BEYOND FIXED FRAMES: DYNAMIC CHARACTER-ALIGNED SPEECH TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural audio codecs are at the core of modern conversational speech technologies, converting continuous speech into sequences of discrete tokens that can be processed by LLMs. However, existing codecs typically operate at fixed frame rates, allocating tokens uniformly in time and producing unnecessarily long sequences. In this work, we introduce **DyCAST**, a **Dy**namic **C**haracter-**A**ligned **S**peech **T**okenizer that enables variable-frame-rate tokenization through soft character-level alignment and explicit duration modeling. DyCAST learns to associate tokens with character-level linguistic units during training and supports alignment-free inference with direct control over token durations at decoding time. Experiments show that DyCAST achieves competitive speech resynthesis quality and downstream performance while using significantly fewer tokens than fixed-frame-rate codecs. **Demo samples are available at https://anonymous.4open.science/w/dycast-web/**.

1 INTRODUCTION

In the past few years, tokenization has emerged as a key building block in multimodal LLMs, enabling joint autoregressive modeling across modalities (Grattafiori et al., 2024; Jiang et al., 2024a; Comanici et al., 2025; Singh et al., 2025; DeepSeek-AI et al., 2025). By representing diverse modalities as sequences of discrete tokens, these models can operate over text, images, and audio within a shared modeling framework. For speech, tokenization is typically achieved through **neural audio codecs**, which map continuous waveforms to discrete tokens. Such representations have driven recent progress in generative speech and audio modeling (Borsos et al., 2023; Chen et al., 2025; Nguyen et al., 2025; Défossez et al., 2024; Labiausse et al., 2025; Zeghidour et al., 2025), enabling LLMs to operate directly on speech. Originally developed for efficient transmission (Zeghidour et al., 2021; Défossez et al., 2023), modern codecs now produce compact yet powerful representations that support high-quality resynthesis as well as competitive performance in both discriminative and generative downstream tasks (Mousavi et al., 2025; Guo et al., 2025).

Despite their success, most existing speech tokenizers rely on frame-level representations at a **fixed frame rate**, producing tokens at a constant temporal resolution. While effective for reconstruction, this design is poorly matched to the inherently variable temporal structure of speech: silence and steady regions are information-poor, whereas rapidly changing segments are information-dense (Van Kuyk et al., 2017; Dieleman et al., 2021; Cuervo et al., 2022). As a result, fixed-frame-rate tokenization leads to inefficient sequence lengths and limited alignment with corresponding text, making generative modeling more challenging. Although **dynamic-frame-rate** codecs and **text-supervised alignment** methods have recently emerged, this area is still developing. Existing approaches often rely on heuristic frame merging or clustering strategies that are weakly grounded in linguistic structure (Wang et al., 2025a; Zhang et al., 2025; Zheng et al., 2026; Li et al., 2025b), or require transcriptions or alignment information at inference time (Tseng et al., 2025; Hsu et al., 2025), limiting flexibility in fully speech-based scenarios.

In this work, we introduce **DyCAST**, a **Dy**namic **C**haracter-**A**ligned **S**peech **T**okenization framework that enables variable-frame-rate coding through **soft character-level alignment**. Each speech token is approximately associated with a character in the underlying transcript, with the degree of alignment controlled by a learned **boundary predictor**. At inference time, this alignment can be adjusted dynamically, allowing a flexible trade-off between strict character alignment and longer

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.

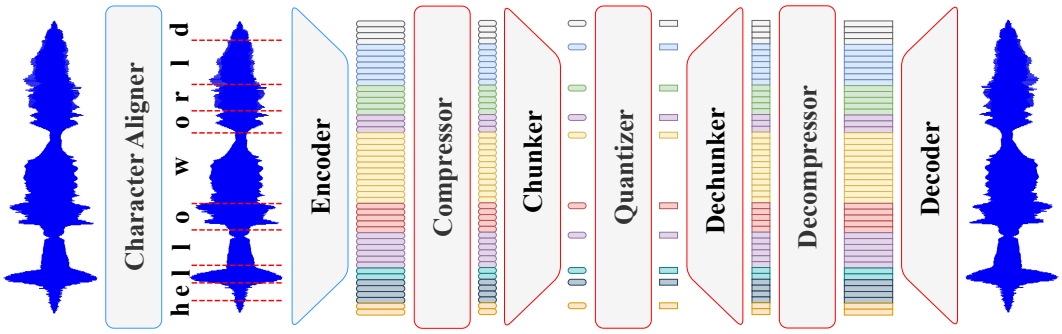

Figure 1: DyCAST architecture. Frame-level representations extracted by a frozen, self-supervised encoder are compressed and dynamically grouped into variable-length chunks, pooled, and quantized into discrete tokens. The decoding stage reverses this process by expanding token-level representations back to frame-level features before waveform reconstruction. Character-level boundaries are provided during training by a frozen aligner.

