SAuBER: NEURAL AUDIO CODING VIA A LEARNABLE MDCT TOKENIZER AND ENERGY-BASED REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative neural audio codecs (NACs) achieve high perceptual quality at low bitrates, but typically rely on Generative Adversarial Networks (GANs) and spatial downsampling. These choices introduce training instability and limit autoencoder reconstruction fidelity. While hybrid approaches avoid spatial downsampling by pairing a fixed signal-processing transform with a learned compressor, designing them independently creates a domain mismatch. To resolve this, we propose SAuBER, a unified NAC framework that co-designs a learnable transform with a neural compressor while replacing adversarial generation with energy-based refinement. Specifically, our analysis/synthesis module operates as a learnable tokenizer built directly over patchified Modified Discrete Cosine Transform (MDCT) coefficients, producing near-transparent tokens without spatial downsampling. Within this token space, an information-bottleneck autoencoder extracts discrete latent representations, and an energy-based refinement head, trained via vector field matching, fits the gradient of an explicit energy function for stable, streamable generation. Autoencoder and refinement head are trained together to jointly optimize feature extraction, quantization, and refinement. At bitrates of 20–40 kbps, SAuBER achieves competitive performance against both classical signal-processing codecs and state-of-the-art GAN-based NACs in terms of perceptual quality and reconstruction fidelity.

1 INTRODUCTION

Neural network-based codecs have rapidly advanced lossy compression for high-dimensional signals such as images, speech, and general audio, particularly at low bitrates. By replacing hand-crafted algorithms with neural networks, neural audio codecs (NACs) learn to map continuous audio into compact discrete representations and reconstruct the input signals. Recent NAC research focuses on extremely low bitrates (typically below 10 kbps), where reconstruction inherently becomes a generative task: the transmitted codes retain only coarse semantic information, delegating the synthesis of fine acoustic details to powerful generative frameworks. Consequently, state-of-the-art generative NACs heavily rely on Generative Adversarial Network (GAN) Goodfellow et al. (2014); Goodfellow (2016) and, more recently, diffusion models Ho et al. (2020); Song et al. (2021); Lipman et al. (2023) to synthesize perceptually convincing audio where classical rule-based codecs fail.

However, fundamental limitations prevent current NAC frameworks from scaling their performance gains to higher bitrates, where rule-based codecs remain competitive. Most conventional NACs Zeghidour et al. (2022); Défossez et al. (2023); Kumar et al. (2023) employ autoencoders directly to time-domain waveforms, learning the analysis and synthesis transforms from scratch Ballé et al. (2021), using aggressive temporal subsampling and spatial compression. Such architectures suffice for plausible synthesis at extremely low bitrates, but impose a structural ceiling once near-transparent reconstruction is required. Unlike classical transforms such as Short-Time Fourier Transform (STFT) and Modified Discrete Cosine Transform (MDCT), that guarantee perfect reconstruction by design , fully learned transforms offer no mathematical guarantees, rendering near-transparent reconstruction computationally expensive and difficult to optimize.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

One way to avoids this ceiling is to separate a fixed signal-processing transform from a learned compressor Lim et al. (2023; 2024); Villemoes et al. (2024); Jiang et al. (2024; 2026); Ai et al. (2024). This inherits perfect reconstruction of the transform and avoids subsampling in a learnable compressor, but since the transform is fixed, the compressor must operate in a domain designed for signal analysis rather than neural compression. Coefficients often spans orders of magnitude across bands and follows heavy-tailed, non-isotropic distributions, which is suboptimal to apply neural networks and generative models.

Meanwhile, the GAN training dynamics central to most NACs present inherent stability challenges. Min-max optimization is notoriously fragile and prone to converge failure Arjovsky & Bottou (2017); Mescheder et al. (2018), requiring delicate hyperparameter tuning and specialized normalization techniques Farnia et al. (2019). Furthermore, GAN-based audio synthesis remains heavily bottlenecked by the discriminator's architecture and capacity, which dictates both the ceiling of perceptual quality and the specific acoustic features the model can faithfully recover.

To overcome these limitations, several diffusion-based NACs have been proposed. In most cases, diffusion models are integrated at the decoder output Wu et al. (2024; 2025); Welker et al. (2025) or within the latent space Yang et al. (2024), operating as post-hoc generative modules on a frozen autoencoder and quantizer. Only a few approaches Pia et al. (2025); Zhang et al. (2024) train the diffusion process jointly with the underlying coding pipeline. Except for FlowDec Welker et al. (2025), these models synthesize audio from pure Gaussian noise conditioned on codec features, shifting signal reconstruction entirely onto the generative model. Finally, to the best of our knowledge, existing diffusion-based NAC lack streaming capability: they generate audio sequences across the full temporal context, which precludes real-time streaming and hinders autoregressive latent generation.

To address these challenges, we propose SAuBER (Subband Autoencoding via Bottlenecked Energy Refinement), a streamable NAC framework that unites a near-transparent learnable transform with an iterative generative refinement module through a three-phase architecture. First, rather than learning an analysis–synthesis transform from scratch, we introduce a learnable suband tokenizer operating directly on patchified MDCT coefficients. Trained under a noisy channel with full-resolution time–frequency loss and shape regularization, this tokenizer maps audio into a degradation-robust token space while preserving near-transparent fidelity. Second, an autoencoder with an information bottleneck over the codeword dimension, rather than across spatial time-frequency axis, is trained within this token space using joint reconstruction and target-rate-matching objectives. Third, we formulate compression and synthesis using an Energy-Based Model (EBM) paradigm, where a refinement head estimates the gradient field of an explicit energy function to iteratively refine decoder outputs at inference. Unlike conventional approaches that generate audio from pure noise or operating post-hoc on a frozen backbone, the refinement head and autoencoder are jointly optimized to complement one another. Finally, by conducting this energy-based refinement frame-by-frame within a strictly causal context, SAuBER enables streamable decoding. Empirical evaluations across various energy function instantiations demonstrate that SAuBER outperforms classical rule-based codecs and state-of-the-art GAN-based NACs at target bitrates of 20–40 kbps.

2 Energy-Based Models

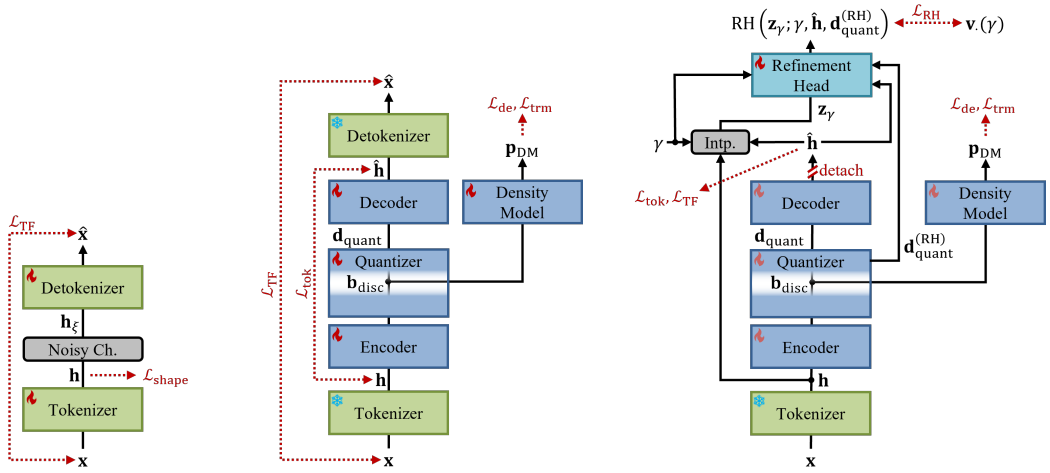
Energy-Based Model (EBMs) represent data probability distributions through an unnormalized scalar energy function, assigning lower values to plausible data configurations LeCun et al. (2006). Classical EBMs are typically trained via maximum likelihood estimation Hinton (2002); Welling & Teh (2011), requiring few-step iteration to sample from model distribution.

