Psychoacoustic Calibration of Loss Functions for Efficient End-to-End Neural Audio Coding

Kai Zhen, Mi Suk Lee, Jongmo Sung, Seungkwon Beack, and Minje Kim,

IEEE Signal Processing Letters, 2020.

April. 1. 2022. (FRI)

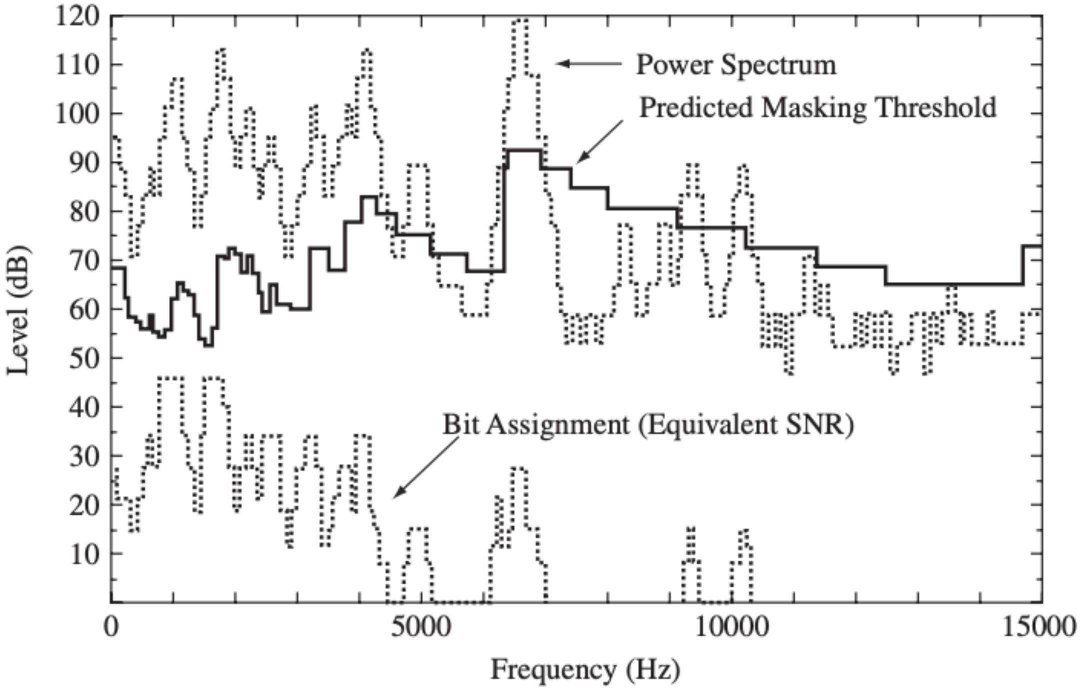
Youngwon Choi

GIST SAPL

Speech and Audio Processing Lab.

Briefing

- □ Conventional codec들은 masking threshold를 계산하여 coding에 사용되는 bit 개수를 최소화함.
  - – Masking threshold 아래로 내려가는 frequency band에 대해서는 bit를 할당할 필요가 없음.
- □ Neural codec에서는 이 원리를 사용하는 경우가 없었음.
- □ 본 논문에서는 neural codec 학습 과정에서 loss를 통해 masking threshold를 coding에 반영.



Psychoacoustic Model 1

□ PAM-1 : 인간의 masking threshold를 추정하여 함수로 표현 가능하도록 하는 모델.

□ 순서:

– (1) 신호의 STFT를 구한 후 SPL로 변환.
  - 4kHz sampled tone with amplitude 1 quantization level => 0dB SPL 로 가정.

– (2) Tonal masker 과 Noise masker 찾기.
  - Tonal masker은 spectral peak를 보이는 frequency bin.
  - Noise masker은 Tonal masker와 k_m이상 떨어진 모든 frequency bin.
  - SPL이 Absolute Threshold 아래로 내려가는 모든 masker들은 삭제됨.

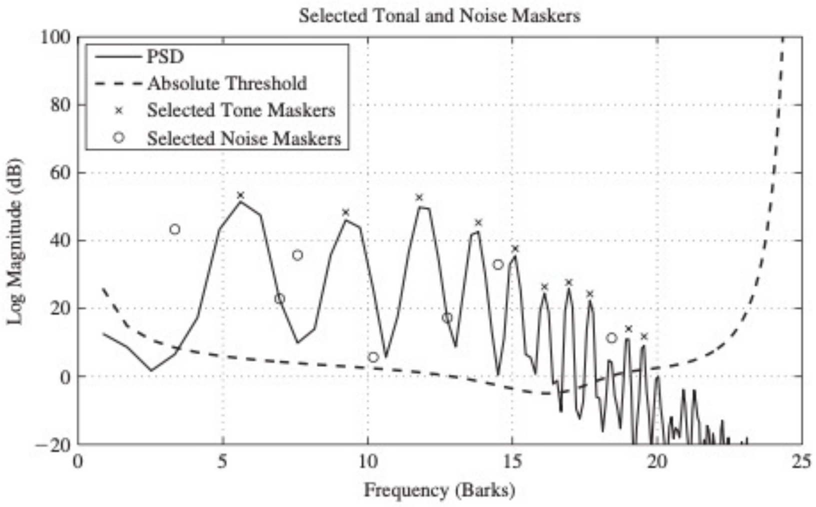
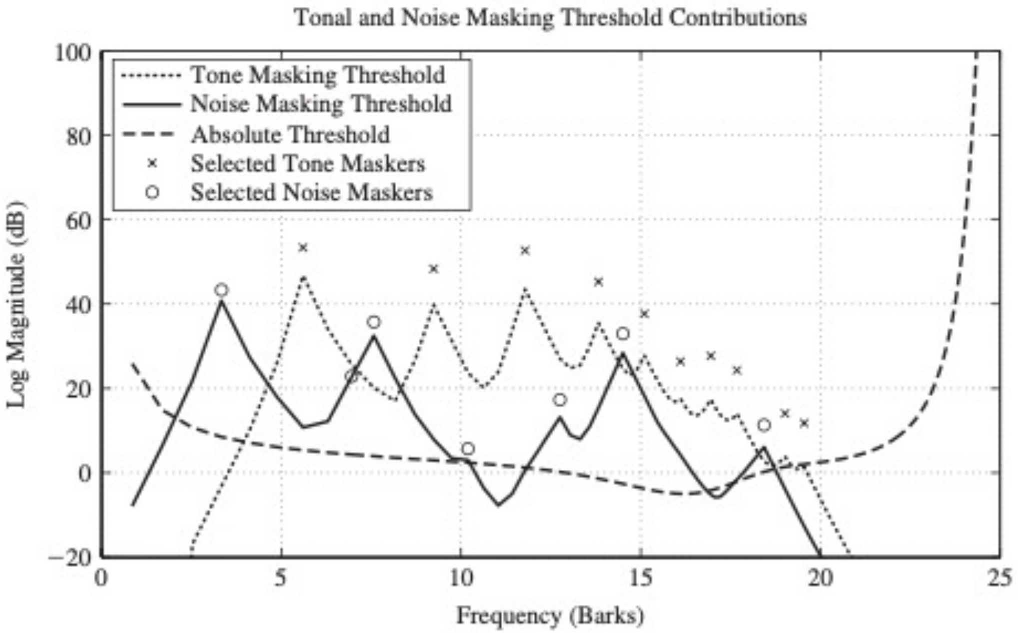


