ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096009

A PERCEPTUAL NEURAL AUDIO CODER WITH A MEAN-SCALE HYPERPRIOR

Joon Byun[†]*, Seungmin Shin*[†]*, Youngcheol Park*[†]*, Jongmo Sung*[‡]*, Seungkwon Beack*[‡]

[†] Intelligent Signal Processing Lab., Yonsei University, Wonju, Korea
[‡]Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea

ABSTRACT

This paper proposes an end-to-end neural audio coder based on a mean-scale hyperprior model together with a perceptual optimization using a psychoacoustic model (PAM)-based loss function. The proposed coder estimates the mean and scale hyperpriors using a sub-network after assuming that the probability distribution of latent samples is Gaussian. The main network is an autoencoder based on Resnet-type gated linear units (ResGLUs), each comprising a generalized divisive normalization (GDN) layer. We train both networks to optimize perceptual attributes estimated using a multi-time-scale scheme to obtain high perceptual quality. Experimental results show that the proposed model accurately predicts the mean and scale hyperpriors. Also, it obtains consistently higher audio quality than the commercial MP3 audio coder at all bitrates.

Index Terms— Neural Audio coder, Hyperprior, PAM, Perceptual Loss Function

1. INTRODUCTION

Audio coding aims to represent various audio cues using discrete data with minimal entropy, but it has to achieve excellent perceptual quality simultaneously. For decades, tremendous studies have been conducted to achieve this goal. Traditional audio codecs are based on linear transforms in the time-frequency domain, such as modified discrete cosine transform (MDCT) [1, 2], and exploit a psychoacoustic model (PAM) to make the quantization noise less perceptible. In recent years, linear transforms were replaced with learning-based ones using deep neural networks (DNNs) that can find meaningful latent representations of the input audio. So far, the DNN-based neural audio coders have shown great potential for achieving better sound quality than conventional transform-based methods, and efforts for improving sound quality and coding efficiency are still underway. Perceptual loss functions were often used when training the DNN-based neural audio and speech coders to improve perceptual audio quality [3–6]. In particular, PAM-based loss functions were incorporated into the rate-distortion (R-D) optimization problem of the neural audio coder. More recently, a method of estimating PAM losses both frame-wise and subframe-wise [6] so that it was achieved more accurate control of local quantization noises within a frame. This scheme was effective for speech coding especially at low bitrates.

Also, there have been efforts to improve the coding efficiency [7–13]. Some utilized additional side information using either sub-network or multi-stage learning method [8–13]. Multi-stage schemes have been tried to integrate into end-to-end neural waveform coder [11–13]. Generative models with multi-stage vector quantizer were also proposed [13] to achieve high coding efficiency in extremely low bit-rate conditions. In image and video coding, a method of providing main network with hyperpriors of the latent representations as side information was introduced [8], which results in superior image compression compared to earlier learning-based methods. In this method, the arithmetic coding was used based on an entropy model which is the key for obtaining an image-independent spatially adaptive compression model.

In this paper, we propose a perceptual neural audio coder with mean-scale hyperpriors. We use a sub-network to estimate the hyperpriors after assuming that latent samples follow Gaussian distribution. The goal is to obtain a frame-dependent entropy model that can increase the efficiency of the arithmetic coding. Also, we utilized the PAM-based losses in our previous work for speech coder [6], measured in different time-scales. Thus, the proposed model is trained to focus more on the perceptually relevant audio cues. Through tests, we confirm the superior performance of the proposed neural audio coder relative to the conventional MP3 audio coder.

2. RELATED WORKS

2.1. End-to-end compression with hyperpriors

In speech and audio coding, sub-networks were used to enhance the coding efficiency by providing side information, such as linear predictive coding (LPC), or by implementing residual coding [11–13]. For image coding, a trainable sub-network incorporated to estimate hyperpriors was proposed [9, 10]. With an assumption that each channel data at the bottleneck is a Gaussian with independent and identical distri-

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [22ZH1200, The research of the basic media·contents technologies]

