NANOCODEC: TOWARDS LOW BITRATE AND LOW COMPLEXITY REAL-TIME NEURAL AUDIO CODEC

Andong Li[⋆*], *Pinglin Xu*[†], *Zhe Han*[†], *Lingling Dai*[⋆*], *Yiqing Guo*[†], *Hua Gao*[†], *Xiaodong Li*[⋆*], Chengshi Zheng[⋆*]

⋆ Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
†ByteDance, China
*University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

In this report, we present NanoCodec, our submitted system for the LRAC Challenge Track 1, which can effectively reconstruct target waveform under ultra-low and low bitrates conditions. Specifically, our architecture operates in the time-frequency (T-F) domain, where we drop the phase and only encode the magnitude feature in the encoder side, and both are estimated in the receiver side. In addition, we propose an efficient convolution-style attention block as the core modeling unit. Given the strict constraint on the decoder complexity, the omnidirectional phase and real-imaginary losses are introduced to enable the effective joint optimization of target magnitude and phase. The submitted system achieves a total latency of 30 ms and a computational complexity of 685 MFlops (390M for the encoder and 295M for the decoder), satisfying the challenge requirements.

Index Terms— Neural audio codec, low-complexity, low bitrate, real-time, speech transmission

1. INTRODUCTION

Audio codecs are designed to convert original waveforms into compact bitstreams for transmission, followed by target decoding at the receiver. In recent years, neural audio codecs (NACs) have surged in popularity, propelled by the advancement of large language models (LLMs). Compared to traditional methods, NACs offer both higher compression ratios and reconstruction quality over [1, 2]. However, while most studies leverage NACs as audio tokenizers for generation tasks, real-time audio transmission remains underexplored [1, 3], where computational cost, causality, and algorithmic delay are regarded as significant factors to hinder the deployment of NACs in practical transmission scenarios.

LRAC Chalenge 2025 aims to gather research attention in real-time (RT) audio transmission under strict constraints on training dataset, calculation complexity and processing delay[1]. Specifically, Track 1 is devised for transparent transmission, with a maximum complexity of 700 MFlops (400 M for the encoder and 300 M for the decoder), and a total latency $\leq$ 30ms. To our best knowledge, existing literature rarely satisfies these requirements, thus posing a significant challenge for neural audio codec design.

To this remedy, in this paper, we present the proposed **NanoCodec**, which contributes in both architecture design and optimization regime. First, the proposed codec is based on time-frequency (T-F) domain, where we ignore the phase and only magnitude is utilized for feature encoding, and both targets are reconstructed in the decoder. The rationale lies in that given the limited calculation resource, it seems challenging for target coding or estimation in the time domain. As such, we employ the Fourier prior to alleviate the learning difficulty. Besides, given the limited bit resource, separate phase encoding can be trivial due to the wrapping effect of phase component. Second, we adopt a convolution-style attention block for spectral modeling, where the attention distribution is generated via large convolution kernels to effectively aggregate the contextual information. Third, it remains an open question for joint magnitude and phase estimation, especially under limited calculation resource. Motivated by [4], we employ an omnidirectional phase loss for phase optimization, efficiently capturing differential relations between centering and neighboring phase bins. we further generalize it into the real and imaginary (RI) parts of the spectrum, and propose an omnidirectional RI loss. By incorporating the above-mentioned tactics together, NanoCodec can reconstruct waveforms with high-quality under both low complexity and low bitrate scenarios.

[1]https://crowdsourcing.cisco.com/lrac-challenge/2025/

2. METHOD ILLUSTRATIONS

2.1. Overall Architecture

The overall diagram of the proposed NanoCodec is presented in Fig. 1(a), where both encoder and decoder are operated in the T-F domain. Given the input waveform $\mathbf{x} \in \mathbb{R}^L$, it is first transformed into the spectrum $\mathbf{X} \in \mathbb{C}^{F\times T}$ via the short-time Fourier transform (STFT), where $\{F, T\}$ denote the frequency and time axes, respectively. Different from previous literature where magnitude and phase are separately encoded [5], here we drop the phase and only preserve the magnitude for feature extraction. The reasons are two-fold. First, due to the restricted computational complexity in the encoder, as well as limited bit resource, the modeling priority should be provided to the magnitude as it exhibits more clear structural patterns over phase. Besides, phase usually exhibits random distribution due to the intrinsic wrapping effect, and it can be trivial for separate feature extraction from phase. Motivated by [6], the energy-content decoupling (ECD) layer is utilized to decouple the spectral energy and content, which is reported to mitigate the extra input energy normalization operation, given by:

$$\mathbf{I}_t = \text{Concat}\left(\log\left(E_t\right), \frac{|\mathbf{X}_t|}{E_t}\right) \in \mathbb{R}^{F+1}, \tag{1}$$

where E_t denotes the calculated energy for the t-th input frame, and Concat $(\cdot)$ is the concatenation operation along the feature dimension. After that, N_e modeling units are stacked for modeing.