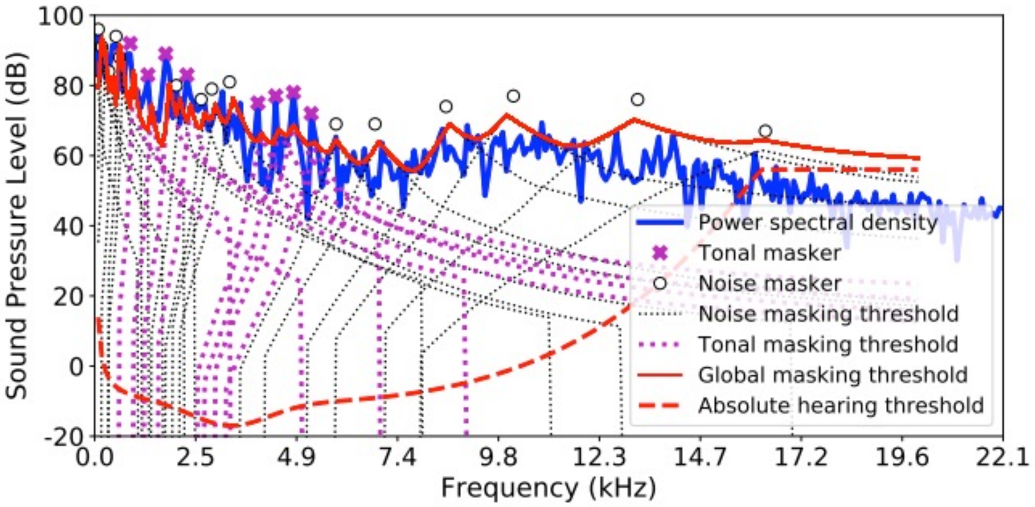
FIGURE 12.25
Tonal and noise masker candidates retained.

Psychoacoustic Model 1

- (3) 각 Tonal masker와 noise masker로 인해 형성되는 masking threshold 구하기.
- (4) Absolute, tone masking, noise masking threshold 를 합쳐 global masking threshold 구하기.

$$m_f = 10\log_{10}(10^{0.1Q_f} + \sum_t 10^{0.1U_{f,t}} + \sum_n 10^{0.1V_{f,n}})$$





- For more information, refer to the chapter 12 of “Theory and Applications of Digital Speech Processing”.

Contents

1. **Introduction**

2. **End-to-End Neural Audio Coding**
   - A. Lightweight NAC Module
   - B. Cross-Module Residual Learning

3. **The Proposed Psychoacoustic Calibration**
   - A. Psychoacoustic Model-1
   - B. Priority Weighting
   - C. Noise Modulation

4. **Experiments**
   - A. Experimental Setup
   - B. Experimental Results

5. **Conclusion**

1. Introduction

Introduction

- □ Audio coding compresses the original signal into a bitstream with a minimal bitrate (encoding) without sacrificing the perceptual quality of the recovered waveform (decoding).

- □ For this end, psychoacoustics is employed to quantify the audibility in both time and frequency domains (for conventional codec systems).
  - – MPEG-1 Audio Layer III (MP3) achieves a near-transparent quality at 128kbps by using a **psychoacoustic model (PAM)**.

- □ Recent efforts on deep neural network-based speech coding systems have made substantial progress on coding gain.
  - – By employing VQ-VAE and WaveNet, [1] yield a competitive speech quality at 1.6kbps with 20 million parameters.
  - – Recent neural speech synthesizers employ traditional DSP techniques, e.g., linear predictive coding, to reduce its complexity [2].
    - ▪ Although it can serve as a decoder of a speech codec, LPC does not generalize well to non-speech signals.

[1] Y. L. C. Garbacea, A. van den Oord, “Low bit-rate speech coding with VQ-VAE and a waveNet decoder,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2019.

[2] J.-M. Valin and J. Skoglund, “LPCNet: Improving neural speech synthesis through linear prediction,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2019.

Introduction

- □ Perceptually meaningful objective functions have shown an improved trade-off between performance and efficiency (in speech-related tasks).
  - – Some recent speech enhancement models successfully employed perceptually inspired objective metrics, e.g., perceptual attractors, energy-based weighting, perceptual weighting filters from speech coding, and global masking thresholds.
  - – Other neural speech enhancement models implement short-time objective intelligibility (STOI) and perceptual evaluation of speech quality (PESQ) as the loss.
    - ▪ These metrics may benefit speech codecs, but do not faithfully correlate with subjective audio quality.

- □ Meanwhile, PAM serves as a subjectively salient quantifier for the sound quality and is pervasively used in the standard audio codecs.
  - – However, integrating the prior knowledge from PAM into optimizing neural audio codecs has not been explored.

Introduction

- □ In this paper, author present a psychoacoustic calibration scheme to improve the neural network optimization process, as an attempt towards efficient and high-fidelity neural audio coding.
  - – With the global masking threshold calculated from a well-known PAM [3], the scheme firstly conducts priority weighting making the optimization process focus more on audible coding artifacts in frequency subbands with the relatively weaker masking effect, while going easy otherwise.
  - – The scheme additionally modulates the coding artifact to ensure that it is below the global masking threshold, which is analogous to the bit allocation algorithm in MP3.
  - – Authors claim that this is the first method to directly incorporate psychoacoustics to neural audio coding.

[3] T. Painter and A. Spanias, "Perceptual coding of digital audio," Proceedings of the IEEE, vol. 88, no. 4, pp. 451–515, 2000.

2. End-To-End Neural Audio Coding

End-To-End Neural Audio Coding

A. Lightweight NAC module

□ Given that neural codecs can suffer from a large inference cost due to their high model complexity, author choose a compact neural audio coding (NAC) module (450K parameters) as the building block.

- – The NAC module is a simplified version of the model used in [4].

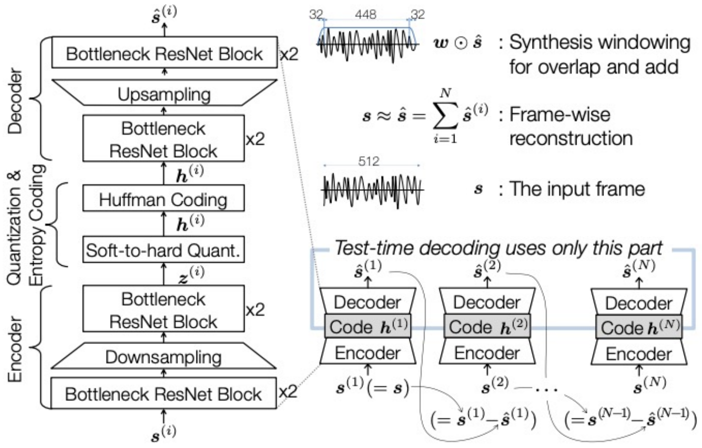


Fig. 1: Schematic diagrams for NAC. The residual coding pipeline for CMRL consists of multiple NAC autoencoding modules. Training and test-time encoding uses all blocks while the test-time decoding uses only the decoder portion.

