ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed
Computing

Bridging Image and Audio Compression: A Spectrogram-based Neural Approach

Master's Thesis

Philipp Dasen

dasenp@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Luca Lanzendörfer, Nathanaël Perraudin
Prof. Dr. Roger Wattenhofer

April 8, 2024

Acknowledgements

I thank my advisors Luca Lanzendörfer and Nathanël Perraudin for their helpful mentorship and boundless patience. I would also like to thank my friends and family for their encouragement, positivity and wisdom. It would be futile to attempt to fairly place them all on a list here, if you are reading this you probably are part of it though.

fam - Lil B

Abstract

This thesis explores a novel approach to audio compression using image compression techniques applied to spectrogram representations of audio signals. By converting audio into two-dimensional spectrograms and leveraging state-of-the-art image compression models, the proposed method aims to achieve competitive compression rates while maintaining perceptual quality. The study focuses on compressing music and general audio, which present unique challenges compared to speech-specific compression. The pipeline involves converting audio into log-magnitude spectrograms, applying an image compression model (ILLM), and reconstructing the audio waveform using phase reconstruction techniques. Experiments demonstrate that while the spectrogram-based approach can achieve bitrates comparable to state-of-the-art neural audio codecs, the perceptual quality currently lags behind. The finetuned ILLM model outperforms its pretrained counterpart on a consistency metric but still introduces artifacts. The study highlights the need for perceptually-aligned evaluation methods and identifies key challenges, such as adapting model architectures to capture the unique characteristics of audio spectrograms and improving phase reconstruction. Despite limitations, the results show the potential for leveraging image compression techniques in the audio domain and serve as a foundation for future research in spectrogram-based audio coding.

Contents

Chapter 1
Introduction

Data compression is a process by which data (usually in digital binary form) is represented using fewer bits of information than in its original representation. In the usual picture a data compression system consists of two algorithms, an encoder and a decoder algorithm (the pair of which is sometimes referred to as "codec"). The encoder takes as input the uncompressed data and returns the compressed representation of the data as an output. The decoder then executes the process by which the compressed data is transformed back into its original data format. The main distinction between different types of compression algorithms is in whether they are lossless or not. Lossless compression refers to such algorithms that can perfectly reconstruct the original data from its compressed representation, while lossy compression algorithms have purposely been designed to discard parts of the source information, which allows a better level of compression, at the cost of only performing an imperfect reconstruction.

Data compression plays a crucial role in conserving storage hardware and in reducing network computing resources, particularly for multimedia data such as images, audio, and video. Efficient compression techniques are essential. Without them, storing and transmitting this content would require an enormous amount of hardware resources and network bandwidth. By minimizing the storage footprint through compression, data centers can operate more efficiently, leading to significant economic savings. Furthermore, compressed files consume less bandwidth during transmission, which translates to lower energy consumption in network infrastructure. As the world becomes increasingly digital, the amount of multimedia content generated and consumed continues to grow exponentially, totalling more than 5000 Exabytes in 2022 [1]. It is estimated that the majority of internet traffic is now being generated by video streaming applications [2].

The field of audio compression is concerned with the development of specialized data compression techniques to compactly represent arbitrary sound signals or some subset of possible audio information (such as speech). Similar to other digital representations of analog signals (images, video), compressed digital audio data usually is not required to be perfectly reconstructed on a bit-by-bit basis.

MP3 [3] has been a widely used audio compression format for many years, offering efficient compression while maintaining acceptable audio quality. More recently, the Opus codec [4] has emerged as a popular alternative, providing improved compression efficiency and audio quality. As technology advances, researchers are developing new compression algorithms that leverage deep learning and machine learning methods. Neural network-based approaches outperform traditional compression methods at very low bitrates, making them an exciting area of research with significant potential for practical applications and as base models for downstream applications.

Neural audio codecs are typically trained end-to-end, meaning the entire compression and decompression pipeline is learned directly from data. This allows the models to optimize for the specific characteristics of the audio signals being compressed. Our approach takes a different route by working with a spectrogram representation of the audio data and applying an image compression model to it.

The key research questions we aim to address with our spectrogram-based audio compression approach are whether it is possible to obtain similar or better audio quality than traditional codecs and end-to-end neural approaches by working with spectrograms, if model training can be simplified by leveraging existing pretrained image compression models and whether our method can provide improved size efficiency, which is crucial for deployment on resource-constrained devices. By exploring these questions, we hope to develop a novel audio compression technique that combines the benefits of deep learning with the advantages of working in the spectrogram domain. This could lead to more efficient and effective audio compression, with applications in various domains such as streaming, telecommunications, and embedded systems.

1.1 Lossless Compression

The invention of Morse code in the middle of the 19th century likely is the first example where a system was deliberately engineered with an early understanding of the principles underlying data compression. By giving the shortest codewords to the most frequently occurring letters in the alphabet, the expected length of telegraph transmissions was reduced. Clearly, the efficiency with which information is represented is of vital importance if one aims to maximize the transmission rate of a communication system. Unsurprisingly, compression was therefore one of the main motivating problems addressed by Shannon's famous 1948 paper [5] that derived many of the main results in the field of information theory.

The most important bound for lossless compression is Shannon's source coding theorem. Let $\mathbf{X}$ be a discrete random variable over an alphabet $\mathcal{X}$ with distribution P. We define the entropy $H(\mathbf{X}) = \mathbb{E}_{\mathbf{x}\sim P}[-\log_2(\mathbf{x})]$, the expected code-length L of an optimal code with symbols in a D-ary alphabet can be bounded [6]:

$$\frac{H(\mathbf{X})}{\log_2(D)} \leq L < \frac{H(\mathbf{X})}{\log_2(D)} + 1 \tag{1.1}$$

We notice that there is an overhead of at most 1 bit per symbol. One may group the sequence of symbols into larger blocks to amortize this overhead over greater portions of a message. One can therefore construct lossless compression codes that achieve an expected code-length per symbol that is arbitrarily close to the entropy $H(\mathbf{X})$. However, constructing a code in this naive manner exponentially increases its computational overhead. In practice, there also exist efficient algorithms to reach these bounds.

1.1.1 Entropy Coding

The term entropy coding describes a lossless data compression method which approaches this lower bound of an expected code length equal to the entropy of the source data distribution. The two most common methods for entropy coding of data are Huffman codes and Arithmetic coding [6]. Huffman coding is a method for determining a binary string for each source data symbol, assigning shorter codes for the more frequently occurring symbols. Among the binary codes that encode a message symbol-by-symbol Huffman codes can be shown to be optimal. However, if a symbol's probability of occurring is not a negative power of two, Huffman codes may suffer an arbitrarily large overhead. Arithmetic coding is a compression method which assigns a single arbitrary precision fraction to encode an entire message. With arithmetic coding not being limited by the per-symbol restriction it is possible to encode messages drawn from an arbitrary source distribution with at most one bit of overhead per message. Its main downsides compared to Huffman coding are a larger computational overhead and the fact that partially decoding a message is nontrivial. Asymmetric Numeral System (ANS) [7] is a recently developed entropy coding method which combines the performance of Huffman codes with the near-optimal compression ratios achieved with arithmetic coding.

We can recap that, under the assumption of probabilistically generated data, optimal lossless compression codecs can at best compress information such that the message size matches the entropy of the data generating distribution. If one has access to the probability distribution of the data then there are well-known algorithms to implement lossless compression.

Here, a connection to generative modeling in machine learning may become quite obvious. In theory, since a generative model of the data gives us access to $P(\mathbf{x})$ we could simply feed it as the input to our entropy coding scheme of choice. However, there may be several issues with this approach. For one, machine learning models usually formally assume and approximate continuous data, requiring some form of quantization. Then, the data is very often drawn from an impractically high-dimensional sample space, making a naive tabulation of all "symbol" probabilities an effectively impossible task. Finally, simply evaluating $P(\mathbf{x})$ may involve an intractable integral for many types of generative models (such as GANs and VAEs). We will briefly mention some approaches to solving this in 1.3.

1.2 Lossy compression

By relaxing the requirement of perfect reconstruction of the input information, much greater compression ratios can be achieved. Lossy compression is particularly relevant in applications involving analog signals (or any other kind of continuous data), since real numbers generally are already represented with finite precision in digital computers, meaning that some amount of error is acceptable. Thus, digital media such as images, audio and video is the most common domain where lossy compression is applied. The bit-rate savings of lossy compression over lossless compression may be substantial, often by an order of magnitude, and with very little perceptible difference to end users.

