INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Perceptual Improvement of Deep Neural Network (DNN) Speech Coder Using Parametric and Non-parametric Density Models

Joon Byun[1], *Seungmin Shin*[1], *Jongmo Sung*[2], *Seungkwon Beack*[2], *Youngcheol Park*[1]

[1]Intelligent Signal Processing Lab., Yonsei University, Wonju, Korea
[2]Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea
{nicesin97, bj9407, young00}@yonsei.ac.kr, {jmseong, skbeack}@etri.re.kr

Abstract

This paper proposes a method to improve the perceptual quality of an end-to-end neural speech coder using density models for bottleneck samples. Two parametric and non-parametric approaches are explored for modeling the bottleneck sample density. The first approach utilizes a sub-network to generate mean-scale hyperpriors for bottleneck samples, while the second approach models the bottleneck samples using a separate sub-network without any side information. The whole network, including the sub-network, is trained using PAM-based perceptual losses in different timescales to shape quantization noise below the masking threshold. The proposed method achieves a frame-dependent entropy model that enhances arithmetic coding efficiency while emphasizing perceptually relevant audio cues. Experimental results show that the proposed density model combined with PAM-based losses improves perceptual quality compared to conventional speech coders in both objective and subjective tests.

Index Terms: Neural Speech Coder, Density Model, Hyperprior, PAM, Perceptual Loss Function

1. Introduction

Speech coding is essential for efficient speech transmission and storage, especially in bandwidth-constrained applications. There are two primary categories of speech coding: waveform coding and parametric coding. Waveform coding aims to reproduce the speech waveform with high fidelity, while parametric coding encodes the speech signal's acoustic features, resulting in a lower bitrate. Nevertheless, it is imperative to model the statistical properties of the speech signal to achieve optimal efficiency in speech coding.

In recent years, DNN-based neural speech coders have been actively researched as they have demonstrated the potential to improve coding efficiency and sound quality compared to traditional methods [1–5]. One of the main research areas focused on enhancing perceptual quality by developing various perceptual loss functions. These include PESQ [6, 7], STOI [8], and PAM-based loss functions [9–13], among others. By incorporating such loss functions into the rate-distortion optimization problem, it is possible to train the network to minimize the perceptual distortion present in the reconstructed speech. There have been other efforts to improve the coding efficiency of neural speech and audio coders as well. For example, multi-stage learning methods [14–16] have been proposed to achieve better results, and sub-networks have been used to utilize additional side information [17–19]. Generative models with multi-stage vector quantizers [16] have also been proposed to achieve high coding efficiency in extremely low bitrate conditions.

In image and video coding, methods to utilize the entropy model have been introduced. The entropy model is a statistical model that characterizes the distribution of the latent representation that would be compressed using arithmetic coding. In [17] and [19], trainable end-to-end models incorporating hyperpriors were proposed. Sub-networks producing side information were used for better statistical representations of the given image or video frame, resulting in superior image compression compared to earlier learning-based methods. This approach was adopted for the audio compression [13], and audio quality better than the conventional MP3 coder was achieved.

In this paper, we investigate a method to enhance the perceptual quality of a neural speech coder using density models for bottleneck samples. We explore two approaches for modeling bottleneck sample density: parametric and non-parametric. The parametric approach generates mean-scale hyperpriors for bottleneck samples, leveraging side information from a sub-network. In contrast, the non-parametric approach models bottleneck samples using another sub-network without side information. Both of these methods are evaluated and compared in terms of their ability to improve the perceptual quality of the neural speech coder. The whole network, including the sub-network, is trained using the PAM-based perceptual losses measured in different timescales [12], which is anticipated to shape the quantization noise below the masking threshold. The proposed method aims to achieve a frame-dependent entropy model that can enhance the efficiency of arithmetic coding while also emphasizing perceptually relevant audio cues. Objective and subjective tests are performed to confirm that the proposed density model, combined with PAM-based losses, yields improved perceptual quality in neural speech coding compared to conventional speech coders like AMR-WB [20] and OPUS [21].