Diffusion models Ho et al. (2020); Song et al. (2021); Lipman et al. (2023); Wang & Du (2026) can be interpreted as a practical form of EBM. For instance, score matching Song et al. (2021) is equivalent to learning the gradient of the negative log density of the noisy variable.

$$s_\theta(x_t, t) \approx \nabla_{x_t} \log p_t(x_t) = -\nabla_{x_t} E(x_t, t), \quad E(x_t, t) = -\log p_t(x_t), \tag{1}$$

where s_θ is score estimator, x_t is noisy variable, and t denotes the noise schedule. Recently, Equilibrium Matching (EqM) Wang & Du (2026) adopts the interpretation of EBM into the Flow Matching Lipman et al. (2023), constructing an equilibrium gradient field estimator

$$v_\theta(x_t) \approx \nabla_{x_t} E(x_t) = c(t)(\epsilon - x), \quad E(x_t) = \frac{c(t)}{2(1-t)} \|x_t - x\|_2^2, \tag{2}$$



(a) Subband tokenizer (b) Bottlenecked autoencoder (c) Refinement head

Figure 1: SAuBER is trained through three consecutive stages. (a) First, train near-transparent and degradation-robust subband tokenizer. (b) Second, in the fozen token space, train an autoencoder equipped with information bottleneck. (c) Finetune the bottlenecked autoencoder jointly with a refinement head, which predicts the gradient of energy function with respect to its input variable $\mathbf{z}_\gamma$.

where x is a target variable, $x_t = tx + (1-t)\epsilon$ is a noisy variable given $\epsilon \sim \mathcal{N}(0, 1)$, and $c(t)$ is schedule-dependent weight to make target vector field diminish near the target.

3 SAuBER: Subband Autoencoding via Bottlenecked Energy Refinement

SAuBER consists of three components: a learnable subband tokenizer built on the MDCT, a token-space autoencoder equipped with an information bottleneck, and an energy-based refinement head. The tokenizer inserts learnable transform and inverse transform layers between the MDCT and its inverse, seamlessly reshaping the latent distribution to render it suitable for the subsequent autoencoder and generative models. Trained on signals with additive noise while maintaining near-transparent quality, it demonstrates robustness to the unknown degradations induced by the autoencoder and refinement head. Within this transformed latent space, we train an autoencoder, consisted of an encoder, a quantizer, a density model, and a decoder, to reconstruct the input signal under various rate-constrained regularization. Following the decoder, we attach a refinement head to estimate the gradient of an explicit energy function; which iteratively refines the decoder output along the learned gradient field during inference. Because this energy-based refinement process operates without adversarial (GAN) training, our framework avoids associated training instabilities.

3.1 Subband Tokenizer

The objective of the subband tokenizer is to preserve the high reconstruction quality of the MDCT while ensuring the token space remains robust against downstream degradations introduced by the bottlenecked autoencoder and refinement head. To this end, we carefully design the learnable transformation layers ($\mathbf{F}_{\text{tok}}, \mathbf{F}_{\text{detok}}$), the loss function, and the overall training strategy.

Architecture. Figure 1a illustrates the architectures of the subband tokenizer and detokenizer. The subband tokenizer transforms a monaural audio signal $\mathbf{x} \in \mathbb{R}^{1\times T}$ of length T into continuous tokens $\mathbf{h} \in \mathbb{R}^{S\times F\times D}$, where S denotes the number of subbands, D the token dimensionality, and $F = \lfloor T/(SD) \rfloor$ the number of frames. The detokenizer acts as the corresponding pseudo-inverse.

An MDCT with window length $2N_{\text{bin}}$ first transforms the input audio into frames of coefficients $\text{MDCT}(\mathbf{x}, N_{\text{bin}}) \in \mathbb{R}^{N_{\text{bin}}\times \lfloor T/N_{\text{bin}} \rfloor}$. Given the number of subbands S and a frame length $N_{\text{frame}} = SD$, the coefficients are grouped into non-overlapping $(N_{\text{bin}}/S) \times (N_{\text{frame}}/N_{\text{bin}})$ patches. From

every N_{frame} audio samples, S patches of dimensionality D are produced and reshaped into $S \times F \times D$. Each subband is then scaled by a fixed equalizing factor $\mathbf{g} \in \mathbb{R}_+^S$, and the equalized patches are processed through a stack of feed-forward network (FFN) layers. The output of the last FFN layer is scaled by a learnable factor $\mathbf{k} \in \mathbb{R}_+^S$. The full tokenization is formulated as

$$\mathbf{k} \overset{s}{\odot} \mathbf{F}_{\text{tok}}(\mathbf{g} \overset{s}{\odot} \text{Patchify}(\text{MDCT}(\mathbf{x}, N_{\text{bin}}), S, N_{\text{frame}})) = \mathbf{T}_{\text{tok}}(\mathbf{x}) = \mathbf{h}, \tag{3}$$

where $\overset{s}{\odot}$ is band-wise multiplication. The detokenizer reverses this order:

$$\text{iMDCT}(\text{Patchify}^{-1}(\mathbf{g} \overset{s}{\oslash} \mathbf{F}_{\text{detok}}(\mathbf{k} \overset{s}{\oslash} \mathbf{h}), S, N_{\text{frame}}), N_{\text{bin}}) = \mathbf{T}_{\text{detok}}(\mathbf{h}) = \tilde{\mathbf{x}}. \tag{4}$$

where $\overset{s}{\oslash}$ is band-wise division.

Each FFN layer incorporates a residual skip connection and a LayerScale Touvron et al. (2021) initialized to a small value, rendering learnable tokenizer components near-identity. Moreover, because no spatial compression is applied, meaning the token dimensionality D is never reduced, the representation remains fully overcomplete without computational bottlenecking.

Robustness To enhance robustness against quantization errors and architectural non-linearities, we inject Gaussian noise into the clean token $\mathbf{h}$ during training. Specifically, for each clip, the clean token is perturbed with a probability of p_{nc}, where the noise scale σ is sampled uniformly from $[0, \sigma_{\max}]$ whenever noise is applied. The detokenizer input $\mathbf{h}_\xi$ is therefore

$$\mathbf{h}_\xi = \mathbf{h} + \mathbb{1}_{\text{nc}} \cdot \sigma\boldsymbol{\epsilon}, \quad \mathbb{1}_{\text{nc}} \sim \text{Bern}(p_{\text{nc}}),\ \sigma \sim \mathcal{U}(0, \sigma_{\max}),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

equivalently $\mathbf{h}_\xi \sim \mathcal{N}(\mathbf{h}, \mathbb{1}_{\text{nc}}\sigma^2\mathbf{I})$.

Training criteria We train the tokenizer to achieve both transparency and robustness: faithfully reconstructing the input audio under a clean channel while preserving an informative token space under degradation. The training objective is

$$\mathcal{L}_{\text{SBT}}(\mathbf{x}) = \mathcal{L}_{\text{TF}}(\mathbf{x}, \mathbf{T}_{\text{detok}}(\mathbf{h}_\xi)) + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}}(\mathbf{h}), \tag{6}$$

where $\mathcal{L}_{\text{TF}}$ is a time–frequency-domain reconstruction loss and $\mathcal{L}_{\text{shape}}$ is a shape regularization loss. While conventional NACs often compute reconstruction losses on mel-spectrograms, this approach is insufficient to preserve the perfect reconstruction property of the MDCT. Therefore, we construct the loss over linear-scale STFT coefficients using two criteria: a μ-law compression applied strictly to the magnitude, and a mean-squared error (MSE) loss evaluated over the real and imaginary components to preserve phase. For the shape loss, we adopt Sketched Isotropic Gaussian Regularization (SIGReg) Balestriero & LeCun (2025) to match the characteristic function of the clean token $\mathbf{h}$ to that of an isotropic Gaussian distribution. This regularization prevents trivial scale-inflation solution and shapes the token distribution to be favorable for autoencoding and generative modeling. Once training is complete, the tokenizer is frozen across all remaining phases, serving as a fixed nonlinear transform for the remainder of the proposed coding pipeline.

3.2 Bottlenecked Autoencoder

The autoencoder, consisting of an encoder, quantizer, density model, and decoder, performs lossy compression in the token space defined by the subband tokenizer ($\mathbf{T}_{\text{tok}}, \mathbf{T}_{\text{detok}}$) via joint reconstruction and rate regularization objectives. All modules are constructed by stacking layers containing temporal causal convolutions, inter-band self-attention, and token-wise FFNs (Figure A.1). Detailed architecture of the layer is described in Appendix A.1. As with the tokenizer, we refrain from applying spatial or dimensional subsampling to avoid quality degradations induced by dimensionality reduction.

The encoder projects clean tokens $\mathbf{h} \in \mathbb{R}^{S \times F \times D}$ into D_h-dimensional hidden states via an input projection matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{D_h \times D}$, then processes them through N_{enc} layers, yielding $\mathbf{d}_{\text{enc}} \in \mathbb{R}^{S \times F \times D_h}$. The quantizer maintains matching output dimensions. Using identical layer units, the decoder processes these quantized representations into intermediate features, which are then passed through a $\tanh$ activation and projected to reconstruct the tokens $\hat{\mathbf{h}}$.

