 128, are first projected via the first fully connected layer to a hidden representation of size 64. The subsequent ReLU activation introduces non-linearity, and the resulting embeddings then pass through the second fully connected layer, which reduces their dimensionality to a single unit. This singular value is then passed through a sigmoid activation, ensuring its range between 0 and 1, representing the probability that the given embedding is genuine. The entire discriminator model is trained using Binary Cross Entropy (BCE) loss, which pushes it to make accurate binary distinctions between authentic and simulated embeddings.

With these variants in architecture and loss, we combined them in diverse ways, aiming to find the optimal configuration. A detailed account of these combinations and their corresponding results follows in the Experiments section.

6.2.2 Experiments

The previously introduced architectures, variants of the EnCodec architecture, and losses, were combined in different ways and trained on the MUSDB18 train set and/or on Bean. The table 2

shows how we combined them, and we will now go into detail about the experiments we ran and the results we obtained.

Losses\Architectures	**FCWE**	**FCWE+dense**	**FCWE+transformer**
MSE	✓	✓	✓
Softmax		✓	
Residual	✓		
Discriminator	✓		

Table 2: Tested combinations

Before delving into the experiments and variants expressed in Table 2, we will state some characteristics and parameters that were maintained across the experiments, and which we hope will help understand the upcoming experimental descriptions:

First of all, it is crucial to emphasise our distinction between the embeddings (those obtained directly from the encoder), the tokens (selected codes from the RVQ dictionaries, which can be thought of as integers from 0 to 1023 or as one-hot encoded vectors of dimension 1024), and the quantized embeddings (decoded with the RVQ from the token). When we talk about the ground truth (GT) embeddings, these are always derived from the frozen encoder, while the others are usually derived from the warm encoder.

For the experiments with MUSDB18, we refer to MUSDB18's *train set* (unless otherwise stated), from which we strategically reserved five songs and treated them as our validation set. Our first set of experiments -the first four to be exact- used Spleeter as the separation tool. For subsequent tests, however, we switched to Demucs.

It is important to note that the audio examples cited in this section and displayed on the website are predominantly from the validation set mentioned above, with a few exceptions such as the overfitting experiments, which will be explicitly stated.

In terms of our optimizer and learning rate hyperparameter, we always use Adam, with a default learning rate of 0.001, although in some cases we tested a more conservative $1 \exp^{-4}$ to potentially smooth the training dynamics.

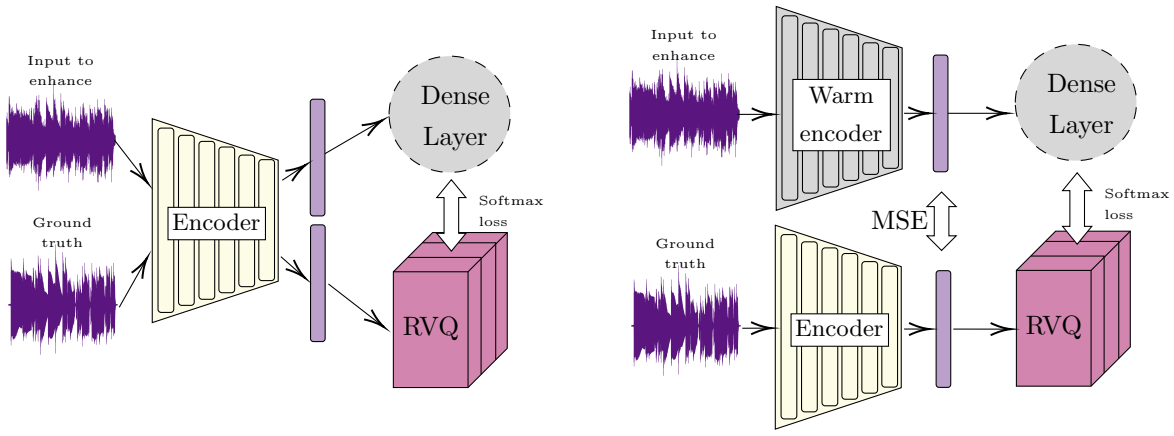
Going forward, it's important to keep these nuances in mind to ensure a full understanding of our experimental approach and subsequent results.

Experiment N1: Prediction of tokens with dense layer

Our first approach was to insert an intermediate dense layer and train it in the classification task of predicting the GT tokens for all the residuals from the embeddings of the separated input ("input to enhance" in Figure 16). This being a classification task, we use the softmax loss on the predicted

probabilities. The encoder was maintained with frozen weights and there was no parallel encoder, as can be seen in Figure 16a.

This was done on 16-second segments of bass recordings from the train set of MUSDB18. The loss converged, but the results were unintelligible; randomly noisy signals, with the dense architecture apparently unable to select tokens with any correlation to the input signal.



