Neural Audio Codec for Latent Music Representations

Luca A. Lanzendörfer
ETH Zurich
lanzendoerfer@ethz.ch

Florian Grötschla
ETH Zurich
fgroetschla@ethz.ch

Amir Dellali
ETH Zurich
dellalia@ethz.ch

Roger Wattenhofer
ETH Zurich
wattenhofer@ethz.ch

Abstract

Neural audio codecs have become increasingly important for audio compression and, more recently, for creating tokenized representations for various generative downstream tasks. Consequently, the performance of neural audio codecs plays a crucial role in many applications. In this work, we introduce DisCodec, a high-fidelity neural audio codec for compressing 44.1kHz music into discrete or continuous latent representations. DisCodec leverages ConvNeXt and attention layers, an affine re-parametrization of the code vectors, and an improved commitment loss for better alignment between codebooks and model embeddings. We study comparisons of DisCodec against existing codecs, perform a comprehensive ablation of the proposed architecture, and demonstrate its performance against state-of-the-art neural audio codecs. We make the DisCodec codebase and model checkpoints available at `https://github.com/ETH-DISCO/discodec`.

1 Introduction

Neural audio codecs have recently emerged as viable alternatives to traditional codecs, such as MP3 [1] and Opus [2]. These neural codecs were initially introduced for general audio compression tasks with the aim of achieving high reconstruction quality at lower bitrates while still enabling real-time encoding and decoding [3, 4, 5]. Neural audio codecs leverage discretized representations based on the vector-quantized variational autoencoder architecture, initially introduced for the image domain [6]. Since the audio data is encoded into discrete tokens, these models have found use beyond just compression. With the advent of the transformer architecture [7] and its use of tokenized representations, neural audio codecs have become a key component in converting audio data into a compatible representation for transformer-based architectures. Many recent applications of generative audio tasks, such as text-conditioned audio generation [8, 9, 10, 11], therefore, rely on high-quality neural audio codecs.

There has been a significant amount of recent research on neural audio codecs [3, 4, 5, 12, 13, 14], with particular focus on extremely low bitrate regimes. However, existing neural audio codecs are all based on the same fundamental architecture. Therefore, we re-evaluate the design choices of existing neural audio codecs, particularly with regard to residual vector quantization, and propose changes to improve these architectures. DisCodec leverages ConvNeXt [15] and attention layers [7], an affine re-parametrization of the code vectors, and an improved commitment loss for better alignment between codebooks and model embeddings [16]. Furthermore, in addition to the discrete latent DisCodec model, we also open-source two unquantized versions of DisCodec at 64 and 128

Neural Information Processing Systems (NeurIPS) 2024 Workshop on AI-Driven Speech, Music, and Sound Generation.

latent dimensions. The 128-dimensional DISCODEC model achieves audio reconstruction nearly indistinguishable from the reference signal at 3x lower bitrate.

Our contributions can be summarized as follows:

- We present DISCODEC, a novel neural audio codec architecture using ConvNeXt layers, attention layers and trained with an improved commitment loss.
- We evaluate DISCODEC on various objective metrics and perform a MUSHRA listening test. Our findings show that the architectural changes of DISCODEC lead to better reconstruction fidelity compared to existing codecs.
- We ablate our model and investigate the trade-off between vocabulary size and the number of codebooks. Furthermore, code and pre-trained checkpoints are open-sourced, along with unquantized versions of the model.

Audio samples are available online.[1]

2 Related Work

Vector Quantization. Vector Quantization (VQ) enables deep neural networks to learn discrete representations by quantizing features into clusters called codebooks, showing impressive results in image and speech generation [6, 17]. However, VQ networks are challenging to optimize and often require specialized training techniques like exponential moving average updates [6] and codebook reset [18]. Residual Vector Quantization (RVQ) [3] extends VQ for efficient audio compression at various bitrates, especially beneficial at lower bitrates. RVQ employs a cascade of vector quantizers that iteratively quantize residual errors from previous stages, enabling refined compression with manageable complexity.

