EXPLORING TOKEN-SPACE MANIPULATION IN LATENT AUDIO TOKENIZERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural audio codecs provide compact discrete representations for speech generation and manipulation. However, most codecs organize tokens as frame-level sequences, making it difficult to study or intervene on global factors of variation. In this work, we propose the Latent Audio Tokenizer for Token-space Editing (LATTE), which appends a fixed set of learnable latent tokens to the audio feature sequence and retains only these tokens for quantization and decoding. This design produces a compact, non-temporally aligned bottleneck in which each token can aggregate global information across the full utterance. We show that the resulting tokenizer preserves competitive reconstruction quality in low-bitrate speech coding settings while enabling simple token-space interventions. In particular, we find that swapping selected latent token positions between utterances can modify global attributes, such as speaker identity and background noise, and we evaluate these interventions on voice conversion and denoising tasks. Our results suggest that compact latent audio tokenizers can support controllable audio manipulation without task-specific supervised editing models. Audio samples are included in the anonymous supplementary material.

1 INTRODUCTION

Neural audio codecs have become the dominant paradigm in speech representation learning, generation, and manipulation (Zeghidour et al., 2021; Défossez et al., 2023; Kumar et al., 2023) with applications to the multimodal domain (Grattafiori et al., 2024; Jiang et al., 2024; Comanici et al., 2025; Singh et al., 2025; DeepSeek-AI et al., 2025). By compressing waveforms into sequences of discrete tokens, they provide a universal language that connects acoustic signal processing with modern language modeling pipelines (Borsos et al., 2023; Kreuk et al., 2023; Wang et al., 2023; Nguyen et al., 2024; Défossez et al., 2024; Labiausse et al., 2025; Zeghidour et al., 2025). The dominant approach consists of generating *frame-level* token streams: one or more codes per fixed-duration frame, organized into long, locally aligned sequences via residual vector quantization (RVQ) (Zeghidour et al., 2021). This structure is well-suited to waveform fidelity and left-to-right generation, but it limits how easily one can study or intervene on *global* factors of variation, such as speaker identity, background noise, or speaking style. Because each token represents a narrow temporal window, isolating or manipulating a global attribute requires identifying *which* frames carry it, an inherently diffuse and challenging problem.

A complementary line of work takes a different structural approach: rather than representing audio as a long sequence of frames, it compresses the entire utterance into a set of learned latent tokens that serve as a bottleneck before reconstruction. TiTok (Yu et al., 2024) introduced this idea for images, replacing patch-level tokens with a fixed number of latent queries that aggregate global information before discrete quantization. ALMTokenizer (Yang et al., 2025) adapted a similar strategy to audio, combining masked autoencoding and autoregressive objectives to produce a compact token sequence suited to language modeling. These architectures suggest a natural hypothesis: if the model must reconstruct the full signal from a small set of non-temporally aligned slots, individual slots may specialize to encode distinct global factors, possibly leading to a structured latent space that is more transparent and more amenable to targeted intervention.

We investigate this hypothesis directly. Concretely, we adapt FocalCodec (Della Libera et al., 2025), a low-bitrate, single-codebook speech codec based on WavLM (Chen et al., 2022) features and focal

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.

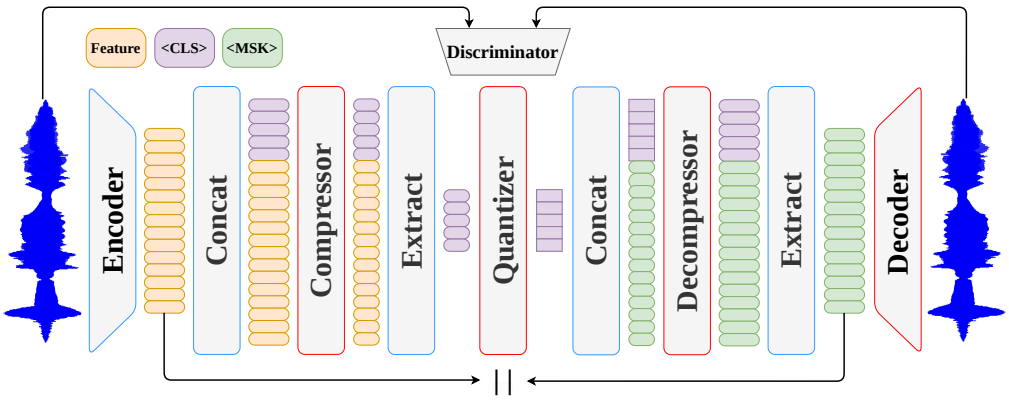


Figure 1: **LATTE turns frame-level codec features into compact latent slots for analysis and editing.** A frozen FocalCodec front-end maps speech to acoustic-semantic features; a learned TiTok-style bottleneck keeps only quantized latent slots; importance scores identify slots associated with global factors, which can then be swapped between utterances for zero-shot speaker and noise edits. Rounded edges denote continuous representations, and squared edges denote discrete representations; blue borders indicate frozen modules, while red borders indicate learned modules. `Feature` represents WavLM layer-6 features. `<CLS>` represents the learnable latent tokens appended to the audio-feature sequence. `<MSK>` represents the learnable mask tokens used to initialize reconstruction.

modulation (Yang et al., 2022; Della Libera et al., 2024), into a TiTok-style latent audio tokenizer. While the same pipeline can be applied to different architectures, we chose FocalCodec for its effective and efficient representation of both acoustic and semantic content. We keep the FocalCodec encoder and decoder frozen, and insert a learned compressor–quantizer–decompressor chain between them. The compressor appends L learned latent queries to the codec feature sequence and runs a shared self-attention encoder; only the query positions are retained, discretized with Binary Spherical Quantization (BSQ) (Zhao et al., 2025), and passed to a symmetric decoder that reconstructs the full codec feature sequence. Because no temporal alignment is imposed, each latent slot is free to pool information from the entire utterance, possibly inducing *interpretable specialization* in the learned slots, independently of codec-specific inductive biases.

Our contributions are as follows:

- We introduce LATTE, a TiTok-style latent audio tokenizer that compresses an entire utterance into a set of global discrete tokens, enabling compact and partially controllable speech representations. Despite compressing a full utterance through a compact, low-bitrate discrete bottleneck, the proposed tokenizer preserves competitive reconstruction quality on standard low-bitrate speech-coding benchmarks, matching or approaching frame-level baselines.
- We perform a Jaccard-relevance analysis of the individual latent positions, showing that they are *not* uniformly involved across factor types: some slots concentrate mass relevant to noise or speaker identity, while others are more diffuse. This pattern is qualitatively different from standard frame-level tokens.
- We use slot-importance scoring to analyze and intervene in global factors, enabling zero-shot speaker and noise-token swaps. We find that swapping selected latent token positions between two utterances produces targeted shifts in global attributes: transferring noise-relevant slots changes the background acoustic environment, and transferring speaker-relevant slots partially transfers speaker identity, without supervised editing models or test-time optimization.

Taken together, these results show that the new tokenization strategy used in LATTE achieves performance competitive with state-of-the-art low-bitrate baselines, while also offering strong controllability and interpretability. Audio samples are included in the anonymous supplementary material.