(a) Experiment N1

(b) Experiment N2

Figure 16: Training setup for experiments

Experiment N2: Adding MSE Loss

Our next step is to try to help the model by inserting the parallel encoder (initialised with the same weights), allowing it to update its weights, and adding a second loss between the embeddings of both encoders (see Figure 16b). Because of the different ranges of the two losses, we multiplied the latter by 100 so as to have them in comparable ranges[†]:

$$\text{Total Loss} = \text{Classification Loss} + 100 \times \text{MSE loss}$$

This new approach allowed the classification loss to converge to a much lower value, but the audio output was still very poor, producing more harmonically structured sounds, but still completely uncorrelated with the input.

To test whether the dense layer was too small, we did another test run where we overfit on many different segments of a single song in the dataset. We reconstructed the output from the selected tokens and also from the embeddings of the warm encoder, the interesting fact being that while the

[†]This is a common practice in the training of multi-loss models, though there are more sound ways of balancing the losses.

reconstruction from the predicted tokens was still terrible, the reconstruction from the embeddings was a decent reconstruction of the input.

This means that not only is the encoder not varying much from its initial state, but the intermediate layer is totally failing at doing the prediction.

Experiment N3: Getting rid of the intermediary

Following the previous results, in order to try a simpler approach that wouldn't be hindered by the training of the dense layer, which doesn't seem to be powerful enough for the task, we removed the intermediate layer and kept the parallel encoder, we trained this new encoder to predict the embeddings on MUSDB18, and we stopped training when the validation loss stopped going down (about 36 hours).

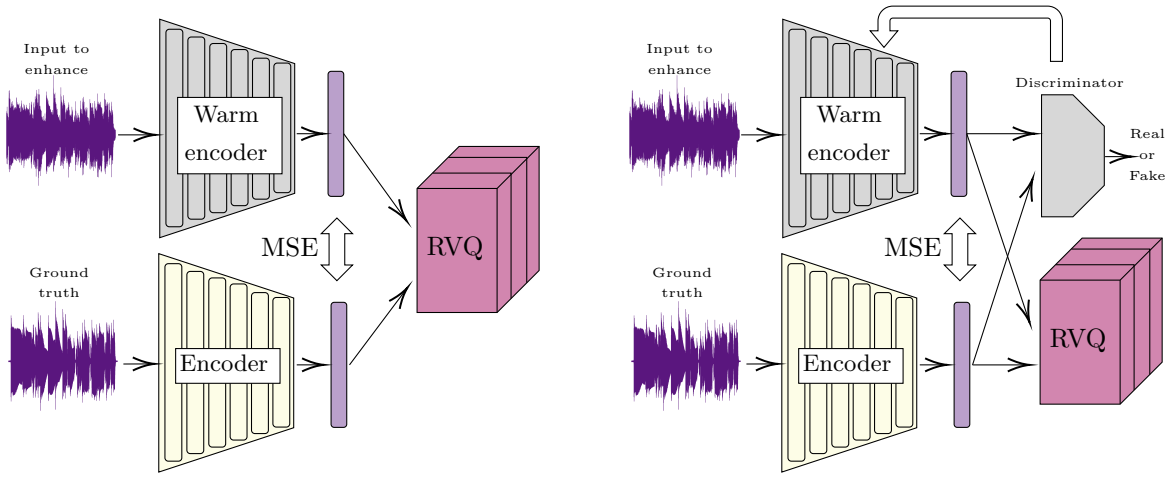
With this approach, we started to see a construction of the target signal where the attacks were mildly reconstructed but without a very natural sound (more noise-like), and some new high-frequency dither-like artefacts appeared. You can hear the reconstruction in example 6, a sample from the validation set.

Again, we repeated this experiment but trained to overfit on a single song, and we also adapted it to always train on the same 16-second segment to try to understand the encoder's ability to generate. Of course, both runs reconstructed the target songs much better than when using the whole dataset, but both showed an interesting insight: the encoder was good at filtering artefacts and recovering some of the high frequencies (at the expense of noise), but was not good at the task of adding strong and clear attacks or other missing frequency information, as can be heard in example 7, which shows the reconstruction of the segment we trained on.

We repeated this training on the drums of MUSDB18 to check that the approach was still generalisable. The training progressed in a similar way, and for the drums we again observed the clear phenomenon of the model struggling to add missing frequency components. It was quite adept at filtering out some artefacts, but it still added some noise and failed to generate much new content.

To further explore this hypothesis that the system was good at filtering (up to a point), we ran a small test in which instead of inputting the output of the Spleeter, we input the ground truth with random white noise added, thus training the warm encoder to denoise the input. Example 8 shows that the model performed this task with reasonable success, although there was still noise over the sound of the bass notes, which is to be expected, as there is actually also audible noise in the quantized target.

In order to see if we could get the encoder to train towards a more generative state, we also tried training for bass reconstruction from the Spleeter output, but initialising the parallel encoder randomly instead of using the pre-trained encoder weights. This did not work well, as it converged to a point with much higher loss, and the audio results were unintelligible.



(a) Experiments N3-5 (N4 w/Residual loss in RVQ)

(b) Experiment N6

Figure 17: Training setup for experiments

Experiment N4: Adding the Residual loss

Under the hypothesis that the MSE loss was not sufficient to force the encoder to learn the correct embeddings, we decided to try another token-oriented approach, but without the need for an intermediate layer. This is the Residual loss that we explained in the methodology, where we approximate the residuals by the sum of the embeddings in the dictionary weighted by the probability of that embedding.