The theoretical foundations of lossy compression, such as the trade-off between reconstruction error and compression rate may be formalized with Shannon's rate-distortion theory [5, 6]. In this setting we define an encoder function e which maps our sequence of source data points $\mathbf{x}_i \in \mathcal{X}$ to some finite intermediate representation $w_i \in \mathcal{W}$ and a decoder function d which maps the $\mathbf{w}_i$ to reconstruction points $\hat{\mathbf{x}}_i \in \mathcal{W}$. The symbols of $\mathcal{W}$ are transmitted losslessly by some entropy code c.

Next, we define a distortion function ρ which is an error measure between two data points in $\mathcal{X}$. A typical choice here is a squared error metric, i.e $\rho(\mathbf{x}, \hat{\mathbf{x}} =\| \mathbf{x} - \hat{\mathbf{x}}$. We also define the length of an encoded symbols $l(\mathbf{x}) = |c(e(\mathbf{x}))|$. In this framework we may define the objective of lossy compression. We want to simultaneously minimize the bitrate:

$$\mathcal{R} = \mathbb{E}_{\mathbf{x}\sim P}[l(\mathbf{x})] \tag{1.2}$$

And the average distortion of our code:

$$\mathcal{D} = \mathbb{E}_{\mathbf{x}\sim P}[\rho(\mathbf{x}, \hat{\mathbf{x}})], \hat{\mathbf{x}} = d(e(\mathbf{x}) \tag{1.3}$$

Again, it is information theory which allows us to establish some fundamental bounds on the trade-off between these two objectives. A useful viewpoint for this process is that our lossy compression algorithm acts as a noisy channel between a random input $\mathbf{X}$ and a noisy output $\hat{\mathbf{X}}$. Their relationship may be modeled with a posterior distribution $Q_{\hat{\mathbf{X}}|\mathbf{X}}$, such that the joint distribution can be written

$P_{data} \cdot Q_{\hat{\mathbf{X}}|\mathbf{X}}$. We define the mutual information $I(\mathbf{X}, \hat{\mathbf{X}})$ as the KL divergence between the joint distribution of $\mathbf{X}$ and $\hat{\mathbf{X}}$ and the product of the marginal distributions of the two variables. An interpretation of the mutual information is that it measures the amount of information that the reconstruction $\hat{\mathbf{X}}$ contains about the source data $\mathbf{X}$, or equivalently, the amount of information needed to transmit $\hat{\mathbf{X}}$, given $\mathbf{X}$. It is always non-negative and equals zero if and only if $\mathbf{X}$ and $\hat{\mathbf{X}}$ are independent.

The best-achievable bitrate at a given level of distortion D may now be characterized by the rate-distortion function:

$$\mathcal{R}_I(D) = \inf_{Q_{\hat{\mathbf{X}}|\mathbf{X}}} I(\mathbf{X}, \hat{\mathbf{X}}) \text{ subject to } \mathbb{E}[\rho(\mathbf{X}, \hat{\mathbf{X}})] \leq D \tag{1.4}$$

$\mathcal{R}_I(D)$ depends only on the source distribution and the choice of distortion measure. It is generally analytically intractable, though explicit solutions for some special cases do exist [6]. In practice, the design of a lossy compression algorithm (or lossy codec) is constrained by various practical considerations, such as the computational complexity, memory usage, or encoding/decoding latency. If we denote the set of all admissible lossy codecs under these practical constraints by $\mathcal{C}$, then we can define the operational rate-distortion function:

$$\mathcal{R}_O(D) = \inf_{c \in \mathcal{C}} \mathbb{E}[l(\mathbf{x})] \text{ subject to } \mathbb{E}[\rho(\mathbf{X}, \hat{\mathbf{X}})] \leq D \tag{1.5}$$

as the infimum of the expected code length over all codecs that satisfy the expected distortion constraint. Compared to the information R-D function $\mathcal{R}_I(D)$, which is defined in terms of an abstract information-theoretic quantity (mutual information), the operational R-D function $\mathcal{R}_O(D)$ is defined directly in terms of the code lengths achieved by actual compression algorithms. In general, we have $\mathcal{R}_O(D) \geq \mathcal{R}_I(D)$ for all D, i.e., the information R-D function provides a lower bound on the operational R-D function. Equality is not necessarily achievable, depending on the type of constraints that define $\mathcal{C}$.

1.3 Data Compression and Machine Learning

The application of machine learning methods to data compression is in a sense quite natural and some authors argue that two subjects essentially describe two sides of the same coin [8]. While statistical inference is the most common theoretical framework underpinning the design of and research into methods used for machine learning, one may also apply information theoretic tools to the problem of learning and is frequently able to justify the same approach from either point of view. As a simple example we may revisit the source coding theorem. A common setting in machine learning is that the data generating probability distribution

$P(\mathbf{X})$ is unknown and that one wishes to approximate it with a tractable distribution Q. If we base our code on the distribution $Q(\mathbf{x})$ instead of $P(\mathbf{x})$, then a new bound for the expected length of an optimal code may be derived [6] (we assume $D = 2$ for simplicity):

$$H(P) + D_{KL}(P \parallel Q) \leq L < H(P) + D_{KL}(P \parallel Q) + 1 \tag{1.6}$$

Where D_{KL} is the Kullback-Leibler divergence. We notice that this bound is equal to the cross-entropy:

$$H[P, Q] = \mathbb{E}_{\mathbf{x} \sim P}(-\log_2 Q(x)) = H(P) + D_{KL}(P \parallel Q) \tag{1.7}$$

Since optimizing the cross-entropy loss of Q is equivalent to maximum-likelihood estimation, one may argue that it is therefore justified from a compression perspective (or vice versa).

The connections between machine learning and data compression extend beyond this simple example. Bayesian inference, a foundational framework in machine learning, can be shown to be equivalent to the minimum description length (MDL) principle under certain conditions [8]. MDL provides an intuitive and theoretically grounded approach to model selection, favoring models that provide the most compact encoding of the data while considering the trade-off between model complexity and goodness of fit. This aligns well with the goals of data compression, which seeks to minimize the number of bits needed to represent information.

Variational inference, another key technique in machine learning, also has close ties to data compression through the bits-back coding scheme [9]. Bits-back coding allows for lossless compression of data using latent variable models by transmitting a stochastic latent code z drawn from a distribution $q(z|x)$ using close to the theoretical minimum number of bits given by the KL divergence between $q(z|x)$ and a prior $p(z)$. The encoding process works as follows:

1. The sender first generates a sample z from $q(z|x)$ by *decoding* a random bit-string ξ under the probability model $q(z|x)$

2. The sender then encodes the data x under the likelihood model $p(x|z)$, encodes z under the prior $p(z)$, and transmits the resulting bits.

3. The receiver decodes z and x from the received bits and may then recover the random bits ξ by encoding z under $q(z|x)$

These extra bits recovered by the receiver are "bits-back", allowing the sender to transmit ξ "for free". Crucially, ξ does not have to be random - it can be well-compressed auxiliary information that also needs to be transmitted. The expected coding cost of bits-back coding is equal to the negative ELBO. Bits-back coding thus provides a compression interpretation of variational inference - minimizing the KL divergence between $q(z|x)$ and the true posterior $p(z|x)$ is equivalent to minimizing the code length.

So far we have assumed that our latent variable model operates with discrete latents z, however many widely used latent variable models in machine learning, including VAEs, assume a continuous latent space. However, by appropriately quantizing the latent space of a VAE, such as maximum entropy quantization [10], it can be shown that bits-back coding can be implemented with arbitrary precision [8]. Another issue with the bits-back coding approach is the "initial-bits problem", where the sender needs to determine auxiliary bits ξ that are practically useful to the receiver. However, if multiple messages are to be sent, then the bitstream of previously coded messages can be used for those auxiliary bits. This requires messages to be decoded in a stack-like fashion, which naturally pairs this method with ANS (see section 1.1.1), which also decodes symbols on a last-in-first-out basis. This method, bits-back coding with chaining and ANS, has been demonstrated as a practical lossless compression scheme in [10].