token spans. As a result, DyCAST can operate either with ground-truth alignments when available (e.g. for text-to-speech) or in a fully alignment-free mode based on the learned soft alignment alone. Additionally, DyCAST provides explicit control over token durations during decoding via a **duration predictor** that learns to invert the pooling operation, predicting appropriate token durations from the context.

In summary, our contributions are as follows:

- We propose **DyCAST**, a novel framework for **dynamic**, **controllable**, **variable-frame-rate** speech tokenization, where the temporal granularity and resulting frame rate can be adjusted at inference time by modulating a learned boundary predictor.
- We introduce **character-level alignment** within the tokenization process, allowing DyCAST to operate both in alignment-free settings and with ground-truth alignments, leading to improved semantic consistency and controllability for text-to-speech.
- We empirically show **competitive performance** on speech resynthesis and a range of other downstream tasks, while using significantly **fewer tokens** compared to fixed-frame-rate baselines.

Code and checkpoints will be released publicly to foster reproducibility and further research. **Demo samples are available at https://anonymous.4open.science/w/dycast-web/**.

2 Related Work

Speech Tokenization. The dominant approach to speech tokenization discretizes audio at a fixed temporal resolution using frame-level tokens. Early research on neural speech codecs primarily focused on high-fidelity acoustic reconstruction at medium bitrates (Zeghidour et al., 2021; Défossez et al., 2023; Kumar et al., 2023). In parallel, a separate line of work explored semantic tokenization by discretizing self-supervised phonetic speech representations to capture linguistic content (Baevski et al., 2020; Hsu et al., 2021; Messica & Adi, 2024; Mousavi et al., 2024b; Chang et al., 2025), at the expense of fine-grained acoustic detail. More recently, hybrid codecs have emerged with the goal of balancing semantic and acoustic information. These approaches combine multiple design strategies, including the use of multiple codebooks (Ju et al., 2024; Jiang et al., 2024b; Zheng et al., 2025), dual-encoder architectures (Liu et al., 2024), knowledge distillation (Zhang et al., 2024; Défossez et al., 2024; Yang et al., 2025; Gong et al., 2025; Li et al., 2025a), or supervised fine-tuning (Har-Tuv et al., 2025), to improve both reconstruction quality and speech language modeling performance. More recently, a growing trend has shifted toward single-codebook designs that jointly encode semantic and acoustic information. These approaches achieve good reconstruction quality at low bitrates while substantially simplifying downstream modeling (Ji et al., 2025; Bai et al., 2024; Xin et al., 2024; Wu et al., 2025; Ye et al., 2025; Della Libera et al., 2025; 2026; Song et al., 2025).

At the same time, recent works have begun exploring dynamic-frame-rate tokenization (Wang et al., 2025a; Zhang et al., 2025; Zheng et al., 2026; Li et al., 2025b). However, these methods typically rely on heuristic token merging, lack explicit semantic alignment, or require transmitting token durations for reconstruction. In contrast, our work introduces dynamic, character-aligned speech tokens whose durations adapt to linguistic content, providing more efficient and linguistically grounded representations without requiring durations as side information.

Text-Aligned Speech Representations. Recent work has explored text-aligned tokenization as a way to bridge the modality gap for joint speech-text modeling. Methods such as TASTE (Tseng et al., 2025) and TASLA (Hsu et al., 2025) learn discrete speech tokens aligned with text by enforcing transcript-speech alignment through cross-attention-based architectures. Similarly, TaDiCodec (Wang et al., 2025b) incorporates textual information in a text-aware diffusion-based speech codec, where speech-text alignment is introduced implicitly through autoregressive conditioning on text rather than being enforced explicitly at the tokenization stage. SSR (Tan et al., 2025) and LST (Lu et al., 2025) pursue a similar goal of speech-text alignment, but are tangential to our work, as they operate at the speech language modeling level and do not define a standalone speech tokenization framework. While effective for joint modeling, these methods typically rely on ground-truth alignments or require textual input at inference time, limiting their applicability to text-free generative settings and making them closer to TTS-oriented pipelines than general-purpose speech tokenizers. In contrast, DyCAST is a fully non-autoregressive, autoencoder-based speech tokenizer that operates end-to-end without requiring text at inference time.