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

End-To-End Neural Audio Coding

1) Encoder

□ The CNN Encoder maps an input frame of T time-domain samples, $s \in \mathbb{R}^T$ to the code vector, $z \leftarrow F_{enc}(s)$.

– Striding during the 1D convolution operation can downsample the feature map.

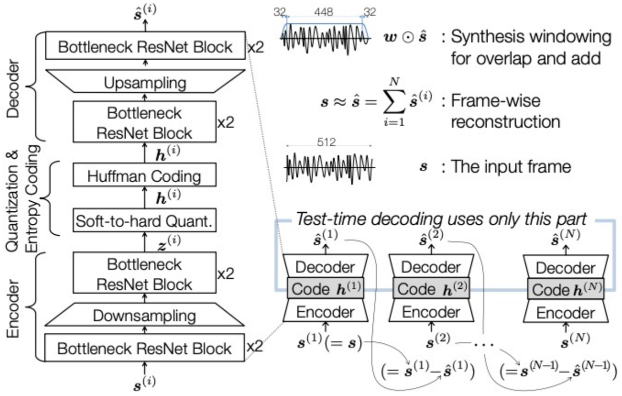
▪ In this case, stride = 2 in downsampling layer.

▪ So, $z \in \mathbb{R}^{T/2}$

3) Decoder

□ The decoder recovers the original signal from the quantized code vector: $\hat{s} \leftarrow F_{enc}(h)$.

– For upsampling, authors use a sub-pixel convolution layer proposed in [5].



$\hat{s}^{(i)}$
Decoder
Bottleneck ResNet Block x2
Upsampling
Bottleneck ResNet Block x2
$h^{(i)}$
Quantization & Entropy Coding
Huffman Coding
$h^{(i)}$
Soft-to-hard Quant.
$z^{(i)}$
Encoder
Bottleneck ResNet Block x2
Downsampling
Bottleneck ResNet Block x2
$s^{(i)}$
32 448 32
$w \odot \hat{s}$: Synthesis windowing for overlap and add
$s \approx \hat{s} = \sum_{i=1}^{N} \hat{s}^{(i)}$: Frame-wise reconstruction
512
s : The input frame
Test-time decoding uses only this part
$\hat{s}^{(1)}$ $\hat{s}^{(2)}$ $\hat{s}^{(N)}$
Decoder
Code $h^{(1)}$ Code $h^{(2)}$ Code $h^{(N)}$
Encoder
$s^{(1)}(= s)$ $s^{(2)}$ ... $s^{(N)}$
$(= s^{(1)} - \hat{s}^{(1)})$ $(= s^{(N-1)} - \hat{s}^{(N-1)})$

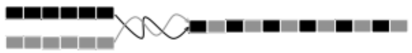
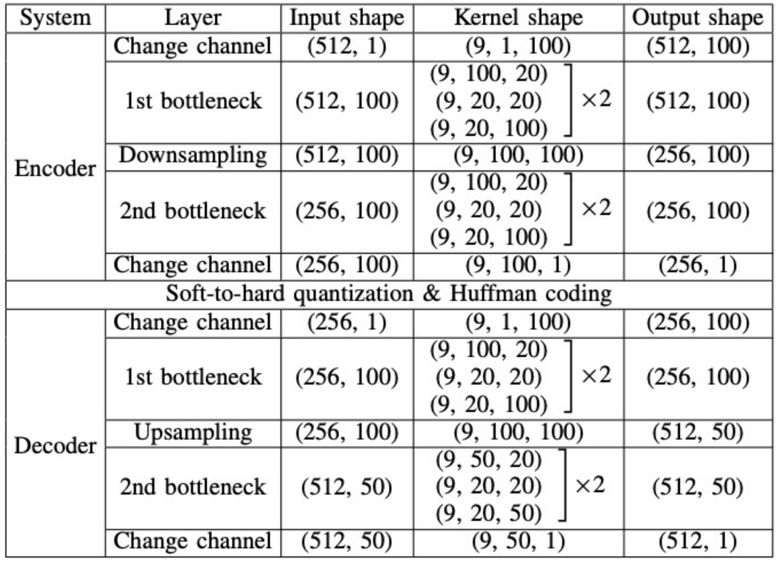


Figure 2: *The interlacing-based upsampling process.*



System	Layer	Input shape	Kernel shape	Output shape
Encoder	Change channel	(512, 1)	(9, 1, 100)	(512, 100)
	1st bottleneck	(512, 100)	(9, 100, 20) (9, 20, 20) (9, 20, 100) ×2	(512, 100)
	Downsampling	(512, 100)	(9, 100, 100)	(256, 100)
	2nd bottleneck	(256, 100)	(9, 100, 20) (9, 20, 20) (9, 20, 100) ×2	(256, 100)
	Change channel	(256, 100)	(9, 100, 1)	(256, 1)
Soft-to-hard quantization & Huffman coding				
Decoder	Change channel	(256, 1)	(9, 1, 100)	(256, 100)
	1st bottleneck	(256, 100)	(9, 100, 20) (9, 20, 20) (9, 20, 100) ×2	(256, 100)
	Upsampling	(256, 100)	(9, 100, 100)	(512, 50)
	2nd bottleneck	(512, 50)	(9, 50, 20) (9, 20, 20) (9, 20, 50) ×2	(512, 50)
	Change channel	(512, 50)	(9, 50, 1)	(512, 1)

[5] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, ´ D. Rueckert, and Z. Wang, “Real-time single image and video superresolution using an efficient sub-pixel convolutional neural network,” in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874–1883.

End-To-End Neural Audio Coding

2) Soft-to-Hard Quantization

□ Author used soft-to-hard quantizer [6], a clustering algorithm compatible with neural networks, where the representatives are also trainable.

□ During **training**, in each feedforward routine, the *c*-th code value z_c is assigned to the nearest kernel out of K, $\boldsymbol{\beta} \in \mathbb{R}^K$, which has been trained so far.

$$\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots, \beta_K\}$$

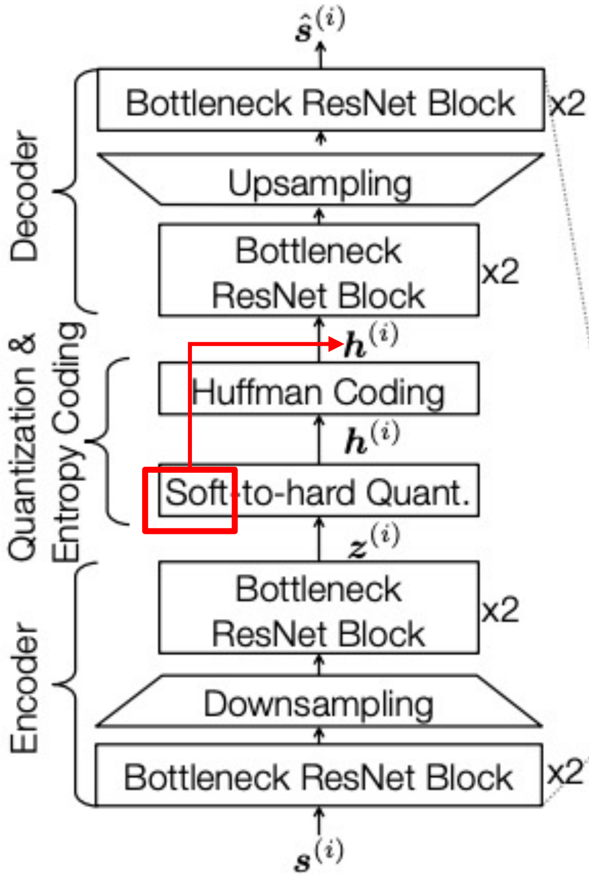
□ This operation is not differentiable, but can be approximated as follows:

$$\boldsymbol{d_c} = [|z_c - \beta_1|, \dots, |z_c - \beta_K|] \in \mathbb{R}^{T/2}$$

$$softmax(-\sigma \boldsymbol{d_c}) \rightarrow soft\ quantization$$

□ Soft dequantization would be,

$$softmax(-\sigma \boldsymbol{d})^{\boldsymbol{T}} \boldsymbol{\beta}$$



[6] 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 Advances in Neural Information Processing Systems (NIPS), 2017, pp. 1141–1151.