For the decoder, similar to the encoder, N_d modeling units are stacked, and separate magnitude and RI heads are adopted for magnitude and phase estimation, respectively. After that, the inverse STFT operation is utilized for target waveform generation.

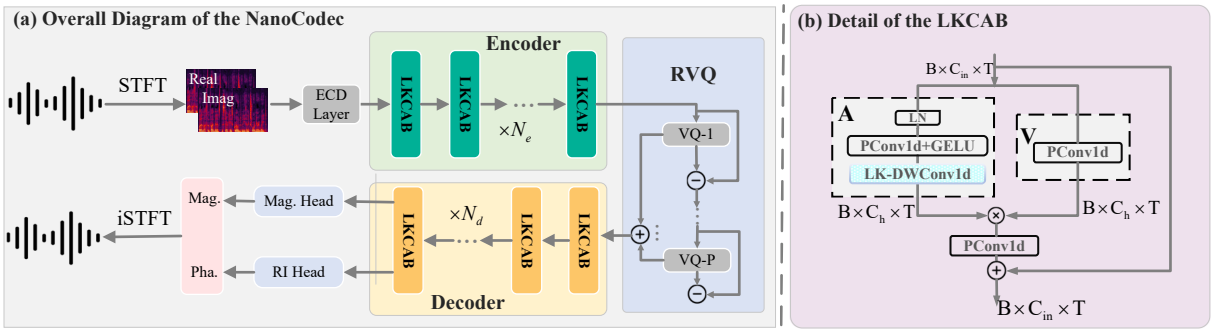

Fig. 1. (a) Overall structure of the proposed NanoCodec; (b) Internal structure of the adopted LKCAB.

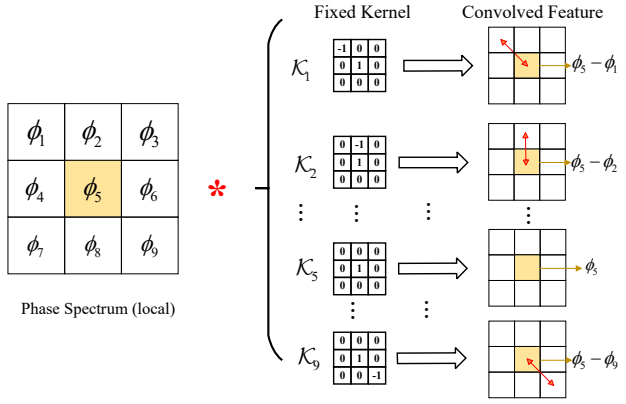

Fig. 2. Illustration of the omnidirectional phase loss.

2.2. Large Kernel Convolution-Style Attention Block

We share the same modeling unit for both encoder and decoder, and detailed internal structure is shown in Fig. 1(b). Given the input $\mathbf{H}_{i-1} \in \mathbb{R}^{B\times C_{in}\times T}$ of the i-th block, where C_{in} represents the input feature channel, it passes the attention branch and value branch to obtain the attention and value feature maps$\{\mathbf{A}_i, \mathbf{V}_i\} \in \mathbb{R}^{B\times C_h\times T}$, respectively. Here C_h indicates the hidden channel size. Motivated by [7], instead of adopting self-attention by calculating the pair-wise similarity scores, we enable it via a depth-wise convolution operation with large kernels (LD-DWConv1d) to enhance the processing efficiency. After that, a point-wise convolution (PConv1d) is adopted to return to the original input space, followed by residual connection. Note that, to reduce the overall computational complexity, we use the group-convolution for PConv1d. The causal setting is adopted, *i.e.*, the padding operation is only applied along the past frames, and no future information is involved. Formally, the process of the LKCAB can be formulated as:

$$\mathbf{A}_i = \text{LK-DWConv1d}\left(\text{GELU}\left(\text{PConv1d}\left(\text{LN}\left(\mathbf{H}_{i-1}\right)\right)\right)\right), \quad (2)$$

$$\mathbf{V}_i = \text{PConv1d}\left(\mathbf{H}_{i-1}\right), \quad (3)$$

$$\mathbf{H}_i = \mathbf{H}_{i-1} + \text{PConv1d}\left(\mathbf{A}_i \otimes \mathbf{V}_i\right), \quad (4)$$

where "$\otimes$" denotes the element-wise multiplication operation.