2 Related Work

Speech Codecs. Most speech tokenizers discretize audio at a fixed temporal resolution (Mousavi et al., 2025). Early neural codecs emphasized acoustic fidelity at moderate bitrates, using con-

volutional encoder–decoder models and RVQ to reconstruct waveforms from frame-level code streams (Zeghidour et al., 2021; Défossez et al., 2023; Kumar et al., 2023). These acoustic codecs preserve fine detail, but their long, locally aligned token sequences make global attributes such as speaker identity or background condition diffuse across time. Semantic tokenizers make the opposite trade-off: they quantize self-supervised speech features, such as HuBERT or WavLM, to produce compact linguistic units for language modeling (Hsu et al., 2021; Chen et al., 2022; Mousavi et al., 2026). This improves semantic abstraction but discards prosody, speaker cues, and fine acoustic detail that must later be recovered by a decoder. Recent hybrid codecs aim to balance semantic usefulness and acoustic reconstruction. SpeechTokenizer (Zhang et al., 2024) and Mimi (Défossez et al., 2024) distill semantic information into codec tokens; SemantiCodec (Liu et al., 2024) combines semantic tokens with a generative acoustic decoder; and WavTokenizer (Ji et al., 2025), BigCodec (Xin et al., 2024), Stable Codec (Parker et al., 2025), and FocalCodec (Della Libera et al., 2025; 2026) explore low-bitrate, single-codebook, or streamable designs with stronger token structure. Our work is orthogonal to this fixed-rate progression: rather than improving a frame-aligned token stream, we ask whether a compact set of non-temporally aligned latent slots can expose and manipulate global factors.

Compact Latent Bottlenecks. Highly compressed tokenizers replace long patch- or frame-level sequences with a small set of learned tokens that aggregate information before quantization. TiTok (Yu et al., 2024) introduced this idea for images by processing patches jointly with learned latent queries and keeping only the query outputs for vector quantization. ALMTokenizer (Yang et al., 2025) adapts related ideas to audio, using learnable queries, semantic-prior initialization, masked autoencoding, and autoregressive objectives to produce compact sequences for audio language modeling. We do not directly compare with ALMTokenizer because public code and checkpoints are unavailable at the time of submission. Instead, we isolate the representation question: whether this learned-token bottleneck induces factor-dependent structure that can be probed and edited.

Latent Space Controllability. Discrete latent spaces can support manipulation when individual token subsets have predictable functional roles (Beyer et al., 2025). Image tokenizers such as TiTok demonstrate that compact learned tokens can retain enough global information for generation and reconstruction (Yu et al., 2024); our question is whether analogous audio tokens expose factor structure that can be manipulated directly. We therefore use an audio-domain causal probe: estimate which latent slots are associated with a global factor, swap those slots across utterances, and measure whether the decoded waveform moves along the intended attribute.

3 Latent Audio Tokenizer

We use FocalCodec (Della Libera et al., 2025) as a frozen speech-codec substrate and replace its original frame-level bottleneck with a learned latent-token bottleneck. This lets us test whether non-temporally aligned discrete slots develop interpretable specialization while keeping the codec feature space and waveform decoder fixed. An overview of the pipeline is presented in Figure 1.

Let $\mathbf{x} \in \mathbb{R}^S$ denote a speech segment sampled at $f_s = 16\,\mathrm{kHz}$, with $S = f_s\tau$ for chunk duration τ ($\tau = 5\,\mathrm{s}$ by default). FocalCodec maps the waveform to WavLM layer-6 features (Chen et al., 2022), yielding a representation space rich in acoustic and semantic content. We denote the frozen front-end as

$$\mathcal{E}_{\mathrm{FC}} : \mathcal{X} \to \mathbb{R}^{T \times H}, \qquad \mathbf{F} = \mathcal{E}_{\mathrm{FC}}(\mathbf{x}), \tag{1}$$

where $H = 1024$ and $T = 50\tau$. We insert a compressor–quantizer–decompressor chain between the frozen encoder and decoder:

$$\mathbf{x} \xrightarrow{\mathcal{E}_{\mathrm{FC}}} \mathbf{F} \xrightarrow{g_\theta} \mathbf{Z} \xrightarrow{q_\phi} \mathbf{C}, \mathbf{k} \xrightarrow{h_\psi} \hat{\mathbf{F}} \xrightarrow{\mathcal{D}_{\mathrm{FC}}} \hat{\mathbf{x}}, \tag{2}$$

where g_θ is a learned compressor, q_ϕ a discrete quantizer that returns quantized vectors and their integer indices, and h_ψ a learned decompressor; $\mathcal{E}_{\mathrm{FC}}$ and $\mathcal{D}_{\mathrm{FC}}$ remain frozen throughout training.

Latent Slot Compression. The compressor summarizes the codec feature sequence into L learned latent slots (Yu et al., 2024; Yang et al., 2025). We set $L = r\tau$, where r is the target token rate; the default $r = 50\,\mathrm{Hz}$ gives $L = 250$ tokens per chunk.

We introduce L learned queries $\mathbf{Q} \in \mathbb{R}^{L\times H}$, concatenate them with the positionally-embedded codec features, and process the joint sequence through a FocalCodec-style encoder:

$$\mathbf{Y} = \left[\mathbf{F} + \mathbf{P}^{\text{feat}}_{1:T} \,;\, \mathbf{Q} + \mathbf{P}^{\text{lat}}_{1:L}\right], \qquad \mathbf{Z} = g_\theta(\mathbf{Y})_{T+1:T+L} \in \mathbb{R}^{L\times d}, \tag{3}$$

where $\mathbf{P}^{\text{feat}}$ and $\mathbf{P}^{\text{lat}}$ are learned positional embeddings. Only the L latent positions are retained, so each slot can pool information from the full feature sequence while reconstruction must flow exclusively through these L vectors.

Binary Spherical Quantization. Each latent vector $\mathbf{z}_\ell \in \mathbb{R}^d$ is quantized with BSQ (Zhao et al., 2025), yielding a discrete code $k_\ell \in \{0, \dots, 2^d - 1\}$ and a quantized vector $\mathbf{c}_\ell \in \mathbb{R}^d$. Collecting over all slots:

$$\mathbf{C} \in \mathbb{R}^{L\times d}, \qquad \mathbf{k} \in \{0, \dots, 2^d - 1\}^L. \tag{4}$$

We use $d = 13$ (vocabulary size $2^{13} = 8192$); at rate r this gives a bitrate of rd bits/s—$650\,\text{bit}\,\text{s}^{-1}$ at default settings. BSQ also contributes an auxiliary regularizer $\mathcal{L}_{\text{BSQ}}$ that promotes balanced codebook usage.

Feature Reconstruction. The decompressor maps the quantized sequence back to the original frame rate. Its input concatenates T learned reconstruction slots with the positionally-embedded quantized codes:

$$\mathbf{U} = \left[\mathbf{m} \otimes \mathbf{1}_T + \mathbf{P}^{\text{mask}}_{1:T} \,;\, \mathbf{C} + \mathbf{P}^{\text{code}}_{1:L}\right], \tag{5}$$

where $\mathbf{m} \in \mathbb{R}^d$ is a learned mask embedding and $\mathbf{P}^{\text{mask}}$, $\mathbf{P}^{\text{code}}$ are learned positional embeddings. Reconstructed codec features are read from the first T output positions:

$$\hat{\mathbf{F}} = h_\psi(\mathbf{U})_{1:T} \in \mathbb{R}^{T\times H}. \tag{6}$$

Training Objective. The tokenizer minimizes reconstruction error in the frozen codec feature space:

$$\mathcal{L}(\theta, \phi, \psi) = \left\|\hat{\mathbf{F}} - \mathbf{F}\right\|_2^2 + \lambda \mathcal{L}_{\text{BSQ}}, \tag{7}$$

where the squared-error term is averaged over non-padded frames and λ weights the quantization regularizer. We use no waveform-domain loss, and the FocalCodec encoder and decoder are not updated.

