S-PRESSO: Ultra Low Bitrate Sound Effect Compression With Diffusion Autoencoders And Offline Quantization

Zineb Lahrichi, Gaëtan Hadjeres, Gaël Richard, Geoffroy Peeters

▶ **To cite this version:**

Zineb Lahrichi, Gaëtan Hadjeres, Gaël Richard, Geoffroy Peeters. S-PRESSO: Ultra Low Bitrate Sound Effect Compression With Diffusion Autoencoders And Offline Quantization. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2026, IEEE Signal Processing Society, May 2026, Barcelone, Spain. ⟨hal-05492477v3⟩

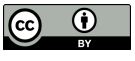
HAL Id: hal-05492477

https://hal.science/hal-05492477v3

Submitted on 13 Feb 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

S-PRESSO: ULTRA LOW BITRATE SOUND EFFECT COMPRESSION WITH DIFFUSION AUTOENCODERS AND OFFLINE QUANTIZATION

Zineb Lahrichi[‡*], *Gaëtan Hadjeres*[‡], *Gaël Richard*[*], *Geoffroy Peeters*[*]

[‡]Sony AI, [*]LTCI, Télécom Paris, Institut Polytechnique de Paris

ABSTRACT

Neural audio compression models have recently achieved extreme compression rates, enabling efficient latent generative modeling. Conversely, latent generative models have been applied to compression, pushing the limits of continuous and discrete approaches. However, existing methods remain constrained to low-resolution audio and degrade substantially at very low bitrates, where audible artifacts are prominent. In this paper, we present **S-PRESSO**, a 48kHz sound effect compression model that produces both continuous and discrete embeddings at *ultra-low bitrates*, down to 0.096 kbps, via offline quantization. Our model relies on a pretrained latent diffusion model to decode compressed audio embeddings learned by a latent encoder. Leveraging the generative priors of the diffusion decoder, we achieve *extremely low frame rates*, down to 1Hz (750x compression rate), producing convincing and realistic reconstructions at the cost of exact fidelity. Despite operating at high compression rates, we demonstrate that S-PRESSO outperforms both continuous and discrete baselines in audio quality, acoustic similarity and reconstruction metrics. Audio samples are available at `https://zineblahrichi.github.io/s-presso/`

Index Terms— Audio Codecs, Diffusion Autoencoders, Low Bitrates

1. INTRODUCTION

In recent years, substantial efforts have been devoted to designing low to ultra-low bitrate codecs, motivated by practical deployment of efficient codecs and the creation of compact representations suitable for latent generative models. However, as reported in the image domain [1], neural compression methods are typically optimized for the rate-distortion trade-off at the expense of perceptual quality and realism, often producing artifacts and less natural images at low bitrates. A similar limitation holds for audio: codecs built on residual vector quantization and adversarial training (RVQ-GANs) [2–4] are deterministic and target exact reconstruction, but at high compression introduce audible degradations such as metallic or robotic timbres. Despite adversarial objectives, their perceptual quality remains fundamentally constrained by the bitrate.

To address these limitations, generative models offer a promising alternative, leveraging their strong generative priors to shift the focus from a bitrate/quality trade-off to a bitrate/acoustic similarity trade-off. Here, acoustic similarirty refers to the perceptual resemblance of two sounds as originating from the same source with comparable characteristics over time. While strict similarity can be critical for certain applications (e.g., lossless music streaming), this is less true in dynamic environments such as video games.

Moreover, the stochasticity of generative models can even be advantageous, providing natural variations that prevents repetitive playback of audio samples. For example, avoiding identical footstep sounds when a character walks in a video game helps to enhance perceptual realism.

Ultra–low bitrate codecs using generative models were initially developed for speech [5] and have since been extended to general audio and music [6,7], achieving bitrates of only a few hundred bits per second. However, to our knowledge, these approaches remain limited in bandwidth ($<$ 32 kHz) and exhibit noticeable quality degradation at very low bitrates.

