Audio codecs for generative music
Descript audio codec

Abstract

Language models have been successfully used to model natural signals such as images, speech and music. A key component of these models is a high-quality neural compression model that compresses high-dimensional natural signals into low-dimensional discrete tokens. To this end, the report presents a high-fidelity generalized neural audio compression algorithm that converts 44.1 KHz audio into tokens at a 90x compression rate with only 8kbps bandwidth. The algorithm achieves this by combining advances in high-fidelity audio generation with better vector quantization techniques in the image domain and improved adversarial and reconstruction loss. The use of a generic model to compress all domains (speech, ambient, music, etc.) makes it broadly applicable to all generative models of audio. The model is compared to similar audio compression algorithms and the new approach's methods are found to significantly outperform them.

Keywords: Language models, Neural compression models, Vector quantization, Generative models.

1 Introduction

Generative music technology aims to lower the threshold of music creation through modern generative modeling technology so that more people can easily participate in music creation. The development of this technology has changed the dependence of traditional music creation on professional knowledge, making music creation gradually move from the professional field to the public. In recent years, some popular generative music creation software has gradually entered the public's field of vision, the most notable of which is Suno, which is known as the "ChatGPT of the music industry", and supports users to generate high-quality music works through simple text prompts, without any knowledge of music theory, and can create music works with a rich style and a variety of genres. Suno supports users to generate high-quality music compositions through simple text prompts, without any knowledge of music theory. Users only need to enter a theme, style or lyrics, and Suno can quickly generate complete music content up to two minutes long, providing great convenience for music lovers and creators.

Neural network architectures play an important role in the technical realization of generative music. Currently, the most commonly used generative models for music composition tasks include Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN). These models are able to learn complex musical features from large amounts of music data to generate musical fragments that are both musically regular and innovative. In addition, Natural Language Processing (NLP)-based models such as Long Short-Term Memory (LSTM) and Transformer are widely used for tasks such as lyric generation and melody composition. These models are able to excel in capturing the temporal dependence and contextual relevance of musical sequences, thus enhancing the coherence and creativity of generative music.

But there are many problems in the process. Generative modeling of high-resolution audio is difficult due to its high-dimensionality (about 44,100 samples per second) [11, 13] and the simultaneous existence of short- and long-term dependent structures on different time scales. To address this problem, audio generation is usually divided into two phases: prediction of audio conditional on some intermediate representation (e.g., Mayer spectrograms); and prediction of intermediate representations based on some conditional information (e.g., text). This can be viewed as a hierarchical generative model with observable intermediate variables. Naturally, an alternative formulation is to learn the intermediate variables via a variational autoencoder (VAE) framework and use a learned derived conditional prior to predict the latent variables based on the conditional information. This formulation combining continuous latent variables and a highly expressive prior trained by normalizing flows has achieved remarkable success in the field of speech synthesis.

A closely related concept is the use of VQ-VAE [14] to train the same variational autocoder (VAE) with discrete latent variables. Arguably, discrete latent variables are the superior choice because more expressive priors can be trained using powerful autoregressive models developed for discrete variable distributions. Specifically, the Transformer language model [16] has demonstrated the ability to learn arbitrarily complex distributions such as text [4], images [7], audio [3], music [1], etc. as data and model capacity grow. While modeling a prior is relatively straightforward, modeling discrete latent codes via quantized self-encoders remains a challenge.

Learning discrete coding can be regarded as a lossy compression task in which the audio signal is compressed into a discrete latent space by vector quantization of the self-encoder representation using a fixed-length codebook. This audio compression model needs to satisfy the following properties: 1) high-fidelity reconstruction: the ability to reconstruct audio with high fidelity and avoid generating artifacts; 2) high compression rate: achieve a high level of compression while downsampling in the time dimension to learn a compact representation that discards low-level, imperceptible details while preserving high-level structure; 3) generality: the ability to handle all types of audio, e.g., speech, music, ambient sounds, etc., as well as different audio coding and different sampling rates, and accomplishes this using a single generalized model.

Although some recent neural audio compression algorithms (e.g., SoundStream [20] and EnCodec [6]) satisfy the above properties to some extent, they typically face problems similar to those of GAN-based generative models. Specifically, these models incur problems such as pitch artifacts, pitch and periodicity artifacts, and imperfect modeling of high frequencies, resulting in generated audio that is significantly different from the original audio. In addition, these models tend to be optimized for specific types of audio signals (e.g., speech or music), but are weak in handling generic sound modeling.The article makes the following contributions:

- ★ Introducing the **improved RVQGAN**, a high-fidelity general-purpose audio compression model that compresses 44.1 KHz audio into discrete code at 8 kbps bit rate (90x compression) while minimizing quality loss and having fewer artifacts. When evaluated using quantitative metrics and qualitative listening tests, the model significantly outperforms state-of-the-art methods even at lower bit rates (higher compression).
- ★ A key problem exists in existing models that underutilize bandwidth due to **codebook folding** (where a small portion of the code is not used) and using improved codebook learning techniques.
- ★ Clarifies that the side effect of **quantizer loss** - a technique designed to allow individual models to support variable bit rates - can actually impair full-bandwidth audio quality, and proposes a solution to mitigate it.