With this loss, we have a loss for each layer (which we add to get a total loss), and we also compute the accuracy of token selection at each layer, which gives us a good insight into how our model is performing. We train to overfit on a single bass song and find that, once again, the loss does not produce gradients strong enough to force the encoder to create a cleaner output. In terms of sound quality, this approach performs worse than the previous experiment, as can be heard in example 9 of our website.

If we look at the accuracy of the token selection, we see a slight improvement throughout the training: the first two layers do not improve much from the starting point, with an accuracy of around 60% and 25% respectively, but the posterior layers gain around a 2% improvement in classification, which is not negligible if we consider that the accuracy is normally very low (between 2 and 8%).

Taking into account these new findings of low performance in token selection, we switched from using the Spleeter output to the output of Demucs v4, which reports a better performance in the 4-stem separation task. We repeated the training in the same way, but with the different input, observing that the accuracy went up to 67% for the first layer, and improved by 2 or 3% for the

subsequent layers. However, we continued to see the phenomenon of the encoder not being able to improve its performance, with the losses stagnating close to their initial values.

Experiment N5: Back to Basics with Demucs

It appeared that this loss did not help our model to improve, even in the case where we were overfitting on one song. To confirm this, we repeated experiment N3 but with the Demucs input and recorded the token selection accuracy, and found that the accuracy was slightly better in this case, by 5% for the first layer and around 1% for the rest. This was even more noticeable when we trained on the whole MUSDB train set, and noticed that in fact, the accuracy for the whole data set was much worse: with an average of about 25% for the first layer and less than 1% for the others.

When this last experience was repeated on the drums dataset, the results were much better, improving the accuracy of the first layer from 25% to 55%, and the posterior five layers to over 10%. The problem with the drums reconstruction is that the errors in the higher layers seem to introduce harmonics in the high frequencies when trying to recreate the spatial component in the drum recordings.

Experiment N6: More losses

Before changing the architecture, we made two other attempts to add losses to the MSE loss that would help us to force the model closer to the goal, these were to use a Residual loss, but only on the first four layers of the RVQ, and then to also add the discriminator loss obtained from the simple discriminator that distinguishes between the embeddings obtained from the ground truth by the frozen encoder and the embeddings generated by the trained parallel encoder from the bass or drums separated with Demucs.

We hoped that these changes, especially the latter, inspired by generative model approaches, would allow the model to add missing components to the separated tracks, but there was virtually no change in the MSE distance between embeddings, in the token accuracy, or in the generated audio quality.

Experiment N7: Adding a Transformer

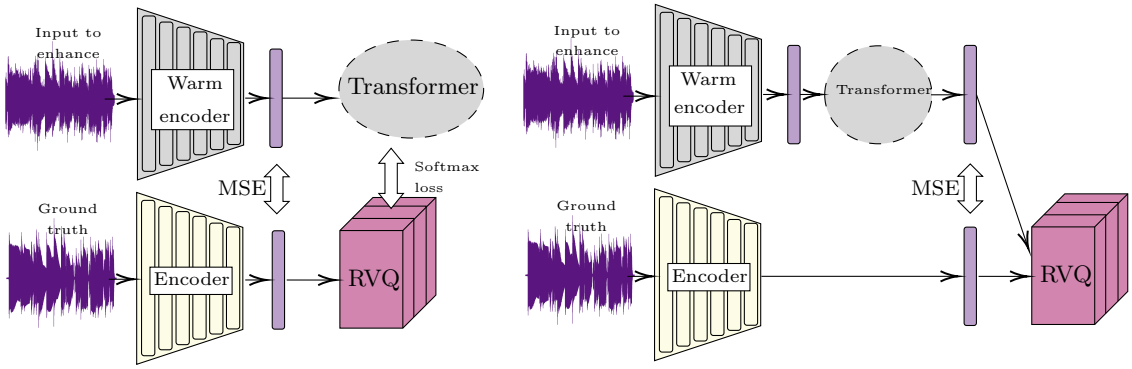
At this point, we had exhausted most of our options for using only EnCodec's encoder to generate the embeddings and achieve decent results in reconstructing the ground truth. We were still hoping to get a good accuracy on the coarse tokens that would allow posterior reconstruction using a coarse-to-fine architecture, but at the moment our best accuracy even for the first layer was only 25% on bass and 55% on drums.

Most SOTA that use RVQs for generation are based on transformer architectures, so we decided to implement a small transformer as an intermediate layer. Instead of using a token translation paradigm, where the transformer received the tokens of the separated signal to predict the tokens

from the ground truth, generating its own embedding space in the process, we decided to take advantage of the meaningful embeddings generated by EnCodec and input them to predict tokens, in the same way as we had done with the dense intermediate layer, but in this case, focusing only on predicting the *first* fine token.