In this paper, we make further progress towards ultra low bitrate compression of high quality audio, focusing on sound effects. We introduce **S-PRESSO**, a diffusion autoencoder that relies on a pretrained latent diffusion model to decode compressed audio embeddings learned by a latent compressor.

In order to encode both *continuous* and *discrete* embeddings, we adopt a three-step training procedure as in [8]. This includes (i) learning compressed representations via continuous diffusion autoencoder training, (ii) offline neural quantization, and (iii) diffusion decoder finetuning, which enables compact yet expressive representations. The proposed model achieves compression ratios up to 750x on 48kHz audio, producing discrete representations at bitrates as low as 0.096kbps while preserving perceptual quality. Finally, leveraging diffusion model priors, our method outperforms strong continuous and discrete baselines in sound quality and acoustic similarity, as corroborated by human evaluations, delivering realistic and high-quality reconstructions even at ultra-low frame rates (down to 1Hz).

2. RELATED WORKS: LOW BITRATE CODECS

Neural audio compression has recently advanced beyond traditional codecs in rate/distortion trade-offs. RVQ-GAN models [2–4] achieve high-fidelity reconstructions at moderate bitrates, but reconstruction quality typically collapses below $\sim$3kbps. Alternative strategies lower the bitrate by reducing frame rates rather than improving quantization [9, 10], yet these remain tailored to speech and narrowband signals.

Advances in generative modeling have enabled compression at much lower rates. Early works applied WaveNet decoders [11] to speech, achieving rates of $\sim$ 2kbps [5], but the limited receptive fields of WaveNets restricted long-range temporal modeling. To address this, these approaches were further improved by using transformers with GAN decoders, capturing longer dependencies and pushing speech compression down to 600 bps [5].

Extensions to general audio and music leverage pretrained semantic or acoustic latent spaces decoded by diffusion models [6, 7], achieving extreme compression down to a few hundred bps. However, these methods remain constrained to narrow bandwidths or domain-specific data, highlighting the need for approaches that generalize across diverse, high-resolution audio.

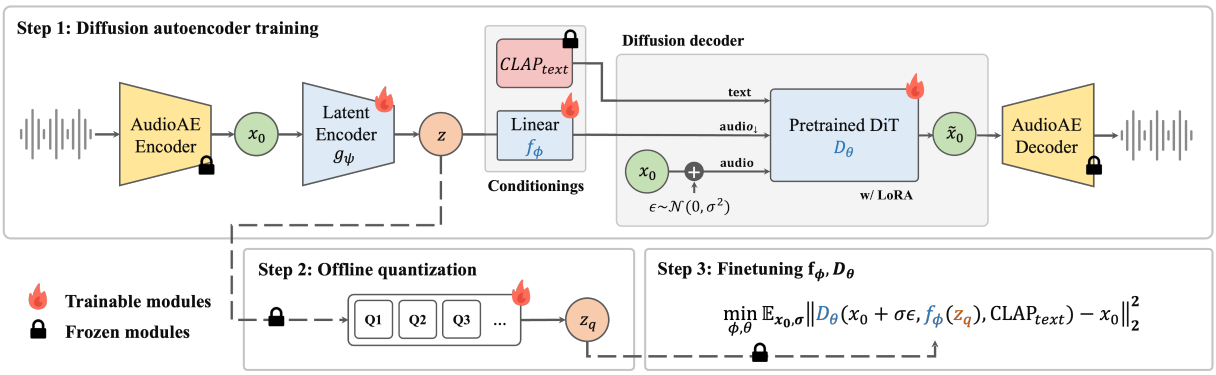


Fig. 1: Overview of our method. **Step 1**: An audio clip is encoded into latent vectors x_0 by a low-compression audio autoencoder. It is then compressed into latents z, which are upsampled by f_ϕ to condition the decoder D_θ, a Diffusion Transformer (DiT) pretrained to reconstruct x_0 from noised inputs. D_θ is finetuned using LoRA adapters, jointly trained with the latent encoder g_ψ and f_ϕ. **Step 2**: The features z are then quantized offline into z_q. **Step 3**: the diffusion decoder D_θ is finetuned on z_q, to compensate for quantization-induced degradation.

