VRVQ: Variable Bitrate Residual Vector Quantization for Audio Compression

Yunkee Chae[1,3*] **Woosung Choi**[1] **Yuhta Takida**[1] **Junghyun Koo**[1]
Yukara Ikemiya[1] **Zhi Zhong**[2] **Kin Wai Cheuk**[1] **Marco A. Martínez-Ramírez**[1]
Kyogu Lee[3,4,5] **Wei-Hsiang Liao**[1] **Yuki Mitsufuji**[1,2]

[1]Sony AI, [2]Sony Group Corporation, Tokyo, Japan
[3]IPAI, [4]AIIS, [5]Department of Intelligence and Information, Seoul National University

Abstract

Recent state-of-the-art neural audio compression models have progressively adopted residual vector quantization (RVQ). Despite this success, these models employ a fixed number of codebooks per frame, which can be suboptimal in terms of rate-distortion tradeoff, particularly in scenarios with simple input audio, such as silence. To address this limitation, we propose variable bitrate RVQ (VRVQ) for audio codecs, which allows for more efficient coding by adapting the number of codebooks used per frame. Furthermore, we propose a gradient estimation method for the non-differentiable masking operation that transforms from the importance map to the binary importance mask, improving model training via a straight-through estimator. We demonstrate that the proposed training framework achieves superior results compared to the baseline method and shows further improvement when applied to the current state-of-the-art codec.

1 Introduction

An audio codec encodes/decodes the audio, aiming to make the decoded signal perceptually indistinguishable from the original while using the minimum number of codes to conserve the bit budget. Conventional audio codecs usually rely on expert knowledge of psycho-acoustics and traditional digital signal processing to code and allocate bits to represent the coefficient of time-frequency domain representation such as modified discrete cosine transform (MDCT) [1]. The emergence of deep neural networks (DNNs) has led to the development of neural audio codecs that outperform conventional ones. Notably, residual vector quantization (RVQ)-based neural audio codecs [2–4] have demonstrated state-of-the-art performance by producing compact discrete representations at low bitrates. These codecs encode raw waveforms into parallel token sequences, subsequently decoded to reconstruct the audio signal from the summations of quantized residual vectors.

Since a typical RVQ-based neural codec allocates the same number of codebooks across all downsampled time frames, it can be considered a form of constant bitrate (CBR) codec.

This approach can be inefficient for encoding less complex segments, such as silence. Traditional codecs like AAC [5] tackle this inefficiency by supporting variable bitrate (VBR), which allocates more bits to complex audio segments and fewer bits to simpler ones. Recently, SlowAE [6] introduced VBR discrete representation learning by modeling the event-based representations using run-length transformers along the time dimension. Additionally, a VBR codec using DNN-based vector quantization is proposed in [7], employing a code allocation scoring function to determine the number of vectors allocated for each audio window. However, introducing VBR to an RVQ-based neural audio codec has not been explored yet.

*Work done during internship at Sony AI

NeurIPS 2024 Workshop on Machine Learning and Compression.

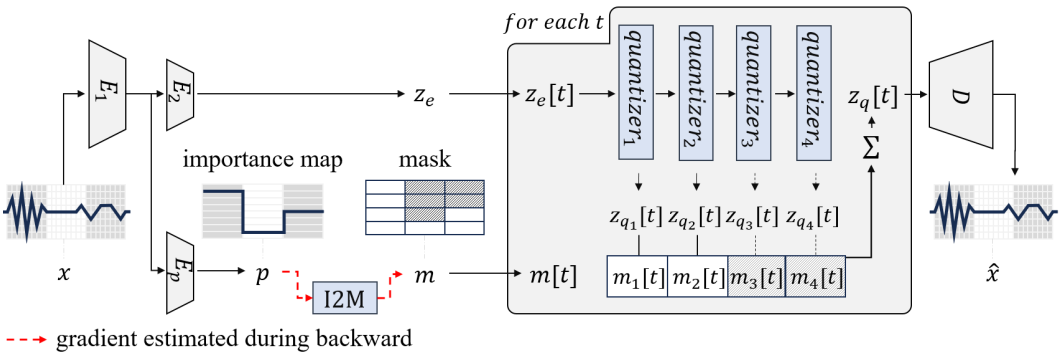


Figure 1: Overall proposed framework of variable bitrate RVQ.

Meanwhile, similar efforts have been made for VBR in DNN-based image compression. For instance, [8] utilized the self-attention map of a self-supervised DINO model [9] as a proxy for capacity to be assigned, leveraging the model's capability to capture semantic saliency. They allocated a variable number of bits to each pixel, applying a bit allocation map to their product quantizer [10]. On the other hand, some works have introduced *importance map* [11–14] to determine how many channels per pixel in the quantized features should be used. These importance maps are trained jointly with the compression model, enabling effective bitrate allocation for each pixel.

Inspired by these approaches, we propose a VBR RVQ (VRVQ) model, which allocates a different number of bits to each downsampled time frame in the latent space based on an importance map. By being jointly trained with rate-distortion loss optimization, the importance map learns to determine the optimal number of codebooks per frame, leading to a more effective rate-distortion curve compared to previous RVQ-based codecs. Consequently, our contributions are as follows:
Variable Bitrate RVQ: We are the first to achieve variable bitrate (VBR) coding to RVQ-VAE-based audio codecs. While many audio codecs utilize VBR, no prior work has attempted to incorporate variable bitrate into RVQ, which is the foundation of current state-of-the-art audio codec models.
Gradient Estimation for Binarized Variables: We adapt the importance map, originally used in image compression models, for audio coding. The challenge lies in the non-differentiable operations involved in generating binary masks. Previous approaches employed identical backward passes, but these methods proved suboptimal in our context. To address this, we propose a more efficient gradient estimation during binarization, ensuring more effective gradient flow during training.

2 Background

This section summarizes RVQ and structured dropout, used in several state-of-the-art CBR RVQ codecs [2–4].

2.1 Residual Vector Quantization

Let $z_e = E(x) \in \mathbb{R}^{D \times T}$ represent the output of the encoder E for the input audio x, where T and D denotes downsampled timeframes and the dimension of the latent representation, respectively. We denote the i-th frame at the time t of z_e as $z_e[t]$. An RVQ with N_q quantizers takes as input $z_e[t]$ and computes $z_q[t]$ as follows:

$$z_q[t] = \sum_{i=1}^{N_q} Q_i(r_i[t]), \tag{1}$$

where Q_i be the i-th quantizer and r_i is defined as $r_i[t] = z_e[t] - \sum_{j=1}^{i-1} Q_j(r_j[t])$, with the initial residual given by $r_1[t] = z_e[t]$. The resulting z_q is then fed to the decoder D to output the audio estimation $\hat{x}$.