- ★ Descript audio codec is a universal audio compression model that can handle speech, music, ambient sound, **different sample rates and audio coding formats**.

2 Related works

Audio codec technology plays a crucial role in generative music. Audio codecs not only significantly improve the efficiency of music generation and distribution, but also excel in ensuring sound quality and compatibility. This technology compresses and optimizes the raw audio signal to produce high-quality music files that are easy to store and transmit, enhancing the user experience and driving the digital transformation of the music industry. Audio encoders have become an indispensable technology in the modern music industry, laying a solid technical foundation for the development of generative music.

2.1 High fidelity neural audio synthesis

Generative Adversarial Networks (GANs) are a solution for generating high-quality audio and fast inference due to their feed-forward (parallel) generators.MelGAN [12] successfully trains a GAN-based spectrogram inversion (neural acoustic code) model. It introduces a multi-scale waveform discriminator (MSD) to penalize structure at different audio resolutions, and a feature matching loss to minimize the L1 distance between the discriminator feature maps of real and synthesized audio.

HifiGAN improves on this approach by introducing a multi-periodic waveform discriminator (MPD) for high-fidelity synthesis and by adding an auxiliary melting reconstruction loss for fast training. UnivNet [9] introduces the Multi-Resolution Spectrogram Discriminator (MRSD) to generate audio with a clear spectrogram.BigVGAN [8] extends the HifiGAN [10] method by introducing a periodic inductive bias using the snake activation function. It also replaces MSD in HifiGAN with MRSD to improve audio quality and reduce pitch and periodic artifacts.

While these GAN-based learning techniques are used for vocoder coding, these methods can easily be applied to neural audio compression as well. Our improved RVQGAN model closely follows the BigVGAN training methodology, but with some key changes. Our model uses a new multi-band, multi-scale STFT discriminator to mitigate aliasing artifacts, and a multi-scale Mel reconstruction loss to better model fast transients.

2.2 Neural audio compression models

VQ-VAE has been the dominant model for training neural audio codecs. The first VQ-VAE-based speech codec model used the original architecture in VQ-VAE with a convolutional encoder and an autoregressive wave-net decoder.SoundStream was one of the first general-purpose compression models capable of handling a wide range of audio types while supporting different bitrates using a single model. They use a fully causal convolutional encoder and decoder network and perform residual vector quantization (RVQ). The model is trained using the VQ-GAN [7] formulation, which adds antagonistic loss and feature matching loss to multiscale spectral reconstruction loss.EnCodec follows SoundStream's approach closely, but with a few improvements that improve quality.EnCodec uses a multiscale STFT discriminator and multiscale spectral reconstruction loss. They use a loss balancer that adjusts the loss weights according to changes in the gradient from the discriminator.

DAC also uses a convolutional encoder-decoder architecture, residual vector quantization, and adversarial perceptual loss. However, there are the following key differences: 1) the use of Snake activations introduces periodic inductive bias 2) codebook learning is improved by projecting the coding into a low dimensional [18] space 3) best practices in adversarial loss and perceptual loss design are used to obtain stable training formulations with fixed loss weights and without the need for complex loss equalizers.

2.3 Language modeling of natural signals

Neural language models have had great success in a variety of tasks, such as open-ended text generation [4] with contextual learning capabilities. A key component of these models is self-attention [16], which is capable of modeling complex long-range dependencies, but at a computational cost that is quadratic with the length of the sequence. This cost is unacceptable for natural signals of very high dimensionality such as images and audio, which require compact mapping into discrete representation spaces. Such mappings are typically learned using VQ-GANs, and then autoregressive transformers are trained on the discrete tokens. This approach has been successful in the image [15, 18, 19], audio [3, 17], video and music [1, 5] domains.Codecs such as SoundStream and EnCodec have been used in audio generation models such as AudioLM [2], MusicLM [1] and VALL-E [17] .DAC can be a direct replacement for the audio tokenization models used in these approaches, resulting in higher audio fidelity and maximizing entropy code representation to improve learning efficiency.

3 Method

3.1 Overview

The Descript audio codec is built on top of the VQ-GAN framework, using the same model as SoundStream and EnCodec. The model uses the fully convolutional encoder-decoder network from SoundStream, which performs temporal downscaling using a selected step factor. The VQ-GAN and SoundStream structures are shown in Figure 1 and Figure 2.

