PROGRESSIVE MULTI-STAGE NEURAL AUDIO CODING WITH GUIDED REFERENCES

Chanwoo Lee[1]*, Hyungseob Lim*[1]*, Jihyun Lee*[1]*, Inseon Jang*[2] *and Hong-Goo Kang*[1]

[1]Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea
[2]Electronics and Telecommunications Research Institution, Daejeon, South Korea

ABSTRACT

In this paper, we propose an effective multi-stage neural audio coding algorithm that encodes full-band audio signals (up to 20 kHz) using an end-to-end training criterion. By predefining several dyadic subband signals as training targets, we progressively encode input audio signals in each stage such that deeper stages of the network encode the residual error terms from the previous encoding stage. Our proposed audio codec successfully decodes full-band audio signals by using an effective multi-stage vector quantization scheme to represent key encoding features extracted in the latent space. Subjective listening tests show that the decoded outputs of the proposed audio codec achieve almost transparent quality at an average bitrate of 132 kbps.

Index Terms— Audio codec, deep neural network, subband coding, cascaded coding, end-to-end model

1. INTRODUCTION

The past several decades have seen the rapid evolution of audio compression techniques for saving the information in audio signals for various applications. Although detailed specifications depend on the application, most audio coding algorithms are able to obtain high quality at around a 10-to-1 compression ratio. Most conventional audio codecs analyze the frequency components of a given audio frame using a linear transformation, such as a modified discrete cosine transform (MDCT) [1–3]. Based on time-frequency analysis, the codecs assign a different number of bits to each frequency component depending on its perceptual importance as measured by a pre-defined psychoacoustic model.

Recently, new kinds of audio codecs based on deep neural networks (DNNs) have been proposed in an attempt to further improve the performance of audio compression. Unlike conventional audio codecs that rely on hand-crafted time-frequency analysis, deep learning-based models aim to find compact latent representations or embeddings that are appropriate for faithfully reconstructing the input signal. The whole codec pipeline—encoding, quantization and decoding—is constructed by a DNN architecture, and the model parameters are trained under the constraint of the rate-distortion trade-off. This end-to-end approach is appealing, as it means that the analysis and synthesis processes are not limited to a linear system category. For example, [4] proposed an convolutional neural network (CNN) based autoencoder structure where the embedding after encoding is quantized with a small number of bits. The network architecture was improved in [5] to allow it to encode full-band audio signals with the help of a perceptual loss function [6] motivated by the psychoacoustic model of [1]. [7] introduced a discriminator network for generative adversarial training and a multi-stage vector quantization scheme to achieve higher coding efficiency than the conventional Opus codec [3] in extremely low bit-rate conditions. Nevertheless, due to structural limitations of the conventional neural audio codec, which used a simple encoder-decoder structure, this method was not able to obtain transparent quality or faithfully reproduce full-band signals.

In this paper, we propose a new framework for end-to-end neural audio compression which can deal with full-band audio signals while preserving transparent audio quality. Our proposed architecture, which we call a progressive neural audio codec, consists of a cascade of multiple coding stages. The cascaded structure is effective because it divides the difficult task of coding full-band signals into relatively easier sub-tasks that each target a narrower bandwidth. Each stage encodes the lower part of the divided frequency band, and the subsequent stage encodes a wider subband signal as well as residuals from all of the preceding encoding stages. Therefore, the target bandwidth at each stage progressively increases, but each encoding stage is able to concentrate on a specific frequency band, as most of the information in the lower bands is encoded in the previous stages. To train the aforementioned progressive architecture, we set a subband signal as a guide for training each stage. We also utilize a blended reference technique to stabilize training, which corrects imperfect stage outputs with the guide signals.