4 Token Importance Scoring

TiTok-style bottlenecks create a fixed set of latent positions, but they do not specify what each position should encode. We therefore use slot importance as a post-hoc probe of factor structure (Beyer et al., 2025). For a factor of variation such as speaker identity, accent, or noise level, we assign a scalar relevance score g_ℓ to every latent index $\ell \in \{1, \dots, L\}$. Given a dataset annotated with a factor label, utterances are partitioned into J groups. For each partition j we accumulate the sample-mean code vector at every token position:

$$\boldsymbol{\mu}_{j,\ell} = \frac{1}{N_j} \sum_{n\in\mathcal{P}_j} \mathbf{c}_{n,\ell} \;\in \mathbb{R}^d, \tag{8}$$

where $\mathcal{P}_j$ is the index set of partition j and $N_j = |\mathcal{P}_j|$. We then form the matrix $\mathbf{M}_\ell \in \mathbb{R}^{J\times d}$ whose j-th row is $\boldsymbol{\mu}_{j,\ell}$, and let $\mathbf{X}_\ell = \mathbf{M}_\ell - \mathbf{1}_J \bar{\boldsymbol{\mu}}_\ell^\top$ be its row-centred form, where $\bar{\boldsymbol{\mu}}_\ell = \frac{1}{J}\sum_j \boldsymbol{\mu}_{j,\ell}$. The importance score is then

$$g_\ell = \frac{\sigma_{\max}(\mathbf{X}_\ell)^2}{J - 1}, \tag{9}$$

where $\sigma_{\max}(\mathbf{X}_\ell)$ is the largest singular value of $\mathbf{X}_\ell$. This equals the leading eigenvalue of the sample between-partition covariance of the mean codes at position ℓ, and measures how much the mean code at that slot spreads across factor groups in all d dimensions simultaneously. A high score indicates that position ℓ carries factor-specific information; a low score indicates factor-invariance.

Scores are computed separately for each factor of interest, yielding one importance vector $\mathbf{g} = (g_1, \dots, g_L) \in \mathbb{R}^L$ per factor. To characterize how concentrated factor information is across slots, we

treat $\mathbf{g}$ as an unnormalized distribution and compute entropy, Gini coefficient, and Jaccard overlap between importance vectors from different factor partitions. These metrics reveal the degree to which individual slots specialize to distinct global attributes rather than spreading information uniformly.

The importance vectors also define a mechanism for zero-shot attribute transfer. Given a source utterance with quantized codes $\mathbf{C}^{(s)} \in \mathbb{R}^{L\times d}$ and a target utterance with codes $\mathbf{C}^{(t)}$, we first rank the token positions by decreasing factor importance. Let π be a permutation such that $g_{\pi_1} \geq g_{\pi_2} \geq \cdots \geq g_{\pi_L}$. We then select the smallest set of top-ranked positions whose cumulative importance reaches a prescribed fraction γ:

$$m_\gamma = \min\left\{ m : \sum_{i=1}^{m} \bar{g}_{\pi_i} \geq \gamma \right\}, \qquad \mathcal{S}_\gamma = \{\pi_1, \ldots, \pi_{m_\gamma}\}, \tag{10}$$

where $\bar{g}_\ell = g_\ell / \sum_{j=1}^{L} g_j$ denotes the normalized importance score. We construct the swapped code matrix as

$$\tilde{\mathbf{C}}_\ell^{(s)} = \begin{cases} \mathbf{C}_\ell^{(t)} & \ell \in \mathcal{S}_\gamma, \\ \mathbf{C}_\ell^{(s)} & \text{otherwise.} \end{cases} \tag{11}$$

The edited code matrix is then mapped back to codec features by h_ψ and decoded by $\mathcal{D}_{\mathrm{FC}}$ to produce the edited waveform. The same operation can equivalently be viewed as replacing the corresponding discrete indices $\mathbf{k}$ before codebook lookup.

5 Experiments

We evaluate LATTE along three axes. First, we measure resynthesis quality to verify that its compact latent representation preserves intelligibility, speaker information, and perceptual quality under standard codec benchmarks. Second, we analyze the learned latent slots by computing slot-wise importance scores for several annotated global factors: speaker identity, speaker gender, accent, and background noise level. Third, we evaluate whether these importance scores can be used to select slots for predictable token-space edits.

Our experiments are designed to answer the following questions: (i) Does LATTE remain competitive with existing neural codec tokenizers in resynthesis quality? (ii) Do different latent slots exhibit non-uniform associations with interpretable audio factors? (iii) Can these slot-factor associations be used to perform targeted edits while preserving non-target properties?

5.1 Datasets and Tasks

All audio is resampled to 16 kHz. For resynthesis, we follow the evaluation protocol of FocalCodec (Della Libera et al., 2025) and report results on LibriSpeech test-clean (Panayotov et al., 2015), VoiceBank (Valentini-Botinhao et al., 2016), and Libri1Mix (Cosentino et al., 2020). These benchmarks cover clean speech, noisy speech, and speech mixtures, allowing us to evaluate reconstruction quality across increasingly challenging acoustic conditions.

We compare LATTE against a broad set of neural codec and speech-tokenizer baselines, including EnCodec (Défossez et al., 2023), DAC (Kumar et al., 2023), WavLM6-KM (Chen et al., 2022), SpeechTokenizer (Zhang et al., 2024), SemantiCodec (Liu et al., 2024), Mimi (Défossez et al., 2024), WavTokenizer (Ji et al., 2025), BigCodec (Xin et al., 2024), Stable Codec (Parker et al., 2025), and FocalCodec at 50 Hz (Della Libera et al., 2025). Unless otherwise stated, baseline numbers follow the common FocalCodec evaluation pipeline; checkpoint and metric details are listed in Appendix C.

For slot-importance analysis, we construct factor partitions from dataset metadata or controlled corruptions. For speaker identity, each speaker is treated as a separate class, and we compute partitions on both VCTK (Yamagishi et al., 2017) and LibriTTS (Zen et al., 2019) to evaluate cross-dataset stability. For speaker gender, we use LibriTTS and VCTK metadata. For accent, we use VCTK labels. For noise-level analysis, we corrupt LibriTTS utterances with additive white noise and WHAM! noise at 0, 5, 10, 20, and 40 dB signal-to-noise ratio (SNR), testing whether the same latent slots are associated with degradation level across noise distributions.

For token-space editing, we evaluate speaker- and noise-intervention methods. Speaker editing uses a parallel VCTK voice-conversion setting: the target is a clean, different-speaker utterance with

Table 1: Speech resynthesis results on clean and noisy speech benchmarks. Higher UTMOS, DNSMOS, and speaker similarity (Sim) are better; lower dWER is better.

Codec	Bitrate (kbps) ↓	*Clean – LibriSpeech test-clean*			*Noisy – VoiceBank*			*Noisy – Libri1Mix*		
		UTMOS ↑	dWER ↓	Sim ↑	DNSMOS ↑	dWER ↓	Sim ↑	DNSMOS ↑	dWER ↓	Sim ↑
Reference	—	4.09	0.00	100.0	3.56	0.00	100.0	3.73	0.00	100.0
EnCodec	1.50	1.58	8.08	93.8	2.76	28.16	87.7	2.40	55.17	86.3
DAC	1.00	1.29	20.04	89.2	2.72	63.90	79.8	2.40	90.92	76.6
WavLM6-KM	0.45	3.75	6.20	90.0	3.06	20.67	82.9	2.87	<u>36.60</u>	85.9
SpeechTokenizer	1.00	2.28	5.14	91.6	2.74	34.51	82.2	2.58	57.26	82.8
SemantiCodec	0.65	2.91	8.97	96.0	3.13	31.46	90.6	2.67	51.18	89.9
Mimi	0.69	3.29	5.73	96.0	3.01	28.00	87.8	2.65	49.14	89.4
WavTokenizer	0.48	3.78	11.55	95.4	3.09	42.12	89.8	2.53	70.10	86.3
BigCodec	1.04	4.11	<u>2.55</u>	**98.5**	3.19	20.67	**92.3**	2.75	53.26	88.3
Stable Codec	0.70	**4.32**	4.97	94.7	**3.33**	20.32	88.8	2.91	43.52	90.0
FocalCodec	0.65	4.05	**2.18**	<u>97.4</u>	3.16	**8.08**	91.3	2.93	**27.89**	<u>91.6</u>
LATTE LARGE	0.65	<u>4.23</u>	5.82	<u>97.4</u>	<u>3.29</u>	<u>16.30</u>	<u>91.6</u>	**3.03**	39.07	91.5
LATTE BASE	0.65	4.20	7.07	96.9	3.26	19.51	91.0	<u>2.96</u>	41.44	**91.7**

the same sentence as the source. Denoising uses non-self-clean targets selected by a deterministic cyclic shift over the corresponding evaluation list, so the target is clean but generally not text- or speaker-matched. Full pairing details are provided in Appendix D.