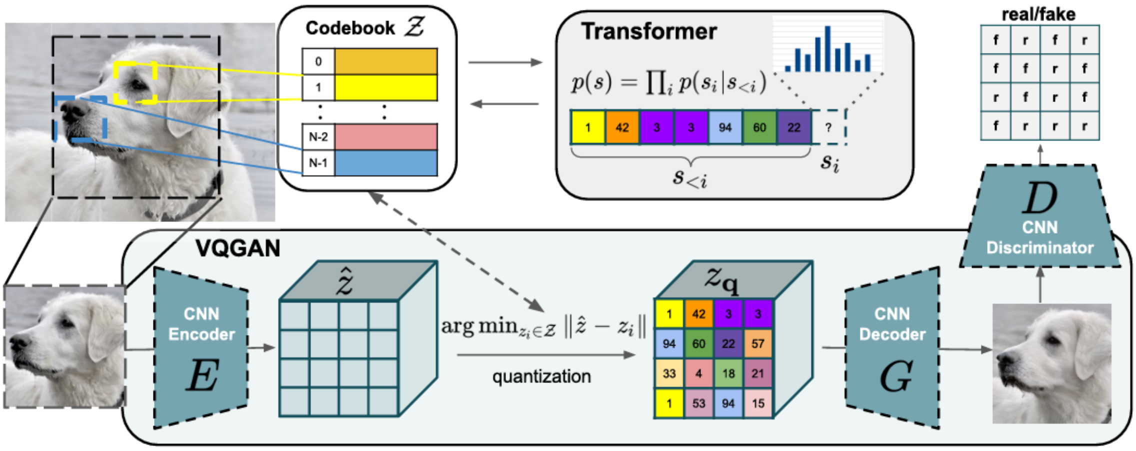

Figure 1. Model uses a convolutional VQGAN to learn a code set of context-rich visual parts and subsequently models their composition using an autoregressive transformer architecture.Discrete coding would have provided an interface to these architectures, while a patch-based discriminator enables strong compression while maintaining high perceptual quality.

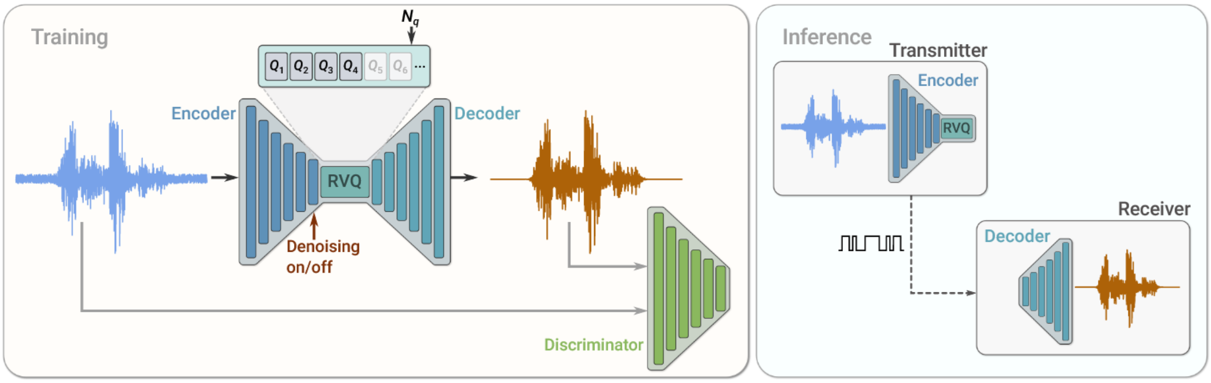

Figure 2. SoundStream model architecture. A convolutional encoder produces a latent representation of the input audio samples, which is quantized using a variable number n_q of residual vector quantizers (RVQ).

The codecs are quantized according to recent literature using Residual Vector Quantization (RVQ), a method that recursively quantizes the residuals with different codebooks after an initial quantization step. Quantizer filtering is applied during training, allowing a single model to operate at multiple target bitrates.The VQ-GAN and SoundStream structures are shown in Figure 3.