1.4 Learned Lossy Compression

Unlike with lossless compression, where the loss can simply be a differentiable objective that measures the bitrate, the loss terms for training a model that performs lossy compression will now be required to capture the rate-distortion trade-off. The analysis transform e, synthesis transform d, and entropy model p are replaced with neural networks or other learned universal function approximators. The transform coefficients then have to be quantized by some method to obtain a discrete representation, which allows for entropy coded transmission. We continue with the notation from section 1.2. To optimize the operational R-D trade-off, a common approach is to minimize the rate-distortion Lagrangian [11]:

$$L(\lambda, c) = \mathcal{R}(D) + \lambda\mathcal{D}(D) = \mathbb{E}[l(\mathbf{x})] + \lambda\mathbb{E}[\rho(\mathbf{x}, \hat{\mathbf{x}})] \tag{1.8}$$

Lossy neural compression, also known as learned lossy compression, is an emerging field that applies neural networks and other machine learning techniques to the task of lossy data compression. It builds upon the classical paradigm of transform coding, replacing hand-designed linear transforms with learned non-linear functions to better adapt to the statistics of the data being compressed.

The core idea behind transform coding is to divide the task of lossy compression into two steps: decorrelation and quantization. First, the sender applies an analysis transform to the input data, resulting in transform coefficients that are ideally decorrelated. These coefficients are then quantized, typically using scalar quantization, to obtain a discrete representation. The quantized coefficients are entropy coded into a bitstream, which is transmitted to the receiver. The receiver decodes the bitstream, dequantizes the coefficients, and applies a synthesis transform (often the inverse of the analysis transform) to reconstruct an approximation of the original data.

In neural lossy compression, the analysis transform, synthesis transform, and entropy model are implemented using neural networks or other learned function approximators. These components are learned end-to-end from data, optimizing a rate-distortion loss that balances the competing objectives of minimizing the bitrate and the distortion between the original and reconstructed data. A useful perspective on neural lossy compression is to view it as a form of variational autoencoder (VAE) with a discrete latent space. In particular, a deterministic autoencoder with uniform quantization of the latents can be seen as a VAE with a degenerate approximate posterior distribution. This connection provides insights into the design and optimization of neural lossy compression models.

A key challenge in neural lossy compression is that the quantization operation, which maps the continuous-valued transform coefficients to a discrete set of symbols, is not differentiable. This prevents the model from being trained end-to-end using gradient descent to minimize the rate-distortion loss. Various techniques have been developed to address this issue.

A popular approach is uniform quantization [9], where each element of the transform coefficients is rounded to the nearest integer. The rounding operation can be approximated using the straight-through estimator, which passes the gradient through the quantizer unchanged during backpropagation. Alternatively, uniform noise can be added to the coefficients during training, which can be seen as a differentiable approximation of uniform quantization.

Stochastic annealing methods have also been proposed, where the quantization operation is replaced by a stochastic rounding process. The probabilities of rounding up or down are determined by the distances between the input and the quantization levels, and are annealed towards deterministic rounding over the course of training.

Vector quantization (VQ) is a classical lossy compression technique that is the most basic and general form of a lossy codec. A VQ scheme consists of:

- A set of integer representations $\mathcal{W} = 1, 2, ..., k$
- An encoder e that assigns input data to an integer representation $w_i \in \mathcal{W}$
- A decoder d that returns a reconstruction given w, usually by indexing into a codebook of k reconstruction vectors

The goal is to determine an optimal quantization scheme for a given data distribution to minimize a reconstruction error (as in k-means) or more generally an operational rate-distortion trade-off. An optimal quantizer always encodes data to its nearest neighbor in the codebook to minimize the reconstruction error ρ. The codebook entries are set to minimize expected reconstruction error between the entry and data points assigned to that index.

For some data sources like uniform or Laplace distributions, optimal quantization can be characterized analytically. In most applications, Lloyd-Max style algorithms (k-means) are used to estimate a quantization scheme from data samples by minimizing an empirical rate-distortion cost over a dataset. Given unlimited data and compute, VQ can approximate any data distribution arbitrarily well. The computational and storage demands of VQ increase quickly with data dimensionality. In high dimensions, a very large number of quantization points and training data are needed to approximate the data distribution well.

Vector quantization has been successfully combined with variational autoencoders (VAEs) in a model called VQ-VAE [12]. In a VQ-VAE, the encoder maps the input to a discrete latent representation by finding the nearest vector in a learned codebook, similar to classical VQ. The decoder then reconstructs the input from this discrete representation. The codebook is learned jointly with the encoder and decoder to minimize the reconstruction error and a commitment loss that encourages the encoder to use the codebook vectors.

A major issue that has not been addressed so far is that typical distortion measures used in lossy compression, such as mean squared error (MSE), do not align well with human perception. These measures penalize any deviation from the original signal equally, regardless of whether those deviations are perceptually noticeable or not. As a result, models trained to minimize MSE may produce blurry or over-smoothed reconstructions that lack perceptually important details. It is important to recognize that neural networks will perform only as well as the losses they are trained with. If the loss function does not capture perceptual quality, the model will not be optimized for it. This highlights the need for perceptually-informed loss functions in neural lossy compression.

Adversarial losses have emerged as a promising approach to address this issue. In an adversarial framework, a discriminator network is trained to distinguish between original and reconstructed images, while the compression model (generator) is trained to fool the discriminator. This encourages the model to generate reconstructions that are perceptually similar to the original images, even if they may perform worse in terms of simpler distortion metrics. Such adversarial losses have become a very common building blocks in many publications concerned with neural compression. Blau et al. [13] suggest that the objective for a lossy compression scheme (as in equation 1.4) should not just include a constraint on the expected level of distortion, but also a constraint on the divergence between the data generating distribution and the distribution of the reconstructed symbols:

$$D_{KL}(P(\mathbf{X}) \parallel P(\hat{\mathbf{X}})) \leq \sigma \tag{1.9}$$

Chapter 2

Related Work

As we have demonstrated, there are many theoretical connections between data compression and various applications and advances in ML research. We will here give a short overview of the literature on neural data compression.

2.1 Image compression models

Image compression using neural networks has been explored since the late 1980s and 1990s, but these early approaches differed markedly from modern methods in their scale, architectures, and encoding schemes [14]. No doubt limited by the computing power available at the time, these early approaches typically used small, fully-connected neural networks with a single hidden layer to compress image blocks or to learn the parameters of traditional codecs, rather than end-to-end optimization of the entire compression pipeline using deep convolutional architectures as is common today.

In 2016, deep generative models began to be explored for data compression [15]. Although the authors did not implement their architecture in a practical image compression scheme, they demonstrated the potential of using variational autoencoders (VAEs) for lossy compression. Many of the results presented in section 1.3, such as the connections between variational inference and both lossless and lossy compression, had been identified earlier. However, the work by Gregor et al. [15] sparked renewed interest in the application of deep learning techniques to image compression.

In addition to the core components of the compression pipeline, auxiliary losses based on adversarial training or deep feature matching have been incorporated to optimize for perceptual quality in addition to rate-distortion performance. Adversarial losses, as used in generative adversarial networks (GANs), encourage the decoder to generate reconstructions that are perceptually similar to the original images. Deep feature matching losses, such as those based on the activations of pretrained convolutional neural networks, aim to preserve the perceptual similarity between the original and reconstructed images in a learned feature space. The combination of these advances in neural architectures, entropy models, and perceptual optimization techniques has led to significant improvements in the performance of neural image compression methods, surpassing traditional codecs in terms of rate-distortion trade-offs and perceptual quality

2.1.1 HiFiC and ILLM

The High Fidelity Compression (HiFiC) model, proposed by Mentzer et al. [16], presents a generative adversarial network (GAN) based approach for lossy image compression. The primary goal of HiFiC is to achieve visually pleasing reconstructions that closely match the input images perceptually, while operating over a wide range of bitrates. This is accomplished by combining an autoencoder-based learned compression model with adversarial losses, which optimize for perceptual quality in addition to traditional distortion losses.

HiFiC utilizes a 2D-convolutional variational autoencoder (VAE) architecture for the compression model. VAEs are well-suited for learning compact and meaningful representations of images, making them a natural choice for image compression tasks. The conditional discriminator plays a crucial role in assessing the perceptual quality of the reconstructed images and providing feedback to the generator (i.e., the decoder of the autoencoder) to improve the visual fidelity of the output. The learned entropy model, based on a hyperprior and autoregressive context modeling, can be used to efficiently compress the quantized latents, enabling effective compression at low bitrates.