3. METHOD

3.1. Overview

An overview of our approach is given in Figure 1. As in [8], we design a three-step training process comprising (i) diffusion autoencoder training, (ii) offline quantization, and (iii) diffusion decoder finetuning. Our approach operates entirely in the latent space x_0 of a pretrained Audio Autoencoder (AudioAE). First, we train a continuous diffusion autoencoder that relies on a diffusion decoder D_θ and a latent encoder g_ψ. The latent encoder maps latent vectors x_0 into compressed representations z which are re-projected by a linear layer f_ϕ and used to condition D_θ. The decoder is a pretrained Diffusion Transformer (DiT), finetuned using LoRA [12] adapters and trained together with g_ψ and f_ϕ, preserving generative priors while enforcing strong audio conditioning. Next, a neural quantizer is trained on the frozen compressed representation z to obtain z_q, minimizing the distortion error. Finally, the diffusion decoder is finetuned by using the audio conditionings $f_\phi(z_q)$ instead of $f_\phi(z)$, compensating for the information loss induced by quantization.

3.2. Pretraining

AudioAE: The latent vectors x_0 are derived from a pretrained low-compression audio autoencoder, yielding high-fidelity reconstructions in a more compact and informative subspace. Following prior work [8], the AudioAE decoder is built upon the design of [13], a GAN-based vocoder trained to predict STFT complex coefficients, and the encoder mirrors this structure. This architecture preserves temporal resolution, reducing upsampling artifacts [14] and enabling efficient training with convolutions over uniformly sized sequences.

DiT: D_θ is a pretrained text-to-audio DiT trained to denoise noisy latent vectors $x_0 + \sigma\epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$, conditioned with a text encoder. It consists of sequential transformer blocks: the first six are multi-modal blocks from [15] operating on the *audio* and *text* modalities followed by six standard transformer blocks operating only on the *audio* modality.

3.3. Diffusion Autoencoder training

Latent encoder: The latent encoder further compresses the latent audio representations x_0 into z. The architecture of the latent encoder g_ψ is provided in Figure 2(a). The encoder downsamples the latent variables $x_0 \in \mathbb{R}^{C \times T}$ along frequency and time by factors c and t, respectively. This is achieved through sequential transformer blocks, followed by a linear layer to reduce the dimensionality and an average pooling layer to downsample in time with kernel and stride t. The transformer blocks employ RoPE positional embeddings [16], which will also be downsampled in the DiT by selecting the central position of each temporal window.

Diffusion decoder: Weights from D_θ are finetuned using LoRA adapters [12] while audio conditioning is injected from the latent encoder via an additional linear layer f_ϕ. The compressed audio conditioning is treated as a third modality termed *audio*↓ similar to the image conditioning from [17] and we initialize dedicated layers for its Q, K, V as depicted in Figure 2(b). The RoPE for the audio conditioning *audio*↓ are a decimated version of the RoPE used for *audio*, so that we maintain the temporal alignment between the two modalities.

3.4. Offline quantization

As in [8], the audio embeddings derived from our latent encoder $z = g_\psi(x_0)$ are quantized offline using the Qinco2 a neural quantizer [18]. Qinco2 extends Residual Vector Quantization (RVQ),