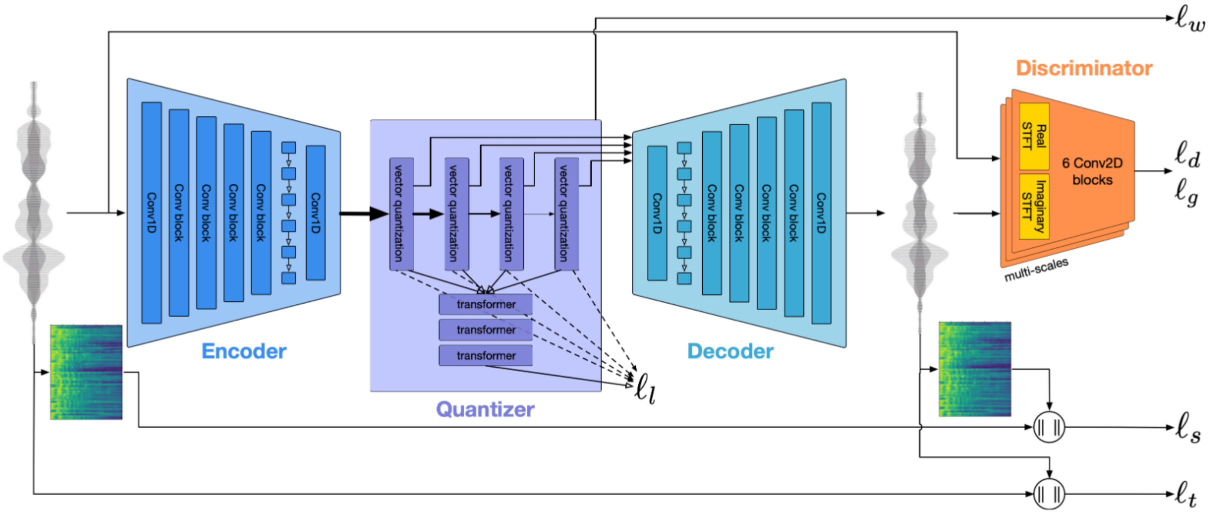

Figure 3. EnCodec : An encoder-decoder codec architecture that is trained by reconstruction (l_f and l_t) and adversarial loss (l_g for generator, l_d for discriminator). The residual vector quantization commitment loss (l_w) applies only to the encoder. Alternatively, we train a small Transformer language model for entropy coding on quantized units using l_l, which further reduces bandwidth.

An audio signal with sample rate f_s(Hz), encoder step factor M, and N_q layer of RVQ generates a discrete code matrix of shape S×N_q, where S is the frame rate defined as f_s/M. The Nq layer of the encoder step factor M and RVQ generate a discrete code matrix of shape S×N_q, where S is the frame rate defined as f_s/M. Compared to all baseline methods, model achieves a higher compression factor while outperforming them in terms of audio quality. Finally, a lower frame rate is required when training the language model on discrete codes, as it leads to shorter sequences.

3.2 Periodic activation function

It is well known that audio waveforms exhibit a high degree of periodicity (especially in vocal components, music, etc.). While current non-autoregressive audio generation architectures are capable of generating high-fidelity audio, they tend to exhibit disturbing pitch and periodicity artifacts. Furthermore, common neural network activations (e.g., Leaky ReLUs) are known to have difficulty extrapolating periodic signals and exhibit poor out-of-distribution generalization in audio synthesis.

In order to add a periodic inductive bias to the signal generator, the snake activation function proposed by Liu et al. is used and introduced into the audio domain of the BigVGAN neural acoustic code model.Replacing the Leaky ReLU activation with the Snake function was found to be an influential change in experiments that significantly improved audio fidelity.

$$\mathrm{snake}(x) = x + \frac{1}{\alpha}\sin^2(\alpha x) \tag{1}$$

where α controls the frequency of the periodic component of the signal.

3.3 Discriminator design

Multi-scale (MSD) and Multi-Periodic Waveform Discriminator (MPD) are used as in previous work, thus improving the audio fidelity. However, the spectrograms of the generated audio can still appear blurry and exhibit over-smoothing artifacts at high frequencies.The Multi-Resolution Spectrogram Discriminator (MRSD) is proposed in UnivNet to fix these artifacts, and is found by BigVGAN to also help in the reduction of pitch and periodicity artifacts. However, using amplitude spectrograms discards phase information that the discriminator could have used to penalize phase modeling errors. In addition, high-frequency modeling remains challenging for these models, especially at high sampling rates.

To address these issues, a sophisticated STFT discriminator was used on multiple timescales and was found to work better in practice and improve phase modeling. In addition, splitting the STFT into sub-bands slightly improves high-frequency prediction and mitigates aliasing artifacts, as the discriminator learns the discriminative features of a particular sub-band and provides a stronger gradient signal to the generator. Multiband processing is used to predict the audio in the subbands and then sum the subbands to produce full-band audio.

3.4 Quantizer dropout rate

Quantizer dropout was introduced in SoundStream to train a single compression model with variable bitrate. The number of quantizers Nq determines the bitrate, so for each input example, randomly sample

$$\mathrm{n} \sim \{1, 2, \ldots, N_q\} \tag{2}$$

and use only the first nq quantizers during training. However, under full bandwidth conditions, the use of quantizer culling degrades the audio reconstruction quality.

At lower bit rates, the discard probability of p = 0.5 is very close to the reconstruction quality of the baseline, while narrowing the gap with the full-bandwidth quality of the model trained without quantizer discards (p = 0.0).

Additional insights into the actual behavior of quantizer loss and its interaction with RVQ were also provided, and it was found that the combination of these techniques leads to quantization coding that learns information from the most to the least significant bit with each additional quantizer. When using $1, \ldots, N_q$ codebooks, more and more detail is added to each codebook. This interaction facilitates the training of hierarchical generative models on top of these codes, e.g., by dividing the codes into "coarse" markers (denoting the most important codes) and "fine" markers.