Neural Audio Codecs. Recent advances in neural audio processing have led to the development of neural audio codecs. WaveNet [19] introduced a deep generative model for high-fidelity audio synthesis from raw samples, revolutionizing speech synthesis. WaveGlow [20] extended this with a flow-based generative model. SoundStream [3] was one of the first neural audio codecs, introducing the VQ-VAE architecture to audio compression, originally proposed for image compression [6]. SoundStream employs adversarial losses, feature-space losses, and multi-scale spectral reconstruction loss. It uses exponential moving average for codebook updates [17] and introduces "quantizer dropout" to adapt to different bitrates, enhancing generalization and outperforming traditional codecs. EnCodec [4] follows a similar architecture to SoundStream, adding LSTM layers in the encoder and decoder and a different loss formulation. It also incorporates a Transformer-based language model for faster-than-real-time compression and decompression, and was trained on large audio datasets [21, 22, 23].

RVQ Adaptations. Descript Audio Codec (DAC)[5] introduces changes inspired by BigVGAN [24], including Snake activations [25] and an improved training recipe. It performs codebook lookup similar to Improved VQGAN [26], using lower-dimensional L2-normalized lookup vectors, improving codebook usage without special initialization methods. DAC was trained on various audio types resampled to 44,kHz. SNAC [14] is a concurrent codec based on DAC, introducing local multi-head attention layers in the encoder and decoder and strided pooling of codebooks to reduce the bitrate.

3 DISCODEC

3.1 Architecture

We provide an overview of DISCODEC (cf. Figure 1). DISCODEC builds on a modified DAC [5] architecture, utilizing 1D convolutions for downsampling, transposed 1D convolutions for upsampling, and interspersed residual connections. We deviate from existing architectures for neural audio compression at two key points.

First, we add a variant of the ConvNeXt [15] block adapted for 1D signals at the end of every encoder and decoder block. Secondly, we introduce multi-headed attention before applying the last

[1] https://lucala.github.io/DisCodec/

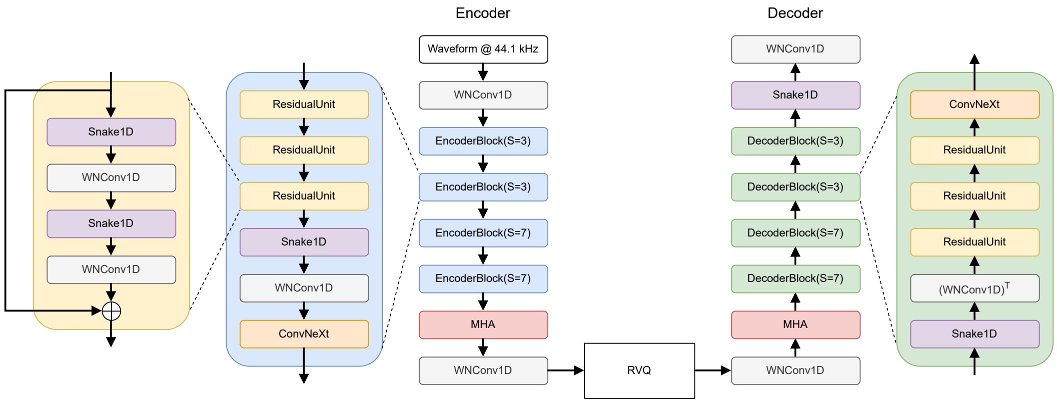

Figure 1: Architectural overview of DisCodec. The model consists of 1D convolution layers with weight normalization (WNConv1D) and stride S, Multi-Headed Attention layers (MHA), snake activations, and ConvNeXt layers. Latent embeddings are discretized with Residual Vector Quantization (RVQ) to obtain discrete representations.

convolution prior to the residual vector quantization layer, as well as after applying the first 1D convolution after quantization. We use a standard multi-stage vector-quantization method to enable the discretization of raw audio signals, with each subsequent step encoding the remaining residual between the true latent and the quantized representation at that stage.

In more detail, the RVQ algorithm works as follows: The input vector e is encoded using a nearest neighbor lookup in an embedding table. The resulting vector q_i is subtracted from e, with the next lookup being performed using $e - q_i$. We modify the RVQ block by splitting the embeddings into multiple disjoint groups and applying separate learned scale and shift parameters [16]. This has been found to reduce the codebook covariate shift, which can negatively affect the reconstruction quality.