The main contributions of our paper are as follows: 1) We propose a novel progressive multi-stage codec which incorporates subband and cascaded coding techniques. This means that each stage is responsible for encoding specific subbands and combining the residuals from all of the preceding stages. 2) We introduce an error correction training method, which we call blended training, that plays a crucial role in successfully training the cascaded architecture.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

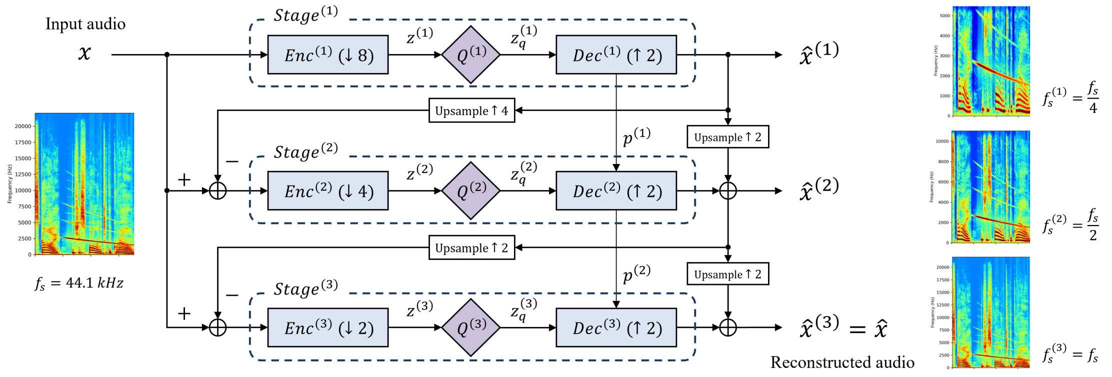

Fig. 1. Architecture of the proposed progressive neural audio codec.

2. RELATED WORKS

2.1. End-to-end neural audio coding

End-to-end neural audio codecs [4–7] are composed of three main components—encoder, quantizer, and decoder. A CNN with nonlinear activation functions is normally used for the encoder to transform the input time-domain waveform into embedding vectors. After encoding, the elements of the embedding vectors are quantized by a scalar or vector quantizer for which centroids are jointly learned during the training process. The quantized embedding vectors are reconstructed back to a time-domain waveform through the decoder, which also consists of a CNN and nonlinear activation functions.

However, we cannot train such audio codecs end-to-end because the quantization function for the embedding vectors is not differentiable. One way to solve this problem is to use a softmax quantization method [8], which obtains a learnable set of centroids using a softmax function with a low temperature, as used in [4–6]. Another way is to use a vector quantized variational autoencoder (VQ-VAE) [9], which directly copies the gradients computed with the quantized vector to the unquantized vector without any approximation [7].

2.2. Subband and cascaded coding

Since the power distribution and perceptual importance vary over frequencies for different types of audio signals, conventional signal processing-based audio codecs utilize subband analysis to improve the quality of decoded signals and coding efficiency. The concept of subband analysis and synthesis has previously been applied to DNN-based neural vocoders [10], but to the best of our knowledge, its usefulness for end-to-end neural audio coding has not yet been studied. [5] introduced a cascaded coding technique [11] where the quantization error is repeatedly encoded through cascaded encoder-decoder modules.

In this work, we integrate the aforementioned ideas into a unified model, motivated by the structure of StackGAN [12]. StackGAN is a model for high-quality image generation that divides the generation process into two-stage sub-tasks; the first stage generates a medium-quality image, and the second sub-task generates the final high-quality image based on the output of the first sub-task. Our proposed model inherits the methodology of subband coding, but the processing of each subband is done sequentially in such a way that the previously decoded subband signal is utilized by the following stages. By using this structure, we not only differentiate the target band at each coding stage, but also enable the network to fix any errors from previous stages for better reconstruction.

3. PROGRESSIVE NEURAL AUDIO CODEC

In this section, we propose a multi-stage progressive coding architecture that incorporates subband and cascaded coding techniques. We explain the role of each stage, the connections between subsequent stages, and the structure of each stage. Then, we introduce the objective functions and a blended training method to improve the training stability of the proposed neural audio codec.