3 DyCAST

The proposed DyCAST framework (see Figure 1) is inspired by the modular compressor-quantizer-decompressor architecture introduced in (Della Libera et al., 2025), while extending it with dedicated modules for *dynamic pooling*. Given an input waveform, a frozen, pretrained self-supervised speech **encoder** first extracts high-dimensional, single-stream acoustic-semantic representations at a fixed frame rate. These features are then projected into a lower-dimensional latent space by a lightweight **compressor**, retaining the most informative components of the representation. A **chunker** module subsequently groups consecutive frames into variable-length chunks, yielding a temporally adaptive chunking of the input. Within each chunk, frame-level features are pooled to form compact chunk-level representations, which are then discretized by a **quantizer**, producing a sequence of discrete tokens. A **dechunker** module reverses the chunking operation by expanding each token-level representation back to frame-level features. A **decompressor** maps the expanded features back to the original encoder dimensionality, and a **decoder** finally reconstructs the time-domain waveform.

3.1 Dynamic Chunking

The *chunker* module consists of two components: a **boundary predictor**, which identifies semantically meaningful chunk boundaries, and a **downsampler**, which aggregates frame-level representations within each chunk into a compact chunk-level representation. To provide training supervision for chunk boundaries, we employ a frozen, pretrained **character aligner**. This aligner takes a waveform as input and returns character-level durations, and can be conveniently implemented using a CTC-based (Graves et al., 2006) ASR model with characters as the vocabulary. The resulting frame-level character boundaries are used to derive target chunk durations that serve as supervision during training. To model the boundary distribution, we adopt a **discrete-time hazard model** (Singer & Willett, 1993; Ren et al., 2019) rather than a standard frame-wise binary cross-entropy objective. Unlike independent boundary classification, the hazard formulation explicitly models the *time to the next boundary*, allowing boundary predictions to be temporally dependent and properly normalized over time. This is particularly important in our setting, where chunk durations are the primary quantity of interest and boundaries are sparse relative to the frame rate. Moreover, the hazard model naturally enforces a single boundary per chunk and provides a principled likelihood over variable-length segments, which facilitates stable training and coherent boundary decoding. Formally, given frame-level representations $x_{1:T}$ extracted by the compressor, the boundary predictor estimates a boundary probability $h_t \in (0, 1)$ at each frame:

$$h_t = \sigma(f_\theta(x_{1:T})_t)\,, \tag{1}$$

where f_θ is neural network operating on the sequence of features. The probability that the next boundary occurs k frames after time t is then given by

$$P(T = k \mid t) = \left(\prod_{i=0}^{k-1} (1 - h_{t+i}) \right) h_{t+k}. \tag{2}$$

The hazard model is trained by maximizing the likelihood of ground-truth next-boundary offsets extracted from forced alignment.

At inference time, chunk boundaries are decoded directly from the hazard predictions without any textual input. Boundaries can be obtained either greedily, yielding deterministic chunking, or via sequential sampling to introduce variability. In both cases, decoding constraints on the minimum and maximum chunk duration, controlled by the hyperparameters `min_gap` and `max_gap`, respectively, are used to regulate the resulting frame rate. In addition, a hazard threshold τ_h controls boundary emission: higher values of τ_h favor longer chunks (lower frame rates), while lower values produce finer-grained chunks (higher frame rates). Formally, a boundary is emitted at frame t when $h_t \geq \tau_h$ subject to the `min_gap` constraint; if no boundary is emitted for `max_gap` consecutive frames, a boundary is forced. Given the decoded boundaries, the downsampler produces chunk-level representations by selecting the *last frame* of each chunk. While alternative strategies such as average pooling per chunk are possible, this design keeps the operation simple and efficient, and preserves the original compressed representations without blending them across frames, which facilitates adaptation to different frame rates.