We use codebook and commitment losses as defined in VQ-VAE [6] to align latent embeddings with quantized codebook vectors. While the codebook loss aims to align the codebook vector to the latent embedding, the commitment loss prevents the embedding space from growing arbitrarily [6]. These losses are defined as follows:

$$\mathcal{L}_{\text{codebook}} = ||\text{sg}[z_e(x)] - z_q(x)||_2^2, \tag{1a}$$

$$\mathcal{L}_{\text{commitment}} = ||z_e(x) - \text{sg}[z_q(x)]||_2^2, \tag{1b}$$

where sg is the stop-gradient operator, and the encoder output z_e and the decoder input z_q are the unquantized and quantized latent vectors, respectively.

In addition, we make improvements to the vector quantization process, adopting an affine reparametrization of the code vectors [16] $\hat{q} = \mu + \sigma * q$, where q is the original code vector, while μ and σ are learned parameters for the mean and standard deviation.

Furthermore, we adopt the synchronized update rule [16] defined as:

$$z_q^{(t+1)} \leftarrow (1 - \eta) \cdot z_q^{(t)} + \eta \cdot z_e^{(t)} + \eta^2 \cdot \frac{\partial \mathcal{L}_{\text{task}}}{\partial z_q}, \tag{2}$$

where $\mathcal{L}_{\text{task}}$ is a combination of $\mathcal{L}_{\text{mel}}$, $\mathcal{L}_{\text{feat}}$, and $\mathcal{L}_{\text{adversarial}}$, and η is the learning rate. These changes make it possible to set considerably higher weights for the codebook and commitment losses, which we found to increase audio reconstruction fidelity.

In addition, we analyze model performance without vector quantization. For these continuous latent models, we do not use the VQ block to discretize the latent, opting instead for a VAE-based approach [27], utilizing a cyclic annealing schedule [28] for the KL loss term.

3.2 Training

Previous approaches often face the issue of codebook collapse, leading to the utilization of only a small subset of codebook tokens for quantization [5]. There have been attempts to mitigate codebook collapse by using methods such as reviving dead codebooks using exponential moving average [6, 3, 4]. DAC [5] addresses codebook collapse by performing the codebook lookup in a lower dimensional space. Similarly to DAC, we also use a learned projection to mitigate codebook collapse. Using D dimensional latent embeddings and M lower-dimensional latent embedding with $D \gg M$. For DISCODEC, we set $D = 1024$ and $M = 8$. The learned projection matrices $W_{\text{in}} \in \mathbb{R}^{D \times M}$ and $W_{\text{out}} \in \mathbb{R}^{M \times D}$ are used to find the closest L2-normalized vector, which is analogous to a cosine similarity. The lookup is defined as follows:

$$z_q(x) = W_{\text{out}} q_k, \tag{3}$$

where $k = \arg\min_j \|\ell_2\left(W_{\text{in}} z_e\left(x\right)\right) - \ell_2(q_j)\|_2$ and q_k is a codebook vector. This differs from the codebook lookup method used in other approaches, which do not L2-normalize the embedding vector z_e and lookup vector z_q. These approaches operate in the dimension of the latent vector z_e, which is significantly higher (1024 compared to 8). This default lookup can be seen as setting $W_{\text{in}} = W_{\text{out}} = I$ and not using L2-normalization.

Losses. To train DISCODEC we use a combination of losses [5] with the following weightings:

$$\mathcal{L}_{\text{total}} = 15 \cdot \mathcal{L}_{\text{mel}} + 2 \cdot \mathcal{L}_{\text{feat}} + 1 \cdot \mathcal{L}_{\text{adv}} + 10 \cdot \mathcal{L}_{\text{codebook}} + 2.5 \cdot \mathcal{L}_{\text{commit}}, \tag{4}$$

where $\mathcal{L}_{\text{mel}}$ is the multi-scale mel-spectogram loss, $\mathcal{L}_{\text{feat}}$ the feature matching loss, $\mathcal{L}_{\text{adv}}$ the adversarial loss. $\mathcal{L}_{\text{codebook}}$ and $\mathcal{L}_{\text{commit}}$ refer to the codebook and commitment losses for the RVQ layer, respectively. We find that the higher codebook and commitment loss weights at 10 and 2.5, compared to 1.0 and 0.25 as proposed by DAC, perform better for us. This is made possible by the affine re-parametrization of the code vectors and the improved commitment loss formulation.