Rather than using the spatial compression typical of Finite Scalar Quantization (FSQ) Mentzer et al. (2024), the quantizer employs an overcomplete binary-state bottleneck with rate regularization. An information bottleneck is instead imposed by jointly training a density model. As shown in Figure 1b, this density model is used to compress and regularize the encoder output.

First, the encoder output $\mathbf{d}_{\text{enc}} \in \mathbb{R}^{S\times F\times D_h}$ is projected into a continuous bottleneck feature $\mathbf{b}_{raw} \in \mathbb{R}^{S\times F\times D_b}$ using band-specific input projections $\mathbf{W}_{\text{key}} \in \mathbb{R}^{S\times D_b\times D_h}$. These continuous features are subsequently bounded by a $\tanh$ non-linearity and quantized into binary bins to yield the discrete bottleneck $\mathbf{b}_{\text{disc}} \in \{0,1\}^{S\times F\times D_b}$. Finally, $\mathbf{b}_{\text{disc}}$ is projected back to form the quantizer output $\mathbf{d}_{\text{quant}} \in \mathbb{R}^{S\times F\times D_h}$ using band-specific output projections $\mathbf{W}_{\text{value}} \in \mathbb{R}^{S\times D_h\times D_b}$. The full quantizer is formulated as:

$$\begin{aligned}\mathbf{b}_{\text{disc}} &= \lfloor 0.5\cdot\tanh\left(\text{einsum}(sfh, sdh\to sfd; \text{DyT}(\mathbf{d}_{\text{enc}}), \mathbf{W}_{\text{key}})\right) + 0.5\rceil,\\ \mathbf{d}_{\text{quant}} &= \text{einsum}(sfd, shd\to sfh; \mathbf{b}_{\text{disc}}, \mathbf{W}_{\text{value}}),\end{aligned} \tag{7}$$

where DyT denotes Dynamic Tanh Zhu et al. (2025). Reusing the autoencoder's causal layer architecture, the density model estimates the Bernoulli distribution of the current frame's discrete bottleneck features given those from previous frames.

In contrast to the unified rate loss commonly used in existing neural audio codecs Ballé et al. (2017; 2018), our formulation separates rate optimization into a density estimation loss $\mathcal{L}_{\text{de}}$ and a target rate matching loss $\mathcal{L}_{\text{trm}}$, defined as follows.

$$\mathcal{L}_{\text{de}} = \frac{1}{BF}\sum_{i,s,f,d} I_{i,s,f,d}, \tag{8}$$

$$\mathcal{L}_{\text{trm}} = \frac{1}{B}\sum_{i=1}^{B}\left|\frac{1}{F}\sum_{s,f,d} I_{i,s,f,d} - r_{tgt}\right|. \tag{9}$$

$I_{i,s,f,d}$ denote the per-bit self-information of the discrete bottleneck under the density model as

$$I_{i,s,f,d} = -\mathbf{b}_{\text{disc},(i,s,f,d)}\log_2 \mathbf{p}_{\text{DM},(i,s,f,d)} - (1-\mathbf{b}_{\text{disc},(i,s,f,d)})\log_2(1-\mathbf{p}_{\text{DM},(i,s,f,d)}), \tag{10}$$

where $\mathbf{p}_{DM}$ represents the probability estimated by the density model. When computing $\mathcal{L}_{\text{de}}$, we detach $\mathbf{b}_{\text{disc}}$ in I to update only the density model; conversely, when computing $\mathcal{L}_{\text{trm}}$, we detach $\mathbf{p}_{\text{DM}}$ so that gradients propagate exclusively to the autoencoder.

We apply two reconstruction losses depending on the module where the distortion is computed. The first is token reconstruction loss, $\mathcal{L}_{\text{tok}}(\mathbf{h}) = \|\mathbf{h} - \text{AE}(\mathbf{h})\|_2^2$, where $\text{AE}(\cdot)$ denotes the sequential execution through the encoder, quantizer, and decoder. The second is the time-frequency audio reconstruction loss used for the subband tokenizer, $\mathcal{L}_{\text{TF}}(\mathbf{x}, \mathbf{T}_{\text{detok}}(\text{AE}(\mathbf{h})))$. The overall training objective for the bottlenecked autoencoder is therefore:

$$\mathcal{L}_{\text{BAE}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}(\mathbf{x},\mathbf{h}) + \mathcal{L}_{\text{de}} + \lambda_{\text{trm}}\mathcal{L}_{\text{trm}}, \tag{11}$$

where $\mathcal{L}_{\text{recon}}$ corresponds to either of these candidate loss formulations.

3.3 Refinement Head

Training the bottlenecked autoencoder solely with a reconstruction loss is insufficient to produce high-fidelity audio, as regression objectives inherently suffer from mean-regression behavior Sajjadi et al. (2017); Kupyn et al. (2018). This manifests as spectral holes and attenuated components, particularly in high-frequency regions where signal energy is lower than in low-frequency regions. To resolve this, a refinement head based on the energy-based model (EBM) framework is applied to the decoder output $\hat{\mathbf{h}}$, recovering rich and realistic spectral details.

Energy Function and Training Criteria Given a clean token $\mathbf{h}$, we define an energy function $E(\mathbf{z},\mathbf{h}) \in \mathbb{R}$, which yields a scalar measuring the quality of a candidate variable $\mathbf{z}$ relative to the target $\mathbf{h}$. Following Equilibrium Matching (EqM) Wang & Du (2026), the refinement head learns the gradient of this energy function with respect to $\mathbf{z}$ via gradient field matching. The variable $\mathbf{z}$ is initialized by sampling from a conditional Gaussian centered at the decoder output, $\mathcal{N}(\hat{\mathbf{h}}, \sigma_{\text{init}}\mathbf{I})$.

During training, the state variable is constructed as a linear interpolation between the noisy initial state and the clean target token:

$$\mathbf{z}_\gamma = \gamma(\hat{\mathbf{h}} + \sigma_{\text{init}}\epsilon) + (1-\gamma)\mathbf{h}, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad \gamma \in [0,1], \tag{12}$$

such that $\mathbf{z}_1 = \hat{\mathbf{h}} + \sigma_{\text{init}}\epsilon$ and $\mathbf{z}_0 = \mathbf{h}$. This formulation enables simulation-free training of the refinement head, analogous to flow-matching Lipman et al. (2023).

We define two distinct energy functions corresponding to the two reconstruction loss criteria required by the proposed bottlenecked autoencoder framework. When using the token reconstruction loss as the energy function, the target vector field is given analytically by

$$\mathbf{v}_{\text{tok}}(\gamma) = \nabla_{\mathbf{z}_\gamma} 0.5\|\mathbf{z}_\gamma - \mathbf{h}\|_2^2 = \mathbf{z}_\gamma - \mathbf{h} = \gamma(\hat{\mathbf{h}} + \sigma_{\text{init}}\epsilon - \mathbf{h}). \tag{13}$$

Conversely, when adopting $\mathcal{L}_{\text{TF}}$ as the energy, the target is computed via automatic differentiation:

$$\tilde{\mathbf{v}}_{\text{TF}}(\gamma) = \nabla_{\mathbf{z}_\gamma} \mathcal{L}_{\text{TF}}(\mathbf{T}_{\text{detok}}(\text{AE}(\mathbf{h})), \mathbf{T}_{\text{detok}}(\text{AE}(\mathbf{z}_\gamma))), \tag{14}$$

which requires back-propagating through the fronzen detokenizer module. To stabilize the gradient, we take only the direction of $\tilde{\mathbf{v}}_{\text{TF}}(\gamma)$ while using the length of $\mathbf{v}_{\text{tok}}(\gamma)$ for each token, by instead targeting :

$$\mathbf{v}_{\text{TF}}(\gamma) = \|\mathbf{v}_{\text{tok}}(\gamma)\| \cdot \frac{\tilde{\mathbf{v}}_{\text{TF}}(\gamma)}{\|\tilde{\mathbf{v}}_{\text{TF}}(\gamma)\|}. \tag{15}$$

Training objective of the energy head is formulated as :

$$\mathcal{L}_{\text{RH}} = \sum_{j \in \{\text{tok}, \text{TF}\}} \lambda_j \left\| \text{RH}\Big(\mathbf{z}_\gamma;\, \gamma, \hat{\mathbf{h}}, \mathbf{d}_{\text{quant}}^{(\text{RH})}\Big) - \mathbf{v}_j(\gamma) \right\|_2^2, \tag{16}$$

where $\text{RH}(\cdot)$ denotes the refinement head. For regularization, we additionally incorporate the reconstruction, density estimation, and target rate matching loss objectives previously used during autoencoder pre-training.