3.2 DECHUNKING

Like the chunker, the *dechunker* module consists of two components: a **duration predictor**, which estimates the number of frames associated with each character-aligned discrete token, and an **upsampler**, which expands each chunk-level representation back to frame-level features. The duration predictor is necessary to recover temporal structure at decoding time, since in general only the token sequence is transmitted, not the chunk boundaries used to derive it. Supervision for duration prediction comes from the same frozen character aligner used during chunking, which provides target token durations corresponding to character spans. To model the distribution of token durations, we adopt a **negative binomial duration model** (Zen et al., 2009). Compared to alternatives such as geometric or Poisson distributions, the negative binomial provides greater flexibility by explicitly modeling over-dispersed count data, which is characteristic of speech durations. In particular, while a geometric model assumes a memoryless process with a fixed hazard rate and a Poisson model ties the mean and variance, the negative binomial decouples these quantities, allowing the model to capture the heavy-tailed and highly variable nature of character-level durations in speech. Formally, given a sequence of discrete tokens $c_{1:N}$, the duration model maps each token to a positive *free* mean duration $\mu_i^{\text{free}} > 0$,

$$\mu_i^{\text{free}} = \text{softplus}(g_\phi(c_{1:N})_i)\,, \tag{3}$$

where g_ϕ denotes a neural network operating on the full sequence of tokens. The actual mean token duration is obtained by enforcing a minimum duration $d_{\min}$,

$$\mu_i = d_{\min} + \mu_i^{\text{free}}. \tag{4}$$

By default, $d_{\min} = 1$. The excess target duration

$$y_i = d_i - d_{\min} \geq 0 \tag{5}$$

is modeled using the negative binomial distribution with mean μ_i^{free} and a global dispersion parameter $\alpha > 0$ shared across all tokens,

$$y_i \sim \text{NB}(\mu_i^{\text{free}}, \alpha). \tag{6}$$

The duration model is trained by minimizing the negative log-likelihood together with a **normalized length regularization** term,

$$\mathcal{L}_{\text{dur}} = \sum_{i=1}^{N} -\log p_{\text{NB}}(y_i \mid \mu_i^{\text{free}}, \alpha) \; + \; \lambda \left(\frac{\sum_{i=1}^{N} \mu_i^{\text{free}} - T_{\text{free}}}{T_{\text{free}} + \epsilon} \right)^2, \tag{7}$$

where $T_{\text{free}} = T - N d_{\min}$ denotes the total number of allocatable frames after enforcing a minimum duration $d_{\min}$ per token, λ controls the strength of the length regularization, and ϵ is a small constant for numerical stability. This normalized penalty encourages globally consistent pacing while remaining invariant to utterance length.

At inference time, the duration model supports two decoding regimes. In the **free decoding** mode, when the total number of frames is unknown, durations are obtained greedily as

$$\hat{d}_i = d_{\min} + \text{round}(\mu_i^{\text{free}}), \tag{8}$$

yielding a predicted total length $\hat{T} = \sum_i \hat{d}_i$. In the **budget-constrained decoding** mode, when the target length T is known (e.g. speech resynthesis), the predicted free means are renormalized to match the available duration budget,

$$\tilde{\mu}_i^{\text{free}} = \mu_i^{\text{free}} \frac{T_{\text{free}}}{\sum_j \mu_j^{\text{free}}}, \tag{9}$$

followed by deterministic integer rounding that enforces $\sum_i \hat{d}_i = T$ exactly. This hybrid formulation enables explicit duration control at decoding time while remaining independent of character alignment during inference.

4 Experimental Setup

4.1 Architecture

We now describe the DyCAST architecture used in our experiments, inspired by FocalCodec (Della Libera et al., 2025). Full hyperparameter specifications are reported in Appendix C.

Character Aligner. We employ MMS (Pratap et al., 2024). Alignment information is extracted via simple `argmax` CTC alignment on the frame-level log probabilities obtained by running the model on the input waveform (see Appendix F.1 for a detailed analysis of alignment quality). To handle silence segments detected by the aligner, we do not discard them; instead, we aggregate them into the subsequent non-silence token. We find this strategy to work well in practice and leave more refined treatments of silence for future work.

Encoder and Decoder. We use the pretrained WavLM-large[1] model (Chen et al., 2022) as the encoder. In particular, we use the representations from the **6-th transformer layer**, which are semantically rich while retaining fine-grained acoustic information (Della Libera et al., 2025; Baas et al., 2023). Waveform reconstruction from these features is performed using Vocos (Siuzdak, 2024) for efficient decoding.

Compressor and Decompressor. Both modules are implemented using focal modulation blocks (Yang et al., 2022; Della Libera et al., 2024), leveraging their demonstrated effectiveness and efficiency for low-bitrate speech coding (Della Libera et al., 2025).

Boundary and Duration Predictor. We adopt the same efficient focal modulation architecture also for the boundary and duration predictor. The boundary predictor uses a binary classification head, while the duration model predicts a free mean parameter together with a shared, learnable dispersion parameter α, as described in Section 3.2. Importantly, the boundary predictor is trained directly on the 1024-dimensional WavLM features rather than the compressed latents, providing richer context for boundary estimation. In contrast, the duration predictor operates on the pooled, quantized low-dimensional latents.