To enable end-to-end optimization of the compression model, HiFiC employs a uniform noise-based relaxation of the quantization operation during training. Specifically, uniform noise is added to the latent representation before quantization, allowing for differentiation of the quantization step. During inference, the latent representation is quantized using rounding, ensuring that the model operates with discrete latents for efficient compression.

The authors conduct extensive experiments to evaluate the model on high-resolution images from several datasets. They evaluate their model on a range of perceptual metrics and in a user study confirmed that HiFiC reconstructions are preferred over those of other popular image codecs, even when they were used at a higher bitrate. This highlights the effectiveness of the GAN-based approach in achieving high perceptual quality at low bitrates. HiFiC does not implement the actual entropy coding step in the paper, however it would be a simple addition to do so and suffices to demonstrate the feasibility of the overall approach.

The Implicit Local Likelihood Model (ILLM), proposed by Muckley et al. [17], was chosen as our only implemented model due to the high quality of its codebase and its competitive performance in image compression. ILLM builds upon the success of HiFiC while introducing a novel approach to improve the statistical fidelity of the reconstructed images. One of the key architectural differences between ILLM and HiFiC is the use of a separate model to obtain a "label" map over the image. This label map is generated by a pretrained vector quantized variational autoencoder (VQ-VAE), which partitions the image into local regions based on their similarity in the latent space.

The label map generated by the VQ-VAE is then used to condition the discriminator, providing it with more localized information to distinguish between real and generated image patches. This local conditioning allows the discriminator to focus on the perceptual quality of specific regions rather than the entire image, enabling it to capture the local statistics more effectively. By incorporating this information, ILLM aims to improve the statistical fidelity of the reconstructed images, ensuring that they not only look perceptually similar to the input images but also maintain the local structure and texture.

The use of a separate label map in ILLM also allows for more flexibility in the design of the discriminator architecture. The authors employ a U-Net-based discriminator, which is well-suited for capturing multi-scale information. The U-Net architecture consists of an encoder that downsamples the input and captures high-level features, and a decoder that upsamples the features and produces the final output. Skip connections between the encoder and decoder allow for the preservation of spatial information at different scales. By taking advantage of the multi-scale information provided by the label map, the U-Net discriminator can effectively capture both local and global characteristics of the image, enhancing its ability to assess the perceptual quality of the reconstructions accurately.

2.2 Audio Compression Models

As with other tasks increasingly accomplished using machine learning techniques, such as classification or text-conditional generation, research progress in using deep learning for audio compression lags behind that in the image domain. Though there is some earlier research on neural approaches to speech compression [18, 19], the first publication to tackle compression of general audio data using a modern ML-paradigm was published in 2021 [20]. Since state-of-the-art hand-designed audio codecs are close to transparent at bitrates above 64kb/s, the research into neural codecs mainly focuses on achieving satisfactory quality with data rates of around 16kb/s and below, where traditional codecs struggle to maintain an acceptable level of quality.

The first major breakthrough in deep learning-based audio compression was marked by the the 2021 publication of the SoundStream codec [21]. SoundStream demonstrated that a fully neural architecture, trained end-to-end, could outperform conventional codecs like Opus and EVS at low bitrates. SoundStream uses a vector-quantized VAE architecture and couples it with an STFT-based discriminator. The adoption of the residual vector quantizer together with the use of

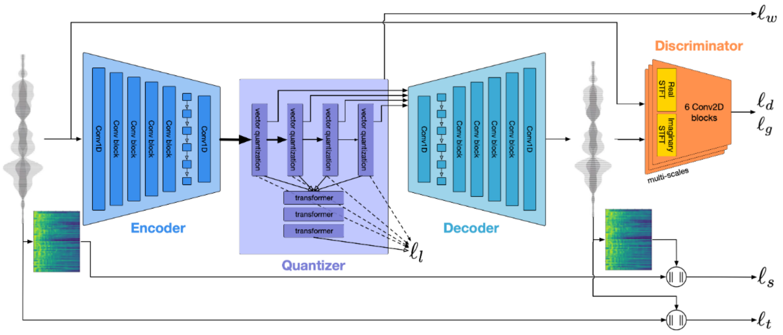

Figure 2.1: Block diagram of the EnCodec model architecture and its losses, from [23]

structured dropout within the quantization layer during training has the advantage of allowing a single model to operate at different bitrates.

The success of SoundStream opened up new avenues for research into deep learning-based audio compression. Subsequent work built upon the core ideas of end-to-end optimization, generative modeling, and learnable quantization to further push the boundaries of compression efficiency and perceptual quality. Innovations such as improved architectures, perceptually-driven loss functions, and joint optimization with enhancement and bandwidth extension tasks aimed to create neural codecs that could adapt to diverse audio content and challenging acoustic conditions. The field also explored techniques for reducing computational complexity and memory footprint, to enable practical deployment of neural codecs on resource-constrained devices. As deep learning-based approaches continue to evolve, they hold the potential to redefine the state-of-the-art in audio compression, offering high coding efficiency, low latency, and perceptually-optimized quality across a wide range of bitrates and audio content types.

VQ-VAEs have become the dominant paradigm in neural audio compression due to their ability to learn a compact, discrete latent representation that enables efficient compression and powerful generative modeling. The flexible architecture, which allows for domain-specific designs and scalability to large datasets, has led to steady improvements in reconstruction quality. The introduction of residual vector quantization has further enhanced VQ-VAEs by enabling variable bitrate compression and improving the expressiveness of the learned codes, making them a versatile tool for a wide range of downstream audio applications. For instance, the Soundstream codec has been leveraged in [22] for text-conditional music generation.

2.2.1 EnCodec

EnCodec [23] builds upon the residual vector quantized (RVQ) autoencoder architecture introduced in SoundStream, but incorporates several key innovations that lead to improved performance and capabilities. One significant difference is the use of a simplified yet effective multi-scale STFT discriminator for the adversarial loss. The EnCodec authors demonstrate that this single discriminator is sufficient to generate high-quality audio while streamlining the training process, in contrast to the combination of discriminators employed by SoundStream. Another notable contribution of EnCodec is the introduction of a novel loss balancer mechanism that stabilizes training by ensuring each loss term contributes a specified fraction of the overall gradient. This approach decouples the loss weights from their typical scales, enhancing the interpretability of these weights. Furthermore, EnCodec explores the use of a lightweight Transformer model to further compress the quantized latent representation by up to 40% while maintaining faster-than-real-time performance, a capability not present in SoundStream.

The EnCodec model is evaluated on a wider range of scenarios compared to SoundStream, including 48 kHz stereo music compression. The paper also presents an extensive subjective evaluation using MUSHRA listening tests, showcasing EnCodec's superior performance against SoundStream and other baselines across speech, noisy speech, and music at bitrates from 1.5 to 12 kbps for 24 kHz audio. Moreover, EnCodec demonstrates the ability to jointly perform compression and background noise suppression without introducing additional latency by conditioning the encoder or decoder, a feature not explored in SoundStream. The model architecture also incorporates an LSTM layer for sequence modeling of the latent representation, enabling the capture of temporal dependencies in the audio signal.

The key advantages of EnCodec, as claimed by the authors, include the simpler yet effective discriminator, the novel loss balancing technique, the Transformer-based entropy coding, the evaluation at higher bandwidths, and the strong subjective evaluation results across a range of audio types, bitrates, and sample rates. The evidence provided suggests that these innovations contribute to EnCodec's superior performance compared to the SoundStream baseline.

Looking ahead, the quantized tokens learned by EnCodec have already been leveraged for downstream audio generation tasks. The AudioGen [24] model, for instance, generates audio conditioned on the discrete EnCodec tokens. This demonstrates the potential of using EnCodec's learned representations for generative audio modeling and opens up exciting possibilities for future research and applications in this domain.

2.2.2 Descript Audio Codec

Kumar et al. [25] again use a largely similar architecture to Encodec and SoundStream. It is designed as a drop-in replacement for applications that have been designed to be used with Encodec. The authors identify a number of issues with the Encodec model and present several incremental improvements to their model based on that.

A simple but significant change introduced with the Descript model is the use of the oscillatory Snake activation function which adds a periodic inductive bias to the generator improving. The LSTM sequence modeling layers and the loss balancer mechanism that were used in the Encodec model were not included.