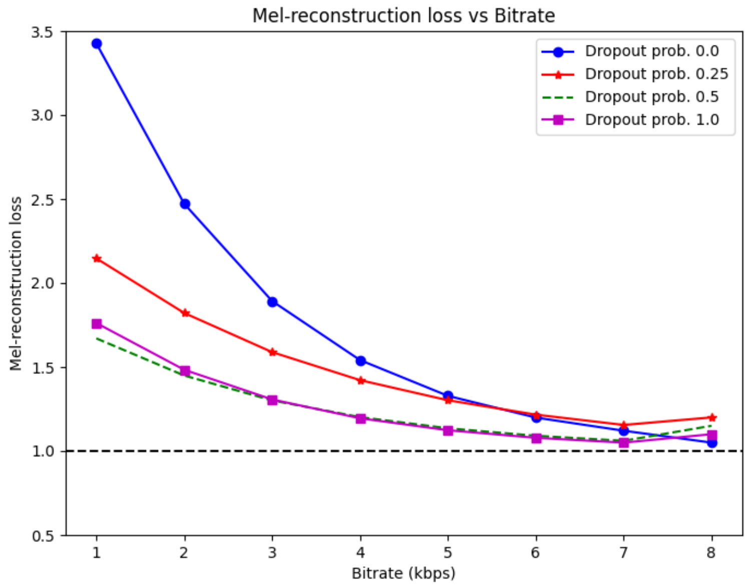

Figure 4. Effect of quantizer dropout on audio quality vs bitrate.

3.5 Loss functions

Codebook learning: Using simple codebook from the original VQ-VAE formulation and commitment loss with stopping gradients, and use a straight-through estimator to find backpropagation gradients through the codebook.

$$L = \log p\left(x \mid z_q(x)\right) + \left\|\mathrm{sg}\left[z_e(x)\right] - e\right\|_2^2 + \beta \left\|z_e(x) - \mathrm{sg}[e]\right\|_2^2 \tag{3}$$

where sg stands for the stopping gradient operator, which is defined to be homogeneous during forward computation and has zero partial derivatives, thus effectively restricting its operands to non-updating constants. The decoder optimizes only the first loss term, the encoder optimizes the first and last loss terms, and the embedding is optimized by intermediate loss terms.

Adversarial Loss: Waveform discrimination is performed using a multi-periodic discriminator and the proposed multi-band multi-scale STFT discriminator is used in the frequency domain. Use HingeGAN against loss formula and apply L1 feature matching loss.

$$L_{GAN}(\mathrm{D}, \mathrm{G}) := E_{x \sim P_S}[\log D(x)] + E_{z \sim P_Z}[\log(1 - D(G(z))] \tag{4}$$

where P_S is the sample distribution; $D(x)$ is the discriminator that takes $x \in X$ as input and outputs a scalar between [0,1]; and $G(z)$ is the generator that maps the sample z drawn from the distribution P_Z to the input space X.

Loss weights: a loss weight of 15.0 was used for the multiscale Mel loss, 2.0 for the feature matching loss, 1.0 for the confrontation loss, and 1.0 and 0.25 for the codebook loss and commitment loss, respectively. These loss weights are consistent with recent findings (which use a Mayer loss weight of 45.0), with simple adjustments to account for the multi-scale and log10 bases used in calculating the Mayer loss, and the model does not utilize the loss balancer proposed in EnCodec.

$$\tilde{g}_i = R \frac{\lambda_i}{\sum_j \lambda_j} \cdot \frac{g_i}{\langle \|g_i\|_2 \rangle_\beta} \tag{5}$$

define $g_i = \frac{\partial \ell_i}{\partial \hat{x}}$, $\langle \|g_i\|_2 \rangle_\beta$ as the exponential moving average of g_i in the last training batch. Given a set of weights (λ_i) and a reference norm R.Then, $\sum_i \tilde{g}_i$ is backpropagated into the network instead of the original $\sum_i \lambda_i g_i$. This changes the optimization problem, but allows (λ_i) to be interpreted independently of the natural scale of each loss.

Frequency-domain reconstruction loss: it is well known that Meier reconstruction loss improves stability, fidelity, and convergence speed, while multiscale spectral loss encourages the modeling of frequencies at multiple time scales. The model proposed in the article combines these two approaches by using L1 loss on mel spectrograms with computational window lengths of [32, 64, 128, 256, 512, 1024, 2048] and hopping length set to window_length / 4.Using the lowest hop size of 8 improves modeling of very quick transients that are especially common in the music domain.

EnCodec uses a similar loss formula, but uses both L1 and L2 loss terms and fixes the mel bin size to 64. Fixing the mel bin size can lead to holes in the spectrogram, especially if the filter length is low. Therefore, the model uses mel bin sizes [5, 10, 20, 40, 80, 160, 320] corresponding to the above filter lengths and verifies their correctness by manual checking.