The Mel-spectrogram loss was initially introduced by Hifi-GAN [29] and is defined as:

$$\mathcal{L}_{\text{mel}}(F, G) = \mathbb{F}_x \left[||\phi(x) - \phi(G(F(x)))||_1 \right], \tag{5}$$

where F is the encoder and G is the decoder of the audio codec, and ϕ transforms a waveform into a mel-spectrogram. We use a multi-scale mel-spectrogram loss with window lengths of 32, 64, 128, 256, 512, 1024, and 2048, and the hop lengths set to 25% of the window length, and 5, 10, 20, 40, 80, 160, and 320 mel bands, respectively.

4 Evaluation

We perform a series of ablations to validate the effectiveness of DISCODEC. We analyze the trade-off between the number of codebooks, vocabulary size, and scaling laws of the codecs in terms of bitrate and compare DISCODEC to various other models. All DISCODEC models are trained on MTG-Jamendo [23] and an internal dataset of 120 hours of vocal tracks.

4.1 Metrics

We evaluate the performance of all models using a multi-scale mel-spectrogram loss (cf. Section 3.2), a multi-scale STFT loss, L1 loss on the waveform, and ViSQOL [30]. We subjectively compare the models by performing listening tests according to the MUSHRA protocol [31] with a group of audio experts. In this setup, listeners are exposed to 5-second audio snippets from an unseen evaluation set. 7 signals, including the hidden reference and a low-passed anchor, are compared in a single test, with an entire run being composed of 10 such tests.

4.2 Number of Codebooks & Codebook Size

To find a good trade-off between vocabulary size and the number of codebooks, we investigate the scaling behavior of the number of codebooks while keeping the bitrate constant. To achieve this, we compensate for a reduced number of codebooks by increasing the vocabulary size. Surprisingly, we find that a model with a vocabulary size of 32k (15 bits per token) and 4 codebooks obtains similar reconstruction quality compared to a model with a vocabulary size of 1024 (10 bits per token) and 6

Table 1: Performance comparison. Latents refer to the latent dimension (continuous) or number of codebooks (discrete). Best scores for discrete models in brown, and for continuous models in blue.

Model	VQ	Latents	Bitrate (kbps)	Mel ↓	SI-SDR ↑	STFT ↓	L1 ↓	ViSQOL ↑	MUSHRA↑	Params
Reference	-	-	706	-	-	-	-	-	98.2 ± 5.80	-
DisCodec	✗	64	102	0.740 ± 0.048	13.862 ± 4.085	7.044 ± 0.497	0.022 ± 0.010	4.45 ± 0.29	87.4 ± 11.3	109M
DisCodec	✗	128	205	0.474 ± 0.038	17.419 ± 5.180	6.016 ± 0.399	0.013 ± 0.007	4.56 ± 0.27	97.3 ± 4.30	109M
DisCodec	✓	8	8	1.139 ± 0.073	10.771 ± 2.866	7.689 ± 0.606	0.034 ± 0.011	4.35 ± 0.21	85.6 ± 11.0	109M
EnCodec	✓	16	12	1.901 ± 0.138	9.274 ± 2.793	14.011 ± 2.531	0.045 ± 0.013	3.46 ± 0.49	70.6 ± 17.9	15M
DAC	✓	9	7.74	1.099 ± 0.059	10.387 ± 2.450	5.680 ± 0.394	0.039 ± 0.011	4.37 ± 0.11	84.5 ± 12.7	76M

codebooks. Overall, our observations support the conventional approach of utilizing more codebooks for higher bitrate models, particularly when considering the trade-off with the number of bits per token. However, our findings indicate that models with higher vocabulary sizes might still be feasible.

To further investigate the behavior of larger vocabulary sizes, we encode samples from our test set and compute the entropy over the tokens used in every codebook. We then normalize the entropy with the maximally achievable entropy by every model, which is equal to the number of bits used per token. The resulting values indicate the percentage of the theoretically possible entropy (and, in turn, codebook usage) that could be achieved by the respective model.

In Figure 2, we observe that exceedingly large vocabulary sizes achieve lower codebook usage. In contrast, models with commonly used vocabulary sizes (10-12 bits per token) can make use of their tokens more uniformly. This aligns with our observations on the performance of these models. We find that models with 15 bits per token retain high codebook utilization, which is close to optimal. Regimes with vocabulary sizes in this range remain under-explored and constitute an interesting avenue for future research.