2. Backgrounds

2.1. End-to-end compression with entropy model

In neural speech and audio coding, sub-networks have often been utilized to improve coding efficiency by providing side information [14–16]. However, there has been limited exploration of using density models for the input or transformed samples. For image coding, methods of using a trainable sub-network to estimate hyperpriors [18, 19] were proposed. These methods assume that each channel data at the bottleneck is a Gaussian with independent and identical distribution (iid), and train a sub-network to estimate the hyperpriors, which consist of mean [18] or mean-scale parameters [19]. In addition, to eliminate the correlation between data and effectively transform the data of various distributions mixed in an image into a Gaussian form, the generalized divisive normalization (GDN) was used [22]. Compared to earlier methods, this approach showed excellent

10.21437/Interspeech.2023-2305

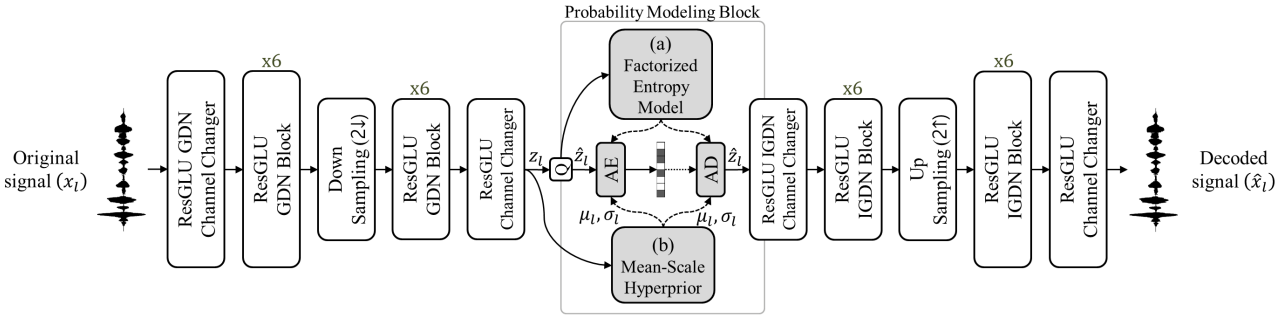

Figure 1: *Architecture of the proposed neural speech coder*

image compression performance when applied to an end-to-end variational autoencoder. Inspired by this sub-network approach in image coding, an autoencoder-type end-to-end neural audio coder was proposed in [13] that employs a sub-network to estimate the probability density function (PDF) of the bottleneck samples. This method aimed to achieve superior audio quality by estimating the PDF of the frame-wise Gaussian bottleneck samples. Thus, it enables us to obtain a frame-dependent entropy model that can increase the efficiency of arithmetic coding.

This paper presents a neural speech coder that utilizes a density model for bottleneck samples. Previous sub-network approaches, as in [13, 17–19], have relied on side information to generate hyperpriors at the decoder. However, the amount of side information may significantly impact speech coder efficiency, particularly at low bitrates. Therefore, we also investigate a non-parametric density model suggested in [17] that does not require any side information. We compare the effects of parametric and non-parametric density models at different bitrates, which can provide insight into how the choice of density model affects the performance of neural speech coders, particularly in bandwidth-constrained scenarios.

2.2. Perceptually optimization

Neural speech coders have been perceptually optimized by designing loss terms that better represent human auditory perception. In particular, the logarithmic noise-to-mask ratio (NMR) calculated through PAM-1 has been successfully used for audio and speech coding [9–12]. This loss term aims to push the quantization noise generated by the neural coder below the masking threshold to make it inaudible. While analyzing all samples in a frame is a possible approach for frame-wise analysis of perceptual loss, it may not effectively control local quantization noises in longer frames designed for higher coding efficiency. A subframe-wise method was introduced in [12] to address this issue. This method partitions frames into overlapped subframes and calculates local perceptual loss. The subframe-wise approach was proven to be effective in controlling quantization noise in an end-to-end neural speech coder, particularly at low bitrates [12]. Therefore, to achieve high perceptual quality, this paper utilizes the PAM-based loss terms developed in [12] to train the proposed speech coder, which includes a sub-network for modeling the probability density of the bottleneck samples.

3. Proposed Neural Speech Coder

3.1. Architecture details

The proposed neural speech coder is an end-to-end system operating in the time domain. It is designed using a variational autoencoder with a trainable quantization module, as depicted in Fig.1. The main component of the proposed speech coder is the Resnet-type gated linear units (ResGLUs) block [12, 13]. The ResGLU block consists of six 1D convolutional layers, each with a different dilation factor ranging from 1 to 32 to cover a wide range of receptive fields. The front-end and back-end channel changers are implemented using a ResGLU layer to transform single-channel input to multi-channel features and vice versa. The kernel size used in all layers is 9, and the number of channels is reduced from 100 to 30 at the bottleneck before being increased back to 100 at the input to the decoding path. To enhance the non-linearity of the ResGLU block, a 1D GDN layer is included as a non-linear activator at the end of each ResGLU block. Down-sampling is achieved through a striding process during 1D convolution, while up-sampling is accomplished through a sub-pixel convolutional layer [23]. The decoding network is a mirrored version of the encoding network and shares the same design.