4 Experiments details

4.1 Data source

The experiments train the model on a large dataset containing speech and music. For the speech part, the DAPS dataset, clear speech clips from DNS Challenge 4, the Common Voice dataset and the VCTK dataset were used. For the music part, the MUSDB18 dataset and the Jamendo dataset were used. Final. All audio was resampled to 44kHz.

During training, short segments were extracted from each audio file and normalized to -24 dB LUFS. the only data enhancement method applied was a random panning of the phase of the segments, with a uniformly distributed pan. For the evaluation, the speech of two retained speakers (F10 and M10) from the DAPS dataset was used, as well as a test division of the MUSDB18 dataset. The dataset is described in detail below:

Music

- **MUSDB18 dataset**: MUSDB18 dataset is a dataset of 150 full-length music tracks of different genres (10 hours duration) and their isolated drums, bass, vocals, and other stems.
- **Jamendo dataset**: This dataset contains over 55,000 full audio tracks covering 195 tags that fall into the categories of genre, instrument and mood/theme.

Speech

- **DNS Challenge 4**: ICASSP 2023 Deep Noise Suppression Challenge with Emotional Speech, French Speech, Italian, etc., totaling 892 GB.
- **Common Voice dataset**: The dataset consists of 28,118 hours of speech in 112 languages, with a total of 18,652 hours of data available for training, totaling 93GB.

- **VCTK dataset**: Released by the University of Edinburgh, 44 hours in total, 110 English speakers, each speaker reading about 400 sentences, sampling rate 48kHz, bit depth 16bit, total 7.8GB .
- **DAPS dataset**: 20speakers, 5 scripts of roughly 14min of speech per speaker, sampling rate 44.1kHz, totaling about 20h, size 21GB .

4.2 Balanced data sampling

I have taken special care with how the dataset is sampled. Although the dataset has been resampled to 44kHz, the data may be band-limited in some way. That is, some of the audio may have been originally sampled at a rate much lower than 44 kHz. this is particularly common in speech data, where the true sample rate of the underlying data can vary greatly (e.g. Common Voice data is typically 8-16 kHz). When I trained the model on different sample rates, I found that the resulting model often failed to reconstruct data above a certain frequency. Upon investigation, I found that this threshold frequency corresponds to the average true sampling rate of the dataset. To solve this problem the balanced data sampling technique was introduced.

First, the dataset was split into two types of data sources, known full-band sources - which were confirmed to contain energy frequencies up to the Nyquist frequency required by the codec (22.05kHz), and sources for which the maximum frequency could not be guaranteed. When sampling in batches, it is important to make sure that we are sampling full-band items. Finally, make sure that each batch has an equal number of items in each domain of speech, music, and ambient sound.

4.3 Model and training recipe

The model consists of a convolutional encoder, a residual vector quantizer, and a convolutional decoder. The basic building blocks of the network are a convolutional layer, which is upsampled or downsampled in certain steps, followed by a residual layer, which consists of a convolutional layer interleaved with nonlinear snake activations. The encoder has 4 such layers, each downsampling the input audio waveform at a rate of [2, 4, 4, 8, 8]. The decoder has 4 such layers and upsamples the input audio waveform at the rate of [8, 8, 4, 4, 2]. The dimension of the decoder is set to 1536. The model has a total of 76 million parameters, of which 22 million are for the encoder and 54 million for the decoder.

The model uses a multi-cycle discriminator and a complex multi-scale STFT discriminator. The former uses [2, 3, 5, 7, 11] cycles, while the latter uses [2048, 1024, 512] window lengths with jumps of 1/4 of the window length. For band splitting of the STFT, band limits [0.0, 0.1, 0.25, 0.5, 0.75, 1.0] are used. For reconstruction loss, the distance between log-pixel spectrograms with window lengths [32, 64, 128, 256, 512, 1024, 2048] is used, and the corresponding number of pixels per window is [5, 10, 20, 40, 80, 160, 320]. The hop length is 1/4 of the window length.

4.4 Objective and subjective metrics

To assess the effectiveness of the experiment, I used the following objective indicators:

1. **Codebook utilization rate**:A high utilization rate indicates that most of the codewords in the codebook are effectively used, while a low utilization rate indicates that many codewords do not contribute to the coding process.

2. **Signal-to-Noise Ratio (SNR)**:In the evaluation of reconstructed music, the signal-to-noise ratio is used to measure the quality of the reconstructed audio, in particular the difference between the reconstructed audio and the original audio.
3. **Mean Square Error (MSE)**:A metric used to quantify the error between predicted and true values, in the evaluation of reconstructed music, MSE is used to measure the difference between reconstructed and original audio.
4. **Mel Frequency Cepstral Coefficient(MFCC)**:A commonly used audio feature to characterize the spectral properties of audio signals. It is often used as a feature extraction tool in audio analysis, but in evaluating reconstructed music, a comparison of MFCCs can measure the similarity of the reconstructed audio to the original audio in terms of audio features.
5. **Time and frequency domain spectrograms**:An intuitive visualization tool in audio signal analysis, used to show signal variations in time and frequency. Often used as a quality comparison tool when evaluating reconstructed music, the fidelity of the model to the spectral information can be assessed by looking at the spectral differences between the original and reconstructed music.