One major issue that the authors identified with Encodec is a suboptimal codebook utilization. Ideally, on a diverse set of testing samples, the distribution of codebook vectors being used is uniform, which maximizes its entropy. However, during training of VQ-VAEs it is common that some codebook vectors end up going unused, meaning that they also do not receive any updates from training. Both Soundstream and Encodec therefore periodically reinitialize underused codebook vectors. However, this procedure does not seem to fully solve the problem. The Descript model instead adopts the use of factorized codes together with L2-normalization from [26]. With this procedure, a learned linear projection maps from the embedding space of the autoencoder into a lower dimensional space where the codebook lookup is performed, which then gets mapped back to a high-dimensional latent vector. Additionally, the L2-normalization step maps the latent variables onto a sphere in the embedding space. Both of these improvements enhance training stability and allow codebook learning with the loss-based objective from [12] without the need for reinitialization and still achieving a near-optimal codebook utilization.

Chapter 3

Methods

Our approach to audio compression can be summarized as "compressing audio using images." The pipeline consists of the following steps:

1. Use a short-time Fourier transform (STFT) to convert an audio file into a two-dimensional complex spectrogram representation.
2. Convert the complex spectrogram into a log-magnitude spectrogram.
3. Map the log-magnitude spectrogram to a grayscale image space.
4. Pass the grayscale image to an image compression model (in our case, ILLM [17]).
5. Decompress the compressed image.
6. Transform the decompressed image back into a log-magnitude spectrogram.
7. Invert the log-magnitude spectrogram back into a waveform using phase reconstruction techniques.

While this approach addresses some of the challenges in modeling audio compared to images, such as the one-dimensional nature of audio signals and their different frequency characteristics, it also introduces new challenges. One significant challenge is the reconstruction of the phase information from the magnitude spectrogram. Modeling the phase of the STFT of a signal is difficult, as it can fluctuate rapidly, and in areas of the time-frequency spectrum where the signal magnitude is low, the phase essentially appears like noise.

Despite these challenges, using spectrograms as input representations for audio tasks has shown promise in various applications [27, 28]. This suggests that a spectrogram-based approach for audio compression could be a promising direction to explore.

Another potential advantage of working in the spectrogram domain is faster training, since the model has direct access to the frequency components of the audio signal. Similar to how interpretability research in convolutional image models

[29] has revealed that image models learn to extract higher-level features based on simpler frequency-specific filters, we expect that audio models also learn low-level filters that can extract different parts of the signal spectrum. By providing the model with a spectrogram representation, we can potentially simplify this learning process and allow the model to focus on higher-level features that are more relevant for compression.

Furthermore, our spectrogram-based approach may enable the compression model to operate more efficiently at a given computational budget. This is particularly important considering that the storage consumption and computational overhead of deep neural networks is one of the main factors hindering the adoption of these technologies on consumer hardware, especially in the context of compression, where excellent compression rates do not matter if the model requires many times the amount of storage that would be saved by the data it compresses. Finally, our spectrogram-based approach provides the attractive prospect of adapting existing image compression models and applying some finetuning on spectrogram images. By representing audio as spectrograms, we can potentially leverage these existing architectures and adapt them for audio compression, benefiting from the progress made in the image domain.

In this work, we chose to focus on music and general audio rather than speech-specific compression. Music and general audio present additional challenges compared to speech, such as a wider frequency range and more complex temporal structures. Music often contains multiple instruments playing simultaneously, resulting in a rich and diverse frequency spectrum. Additionally, music and general audio can have more complex temporal dynamics, with variations in rhythm, tempo, and structure over time. Developing effective compression methods for these types of audio is important given the prevalence of music and other non-speech audio in many applications, such as streaming services, audio archives, and multimedia content. By focusing on music and general audio, we aim to address the unique challenges and requirements of compressing these types of signals efficiently while maintaining perceptual quality.

3.1 Phaseless Spectrogram Inversion

While there are many advantages to spectrogram-based approaches for audio-related tasks, it introduces a major challenge for any application where a waveform output is desired. Since it is much easier to train models that generate magnitude spectrograms, their inversion back into a waveform representation require a method for estimating the phase from the magnitude alone. However, all currently known methods for phase reconstruction are non-differentiable which limits there applicability for fully end-to-end trained architectures. We reconstruct the phase from magnitude spectrograms using the Tifresi [27] library's implementation of the Phase-gradient heap integration (PGHI) algorithm [30].

Despite being a computationally more efficient method than the traditionally employed Griffin-Lim algorithm (GLA) [31], which relies on an expensive iterative scheme, it has been shown to produce better results. It is based on the observation that, under the condition that the STFT is computed with a Gaussian window, there exists an explicit analytical relationship between the gradient of the phase and that of the magnitude, which can be discretely approximated as [30, 27]:

$$\nabla\phi[m,n] = \left[\frac{aM}{\lambda}\partial_m\mathbf{M}[m,n], -\frac{\lambda}{aM}\partial_n\mathbf{M}[m,n] - 2\pi na/M\right] \tag{3.1}$$

Where, a, M and λ are parameters of the discrete STFT and the window function and $\mathbf{M}$ is the logarithm of the magnitude spectrogram. Therefore, the phase can be recovered by integrating equation 3.1 numerically, though additional care has to be taken to first integrate along contours of large magnitude (where the error is lower) and to completely avoid visiting areas where the STFT-magnitude is too small.

3.2 ILLM Finetuning

To adapt the ILLM image compression model for audio compression, we finetuned the pretrained model on a dataset of spectrogram images. The spectrograms were generated on-the-fly from the musdb18-hq and FMA datasets using the short-time Fourier transform (STFT) with a window size of 512 samples and a hop length of 64 samples. The resulting complex spectrograms were converted to log-magnitude spectrograms and then mapped to a grayscale image space with floating-point pixel values in the interval $[0.0, 1.0]$. During the finetuning process, the ILLM model was trained to compress and reconstruct the spectrogram images using the same architecture as in the original implementation.

With the goal of improving the perceptual quality of the reconstructed audio, we incorporated a novel loss term based on the mel spectrogram representation. The mel spectrogram is a perceptually-motivated representation of audio that better aligns with human auditory perception compared to the linear frequency scale used in the standard STFT. We used the implementation of the mel spectrogram loss from the Descript audio tools library also used in [25]. This loss function computes the mean squared error (MSE) between the mel spectrograms of the original and reconstructed audio signals. By minimizing the difference in the mel spectrogram domain, the model is encouraged to preserve perceptually relevant features of the audio during compression. The mel spectrogram loss was added to the existing loss terms used in the ILLM model. By incorporating the mel spectrogram loss and finetuning the ILLM model on a diverse dataset of music and general audio spectrograms, we aim to adapt the image compression model to the specific characteristics of audio data and improve the perceptual quality of the reconstructed audio signals.

During training we also track the consistency of the generated spectrograms. In the context of generative modeling, a spectrogram can be considered "inconsistent" if it does not correspond to a valid STFT representation of a time-domain signal. Inconsistent STFT representations can lead to artifacts and distortions in the synthesized time-domain signal. When the generated STFT deviates significantly from a valid STFT, the resulting audio quality may be degraded, containing undesirable noise or aliasing components. Typically consistency is assessed by measuring the projection error between the source spectrogram and the spectrogram of the resulting waveform after phase reconstruction and inverse STFT. We use the consistency metric proposed in [27] measures the consistency of a generated magnitude spectrogram with the properties of a valid short-time Fourier transform (STFT) magnitude. By comparing how well the partial derivatives of the generated magnitude match this expected relationship, the metric provides a computationally efficient way to assess the consistency of the generated magnitude spectrogram without having to perform a potentially demanding transform operation.

3.3 Datasets and Metrics

The datasets used for training and evaluation in this study were the musdb18-hq [32] dataset and the Free Music Archive (FMA) [33] dataset. The musdb18-hq dataset is a widely used benchmark for music source separation tasks, containing 150 full-length music tracks along with their isolated vocals, drums, bass, and other stems. Though our model was not designed for source separation, it is a dataset with full-band (i.e. containing frequencies up to 22.05kHz) uncompressed high quality audio files. The FMA dataset, on the other hand, is a large-scale dataset containing 106,574 compressed mp3 tracks. We adopted the balanced sampling approach from [25] to using FMA during model training, however for the evaluation of our method and the baselines we chose to only use musdb18-hq to avoid the presence of unrelated compression artifacts.