End-To-End Neural Audio Coding

2) Soft-to-Hard Quantization

□ For (hard) quantization, we need a loss function favoring soft quantization close to one-hot vector:

$$Q(c) = \frac{1}{T/2}\sum_{i=0}^{T/2}\left[\left(\sum_{j=0}^{K-1}\sqrt{c_{i,j}}\right) - 1.0\right]$$

- where $c_{i,j} = softmax(-\sigma \boldsymbol{d_i})_{\boldsymbol{j}}$
- $Q(c) = 0$ when all soft quantization are one-hot vectors.
- σ and $\boldsymbol{\beta}$ are learnable parameters. (Initially, σ=300)

□ At the **test** time, one hot vector replace soft quantization by turning on only the maximum element.

□ Huffman coding follows to generate the final bitstream.

End-To-End Neural Audio Coding

4) Bitrate Analysis and Control

□ The lower bound of the bitrate is defined as $|\boldsymbol{h}|H(\boldsymbol{h})$.

- $|\boldsymbol{h}|$ is the number of down-sampled and quantized features per second.
  - ex) In this paper, $|\boldsymbol{h}| = 256\left(\frac{features}{frame}\right) * \frac{44100}{512-32}\left(\frac{frames}{second}\right)$.
- Entropy $H(\boldsymbol{h})$ forms the lower bound of the average amount of bits per features and is adaptable during training.
- $H(\boldsymbol{h}) = -\sum_k p(\beta_k)\log_2 p(\beta_k)$, where $p(\beta_k)$ denotes the occurrence probability of the k-th cluster defined in the soft-to-hard quantization.

□ Therefore, during model training, $\lambda_{entropy}H(\boldsymbol{h})$ is added to the loss function as a regularizer navigatin g the model towards the target bitrate.

- Initially, $\lambda_{entropy} = 0$
- If bitrate overshoots the target, $\lambda_{entropy} = 0.015$
- Otherwise, $\lambda_{entropy} = -0.015$.

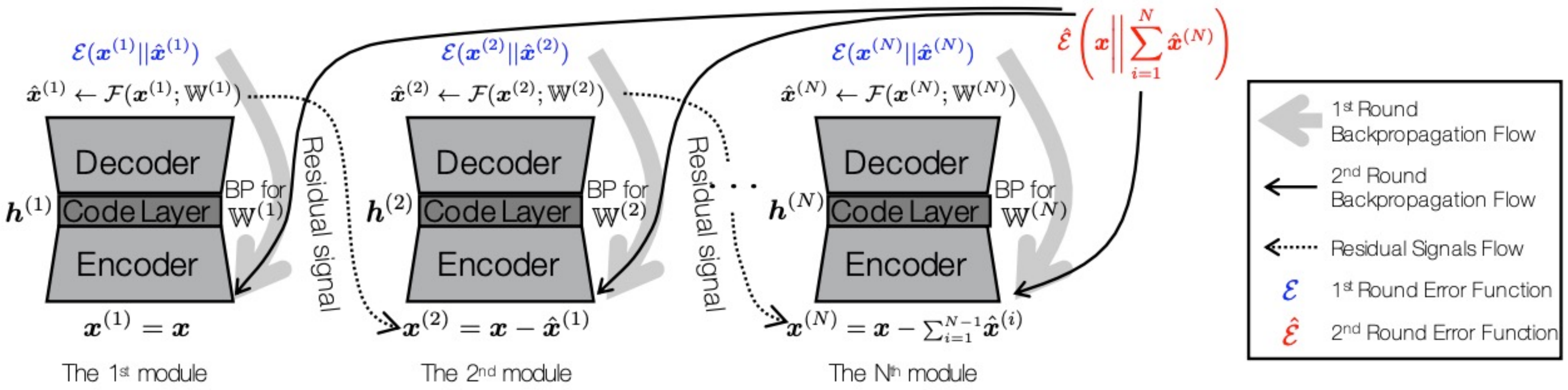
End-To-End Neural Audio Coding

B. Cross-Module Residual Learning (CMRL)

□ To scale up for high bitrates, CMRL [7] implants the multistage quantization scheme by cascading residual coding block.

□ Each module encodes what is not reconstructed from preceding modules, making the system scalable.

– Input of *i*-th module is $s^{i} = s - \sum_{j=1}^{i-1} \hat{s}^{(j)}$



$\hat{\mathcal{E}}\left(\boldsymbol{x} \middle\| \sum_{i=1}^{N} \hat{\boldsymbol{x}}^{(N)}\right)$
$\mathcal{E}(\boldsymbol{x}^{(1)}||\hat{\boldsymbol{x}}^{(1)})$
$\hat{\boldsymbol{x}}^{(1)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(1)}; \mathbb{W}^{(1)})$
Decoder
$\boldsymbol{h}^{(1)}$
Code Layer
BP for $\mathbb{W}^{(1)}$
Encoder
$\boldsymbol{x}^{(1)} = \boldsymbol{x}$
The 1st module
Residual signal
$\mathcal{E}(\boldsymbol{x}^{(2)}||\hat{\boldsymbol{x}}^{(2)})$
$\hat{\boldsymbol{x}}^{(2)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(2)}; \mathbb{W}^{(2)})$
Decoder
$\boldsymbol{h}^{(2)}$
Code Layer
BP for $\mathbb{W}^{(2)}$
Encoder
$\boldsymbol{x}^{(2)} = \boldsymbol{x} - \hat{\boldsymbol{x}}^{(1)}$
The 2nd module
Residual signal
· · ·
$\mathcal{E}(\boldsymbol{x}^{(N)}||\hat{\boldsymbol{x}}^{(N)})$
$\hat{\boldsymbol{x}}^{(N)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(N)}; \mathbb{W}^{(N)})$
Decoder
$\boldsymbol{h}^{(N)}$
Code Layer
BP for $\mathbb{W}^{(N)}$
Encoder
$\boldsymbol{x}^{(N)} = \boldsymbol{x} - \sum_{i=1}^{N-1} \hat{\boldsymbol{x}}^{(i)}$
The Nth module
1st Round Backpropagation Flow
2nd Round Backpropagation Flow
Residual Signals Flow
$\mathcal{E}$ 1st Round Error Function
$\hat{\mathcal{E}}$ 2nd Round Error Function

Figure 3: *Cross-module residual learning pipeline*

[7] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, “Cascaded crossmodule residual learning towards lightweight end-to-end speech coding,” in Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), 2019.