4.3 Comparative Analysis

We evaluate against pre-trained DAC [5] and EnCodec [4] models. All metrics in Table 1 are computed on the MUSDB-test set [32]. We observe in Table 1 that DAC exhibits strong performance. The VQ version of DisCodec obtains comparable performance on objective metrics and outperforms EnCodec on the subjective listening test conducted with audio experts, achieving an average score of 85.6 compared to 70.6 for EnCodec and 84.5 for DAC. The mean rating of the low anchor for the MUSHRA test is 32.3. While the continuous models use significantly more bandwidth, they are comparable in audio fidelity to the reference signal, making them ideal for downstream tasks where a continuous latent representation can be utilized (e.g., diffusion-based approaches). Providing the continuous models of DisCodec thus allows for further use cases without the loss in audio fidelity induced by vector quantization.

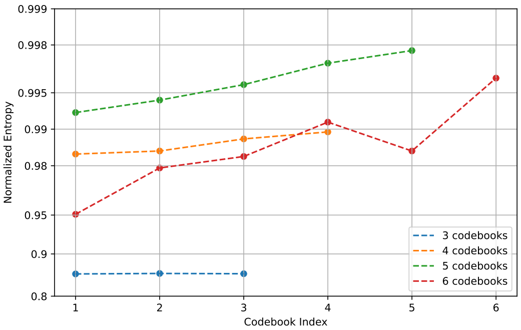

Figure 2: Visualization of entropy over multiple codebooks, calculated on our MTG-Jamendo test set. Values were computed as the entropy for every codebook, normalized by the number of bits available for the codebook. The resulting numbers indicate the codebook usage percentage.

5 Conclusion

Neural audio codecs have been shown to significantly outperform traditional codecs at low bitrates. In addition, these codecs transform audio into highly compressed latent representations, which transformer-based architectures can directly use to process and generate audio. However, since these models depend on the codec's latent representation, the resulting audio fidelity is inherently tied to the codec's reconstruction quality. To that end, we present DisCodec, a high-fidelity neural audio codec specifically trained for music that incorporates ConvNeXt and attention layers, as well as recent advancements in residual vector quantization. To further research on downstream tasks, we open-source the codebase and model checkpoints of DisCodec.

References

[1] M. Nilsson, “The audio/mpeg Media Type,” RFC 3003, Nov. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc3003

[2] J.-M. Valin, K. Vos, and T. Terriberry, “Definition of the opus audio codec,” Tech. Rep., 2012.

[3] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” 2021.

[4] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” 2022.

[5] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” 2023.

[6] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.

[8] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” 2023.

[9] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, “Audiolm: a language modeling approach to audio generation,” 2023.

[10] 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.

[11] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” 2024.

[12] S. Ji, M. Fang, Z. Jiang, R. Huang, J. Zuo, S. Wang, and Z. Zhao, “Language-codec: Reducing the gaps between discrete codec representation and speech language models,” *arXiv preprint arXiv:2402.12208*, 2024.

[13] D. Yang, S. Liu, R. Huang, J. Tian, C. Weng, and Y. Zou, “Hifi-codec: Group-residual vector quantization for high fidelity audio codec,” 2023.

[14] H. Siuzdak, F. Grötschla, and L. A. Lanzendörfer, “Snac: Multi-scale neural audio codec,” in *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*, 2024.

[15] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” 2022.

[16] M. Huh, B. Cheung, P. Agrawal, and P. Isola, “Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks,” 2023.

[17] A. Razavi, A. van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with vq-vae-2,” 2019.

[18] A. Łańcucki, J. Chorowski, G. Sanchez, R. Marxer, N. Chen, H. J. Dolfing, S. Khurana, T. Alumäe, and A. Laurent, “Robust training of vector quantized bottleneck models,” in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[19] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[20] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[21] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[22] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[23] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo 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. [Online]. Available: http://hdl.handle.net/10230/42015

[24] S. gil Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," 2023.

[25] L. Ziyin, T. Hartwig, and M. Ueda, "Neural networks fail to learn periodic functions and how to fix it," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1583–1594, 2020.

[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.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[28] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical annealing schedule: A simple approach to mitigating kl vanishing," 2019. [Online]. Available: https://arxiv.org/abs/1903.10145

[29] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.

[30] 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.

[31] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[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