While there is a wealth of objective metrics for evaluating the quality of speech-only codecs and models (PESQ [34] to name one example), this is not the case for applications involving music and general audio. The "gold standard" for evaluation of the perceptual quality of audio compression algorithms and audio generation system remains the subjective evaluation of the audio samples by humans in controlled listening tests. However automated metrics are still very useful in providing early feedback and to establish objective baselines:

VisQOL [35] is an objective metric intended to reproduce a result on the same scale as in human trials that are using the mean opinion score scale (where 1 is the worst rating and 5 is a perfect score). To compute a score, VisQOL receives both an evaluation and a reference sample as input, computes a frequency-weighted magnitude spectrogram on both of the samples, extracts time-aligned patches of those spectrogram images and then computes the structural similarity index measure (SSIM, see below) to get a numerical score of their difference. The SSIM scores of the individual patches are then aggregated and mapped onto the 1-to-5 numerical scale by using a regression model that was trained to match human evaluation scores. Fréchet audio distance (FAD) [36] is an adaption of the Fréchet inception distance (FID) image metric to the audio domain. It is a reference-free metric intended for estimating the perceptual quality and statistical fidelity of the samples produced by a generative audio model. This is achieved by feeding both a reference dataset and the evaluation samples into an audio embeddings model (in our case, CLAP[37]). The difference in the statistical distribution of the latent activations between the two datasets is then compared with the Fréchet distance. Scale-invariant signal-to-distortion ratio (SI-SDR) [38] is an L_2-error metric similar to SNR, but which has been designed such that multiplicative scaling factors on the estimated signal do not affect the result. Though we did not expect our method to perform well in terms of this metric we have included it for the sake of completeness and to verify our results against [25].

In addition to metrics that are calculated on audio waveforms, our setup also gives us the opportunity to evaluate the metrics that are used in the evaluation of image compression methods. Two popular methods for measuring the difference in visual fidelity between two images are peak signal-to-noise-ration (PSNR) and the Multiscale Structural Similarity Index (MS-SSIM) [39].

The baselines that we chose to compare our method against are the encodec and descript neural audio codecs and the "PGHI only" method which subjects the input signal to the magnitude spectrogram computation and its inversion operation with tifresi without any compression operation on the spectrogram images themselves. All metrics were evaluated on 500 audio snippets of 6 seconds duration each. To establish some results for human perceptual quality we also conducted a small human evaluation study, with three subjects giving an opinion score of blinded samples relative to a known uncompressed reference sample. Each sample was rated on a scale of 1-to-5 which was then aggregated into a mean-opinion-score (MOS) [40].

Chapter 4
Results

A comparison between a sample spectrogram before and after our compression pipeline has been applied to it can be seen in Figures 4.1 and 4.2. The compressed spectrogram maintains much of the overall structure and high-frequency details of the original, but there is a noticeable loss of fidelity, especially in the low-frequency components.

While we have not conducted an extensive user study for validating our method, subjective listening tests indicate that it qualitatively performs considerably worse than our baselines, as shown in Figure 4.3. The mean opinion scores from a small group of listeners reveal that the audio quality of our method is rated significantly lower than the Encodec and Descript neural audio codecs. Despite these limitations in perceptual quality, our method is able to achieve competitive bitrates compared to the baselines, even though the spectrogram representation has increased redundancy compared to the raw audio waveform. Table 4.1 shows that our approach, both with the pretrained and finetuned ILLM model, achieves a bitrate of 13.1 kbps, which is comparable to Encodec at 12.0 kbps and Descript at 8.0 kbps. This demonstrates the potential for spectrogram-based methods to attain high compression ratios. While the 13.1 kbps bitrate of our method is higher when compared with our baseline codecs, it is important to note that these are still relatively low bitrates for audio compression. For reference, MP3 bitrates typically range from 128 kbps to 320 kbps, and even low-bitrate codecs like Opus operate at around 32 kbps for high-quality speech compression. The fact that our method can achieve audible reconstructions at such low bitrates is a promising sign, even if the perceptual quality is not yet on par with state-of-the-art codecs.

We notice that our spectrogram-based approaches consistently resulted in very low values for the SI-SDR metric. This is to be expected, as SI-SDR captures an L_2-distance between two signals and heavily penalizes significant phase differences. However, since the phase reconstruction process is imperfect, avoiding such phase mismatches is very difficult. Interestingly, although our method does not perform well in subjective evaluation experiments, it achieves very good scores with the VisQOL metric, outperforming both Encodec and the Descript

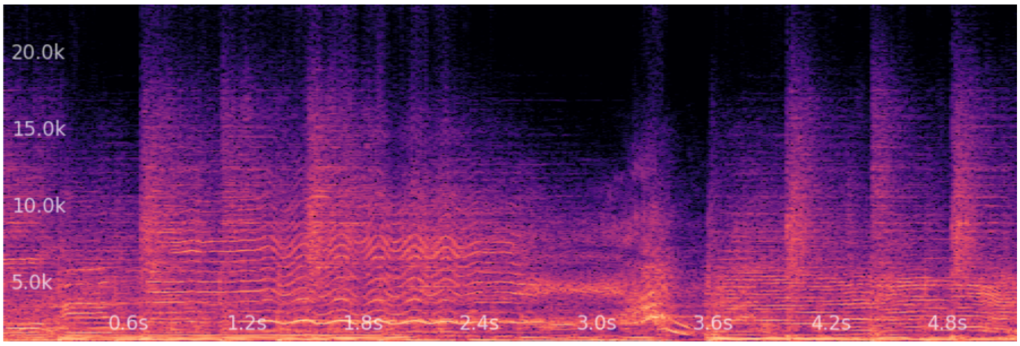

Figure 4.1: Ground-truth spectrogram of audio sample

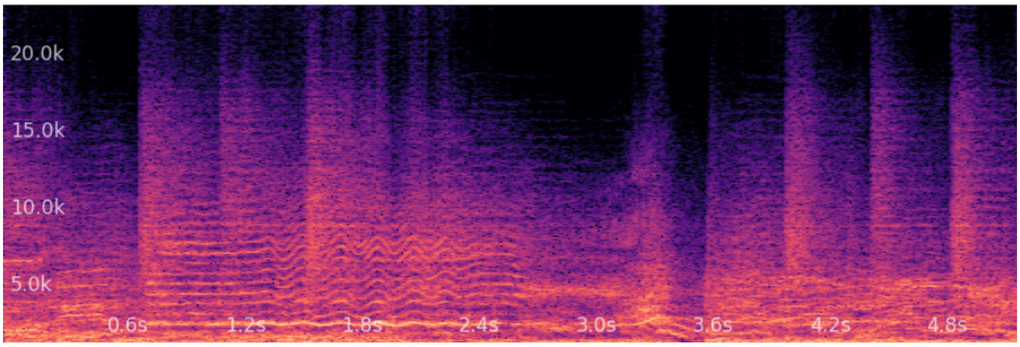

Figure 4.2: Spectrogram of audio sample after compression and phase reconstruction

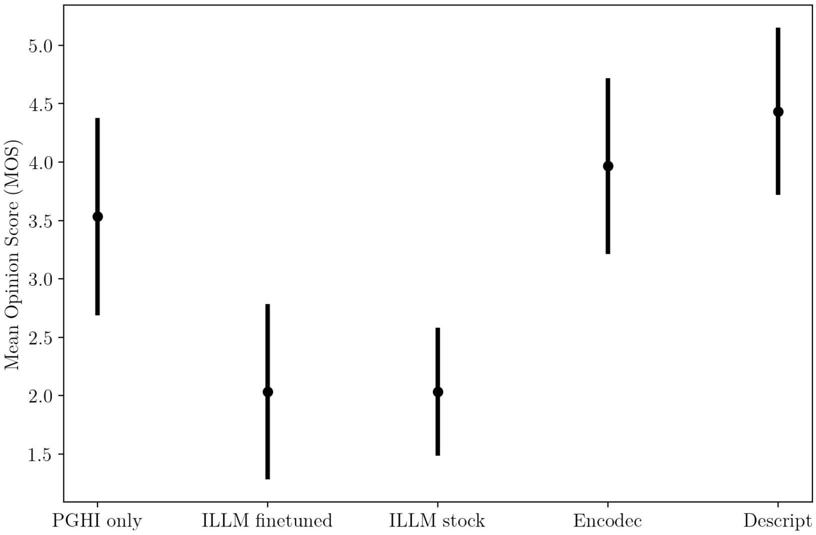

Figure 4.3: Mean opinion score during subjective evaluation test

Method	**Visqol**	**FAD**	**SI-SDR**	**Bitrate (kbps)**
PGHI only	4.68	0.025	-26.12	6444.0
Our Method (finetuned)	4.39	0.32	-35.89	13.1
Our Method (pretrained)	4.39	0.28	-37.34	13.1
Encodec	4.21	0.035	10.23	12.0
Descript Codec	4.16	0.026	10.23	8.0

Table 4.1: Objective Audio Metric Scores, ILLM

codec as seen in Table 4.1. Upon further investigation, it becomes clear that VisQOL is not a reliable indicator of perceptual quality for the spectrogram-based approaches. We hypothesize this is due to the design of VisQOL, which computes the SSIM on spectrogram patches to estimate quality. This makes it unable to detect and penalize the phase distortions that harm perceptual quality in our method. Qualitatively, this was also observed when evaluating the performance of the PGHI reconstruction with very low redundancy. Even when phase distortion artifacts were clearly audible, the VisQOL scores continued to be near-perfect.