The main encoding network transforms the input frame $\mathbf{x}_l \in \mathbb{R}^T$ into a latent feature $\mathbf{z}_l \in \mathbb{R}^{T/2}$, where l denotes the frame index. We set $T = 512$, and adjacent frames are overlapped by 32 samples using a half-sine window. The latent feature $\mathbf{z}_l$ is then quantized as $\hat{\mathbf{z}}_l$ using a uniform scalar quantizer (Q) and losslessly compressed using the arithmetic encoder (AE) and decoder (AD), as shown in Fig.1. During training, additive stochastic noise is added to approximate the quantization process. Therefore, we have $\tilde{\mathbf{z}}_l = \mathbf{z}_l + \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)$. During testing, $\mathbf{z}_l$ is mapped into the nearest integer $\hat{\mathbf{z}}_l$ using rounding operation. The main decoding network reconstructs the output frames $\hat{\mathbf{x}}_l$ from a given bitstream of $\hat{\mathbf{z}}_l$. Consequently, the proposed neural speech coder consists of $2.35M$ trainable network parameters.

3.2. Probability density modeling

An accurate estimation of the PDF is essential to compress the bottleneck samples effectively. In this paper, we employ both the fully factorized entropy model (FEM) [17], and the mean-scale hyperprior model (HPM) [13]. While HPM has been shown to be effective in compressing frame-wise audio samples, it may not perform optimally when the bit budget is insufficient at low bitrates. To address this issue, we also test the effect of FEM to model the density of the bottleneck samples without any side information.

3.2.1. Non-parametric density model

The detailed structure of FEM is illustrated in Fig.2(a), which is similar to that in [17]. Because there is no prior assumption about $\hat{\mathbf{z}}_l^h$, the FEM estimates the PDF of $\hat{\mathbf{z}}_l^h$, which is shared by the arithmetic encoder (AE_h) and decoder (AD_h). The FEM shown in Fig.2(a) comprises several trainable pa-

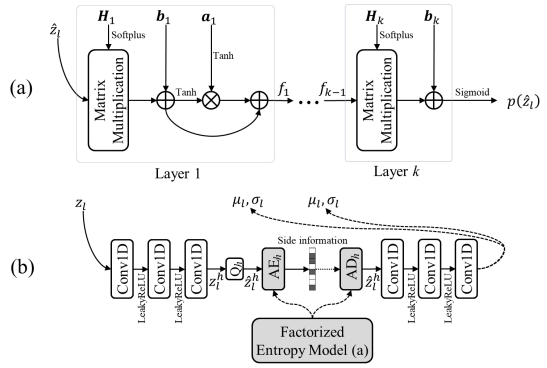

Figure 2: *Probability modeling blocks: (a) Factorized Entropy Model (FEM) and (b) Mean-Scale Hyperprior Model (HPM)*

rameters that govern the expansion or contraction rate, including matrices $\boldsymbol{H}_i$ and biases $\boldsymbol{b}_i$, as well as a control vector $\boldsymbol{a}_i$ where i is the layer index. The i-th layer output is calculated as $f_i = \mathcal{O}_i\left(\text{softplus}(\boldsymbol{H}_i) \cdot \boldsymbol{x} + \boldsymbol{b}_i\right)$, where $\mathcal{O}_i$ are nonlinearities defined as $\mathcal{O}_i(\boldsymbol{x}) = \boldsymbol{x} + \tanh(\boldsymbol{a}_i) \odot \tanh(\boldsymbol{x})$. The elements of $\boldsymbol{H}_i$ are constrained to be non-negative, and those of $\boldsymbol{a}_i$ to be lower-bounded by -1 [17]. In the last layer, to ensure that the output $p(\hat{\mathbf{z}}_l)$ is an appropriate density function, sigmoid function is used instead of o_i. We use four ($k = 4$) layers, resulting in 43 trainable parameters.