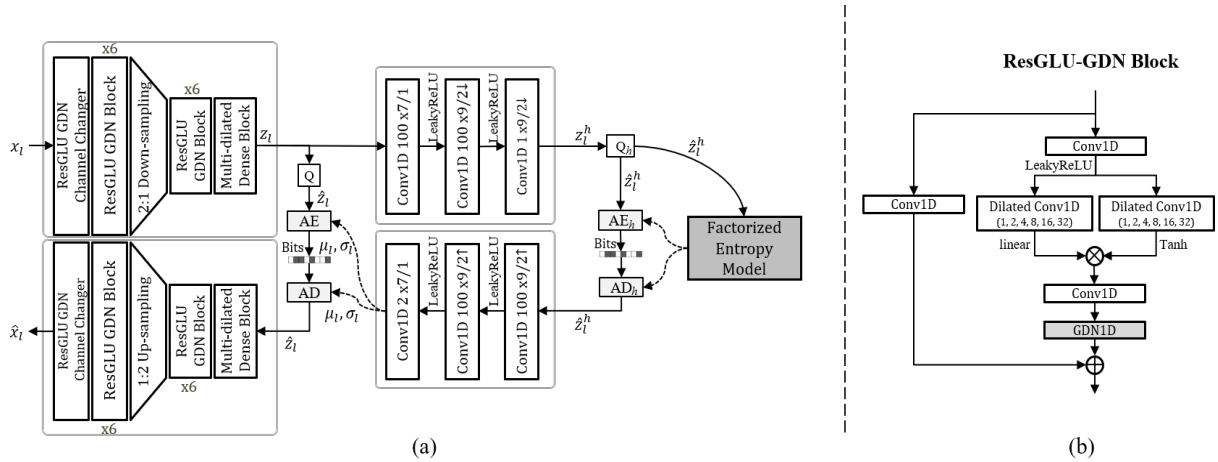


Fig. 1. Architecture of the proposed neural audio coder: (a) the main and sub-network interconnection and (b) the ResGLU-GDN block used in the main network.

bution (iid), a sub-network was trained to estimate the hyperpriors: mean and scale. The methods in [9, 10] also used the generalized divisive normalization (GDN) to eliminate the correlation between data and effectively transform the data of various distributions mixed in an image into a Gaussian form [7]. Compared to earlier methods, it showed excellent image compression performance when tested on an end-to-end variational autoencoder. Inspired by this, this paper proposes an autoencoder-type end-to-end neural audio coder employing a sub-network estimating hyperpriors of the Gaussian bottleneck samples.

2.2. PAM-based loss functions

Since the mean-square error (MSE) alone can not faithfully reflect human auditory perception, various forms of perceptual loss terms were proposed for training the network. Perceptual distortion was often measured using the logarithmic noise-to-mask ratio (NMR) calculated by PAM-1 [3–6].

The measure of perceptual loss can be performed frame-wise using the entire samples in a frame. However, it might be too coarse a time resolution to control the local quantization noises occurring within the frame, especially when the frame length becomes longer for higher coding efficiency. In [6], a local perceptual loss was calculated subframe-wise after dividing a frame into several overlapped subframes to remedy this. This approach was proven effective when applied to an end-to-end neural speech coder, especially at low bitrates [6].

3. PERCEPTUAL AUDIO CODER BASED ON MEAN-SCALE HYPERPRIORS

3.1. Architecture of the main and sub-networks

This paper proposes a time-domain end-to-end neural audio coder combined with a hyperprior model. A block diagram of the proposed audio coder is shown in Fig.1, which is based on an autoencoder consisting of Resnet-type gated linear units (ResGLUs) [14–16]. The ResGLU block has six layers of 1D convolution with dilation factors 1, 2, 4, 8, 16, and 32, respectively. The kernel size was 9, and the number of channels is reduced from 100 to 30 at the bottleneck and restored to 100 at the input to the decoding path. Each ResGLU block includes a 1D GDN layer, as shown in Fig. 1(b), acting as a non-linear activator at the end of GLU. The channel changer in Fig. 1(a) is also a ResGLU layer, transforming the single-channel input to multi-channel features or vice versa. A down-sampling is conducted by striding during the 1D convolution, and a sub-pixel convolutional layer [17] is adopted for up-sampling. To further capture the long-term dependencies of the input features, a multi-dilated convolution layer is used before and after the quantization block (Q) of the main autoencoder that is known to show superior performance to the LSTM [18]. The main autoencoder transforms the input vector $\mathbf{x}_l \in \mathbb{R}^T$ to the bottleneck data vector $\mathbf{z}_l \in \mathbb{R}^{T/2}$ having half dimension, where l denotes the frame index. The input $\mathbf{x}_l$ has a size of $512(= T)$ samples and is framed using a window with half-sines of the length of 32 samples at both ends. The output $\hat{\mathbf{x}}_l$ is synthesized via the overlap-add procedure using the same window.