We evaluate two asymmetric encoder–decoder configurations, denoted LATTE BASE and LATTE LARGE. In our notation, the first label indicates encoder scale and the second indicates decoder scale: BL corresponds to a Base encoder with a Large decoder, while LATTE LARGE uses a Large encoder with an XL decoder. Both variants share the same tokenization objective and latent interface, but differ in how representational capacity is allocated between encoder and decoder. Exact hyperparameters for both settings are reported in Appendix F.

We report UTokyo-SaruLab mean opinion score (UTMOS) (Saeki et al., 2022) for clean speech, Deep Noise Suppression mean opinion score (DNSMOS) (Reddy et al., 2022) for noisy speech, differential word error rate (dWER) (Wang et al., 2021) for linguistic preservation, and speaker similarity (Sim) from a pretrained speaker-verification model. Metric checkpoints and preprocessing details are provided in Appendix C.

6 RESULTS

6.1 RESYNTHESIS QUALITY

Table 1 reports resynthesis quality on LibriSpeech test-clean, VoiceBank, and Libri1Mix. Across all three benchmarks, we evaluate LATTE LARGE, the full model with overlap-add (OLA) chunking at a one-second window, and LATTE BASE, a smaller asymmetric variant at the standard 50 Hz token rate.

Clean Speech. On LibriSpeech test-clean, LATTE LARGE achieves an UTMOS of 4.23, the highest among the latent tokenizer variants and comparable to FocalCodec (4.05) and Stable Codec (4.32), the two strongest baselines on this metric. Speaker similarity (Sim = 97.4) matches FocalCodec, indicating that the latent slot bottleneck preserves speaker identity faithfully despite operating on a non-temporally aligned representation. The dWER of 5.82 is higher than FocalCodec (2.18) and BigCodec (2.55), suggesting a moderate intelligibility cost attributable to the latent-slot compression; the bottleneck must reconstruct fine phonetic detail from global, position-unaligned tokens rather than frame-level codes.

Noisy Speech. On VoiceBank, LATTE LARGE achieves a DNSMOS of 3.29, second only to Stable Codec (3.33) and above all other baselines, including FocalCodec (3.16). Its dWER of 16.30 is the second-best noisy-speech result, behind FocalCodec (8.08), and substantially better than most higher-bitrate baselines such as EnCodec (28.16) and DAC (63.90). On Libri1Mix, LATTE LARGE achieves the highest DNSMOS (3.03) and approximately matches FocalCodec in Sim (91.5 vs. 91.6), while its dWER of 39.07 is again mid-table.

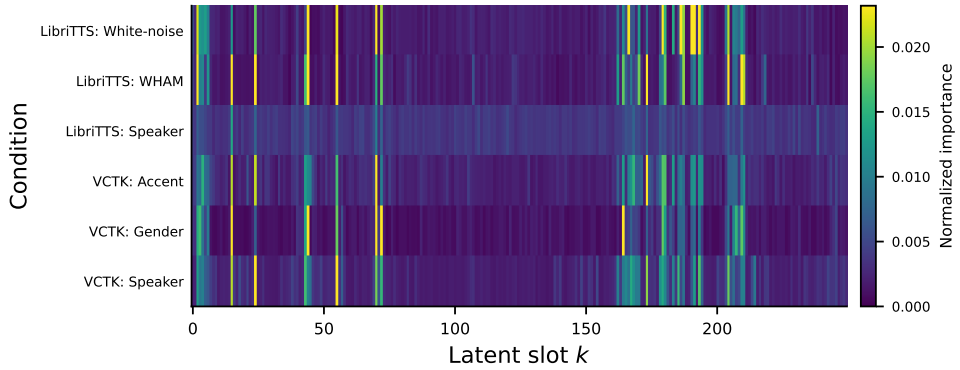


Figure 2: Row-normalized slot-importance profiles for different factor partitions. Each row is ℓ_1-normalized across slots. The heatmap shows that importance is non-uniform and factor-dependent, with noise-related partitions emphasizing similar slot regions across corruption types.

Table 2: Structure of slot-importance profiles. Left: profile concentration, measured by entropy over the row-normalized importance distribution and by the Gini coefficient. Right: similarity between selected profile pairs, measured by Spearman correlation over all slots and Jaccard overlap among the top-ranked slots. Lower entropy and higher Gini indicate stronger concentration. We abbreviate speaker as Spk., LibriTTS as L, and VCTK as V.

Profile concentration

Profile	Ent. ↓	Gini ↑
Noise: White	4.86	0.55
Noise: WHAM!	4.78	0.57
Speaker: L	5.49	0.13
Accent: V	5.16	0.42
Gender: V	4.04	0.70
Speaker: V	5.18	0.42

Profile similarity

Comparison	ρ	Jacc@5	Jacc@10
White vs WHAM!	0.735	0.111	0.177
Spk.: L vs Spk.: V	0.308	0.429	0.429
Spk.: V vs Accent: V	0.515	0.667	0.429
Spk.: V vs Gender: V	0.433	0.250	0.250

Across all three benchmarks, the gap between LATTE LARGE and FocalCodec on dWER is the most consistent difference: the latent-slot design sacrifices some word-level intelligibility relative to a frame-rate codec at the same bitrate, while matching or exceeding it on perceptual quality and speaker similarity. This trade-off reflects the core design choice: by discarding temporal frame alignment in favour of a compact set of global tokens, the model gains a representation that is more amenable to token-space manipulation at the cost of slightly looser phonetic reconstruction.

6.2 SLOT-IMPORTANCE ANALYSIS

We next ask whether LATTE's latent positions behave as interchangeable containers or develop factor-dependent structure. For each annotated partition, we compute the TiTok-style slot-importance score from Section 4 and normalize the resulting profile to sum to one across slots. We evaluate speaker identity, speaker gender, accent, and noise level. Speaker and gender partitions are computed on both LibriTTS and VCTK; noise partitions are computed on LibriTTS using white noise and WHAM! (Wichern et al., 2019) noise across multiple SNRs.

Figure 2 shows that slot importance is strongly non-uniform. Most factors place a considerable share of their mass on a small subset of latent positions, and the active regions vary by factor. This rules out a purely exchangeable view of the latent sequence: even though the tokens are not temporally aligned, their positions acquire distinct functional roles. At the same time, the profiles are not one-hot or perfectly disjoint, which already suggests partial specialization rather than hard disentanglement. Table 2-left quantifies this pattern with entropy and the Gini index. Noise-related profiles have similar concentration across white noise and WHAM! noise. VCTK gender is the most concentrated profile,

Table 3: Noisy-set resynthesis compared with mass-based noise-slot replacement. Each corpus reports LATTE LARGE and mass-based slots at cumulative thresholds $\gamma \in \{0.02, 0.10\}$. Noise type appears as two four-column groups on each row (**WHAM** vs **white noise**), with DNSMOS, dWER, and Sim (higher DNSMOS and Sim are better; lower dWER is better). The resynthesis baseline is noise-family agnostic, so DNSMOS / dWER / Sim are repeated under both groups.