3.2.2. Mean-scale hyperprior model

Another model for the bottleneck samples is the mean-scale HPM developed in [13]. We employ an autoencoder-type HPM sub-network, as illustrated in Fig.2(b). The HPM sub-network includes simple 1D convolutional layers with kernel sizes of $\{7, 9, 9\}$ and strides of $\{1, 2, 2\}$.

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

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

where $\mathcal{N}$ denotes the cumulative distribution function of a standard normal Gaussian distribution. The HPM sub-network accepts the latent vector $\mathbf{z}_l$ and estimates the hyper-representation $\mathbf{z}_l^h$. $\mathbf{z}_l^h$ is then quantized using a uniform quantizer Q_h, producing discrete symbols $\hat{\mathbf{z}}_l^h$ that are arithmetically encoded (AE_h) and transmitted as side information. The arithmetic decoder (AD_h) in the hyper-decoder recovers the discrete symbols $\hat{\mathbf{z}}_l^h$, which are used to predict the means (μ_l) and scales (σ_l) of the latent samples in $\hat{\mathbf{z}}_l$. The HPM sub-network comprises the FEM in Fig.2(a) for the arithmetic coding, resulting in $0.18M$ trainable network parameters.

For the model with HPM, the bit-per-sample usage of the main and sub-networks was measured at five different bitrates, and the results are shown in Table 1. As the bitrate decreases, the sub-network consumes more bits for hyperprior modeling, but the ratio decreases. This implies that as the bitrate decreases, the need for side information for hyperprior generation may affect the coding efficiency.

3.3. Loss function

A loss function for training the proposed neural speech coder can be rewritten as a combination of several loss terms:

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

Table 1: *Bit allocations in the main- and sub-networks.*

Bitrates	9kbps	13kbps	16kbps	20kbps	24kbps
Main-network	0.469	0.678	0.834	1.028	1.256
Sub-network	0.119	0.158	0.191	0.225	0.264
Side Info Ratio	20.24%	18.90%	18.63%	17.96%	17.37%

where $\lambda_1, \lambda_2, \lambda_3$ are the blending factors. The loss function includes the mean-square error (MSE), calculated as $\mathcal{L}_{mse} = ||\hat{\mathbf{x}}_l - \mathbf{x}_l||_2^2$. In addition, it has two PAM-based loss terms: $\mathcal{L}_p^G$ and $\mathcal{L}_p^L$, denoting perceptual distortions measured using globally long and locally short windows, respectively. $\mathcal{L}_p^G$ is calculated from a frame of 512 samples, and $\mathcal{L}_p^L$ is 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 [12]. The obtained parameters are transformed using three Mel filterbanks with 16, 32, and 64 bands.

L_e in Eq. (2) is the rate term, estimated using discrete latent vector representations. The rate term can be estimated as $\mathcal{L}_e = \sum\text{-}\log_2 p(\hat{\mathbf{z}}_l|\hat{\mathbf{z}}_l^h) + \sum\text{-}\log_2 p(\hat{\mathbf{z}}_l^h)$. 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. Through experiments, we could confirm that the estimated bitrate coincides with the actual bitrate obtained using the arithmetic encoder.

4. Experimental Results

4.1. Test setup

For the experiments, we constructed a dataset using the TIMIT corpus [24] with a sampling frequency of $16kHz$. We selected 5,600 utterances for training, 200 for validation, and 500 for testing. All sources underwent min-max normalization. We used a fixed mini-batch size of 128 and adjusted the learning rate using cosine annealing between 0.0001 and 0.00005. The model was trained on PyTorch using the Adam [25] optimizer. Considering the dynamic range of each loss term, we initialized the blending factors as $\lambda_1 = 60$, $\lambda_2 = 0.003$, and $\lambda_3 = 0.0005$. Still, they were adjusted during training until the DNN speech coder reached the pre-set bitrate range. We evaluated the performance of our proposed DNN-based neural speech coder against commercial speech coders such as AMR-WB and OPUS, as well as our previous neural speech coder proposed in [12]. We trained all neural speech coders for at least 200 epochs to ensure a fair comparison.

4.2. Objective quality measure

We conducted measurements on the output speech quality using four specific metrics: Signal-to-Noise Ratio (SNR), Perceptual Evaluation of Speech Quality (PESQ), Virtual Speech Quality Objective Listener (ViSQOL) [26], and the 2f-model [27]. These measurements were taken at bitrates of 9, 13, 16, 20, and $24kbps$, respectively. The 2f-model is a recently proposed metric that effectively predicts perceived audio quality and is strongly correlated with actual subjective tests. The measurement results are presented in Table 2.