3.1. Multi-stage progressive coding

Fig. 1 illustrates the proposed architecture. It consists of three coding stages, each with its own output signal. The output of each stage is designed to have a lower sampling rate than its input, except for the final stage. The purpose of this design is to structurally hinder the model from reconstructing the whole frequency band at once, so that the encoder, decoder, and VQ codebook pair can only concentrate on patterns in specific frequency bands.

However, unlike conventional subband analysis approaches, we do not explicitly limit the bandwidth of each stage's input by means of band-pass filtering. Instead, we subtract the up-sampled previous stage output from the original input signal before feeding it to the next stage. The output of each stage is then added to the up-sampled output of the previous stage. To match the sampling rate between the stages, we perform sub-sampling, as shown in Fig. 1. We apply low-pass filtering after up-sampling to remove any artifacts resulting from the up-sampling. Here, we note that the target bandwidth of each stage can be arbitrarily determined by changing the subsampling ratio or the number of stages. Furthermore, inspired by [12], we provide the hidden representation of the previous stage's decoder network to the current one ($p^{(1)}$ and $p^{(2)}$ in Fig. 1) in order to improve the quality of the decoded signal. In addition to the signal flow control, we explain a training method to encourage the band splitting in Section 3.2.

Next, we explain the model components more in detail.
Encoder. The encoder network $Enc^{(i)}$ consists of CNN layers with residual connections as in [4,5] to extract a high-level representation appropriate for the subsequent vector quantization. Down-sampling is performed with strided convolutional layers depending on the desired down-sampling ratio. After encoding, we obtain a sequence of vectors whose length is the number of input samples divided by the down-sampling ratio.

Vector quantizer. The vector quantizer $Q^{(i)}$ reduces the number of bits needed to represent the embedding vectors by selecting code vectors that are the closest to each embedding vector. We adopt the scheme of VQ-VAE for the quantization of the vectors, which is explained in Section 2.1.

Decoder. The decoder network $Dec^{(i)}$ also consists of CNN layers with residual connections to reconstruct the waveform from the embedding vectors. Up-sampling is performed inside the network with subpixel convolution layers [13], where the ratio is usually made to be smaller than the down-sampling ratio of the encoder to reduce the bandwidth. The output of the last coding stage added with the up-sampled previous output is the final decoding result.

3.2. Network training

Training loss. To train the proposed progressive coding architecture, we use mean squared error (MSE), adversarial loss, feature matching loss, and VQ codebook loss as the objective functions. The losses are computed on all of the intermediate stage outputs $\hat{x}^{(i)}, \forall i$, rather than only considering the final decoding result. Down-sampled input signals $\tilde{x}^{(i)}$ are given as reference signals to each stage to be used as guides during training.

MSE loss is used to measure distortion at the sample level. However, this loss function is biased towards differences in high energy components, such as lower frequency bands. Thus, this causes reconstruction performance to degrade in low energy components, which are usually higher frequency components. To alleviate this problem, we also use least-square GAN-based adversarial loss [14] along with the feature matching loss. Among various types of discriminators, we use the multi-period discriminators (MPD) originally proposed in HiFi-GAN [15] because they can reliably capture the complicated harmonic structure of audio signals.

Blended training for progressive network. As our proposed model is a cascade of multiple coding stages, later stages are highly affected by any errors in the preceding stages. Although our model is designed to remedy these sorts of errors through the cascaded coding, they still make the optimization difficult and unstable at the early training steps. More specifically, an ill-posed previous stage output may distort both encoder input and decoder outputs due to the connection between the stages, disrupting the stage-wise optimization.

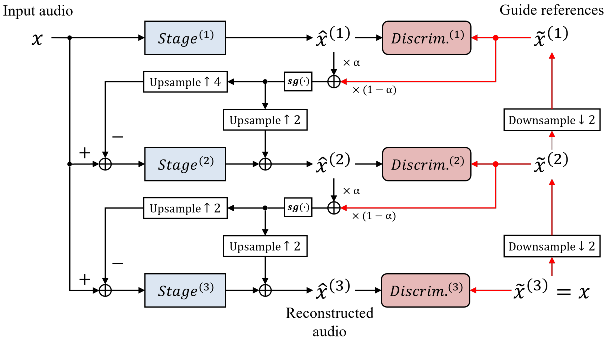

Fig. 2. Training scheme with blended training technique.