Architecture The refinement head is composed of autoregressive context extractor and frame-wise prediction head, which share similar layer unit as the autoencoder. Appendix A.2 details the internal architecture of the refinement head, which comprises a context extractor that performs autoregressive modeling over refined frames and a prediction head that iteratively refines input variables frame-by-frame. To provide the refinement head with auxiliary information from the discrete bitstream and enable joint training, we instantiate a dedicated output projection $\mathbf{W}_{\text{quant}}^{(\text{RH})}$ by cloning the pre-trained weights of $\mathbf{W}_{\text{value}}$. The refinement head then processes the detached decoder output $\hat{\mathbf{h}}$, given the side-information projection $\mathbf{d}_{\text{quant}}^{(\text{RH})} = \text{einsum}(sfd, shd \to sfh; \mathbf{b}_{\text{disc}}, \mathbf{W}_{\text{quant}}^{(\text{RH})})$, and the schedule γ as conditional inputs.

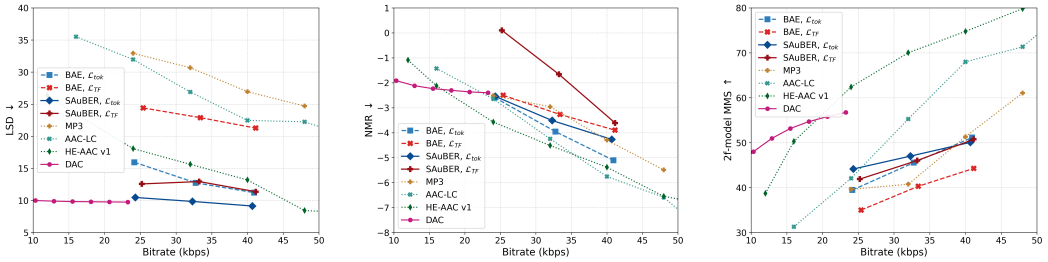
Inference During inference, the state variable $\mathbf{z}_\gamma$ is initialized from the decoder output $\hat{\mathbf{h}}$. The refinement head perform iterative gradient descent based on its estimation with respect to the current state. To minimize degradation due to the stochasticity, we decide σ_{init} of each token relatively based on the decoder $\hat{\mathbf{h}}$, introducing less stochasticity on salient tokens, where the stochasticity is controlled by a single parameter τ. Details are explained in Appendix A.2.

4 Experiments

4.1 Experimental Setup

For training, we use the clean speech subset of the DNS Challange 4 dataset Dubey et al. (2022), MUSDB18-HQ Rafii et al. (2019), and a subset of the MTG-jamendo Bogdanov et al. (2019) for music signals, and FSD50K Fonseca et al. (2022) for general audio signals. All audio clips are downmixed to monaural and resampled to 44.1 kHz where necessary. For evaluation, we use 43 test samples ISO/IEC JTC 1/SC 29/WG 11 N9639 (2008) from the USAC ISO (2020) standardization test set, consisting of a diverse and challenging range of audio content. Three objective evaluation metrics are used: (i) 2f-model Mean MUSHRA score (MMS) Kastner & Herre (2019), (ii) Noise-to-Mask Ratio (NMR) from intermediate PEAQ Thiede et al. (2000) metrics, and (iii) Log-Spectral Distance (LSD). Further architectural details, training configuration, and hyperparameter setups are provided in Appendix A.3.

4.2 Main Results



(a) LSD (b) NMR (c) 2f-model MMS

Figure 2: Objective evaluation results for comparing BAE and SAuBER against DAC and classical audio codecs. BAE, even constructed without psychoacoustic models or perceptual loss, tracks the performance of classical codecs, while the performance of DAC plateaus at higher bitrates. Attaching refinement head improve both LSD and 2f-model score, at the expense of degraded NMR.

We compare our proposed framework against classical signal processing-based audio codecs, including LAME MP3 ISO (1993), Fraunhofer FDK AAC-LC ISO (1997), and HE-AAC v1 ISO (2003), as well as state-of-the-art GAN-based neural audio codec. Figure 2 presents objective evaluation results across three metrics: LSD, NMR, and 2f-model MMS. Throughout this section, SAuBER denotes our full framework after joint training of the autoencoder and refinement head, while BAE denotes the baseline bottlenecked autoencoder trained without the refinement head. Both SAuBER and BAE are trained independently at target bitrates of 24, 32, and 40 kbps. For SAuBER, the refinement head is trained exclusively on $\mathbf{v}_{\text{tok}}$. By default, inference refinement employs $N_{step} = 30$, learning rate $lr = 0.08$, token-relative initialization temperature $\tau = 1.5$, and Classifier-Free Guidance Ho & Salimans (2021) scale $w_{CFG} = 2.0$.

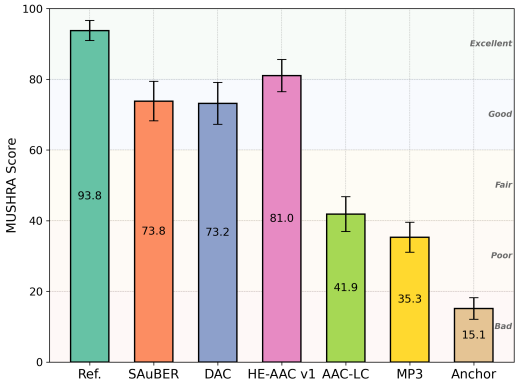


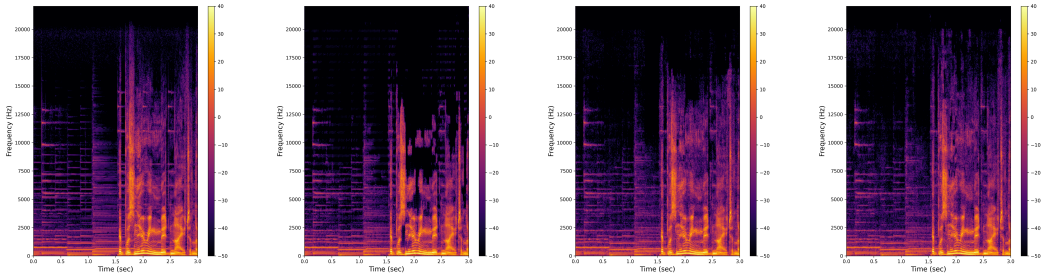
Figure 3: MUSHRA test results at 24 kbps.

First, the BAE alone exhibits a clear tendency: $\mathcal{L}_{\text{tok}}$ outperforms $\mathcal{L}_{\text{TF}}$ across all metrics and bitrates. As shown in Appendix A.5 (Figure A.7e–A.7g), BAE trained with $\mathcal{L}_{\text{TF}}$ consistently recovers low-frequency and high-magnitude components bu t largely discards high-frequency and low-magnitude regions compared to training with $\mathcal{L}_{\text{tok}}$. Conversely, while $\mathcal{L}_{\text{tok}}$ preserves high-frequency details better, it introduces repetitive high-frequency artifacts. Notably, without psychoacoustic models or perceptual losses, relying solely on reconstruction and rate regularization, BAE scales with bitrate similarly to classical codecs such as MP3 and AAC-LC rather than DAC, improving steadily across the operating range without plateauing.

Comparing BAE and SAuBER, the refinement head consistently improves both objectives, lowering LSD and raising 2f-model MMS, with the largest gains at 24 kbps. This improvement is most pronounced for models using $\mathcal{L}_{\text{TF}}$ reconstruction loss, whereas models trained with $\mathcal{L}_{\text{tok}}$ achieves smaller yet consistent performance gains. The refinement head slightly degrades NMR due to the added stochasticity and generative modeling, though this trade-off is confined to lower bitrates. Overall, applying refinement head is beneficial at lower bitrates, where the BAE alone with mean-regression behavior cannot recover high-fidelity audio signals (Figure 4b and 4c). Meanwhile, DAC plateaus across all three metrics and shows no further gain at 24 kbps, implying that spatial compression paired with GAN-based generation fails to scale effectively as bitrate increases.