Training was done on the bass from the MUSDB train set, using the softmax loss. It achieved 60% accuracy on the MUSDB train and validation subset, but the audio reconstruction from this first layer was terrible, so we tried predicting the first four quantizers (the first four quantizers are often referred to as the coarse tokens[11][56]). The accuracy here was terrible, dropping to an average accuracy of 10% across the four layers, and the reconstruction was still unintelligible.

We tried to improve performance by changing hyperparameters such as batch size, using a larger transformer, allowing the parallel encoder to train as well as the transformer, or loading a pre-trained transformer that had been trained to do the quantization (go from the GT embedding to the GT token), but this was not possible.



(a) Experiment N7

(b) Experiment N8

Figure 18: Training setup for experiments

Experiment N8: Last ditch effort with transformers

We believe that the bad performance had to do with how the transformer handled the transfer from the embedding space to the corresponding tokens in its internal decoder. The relatively small amount of data available, coupled with our limited experience with transformers - a notoriously complex and difficult-to-train architecture - was also a major source of difficulty. We decided to target the embeddings of the EnCodec encoder directly, rather than the tokens. In this way, we relied on an internal embedding-to-embedding layer, hoping that the transformer would team up with the encoder and provide us with an architecture that offered certain generative capabilities.

We ran two variants of this experiment, one with the transformer receiving the embedding from the original frozen EnCodec encoder and predicting the ground truth embedding from the same

encoder. The other was with the previously used parallel encoder, also trained. The best results were obtained with the latter method, with an average accuracy over the 32 quantizers of around 10%, but the results do not represent a significant improvement, as can be heard in example 11.

In summary, the process of fine-tuning EnCodec for audio enhancement has presented numerous challenges. Despite extensive experimentation with different architectures and losses, achieving high-quality sound reconstruction remained elusive. Further research and exploration of more advanced models or techniques may be essential to achieve the desired results.

6.3 Mixture Separation Approach

From the moment we noticed the good filtering capabilities of the models, we decided to explore the avenue of filtering with the encoder, instead of attempting enhancement. Our first attempt was to input white noise and see how the system learnt to generate bass sounds. We trained this using the MSE loss and overfitting on a single bass song. The results were rhythmic outputs but no clear bass sounds, and as the potential of this test was limited, we moved on to the more relevant task of inputting the mix and using the system to filter and output an instrument; basically performing the separation with the encoder.

This new approach performed decently when trained on the MUSDB train set using the basic FCWE architecture with MSE loss. Examples 12 and 13 show the separation obtained on samples from the validation set for both bass and drums.

We then ran the same experiments as we had done with the separated signal as input, but with the mixture as input. We ran the same configuration as in Experiment 6 (with the discriminator), which again did not improve performance. The addition of the transformer (embedding-to-embedding), however, gave an improvement in the generated sound, with fewer artefacts in the generated output. Examples 14, 15 and 16 show the results obtained for the separation of bass, drums and the other track.

The transformer architecture gives the cleanest results, so we decided to test it with two other encoders: the RAVE codec, which we had trained ourselves for bass reconstruction, and the DAC codec, which was brought in towards the end of the internship[58].

We trained the architecture using the RAVE codec and its 16 layer RVQ instead of EnCodec, with the intermediate transformer layer, on MUSDB using the MSE loss. The reconstruction quality (which you can hear in the quantized target of example 17) was worse than EnCodec's reconstruction, so our ceiling is a little lower. The separation itself was also slightly worse, as you'll hear. We think that this may be due to the Pseudo Quadrature Mirror Filters (PQMF) layer at the input and output of the model, which is part of RAVE's focus on exploring musical changes in latent space, rather than just timbral changes. Also, since the encoder hasn't seen any non-bass music in its training, it's unlikely to be able to extract the relevant information from the

input mix, making reconstruction more difficult.

With the DAC encoder we did the same with its 14 quantizers, achieving good results, but we also experimented with changing our target to be the quantized embedding using only 4 quantizers, in order to get a higher accuracy on the first 4 layers, which are the coarse tokens for the VampNet coarse2fine transformer [56], this improved the performance. A final experiment was carried out without the transformer in order to be able to insert the separating encoder into the VampNet architecture.

7 Token Prediction with Transformers

Transformers, well known for their contributions to natural language processing, have often been used for translation tasks. Given the idea behind our work of aligning separated audio codec tokens to ground truth tokens, it was easy to view the task through the lens of a translation problem. Transformers are also present in the coarse-to-fine paradigm presented in AudioLM, proving their suitability for such tasks.

While we'd already tried to incorporate a transformer as an intermediate layer, focusing on refining the embedding itself, we also decided to try a more direct translation approach, where the transformer would receive the tokens derived from the separated audio, process them, generating its own embeddings, and output tokens that matched the ground truth. Despite our hopes, this approach did not produce satisfactory results.

However, towards the end of the internship, VampNet presented a transformer specifically designed for the coarse and coarse-to-fine prediction tasks (see Figure 6). Accompanied by training and fine-tuning codes and pre-trained models, it was an exciting avenue to explore. To assess its potential, we trained these transformers on our bean dataset. We also fine-tuned the DAC codec used for bass extraction in the same manner we had done with EnCodec and RAVE.