2.2 Structured Codebook Dropout

Given a target bitrate, a separate model can be trained with the corresponding N_q. However, instead of this separate training, structured dropout [2, 15] can be used to enable a single RVQ model to support

multiple target bitrates. During training, rather than utilizing all the codebooks, only the first n_q quantizers are used (i.e., $z_q[t] = \sum_{i=1}^{n_q} Q_i(r_i[t])$), where n_q is randomly sampled from $\{1, \cdots, N_q\}$ for each batch (or batch item [4]). During inference, we can select the desired n_q for the target bitrate. We can rewrite the structured codebook dropout using masking notation as follows:

$$z_q[t] = \sum_{i=1}^{N_q} \mathbb{1}_{i \leq n_q} \cdot Q_i(r_i[t]), \tag{2}$$

where $\mathbb{1}$ denotes the indicator function. A neural codec trained with this trick remains a CBR codec, as it uses the same number of codebooks across all frames once a target bitrate is set.

3 Variable Bitrate RVQ

We propose a VRVQ by introducing the *importance map*. First, we outline the bit allocation method based on the importance map (3.1). Next, we propose a gradient estimation method (3.2) to jointly train the importance map within an encoder-decoder framework. Finally, we describe the joint training method (3.3) in detail.

3.1 Importance Map-based Bit Allocation

As shown in Fig. 1, the proposed bit allocation at time frame index t is controlled by the binary mask $m[t] \in \{0, 1\}^{N_q}$, where $m_i[t]$ determines the usage of the i-th codebook at t. Note that this mask is time-varying over t unlike the indicator function in Eq. (2). We design $m[t]$ to decrease monotonically over i (i.e., $m_i[t] \geq m_j[t]$ for $i < j$) for the compression efficiency. For a desirable mask generation, we adopt the *importance map*, initially proposed for image compression [11–14]. We compute the importance map from an intermediate feature map of the encoder E and generate a mask based on the importance map for the bit allocation.

We start by decomposing E into two components $E = E_2 \circ E_1$ to obtain the intermediate feature map: a shared part E_1 and an encoding-specific part E_2, borrowing notations from [13]. Given an audio x, a learnable subnetwork E_p takes $E_1(x)$ as input, and outputs an importance map $p = E_p(E_1(x)) \in (0, 1)^T$. Then, for each time frame index t, we generate $m[t] \in \{0, 1\}^{N_q}$ based on $p[t]$ with the following *importance map-to-mask* function I2M : $(0, 1) \to \{0, 1\}^{N_q}$:

$$m[t] = \text{I2M}(p[t]) := [H^0(S(p[t])), \cdots, H^{N_q-1}(S(p[t]))]^\mathsf{T}, \tag{3}$$

where $S(p) = N_q \cdot p$ is a scaling function and each $H^k(s)$ is defined as a Heaviside step function as follows:

$$H^k(s) := \begin{cases} 1 & \text{if } k \leq s, \\ 0 & \text{if } k > s \end{cases}. \tag{4}$$

The obtained mask $m[t]$ is then used to calculate the quantized vector z_q by replacing the indicator function in Eq. (2) as follows:

$$z_q[t] = \sum_{i=1}^{N_q} m_i[t] \cdot Q_i(r_i[t]). \tag{5}$$

3.2 Gradient Estimation for Binarized Variables

3.2.1 Baseline

Since H^k is a non-continuous function, gradient estimation for backpropagation is required to jointly train the importance map within the encoding-decoding framework. In the image domain [11, 12], the saturated identity function f_I^k is employed as a surrogate for H^k during backpropagation, defined as follows, and serves as the baseline function for our work:

$$f_I^k(s) := \max(\min(s - k, 1), 0) \tag{6}$$

However, this function presents an issue where the gradient does not flow through large regions due to the $\max$ and $\min$ operations (note that non-zero gradient can exist for only a single $k \in \{0, ..., N_q - 1\}$ in the I2M function). This problem makes the model suboptimal and degrades the performance of VRVQ, as demonstrated in our experiments.

3.2.2 Smooth approximation

To address this, we propose a smooth surrogate function that relaxes the f_I^k, allowing the gradient to be computed across all ranges, which is defined as follows:

$$f_\alpha^k(s) := \frac{1}{2\alpha} \log \left(\frac{\cosh(\alpha(s-k))}{\cosh(\alpha(-s+k+1))} \right) + \frac{1}{2} \tag{7}$$

This surrogate function is parameterized by $\alpha \in \mathbb{R}_{>0}$, which serves as a hyperparameter. In the extreme case where alpha approaches infinity, this function approaches f_I^k for all $s \in \mathbb{R}$ (i.e., $f_\infty^k(x) := \lim_{\alpha\to\infty} f_\alpha^k(s) = f_I^k(s)$). The shape of the function and the proof of convergence are explained in Appendix A.

By substituting the H^k in I2M with f_α^k, we define $\text{I2M}_{\text{soft}}^\alpha$, the surrogate function of I2M, and then apply the straight-through estimator for backpropagation as follows:

$$\text{I2M}_{\text{soft}}^\alpha(p[t]) := [f_\alpha^0(S(p[t])), \cdots, f_\alpha^{N_q-1}(S(p[t]))], \tag{8}$$

$$p[t] \mapsto \text{I2M}_{\text{soft}}^\alpha(p[t]) + \text{sg}\left(\text{I2M}(p[t]) - \text{I2M}_{\text{soft}}^\alpha(p[t])\right), \tag{9}$$

where sg denotes the stop-gradient operation.

3.3 Training

For the rate-distortion tradeoff, we define the rate loss $\mathcal{L}_R$ as follows:

$$\mathcal{L}_R = \frac{1}{T} \sum_{t=1}^{T} E_p(E_1(x))[t] \tag{10}$$

We basically follow the training scheme of [4], such as GAN training. During the generator training phase, we optimize the rate-distortion (R-D) tradeoff loss $\mathcal{L}_D + \beta\mathcal{L}_R$, where $\mathcal{L}_D$ is the training objective for generator of [4] and $\mathcal{L}_R$ is the proposed rate loss. The balancing of these two is controlled by the tradeoff parameter β, which serves as a hyperparameter.

The downside of the importance map is that once the importance map is trained, it consistently generates fixed output p for the identical input audio x, which restricts the flexibility of the model in terms of rate control. Inspired by random sampling approach used in structured dropout [15, 2], we address this problem by sampling the scaling factor l from the continuous uniform distribution $U([L_{\text{min}}, L_{\text{max}}])$, where $L_{\text{max}} > N_q$, replacing the scaling function $S(p)$ in Eqs. (3) and (8) with $S_l(p) = l \cdot p$ during training. This approach enables a single model to handle multiple bitrates in VBR mode. Here, we claim that if we interpret our framework from the perspective of the β-VAE objective [16], we can formulate the rate loss independently of the randomly sampled l. A detailed discussion on this can be found in Appendix D.

In this study, we opt not to train a separate entropy model or employ entropy coders such as arithmetic coding. Instead, we calculate the bitrate based on the number of codebooks used for discrete representation of the audio on frame-by-frame basis, allowing for potential future enhancements through the incorporation of entropy modeling.