		WHAM			**White noise**		
Corpus	**Setting**	**DNSMOS ↑**	**dWER ↓**	**Sim ↑**	**DNSMOS ↑**	**dWER ↓**	**Sim ↑**
VoiceBank	LATTE (LARGE)	3.29	16.30	**91.6**	3.29	16.30	**91.6**
	γ=0.02	<u>3.50</u>	<u>11.23</u>	89.8	<u>3.59</u>	**11.52**	<u>90.8</u>
	γ=0.10	**3.58**	**9.54**	<u>90.4</u>	**3.60**	<u>11.61</u>	90.7
Libri1Mix	LATTE (LARGE)	3.03	39.07	**91.5**	3.03	39.07	**91.5**
	γ=0.02	<u>3.14</u>	**30.77**	<u>89.1</u>	<u>3.40</u>	<u>36.07</u>	87.9
	γ=0.10	**3.36**	<u>35.13</u>	86.3	**3.41**	**31.26**	<u>88.0</u>

whereas LibriTTS speaker identity is substantially more diffuse, indicating that different attributes occupy the latent sequence at different sparsity levels.

We then evaluate whether the same factors select stable slots across related partitions. Table 2-right reports Spearman correlations and top-k Jaccard overlaps between selected profile pairs. Noise-related profiles are the most stable: white-noise and WHAM!-noise partitions have a Spearman correlation of 0.735. Thus, the estimator appears to recover slots associated with degradation level rather than artifacts of a particular corruption process.

Speaker identity, gender, and accent show a more entangled pattern. VCTK speaker importance overlaps with both accent and gender, which is expected because these attributes are statistically correlated in the dataset. Speaker-related profiles also transfer less cleanly across datasets than noise profiles, suggesting that speaker importance depends more strongly on the population and partition used to estimate it. We therefore do not interpret the slots as isolated semantic factors. The evidence supports a weaker and more useful conclusion: the latent representation has structured, factor-dependent importance profiles, but related attributes remain partially entangled.

Overall, the latent slots are not interchangeable: different global factors induce distinct and, for noise, highly stable importance profiles. The profiles are nevertheless not fully separated across speaker-related attributes. This motivates the token-space editing experiments below, where we test whether structured but entangled importance profiles are still useful for targeted interventions. Cumulative-mass curves and per-factor diagnostic plots are provided in Appendix B.

6.3 TOKEN-SPACE EDITING

We evaluate denoising and parallel voice conversion by replacing importance-ranked latent slots with tokens from a reference utterance (Tables 3 and 4). These experiments are controlled interventions on the tokenizer, not fully trained conversion or enhancement systems.

Denoising. Mass-based noise-slot replacement Equations (10) and (11) improves perceptual quality on both noisy benchmarks. On VoiceBank, dWER drops from 16.30 under no-swap resynthesis to roughly 9.5–11.6 for the reported settings, while mean opinion score (MOS) also increases. On Libri1Mix, MOS increases and dWER generally falls into the low-to-mid 30s, a smaller but consistent gain under harder mixture conditions. Speaker similarity decreases slightly because clean references are generally unmatched in text and speaker.

Voice Conversion. Mass-based speaker-slot swaps reach high UTMOS while keeping parallel-VCTK dWER in the single-digit-to-low-teens range. Increasing the number of replaced slots raises target-speaker similarity but can increase dWER, revealing the expected edit-strength trade-off. We compare against two matched-budget controls: a random control, which swaps the same number of token slots chosen uniformly at random, and a least-important control, which swaps the same number of slots with the lowest importance scores. Both controls yield substantially lower target-speaker similarity, suggesting that the importance scores identify functionally relevant slots rather

Table 4: One-shot parallel voice conversion on VCTK (Yamagishi et al., 2017). We report reference systems and codec baselines in the first block, followed by LATTE LARGE token-swap results on `vctk_parallel_test_mic1`. For token swapping, γ denotes the cumulative-mass threshold and k the number of swapped tokens. Importance-guided token selection improves target-speaker similarity over random and least-important controls at matched edit budgets.

Method	γ	k	**Bitrate (kbps)** ↓	**UTMOS** ↑	**dWER** ↓	**Sim** ↑
Reference and codec baselines						
Reference	—	—	—	4.09	0.00	100.0
EnCodec	—	—	1.50	1.24	86.52	72.2
DAC	—	—	1.00	1.25	104.00	67.2
WavLM6-KM	—	—	**0.45**	2.90	26.68	**92.4**
SpeechTokenizer	—	—	1.00	1.49	<u>20.32</u>	81.2
SemantiCodec	—	—	0.65	2.02	106.00	72.8
Mimi	—	—	0.69	2.40	110.00	89.7
WavTokenizer	—	—	<u>0.48</u>	3.13	43.15	73.4
BigCodec	—	—	1.04	1.31	99.96	68.9
Stable Codec	—	—	0.70	3.76	27.63	71.1
FocalCodec	—	—	0.65	3.38	21.27	<u>92.2</u>
Importance-guided token swap						
LATTE (LARGE)	0.05	3	0.65	4.15	**11.24**	89.5
LATTE (LARGE)	0.07	4	0.65	4.14	12.53	89.6
LATTE (LARGE)	0.10	5	0.65	**4.16**	12.32	90.0
Matched-budget editing controls						
Random control	0.10	5	—	4.08	16.26	67.8
Least control	0.10	5	—	<u>4.10</u>	12.03	68.1

than arbitrary perturbation directions. Additional protocol details and interpretation guidelines are provided in Appendix D.

7 CONCLUSIONS

We introduced LATTE, a TiTok-style latent audio tokenizer that compresses speech into a compact set of non-temporally aligned discrete slots. Despite this global bottleneck, LATTE preserves competitive low-bitrate resynthesis quality and exposes a token interface suitable for analysis and intervention. Slot-importance profiles show that latent positions are not exchangeable: different factors concentrate on different subsets of slots, with noise profiles showing the strongest stability and speaker-related attributes remaining partially entangled. Using these profiles for token replacement enables simple zero-shot denoising and speaker-transfer interventions, outperforming random and least-important slot controls at matched edit budgets.

Overall, our results suggest that compact latent audio tokenizers can serve not only as efficient compression interfaces, but also as probes of global factor structure in audio representations. The current evidence supports partial specialization rather than hard disentanglement, and the token-swap results should be viewed as controlled interventions rather than full editing systems. Scaling latent-slot tokenization to more diverse audio domains and developing objectives that encourage cleaner factor separation are natural directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and debugging experimental methods, drafting portions of the manuscript, editing text for readability and conciseness, formatting references and LaTeX, brainstorming presentation choices, and searching for relevant literature. We did not use generative AI tools to develop the core model or conceptual framework, formulate the research hypotheses, design the experimental methodology, clean or reformat datasets, conduct qualitative or thematic data analysis, interpret the experimental results, generate synthetic datasets, or translate material. All AI-assisted material was reviewed by the authors: the experimental code was tested, reported results were checked against the underlying outputs, and manuscript text and citations were manually verified. The authors take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

The model architecture, latent-slot compression and decompression procedure, quantization objective, and training losses are specified in Section 3. The slot-importance estimator and token-swapping intervention are described in Section 4. The datasets, baselines, evaluation protocol, and model configurations used in the main experiments are reported in Section 5. The appendix provides the evaluation checkpoints and metric implementations in Section C, the editing and control protocols in Section D, the overlap-add inference procedure in Section E, and the architecture, optimization, data, and runtime hyperparameters in Section F. Anonymous supplementary material includes audio examples supporting the resynthesis, denoising, and speaker-transfer evaluations.