Experiments

Using the VampNet architecture presented in the SOTA section 2.2.3, we sought to approach the challenge as a token translation task, capitalising on the effectiveness of transformers.

Despite the recognised power of transformers, our first-hand experience with them was limited. Their complex architecture, combined with the nuances of hyper-parameter tuning, made them somewhat difficult to train.

Initial attempts to fine-tune a T5 architecture for token prediction were unsuccessful, as were our efforts with the architecture formulated by LucidRains.

However, a turning point came towards the end of the internship with the release of the VampNet transformer. This prompted us to start some experiments such as training from scratch, fine-tuning

the model and exploring its reconstruction capabilities.

Below are summarised experiments using the VampNet transformer models (specifically the coarse prediction model and the coarse2fine prediction model). It's worth noting that VampNet uses the Descript Audio Codec and its associated tokens. The audio samples in the examples were taken from the MUSDB18 training set, which was not part of the training set of the VampNet transformers, and served as the validation set.

Experiment 1: Reconstruction using Descript pre-trained models (baseline)

This experiment was anchored on VampNet transformers that had been previously trained on different sound types.

We have listened to and displayed the results of the following two experiments:

- Reconstructions where the latter 10 fine tokens are predicted based on the coarse tokens of the input only.
- Reconstructions where the 2nd, 3rd and 4th tokens are derived from the initial token using the coarse model, and the subsequent 10 tokens are then predicted from these primary four.

Table 3 shows the objective separation metrics computed on the outputs of the two experimentas and compared with the input's score, and with the input's reconstruction's (using the DAC codec and its 14 tokens) score. The metrics were computed over 10-second segments extracted from the 100 songs of the MUSDB train set.

	Input (Demucs)	**Reconstructed**	**Pretrained VampNet**		**Bass VampNet**	
			c2f	**1 coarse**	**c2f**	**1 coarse**
SDR	10.62	10.03	8.32	8.31	7.44	7.40
SAR	11.46	10.83	9.13	9.14	8.52	8.45
ISR	16.72	16.32	15.62	15.61	14.87	14.85

Table 3: Objective separation metrics on VampNet reconstruction. Input separated w/Demucs

The reconstruction here presented some interesting characteristics, that you can hear on our website. The first we noticed was that the input had some minimal bleeding from drums and other instruments, but the reconstruction had none of this: clearly the bleeding was in the fine tokens and when reconstructing them from the coarse, there was none of this bleeding. The second one, was that even with the reduction of this bleeding, and with a very good sounding reconstruction, that was to our ears at least equal to the separated input. As can be seen in Table3 SDR went down, as the reconstructed fine tokens had nothing forcing them to reconstruct the target signal.

One pitfall we heard, is that the reconstruction sometimes added sounds that do not belong to the family of sounds that could be produced by a bass. This is of course understandable, as the model was trained to reconstruct music, and when it receives bass at the input, it has no reason

to believe there shouldn't be other instruments in the background. This could be addressed with the next experiment.

Experiment 2: Reconstruction using coarse and coarse2fine transformers trained from scratch on Bean

We repeated the same two experiments, but with a VampNet model trained exclusively on predicting bass tokens, thus offering a narrower range of sound prediction.

Again we observed an improvement in the sound, in this case it was even better because it reconstructed a clear-sounding bass, and in this case with no non-bass artefacts. The sound quality of the reconstruction was not as great as with the pre-trained model(though it was extremely good), and again the SDR went down when compared with the input (Demucs' output), very possibly because of this.

Experiment 3: Training coarse and c2f models from scratch on Bean with input from Demucs and target from GT

In this experiment, we attempted to train the transformers to do the token domain translation, from the separated input to the ground truth, so as to perform the enhancing in a deliberate fashion. We did not alter the VampNet training pipeline for this, only the target tokens would now be different from the input, keeping the masking scheme they propose. Unfortunately, the model training for this experiment encountered a snag in its training. Despite several attempts to fix it, the problem persisted. Addressing this challenge is planned for future research.

Experiment 4: Reconstruction using the codec fine-tuned for separation

Our final experiment with VampNet was to repeat the first two reconstruction experiments, but instead of using the pre-trained DAC codec, we replaced it with our fine-tuned DAC codec, which we had fine-tuned to receive the mixture and filter out the instruments to perform bass extraction. In this way, we input the mixture, and in the output we got the extraction done by the codec's encoder, enhanced by VampNet's coarse and coarse2fine predictions.

	Input (Demucs)	**Reconstructed**	**Pretrained VampNet**		**Bass VampNet**	
			c2f	**1 coarse**	**c2f**	**1 coarse**
SDR	-0.41	2.76	2.84	2.83	3.05	3.05
SAR	-6.79	0.55	0.48	0.49	0.31	0.38
ISR	5.74	4.21	4.63	4.66	5.52	5.51

Table 4: Separation metrics on VampNet reconstruction from mixture, separated w/DAC encoder

The results were decent: the simple reconstruction of the separated bass using only the codec already improved the SDR compared to the mixture, and the interesting fact was that the reconstructions obtained using the VampNet transformers improved the SDR even more in this case.