Quantizer. Della Libera et al. (2025) employ binary spherical quantization (BSQ) (Zhao et al., 2025), which tightly couples bitrate to latent dimensionality L through an implicit codebook of size $|\mathcal{C}| = 2^L$. At the low frame rates considered in this work, this coupling limits representational expressivity. We therefore generalize this formulation to **scalar spherical quantization** (SSQ). SSQ

[1] https://github.com/microsoft/unilm/tree/master/wavlm

preserves the spherical constraint of BSQ while allowing each latent dimension to take one of K scalar levels in $[-\frac{1}{\sqrt{L}}, \frac{1}{\sqrt{L}}]$, enabling more flexible bitrates. Compared to finite scalar quantization (FSQ) (Mentzer et al., 2024), SSQ explicitly enforces spherical geometry via L_2 normalization and includes a factorized entropy regularizer for improved codebook utilization. The resulting implicit codebook has size $|\mathcal{C}| = K^L$. In our experiments, we set $L = 32$ and $K = 4$, yielding $|\mathcal{C}| = 4^{32}$. To make this high cardinality tractable in practice, we adopt a factorized representation in which each token is decomposed into 32 parallel streams, each with a vocabulary size of 4.

4.2 Training

We train DyCAST on LibriTTS (Zen et al., 2019), resampled to 16 kHz. We adopt a multi-stage training strategy for improved stability:

1. **Reconstruction.** We train the compressor-quantization-decompressor for WavLM features reconstruction using L_2 loss, *without* boundary prediction, or duration modeling. Dynamic downsampling and upsampling use the character durations from the character aligner in a teacher-forced manner. In parallel, we train the decoder to map WavLM features back to waveform.
2. **Boundary Predictor Training.** We train the boundary predictor to map WavLM features to character boundaries, using character aligner durations as supervision.
3. **Adaptation to Predicted Boundaries.** We fine-tune the compressor-quantizer-decompressor to operate on durations predicted by the boundary predictor. To improve robustness, during training we randomly sample among character aligner boundaries and predicted boundaries greedily decoded with $\tau_h = 0.5$, `min_gap` $\in \{1, 3, 5\}$, and no `max_gap` constraint (see Section 3.1).
4. **Duration Predictor Training.** Finally, we freeze all the other components and train the duration model to predict chunk durations from pooled quantized latents, using the same boundary sampling strategy as in the previous step. This procedure yields a robust system that generalizes well to different boundary configurations, including `min_gap` values outside the training range.

Hyperparameters and training details are provided in Appendix C.

4.3 Inference

DyCAST provides substantial flexibility at inference time for both encoding and decoding. For encoding, chunk boundaries can be obtained in two ways:

1. **Character Aligner.** We use boundaries provided by the character aligner. This is particularly useful for downstream applications that benefit from accurate speech-text alignment, such as text-to-speech.
2. **Boundary Predictor.** We greedily decode or sample boundaries from the boundary predictor. By adjusting the decoding hyperparameters, we obtain direct control over the effective frame rate. For example, increasing the `min_gap` parameter enforces a larger minimum gap between boundaries, resulting in fewer, more uniformly spaced tokens.

For decoding, three operating modes are available:

1. **Tokens + Durations.** We transmit discrete tokens together with their original durations and use them as side information during decoding. The effective bitrate is the token bitrate plus the overhead required to transmit one duration per token, assuming a bounded duration range.
2. **Tokens + Utterance Length.** We transmit discrete tokens and a single global utterance length, which incurs negligible overhead. The duration model is then used in budget-constrained decoding mode to sample durations whose total length matches the provided value. This mode is particularly suitable for resynthesis tasks.
3. **Tokens Only.** We transmit only discrete tokens and let the duration model infer the most likely durations. This mode is useful for modeling tasks such as text-to-speech and speech language modeling, where tokens do not come with explicit duration information.

Note that, independently of the chosen setting, durations can always be resampled using the duration model or an external model to introduce natural variability in speaking rate, or manually overridden to stretch or compress individual characters. Overall, this design offers significantly greater flexibility than traditional fixed-rate codecs, enabling fine-grained control over bitrate, timing, and prosody.