In contrast, the FAD metric, which measures the Fréchet distance between embeddings from a pretrained audio classifier, correlates much better with human perception in our experiments. Table 4.1 shows FAD scores that align with the subjective evaluation results, with our method having a much higher (worse) score than the baselines.

During the finetuning process of the ILLM model on spectrograms, we tracked the consistency metric of the generated spectrograms (see section 3.2). Figure 4.4 plots this metric over the course of training, revealing that while consistency rapidly improves initially, the rate of improvement slows down and stabilizes at a value of about 0.65. This suggests there is still room for improvement in adapting image compression models to better suit the structure of audio spectrograms. While the finetuned model shows improved consistency during training (Figure 4.4), the overall performance in terms of audio quality metrics (Table 4.1) is quite similar between the two models. This suggests that the pretraining on image data provides a strong initialization for the spectrogram compression task, and that further finetuning on audio spectrograms may not always be necessary to achieve competitive results.

Figure 4.5 plots the MS-SSIM and PSNR metrics on the spectrogram images over the course of training the ILLM model. Interestingly, both metrics actually worsen as training progresses. However, this does not necessarily indicate a degradation in perceptual quality. As discussed in [13] and [16], these metrics do not always align well with human perception of image quality. In partic-

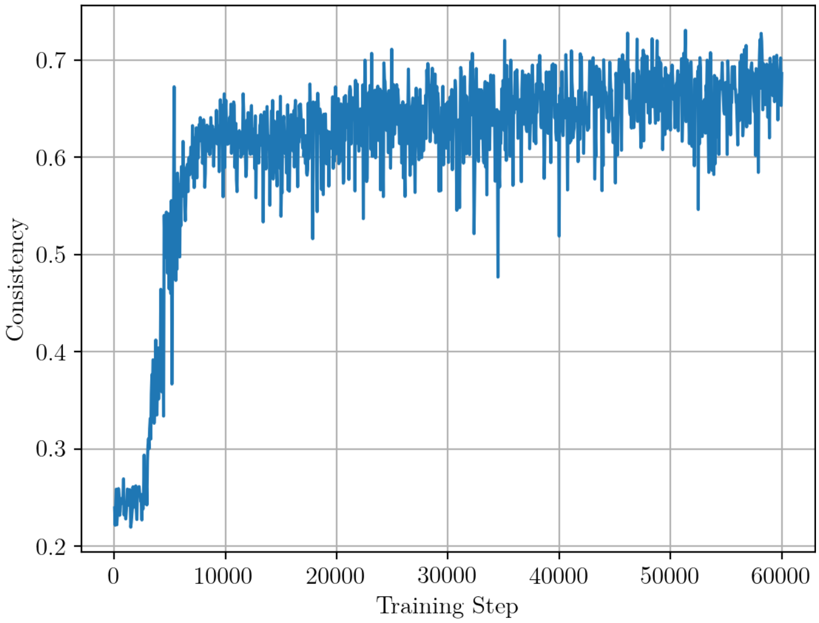

Figure 4.4: Consistency metric of generated data samples during training

ular, optimizing for PSNR tends to introduce blurriness, as the metric favors smoothed-out reconstructions that minimize pixel-wise error. In the context of audio spectrograms, a similar phenomenon may be occurring. The model may be learning to produce spectrogram reconstructions that appear less blurry, thereby risking a greater squared error, but that actually better capture the perceptually relevant features. The worsening scores observed in Figure 4.5 suggest that the adversarial training is pushing the model towards perceptually meaningful reconstructions, even at the cost of pixel-wise fidelity.

Finally, an interesting discrepancy we observed between our results and those presented in [25] is that Encodec and the Descript codec achieve very similar objective performance scores in our tests, in contrast to the clear performance advantage for the Descript codec reported by the authors. We were able to reproduce scores closer to those in [25] when using the 24 kHz model of the Encodec system, as shown in Table A.1. However, we believe that the scores for Encodec at 48 kHz are a fairer representation of its audio quality and compression performance.

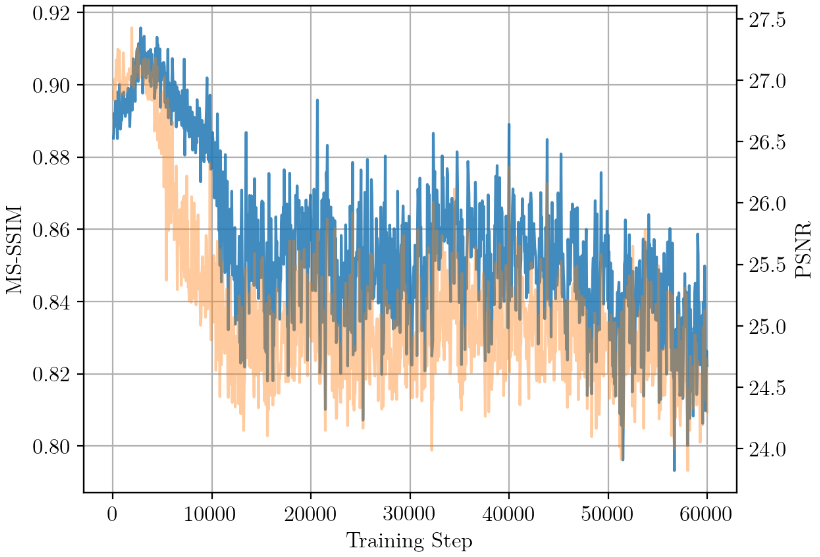

Figure 4.5: MS-SSIM (dark blue) and PSNR (light orange) image metrics during Training

Chapter 5

Conclusion

Our experiments demonstrate that while a spectrogram-based approach using an image compression model can achieve competitive bitrates, the perceptual quality currently lags behind state-of-the-art neural audio codecs. The finetuned ILLM model outperforms its pretrained counterpart on the consistency metric, but still introduces significant artifacts, especially at lower frequencies. The VisQOL metric fails to capture these quality issues, highlighting the need for perceptually-aligned evaluation methods. Key challenges to address include adapting the model architectures to better capture the unique characteristics of audio spectrograms and improving the phase reconstruction process. Despite these limitations, our results show the potential for leveraging image compression techniques in the audio domain and serve as a foundation for spectrogram-based audio coding.

While high-frequency components of the audio signal are reproduced reasonably well with our method, it fails at faithfully reproducing the low frequencies. We hypothesize that the underlying issue is related to how standard convolutional architectures used in image compression models are not well-suited to capture the differing perceptual importance of low versus high frequencies in audio spectrograms. The 2D convolutional layers apply filters uniformly across the input, whereas for audio, more representational capacity should be dedicated to the lower frequencies.

The use of spectrograms as an intermediate representation for audio compression offers several potential advantages. By working in the frequency domain, the model can more easily capture and exploit the spectral characteristics of the audio signal. This is analogous to how image compression models learn to extract and compress features at different spatial scales. Furthermore, the two-dimensional structure of spectrograms allows for the application of well-established image processing techniques and architectures, such as convolutional neural networks. This opens up the possibility of leveraging advances in image compression for the audio domain.