End-To-End Neural Audio Coding

□ For an input signal *s*, the encoding process runs all N autoencoder modules in a sequential order, which yields the bitstring as a concatenation of the quantized code vectors:

$$\boldsymbol{h} = \left[\boldsymbol{h}^{(1)T}, \boldsymbol{h}^{(2)T}, \dots, \boldsymbol{h}^{(N)T}\right]$$

□ During decoding, all decoders, $F_{dec}\left(h^{(i)}\right)\forall i$,run to produce the reconstructions that sum up to approximate the initial input signal as $\sum_{i=1}^{N} \hat{s}^{(i)}$.

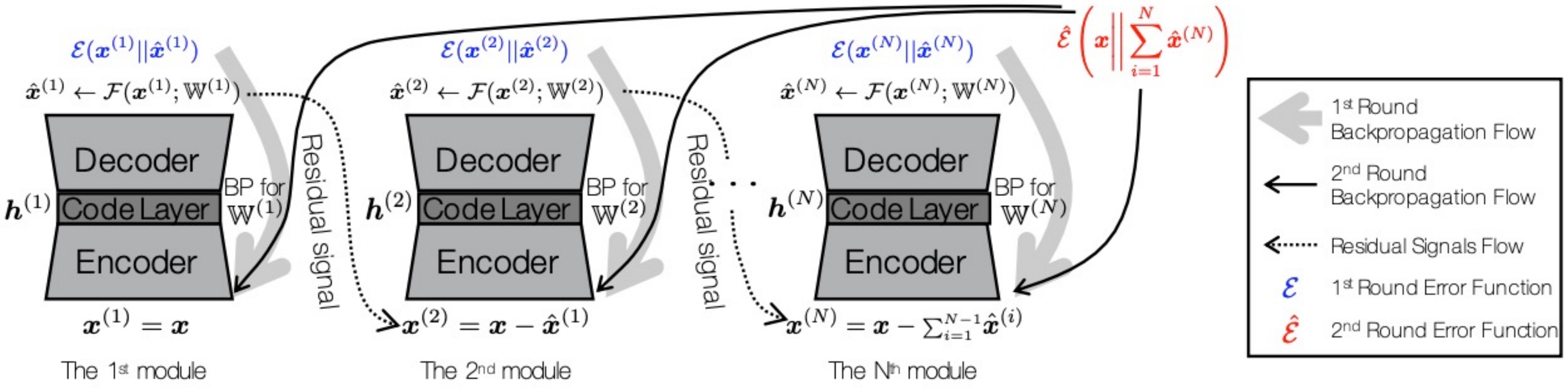

$\mathcal{E}(\boldsymbol{x}^{(1)}||\hat{\boldsymbol{x}}^{(1)})$
$\hat{\boldsymbol{x}}^{(1)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(1)}; \mathbb{W}^{(1)})$
Decoder
$\boldsymbol{h}^{(1)}$
Code Layer
BP for $\mathbb{W}^{(1)}$
Encoder
$\boldsymbol{x}^{(1)} = \boldsymbol{x}$
The 1st module
Residual signal
$\mathcal{E}(\boldsymbol{x}^{(2)}||\hat{\boldsymbol{x}}^{(2)})$
$\hat{\boldsymbol{x}}^{(2)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(2)}; \mathbb{W}^{(2)})$
Decoder
$\boldsymbol{h}^{(2)}$
Code Layer
BP for $\mathbb{W}^{(2)}$
Encoder
$\boldsymbol{x}^{(2)} = \boldsymbol{x} - \hat{\boldsymbol{x}}^{(1)}$
The 2nd module
Residual signal
$\mathcal{E}(\boldsymbol{x}^{(N)}||\hat{\boldsymbol{x}}^{(N)})$
$\hat{\boldsymbol{x}}^{(N)} \leftarrow \mathcal{F}(\boldsymbol{x}^{(N)}; \mathbb{W}^{(N)})$
Decoder
$\boldsymbol{h}^{(N)}$
Code Layer
BP for $\mathbb{W}^{(N)}$
Encoder
$\boldsymbol{x}^{(N)} = \boldsymbol{x} - \sum_{i=1}^{N-1} \hat{\boldsymbol{x}}^{(i)}$
The Nth module
$\hat{\mathcal{E}}\left(\boldsymbol{x} \middle|\middle| \sum_{i=1}^{N} \hat{\boldsymbol{x}}^{(N)}\right)$
1st Round Backpropagation Flow
2nd Round Backpropagation Flow
Residual Signals Flow
$\mathcal{E}$ 1st Round Error Function
$\hat{\mathcal{E}}$ 2nd Round Error Function

Figure 3: *Cross-module residual learning pipeline*

3. The proposed Psychoacoustic Calibration

The Proposed Psychoacoustic Calibration

□ The baseline model uses the sum of squared error defined in the time domain

$$L_1(\boldsymbol{s}||\hat{\boldsymbol{s}}) = \sum_{i=1}^{N}\sum_{t=1}^{T}(\hat{s}_t^{(i)} - s_t^{(i)})^2$$

□ In addition, another loss is defined in the mel-scaled frequency domain to weight more on the low frequency area, as the human auditory system does.

$$L_2(\boldsymbol{y}||\hat{\boldsymbol{y}}) = \sum_{i=1}^{N}\sum_{l=1}^{L}(y_l^{(i)} - \hat{y}_l^{(i)})^2$$

– $\boldsymbol{y}$ stands for a mel spectrum with L frequency subbands.

The Proposed Psychoacoustic Calibration

A. Psychoacoustic Model-1

□ Author choose a basic PAM that computes simultaneous masking effects for the input signal as a function of frequency, while the temporal masking effect is not integrated.

□ According to PAM-1, for input frame,

- (a) calculates the logarithmic power spectral density (PSD) p,
- (b) detects tonal and noise maskers,
- (c) calculates masking threshold for individual tonal and noise maskers, $\boldsymbol{U} \in \mathbb{R}^{F \times R}, \boldsymbol{V} \in \mathbb{R}^{F \times B}$, where R and B are the number of maskers.

The Proposed Psychoacoustic Calibration

□ The global masking threshold at frequency bin *f* is accumulated from each individual masker in (c) along with the absolute hearing threshold ***Q***, as

$$m_f = 10\log_{10}(10^{0.1Q_f} + \sum_r 10^{0.1U_{f,r}} + \sum_b 10^{0.1V_{f,b}})$$