Obviously, the reconstruction is still very much conditioned by the extracted bass signal, and since the separation with the encoder is not as good as that obtained with Demucs, the predicted coarse and fine tokens improve the separation, not only in terms of sound quality but also in terms of objective metrics.

This approach is a first attempt at what could be an end-to-end approach to source separation following the paradigm proposed in our work.

8 Discussion

Having chosen to work on such an active topic as neural codecs and language models, our choices throughout the internship were affected by a large amount of late research in the field. MusicLM was published only a month before the start of the internship, MusicGen just after the halfway point but without training code, and VampNet only six weeks before the end of the internship.

Throughout our experiments, we pursued various approaches and methodologies with the goal of achieving a viable token prediction mechanism for audio reconstruction, enhancement and separation. Upon reflecting on the experiments and the results obtained, several key observations emerge.

One of the first observations is that the existing methods for sound separation enhancement, which follow the lines of state-of-the-art source separation techniques, does not notably improve the sound quality, despite their influence on the SDR score. This underlines the inherent challenges of the task and questions the extent to which existing benchmarks can be pushed further.

With this in mind, we set out to explore a new avenue using token-based audio generation. The aim was not just to approximate the ground truth of separation, but also to improve the sound quality, even if it meant deviating from the ground truth. However, the path to achieving this paradigm of separation enhancement proved to be a complex one.

Our proposed approach for mixture separation presents a potentially intriguing prospect. The incorporation of an encoder as the primary driver for sound separation offered an interesting path for end-to-end separation, and, when combined with the transformer, yielded encouraging results that could certainly be further improved.

We explored numerous architectural variations, but the level of success did not always correspond to our efforts. The field of token-based audio generation is still at an early stage; in an ideal scenario we would have had open-source transformer architectures with training pipelines, which whilst intended for different tasks such as unconditional music generation or speech generation, would have greatly expedited our progress. Adapting these models to our specific needs would have been a simpler undertaking than constructing anew or building from non-operational work in progress systems, especially given our limited prior experience.

Our trials with the T5 transformer and the architectures implemented by LucidRains did not produce the desired results. Although the VampNet transformer holds great promise for future developments, it came too late for us to explore in depth within the timeframe of the internship. If we had been able to allocate more time and examine these models more extensively, our account might have been very different.

Looking forward, it's clear that there are still many unexplored avenues in the field. The nexus of token prediction, sound reconstruction and enhancement remain fertile ground for innovation. While progress has been made, the road ahead is laden with potential and hopefully invites future research to continue in this direction.

9 Conclusions

The path of this research has been both rewarding and challenging, delving deep into the realms of token-based audio generation and codec-based architectures for sound separation and separation enhancement.

9.1 Main Contributions

In this study we have set out to confront what we believe are the prevailing challenges encountered within the domain of source separation. One of the main contributions within this dissertation resides in the conceptualisation and systematic exploration of a novel approach to source separation. This approach distinctly diverges from conventional methodologies that primarily concentrate on the improvement of quantifiable metrics, notably the SDR, and that are inherently limited by the confines of the traditional 4-stem separation task. We believe that although there has been considerable progress in the field, it has become apparent that we have reached a plateau in performance, particularly when assessing the sonic quality of the separation. Frequently, artefacts are introduced into the separated tracks, rendering the output unconvincing and inconsistent with expectations of how a particular instrument -such as drums or bass- should sound.

We propose a novel approach that seeks to prioritise the consistency of the generated sound with the expected timbre, and the dynamics and overall manner of sounding of the instrument, rather than rigidly adhering to the ground truth. The proposed paradigm of separation and separation enhancement focuses on improving the perceptual quality of the output audio, and in pursuit of this goal chooses not to attach great significance to deviations from the target in terms of classical evaluation metrics.

The choice of focusing on token-based audio generation - particularly with strict conditioning - for the enhancement of complex musical sounds obtained from the separation of a song proved to be an interesting and promising one. This paradigm is particularly interesting given that token-based methods have only recently entered the realm of audio processing and generation.

Codec-based frameworks are gaining popularity and are becoming the cornerstone of cutting edge generative applications. While there has been a surge of activity in this area, their application to musically driven tasks such as separation or enhancement remains unexplored. In this context, our exploration serves as a first step in delineating the scope and potential of codecs when tailored to highly conditioned tasks.

Our proposal of training and fine-tuning models by aiming at a compressed and informative representation of the signals aligns with SOTA proposals in the field of generative machine learning. By focusing on the representations obtained from neural codecs, investigating how certain information is encoded in them and exploiting their architectural flexibility, our work offers a valuable contribution to future work in the area of token-based audio generation.