In terms of SNR, the results indicate that neural coders produce significantly higher SNRs than both AMR-WB and OPUS at all bitrates. This suggests that neural coders are able to synthesize speech with high accuracy and fidelity. As for PESQ, the neural coders achieved similar scores to each other, with scores significantly higher than AMR-WB but slightly lower than OPUS at bitrates higher than $16kbps$. However, OPUS had difficulties reconstructing high-frequency components at

Table 2: *Objective evaluation results.*

	SNR (dB)					PESQ-WB					ViSQOL					2f-model				
Bitrate (kbps)	∼9	∼13	∼16	∼20	∼24	∼9	∼13	∼16	∼20	∼24	∼9	∼13	∼16	∼20	∼24	∼9	∼13	∼16	∼20	∼24
AMR-WB	9.75	11.27	11.85	12.37	12.64	3.365	3.761	3.875	3.953	3.990	3.896	4.025	4.078	4.107	4.121	38.91	45.91	48.42	51.16	53.41
OPUS	8.11	9.70	11.18	12.33	13.03	3.305	4.010	4.230	4.375	4.453	3.608	4.000	4.098	4.173	4.228	32.71	48.71	51.30	53.28	56.04
Model in [12]	14.52	16.42	16.95	18.26	19.19	3.771	4.104	4.143	4.279	4.344	3.949	4.111	4.151	4.199	4.256	44.93	51.58	53.53	59.24	64.46
Proposed w FEM	14.05	16.32	16.97	18.66	19.29	3.807	4.111	4.162	4.331	4.382	3.960	4.122	4.159	4.227	4.257	45.32	51.67	55.25	62.95	64.78
Proposed w HPM	13.61	16.05	17.58	19.34	19.97	3.669	4.001	4.164	4.342	4.418	4.007	4.140	4.176	4.256	4.276	46.86	54.13	56.90	64.29	66.46

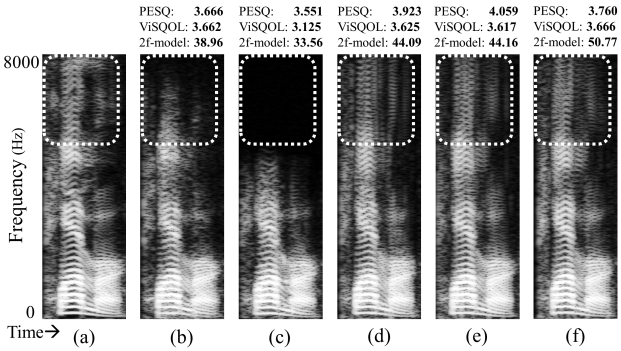

Figure 3: *Spectrograms of the output signals at* $9kbps$*: (a) Original, (b) AMR-WB, (c) OPUS, (d) the model in [12], (e) the proposed model with FEM, and (f) the proposed model with HPM.*

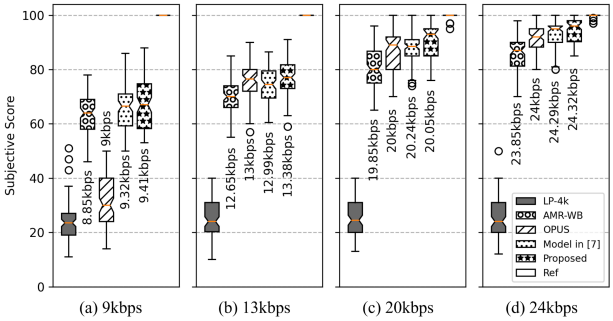

Figure 4: *MUSHRA test results.*

$9kbps$, resulting in the lowest PESQ. Among the neural coders, the model with HPM achieved the best scores at 20 and $24kbps$ by small margins, while the model with FEM achieved the best scores at 9 and $13kbps$ also by small margins. However, the model with HPM showed lower PESQs than the other two neural coders at 9 and $13kbps$, which might be attributed mainly to the extra bit demand for the hyperprior side information. Overall, it can be said that the use of the density model (either FEM at low bitrates or HPM at high bitrates) can generally improve PESQs compared to the previous model described in [12].