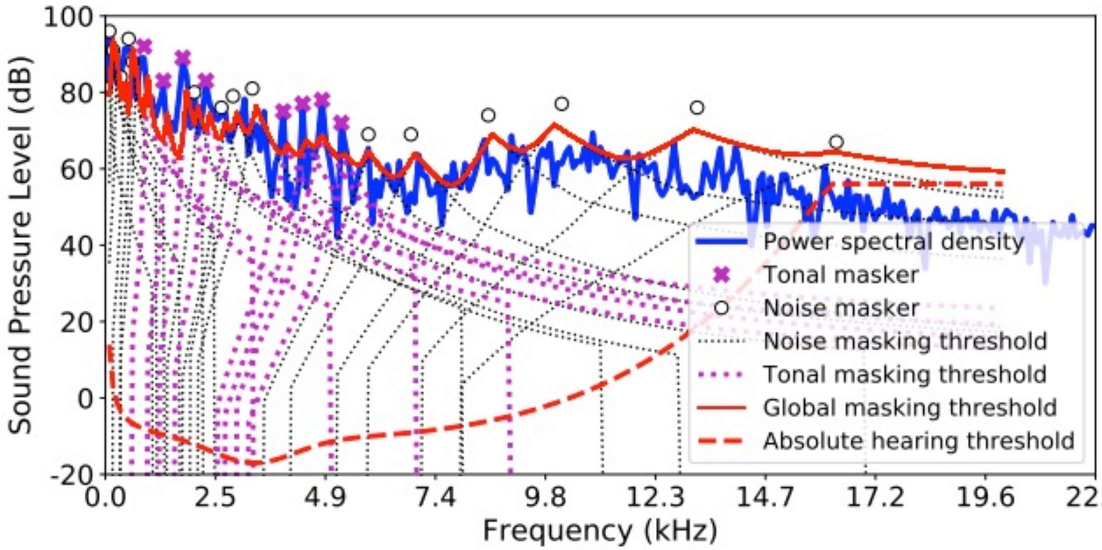


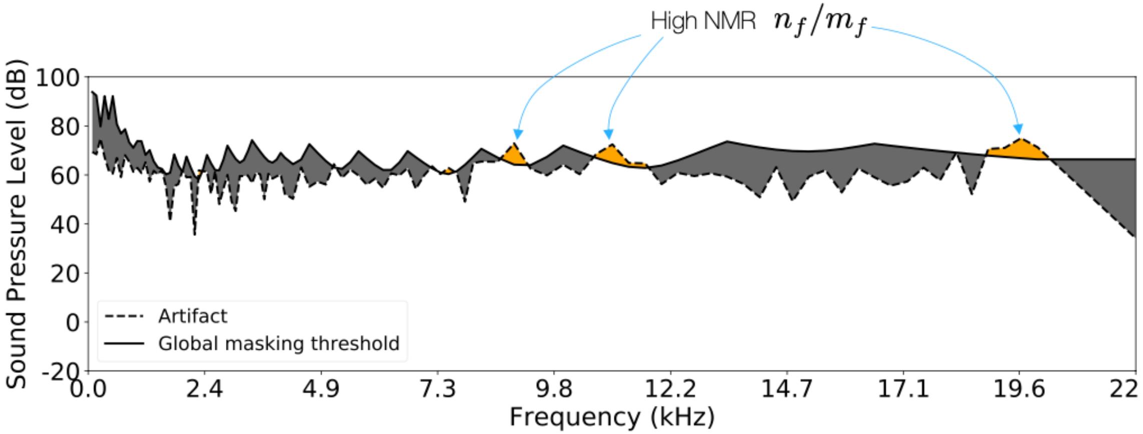
Fig. 2: Visualization of the masker detection, individual and global masking threshold calculation for an audio input.

The Proposed Psychoacoustic Calibration

□ Global masking threshold is used in various conventional audio codecs to allocate minimal amount of bits without losing the perceptual audio quality.

□ Typically, the bit allocation algorithm optimizes n_f/m_f(Noise-to-mask ratio; NMR), where n_f denotes the power of the noise (i.e., coding artifacts) in the subband f.

- In an iterative process, each time the bit is assigned to the subband with the highest NMR until no more bit can be allocated.



□ Author propose two mechanisms to integrate PAM-1 into NAC optimization: **priority weighting** and **noise modulation**.

The Proposed Psychoacoustic Calibration

B. Priority Weighting

□ During training, proposed models estimate the logarithmic PSD $\boldsymbol{p}$ out of an input frame s, as well the global masking threshold $\boldsymbol{m}$.

□ The log ratio between the signal power and the masked threshold rescaled from decibel is,

$$\boldsymbol{w} = \log_{10}(\frac{10^{0.1}\boldsymbol{p}}{10^{0.1}\boldsymbol{m}} + 1)$$

□ **Priority weighting loss** is,

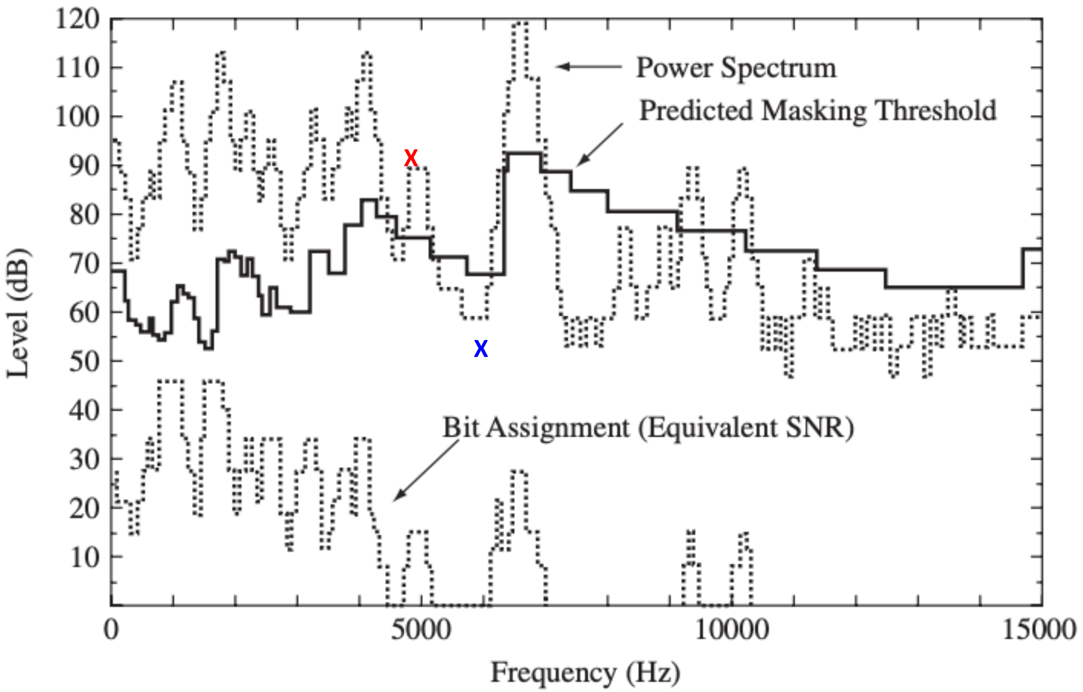
$$L_3(\boldsymbol{s}||\hat{\boldsymbol{s}}) = \sum_i \sum_j w_f \left(x_f^{(i)} - \hat{x}_f^{(i)}\right)^2 ,$$

where $x_f^{(i)}$ and $\hat{x}_f^{(i)}$ are the f-th magnitude of the Fourier spectra of the input and the recovered signals for the *i*-th CMRL module.

The Proposed Psychoacoustic Calibration

□ The intuition is that, if the signals power is greater than its masking threshold at the f-th frequency bin, the model tries hard to recover this audible tone precisely": a large w_f will force it.

- – Otherwise, for a masked tone, the model is allowed to generate some reconstruction error.
- – The weights are bounded between 0 and ∞.