9.2 Future Work

Looking ahead, there are many avenues to explore and challenges to overcome. One salient task would be to focus on relevant metrics and audio evaluation. While classical metrics such as SDR provide a satisfactory foundation, they do not cover the full range of perceptual features in audio. Following the suggestions of Schaffer et al.[22], there is potential in integrating classical MIR features such as spectral centroid, spectral flatness, contrasts and loudness, among others, so as to enrich the quantitative evaluation possibilities. A comprehensive evaluation should also integrate subjective evaluation methods in an attempt to bridge the gap between empirical metrics and auditory perception.

While our research was ambitious in scope, a frank introspection reveals a relative scarcity of rigorous evaluations. As our outputs often appeared to be qualitatively inferior to the prevailing state-of-the-art frameworks, extensive evaluation was often dispensed with. However, as the research moves forward, this will undoubtedly become crucial.

In addition, forthcoming efforts should examine the capability of these frameworks to generalise to a wide range of instruments or sources. To this end, the use of resources such as the MoisesAI multi-instrument source separation database[67] could be advantageous. The development of specialised instrument codecs, together with an adeqaute framework to exploit them would be of great value, not only to improve fidelity, but also to optimise efficiency, as the narrower range of possibilities could allow for smaller codebooks. Conversely, the widening of the information bottleneck imposed by the codec could improve audio quality, given that we are not interested in audio compression.

In closing, we believe that this research, with its novelties and challenges, serves as both an endpoint and a starting point - a culmination of our efforts to date and a pointer to the many possibilities that lie ahead.

References

[1] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ - an uncompressed version of MUSDB18, 2019-08-01. Type: dataset.

[2] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, Fabian-Robert Stöter, Alexandre Défossez, Minseok Kim, Woosung Choi, Chin-Yun Yu, and Kin-Wai Cheuk. Music demixing challenge 2021. *Frontiers in Signal Processing*, 1, 2022.

[3] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, June 2020.

[4] Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung. LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation, April 2021. arXiv:2010.11631 [cs, eess].

[5] Alexandre Défossez. Hybrid Spectrogram and Waveform Source Separation, August 2022. arXiv:2111.03600 [cs, eess, stat].

[6] 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.

[7] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms, January 2019. arXiv:1812.08466 [cs, eess].

[8] 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, 2022.

[9] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music, April 2020. arXiv:2005.00341 [cs, eess, stat].

[10] Marco Pasini and Jan Schlüter. Musika! Fast Infinite Waveform Music Generation, August 2022. arXiv:2208.08706 [cs, eess].

[11] 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, February 2023. arXiv:2301.12503 [cs, eess].

[12] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating Music From Text, January 2023. arXiv:2301.11325 [cs, eess].

[13] Nikhil Kandpal, Oriol Nieto, and Zeyu Jin. Music Enhancement via Image Translation and Vocoding, April 2022. arXiv:2204.13289 [cs, eess].

[14] Stefan Lattner and Javier Nistal. Stochastic Restoration of Heavily Compressed Musical Audio Using Generative Adversarial Networks. *Electronics*, 10(11):1349, January 2021. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec, July 2021. arXiv:2107.03312 [cs, eess].

[16] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High Fidelity Neural Audio Compression, October 2022. arXiv:2210.13438 [cs, eess, stat].

[17] Jiaqi Su, Zeyu Jin, and Adam Finkelstein. HiFi-GAN: High-Fidelity Denoising and Dereverberation Based on Speech Deep Features in Adversarial Networks, September 2020. arXiv:2006.05694 [cs, eess].

[18] Jiaqi Su, Zeyu Jin, and Adam Finkelstein. HiFi-GAN-2: Studio-Quality Speech Enhancement via Generative Adversarial Networks Conditioned on Acoustic Features. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 166–170, New Paltz, NY, USA, October 2021. IEEE.

[19] Yunpeng Li, Beat Gfeller, Marco Tagliasacchi, and Dominik Roblek. Learning to Denoise Historical Music, June 2022. arXiv:2008.02027 [cs, eess].

[20] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Adversarial Semi-Supervised Audio Source Separation applied to Singing Voice Extraction, April 2018. arXiv:1711.00048 [cs].

[21] Genís Plaja-Roglans, Marius Miron, and Xavier Serra. A diffusion-inspired training strategy for singing voice extraction in the waveform domain. In *Proceedings of the First MiniCon Conference*, December 2022.

[22] Noah Schaffer, Boaz Cogan, Ethan Manilow, Max Morrison, Prem Seetharaman, and Bryan Pardo. Music Separation Enhancement with Generative Modeling, August 2022. arXiv:2208.12387 [cs, eess].

[23] Emmanuel Vincent, Nancy Bertin, Remi Gribonval, and Frederic Bimbot. From Blind to Guided Audio Source Separation: How models and side information can improve the separation of sound. *IEEE Signal Processing Magazine*, 31(3):107–115, May 2014.

[24] 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, September 2016.

[25] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, October 1998. Conference Name: Proceedings of the IEEE.

[26] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. Number: 6755 Publisher: Nature Publishing Group.