5 DOWNSTREAM EVALUATION

For downstream evaluation, we consider four inference configurations: **DyCAST-CA**, **DyCAST-BP1**, **DyCAST-BP3**, and **DyCAST-BP5**. DyCAST-CA uses boundaries provided by the character aligner, while the remaining variants rely on the boundary predictor with `min_gap` set to 1, 3, and 5, respectively, corresponding to average frame rates of approximately 14 Hz, 9 Hz, and 6 Hz. Unless otherwise stated, all experiments use the *tokens + utterance length* decoding mode. We provide analysis of alternative decoding strategies and visualizations in Appendix F.2 and F.3.

Table 1: Codecs considered in our experiments.

Codec	Frame Rate (Hz)	Bitrate (kbps)	Sample Rate (kHz)	Codebooks	Params (M)
EnCodec	75.0	1.50	24	2×1024	15
DAC	50.0	1.00	16	2×1024	74
WavLM6-KM	50.0	0.45	16	1×512	127
SpeechTokenizer	50.0	1.00	16	2×1024	108
SemantiCodec	25.0	0.65	16	2×8192	1033
Mimi	12.5	0.69	24	5×2048	82
WavTokenizer	40.0	0.48	24	1×4096	85
BigCodec	80.0	1.04	16	1×8192	160
Stable Codec	25.0	0.70	16	2×15625	950
FocalCodec	50.0	0.65	16	1×8192	142
DyCAST-CA	$\sim$14.4	$\sim$0.92	16	1×4^{32}	345
DyCAST-BP1	$\sim$17.5	$\sim$1.12	16	1×4^{32}	345
DyCAST-BP3	$\sim$9.0	$\sim$0.57	16	1×4^{32}	345
DyCAST-BP5	$\sim$6.2	$\sim$0.40	16	1×4^{32}	345

To evaluate the effectiveness of our framework, we closely follow the experimental setup of Della Libera et al. (2025) and compare against recent state-of-the-art low-bitrate codecs operating at fixed frame rates. As our focus is on low-bitrate regimes, when multiple quantizers are available, we configure them to achieve a bitrate below 1.50 kbps, ensuring a fair comparison. In particular, we include **EnCodec** (Défossez et al., 2023), **DAC** (Kumar et al., 2023)[2], **WavLM6-KM** (Wang et al., 2024), **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** (Della Libera et al., 2025). The configurations and details of each model are summarized in Table 1. Additional details about these baselines are provided in Appendix B. We follow the evaluation protocol of Della Libera et al. (2025), considering speech resynthesis under different conditions, voice conversion, and both discriminative and generative downstream tasks. For generative evaluation, we focus on text-to-speech in limited-data regimes.

Details about the datasets and experimental settings can be found in Appendix A and D. We further evaluate voice conversion (VC) on **VCTK** (Yamagishi et al., 2017), following the protocol of Della Libera et al. (2025), which is described in detail in Appendix D.2.

5.1 SPEECH RESYNTHESIS AND VOICE CONVERSION

We evaluate the speech resynthesis (SR) capabilities of DyCAST on **LibriSpeech** (Panayotov et al., 2015), **MLS** (Pratap et al., 2020), **VoiceBank** (Valentini-Botinhao et al., 2016), and **Libri1Mix**. Naturalness, intelligibility, and speaker similarity are assessed using **UTMOS** (Saeki et al., 2022) for clean speech and **DNSMOS** (Reddy et al., 2022) for noisy speech, **dWER** (Wang et al., 2021), and **Sim** (Della Libera et al., 2025), respectively. Additional details are provided in Appendix D.1.

As shown in Table 2, DyCAST achieves a strong balance between reconstruction quality and token efficiency, operating at substantially lower frame rates than fixed-frame-rate baselines while maintaining competitive performance across all metrics. On LibriSpeech, DyCAST variants consistently reach high naturalness scores and low intelligibility degradation, closely matching strong fixed-rate codecs such as FocalCodec and Stable Codec, despite using 3–8x fewer frames. Notably, DyCAST-CA and DyCAST-BP1 achieve dWER values comparable to high-rate baselines, confirming that character-aligned or predicted durations preserve linguistic content even at reduced token rates. For multilingual speech, DyCAST maintains robust performance across languages, with only a moderate increase in dWER as the frame rate is reduced. Importantly, speaker similarity remains consistently high across all DyCAST variants, indicating that variable-rate tokenization does not compromise speaker identity, even in low-rate regimes. Notably, aside from the character aligner, which naturally supports multilingual inputs, the codec itself is trained exclusively on English speech, highlighting the strong generalization capability of the learned discrete representations. In noisy conditions, DyCAST remains competitive with dedicated speech codecs. While very aggressive compression leads

[2]We use the 16 kHz checkpoint, whereas Kumar et al. (2023) report results at 24 kHz.