The Proposed Psychoacoustic Calibration

C. Noise Modulation

□ The priority weighting mechanism can accidentally result in audible reconstruction noise, exceeding the mask value m_f if w_f is small.

□ Noise modulation loss exploit NMR n_f/m_f directly, where $\boldsymbol{n}$ is the power spectrum of the reconstruction error $\boldsymbol{s} - \sum_{i=1}^{N} \hat{\boldsymbol{s}}^{(i)}$ from all N autoencoding modules.

□ Author tweak the greedy bit allocation process in the MP3 encoder that minimizes NMR iteratively, such that it is compatible to the stochastic gradient descent algorithm as follows:

$$L_4 = max_f\left(ReLU\left(\frac{n_f}{m_f} - 1\right)\right)$$

The Proposed Psychoacoustic Calibration

$$L_4 = max_f\left(ReLU\left(\frac{n_f}{m_f} - 1\right)\right)$$

□ The ReLU function excludes the contribution of the inaudible noise to the loss when $\frac{n_f}{m_f} - 1 < 0$

□ Out of those frequency bins where the noise is audible, the max operator selects the one with largest NMR, which counts towards the total loss.

□ The process as such resembles MP3’s bit allocation algorithm, as it tackles the frequency bin with the largest NMR for each training iteration.

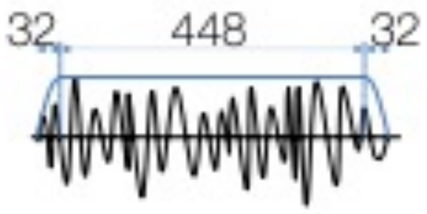
4. Experiments

Experiments

A. Experimental Setup

□ Dataset

- 1,000 single-channel clips of commercial music, 20 seconds long, spanning 13 genres.
- Sampling rate is 44.1kHz.
  - For lower bitrate setup, downsampled to 32kHz.
- Each frame contains 512 samples with an overlap of 32 samples.
  - Hann window is applied to the overlapping region.



□ Hyperparameters

- Batch size: 128 (frames)
- Initial softmax scaling factor: $\alpha = 300$
- Learning rate:
  - Initial learning rate
    - 2×10^{-4} for training the first modules
    - 2×10^{-5} for training the second cascaded modules.
- 64 and 32 kernels for the quantization for low and high bitrate cases.
- 50 and 30 for the number of epochs to train the first and second modules in CMRL.

Experiments

□ Competing Models

- Baseline models
  - Model-A : Model trained by loss,
    $$L = L_1$$
  - Model-B : Model trained by loss,
    $$L = L_1 + \lambda L_2$$
- Proposed models
  - Model-C : Model trained by loss,
    $$L = L_1 + \lambda(L_2 + L_3)$$
  - Model-D: Model trained by loss,
    $$L = L_1 + \lambda(L_2 + L_3 + L_4)$$

* In this experiments, $\lambda = 0.1$

Loss review

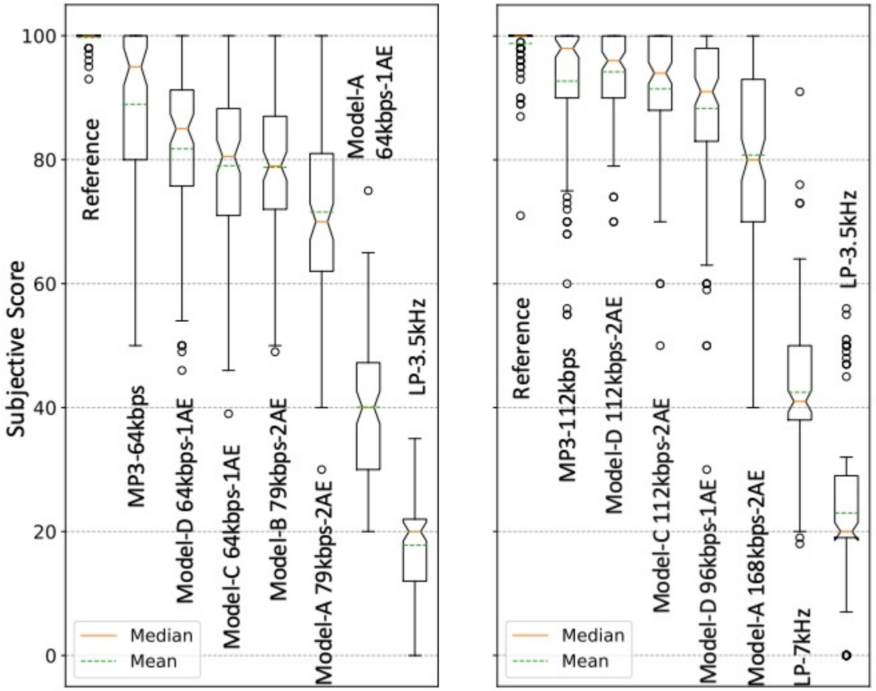
- L1: Sum of squared error (SSE)
  $$L_1(\boldsymbol{s}||\hat{\boldsymbol{s}}) = \sum_{i=1}^{N} \sum_{t=1}^{T} (\hat{s}_t^{(i)} - s_t^{(i)})^2$$
- L2: Mel-frequency Loss
  $$L_2(\boldsymbol{y}||\hat{\boldsymbol{y}}) = \sum_{i=1}^{N} \sum_{l=1}^{L} (y_l^{(i)} - \hat{y}_l^{(i)})^2$$
- L3: Priority weighting loss
  $$L_3(\boldsymbol{s}||\hat{\boldsymbol{s}}) = \sum_i \sum_j w_f \left(x_f^{(i)} - \hat{x}_f^{(i)}\right)^2$$
- L4: Noise modulation
  $$L_4 = max_f \left(ReLU \left(\frac{n_f}{m_f} - 1 \right) \right)$$

Experiments

B. Experimental Results

□ Subjective test using MUSHRA

- Ten audio experts participated on MUSHRA listening tests for low and high bitrate settings using headphones.
- Author randomly sample 13 songs, one per genre, and fix them throughout all tests.
- In figure, each model is specified by the target bitrate and model complexity.
    - ▪ ”Model-A 168kbps-2AE” is equipped with two concatenated AEs, trained by L_1 for a bitrate of 168kbps.