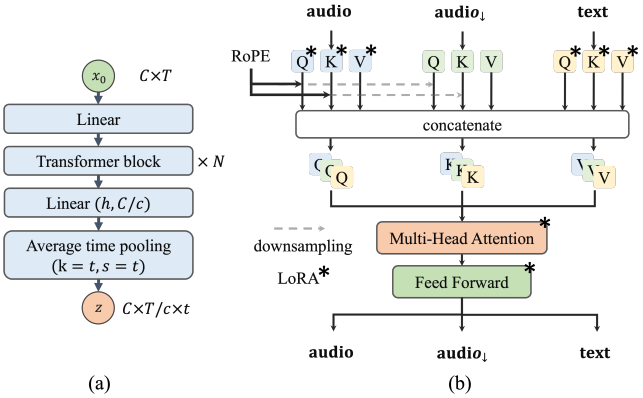


Fig. 2: (a) Overview of the latent encoder architecture (b) Conditioning mechanism within the diffusion decoder.

where continuous vectors are quantized through hierarchical codebooks. Instead of fixed codebooks, it employs neural networks to generate adaptive centroids conditioned on prior reconstructions, enabling finer modeling of residual distributions and capturing inter-codebook dependencies. The bitrate is determined by the number of codebooks M, codebook size K, and frame rate f: $M \times \log_2 K \times f$.

3.5. Finetuning

After training the codebooks, the latent embeddings z are quantized into z_q, which are then re-injected in replacement of z into the diffusion autoencoder pipeline. To mitigate the degradation induced by quantization, the LoRA weights of the diffusion decoder D_θ and the projection layer f_ϕ of the transformer are finetuned.

4. EXPERIMENTS

4.1. Datasets

All models are trained on a combination of four internal sound effect datasets, reaching $\sim$ 5000 hours, sampled at 48kHz and clipped to 5 seconds. Sound effects cover a wide range of audio types, including Foley sounds, environmental sounds, individual events, as well as music samples and background speech. Evaluation uses three datasets: Freesound Effects, BBC Sound Effects, and an internal studio-quality dataset, in order to assess performance on both public benchmarks and professional audio. We randomly sample 500 clips per dataset, each clipped or zero-padded to 5 seconds.

4.2. Training details

4.2.1. Pretraining

AudioAE: Input STFTs are computed using $n_{\text{fft}} = 960$ and a hop size of 480, yielding roughly 100 frames per second. The latent dimension is $C = 128$. Training follows [4], reusing its discriminators, loss functions and optimization parameters.

DiT: We adopt the EDM2 [19] parametrization and training strategy with text conditioning, using AdamW with a $1\text{e}{-4}$ learning rate. Text conditioning is provided by a CLAP encoder [20] trained on our datasets.

4.2.2. Three-step training

Latent encoder: We set the encoder depth by framerate, with lower rates requiring more complexity: 6 blocks at 25Hz, 10 at 11Hz, and 12 at 5Hz and 1Hz. We set the frequency compression factor to $c = 2$, yielding 64-dimensional representations, and choose t according to the target frame rate.

Diffusion Decoder: The DiT is finetuned with LoRA weights under the same EDM2 strategy, sampling σ with a log normal distribution. To prioritize audio over text conditioning, we apply strong dropout (0.8) on the text embeddings, encouraging the model to rely mainly on audio conditioning. Our models are trained using the AdamW optimizer with a learning rate of 1e^{-4}. Each of them is trained across four A100 GPUs, with a batch size of 32.

Offline quantization: We follow the default parameters from Qinco2 [18], adjusting only the codebook size K to match the task complexity. For

high temporal compression rates ($t = 100, 20$), larger codebooks (12 bits vs. 10 bits) are used to reduce MSE in early quantization layers, as fewer frames encode more information and a larger vocabulary improves reconstruction. Each quantizer is trained with $M = 20$ codebooks, and a batch size of 8000 vector frames.

Finetuning: In the final finetuning stage, we replace z with z_q using M codebooks, and continue training f_ϕ and the LoRA weights for 40 additional epochs. The choice of M is determined by evaluating reconstructions with z_q substituted for z without decoder finetuning. As expected, fewer codebooks increase distortion and induce distributional drift. Empirically, we found that a total vocabulary size of about 100 bits is sufficient to retain most salient elements to stay close from the original sound. Accordingly, we finetune our models with $M = 10$ for $K = 10$ and $M = 8$ for $K = 12$. To stabilize training when replacing z with z_q, we retain the original z 10% of the time, reducing the abrupt distributional shift.