To facilitate the training at each stage, we use a blended training technique inspired by the works of [16], [17] and [18]. As shown in Fig. 2, we take a weighted sum between each stage's output and the corresponding guide reference using weights α and $1-\alpha$, respectively. The constant α is set to zero at the beginning of training and gradually increases to one as training progresses. In addition, we block the gradient flows between stages with the stop-gradient operator $\mathrm{sg}(\cdot)$ [9]. By combining these two methods, each stage starts off by learning to reconstruct its own specific subband using the previous stage's reference, and is only gradually required to deal with the errors from the preceding stages.

4. EXPERIMENTS

4.1. Training details

We used BBC sound effect data[1] for training the proposed neural audio codec. The dataset contains various types of CD-quality sound effects from nature recorded at a 44.1 kHz sampling rate. Here, there is a different amount of audio available for each category. Therefore, we set a limit to the maximum amount of data from each category to equalize data across categories, obtaining a total of 5.5 hours of training data. For training input, an audio frame of length 1,024 samples is used.

We used the AdamW optimizer [19] with $\beta_1 = 0.8, \beta_2 = 0.99$ and weight decay $\lambda = 0.01$, referring to [15]. The learning rate was set to a constant value of 10^{-4} throughout the training. For blended training, α was set to zero during the first 20% of training. From then until 60% of the training progression, α was linearly increased to one. Afterwards, α was fixed to one until training ended.

[1] https://sound-effects.bbcrewind.co.uk/

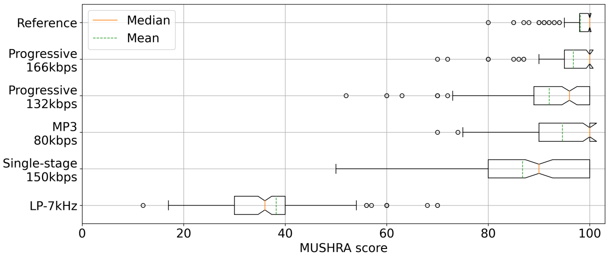

Fig. 3. MUSHRA test results.

4.2. Evaluation methods

For evaluation, we used 44 single-channel audio samples consisting of speech, music, and mixed signals, including test items used for the USAC (unified speech and audio coding) verification test. The analysis frame and shift length of the proposed encoding model are 2,048 and 1,024 samples, respectively. We assume that the middle 1,024 samples correspond to the current time window, with 512 samples corresponding to the past, and 512 samples to the future (look-ahead).

We performed MUSHRA [20] listening tests to evaluate the subjective sound quality. MUSHRA tests were conducted on two versions of our proposed model with different bitrates, which only differed in the maximum available number of code vectors. The higher bitrate model was assigned 32 codes (5 bits) at each stage, and the lower bitrate model was assigned 16 (4 bits). We also used a single-stage model with only one encoding-decoding stage (with increased capacity for fairness) for comparison to verify the benefits of the proposed multi-stage structure. This model was assigned a large number of 1,024 codes (10 bits).

To compare the quality of our model with an existing audio codec, we also included the MP3 codec [1] operating at 80 kbps bitrate on mono signal into the comparison group. In accordance with the guidelines, a hidden reference and a low-pass anchor at 7 kHz were included along with the aforementioned systems. Additionally, we also evaluated objective scores (SDR and ViSQOL [21]) of the different methods.

4.3. Results

Fig. 3 summarizes the MUSHRA test results. We selected a subset of 15 samples from the full test set to reduce listener fatigue. 12 audio experts participated in our MUSHRA test, of which 3 were disqualified because of their low scores in determining hidden references.