3. MISCELLANEOUS CONFIGURATIONS

3.1. Loss Setups

We incorporate the reconstruction, adversarial, and perceptual losses for training. For the first term, we include the log-spectral amplitude loss $\mathcal{L}_a$, multi-resolution Mel loss $\mathcal{L}_m$, consistency loss $\mathcal{L}_{cons}$, phase loss $\mathcal{L}_p$, and RI loss $\mathcal{L}_{ri}$.

The amplitude loss evaluates the mean-square error (MSE) between $\left|\tilde{\mathbf{X}}\right|$ and $|\mathbf{X}|$ in the log-domain:

$$\mathcal{L}_a = \frac{1}{FT}\sum_{f,t}\left\|\log\left|\tilde{\mathbf{X}}_{f,t}\right| - \log\left|\mathbf{X}_{f,t}\right|\right\|_2^2. \quad (5)$$

Inconsistency can arise when the generated spectrum in the T-F domain is not necessarily equal to the STFT of it time-domain counterpart [8]. To mitigate this issue, the consistent spectrum is defined as $\hat{\mathbf{S}} = \text{STFT}\left(\text{iSTFT}\left(\tilde{\mathbf{S}}\right)\right)$, and consistency loss is given by:

$$\mathcal{L}_c = \frac{1}{FT}\sum_{f,t}\left(\left\|\mathcal{R}(\tilde{\mathbf{S}}_{f,t}) - \mathcal{R}\left(\hat{\mathbf{S}}_{f,t}\right)\right\|_2^2 + \left\|\mathcal{I}(\tilde{\mathbf{S}}_{f,t}) - \mathcal{I}(\hat{\mathbf{S}}_{f,t})\right\|_2^2\right). \quad (6)$$

Motivated by [9], we use multi-resolution Mel loss, which was reported to yield better performance over the single-resolution version, given by:

$$\mathcal{L}_{mel} = \frac{1}{FTS}\sum_{f,t}\sum_{s}\left\|\tilde{\mathbf{X}}_{f,t}^{mel,(s)} - \mathbf{X}_{f,t}^{mel,(s)}\right\|_1, \quad (7)$$

where $\left\{\tilde{\mathbf{X}}^{mel}, \mathbf{X}^{mel}\right\}$ are the estimated and target Mel spectra, respectively. $(\cdot)^{(s)}$ denotes the Mel spectrum under the s-th resolution scale. Here seven window sizes are adopted: {32, 64, 128, 256, 512, 1024, 2048}, and hop length set to window_length / 4. Besides, we use mel bin sizes $\{5, 10, 20, 40, 80, 160, 320\}$.

Motivated by [4], we employ an omnidirectional phase loss, as shown in Fig. 2. To be specific, a specially devised kernel $\mathcal{K} \in \mathbb{R}^{9\times3\times3}$ is applied to the estimated and target phase, to obtain the omnidirectional differential between the centering and neighboring phase bins:

$$\hat{\tilde{\mathbf{\Phi}}}_{est} = \tilde{\mathbf{\Phi}} * \mathcal{K}, \hat{\mathbf{\Phi}} = \mathbf{\Phi} * \mathcal{K}, \quad (8)$$

where "*" denotes the convolution operation, and $\left\{\hat{\tilde{\mathbf{\Phi}}}, \hat{\mathbf{\Phi}}\right\} \in \mathbb{R}^{9\times F\times T}$ are the convolved results for estimated and target phase, respectively. The phase loss can be calculated as:

$$\mathcal{L}_p = \frac{1}{FT}\sum_{f}\sum_{t}\left\|\hat{\tilde{\mathbf{\Phi}}} - \hat{\mathbf{\Phi}}\right\|_1. \quad (9)$$

We further generalize it into the RI loss. Concretely, we first decouple the magnitude and phase, then the omnidirectional operation is employed to extract the differential phase representation, *i.e.*,