4 Experiments

4.1 Model Architecture

In our experiments, we implement VRVQ audio codecs based on DAC [4], which itself builds upon the improved RVQGAN architecture. The improved RVQGAN is derived from the Improved VQGAN [17], which employs factorized codes and L2-normalized codebook lookups. Specifically, the codebook has an 8-dimensional space with a size of 1024, corresponding to 10 bits per codebook. The downsampling rate of the time frame is 512, resulting in a frame rate of 86 Hz in the latent space. In our experiment, we set the maximum number of codebooks N_q, to 8.

The importance map subnetwork is designed as a simple neural network composed of 1D convolutional layers with weight normalization [18] and Snake activation functions [19]. It takes the intermediate feature map $E_1(x)$, which has a channel dimension of 1024, extracted from the layer immediately preceding the encoder's final convolutional block (i.e., E_2). It processes the input feature

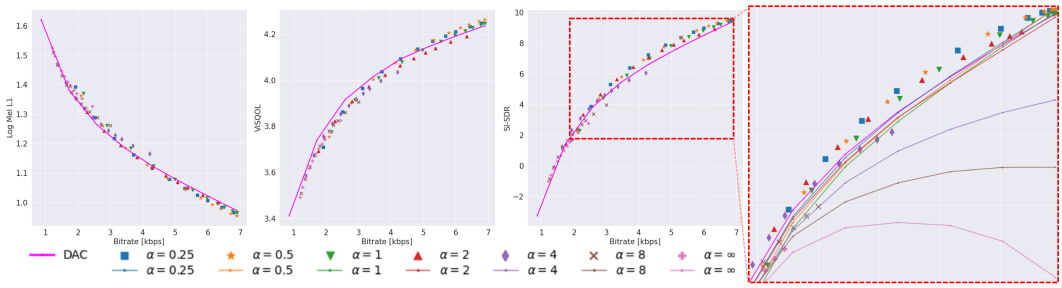


Figure 2: The results of VRVQ across different α. The points marked with various markers represent the results of inference at different scaling factor, l=4, 6, 8, 10, 12, 14, 16, 18, 20, 24, and 32, in VBR mode. In the rightmost plot, we display solid lines representing the results of inference in CBR mode for each model.

map through five sequential Conv1D blocks with kernel sizes of [5, 3, 3, 3, 1], progressively reducing the channel dimensions from 1024 to 512, 128, 32, 8, and finally 1. The output is then passed through a sigmoid activation function to produce the importance map.

4.2 Experimental Setup

We set the weight of the rate loss $\beta = 2$, and kept it fixed for all experiments. During training, we randomly sample the scaling factor l between $L_{\text{min}} = 1$ and $L_{\text{max}} = 48$. Note that l can be any positive real number between L_{min} and L_{max} for inference. Every model was trained with a batch size of 32 for 300k iterations, using audio segments of 0.38 seconds in duration.

For evaluation, we measure the ViSQOL [20], as well as the SI-SDR [21]. ViSQOL is a perceptual quality metric ranging from 1 to 5, designed to estimate a mean opinion score. We also report the sum of L1 losses of log mel spectrograms, using window sizes of [32, 64, 128, 256, 512, 1024, 2048] with a hop length of 1/4 of each window size, following the same configuration as in [4].

4.3 Dataset

We train our model using a dataset similar to that used in the original DAC model [4]. For the speech dataset, we utilize DAPS [22], Common Voice [23], VCTK [24], and clean speech segments from DNS Challenge 4 [25]. For the music dataset, we use MUSDB18 [26] and Jamendo [27]. Additionally, for general sounds, we incorporate AudioSet [28]. All audio is resampled to 44.1 kHz, and the training samples were normalized to -16 dB LUFS with phase shift augmentation applied.

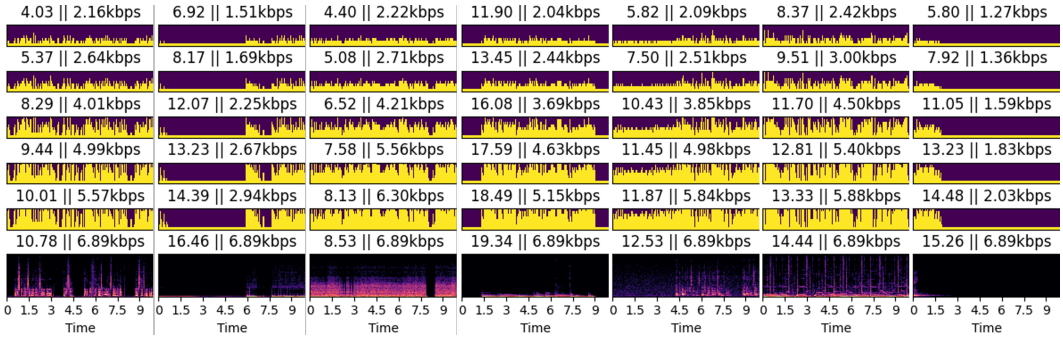
For evaluation, we use the F10 and M10 speakers from the DAPS dataset and the test sets from MUSDB18 and AudioSet. We randomly extract 100 10-second segments from each domain, resulting in a total of 300 samples for evaluation. The audio is not normalized during the evaluation phase.

5 Result

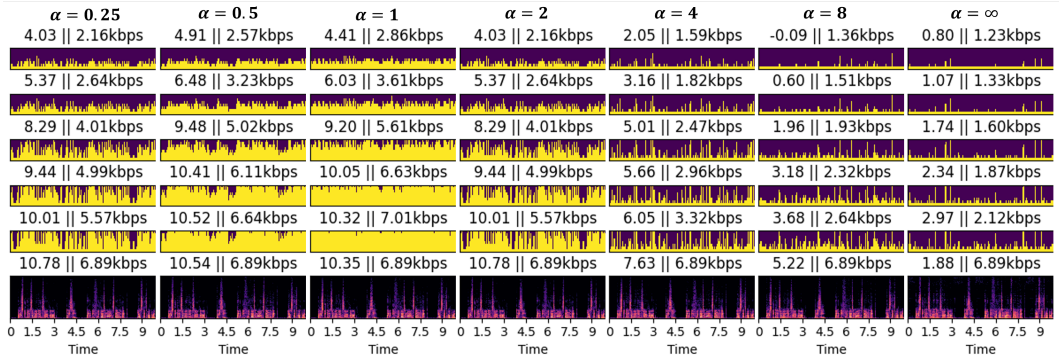
In this section, we discuss the impact of the f_α on VRVQ. The results of the additional ablation study are described in Appendix C.

Fig. 2 summarizes the result. For comparison, we also report the results of CBR RVQ with 8 codebooks trained from scratch for the same number of iterations with structured codebook dropout with a dropout rate of 0.5. For each model, we plot the rate-distortion curve for each metric. For our VBR models, we compute the average bitrate over the test dataset as well as the mean of each metric for the scaling factor l in the range of [L_{min}, L_{max}]. We also include the additional transmission costs (i.e., $\log_2(N_q) = 3$) for the number of codebooks used per frame into the bitrate calculation which amounts to approximately 0.258 kbps. Note that our VRVQ models can also be evaluated in a fixed-rate RVQ setup, i.e., in CBR mode not using an importance map. In this case, we do not calculate the transmission cost.