We conduct a Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) listening test at a target bitrate of 24 kbps, comparing five codecs: SAuBER (trained with $\mathcal{L}_{\text{tok}}$ and $\lambda_{\text{tok}} = 0.5$),

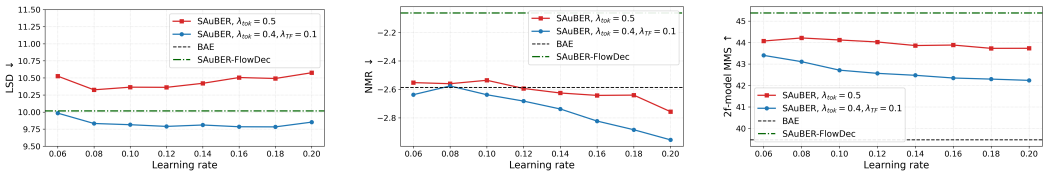
DAC, HE-AAC v1, AAC-LC, and MP3. For a fair comparison, DAC is trained on the same dataset as SAuBER and configured at a comparable parameter scale. The test evaluates nine audio clips from the USAC evaluation set, covering speech, music, and mixture signals (Figure 3). At 24 kbps, SAuBER achieves a MUSHRA score comparable to DAC and slightly below HE-AAC v1, while surpassing AAC-LC and MP3 by a large margin. Performance varies by signal type: SAuBER outperforms DAC on music and mixed audio but underperforms on speech. A detailed analysis of the MUSHRA results is provided in Appendix A.4.1.



(a) Input signal (b) BAE only (c) SAuBER, $\lambda_{\text{tok}} = 0.5$ (d) SAuBER, $\lambda_{\text{tok}} = 0.4$, $\lambda_{\text{TF}} = 0.1$

Figure 4: Spectrograms of (a) an input signal, (b) a reconstruction without the refinement head, and SAuBER, whose refinement head trained with (c) $\lambda_{\text{tok}} = 0.5$ and (d) $\lambda_{\text{tok}} = 0.4$, $\lambda_{\text{TF}} = 0.1$. All models use $\mathcal{L}_{\text{tok}}$ as the reconstruction loss.

4.3 Refinement Process Analysis



(a) LSD (b) NMR (c) 2f-model MMS

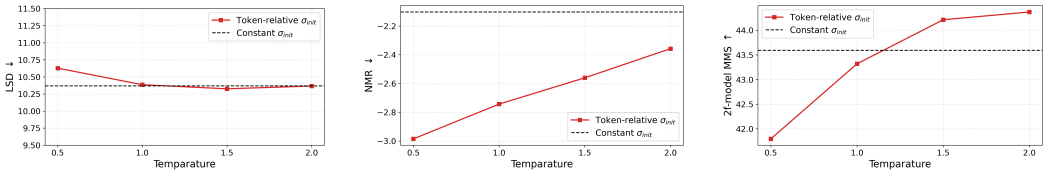
Figure 5: Effect of the learning rate during the inference-time refinement process. Increase of the learning rate improves NMR, slightly degrades 2f-model score, while LSD plateaus with respect to the learning rates.

We analyze the inference-time refinement process along two key factors: the refinement learning rate and the token-relative initialization of the state variable. Sensitive analyses on classifier-free guidance and step count are provided in Appendix A.4.2. We evaluate two SAuBER variants: one trained on the gradient of $\mathcal{L}_{\text{tok}}$ ($\lambda_{\text{tok}} = 0.5$), and another targeting a weighted sum of the gradients ($\lambda_{\text{tok}} = 0.4, \lambda_{\text{TF}} = 0.1$). As a baseline, we train a FlowDec variant of SAuBER, where the autoencoder is frozen during refinement head training and conditional inputs ($\hat{\mathbf{h}}$) and $\mathbf{d}_{\text{quant}}^{(RH)}$ are omitted. Its target vector field is formulated as $\mathbf{h} - \hat{\mathbf{h}} - \sigma_{\text{init}}\epsilon$. Unless specified otherwise, we set $N_{step} = 30$, $lr = 0.08$, $\tau = 1.5$, and $w_{CFG} = 1.0$.

Figure 5 shows objective evaluation results across various learning rates. For both SAuBER variants, LSD plateaus while NMR improves and 2f-model score degrades with increasing learning rate. This trade-off suggests that higher learning rates recover salient low-frequency components more accurately at the expense of slight perceptual degradation.

Figure 6 demonstrates the effectiveness of token-relative initialization, based on weights $\lambda_{\text{tok}} = 0.5$. Lowering the temperature τ reduces the stochasticity of the initial state $\mathbf{z}_1$, degrading LSD and 2f-model MMS while improving NMR. With an appropriate τ, token-relative initialization outperforms the baseline using constant σ_{init} across all metrics.

Based on LSD and NMR results, training the refinement head to target the gradient of explicit energy function ($\mathbf{v}_{\text{TF}}(\gamma)$) improves both high-frequency recovery and the reconstruction accuracy of salient

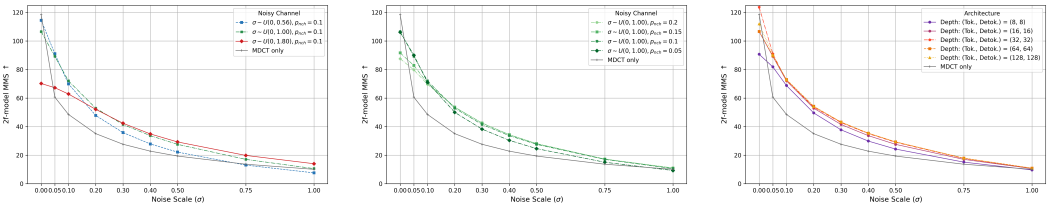


(a) LSD (b) NMR (c) 2f-model MMS

Figure 6: Effect of the token-relative initialization of the state variable $\mathbf{z}_\gamma$. By selecting appropriate temperature τ of the relative initialization, it minimized degradation due to added stochasticity at salient components while recovering plausible high frequencies.

spectral components. Comparing Figure 4c and 4d, the latter recovers richer high-frequency details, which is consistent with our objective evaluations. Furthermore, comparison with the FlowDec variant confirms that jointly training the autoencoder and refinement head enables plausible high-frequency generation without degrading low frequency components.

4.4 Analysis on the Subband Tokenizer



(a) Maximum noise scale $\sigma_{\max}$ (b) Perturbation probability p_{nc} (c) Tokenizer depth

Figure 7: Reconstruction quality of tokenizer under clean and noisy channels. We compare different setups for (a) the probability p_{nc} for perturbing tokens, and (b) the maximum scale $\sigma_{\max}$ of the added Gaussian noise during training, and (c) the depth of FFNs in the tokenizer.

We analyze how tokenizer hyperparameters affect its two core properties: near-transparency and degradation-robustness. For each evaluation audio clip, we estimate the standard deviation of each subband and add Gaussian noise proportional to the estimated scale of each band. As shown in Figure 7a, increasing the maximum noisy-channel scale $\sigma_{\max}$ enhances token-space noise-robustness, though an excessively high $\sigma_{\max}$ degrades the reconstruction quality under clean channel ($\sigma = 0$). After fixing $\sigma_{\max} = 1.0$, we evaluate the impact of the perturbation probability p_{nc}. As the probability increases until $p_{\text{nc}} = 0.1$, the noise-robustness improves while the reconstruction quality under clean channel barely affected, but further increasing p_{nc} do not improve the robustness much but degenerates the near-transparency. Finally, with $\sigma_{\max} = 1.0$ and $p_{\text{nc}} = 0.1$, stacking 32 FFN layers per tokenizer and detokenizer serve as a practical upper bound for performance scaling.

5 Conclusion

In this paper, we presented SAuBER, a novel neural audio codec designed for high-fidelity audio compression. By employing a learnable MDCT-based tokenizer and an information-bottlenecked autoencoder without spatial downsampling, SAuBER avoids loss of fine time-frequency details and maintains an overcomplete feature space throughout the processing. To counteract mean-regression artifacts and loss of high-frequency components at low bitrates, we introduced an energy-based refinement head trained via estimating gradient of explicit energy function. Comprehensive evaluations demonstrate that SAuBER scales continuously across bitrates (24-40 kbps), achieving competitive performance with classical and generative baselines in objective audio quality and MUSHRA listening tests on music and mixed audio signals. Future work includes adopting psychoacoustic modeling or perceptual objectives in the training loss, and distilling the multi-step refinement into a few-step model for lower inference cost.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology or experiments, and implementing methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, or interpreting results, and generating synthetic datasets are not applicable to this work. Additionally, we used generative AI tools for suggesting experimental parameters, editing software code, and drafting parts of a research paper. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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 Transactions on Audio, Speech, and Language Processing*, 32:3256–3269, 2024.

Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk4_qw5xe.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

Johannes Ballé, Philip A. Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE J. Sel. Top. Signal Process.*, 15(2):339–353, 2021.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The MTG-Jamendo dataset for automatic music tagging. In *Proc. ICML Workshop Mach. Learn. Music Discovery*, 2019.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Trans. Mach. Learn. Res.*, 2023. ISSN 2835-8856.

Harishchandra Dubey, Vishak Gopal, Ross Cutler, Ashkan Aazami, Sergiy Matusevych, Sebastian Braun, Sefik Emre Eskimez, Manthan Thakker, Takuya Yoshioka, Hannes Gamper, and Robert Aichner. ICASSP 2022 deep noise suppression challenge. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 9271–9275, 2022.

Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hyx4knR9Ym.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: An open dataset of human-labeled sound events. *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 30: 829–852, 2022.

Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pp. 2672–2680, 2014.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio. ISO/IEC 11172-3, 1993.

Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 7: Advanced Audio Coding (AAC). ISO/IEC 13818-7:1997, 1997.

Information Technology—Coding of Audio-Visual Objects—Part 3: Audio. ISO/IEC 14496-3:2001/Amd 1:2003, 2003.

ISO/IEC JTC 1/SC 29/WG 11 N9639. Workplan on speech and audio material selection, Jan. 2008.

Information Technology—MPEG Audio Technologies—Part 3: Unified Speech and Audio Coding. ISO/IEC Standard 23003-3:2020, 2020.

Xiao-Hang Jiang, Yang Ai, Rui-Chen Zheng, Hui-Peng Du, Ye-Xin Lu, and Zhen-Hua Ling. Mdctcodec: A lightweight mdct-based neural audio codec towards high sampling rate and low bitrate scenarios. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 540–547. IEEE, 2024.

Xiao-Hang Jiang, Yang Ai, Hui-Peng Du, Zhen-Hua Ling, and Ji Wu. Cfmdctcodec: A low-bitrate neural speech codec with noise-prior-aware conditional flow matching for mdct-spectral enhancement. *IEEE Transactions on Audio, Speech and Language Processing*, 2026.

Thorsten Kastner and Jürgen Herre. An efficient model for estimating subjective quality of separated audio source signals. In *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, pp. 95–99, 2019.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 59223–59241, 2023.

Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8183–8192, 2018.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Hyungseob Lim, Jihyun Lee, Byeong Hyeon Kim, Inseon Jang, and Hong-Goo Kang. End-to-end neural audio coding in the mdct domain. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Hyungseob Lim, Jihyun Lee, Byeong Hyeon Kim, Inseon Jang, and Hong-Goo Kang. Perceptual neural audio coding with modified discrete cosine transform. *IEEE Journal of Selected Topics in Signal Processing*, 18(8):1490–1505, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Nicola Pia, Martin Strauss, Markus Multrus, and Bernd Edler. FlowMAC: Conditional flow matching for audio coding at low bit rates. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2025.

Prolific. Prolific online participant recruitment platform, 2020. URL https://www.prolific.com. Accessed: 2026-09-26.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ - an uncompressed version of MUSDB18, Aug. 2019.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15781–15791. IEEE, 2025.

Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE international conference on computer vision*, pp. 4491–4500, 2017.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

Thilo Thiede, William C. Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G. Beerends, and Catherine Colomes. PEAQ—the ITU standard for objective measurement of perceived audio quality. *J. Audio Eng. Soc.*, 48(1/2):3–29, 2000.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 32–42. IEEE, 2021.

Lars Villemoes, Mark Vinton, Per Ekstrand, Lie Lu, Grant Davidson, and Cong Zhou. Mdctnet: a hybrid approach to neural audio coding. *IEEE Journal of Selected Topics in Signal Processing*, 18(8):1506–1516, 2024.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models, 2026. URL https://openreview.net/forum?id=xqlxtkOhDX.

Simon Welker, Matthew Le, Ricky T. Q. Chen, Wei-Ning Hsu, Timo Gerkmann, Alexander Richard, and Yi-Chiao Wu. FlowDec: A flow-based full-band general audio codec with high perceptual quality. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Yi-Chiao Wu, Dejan Marković, Steven Krenn, Israel D. Gebru, and Alexander Richard. ScoreDec: A phase-preserving high-fidelity audio codec with a generalized score-based diffusion post-filter. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 361–365, 2024.

Yi-Chiao Wu, Dejan Marković, Steven Krenn, Israel D. Gebru, and Alexander Richard. ComplexDec: A domain-robust high-fidelity neural audio codec with complex spectrum modeling. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2025.

Haici Yang, Inseon Jang, and Minje Kim. Generative de-quantization for neural speech codec via latent diffusion. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1251–1255, 2024.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 30: 495–507, 2022.

Shuangfei Zhai. Exclusive self attention. *arXiv preprint arXiv:2603.09078*, 2026.

Zhengpu Zhang, Jianyuan Feng, Yongjian Mao, Yehang Zhu, Junjie Shi, Xuzhou Ye, Shilei Liu, Derong Liu, and Chuanzeng Huang. High-fidelity diffusion-based audio codec. In *2024 18th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 344–348. IEEE, 2024.

Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. Transformers without normalization. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14901–14911. IEEE, 2025.

A Appendix

A.1 Bottlenecked Autoencoder

Layer Architecture Each layer of the bottlenecked autoencoder consist for three sub-layers: temporal causal convolution, inter-band self-attention, and token-wise FFN, as illustrated in Figure A.1. For a batch of hidden features of shape $B \times S \times F \times D_h$, the temporal sub-layer reshapes features into $(BS) \times F \times D_h$, treating the frame axis F as the sequence length and BS as the effective mini-batch. The inter-band sub-layer reshapes them into $(BF) \times S \times D_h$, treating the subband axis as the sequence length.

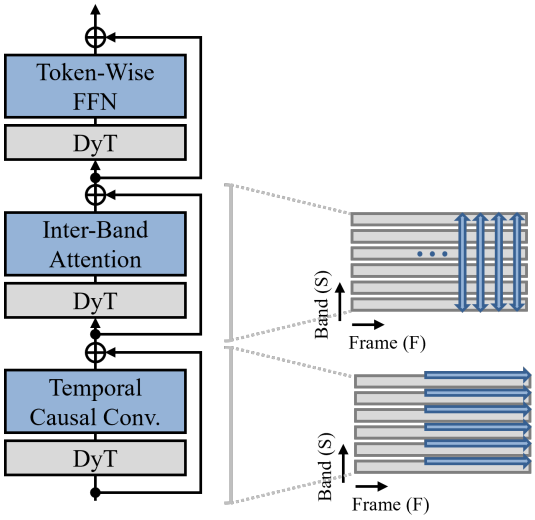


Figure A.1: Architecture of the layer used in the autoencoder, the density model, and context extractor.

Quantizer and Density Model Figure A.2 illustrates the detailed architecture of the quantizer and the density model. The input and output projections of the quantizer, $\mathbf{W}_{\text{key}}$ and $\mathbf{W}_{\text{value}}$, can be interpreted as a codebook or key-value matrices. To enable the quantizer to capture different aspects of the audio signals across frequency ranges, we assign different weights to these matrices across frequency bands, unlike the FFNs in the rest of the model.

The density model perform density estimation of the binary bottleneck features assuming causal dependency. The density model is constructed using the same layer architecture used in autoencoder. Figure A.2 describes the density modeling process. The density model directly takes the discrete bottleneck features $\mathbf{b}_{\text{disc}}$ as input, prefixing these features with a learnable embedding $\mathbf{e}_{\text{BOS}} \in \mathbb{R}^{S \times 1 \times D_b}$, and drops the last frame. It then processes the prefixed sequence and applies a sigmoid activation to output the conditional probability estimates. Probability estimation $\mathbf{p}_{\text{DM}}$ is then used for calculating the self information in Eq. 3.2.

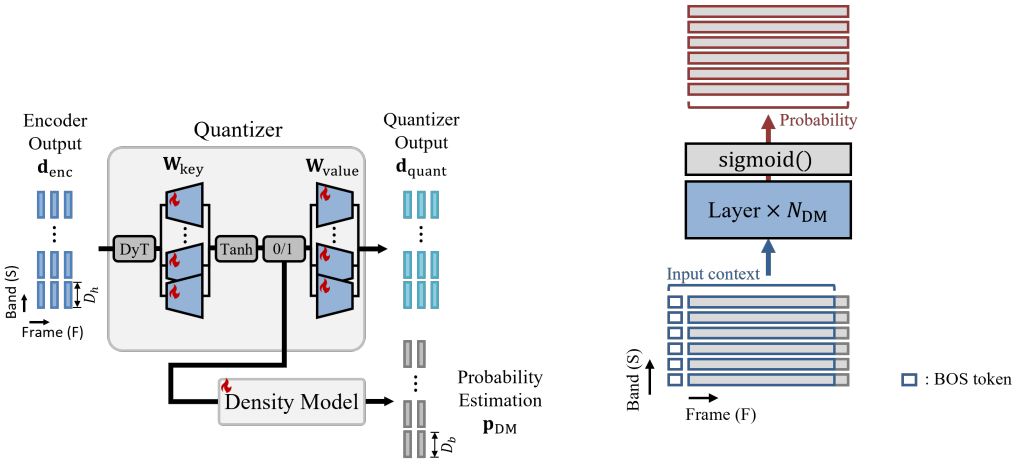


Figure A.2: Architecture of the quantizer and the density model.