In terms of objective evaluation using ViSQOL and 2f-model, the proposed neural audio coding model, particularly when utilizing HPM, consistently achieves the highest scores across all bitrates. Even at 9 and $13kbps$, where HPM exhibits lower PESQ scores than the other neural coders, it still achieves the highest overall scores. Compared to OPUS, the proposed model with HPM outscores OPUS by $5.42 \sim 14.15$. Similarly, compared to FEM, HPM still performs better at all bitrates. Overall, our proposed model with HPM demonstrates a meaningful improvement over the other coders in terms of ViSQOL and 2f-model scores.

The discrepancy between PESQ and 2f-model scores can be somewhat validated from the spectrogram analysis in Fig. 3, where we compare the output spectrograms obtained at $9kbps$. In this case, HPM scored lower PESQ than other neural coders but achieved the highest ViSQOL and 2f-model scores. One particular note was that HPM enabled the coder to reconstruct a more accurate spectral structure, as indicated by the white boxes. The improved spectral accuracy could be one of the reasons for the higher ViSQOL and 2f-model scores achieved by the proposed model with HPM, despite the lower PESQ scores.

4.3. Subjective test results

To evaluate the subjective performance of the proposed neural audio coding model, Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) [28] tests were conducted with ten experienced listeners. In the tests, ten different audio utterances were randomly selected, and they were encoded and decoded at four different bitrates: 9, 13, 20, and 24kbps. The model with FEM was not included in this test. The results are shown in Fig. 4. Based on the median scores obtained from the MUSHRA tests, it can be observed that the proposed model with HPM outperforms both AMR-WB and OPUS, although there are overlaps between the scores of coders. Compared to AMR-WB, the improvement is significant at all bitrates except $9kbps$. When compared to OPUS, the improvement is still noticeable except for $13kbps$. Furthermore, compared to the previous model presented in [12], consistent improvements are observed at all bitrates, indicating that combining HPM with multi-time-scale PAM-based loss function can improve both coding efficiency and speech quality. At lower bitrates, the contribution of HPM to perceptual improvement was found to be limited as the additional bit demand for the side information was critical. In contrast, the improvement was relatively significant at higher bitrates such as 20 and $24kbps$. The results of the MUSHRA tests tend to agree more with the 2f-model scores than the PESQ scores presented in Table 2.

5. Conclusions

We proposed a perceptually improved end-to-end DNN speech coder using density models. By estimating more accurate bottleneck sample density, we can achieve a frame-dependent entropy model enhancing arithmetic coding efficiency. Test results showed consistently higher speech quality than AMR-WB and OPUS speech coders. Test samples are available online[1].

6. Acknowledgements

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZH1200, The research of the basic media·contents technologies]

[1] https://drive.google.com/uc?export=download&id=1jgnnJv4COAZSnVuX-UyvuQicSvzMOkOZ

7. References

[1] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2013.

[3] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[4] W. B. Kleijn, F. S. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "Wavenet based low rate speech coding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 676–680.

[5] J. Klejsa, P. Hedelin, C. Zhou, R. Fejgin, and L. Villemoes, "High-quality speech coding with sample RNN," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7155–7159.

[6] J. M. Martin-Doñas, A. M. Gomez, J. A. Gonzalez, and A. M. Peinado, "A deep learning loss function based on the perceptual evaluation of the speech quality," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1680–1684, 2018.

[7] S.-W. Fu, C.-F. Liao, and Y. Tsao, "Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality," *IEEE Signal Processing Letters*, vol. 27, pp. 26–30, 2019.

[8] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1570–1584, 2018.

[9] 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 Processing Letters*, vol. 27, pp. 2159–2163, 2020.

[10] 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 *Proc. INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association*, 2021, pp. 1694–1698.

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

[12] 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 *Proc. INTERSPEECH 2022 – 23rd Annual Conference of the International Speech Communication Association*, 2022, pp. 4411–4415.

[13] ——, "A perceptual neural audio coder with a mean-scale hyperprior," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[14] K. Zhen, M. S. Lee, J. Sung, S. Beack, and M. Kim, "Efficient and scalable neural residual waveform coding with collaborative quantization," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 361–365.

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

[16] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[17] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *International Conference on Learning Representations (ICLR)*, 2018.

[18] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

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

[20] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 620–636, 2002.

[21] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," *arXiv preprint arXiv:1602.04845*, 2016.

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

[23] 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 sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.

[24] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[26] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An open source production ready objective speech and audio metric," in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[27] 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*, vol. 29, 2021, pp. 1530–1541.

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