It is observable that the models trained with the proposed smooth surrogate functions outperform the model trained with the baseline function f_I. As α increases, the performance approaches that of f_I. This observation aligns with the fact that f_α converges to f_I. Our models, trained with $\alpha < 2$, show



(a) Codebook usage based on importance map with varying l for seven audio samples when $\alpha = 2$. The bottom row shows the spectrogram of the input audio.



(b) Codebook usage based on importance map for the same sample with varying α values. The bottom row shows the spectrogram of the reconstructed audio with the full number of codebooks.

Figure 3: Visualization of the codebook usages. From the first row, the scaling factor l are set to 6, 8, 14, 20, and 26. In each plot, the bitrate of the mask and the corresponding SI-SDR are also noted. For the bottom row, the bitrate of the sample is reported when it is inferred with the full number of codebooks in all frames (i.e., 8 codebooks in CBR mode), and thus, the bitrate is calculated without transmission cost.

better performance than the DAC trained for the same number of iterations under some conditions. Notably, for SI-SDR, our models perform significantly better than DAC. For the log mel loss and ViSQOL, our models outperform the DAC when α is 1 or lower and the bitrate is above 3-4 kbps.

However, across all metrics, we observe that the benefit of using VBR diminishes at lower bitrates. This is likely due to the overhead of additional transmission costs, which become relatively larger for lower bitrates. The performance at lower bitrates improves for a larger α. This occurs because, as shown in Fig. 3b, increasing α limits the model's ability to generate the importance map in various ranges, leading it to predominantly sample lower bitrates. This results in improved performance within that bitrate range. However, reduced diversity in the importance map results in significant underperformance at higher bitrates. It is also notable in Fig. 2 and 3b that when $\alpha = \infty$, the SI-SDR of the sample using the full codebook is worse than in cases with lower bitrates. This is because the importance map during training rarely utilized all codebooks, leading to poorer performance in the full codebook scenario and failing to optimize the rate-distortion tradeoff.

6 Conclusion

In this paper, we introduced a VBR RVQ for audio coding, leveraging an importance map. While developing the framework, we addressed key challenges such as managing the non-continuous function that converts the importance map into a binary mask by introducing the surrogate functions for gradient estimation, demonstrating that our approach provides better gradient flow compared to the function previously used in the image domain. As a result, we achieved efficient bit allocation within the RVQ-VAE framework and demonstrated the effectiveness of our models through rate-distortion curves across various metrics.

References

[1] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12. IEEE, 1987, pp. 2161–2164.

[2] 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.

[3] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[4] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] "Generic coding of moving pictures and associated audio information - part 7: Advanced audio coding (aac)," *ISO/IEC 13818-7*, 2006.

[6] S. Dieleman, C. Nash, J. Engel, and K. Simonyan, "Variable-rate discrete representation learning," *arXiv preprint arXiv:2103.06089*, 2021.

[7] S. Li, H. H. Mao, and J. McAuley, "Variable bitrate discrete neural representations via causal self-attention," in *2nd Pre-registration workshop (NeurIPS 2021), Remote*, 2021.

[8] F. Baldassarre, A. El-Nouby, and H. Jégou, "Variable rate allocation for vector-quantized autoencoders," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[9] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[10] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[11] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3214–3223.

[12] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[13] M. Li, W. Zuo, S. Gu, J. You, and D. Zhang, "Learning content-weighted deep image compression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3446–3461, 2020.

[14] J. Lee, S. Jeong, and M. Kim, "Selective compression learning of latent representations for variable-rate image compression," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 146–13 157, 2022.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework." *ICLR (Poster)*, vol. 3, 2017.

[17] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved vqgan," *arXiv preprint arXiv:2110.04627*, 2021.

[18] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

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

[20] 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.

[21] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[22] G. J. Mysore, "Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1006–1010, 2014.

[23] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[24] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, vol. 6, p. 15, 2017.

[25] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, and R. Aichner, "Icassp 2022 deep noise suppression challenge," 2022. [Online]. Available: https://arxiv.org/abs/2202.13288

[26] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The musdb18 corpus for music separation," 2017.

[27] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," in *International Conference on Machine Learning (ICML)*, 2019.

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

[29] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[30] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3146–3154.

[31] Y. Zhang, G. Lu, Y. Chen, S. Wang, Y. Shi, J. Wang, and L. Song, "Neural rate control for learned video compression," in *The Twelfth International Conference on Learning Representations*, 2023.

[32] J. Bae, M. R. Zhang, M. Ruan, E. Wang, S. Hasegawa, J. Ba, and R. Grosse, "Multi-rate vae: Train once, get the full rate-distortion curve," *arXiv preprint arXiv:2212.03905*, 2022.

[33] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, "Asymmetric gained deep image compression with continuous rate adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 532–10 541.

[34] F. Yang, L. Herranz, J. Van De Weijer, J. A. I. Guitián, A. M. López, and M. G. Mozerov, "Variable rate deep image compression with modulated autoencoder," *IEEE Signal Processing Letters*, vol. 27, pp. 331–335, 2020.

Appendix

A Smooth approximation of baseline function

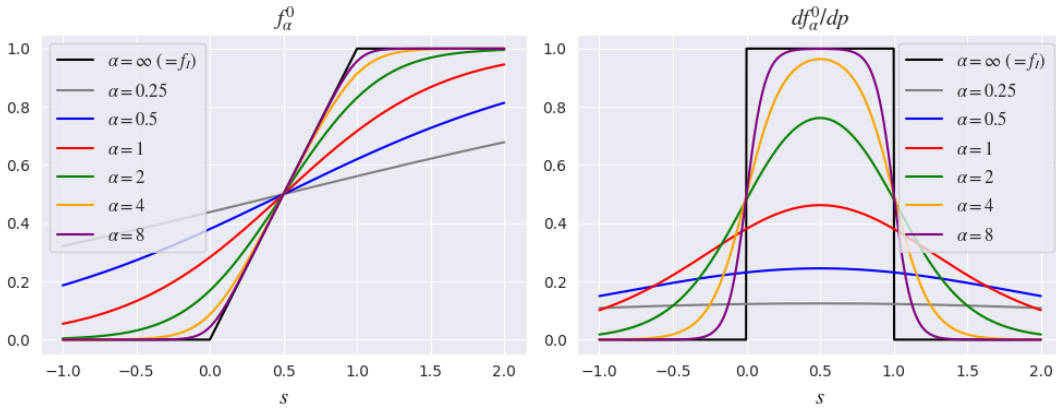


Figure 4: Gradient estimator f_α^k when $k = 0$

Eq. (7) is defined to approximate Eq. (6) smoothly, as illustrated in Fig. 4. This function is constructed to converge to Eq. (6) for all $s \in \mathbb{R}$ as α increase, as can be proven as follows.