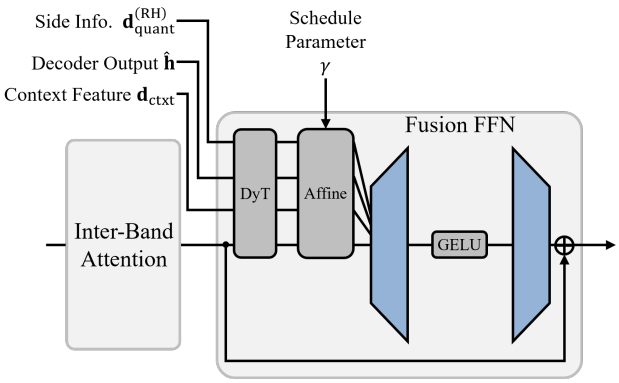


Figure A.3: Layer architecture of the prediction head.

A.2 Refinement Head

Architecture Figure A.3 illustrates detailed architecture of the prediction head. The prediction head consists of an input projection, a stack of fusion layers, and an output projection with a $\tanh$ activation function. To enforce strictly frame-wise operation, each fusion layer contains two sub-layers—an inter-band self-attention module and a token-wise fusion FFN. The fusion FFN, as shown in Figure A.3, incorporates the main residual stream together with three D_h-dimensional conditional signals: the context extractor's output, the clean decoder output $\hat{\mathbf{h}}$, and the side-information projection $\mathbf{d}_{\text{quant}}^{(\text{RH})}$.

Reusing the autoencoder's layer architecture, the context extractor models previously generated frames in an autoregressive manner. During training, it receives a noisy context Ren et al. (2025), in which past frames are populated by the interpolated state variables $\mathbf{z}_\gamma$. We use the same interpolation schedule $\gamma \sim \mathcal{U}(0, 1)$ for constructing both the input of the refinement head and the noisy context. At inference, refinement proceeds frame by frame: the refinement head optimizes the current frame's decoder output by gradient descent, and the resulting refined token is then committed to the context extractor as the newest frame of its autoregressive context.

Gradient Descent Algorithm 1 describes how during the inference the refinement head perform iterative optimization along the learned gradient. The side information transmitted from the quantizer serves as an conditional input, thus dropping that side information proportionally during training enables the utilization of the CFG.

Token-Relative Initialization We quantify the saliency of each token by a root-mean-squared error (RMSE) from a null token. Null token $\mathbf{n} \in \mathbb{R}^{S \times D_h}$ is computed by passing a white noise of -50 dB through the tokenizer and average the clean token $\mathbf{h}$ across the frame axis. Then noise scale of the initial variable is determined as :

$$\hat{\sigma}_{\text{init}}(\hat{\mathbf{h}}) = \sigma_{\text{init}} \cdot \min\left(0, \max\left(1, \frac{\tau}{D_h \|\hat{\mathbf{h}} - \mathbf{n}\|_2}\right)\right), \tag{17}$$

where the formula is applied for each band, independently.

Algorithm 1 Energy-Based Refinement by Gradient Descent

Require: initial variable $\mathbf{z}_1$; anchor $\hat{\mathbf{h}}$; steps N_{step}; learning rate η; CFG scale w_{CFG}; side information $\mathbf{d}_{\text{quant}}^{(\text{RH})}$;
Ensure: refined token $\tilde{\mathbf{h}}$

$\mathbf{z} \leftarrow \mathbf{z}_1$; $\gamma \leftarrow 1$;
for $k = 1, 2, \ldots, K$ **do**
  $\hat{\mathbf{v}}_c \leftarrow \mathrm{RH}(\mathbf{z}; \gamma, \hat{\mathbf{h}}, \mathbf{d}_{\text{quant}}^{(\text{RH})})$ {conditional prediction}
  if $\omega \neq 1$ **then**
    $\hat{\mathbf{v}}_u \leftarrow \mathrm{RH}(\mathbf{x}; \gamma, \hat{\mathbf{h}}, \varnothing)$ {unconditional prediction}
    $\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}}_u + \omega\,(\hat{\mathbf{v}}_c - \hat{\mathbf{v}}_u)$
  else
    $\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}}_c$
  end if
  $\mathbf{z} \leftarrow \mathbf{z} - \eta\,\hat{\mathbf{v}}$ {gradient step}
  $\gamma \leftarrow \gamma - \eta\,\gamma$ {schedule advance}
end for

return $\mathbf{z}$

A.3 Experimental Setup

Table 1 shows the detailed configuration of the SAuBER. Across all neural networks, we remove all the bias terms to avoid tonal artifacts. We adopt the exclusive self-attention (XSA) Zhai (2026) in all self-attention layers. LayerScale Touvron et al. (2021) is applied only to the FFN sub-layers.

A.4 Quantitative Results

A.4.1 MUSHRA Tests

Experimental Setup We selected 9 audio clips from the USAC evaluation set, three clips of clean speech, three clips of music without singing voice, and three mixtures composed of music, speech, and environmental audio. Participants for this study were recruited using Prolific Prolific (2020). We aggregated 20 submission from participants. After the screening process, the results from 9 participants are selected for analyzing results.

The test includes a hidden reference, a 3.5 kHz low-pass-filtered anchor, and five comparison groups: SAuBER, DAC, HE-AAC v1, AAC-LE, and MP3.

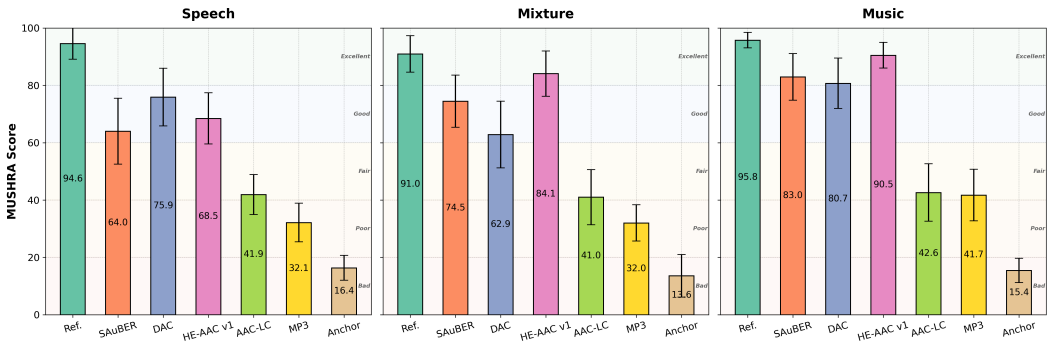


Figure A.4: Aggregated MUSHRA test results for different audio types.

Analysis Figure A.4 reports MUSHRA test results at 24 kbps, broken down into three groups by audio types. The result reveals different tendencies among audio types. On speech, DAC outperforms SAuBER and all classical codecs. On mixture and music, however, the ordering reverses:

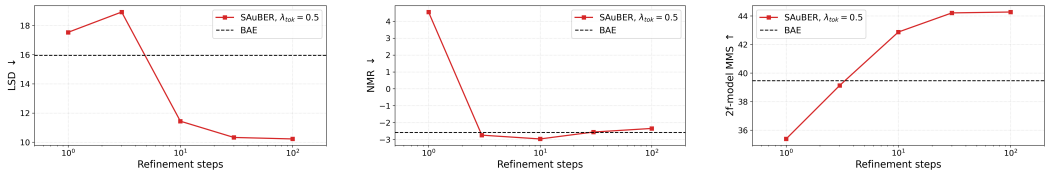
Table 1: Configuration of the proposed framework. The configuration described in this table is unchanged unless specified.