For modeling the mean-scale hyperprior of the bottleneck data, an autoencoder-type sub-network is added to the bottleneck, as in [12]. The hyperprior sub-network comprises simple 1D convolutional layers with kernel sizes of $\{7, 9, 9\}$ and strides of $\{1, 2, 2\}$. The number of channels expands from 1 to 100 at the first convolutional layer and reduces to 1 at the bottleneck. After the quantization, the latent samples are finally arithmetically encoded to form a compressed bitstream. The arithmetic encoder (AE) and decoder (AD) are cascaded and share the same entropy model. In the training phase, the quantization block (Q) is replaced with the uniform noise model used in a neural speech coder [6]. Since this model replaces the uniform quantization with additive stochastic noise [7], it is intuitive and, furthermore, provides stable convergence at any bitrates without the additional penalty. The main au-

toencoder and sub-network have symmetric-shaped encoding and decoding modules, and the decoding module in the sub-network has two output channels producing means and scales, respectively. As a result, the main autoencoder and hyperprior sub-network have $2.35M$ and $0.18M$ trainable network parameters, respectively.

3.2. Hyperprior modeling

Setting a proper entropy model for the discrete latents strongly affects the coding efficiency and, thus, the output audio quality. Since the probability distribution might differ from one frame to another, a frame-dependent conditional probability of the bottleneck data needs to be estimated. The overall procedure for hyperprior modeling is similar to those in image compression [9, 10]. As shown in Fig. 1(a), the hyperprior sub-network consisting of 1D convolutional layers accepts the latent vector $\mathbf{z}_l$ and passes it to the hyper-encoder estimating hyper-representation $\mathbf{z}_l^h$. $\mathbf{z}_l^h$ is quantized (Q_h) to discrete symbols $\hat{\mathbf{z}}_l^h$, and then arithmetically encoded. The probability distribution of $\hat{\mathbf{z}}_l^h$ is estimated using the fully factorized entropy model and shared by the arithmetic encoder (AE_h) and decoder (AD_h). The decoded discrete symbols $\hat{\mathbf{z}}_l^h$ are passed to the hyper-decoder predicting means (μ_l) and scales (σ_l) of the latent samples in $\hat{\mathbf{z}}_l$, which are assumed to have a Gaussian distribution.

Since the main quantizer Q rounds the latent vector $\mathbf{z}_l$ to $\hat{\mathbf{z}}_l$, the likelihood of $\hat{\mathbf{z}}_l$ is calculated as

$$p(\hat{\mathbf{z}}_l|\hat{\mathbf{z}}_l^h) = \mathcal{N}\left(\frac{\hat{\mathbf{z}}_l - \mu_l + \frac{1}{2}}{\sigma_l}\right) - \mathcal{N}\left(\frac{\hat{\mathbf{z}}_l - \mu_l - \frac{1}{2}}{\sigma_l}\right), \quad (1)$$

where $\mathcal{N}$ denotes the cumulative distribution function of a standard normal Gaussian distribution.

We ran a test to validate the proposed hyperprior modeling. After training the main and sub-networks using a dataset with a $32kHz$ sampling rate, the effect of transforming using the estimated hyperpriors was measured. Results are shown in Fig. 2. Fig. 2(a) shows the estimated mean-scale hyperpriors along with the latent samples, which shows that the hyperpriors closely track the shape of the latent samples. Probability distributions before and after transforming with hyperpriors are demonstrated in Fig. 2(b), from which we can see that the latent samples are accurately transformed to a Gaussian distribution. To confirm the consistency, probability distributions of 100 frames are superimposed in Fig. 2(b).