4.3. Baselines

In the continuous case (step 1), we benchmark S-PRESSO against continuous baselines trained on high-quality audio: Stable Audio Open [21] (44.1 kHz) and Music2Latent [22] (48 kHz). Music2Latent is trained exclusively on music, and is therefore not directly comparable but remains relevant since our evaluation sets spans background voices and music samples. For fairness, we train S-PRESSO with different downsampling factors t to match the compression rates of the baselines. Table 1 summarizes the latent dimensionality D, frame rate, and overall compression factor R of each method. Additionally, we include the performance of our AudioAE, which serves as the upper bound for our models.

In the discrete case (step 3), the main baseline is SemantiCodec [6], a 16 kHz diffusion-based codec trained on general sounds at comparable bitrates. We evaluate both low bitrate (1–2 kbps) and ultra-low bitrate (100–300 bps) configurations. Table 2 summarizes the bitrate and frame rate of the compared methods. Finally, for consistency across models, audio clips are first resampled to each model's native rate and then converted back to 48 kHz.

4.4. Evaluation metrics

Since our method relies on a generative decoder, we assess audio quality with the Fréchet Audio Distance (FAD) using VGGish and LAION-CLAP embeddings. We additionally report the Kernel Audio Distance (KAD) [23] with LAION-CLAP embeddings, a distribution-free alternative to FAD which shows stronger correlation with human perception. We measure the reconstruction using the Si-SDR and the global acoustic similarity using the cosine distance between the CLAP audio embeddings. For subjective evaluation, we conducted a MUSHRA test [24] against SemantiCodec at low and ultra–low bitrates, using three samples per test set (foleys, music samples, ambiances) and 20 listeners (experts and non-experts) on headphones, who rated quality and similarity.

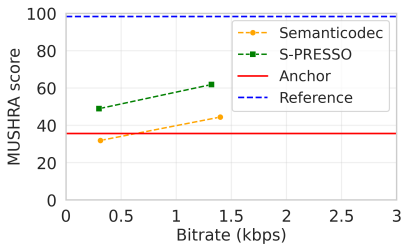


Fig. 3: MUSHRA scores for S-PRESSO, SemantiCodec and a 3.5kHz low-pass anchor, evaluated at $\sim$ 1.35 kbps and $\sim$ 0.3 kbps.

Method	Variant	D	Framerate	R	FAD ↓	FAD_{CLAP} ↓	KAD_{CLAP} ↓	$\text{CLAP}_{\text{audio}}$ ↑	Si-SDR↑
AudioAE	–	128	100 Hz	4	0.008	0.008	0.15	0.90	22.3
StableAudio Open	–	64	21.5 Hz	32	0.78	0.066	1.25	**0.78**	0.48
S-PRESSO	$t = 4$	64	25 Hz	30	**0.48**	**0.038**	**0.57**	0.76	**3.21**
Music2Latent	–	64	11 Hz	64	1.28	0.168	3.29	0.69	-10.5
S-PRESSO	$t = 9$	64	11 Hz	68	**0.59**	**0.050**	**0.77**	**0.76**	**-2.40**
S-PRESSO	$t = 20$	64	5 Hz	150	0.76	0.059	0.92	0.71	-8.80
	$t = 100$	64	1 Hz	750	0.64	0.059	0.89	0.73	-27.7

Table 1: Performance in the continuous case of S-PRESSO vs. continuous audio compression baselines at equivalent compression rates R.