REFERENCES

Lukas Lao Beyer, Tianhong Li, Xinlei Chen, Sertac Karaman, and Kaiming He. Highly compressed tokenizer can generate without training. In *International Conference on Machine Learning (ICML)*, 2025.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 31:2523–2533, 2023.

Jonah Casebeer, Ge Zhu, Zhepei Wang, and Nicholas J. Bryan. A generative-first neural audio autoencoder. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 15107–15111, 2026.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, pp. 1505–1518, 2022.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. LibriMix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2025.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research (TMLR)*, 2023.

Luca Della Libera, Cem Subakan, and Mirco Ravanelli. Focal modulation networks for interpretable sound classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 853–857, 2024.

Luca Della Libera, Francesco Paissan, Cem Subakan, and Mirco Ravanelli. FocalCodec: Low-bitrate speech coding via focal modulation networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Luca Della Libera, Cem Subakan, and Mirco Ravanelli. FocalCodec-Stream: Streaming low-bitrate speech coding via causal distillation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17002–17006, 2026.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: A speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460, 2021.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, Wen Wang, and Zhou Zhao. WavTokenizer: An efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations (ICLR)*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, and Emma Bou Hanna others. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. AudioGen: Textually guided audio generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Tom Labiausse, Laurent Mazaré, Edouard Grave, Alexandre Défossez, and Neil Zeghidour. High-fidelity simultaneous speech-to-speech translation. In *International Conference on Machine Learning (ICML)*, 2025.

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. *arXiv preprint arXiv:2405.00233*, 2024.

Pooneh Mousavi, Gallil Maimon, Adel Moumen, Darius Petermann, Jiatong Shi, Haibin Wu, Haici Yang, Anastasia Kuznetsova, Artem Ploujnikov, Ricard Marxer, Bhuvana Ramabhadran, Benjamin Elizalde, Loren Lugosch, Jinyu Li, Cem Subakan, Phil Woodland, Minje Kim, Hung yi Lee, Shinji Watanabe, Yossi Adi, and Mirco Ravanelli. Discrete audio tokens: More than a survey! *Transactions on Machine Learning Research*, 2025.

Pooneh Mousavi, Jarod Duret, Darius Petermann, Artem Ploujnikov, Luca Della Libera, Anastasia Kuznetsova, Cem Subakan, and Mirco Ravanelli. DASB - discrete audio and speech benchmark. *Transactions on Machine Learning Research*, 2026.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux. SpiRit-LM: Interleaved spoken and written language model. *arXiv preprint arXiv:2402.05755*, 2024.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

Julian D. Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In *International Conference on Learning Representations (ICLR)*, 2025.

Chandan KA Reddy, Vishak Gopal, and Ross Cutler. DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Interspeech*, pp. 4521–4525, 2022.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Investigating RNN-based speech enhancement methods for noise-robust text-to-speech. In *Speech Synthesis Workshop*, pp. 146–152, 2016.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Zhong-Qiu Wang et al. Sequential multi-frame neural beamforming for speech separation and enhancement. In *IEEE Spoken Language Technology Workshop (SLT)*, pp. 905–911, 2021.

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

Detai Xin, Xu Tan, Shinnosuke Takamichi, and Hiroshi Saruwatari. BigCodec: Pushing the limits of low-bitrate neural speech codec. *arXiv preprint arXiv:2409.05377*, 2024.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 6:15, 2017.

Dongchao Yang, Songxiang Liu, Haohan Guo, Jiankun Zhao, Yuanyuan Wang, Helin Wang, Zeqian Ju, Xubo Liu, Xueyuan Chen, Xu Tan, Xixin Wu, and Helen M. Meng. ALMTokenizer: A low-bitrate and semantic-rich audio codec tokenizer for audio language modeling. In *International Conference on Machine Learning (ICML)*, 2025.

Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

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*, pp. 495–507, 2021.

Neil Zeghidour, Eugene Kharitonov, Manu Orsini, Václav Volhejn, Gabriel de Marmiesse, Edouard Grave, Patrick Pérez, Laurent Mazaré, and Alexandre Défossez. Streaming sequence-to-sequence learning with delayed streams modeling. *arXiv preprint arXiv:2509.08753*, 2025.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A corpus derived from LibriSpeech for text-to-speech. In *Interspeech*, 2019.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified speech tokenizer for speech large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. In *International Conference on Learning Representations (ICLR)*, 2025.

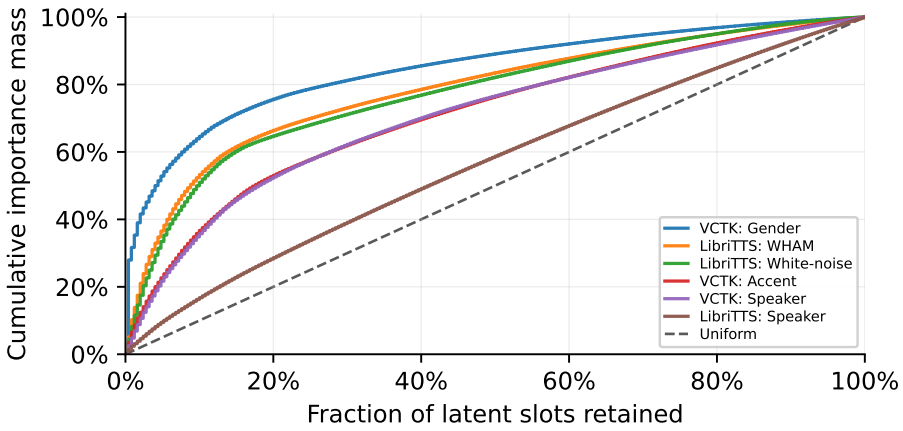


Figure 3: Cumulative importance mass after sorting slots by descending importance. Curves above the uniform baseline indicate that a factor is concentrated in a subset of slots. The absence of near-step-function behavior supports partial specialization rather than hard slot-factor disentanglement.

A Limitations and Broader Impacts

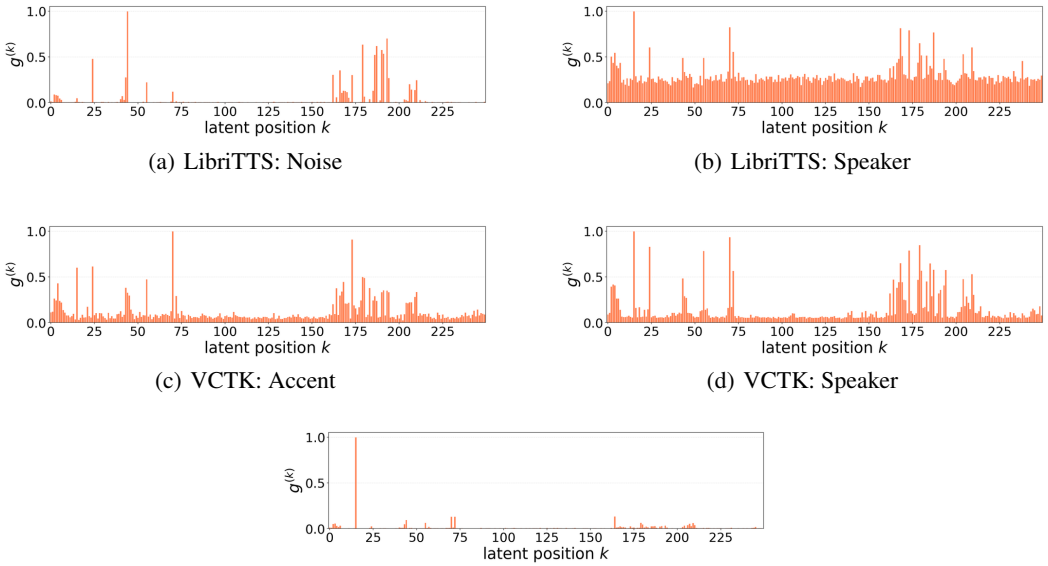
Limitations. A limitation of the present LATTE model is the scale and diversity of its training data. Although the architecture builds on a strong pretrained WavLM front-end, the LATTE-specific components inherited from the FocalCodec pipeline are trained on only a few hundred hours of clean English speech. This is substantially lower than that of recent competing tokenizers such as WavTokenizer, StableCodec, and Mimi, which are trained on much larger and more diverse datasets. This data gap may affect not only reconstruction quality and robustness, but also the reliability with which individual latent slots specialize to interpretable factors such as speaker identity, background noise, or channel conditions. Future work should therefore scale LATTE training to multilingual speech, noisy and reverberant conditions, mixtures, and broader audio domains to test whether the observed slot structure becomes more stable and general.