We also measured bit-per-samples used by the main and sub-networks after training. Results are shown in Fig. 3. As the bitrate increases, the sub-network consumes more bits for the hyperprior modeling, but the ratio becomes lower. At $48kbps$, the sub-network uses 16% of the total bits, but the portion reduces to 14% at $64kbps$.

3.3. Loss function

The R-D optimization problem for training the proposed neural coder can be rewritten as a combination of several loss

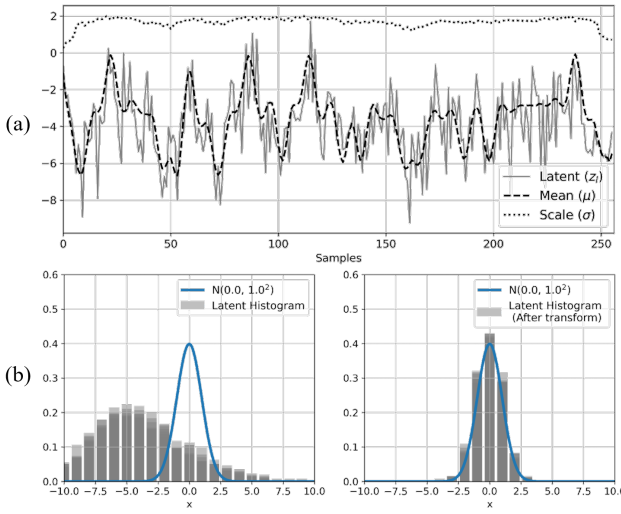


Fig. 2. Result of hyperprior modeling: (a) the mean and scale obtained for a given audio frame, (b) the probability distribution of the latent samples before (left) and after (right) transforming by the estimated hyperpriors.

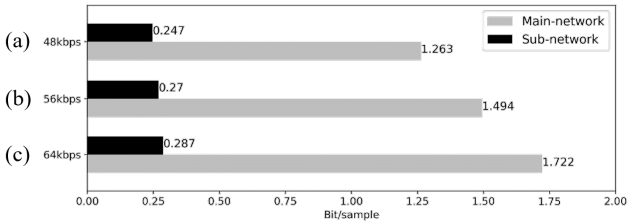


Fig. 3. Bit-per-samples used by the main and sub-networks at (a) $48kbps$, (b) $56kbps$, and (c) $64kbps$.

terms:

$$\mathcal{O} = \mathcal{L}_e + \lambda_1 \mathcal{L}_{MSE} + \lambda_2 \mathcal{L}_p^G + \lambda_3 \mathcal{L}_p^L, \quad (2)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the blending factors. Here, the signal distortion is calculated as $\mathcal{L}_{MSE} = \|\hat{\mathbf{x}}_l - \mathbf{x}_l\|_2^2$. $\mathcal{L}_p^G$ and $\mathcal{L}_p^L$ denote perceptual distortions measured in different time scales. $\mathcal{L}_p^G$ is a global perceptual loss calculated using a frame of 512 samples, and $\mathcal{L}_p^L$ is a local one calculated by averaging the losses obtained from seven subframes of 128 samples overlapped by 50%. More detailed procedures for calculating these perceptual loss terms can be found in [6]. The difference is the number of Mel filter banks transforming the obtained parameters. Considering the input audio signal's bandwidth, we use four Mel filterbanks jointly, having 16, 32, 64, and 256 bands.

L_e in Eq. (2) is the rate term, estimated using discrete representations of the latent vector. The rate term can be estimated as $\mathcal{L}_e = \sum$ - $\log_2 p(\hat{\mathbf{z}}_l|\hat{\mathbf{z}}_l^h) + \sum$ - $\log_2 p(\hat{\mathbf{z}}_l^h)$. The average bitrate is also estimated using L_e as $\frac{fs}{T-32} \times \mathcal{L}_e$ where f_s is the sampling frequency and $T(=512)$ is the input frame size. The estimated bitrate is exactly the same as the actual bitrate obtained using the arithmetic encoder.