Table 2: Speech resynthesis (SR) and voice conversion (VC). **Best** and second-best results are highlighted.

Codec	Frame Rate (Hz) ↓	Bitrate (kbps) ↓	SR – LibriSpeech			SR – MLS			SR – VoiceBank			SR – Libri1Mix			VC – VCTK		
			UTMOS ↑	dWER ↓	Sim ↑	UTMOS ↑	dWER ↓	Sim ↑	DNSMOS ↑	dWER ↓	Sim ↑	DNSMOS ↑	dWER ↓	Sim ↑	UTMOS ↑	dWER ↓	Sim ↑
Reference	—	—	4.09	0.00	100.0	2.84	0.00	100.0	3.56	0.00	100.0	3.73	0.00	100.0	4.09	0.00	100.0
EnCodec	75.0	1.50	1.58	8.08	93.8	1.33	29.60	95.5	2.76	28.16	87.7	2.40	55.17	86.3	1.24	86.52	72.2
DAC	50.0	1.00	1.29	20.04	89.2	1.24	56.08	89.1	2.72	63.90	79.8	2.40	90.92	76.6	1.25	104.00	67.2
WavLM6-KM	50.0	0.45	3.75	6.20	90.0	2.97	44.54	89.5	3.06	20.67	82.9	2.87	36.60	85.9	2.90	26.68	92.4
SpeechTokenizer	50.0	1.00	2.28	5.14	91.6	1.55	56.32	92.0	2.74	34.51	82.2	2.58	57.26	82.8	1.49	**20.32**	81.2
SemantiCodec	25.0	0.65	2.91	8.97	96.0	1.87	36.21	97.7	3.13	31.46	90.6	2.67	51.18	89.9	2.02	106.00	72.8
Mimi	12.5	0.69	3.29	5.73	96.0	2.08	30.96	96.7	3.01	28.00	87.8	2.65	49.14	89.4	2.40	110.00	89.7
WavTokenizer	40.0	0.48	3.78	11.55	95.4	2.64	49.73	97.0	3.09	42.12	89.8	2.53	70.10	86.3	3.13	43.15	73.4
BigCodec	80.0	1.04	4.11	2.55	**98.5**	2.86	15.24	**99.1**	3.19	20.67	92.3	2.75	53.26	88.3	1.31	99.96	68.9
Stable Codec	25.0	0.70	**4.32**	4.97	94.7	**3.47**	56.99	95.9	3.33	20.32	88.8	2.91	43.52	90.0	**3.76**	27.63	71.1
FocalCodec	50.0	0.65	4.05	**2.18**	97.4	2.96	**12.57**	98.3	3.16	**8.08**	91.3	2.93	**27.89**	91.6	3.38	21.27	92.2
DyCAST-CA	∼14.4	∼0.92	4.06	3.22	97.5	2.92	16.63	98.4	**3.36**	14.98	91.8	**3.01**	30.54	91.9	3.24	26.52	91.1
DyCAST-BP1	∼17.5	∼1.12	3.99	3.63	97.7	2.96	16.50	98.6	3.29	13.63	**92.4**	3.00	31.06	**92.1**	3.31	25.07	**92.3**
DyCAST-BP3	∼9.0	∼0.57	4.01	4.60	97.4	2.99	21.29	98.4	3.31	17.93	91.9	3.00	36.51	91.4	3.29	27.90	92.2
DyCAST-BP5	∼**6.2**	∼**0.40**	4.03	8.50	96.6	3.00	35.03	97.8	3.31	27.21	90.3	2.98	46.29	90.0	3.26	36.03	92.0

to higher dWER, naturalness and speaker similarity remain stable, suggesting that explicit duration modeling provides robustness to noise by avoiding over-fragmentation of tokens.

For voice conversion, DyCAST achieves strong speaker similarity and intelligibility, comparable to FocalCodec. Performance degrades gracefully as the frame rate is reduced, indicating that the learned discrete units retain speaker-relevant information even at low temporal resolution.

5.2 Discriminative Tasks

We evaluate the representational quality of DyCAST discrete tokens on automatic speech recognition (ASR), speaker identification (SI), and speech emotion recognition (SER) probing tasks (Mousavi et al., 2024a). For ASR and SI, we use **LibriSpeech**, while for SER we use **IEMOCAP** (Busso et al., 2008). All experiments follow the same shallow probing setup as in (Della Libera et al., 2025). We report word error rate (**WER**) for ASR and error rate (**ER**) for SI and SER. Additional details are provided in Appendix D.3–D.5.