$\left\{\hat{\tilde{\mathbf{\Phi}}}, \hat{\mathbf{\Phi}}\right\}$. The corresponding omnidirectional RI loss can be defined as:

$$\mathcal{L}_{ri} = \frac{1}{FT}\sum_{f}\sum_{t}\Bigg(\left\|\text{Rep}\left(\left|\tilde{\mathbf{X}}\right|\right)\cos\left(\hat{\tilde{\mathbf{\Phi}}}\right) - \text{Rep}\left(|\mathbf{X}|\right)\cos\left(\hat{\mathbf{\Phi}}\right)\right\|_1 + \left\|\text{Rep}\left(\left|\tilde{\mathbf{X}}\right|\right)\sin\left(\hat{\tilde{\mathbf{\Phi}}}\right) - \text{Rep}\left(|\mathbf{X}|\right)\sin\left(\hat{\mathbf{\Phi}}\right)\right\|_1\Bigg), \tag{10}$$

where $\text{Rep}\left(\cdot\right)$ denotes the tensor repeat operation, *i.e.*, $\mathbb{R}^{1\times F\times T} \rightarrow \mathbb{R}^{9\times F\times T}$. The overall reconstruction loss $\mathcal{L}_{recon}$ can be defined as:

$$\mathcal{L}_{recon} = \lambda_a\mathcal{L}_a + \lambda_c\mathcal{L}_c + \lambda_{mel}\mathcal{L}_{mel} + \lambda_p\mathcal{L}_p + \lambda_{ri}\mathcal{L}_{ri}, \tag{11}$$

where $\{\lambda_a, \lambda_c, \lambda_{mel}, \lambda_p, \lambda_{ri}\}$ are the corresponding weighting hyper-parameters, and set to $\{45.0, 20.0, 45.0, 50.0, 45.0\}$, respectively.

For adversarial loss, we incorporate the multi-period discriminator (MPD) [10], multi-resolution STFT discriminator (MRSTFTD) [5], and multi-band discriminator (MBD) [9], and the hinge loss is adopted to calculate the adversarial loss. For each sub-discriminator in MPD, the 1-D raw audio waveform is reshaped into 2-D format with period p, then processed through consecutive Conv2D layers and leaky ReLU for score computation. The periods are set to $\{2, 3\}$[2]. For MRD, three sub-discriminators process magnitude spectra via stacked Conv2d layers to calculate the discriminative score. The {window_size, hop_size, nfft} are set to $(128, 32, 128)$, $(256, 64, 256)$, $(512, 128, 512)$, $(1024, 256, 1024)$, and $(2048, 512, 2048)$, respectively. For MBD, we divide the overall spectrum into five band regions: {(0, 0.1), (0.1, 0.25), (0.25, 0.5), (0.5, 0.75), (0.75, 1.0)}. The {window_size, hop_size, nfft} are set to $(256, 64, 256)$, $(512, 128, 512)$, $(1024, 256, 1024)$, and $(2048, 512, 2048)$, respectively. The trainable parameters of the three discriminators are 3.4 M, 6.3 M, and 7.5 M, respectively. The weighting hyper-parameters of the adversarial and feature-matching losses are set to 1.0, 2.0, respectively.

Besides, the feature matching loss is also incorporated. For perceptual-based loss, to promote the performance on objective metrics, we include the PESQ loss[3] and UTMOS loss[4] for optimization. We also utilize the pre-trained SCOREQ model[5] and maximize the output similarity score between the estimation and target waveforms. Note that, to accelerate the network training, we only add the perceptual loss in the finetune stage, and the weighting hyper-parameters $\{\lambda_{pesq}, \lambda_{utmos}, \lambda_{scoreq}\}$ are set to $\{5.0, 5.0, 5.0\}$, respectively.

3.2. Dataset Setups

For codec training, we use the speech clips from LibriSpeech [11], DNS-Challenge [12], VCTK [13] and EARS [14]. Note that we did not use the CommonVoices [15] due to its relatively low quality. For noise set, we include DNS-Challenge noise set[6], WHAM! [16] and FSD50K [17]. For reverberation generation, we include the RIRs from Open SLR 28[7] and our synthesized 100 k RIR clips. To adapt to practical acoustic scenarios, we adopt the on-the-fly (OTF) training strategy, that is, we randomly combine noise and reverberation during the training process. For noise, the average SNR value is 15 dB, with the variance of 7.5 dB. The probability to include noise and reverberation are 0.15 and 0.15, respectively. We also include the multi-speaker case[8] with the overlap ratio randomly sampled in the range of $[0.5, 0.95]$, and the probability is set to 0.15. To mitigate the possible audio clip, we randomly rescale the waveform value from the range of $[0.218, 0.917]$. No other data augmentation strategies are adopted. All training clips are chunked to 2.0 second to stabilize the training.