	Method	kbps	Framerate	M	FAD ↓	FAD_{CLAP} ↓	KAD_{CLAP} ↓	$\text{CLAP}_{\text{audio}}$ ↑	Si-SDR ↑
Low bitrates	DAC	1.7	86 Hz	2	3.24	0.108	1.71	0.63	**-4.11**
	SemantiCodec	1.4	100 Hz	1	1.79	0.136	4.93	0.60	-31.8
	S-PRESSO	1.32	11 Hz	12	**0.55**	**0.048**	**0.728**	**0.73**	-4.48
Ultra low bitrates	SemantiCodec	0.3125	25 Hz	1	1.23	0.271	2.70	0.48	-34.5
	S-PRESSO	0.3	1 Hz	25	**0.64**	**0.052**	**0.78**	**0.71**	**-27.8**
	S-PRESSO	0.096	1 Hz	8	0.68	0.060	0.89	0.67	-30.4

Table 2: Performance in the discrete case of S-PRESSO vs. baseline audio codecs at equivalent bitrates.

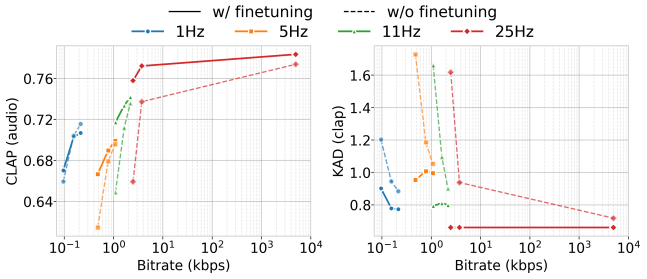


Fig. 4: Evaluation of S-PRESSO at varying bitrates and framerates.

5. RESULTS

For all experiments, we sample S-PRESSO reconstructions using the Heun solver [19] with 64 steps and the default parameters.

Overall Performance: As shown in the continuous compression benchmark Table 1, the AudioAE provides the upper bound on achievable quality. Relative to this reference, S-PRESSO consistently outperforms continuous baselines at comparable compression rates, achieving superior audio quality (FAD, KAD, Si-SDR) and acoustic similarity (CLAP), even at extreme compression rates (1 Hz). As the framerate increases, i.e. more temporal information is preserved, SI-SDR improves accordingly. By contrast, CLAP similarity remains remarkably stable across framerates. This stability shows that reconstructions globally preserve acoustic category and source identity regardless of temporal downsampling.

Table 2 reports results in the discrete case. S-PRESSO outperforms Semanticodec across all metrics, with the latter's 16 kHz bandwidth likely explaining its weaker FAD and KAD. These results confirm that bitrate does not constrain either perceptual quality or global similarity. Finally, as shown in Figure 3, the MUSHRA test corroborates these findings, showing superior ratings for S-PRESSO at low and ultra–low bitrates. However, both models remain below the reference, partly because the variability of diffusion sampling complicates subjective judgments, forcing listeners to balance quality against similarity.

Impact of the bitrate: In Figure 4, we compare the performance of S-PRESSO in the discrete case for varying bitrates and framerates. The results further show that finetuning (step 3) consistently improves performance across bitrates, even though the model was not explicitly trained for variable bitrate settings. At a fixed framerate, more codebooks yield higher scores, reflecting finer residual modeling. Conversely, at a fixed bitrate, higher framerates perform better, highlighting the cost of framerate reduction.

Reconstruction diversity: Empirically, at fixed compression rate, we observe more diversity with low frame rates. Specifically, higher frame rates capture fine local structure, while lower frame rates (e.g., 1Hz, encoding a single vector) emphasize global information, leading to coarser reconstructions and increased variability across samples. This usually implies subtle differences in audio textures, high frequency details and background noise. Interestingly, we also observe that the model tends to replace background noise with other noise patterns, suggesting that it prioritizes preserving salient audio elements while freely re-synthesizing less critical components.