(a) Training configuration	
Optimizer	AdamW (β_1=0.9, β_2=0.95, $\lambda = 10^{-4}$)
Learning rate	6×10^{-4}
LR schedule	Linear warmup (until 5 K iter.)
	Exponential decay (after 5 K iter., γ=0.9999965)
Gradient clipping	100.0
Batch size	16
Training iterations	200 K
(b) Subband tokenizer	
Window length ($2N_{\mathrm{bin}}$)	256
Number of subbands (S)	32
Token dim. (D)	32
Number of layers (tokenizer, detokenizer)	(32, 32)
FFN channel expansion	4
Layerscale initialization	10^{-6}
Model size (parameters)	0.53 M
(c) Bottlenecked autoencoder	
Hidden dim. (D_h)	256
Quantizer dim. (D_b)	256
Number of layers (encoder, decoder, density model)	(8, 8, 4)
Kernel size	7
Group size	32
Attention heads	8
FFN channel expansion	4
Model size (parameters)	23.60 M
(d) Refinement head	
Hidden dim. (D_h)	256
Side-information dropout	0.1
Number of layers (Context extractor, prediction head)	(4, 4)
Noise scale of the initial state (σ_{init})	0.5
Model size (parameters)	10.45 M
(e) Training objective	
Time-frequency reconstruction loss ($\mathcal{L}_{\mathrm{TF}}$)	
STFT window size	[256, 512, 1024, 2048]
Hop length	[64, 128, 256, 512]
Magnitude compression (μ)	100.0
Loss ratio (subband tokenizer training)	
Shape regularization (λ_{shape})	1×10^{-2}
Loss ratio (bottlenecked autoencoder training)	
Reconstruction loss (λ_{recon} for ($\mathcal{L}_{\mathrm{tok}}, \mathcal{L}_{\mathrm{TF}}$))	(1.0, 100.0)
Target rate matching loss (λ_{trm})	20.0
Loss ratio (refinement head training)	
Reconstruction loss (λ_{recon} for ($\mathcal{L}_{\mathrm{tok}}, \mathcal{L}_{\mathrm{TF}}$))	(0.5, 50.0)
Target rate matching loss (λ_{trm})	20.0

SAuBER surpasses DAC by a large margin on mixture and slightly outperforms DAC on music, while HE-AAC v1 remains the highest.

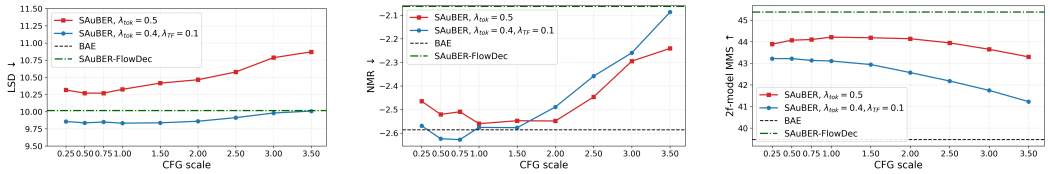
SAuBER becomes advantageous against DAC on mixtures and music, where the audio signals have complex spectral structures and thus reconstruction of high frequencies matters. This is consistent with the behavior of the refinement head, which generates plausible high-frequency components not recovered from the bottlenecked autoencoder. On speech, by contrast, where clean reconstruction of low-frequency components is critical, SAuBER underperforms DAC and HE-AAC v1. Since SAuBER is trained without psychoacoustic models or perceptual losses, its ability to approach classical codecs on mixture and music, and to exceed DAC on both, suggests that the learned token space and energy-based refinement generalize across content types more evenly than GAN-based NACs.

A.4.2 Further Analysis on the Refinement Process



(a) LSD (b) NMR (c) 2f-model MMS

Figure A.5: Performance improvement from the refinement head with respect to the number of steps.



(a) LSD (b) NMR (c) 2f-model MMS

Figure A.6: Effect of the classifier-free guidance during the refinement process.

Experiments are conducted under same setup as Section 4.3.

We first analyze the effect of the number of refinement step N_{step}. Results are shown in Figure A.5. SAuBER outperforms the autoencoder as using about $N_{\text{step}} = 10$ steps. However, the performance improvement plateaus after increasing into 100 steps..

Figure A.6 shows the effect of CFG scale (w_{CFG}) to the performance. SAuBER trained with $\lambda_{\text{tok}} = 0.5$ acheives best LSD using $w_{CFG} \in [0.5, 0.75]$, while the NMR and 2f-model scores become optimal at a range of $[1.0, 2.0]$. Similarly, SAuBER with $\lambda_{\text{tok}} = 0.4, \lambda_{\text{TF}} = 0.1$ consistently shows optimal results in a range $[0.5, 0.75]$.

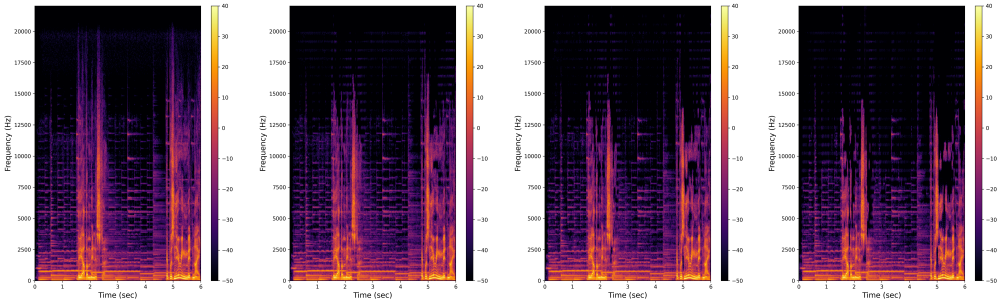
A.4.3 Autoencoding Directly on MDCT Coefficients

To verify the robustness of our subband tokenzier, we train two bottlenecked autoencoder, each with loss $\mathcal{L}_{\text{tok}}$ or $\mathcal{L}_{\text{TF}}$, directly on the patchified MDCT coefficients. Results are shown in Table 2. The MDCT-space autoencoder trained with $\mathcal{L}_{\text{TF}}$ attains slightly better NMR, and the one trained with $\mathcal{L}_{\text{tok}}$ attains better LSD, than their token-space counterparts; however, both lag far behind the token-space BAE trained with $\mathcal{L}_{\text{tok}}$ in 2f-model score. Moreover, attaching the refinement head to the MDCT-space autoencoder fails to improve any of the three metrics, whereas the same head consistently improves the token-space BAE. These results support our claim that a learnable tokenizer is better suited to neural compression and generative modeling than a fixed time–frequency transform.

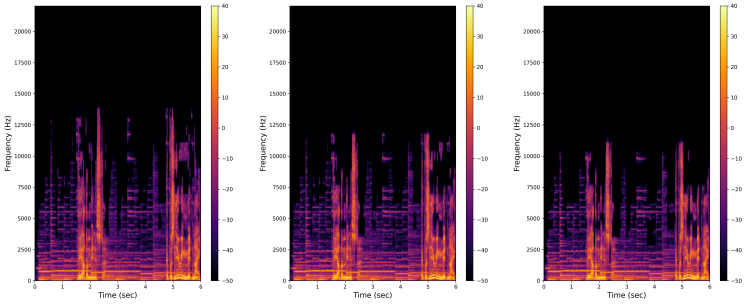
Table 2: Ablation of the learnable tokenizer at 24 kbps. When the autoencoder and the refinement head are applied on the MDCT coefficients directly, only patchifying and euqlizing operations are performed.

Reconstruction Loss	Objective	LSD ↓	NMR ↓	MMS ↑
BAE on subband token space	$\mathcal{L}_{\text{tok}}$	15.94	-2.58	39.46
	$\mathcal{L}_{\text{TF}}$	24.41	-2.50	35.02
BAE on MDCT coefficients	$\mathcal{L}_{\text{tok}}$	15.24	2.82	27.90
	$\mathcal{L}_{\text{TF}}$	19.65	-2.64	19.66
SAuBER on subband token space	$\mathcal{L}_{\text{tok}}$	**9.83**	**-2.57**	**43.11**
SAuBER on MDCT coefficients	$\mathcal{L}_{\text{tok}}$	21.55	24.76	3.70

A.5 Qualitative Results



(a) Input signal (b) $\mathcal{L}_{\text{tok}}$, 40 kbps (c) $\mathcal{L}_{\text{tok}}$, 32 kbps (d) $\mathcal{L}_{\text{tok}}$, 24 kbps



(e) $\mathcal{L}_{\text{TF}}$, 40 kbps (f) $\mathcal{L}_{\text{TF}}$, 32 kbps (g) $\mathcal{L}_{\text{TF}}$, 24 kbps

Figure A.7: Spectrograms of the input signal and reconstructed signals from the bottlenecked autoencoder trained without the refinement head.