We found that with a higher bitrate, our proposed model achieves almost the same score as the reference. The lower bitrate version of our model performs on par with the MP3 at 80 kbps, which is well-known for transparent quality. It is worth noting that our model's quality was achieved without any knowledge on human perception. In contrast, the single-stage model shows significantly lower scores, even at a higher bitrate than our proposed method. These results are also reflected in the objective measurements of SDR and ViSQOL. The scores in Table 1 show similar patterns with the MUSHRA test results, where our proposed model at both bitrates outperformed the single-stage model in terms of both faithful reconstruction and bitrate efficiency.

Table 1. Objective scores.

System	Bitrate (kbps)	ViSQOL	SDR (dB)
Progressive	166 ± 3	4.64 ± 0.02	38.27 ± 1.40
	132 ± 3	4.54 ± 0.04	31.75 ± 1.11
Single-stage	150 ± 2	4.19 ± 0.08	26.73 ± 0.77
MP3	80	4.63 ± 0.01	24.83 ± 1.38

Table 2. Stage-wise bitrate analysis.

System	Bitrate (kbps)		
	Stage 1	Stage 2	Stage 3
Progressive	23.7 ± 0.5	50.1 ± 0.7	92.1 ± 2.3
	18.6 ± 0.4	40.4 ± 0.6	72.6 ± 2.3
Single-stage	-	-	150.0 ± 2.1

We calculated the bitrates for each stage; these values are shown in Table 2. The bitrates were calculated based on the rate of code vectors and entropy of the codebooks, taking entropy coding into consideration. We see that the third stage, which is designed to encode the highest frequency band, has the largest bitrate. These high-frequency components have low signal energy and are relatively less important for human perception, and are thus usually treated with very few bits or with a parametric coding method in conventional approaches. Therefore, we expect that the coding efficiency of our method can be further improved by adjusting the bit allocation based on knowledge of psychoacoustics.

5. CONCLUSION

We introduced a progressive multi-stage architecture for neural audio coding that is able to effectively encode full-band audio signals. Our model uses a cascade of subband encoding stages to progressively encode audio signals in such a way that each stage can focus on learning the coding of a specific frequency band. Experimental results demonstrated that our method is able to achieve an almost transparent audio quality comparable to or better than the MP3 codec without psychoacoustic knowledge. We believe that the proposed structure can be the baseline for a new series of more advanced deep learning-based audio codecs.

6. ACKNOWLEDGEMENTS

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [21ZH1200, The research of the basic media contents technologies]

7. REFERENCES

[1] International Organization for Standardization/International Electrotechnical Commission et al., “Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s,” *ISO/IEC 11172*, 1993.

[2] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, “Iso/iec mpeg-2 advanced audio coding,” *Journal of the Audio engineering society*, vol. 45, no. 10, pp. 789–814, 1997.

[3] J. M. Valin, K. Vos, and T. B. Terriberry, “Definition of the opus audio codec,” *RFC*, vol. 6716, pp. 1–326, 2012.

[4] S. Kankanahalli, “End-to-end optimized speech coding with deep neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[5] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, “Cascaded cross-module residual learning towards lightweight end-to-end speech coding,” in *Interspeech*, 2019.

[6] K. Zhen, M. S. Lee, J. Sung, S. Beack, and M. Kim, “Psychoacoustic calibration of loss functions for efficient end-to-end neural audio coding,” *IEEE Signal Process Letters*, vol. 27, pp. 2159–2163, 2020.

[7] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *CoRR*, vol. abs/2107.03312, 2021.

[8] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in *30th Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[9] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *30th Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[10] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, “Multi-band melgan: Faster waveform generation for high-quality text-to-speech,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[11] G. Schuller, B. Yu, and D. Huang, “Lossless coding of audio signals using cascaded prediction,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[12] H. Zhang, T. Xu, and H. Li, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[16] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[17] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in *28th Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” in *6th International Conference on Learning Representations (ICLR)*, 2018.

[19] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations (ICLR)*, 2019.

[20] ITU-R, “Bs. 1534-1, method for the subjective assessment of intermediate quality levels of coding systems (mushra),” *International Telecommunication Union*, 2003.

[21] 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,” in *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020.