6. CONCLUSION

We introduced **S-PRESSO**, a diffusion autoencoder for ultra–low bitrate compression of 48 kHz sound effects. Leveraging diffusion priors and strong multimodal conditioning, S-PRESSO achieves up to 750× compression while preserving perceptual quality. Our results suggest to shift the focus from strict fidelity to acoustic similarity enabling realistic and semantically consistent reconstructions even at 1Hz frame rates. Beyond surpassing continuous and discrete baselines, this work highlights the potential of generative models to redefine the limits of neural audio compression. While this work focused on audio quality, future efforts will make S-PRESSO more practical by speeding up inference time and extending it to general audio.

7. REFERENCES

[1] Marlene Careil, Matthew J Muckley, Jakob Verbeek, and Stéphane Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *The Twelfth International Conference on Learning Representations*, 2023.

[2] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco 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] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[4] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar, "High-fidelity audio compression with improved rvqgan," *Advances in Neural Information Processing Systems*, vol. 36, pp. 27980–27993, 2023.

[5] Ali Siahkoohi, Michael Chinen, Tom Denton, W. Kleijn, and Jan Skoglund, "Ultra-low-bitrate speech coding with pretrained transformers," 09 2022, pp. 4421–4425.

[6] 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 Journal of Selected Topics in Signal Processing*, 2024.

[7] Yaoxun Xu, Hangting Chen, Jianwei Yu, Wei Tan, Rongzhi Gu, Shun Lei, Zhiwei Lin, and Zhiyong Wu, "Mucodec: Ultra low-bitrate music codec," *arXiv preprint arXiv:2409.13216*, 2024.

[8] Zineb Lahrichi, Gaëtan Hadjeres, Gaël Richard, and Geoffroy Peeters, "QINCODEC: neural audio compression with implicit neural codebooks," in *European Signal Processing Conference (EUSIPCO)*, 2025.

[9] Edresson Casanova, Ryan Langman, Paarth Neekhara, Shehzeen Hussain, Jason Li, Subhankar Ghosh, Ante Jukić, and Sang-gil Lee, "Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[10] Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu, "Dualcodec: A low-frame-rate, semantically-enhanced neural audio codec for speech generation," *arXiv preprint arXiv:2505.13000*, 2025.

[11] W Bastiaan Kleijn, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, Florian Stimberg, Quan Wang, and Thomas C Walters, "Wavenet based low rate speech coding," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018, pp. 676–680.

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al., "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, pp. 3, 2022.

[13] Hubert Siuzdak, "Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis," *arXiv preprint arXiv:2306.00814*, 2023.

[14] Jordi Pons, Santiago Pascual, Giulio Cengarle, and Joan Serrà, "Upsampling artifacts in neural audio synthesis," *CoRR*, 2020.

[15] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al., "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first international conference on machine learning*, 2024.

[16] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, pp. 127063, 2024.

[17] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al., "Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space," *arXiv e-prints*, pp. arXiv–2506, 2025.

[18] Théophane Vallaeys, Matthew Muckley, Jakob Verbeek, and Matthijs Douze, "Qinco2: Vector compression and search with improved implicit neural codebooks," *arXiv preprint arXiv:2501.03078*, 2025.

[19] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine, "Elucidating the design space of diffusion-based generative models," *Advances in neural information processing systems*, vol. 35, pp. 26565–26577, 2022.

[20] Yusong Wu, K. Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2022.

[21] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons, "Stable audio open," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[22] Marco Pasini, Stefan Lattner, and George Fazekas, "Music2latent: Consistency autoencoders for latent audio compression," *arXiv preprint arXiv:2408.06500*, 2024.

[23] Yoonjin Chung, Pilsun Eu, Junwon Lee, Keunwoo Choi, Juhan Nam, and Ben Sangbae Chon, "Kad: No more fad! an effective and efficient evaluation metric for audio generation," *arXiv preprint arXiv:2502.15602*, 2025.

[24] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre, "webmushra — a comprehensive framework for web-based listening tests," *Journal of open research software*, vol. 6, 2018.