3.3. Network Setups

For both STFT and iSTFT, the target sampling rate is 24 kHz. The window size is set to 30 ms, with 10 ms overlap between adjacent frames. 720-point FFT is adopted, leading to 361-D input features. Thus, the overall system latency is 10 + 20 = 30 ms, which satisfies the challenge rule. For network encoder, the input and hidden channel $\{C_{in}, C_h\}$ are set to $\{372, 372\}$, and $N_e = 6$ blocks are adopted. For the decoder, the input and hidden channel $\{C_{in}, C_h\}$ are set to $\{260, 360\}$, and $N_d = 6$ are adopted. For both sides, we set the kernel size of the LK-DWConv1d to 7, and the number of groups for PConv1d is set to 2 to reduce the computational complexity. For the quantization process, motivated by [9], we adopt the factorized quantizer, and the codebook dimension is set to 8. For 1 kbps and 6 kbps settings, $\{1, 6\}$ codebooks are utilized, respectively, and the codebook size is set to 1024. As a result, the average computational complexity of the encoder and decoder are around 390.18 MFlops (including 8.76 MFlops for quantization) and 295.12 MFlops. The trainable parameters of the encoder and decoder are 2.04 M and 1.48 M, respectively.

3.4. Training Setups

The training is based on the Pytorch-Lightning platform, and Two NVIDIA A100 are employed. The total batch size is 32, and we train the network for 1.5 M steps in total, where the discriminators are updated per three steps to reduce the GPU assumption. For the first 1.2 M steps, only reconstruction loss and adversarial loss are adopted. After that, we incorporate the perceptual loss in the remaining finetune stage. The AdamW optimizer [18] is employed, and the learning rate is initialized at 2e-4, with the exponential decay in the batch level, and the decay rate is set to 0.999996. Besides, the exponential moving average (EMA) strategy for generator update, and the decay rate is set to 0.999.

4. REFERENCES

[1] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[2] Haohe Liu, Xuenan Xu, Yi Yuan, Mengyue Wu, Wenwu Wang, and Mark D Plumbley, "Semanticodec: An ultra low bitrate semantic audio codec for general sound," *IEEE J. Sel. Top. Signal Process.*, 2024.

[3] Yi-Chiao Wu, Israel D Gebru, Dejan Marković, and Alexander Richard, "Audiodec: An open-source streaming high-fidelity neural audio codec," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[4] Andong Li, Tong Lei, Zhihang Sun, Rilin Chen, Erwei Yin, Xiaodong Li, and Chengshi Zheng, "Learning neural vocoder from range-null space decomposition," *arXiv preprint arXiv:2507.20731*, 2025.

[2] We empirically observe that more period settings can damage the performance in the light-weight audio codec design.

[3] https://github.com/audiolabs/torch-pesq

[4] https://github.com/tarepan/SpeechMOS/tree/main

[5] https://github.com/alessandroragano/scoreq

[6] https://github.com/microsoft/DNS-Challenge

[7] https://www.openslr.org/28/

[8] In practical synthesis, we only consider the 2-speakers remixing case.

[5] Yang Ai, Xiao-Hang Jiang, Ye-Xin Lu, Hui-Peng Du, and Zhen-Hua Ling, "Apcodec: A neural audio codec with parallel amplitude and phase spectrum encoding and decoding," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 3256–3269, 2024.

[6] Yi Luo, Jianwei Yu, Hangting Chen, Rongzhi Gu, and Chao Weng, "Gull: A generative multifunctional audio codec," *arXiv preprint arXiv:2404.04947*, 2024.

[7] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng, "Conv2former: A simple transformer-style convnet for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 8274–8283, 2024.

[8] Scott Wisdom, John R Hershey, Kevin Wilson, Jeremy Thorpe, Michael Chinen, Brian Patton, and Rif A Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. ICASSP*. IEEE, 2019, pp. 900–904.

[9] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar, "High-fidelity audio compression with improved rvqgan," *Proc. NeurIPS*, vol. 36, pp. 27980–27993, 2023.

[10] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, pp. 17022–17033, 2020.

[11] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[12] Jianwei Yu, Hangting Chen, Yi Luo, Rongzhi Gu, Weihua Li, and Chao Weng, "Tspeech-ai system description to the 5th deep noise suppression (dns) challenge," in *Proc. ICASSP*. IEEE, 2023, pp. 1–2.

[13] Junichi Yamagishi, "English multi-speaker corpus for CSTR voice cloning toolkit," `Lhttp://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html/`, 2012.

[14] Julius Richter, Yi-Chiao Wu, Steven Krenn, Simon Welker, Bunlong Lay, Shinji Watanabe, Alexander Richard, and Timo Gerkmann, "Ears: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation," in *Proc. Interspeech*, 2024, pp. 4873–4877.

[15] 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," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[16] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux, "Wham!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019, pp. 1368–1372.

[17] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "FSD50K: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[18] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.