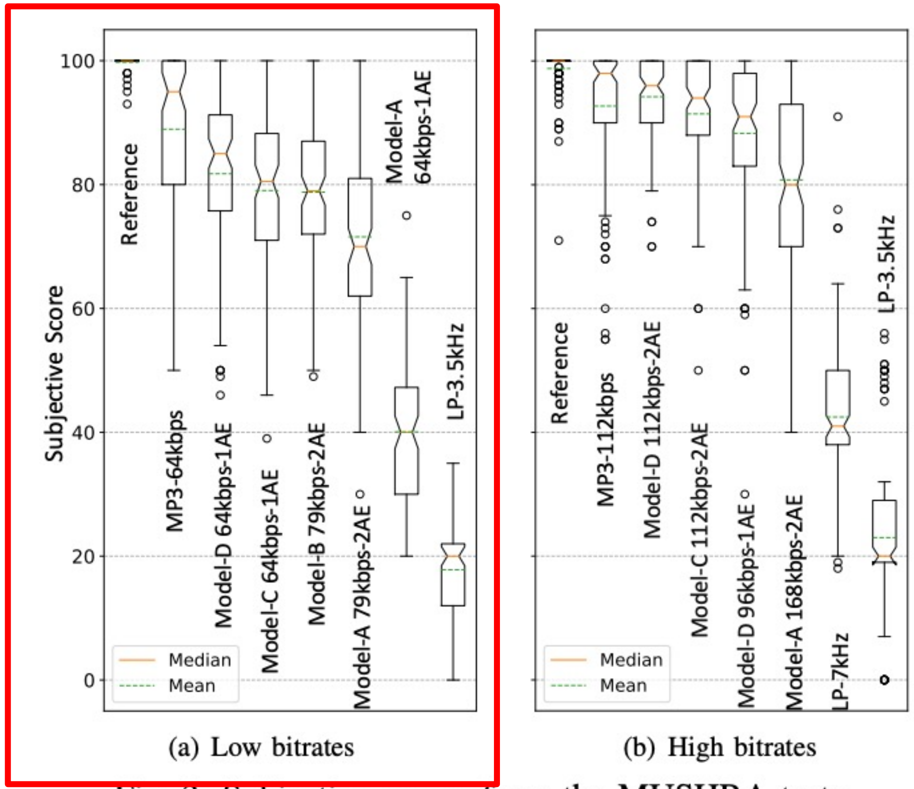
(a) Low bitrates (b) High bitrates

Fig. 3: Subjective scores from the MUSHRA tests.

Experiments

□ Low bitrate session

- Targets at 64kbps with the sample rate of 32kHz.
- The results show that the Model-C and Model-D with a smaller architecture and lower bitrate outperforms the Model-A and Model-B.
- Model-D is superior to Model-C.
- However, model-D does not outperform the commercial MP3 codec, due to its lightweight network.



(a) Low bitrates

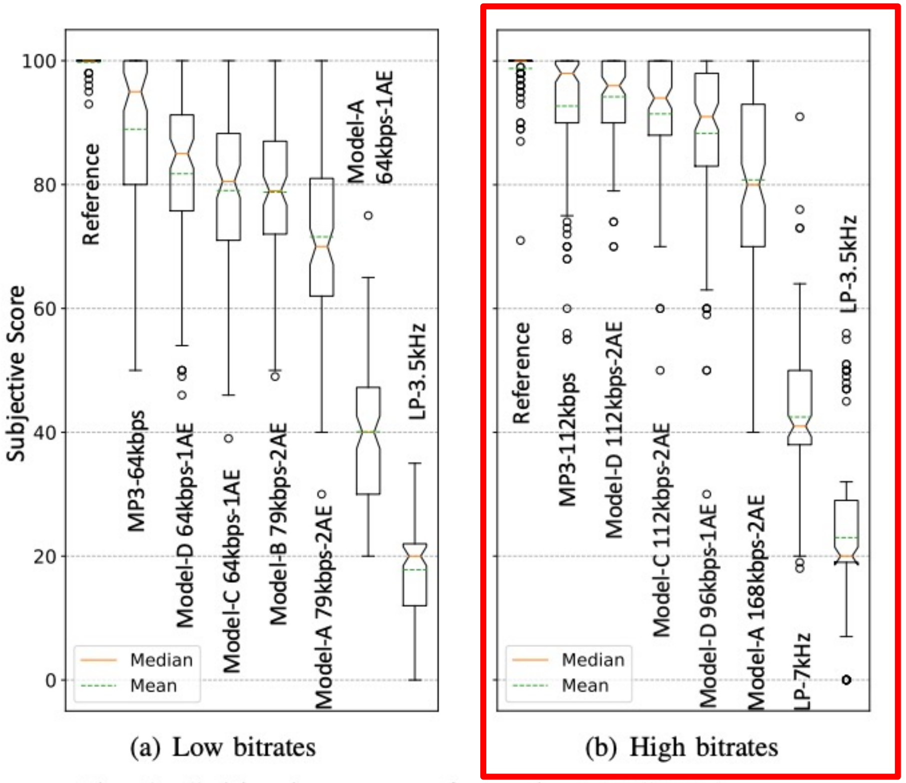
(b) High bitrates

Fig. 3: Subjective scores from the MUSHRA tests.

Experiments

□ High bitrate session

- Targets at 112kbps with the sample rate of 44.1kHz.
- Model-D outperforms Model-A which is twice as large and performs at a 64.3% higher bitrate.
- With 900K parameters, Model-D achieves almost transparent quality similar to MP3 at the same bitrate.



(a) Low bitrates

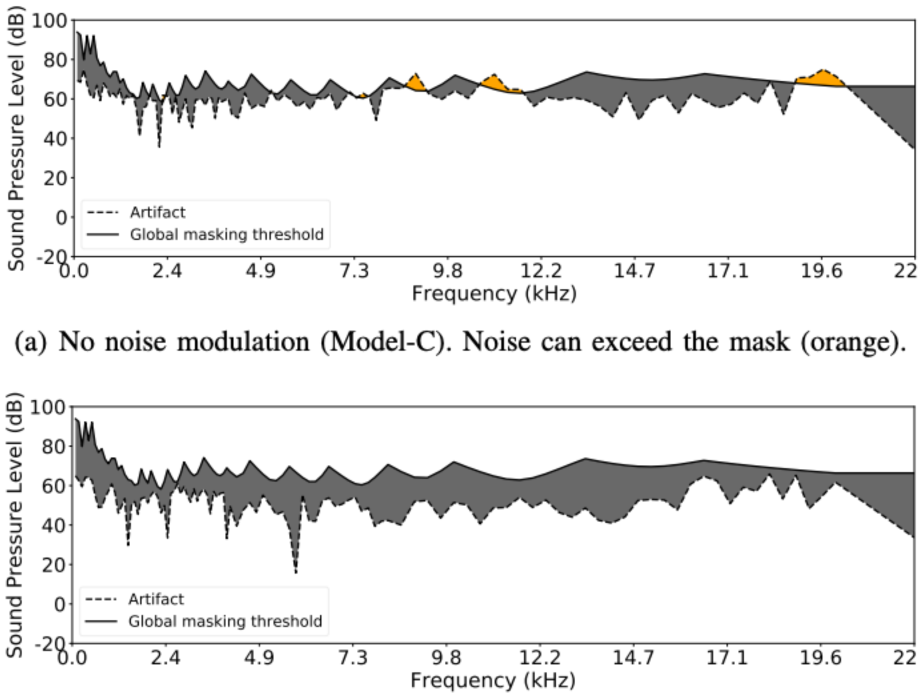
(b) High bitrates

Fig. 3: Subjective scores from the MUSHRA tests.

Experiments

□ The effect of the proposed noise modulation loss

– While Model-C can result in audible reconstruction error, the noise modulation loss in Model-D suppresses it under the masking curve in b.



(a) No noise modulation (Model-C). Noise can exceed the mask (orange).

(b) With noise modulation (Model-D)

Fig. 4: The effect of the proposed noise modulation loss.

5. Conclusion

Conclusion

□ Authors showed that incorporating the simultaneous masking effect in the objective function is advantageous to NAC in terms of the coding gain and model efficiency.

□ Although the system is based on PAM-1, it successfully proved the concept and suggests that a more advanced PAM, e.g., by employing temporal masking, will improve the performance further.

Thank you for listening
Q & A