Lemma 1. $\lim_{\alpha\to\infty} f_\alpha^k(s) = f_I^k(s), \quad \forall s \in \mathbb{R}$

Proof. Without loss of generality, we prove the case for $k = 0$.

$$\begin{aligned}\lim_{\alpha\to\infty} f_\alpha^0(s) &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{\cosh(\alpha s)}{\cosh(\alpha(-s+1))}\right) + \frac{1}{2} \\ &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{\alpha s} + e^{-\alpha s}}{e^{\alpha(-s+1)} + e^{\alpha(s-1)}}\right) + \frac{1}{2}\end{aligned}$$

Case 1. $s < 0$

$$\begin{aligned}\lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{\alpha s} + e^{-\alpha s}}{e^{\alpha(-s+1)} + e^{\alpha(s-1)}}\right) + \frac{1}{2} &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{2\alpha s} + 1}{1 + e^{2\alpha(s-1)}} \cdot \frac{e^{-\alpha s}}{e^{\alpha(-s+1)}}\right) + \frac{1}{2} \\ &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \cdot (-\alpha) + \frac{1}{2} \\ &= 0\end{aligned}$$

Case 2. $0 \le s < 1$

$$\begin{aligned}\lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{\alpha s} + e^{-\alpha s}}{e^{\alpha(-s+1)} + e^{\alpha(s-1)}}\right) + \frac{1}{2} &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{-2\alpha s} + 1}{1 + e^{2\alpha(s-1)}} \cdot \frac{e^{\alpha s}}{e^{\alpha(-s+1)}}\right) + \frac{1}{2} \\ &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \cdot \alpha(2s-1) + \frac{1}{2} \\ &= s\end{aligned}$$

Case 3. $s \ge 1$

$$\begin{aligned}\lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{\alpha s} + e^{-\alpha s}}{e^{\alpha(-s+1)} + e^{\alpha(s-1)}}\right) + \frac{1}{2} &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \log\left(\frac{e^{-2\alpha s} + 1}{e^{2\alpha(-s+1)} + 1} \cdot \frac{e^{\alpha s}}{e^{\alpha(s-1)}}\right) + \frac{1}{2} \\ &= \lim_{\alpha\to\infty} \frac{1}{2\alpha} \cdot \alpha + \frac{1}{2} \\ &= 1\end{aligned}$$

□

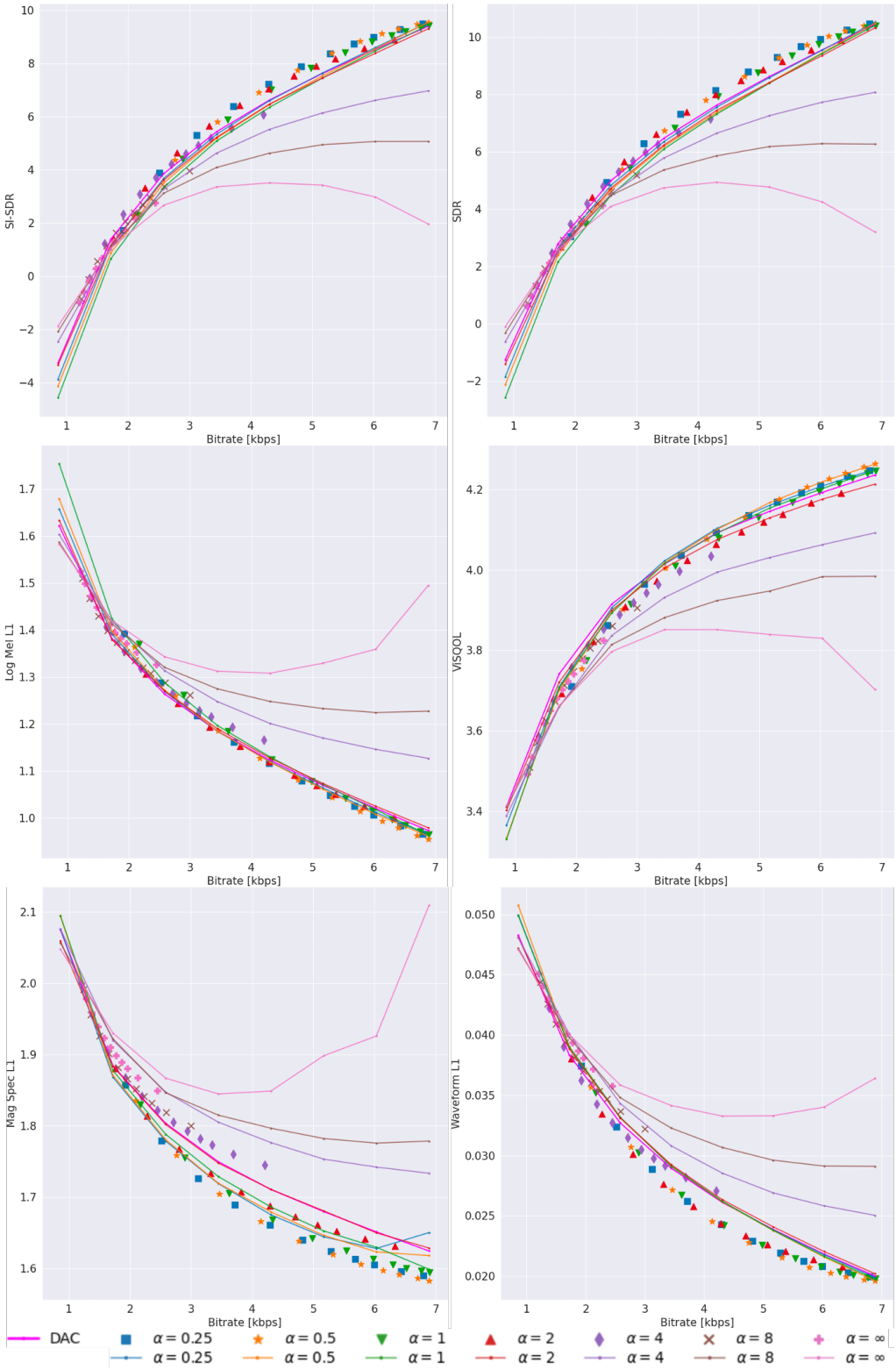
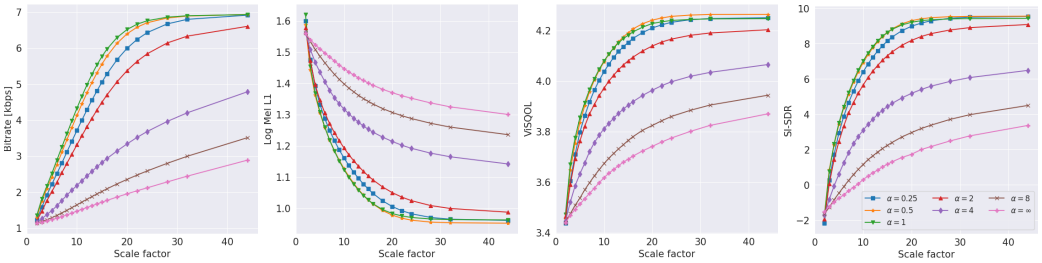


Figure 5: The results of VRVQ across different α values on various metrics. The scaling factors of VBR mode are the same as in Fig. 2. The solid lines represent the results inferred in CBR mode for each model.