Broader Impacts. Controllable speech tokenization can support useful applications such as speech restoration, privacy-preserving factor analysis, and more efficient generative audio modeling. The same capability can also be misused for voice conversion or speaker-identity manipulation. We therefore frame the token-swap experiments as analysis tools rather than deployment-ready conversion systems, and any release of checkpoints should be accompanied by clear usage restrictions, provenance tracking, and detection or watermarking mechanisms where appropriate.

B Additional Slot-Importance Diagnostics

Cumulative Concentration. Figure 3 complements the entropy and Gini statistics in Table 2 by showing how quickly each importance profile accumulates mass when slots are sorted from most to least important. All curves lie above the uniform baseline, confirming that factor information is not spread evenly across latent positions. At the same time, none of the curves approaches a step function: even the most concentrated attributes require multiple slots to account for most of the importance mass. This supports the interpretation that LATTE develops partially specialized latent positions, rather than a hard one-slot-per-factor decomposition.

LATTE Slot Profiles. Figure 4 provides the unnormalized per-slot importance profiles underlying the aggregate results in Figure 3 and Table 2. The profiles show that high-importance slots are sparse but not isolated: several factors activate small groups of latent positions, and related factors often share part of their support. Noise profiles on LibriTTS exhibit recurring high-importance regions, consistent with the strong white-noise/WHAM! agreement reported in the main text. Speaker, accent, and gender profiles show more overlap, especially on VCTK, supporting our conclusion that speaker-related attributes are structured but partially entangled.

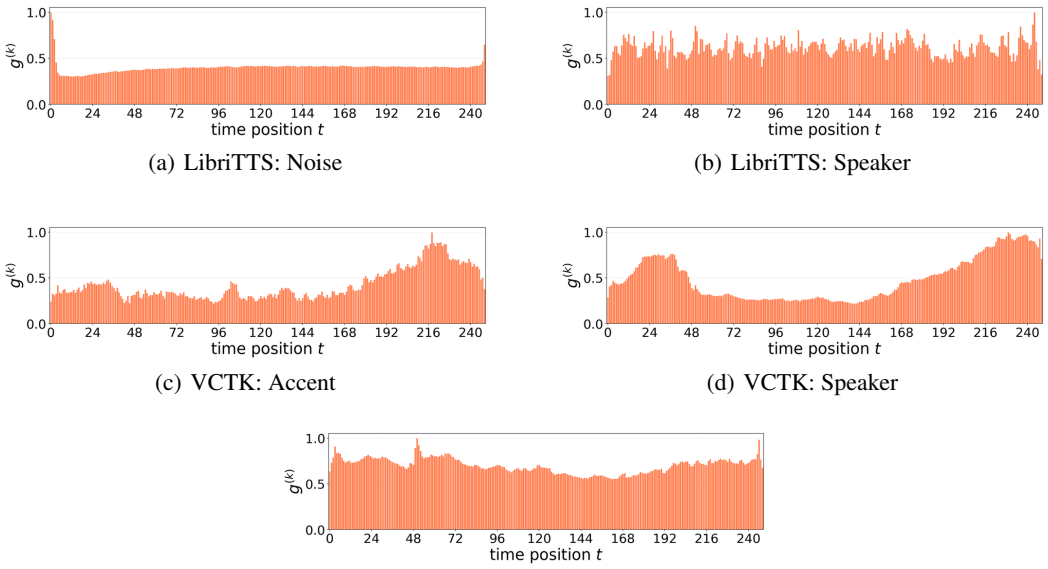


(a) LibriTTS: Noise (b) LibriTTS: Speaker

(c) VCTK: Accent (d) VCTK: Speaker

(e) VCTK: Gender

Figure 4: Per-factor importance scores for LATTE. Higher values indicate latent positions whose mean codes vary more strongly across the corresponding factor partition.



(a) LibriTTS: Noise (b) LibriTTS: Speaker

(c) VCTK: Accent (d) VCTK: Speaker

(e) VCTK: Gender

Figure 5: Per-factor importance scores for FocalCodec tokenization, computed with the same partition-based scoring procedure.

Comparison with Frame-Level Tokenization. Figure 5 applies the same scoring procedure to the original frame-level FocalCodec tokenization. Compared with Figure 4, these profiles are less naturally interpretable as slot-level factors, since token positions are tied to temporal frames rather than fixed latent roles. This contrast is important: in LATTE, a high-importance index refers to the same latent slot across utterances, making it meaningful to rank and replace positions for token-space intervention. In the frame-level representation, high scores instead reflect where factor-dependent variation appears over time, which is less directly suited to global attribute swapping.

C Evaluation Checkpoints and Metric Details

This appendix summarizes the external models, checkpoints, and preprocessing used for evaluation. Unless otherwise stated, all audio is converted to mono and resampled to 16 kHz before computing metrics. Our implementation follows the public evaluation protocol associated with FocalCodec (Della Libera et al., 2025).

UTMOS. For LibriSpeech test-clean, we report UTMOS as a non-reference perceptual speech-quality metric. We use the `utmos22_strong` model loaded via `torch.hub` from `tarepan/SpeechMOS:v1.2.0`. UTMOS is computed directly on the hypothesis waveform at 16 kHz.

DNSMOS. For VoiceBank and Libri1Mix, we report DNSMOS. We use the implementation bundled with the evaluation code, which loads the checkpoint and computes a P.808-style MOS estimate from Mel features. The final score is averaged over analysis windows. We do not use DNSMOS P.835. In our resynthesis evaluation, DNSMOS is used only for VoiceBank and Libri1Mix; LibriSpeech uses UTMOS.

dWER. We report dWER as an audio-only measure of linguistic preservation. Unless a reference transcript is explicitly provided, both the reference waveform and the hypothesis waveform are transcribed with `faster-whisper`. We use `WhisperModel` with the default model hub identifier `small`, corresponding to Whisper small, and the tokenizer `openai/whisper-small`. The ASR device and compute type are configurable; the default compute type is `int8`. Both ASR outputs are normalized with `WhisperTokenizer.normalize`, tokenized by whitespace, and compared using `SpeechBrain`'s `ErrorRateStats`. Thus, dWER is the WER between the two ASR transcripts, not the difference between two WER values.

Speaker Similarity. We report speaker similarity (Sim) using a pretrained speaker-verification model. We use the WavLM speaker-verification backend, `microsoft/wavlm-base-sv`.

Baseline and Model Checkpoints. For the original FocalCodec baselines, we use the public `lucadellalib/focalcodec` hub models. Unless otherwise stated, non-FocalCodec baseline numbers in the resynthesis tables follow the common FocalCodec evaluation protocol. For all the other baselines, refer to the `audiocodecs` repository.