[27] Alexey Ozerov and Cédric Fevotte. Multichannel Nonnegative Matrix Factorization in Convolutive Mixtures for Audio Source Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, March 2010. Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

[28] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software*, 2019.

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

[30] Woosung Choi, Minseok Kim, Jaehwa Chung, Daewon Lee, and Soonyoung Jung. Investigating U-Nets with various Intermediate Blocks for Spectrogram-based Singing Voice Separation, October 2020. arXiv:1912.02591 [cs, eess, stat].

[31] Qiuqiang Kong, Yin Cao, Haohe Liu, Keunwoo Choi, and Yuxuan Wang. Decoupling Magnitude and Phase Estimation with Deep ResUNet for Music Source Separation, September 2021. arXiv:2109.05418 [cs, eess].

[32] 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, August 2019. arXiv:1809.07454 [cs, eess].

[33] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation, March 2020. arXiv:1910.06379 [cs, eess].

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. arXiv:1505.04597 [cs].

[35] 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.

[36] Recommendation ITU-R BS.1534-3. Method for the subjective assessment of intermediate quality level of audio systems. *R BS.*, 2014.

[37] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre. webMUSHRA — a comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1):8, 2018-02-05. Number: 1 Publisher: Ubiquity Press.

[38] Sławomir Zieliński, Philip Hardisty, Christopher Hummersone, and Francis Rumsey. Potential biases in MUSHRA listening tests. *Audio Engineering Society - 123rd Audio Engineering Society Convention 2007*, 2, 2007-01-01.

[39] Max Morrison, Brian Tang, Gefei Tan, and Bryan Pardo. Reproducible subjective evaluation, 2022-03-08.

[40] Guanxin Jiang, Lars Villemoes, and Arijit Biswas. Generative Machine Listener, August 2023. arXiv:2308.09493 [cs, eess].

[41] Mark Cartwright, Bryan Pardo, Gautham J. Mysore, and Matt Hoffman. Fast and easy crowdsourced perceptual audio evaluation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 619–623, 2016.

[42] Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, and Jesse Engel. SingSong: Generating musical accompaniments from singing, January 2023. arXiv:2301.12662 [cs, eess].

[43] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and Controllable Music Generation, June 2023. arXiv:2306.05284 [cs, eess].

[44] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014. arXiv:1406.2661 [cs, stat].

[45] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. arXiv:2006.11239 [cs, stat].

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. arXiv:1706.03762 [cs].

[47] 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, September 2022. arXiv:2209.03143 [cs, eess].

[48] Antoine Caillon and Philippe Esling. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis, December 2021. arXiv:2111.05011 [cs, eess].

[49] Gaku Narita, Junichi Shimizu, and Taketo Akama. GANStrument: Adversarial Instrument Sound Synthesis with Pitch-invariant Instance Conditioning, March 2023. arXiv:2211.05385 [cs, eess].

[50] Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. BigVGAN: A Universal Neural Vocoder with Large-Scale Training, February 2023. arXiv:2206.04658 [cs, eess].

[51] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models, February 2021. arXiv:2102.09672 [cs, stat].

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

[53] Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Mo\ˆusai: Text-to-Music Generation with Long-Context Latent Diffusion, January 2023. arXiv:2301.11757 [cs, eess].

[54] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. AudioLDM 2: Learning Holistic Audio Generation with Self-supervised Pretraining, August 2023. arXiv:2308.05734 [cs, eess].

[55] Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, Hannah Sheahan, Neil Zeghidour, Jean-Baptiste Alayrac, João Carreira, and Jesse Engel. General-purpose, long-context autoregressive modeling with perceiver ar, 2022.

[56] Hugo Flores Garcia, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo. VampNet: Music Generation via Masked Acoustic Token Modeling, July 2023. arXiv:2307.04686 [cs, eess].

[57] Antoine Caillon. *Hierarchical temporal learning for multi-instrument and orchestral audio synthesis*. phdthesis, Sorbonne Université, February 2023.

[58] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-Fidelity Audio Compression with Improved RVQGAN, June 2023. arXiv:2306.06546 [cs, eess].

[59] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning, May 2018. arXiv:1711.00937 [cs].

[60] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2, June 2019. arXiv:1906.00446 [cs, stat].

[61] 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, June 2022. arXiv:2110.04627 [cs].

[62] 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, July 2020. arXiv:1910.10683 [cs, stat].

[63] Laure Prétet, Romain Hennequin, Jimena Royo-Letelier, and Andrea Vaglio. Singing voice separation: a study on training data. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 506–510, May 2019. arXiv:1906.02618 [cs, eess].

[64] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music Source Separation in the Waveform Domain, April 2021. arXiv:1911.13254 [cs, eess, stat].

[65] Phil Wang. Audiolm - pytorch. `https://github.com/lucidrains/audiolm-pytorch`, 2023.

[66] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture, April 2023. arXiv:2301.08243 [cs, eess].

[67] Igor Pereira, Felipe Araújo, Filip Korzeniowski, and Richard Vogl. Moisesdb: A dataset for source separation beyond 4-stems, July 2023. arXiv:2307.15913 [cs, eess].