(a) Bitrate (b) Log Mel L1 (c) ViSQOL (d) SI-SDR

Figure 6: Bitrate and various metrics as a function of scale factor in VBR mode

B Evaluation across additional metrics

In this section, we report additional results for the models presented in Fig. 2. We also plot the results for inference in CBR mode for the previously reported metrics. In addition to SI-SDR [21], ViSQOL [20], and Log Mel L1, we also report SDR, L1 loss on the raw waveform (Waveform L1), and L1 loss on the magnitude spectrogram (Mag Spec L1). For the Mag Spec L1 loss, we used multiscale STFT loss [29], with window sizes of 512 and 2048, and a hop size of 1/4 of those values.

As mentioned in the main text, in most cases, our proposed method outperforms both the Baseline and DAC for $\alpha < 2$, with particularly strong performance in waveform L1 loss and magnitude spectrogram L1 loss. As a reference, we also include plots of bitrate vs. scale factor, along with various metrics vs. scale factor, for the models inferred in VBR mode in Fig. 6.

C Ablation study

We conduct additional experiments to assess the impact of various configurations. First, we investigate whether allocating the full number of codebooks for a subset of samples in each mini-batch during training, as proposed in [4], could improve rate-distortion tradeoff, rather than solely relying on importance map-based allocation. Thus, we conduct ablation studies with $\alpha = 2$, where the performance is slightly worse than smaller α values at higher bitrates. Specifically, for 25% of the samples in each mini-batch, we use the full set of codebooks regardless of the importance map. Additionally, we investigate the effect of removing LUFS normalization preprocessing during training. We also experiment with increasing the number of codebooks to 16 to assess the scalability of the proposed method with respect to the maximum number of the codebooks. The corresponding increase in additional transmission cost for VBR is 0.344 kbps. For consistency, a DAC model is also trained from scratch using the same configuration. All ablation studies adhere to the settings outlined in Section 4.2.

The results for the first studies are shown in Fig. 7. From the results, we can observe that using the full codebooks for a subset of samples does not provide a significant advantage for SI-SDR, SDR, Log Mel L1 loss and Waveform L1 loss. This is likely because, when $\alpha = 2$, the model have observed a sufficiently wide range of the codebook usage cases during training, simply by varying the scaling factor l. Instead, it results in the performance degradation at lower bitrates. On the other hand, the models trained without LUFS normalization shows a significant performance improvement in SI-SDR, SDR, ViSQOL, and Waveform L1 loss, but not in Log Mel L1 loss. This suggests that the importance subnetwork may have generated a higher-quality importance map by learning the energy distribution of frames.

For $N_q = 16$, VRVQ shows better performance than DAC in SI-SDR, SDR, and Waveform L1 loss similar to the results for $N_q = 8$, as shown in Fig. 8. In particular, the CBR performance of VRVQ almost matched that of DAC in SI-SDR, SDR and Waveform L1 loss. Although not all metrics outperformed the DAC, such as Log Mel L1, ViSQOL, and Mag Spec L1 loss, the VRVQ still showed comparable results to the DAC. This shows the scalability that our method can be applied to higher bitrates without impairing the performance of existing CBR codecs.

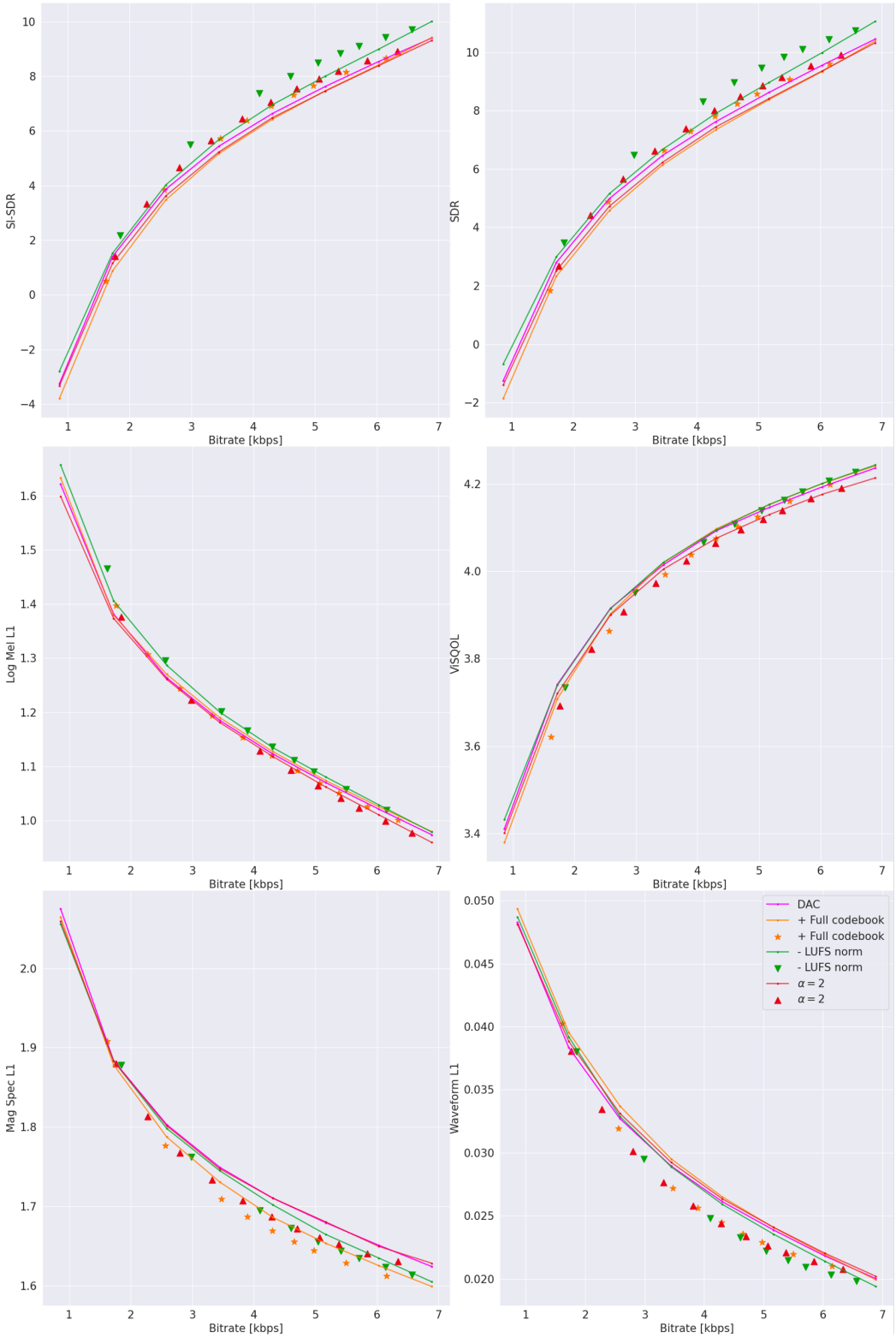


Figure 7: The results for ablation studies. The scaling factors of VBR mode are the same as in Fig. 2.