D Token-Swap Protocol Details

Cumulative-Importance Slot Replacement as an Evaluation Protocol. Using cumulative-importance replacement turns slot importance into a testable intervention: if a factor is concentrated in a subset of slots, replacing the smallest set of top-ranked slots that accounts for a fraction γ of the total importance should move task-specific metrics while preserving non-target attributes. Varying γ provides a controllable edit strength. Small values of γ probe highly concentrated, high-confidence slots, while larger values test whether useful signal is distributed across a broader set of slots. In this sense, cumulative-importance replacement functions as a causal probe of representational structure rather than a recipe for optimal downstream quality.

Speaker Editing with Parallel VCTK References. For speaker editing, the reference is a different speaker reading the same sentence. This parallel condition reduces lexical confounds and makes Sim and dWER changes easier to attribute to speaker-token transfer instead of content mismatch. The setup is intentionally favourable for analyzing speaker transfer under controlled conditions; more realistic non-parallel conversion remains an important extension.

Denoising. For denoising, the target utterance is clean and selected by a deterministic cyclic shift over the clean evaluation subset; it is therefore typically unmatched in both transcript and speaker. This prevents trivial identity-copy behavior and avoids relying on paired clean targets. Any gains from slot replacement therefore indicate that the identified noise-associated slots can be manipulated robustly through token swapping.

Importance-Guided vs Control Slot Selections. Random and least-important slots provide the critical controls. These controls test whether gains come from targeted factor localization rather than generic perturbation. If importance-guided edits dominate these controls at matched γ, this supports the claim that slot-importance profiles identify functionally relevant positions.

No-Swap Source Reconstruction as a Reference. The no-swap setting reconstructs the source utterance without slot replacement. It provides a reconstruction baseline, helping distinguish the effect of swapping from artifacts already introduced by the tokenizer and decoder.

Asymmetric Encoder–Decoder Capacity. Our architecture uses a lighter encoder than decoder. This choice is motivated by generative workloads, where encoding is often performed offline or at large scale for preprocessing, augmentation, and conditioning extraction, while decoding quality directly controls perceptual fidelity. A stronger decoder better handles the one-to-many inverse mapping from compressed latents to waveform detail, whereas a leaner encoder reduces front-end compute. This asymmetry follows the same motivation as recent generative-first neural audio autoencoders, where encoder-side efficiency and decoder capacity are central to tokenization throughput and downstream usability (Casebeer et al., 2026).

E Overlap-Add Inference

Long utterances may exceed the fixed temporal context used to train LATTE. At inference time, we therefore process long recordings with feature-domain overlap-add (OLA). The waveform is first encoded into FocalCodec features, overlapping feature chunks are passed through LATTE, and the resulting reconstructed feature chunks are merged before a single vocoder pass.

Let $x[n]$ be a mono waveform sampled at $16\,\mathrm{kHz}$. The FocalCodec encoder maps it to a feature sequence $\mathbf{X} \in \mathbb{R}^{T\times D}$ at $50\,\mathrm{Hz}$, corresponding to one feature frame every $20\,\mathrm{ms}$. We use feature chunks matching the training chunk duration. If the training chunk contains c waveform samples, the corresponding feature length is

$$K = \mathrm{round}\left(\frac{c}{f_s}\cdot 50\right),$$

where $f_s = 16\,000\,\mathrm{Hz}$.

For long utterances, we slide this window over the feature sequence with a fixed overlap of $1\,\mathrm{s}$, i.e., 50 feature frames. Thus the hop size is

$$h = K - 50.$$

The feature sequence is right-padded when needed so that the final window fits the regular sliding-window grid. For each window i, we extract a feature chunk $\mathbf{X}^{(i)} \in \mathbb{R}^{K\times D}$, run it through LATTE quantization and decoding, and obtain a reconstructed feature chunk $\widetilde{\mathbf{X}}^{(i)}$.

Overlapping chunks are merged using a non-periodic Hann window $\mathbf{w} \in \mathbb{R}^{K}$:

$$\mathbf{\Omega} \leftarrow \mathbf{\Omega} + \widetilde{\mathbf{X}}^{(i)} \odot \mathbf{w}, \qquad \omega \leftarrow \omega + \mathbf{w},$$

where $\mathbf{w}$ is broadcast over the feature dimension and ω stores the accumulated window envelope. The stitched feature sequence is then

$$\widetilde{\mathbf{X}} = \frac{\mathbf{\Omega}}{\mathrm{clamp}(\omega, 10^{-8})}.$$

After fusion, we crop the sequence back to the original feature length T and decode the full stitched feature trajectory with the FocalCodec vocoder. The decoded waveform is finally cropped to the original waveform length before evaluation.

For utterances shorter than the training chunk, we right-pad the waveform, process it with a single LATTE forward pass, and crop the decoded audio back to the original duration. This avoids applying temporal fusion when only one chunk is needed.

F TiTok Training Hyperparameters

This appendix reports the hyperparameters used for the TiTok-style LATTE BASE and LATTE LARGE tokenizer variants. Unless otherwise stated, both variants use the same training, optimization, data, and runtime settings. The only architectural difference between LATTE BASE and LATTE LARGE is the width preset used for the encoder and decoder compressor stacks. All LATTE models used in this paper use a rate of 50 tokens per second.

F.1 Model Variants

Table 5 summarizes the architectural presets used for the two variants. We use the shorthand BL to denote a base encoder with a large decoder, and LATTE LARGE to denote a large encoder with an XL decoder.

Table 5: TiTok-style model variants used in our experiments. Width presets are reported as `hidden_dims` for the corresponding compressor or decompressor stack.

Variant	Encoder width preset	Decoder width preset
BL	`base`: $(1024, 512, 256)$	`large`: $(2048, 1024, 512)$
LATTE LARGE	`large`: $(2048, 1024, 512)$	`xl`: $(4096, 2048, 1024)$

All compressor and decompressor temporal factors are set to 1, so the TiTok compressor stack does not perform additional time-axis scaling.

F.2 Shared Model Hyperparameters

Table 6 reports the model hyperparameters shared by the LATTE BASE and LATTE LARGE variants.

Table 6: Shared model hyperparameters for the LATTE BASE and LATTE (LARGE) variants.

Hyperparameter	Value
Input feature dimension	1024
Quantizer	Binary spherical quantizer (BSQ)
Code dimension	13
Chunk duration	5 s
Maximum sequence length (Positional Embedding)	10,000

F.3 Shared Optimization Hyperparameters

Table 7 reports the optimization hyperparameters used for both variants.

Table 7: Shared optimization hyperparameters for LATTE training.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	5×10^{-4}
Adam betas	$(0.9, 0.98)$
Weight decay	0.01
Gradient clipping	Max norm 5.0
BSQ entropy auxiliary loss weight	0.1
BSQ inverse temperature	100.0
Diversity weight	1.0

The learning rate is scheduled with `ReduceLROnPlateau`. The scheduler is configured with `mode=min`, factor 0.9, patience 0, improvement threshold 0.0025, and minimum learning rate 10^{-6}.

Table 8: Data and dataloader settings used for LATTE training.

Setting	Value
Training splits	`train-clean-100`, `train-clean-360`, `train-other-500`
Validation split	`dev-clean`
Dataset	LibriTTS
Audio sampling rate	16 kHz
Chunk size	80,000 samples
Chunk duration	5 s
Number of workers	4
Batch size	4

F.4 Data and Dataloader Settings

Table 8 reports the data and dataloader settings.

The chunk size corresponds to $5\,\mathrm{s} \times 16\,000\,\mathrm{Hz}$.

F.5 Computational Resources

The model is trained on a node with four NVIDIA H100 GPUs, each with 80 GB of memory.