5 Results and analysis

By comparing the input raw audio with the reconstructed audio, our calculations and analyses from the five perspectives of codebook utilization, signal-to-noise ratio, mean-square error, Mel frequency cepstrum coefficients, and time-frequency spectrograms yielded the results shown in Table 1.

Table 1. Objective evaluation of Descript audio codecs

Codec	Codebook Rate↑	Signal-to-Noise Ratio(SNR)↑	Mean Square Error(MSE)↓
DAC	92.45%	27.86 dB	0.001835

Regarding the objectively evaluated Mel Frequency Cepstral Coefficient (MFCC) and Time-frequency spectrogram are shown in Fig 5, Fig 6 and Fig 7, by comparing and observing we find that the reconstructed audio is almost similar to the original audio and the existence of error is very small.

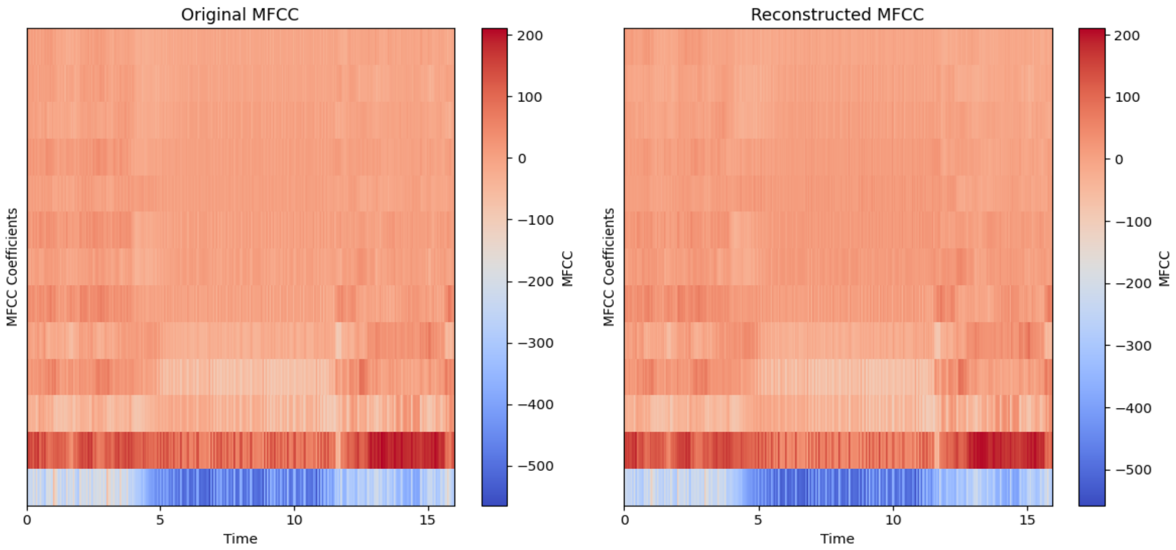

Figure 5. MFCC of original and reconstructed audio.

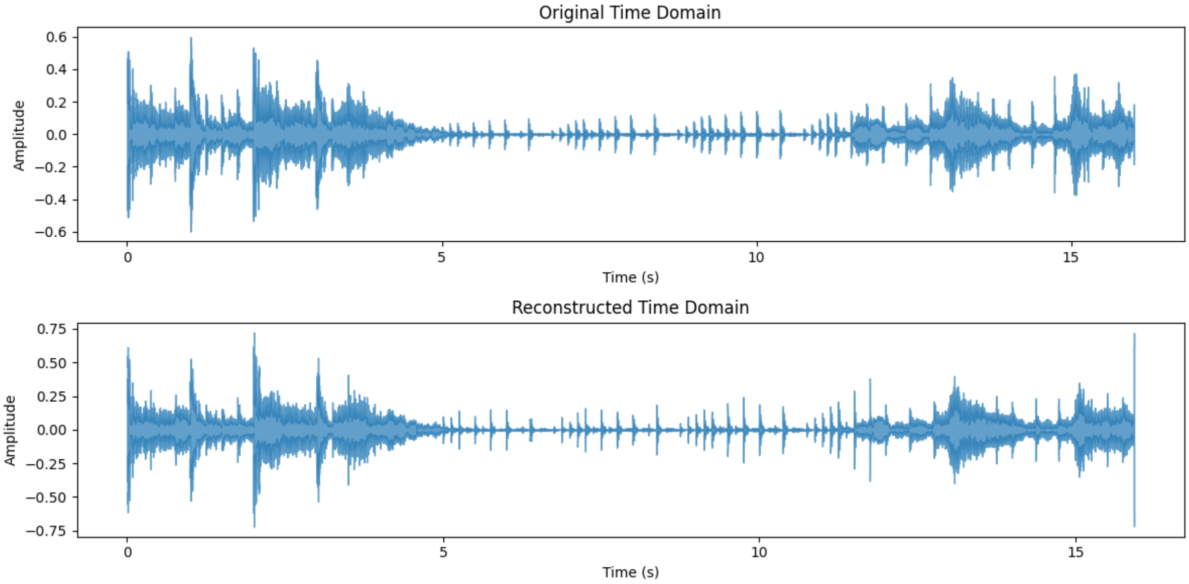

Figure 6. Comparison of the time-domain spectrograms of the original and reconstructed audio.