4. EXPERIMENTAL RESULTS

4.1. Test setup

For experiments, we constructed an audio dataset using $1,000$ commercial movie clips sampled at 32kHz. Among them, 700 and 200 clips were used for training and validation, respectively. The other 100 clips were used for the test. All clips were normalized by standard deviations. The mini-batch size was 128 and the learning rate was adjusted using cosine annealing from an initial value 0.0001 to a final value 0.00005. The model was trained in PyTorch using Adam optimizer. In consideration for the dynamic range of each loss functions, blending factors were initialized as $\lambda_1 = 60$, $\lambda_2 = 0.003$ and $\lambda_3 = 0.0005$. The performance of the proposed model was compared with the commercial MP3 audio coder included in Adobe Audition, licensed from Fraunhofer IIS and Thomson. Our previous neural audio coder proposed in [5] was also included for the performance comparison to assess the performance improvement obtained using the multi-time-scale PAM loss and the hyperprior modeling. DNN-based neural audio coders were trained for at least over 200 epochs.

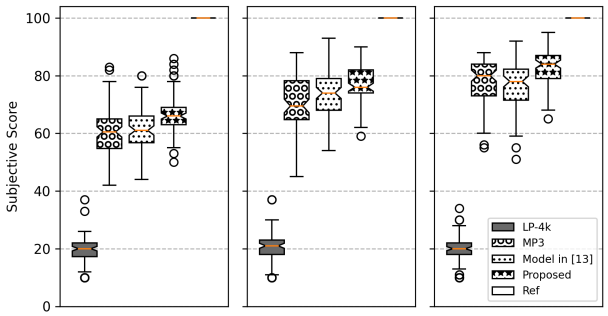
4.2. Objective quality measure

We first measured the output audio quality of each coder in terms of SNR and 2f-model in [19]. The operating bitrates were 48, 56, and $64kbps$. The 2f-model is a newly proposed metric for predicting the perceived audio quality and is known to have a high correlation with the actual subjective tests. We trained the proposed model using our audio dataset. But the test items were selected from three different sources: our database, unified speech and audio coding (USAC) test items [20], and the BBC sound effect dataset [21]. We selected twenty test items from our database, among those not used for training. USAC test items consisted of three speech, three music, and four speech and music mixed items. BBC sound effect dataset consists of various types of CD-quality sound effects from nature, sampled at $44.1kHz$. There are over $15,000$ sound clips in 23 categories, and 20 clips were randomly selected across the categories. All test items were downsampled to $32kHz$ before encoding.

The results are summarized in Table 1, where the mean and standard deviation are shown together. Results show that regardless of the test sets, the proposed model obtains higher SNRs and 2f scores than MP3 and our previous model [5]. The 2f scores, in particular, shows consistently higher mean values and smaller standard deviations for all test items from different sources. On the other hand, our previous model generally obtained lower 2f scores than the MP3 coder at all bitrates. The proposed model obtained higher SNRs, mainly due to the accurate estimation of the mean-scale hyperprior of the latent samples. More importantly, we may mention that the proposed model reaches a certain level of generalization since it shows a similar performance margin relative to the MP3 coder for all test items from three different sources.

Table 1. Objective evaluation results for test items (a) selected from the training dataset (but unseen), (b) USAC test items, and (c) selected from BBC sound sound effect dataset.

	Bitrate	**SNR (dB)** mean(std)			**2f-model** mean(std)		
	(kbps)	MP3	Model in [5]	Proposed	MP3	Model in [5]	Proposed
(a)	~ 48	17.65(2.08)	19.15(1.45)	**20.46(2.34)**	58.71(4.13)	59.92(2.03)	**66.51(2.76)**
	~ 56	18.94(2.03)	20.47(1.97)	**22.06(2.76)**	67.30(3.90)	64.36(2.48)	**74.19(3.03)**
	~ 64	20.21(1.99)	21.80(2.58)	**23.10(3.18)**	75.35(3.62)	70.68(2.91)	**78.08(3.07)**
(b)	~ 48	20.97(3.66)	19.72(2.07)	**22.18(3.62)**	59.38(9.08)	57.12(3.62)	**63.33(5.89)**
	~ 56	22.27(3.58)	21.48(2.63)	**23.95(4.15)**	66.69(9.35)	62.04(4.87)	**70.61(6.66)**
	~ 64	23.41(3.56)	22.89(3.45)	**25.29(4.64)**	73.19(9.39)	67.19(6.04)	**75.15(7.48)**
(c)	~ 48	18.00(5.99)	18.15(4.32)	**19.86(5.60)**	60.37(13.01)	58.33(5.89)	**65.76(7.95)**
	~ 56	19.43(5.99)	19.43(4.89)	**21.72(6.29)**	67.76(13.10)	63.66(7.11)	**72.11(8.40)**
	~ 64	20.87(5.79)	20.91(5.70)	**22.68(6.86)**	75.09(12.23)	69.22(7.81)	**77.06(10.51)**