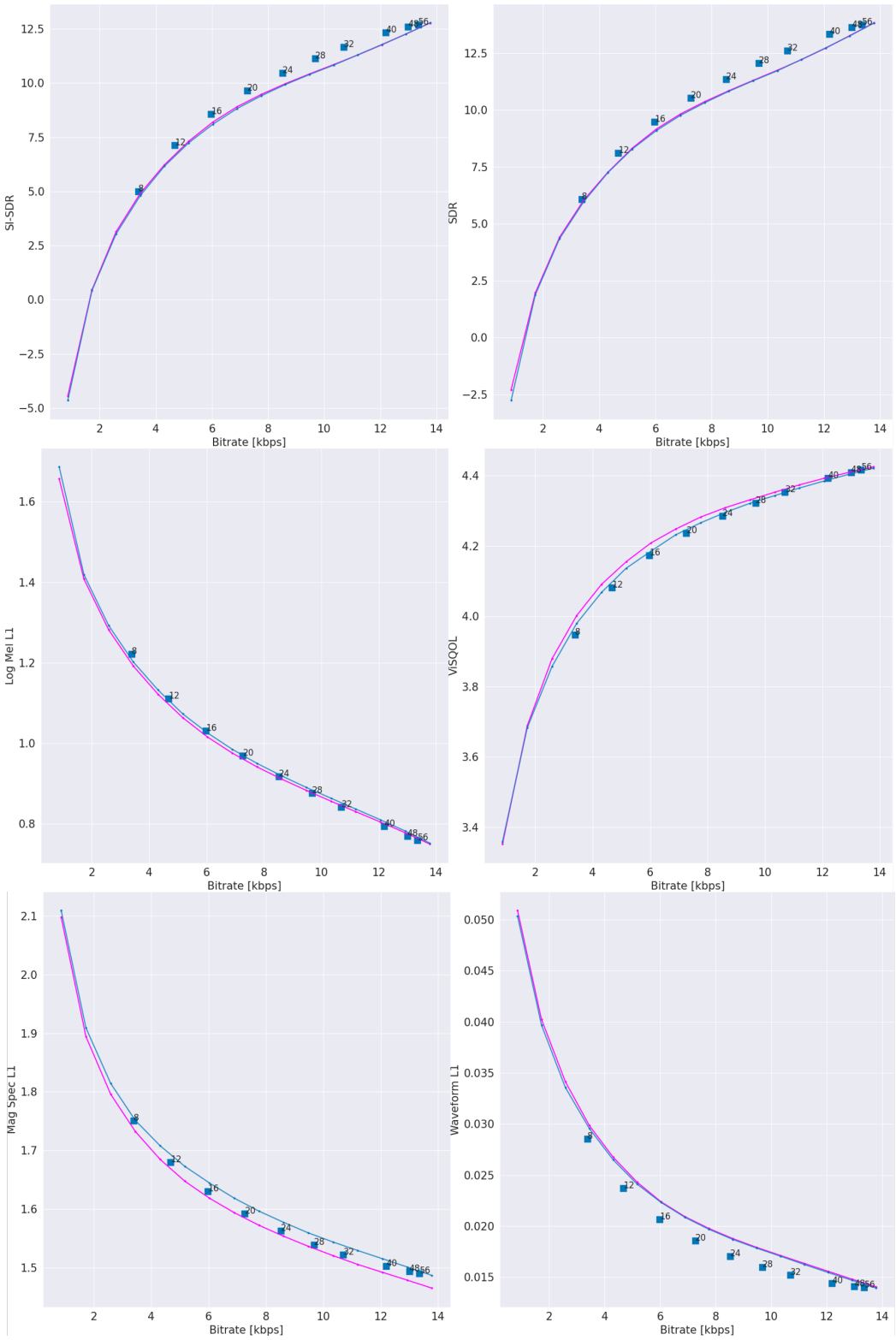


Figure 8: Results for the model trained with 16 codebooks ($N_q = 16$) shown in blue. The scaling factors l for VBR mode are set to 8, 12, 16, 20, 24, 28, 32, 40, 48, and 56, with each scaling factor annotated above the corresponding points. The DAC model, shown in magenta, is trained with the same configuration.

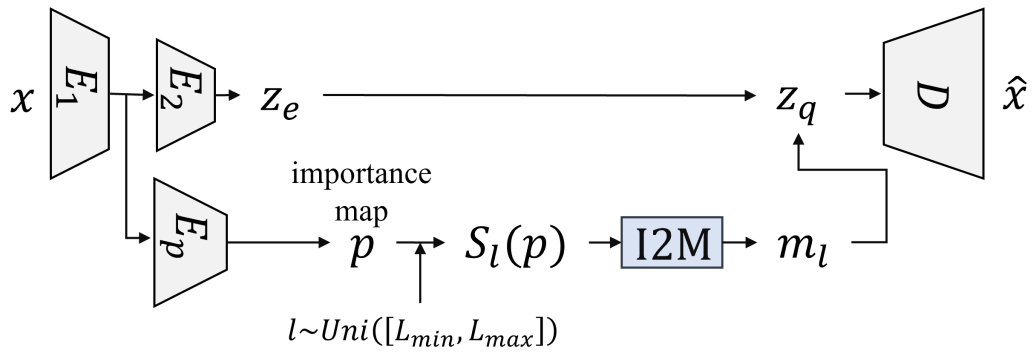


Figure 9: Simplified overall framework graph of VRVQ

D Independence of rate loss from random sampling

Typically, rate control is achieved by setting different weights β for the rate loss, which leads to the inconvenience of deploying a separate model for each β. Some studies have addressed this by allowing the model to respond to changes in β by incorporating β as an input condition [30, 14, 31–34]. In our research, we avoided inputting β as a condition into the model. Instead, we fixed β and introduced an external operator S_l that is randomly sampled to control the rate loss. This section justifies our approach, demonstrating that rate loss can be formulated in such a way that the sampled l does not influence it—in other words, proving that the rate loss and l are independent.