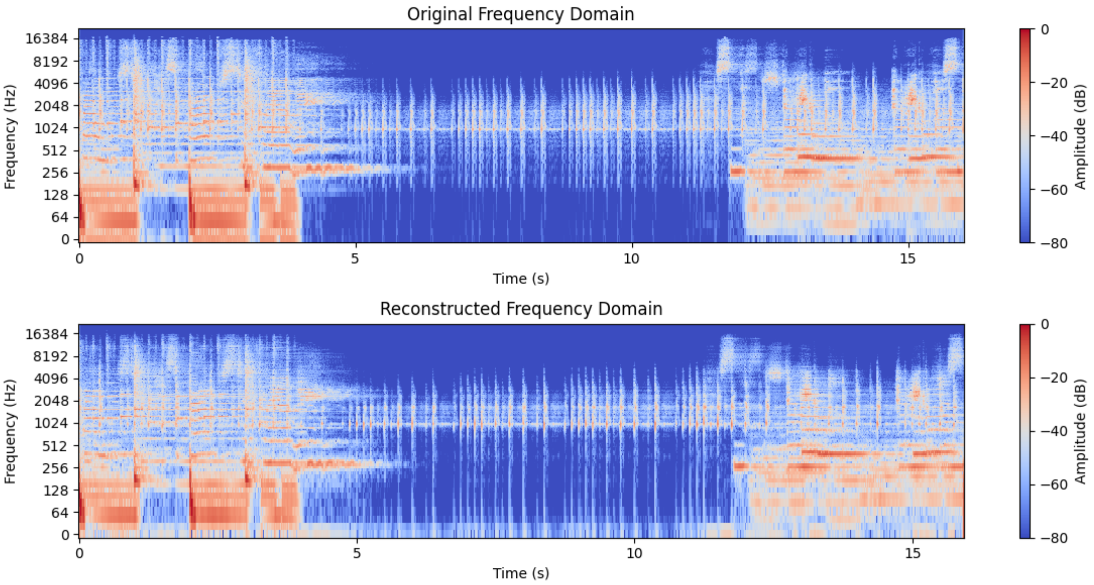

Figure 7. Comparison of frequency domain spectrograms of original and reconstructed audio.

In terms of subjective assessment, the original and reconstructed audio were subjectively evaluated artificially, and it was found that the reconstructed pure music was of very high quality, and the quality of the audio with lyrics and discourse was a little bit worse compared to the pure music, but the result was still very good. The relevant audio was submitted as supporting material with the zip package.

6 Conclusion and future work

In this article, we propose a high-fidelity general-purpose neural audio compression algorithm that achieves excellent compression ratios while preserving the audio quality of all types of audio data.Descript audio codec combines recent advances in audio generation, vector quantization techniques, and improved combating of loss and reconstruction loss. The codebook utilization, signal-to-noise ratio, and mean square error are calculated by objective and subjective evaluation of the reconstructed audio, and it is concluded that the article's approach achieves very good results.

The article's model can make generative modeling of full-band audio much easier. While this will lead to many useful applications such as media editing, text-to-speech synthesis, music synthesis, etc., it may also lead to harmful applications such as deep forgery. Care should be taken to avoid these applications. One possible approach is to add a watermark and/or train a classifier that can detect whether a codec has been applied, so that synthetic media generated by article-based codecs can be detected. In addition, the model of the paper is not perfect and still struggles to reconstruct some challenging audio. By slicing and dicing the results by domain, it was found that although the proposed codec outperforms other competing methods in all domains, it performs best for speech and is more problematic for ambient sounds, where it does not perfectly model the sound of some instruments, such as glockenspiels or synthesizers. In subsequent work, it is important to focus on these problems and develop different training tests for the code depending on the problem.

References

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

[2] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtgjamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019.

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

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

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

[6] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[8] Sang gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. Bigvgan: A universal neural vocoder with large-scale training. *arXiv preprint arXiv:2206.04658*, 2022.

[9] Won Jang, Dan Lim, Jaesam Yoon, Bongwan Kim, and Juntae Kim. Univnet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation. *arXiv preprint arXiv:2106.07889*, 2021.

[10] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033, 2020.

[11] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in Neural Information Processing Systems*, 32, 2019.

[12] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[13] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

[14] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

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

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[17] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

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

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

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