(a) $48kbps$ (b) $56kbps$ (c) $64kbps$

Fig. 4. MUSHRA test results using USAC test items.

4.3. Subjective test results

A Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) [22] test was conducted using USAC items showing the lowest 2f scores in Table 1. After training the proposed model on our audio dataset, we encoded and decoded USAC items in the trained model. Nine experienced subjects participated in the test. Test results are shown in Fig. 4. At all bitrates, the proposed model's audio quality was significantly better than the MP3 coder and our previous model in [5]. In addition, the proposed model outperformed our previous model, proving that the hyperprior modeling and the multi-time-scale PAM-based loss contributed to improving the coding efficiency and audio quality. Test samples are available online[1].

5. CONCLUSIONS

We proposed an end-to-end neural audio coder with mean-scale hyperpriors. By training the proposed coder using a multi-time-scale PAM-based loss function, we could achieve consistently higher audio quality than the commercial MP3 audio coder. Also, we could confirm that the proposed model accurately predicted the mean and scale hyperpriors.

[1] https://sites.google.com/view/isplab-yonsei/research/listening-icassp2023

6. 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," in *ISO/IEC 11172*, 1993.

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

[3] 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," in *IEEE Signal Processing Letters*, 2020, vol. 27, pp. 2159–2163.

[4] J. Byun, S. Shin, Y. Park, J. Sung, and S. Beack, "Development of a psychoacoustic loss function for the deep neural network (DNN)-based speech coder," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021, pp. 1694–1698.

[5] S. Shin, J. Byun, Y. Park, J. Sung, and S. Beack, "Deep neural network (DNN) audio coder using a perceptually improved training method," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 871–875.

[6] J. Byun, S. Shin, Y. Park, J. Sung, and S. Beack, "Optimization of deep neural network (DNN) speech coder using a multi time scale perceptual loss function," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2022, pp. 4411–4415.

[7] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations (ICLR)*, 2017.

[8] J. Ballé, V. Laparra, and E. P. Simoncelli, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations (ICLR)*, 2018.

[9] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in neural information processing systems (NeurIPS)*, 2018.

[10] J. Lee, S. Cho, and S. K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *International Conference on Learning Representations (ICLR)*, 2019.

[11] Kai Zhen, Mi Suk Lee, Jongmo Sung, Seungkwon Beack, and Minje Kim, "Efficient and scalable neural residual waveform coding with collaborative quantization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 361–365.

[12] C. Lee, H. Lim, J. Lee, I. Jang, and H. G. Kang, "Progressive multi-stage neural audio coding with guided references," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 876–880.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning Research (PMLR)*, 2017, pp. 933–941.

[16] K. Tan, J. Chen, and D. Wang, "Gated residual networks with dilated convolutions for monaural speech enhancement," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 1, pp. 189–198, 2018.

[17] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[18] N. Takahashi and Y. Mitsufuji, "Densely connected multidilated convolutional networks for dense prediction tasks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 993–1002.

[19] M. Torcoli, T. Kastner, and J. Herre, "Objective measures of perceptual audio quality reviewed: An evaluation of their application domain dependence," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021, vol. 29, pp. 1530–1541.

[20] ISO/IEC JTC1/SC29/WG11, *Unified Speech and Audio Coding Verification Test Report*, MPEG2011/N12232, 2011.

[21] "BBC sound effects dataset," https://sound-effects.bbcrewind.co.uk/.

[22] ITU-R Recommendation BS 1534-1, *Method for the subjective assessment of intermediate quality levels of coding systems (MUSHRA)*, 2003.