Typically, the β-VAE objective [16] is formulated with a reconstruction loss and a KL divergence term, also known as the rate loss, as follows:

$$\mathcal{L}_\beta(\phi, \theta) = \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{\mathrm{KL}}(q_\phi(z|x)||p_r(z)) \tag{11}$$

In our case, z can be viewed as a joint variable of the latent variable z_e, the importance map p, and the scaling factor l, i.e., $z = (z_e, p, l)$. Once z_e, p, and l are determined, the mask m_l is also determined, allowing the decoder input z_q to be calculated as shown in Fig. 9. The first term corresponds to the reconstruction loss, which is calculated using the reconstructed audio with the masked latent z_q. Therefore, we focus on maximizing the second term, the rate loss, which can be expanded as follows:

$$\min_\phi \int_z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_r(z)} dz \tag{12}$$

$$= \int_z q_\phi(z_e, p, l|x) \log \frac{q_\phi(z_e, p, l|x)}{p_r(z_e, p, l)} dz \tag{13}$$

$$= \int_z q_\phi(p, l|x) q_\phi(z_e|p, l, x) \log \left(\frac{q_\phi(p, l|x)}{p_r(p, l)} \cdot \frac{q_\phi(z_e|p, l, x)}{p_r(z_e|p, l)} \right) dz \tag{14}$$

$$= \mathbb{E}_{q_\phi(p,l|x) q_\phi(z_e|p,l,x)} \left[\log \frac{q_\phi(p, l|x)}{p_r(p, l)} \right] \tag{15}$$

$$+ \mathbb{E}_{q_\phi(p,l|x)} \left[D_{KL}(q_\phi(z_e|p, l, x)||p_r(z_e|p, l)) \right] \tag{16}$$

Due to the independence of l from other variables and the conditional independence of z_e and p given x, Eq. (15) can be simplified as follows:

$$\mathbb{E}_{q_\phi(p,l|x) q_\phi(z_e|p,l,x)} \left[\log \frac{q_\phi(p, l|x)}{p_r(p, l)} \right] = \mathbb{E}_{q_\phi(p|x) p_r(l) q_\phi(z_e|x)} \left[\log \frac{q_\phi(p|x)}{p_r(p)} \right] \tag{17}$$

$$= \mathbb{E}_{q_\phi(p|x)} \left[\log \frac{q_\phi(p|x)}{p_r(p)} \right] \tag{18}$$

$$= D_{\mathrm{KL}}(q_\phi(p|x)||p_r(p)) \tag{19}$$

For the second term, Eq. (16), we have:

$$\mathbb{E}_{q_\phi(p,l|x)}\left[D_{\mathrm{KL}}(q_\phi(z_e|p,l,x)||p_r(z_e|p,l))\right] = \mathbb{E}_{q_\phi(p|x)p_r(l)}\left[D_{\mathrm{KL}}(q_\phi(z_e|x)||p_r(z_e|p))\right] \tag{20}$$

$$= \mathbb{E}_{q_\phi(p|x)}\left[D_{\mathrm{KL}}(q_\phi(z_e|x)||p_r(z_e|p))\right] \tag{21}$$

Consequently, both terms Eq. (19) and Eq. (21)) are independent of l, meaning the calculation of the rate loss is unaffected by l.

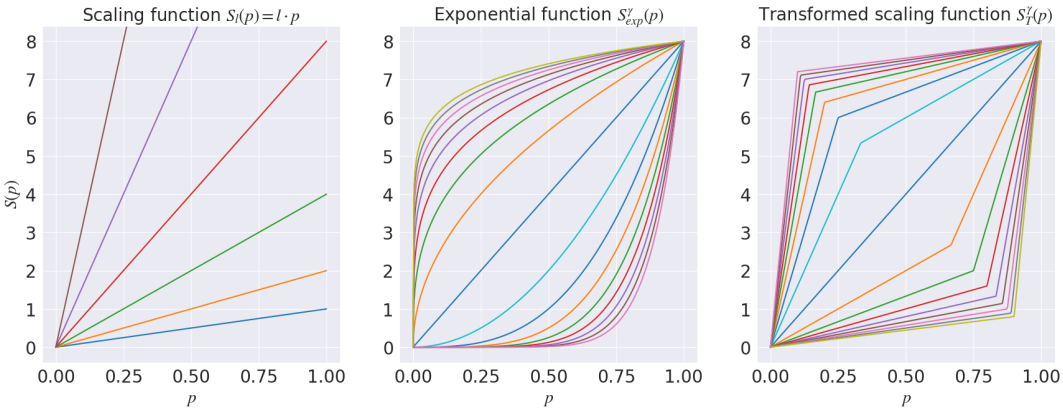
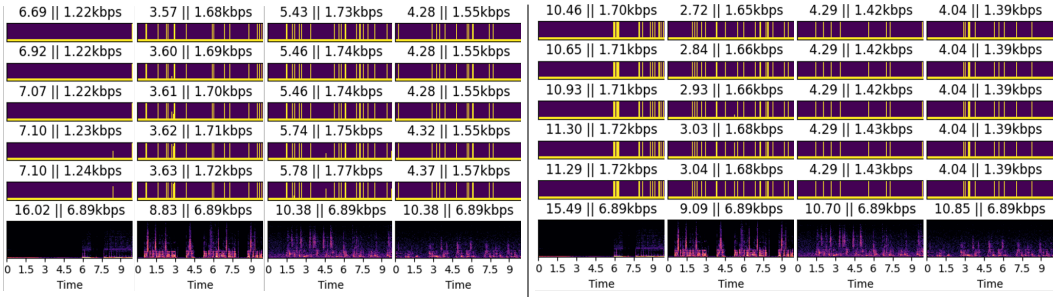


Figure 10: Comparison of S_l and other alternative scaling functions



(a) Samples trained on S^{γ}_{exp}

(b) Samples trained on S^{γ}_{T}

Figure 11: Visualization of codebook usage based on alternative functions. In each plot, the bitrate of the mask and the corresponding SI-SDR are noted. The bottom row displays the reconstructed samples using the full number of codebooks. From the first row, the γ values are set to 0.1,0.5, 1, 4, and 8

E Exploration of alternative scaling operators

We randomly sample the scale factor l and apply the corresponding operator S_l to the importance map p, as shown in Fig. 9 . In this work, $S_l(p) = l \cdot p$ is implemented as a simple scaling function. However, this approach presents a limitation: if the sampled scale factor is less than $N_q - 1$, the upper bound of codebook usage is constrained to that value.

To address this limitation, we have explored alternative operators as shown in Fig. 10, such as an exponential function $S^{\gamma}_{\text{exp}}(p) = N_q \cdot p^{\gamma}$, inspired by [14], and a *Transformed scaling function*, which we define below. The expectation is that the gradient around 0 will neither be infinite nor zero unlike S_{exp}, thereby stabilizing the training process:

$$S^{\gamma}_{T}(x) = \begin{cases} N_q \gamma x & \text{if } 0 < x \leq 1/(\gamma+1), \\ \dfrac{N_q}{\gamma}(x + \gamma - 1) & \text{if } 1/(\gamma+1) < x < 1 \end{cases} \tag{22}$$

Despite experimenting with these alternative operators, the importance map is not trained effectively, as shown in Fig. 11. We set $\alpha = 2$ and sampled the random value γ from a log-uniform distribution: $\log \gamma \sim U([\log 0.1, \log 6])$ during training. The results indicate only a few extreme spikes of codebook usage in some time frames, but for most frames, the model consistently uses a single codebook, thereby limiting the performance of the VRVQ framework.

Addressing this issue remains part of future work, as these alternative operators have the potential to avoid limiting the upper bound of codebook usage at low γ, which is a drawback of the simple scaling function S_l used in this study.