Another key challenge is the reconstruction of the phase information from the compressed magnitude spectrogram. While the PGHI algorithm provides a computationally efficient solution, it introduces artifacts that degrade the perceptual quality. The development of differentiable phase reconstruction methods could enable end-to-end optimization of the compression pipeline, potentially leading to improved quality. Alternatively, the use of complex-valued neural networks or phase-aware loss functions could allow the model to jointly compress both magnitude and phase information.

Despite these challenges, our work demonstrates the feasibility of a spectrogram-based approach to audio compression using image compression models. The competitive bitrates achieved by our method, even with the additional redundancy of the spectrogram representation, highlight its potential. By addressing the limitations identified in this study, such as the need for perceptually-aligned architectures and improved phase reconstruction, spectrogram-based methods could rival or even surpass the performance of state-of-the-art neural audio codecs.

5.1 Further Work

It is worth noting that research into the use of deep learning for audio task has traditionally lagged behind the work in computer vision. This is partly due to the historical focus on image-related tasks in the machine learning community, as well as the lack of large-scale audio datasets comparable to ImageNet or Open Images in the image domain. As a result, the development of neural audio compression methods has not received the same level of attention and investment as image compression. However, as the demand for efficient audio compression continues to grow, driven by applications like streaming services, teleconferencing, and immersive audio experiences, there is a clear need for increased research efforts in this area. By leveraging the advances made in image compression and adapting them to the specific challenges of audio data, we can work towards closing this gap and unlocking the full potential of deep learning for audio compression.

Further research into spectrogram-based audio compression using image compression models holds great promise for advancing the state-of-the-art in this field. One key area for future exploration is the development of custom architectures specifically tailored to the unique characteristics of audio spectrograms. While our approach of adapting existing image compression models has shown potential, it is important to recognize the inherent differences between working with audio and image data in machine learning research. Audio signals exhibit distinct temporal and spectral properties that may not be optimally captured by architectures designed for two-dimensional image data. By designing novel architectures that take into account the specific structure and perceptual relevance of audio features, we may be able to achieve significant improvements in both compression efficiency and perceptual quality.

Another interesting avenue for future research is to investigate the conditions under which objective metrics like VisQOL provide reliable assessments of audio quality, and when they may fall short. Our experiments revealed that VisQOL scores did not always align with subjective evaluations, particularly in the presence of phase distortions introduced by the spectrogram-based approach. Developing a deeper understanding of the strengths and limitations of these metrics could guide the development of more perceptually-aligned evaluation methods, which are crucial for accurately assessing the performance of audio compression algorithms. Additionally, exploring alternative spectrogram representations, such as mel-spectrograms or constant-Q transforms, may offer benefits in terms of capturing perceptually relevant audio features while reducing redundancy in the representation.

One major challenge in our spectrogram-based approach to audio compression is the non-differentiability of current phase reconstruction algorithms. While there are autograd-compatible implementations for the short-time Fourier transform (STFT) and its inverse, the same does not hold true for the phaseless inversion problem. This limitation prevents the implementation of a fully end-to-end trained compression pipeline, as the gradient cannot be propagated through the phase reconstruction step. To address this issue, future research could focus on developing a differentiable relaxation of the phase-gradient heap integration (PGHI) algorithm, described in [30] and section 3.1.

Spectrogram-based audio compression using image compression models presents a promising direction for future research. By addressing the challenges identified in this work, such as the need for custom architectures, perceptually-aligned evaluation metrics, and differentiable phase reconstruction methods, we can continue to push the boundaries of audio compression performance.

Bibliography

[1] "ITU Facts and Figures 2023," Tech. Rep., 2023. [Online]. Available: https://www.itu.int/itu-d/reports/statistics/2023/10/10/ff23-internet-traffic/

[2] "Sandvine Global Internet Phenomena Report 2024," Tech. Rep., 2024. [Online]. Available: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2024/GIPR/GIPR%202024.pdf

[3] K. Brandenburg and H. Popp, "An introduction to MPEG Layer-3," 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:14474869

[4] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-Quality, Low-Delay Music Coding in the Opus Codec," 2016, _eprint: 1602.04845.

[5] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[6] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2012. [Online]. Available: https://books.google.ch/books?id=VWq5GG6ycxMC

[7] J. Duda, "Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding," 2014, _eprint: 1311.2540.

[8] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, first edition ed. Cambridge University Press, Oct. 2003, published: Hardcover. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/itila/book.html

[9] Y. Yang, S. Mandt, and L. Theis, "An Introduction to Neural Data Compression," 2023, _eprint: 2202.06533.

[10] J. Townsend, T. Bird, and D. Barber, "Practical Lossless Compression with Latent Variables using Bits Back Coding," 2019, _eprint: 1901.04866.

[11] P. Chou, T. Lookabaugh, and R. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 31–42, 1989.

[12] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," 2018, _eprint: 1711.00937.

[13] Y. Blau and T. Michaeli, "Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff," 2019, _eprint: 1901.07821.

[14] J. Jiang, "Image compression with neural networks – A survey," *Signal Processing: Image Communication*, vol. 14, no. 9, pp. 737–760, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596598000411

[15] K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, and D. Wierstra, "Towards Conceptual Compression," 2016, _eprint: 1604.08772.

[16] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson, "High-Fidelity Generative Image Compression," 2020, _eprint: 2006.09965.

[17] M. J. Muckley, A. El-Nouby, K. Ullrich, H. Jégou, and J. Verbeek, "Improving Statistical Fidelity for Neural Image Compression with Implicit Local Likelihood Models," 2023, _eprint: 2301.11189.

[18] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "Wavenet based low rate speech coding," 2017, _eprint: 1712.01120.

[19] C. Garbacea, A. v. den Oord, Y. Li, F. S. C. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low Bit-rate Speech Coding with VQ-VAE and a WaveNet Decoder," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2019.8683277

[20] D. N. Rim, I. Jang, and H. Choi, "Deep Neural Networks and End-to-End Learning for Audio Compression," 2021, _eprint: 2105.11681.

[21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," 2021, _eprint: 2107.03312.

[22] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "MusicLM: Generating Music From Text," 2023, _eprint: 2301.11325.

[23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," 2022, _eprint: 2210.13438.

[24] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and Controllable Music Generation," 2024, _eprint: 2306.05284.

[25] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-Fidelity Audio Compression with Improved RVQGAN,” 2023, _eprint: 2306.06546.

[26] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, “Vector-quantized Image Modeling with Improved VQGAN,” 2022, _eprint: 2110.04627.

[27] A. Marafioti, N. Holighaus, N. Perraudin, and P. Majdak, “Adversarial Generation of Time-Frequency Features with application in audio synthesis,” 2019, _eprint: 1902.04072.

[28] S. Forsgren and H. Martiros, “Riffusion - Stable diffusion for real-time music generation,” 2022. [Online]. Available: https://web.archive.org/web/20230105130355/https://www.riffusion.com/about

[29] N. Cammarata, G. Goh, S. Carter, L. Schubert, M. Petrov, and C. Olah, “Curve Detectors,” *Distill*, 2020.

[30] Z. Průša, P. Balazs, and P. L. Søndergaard, “A Noniterative Method for Reconstruction of Phase From STFT Magnitude,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[31] D. Griffin and J. Lim, “Signal estimation from modified short-time Fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

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

[33] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A Dataset For Music Analysis,” 2017, _eprint: 1612.01840.

[34] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, 2001, pp. 749–752 vol.2.

[35] M. Chinen, F. S. C. Lim, J. Skoglund, N. Gureev, F. O’Gorman, and A. Hines, “ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric,” 2020, _eprint: 2004.09584.

[36] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms,” 2019, _eprint: 1812.08466.

[37] Y. Wu*, K. Chen*, T. Zhang*, Y. Hui*, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[38] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR - half-baked or well done?” 2018, _eprint: 1811.02508.

[39] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, 2003, pp. 1398–1402 Vol.2.

[40] R. C. Streijl, S. Winkler, and D. S. Hands, “Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives,” *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, Mar. 2016. [Online]. Available: https://doi.org/10.1007/s00530-014-0446-1

Appendix A
Appendix

Method	**Visqol**	**FAD**	**SI-SDR**	**Bitrate (kbps)**
Encodec @ 24kHz	2.76	0.053	8.07	12.0
Encodec @ 48kHz	4.21	0.035	10.23	12.0
Descript	4.16	0.026	10.23	8.0

Table A.1: Objective Metrics for the 24 kHz Encodec model