Results for discriminative probing tasks are reported in Table 3. Overall, DyCAST performs well despite operating at substantially lower frame rates than fixed-rate baselines. On ASR, DyCAST-CA achieves the best WER among all codecs, outperforming both fixed-frame-rate speech codecs and prior discrete tokenizers, while using significantly fewer tokens. This highlights the benefit of character-aligned tokenization for preserving fine-grained linguistic information that is directly relevant for recognition. As the frame rate is reduced further, performance degrades gracefully. Variants relying on predicted boundaries maintain competitive WER at moderate rates, while extremely low-frame-rate configurations exhibit higher WER, reflecting the expected trade-off between compression and phonetic resolution. Importantly, this degradation is consistent with trends observed in fixed-rate codecs at comparable or higher bitrates. For SI, DyCAST achieves error rates comparable to strong baselines, indicating that speaker-related cues are well preserved even under aggressive temporal compression. Although fixed-frame-rate tokenizers optimized for speaker modeling (e.g. WavTokenizer) achieve lower SI error rates, DyCAST offers a more favorable balance between speaker information retention and token efficiency. On SER, performance across all codecs remains relatively close, suggesting that coarse prosodic and affective cues are robust to different tokenization strategies. DyCAST performs on par with fixed-rate baselines, indicating that variable-rate tokenization does not impair the extraction of emotional content, despite operating at lower frame rates. Overall, these results indicate that DyCAST tokens retain rich linguistic, speaker, and paralinguistic information even at substantially reduced rates. In particular, the strong ASR performance of the character-aligned variant suggests that explicitly modeling durations and linguistic alignment yields discrete tokens that are well suited for speech understanding tasks.

5.3 Text-To-Speech

We evaluate the generative capabilities of DyCAST on text-to-speech (TTS) using **LibriSpeech**, with character-level text as input and speaker embeddings for speaker conditioning. Performance is assessed using the same metrics as for speech resynthesis, namely **UTMOS**, **dWER**, and **Sim**. Following (Della Libera et al., 2025), we employ a transformer-based (Vaswani et al., 2017) autoregressive architecture over speech tokens for TTS, reflecting the formulation of TTS as text-

Table 3: Downstream modeling performance. Shallow probing heads are used for discriminative tasks (ASR – LibriSpeech, SI – LibriSpeech, SER – IEMOCAP), transformer-based models are used for generative tasks (TTS – LibriSpeech). **Best** and second-best results are highlighted.

Codec	Frame Rate (Hz) ↓	Bitrate (kbps) ↓	ASR	SI	SER	TTS		
			WER ↓	ER ↓	ER ↓	UTMOS ↑	dWER ↓	Sim ↑
EnCodec	75.0	1.50	27.89	3.00	47.00	1.71	64.28	83.2
DAC	50.0	1.00	35.89	3.27	45.90	1.34	47.06	85.9
WavLM6-KM	50.0	0.45	19.04	22.30	42.90	3.74	38.67	88.7
SpeechTokenizer	50.0	1.00	14.97	2.73	**41.50**	2.69	35.46	89.2
SemantiCodec	25.0	0.65	41.42	15.90	51.60	2.82	48.38	91.4
Mimi	12.5	0.69	22.98	5.43	44.70	3.11	28.63	93.6
WavTokenizer	40.0	0.48	35.62	2.44	49.80	3.68	47.56	92.8
BigCodec	80.0	1.04	26.41	**2.34**	47.50	3.43	54.43	89.4
Stable Codec	25.0	0.70	16.85	16.50	46.54	3.19	49.28	88.8
FocalCodec	50.0	0.65	17.63	4.48	45.60	4.11	28.10	93.3
DyCAST-CA	∼14.4	∼0.92	**13.05**	9.03	49.54	**4.23**†	**3.37**†	**96.2**†
DyCAST-BP1	∼17.5	∼1.12	19.98	7.34	51.61	3.91	11.57	93.2
DyCAST-BP3	∼9.0	∼0.57	27.80	7.42	50.00	4.00	14.56	92.6
DyCAST-BP5	**∼6.2**	**∼0.40**	40.27	11.50	50.23	4.02	15.85	92.8

† Non-autoregressive one-to-one architecture, defined only for DyCAST-CA.

conditioned speech language modeling. The only exception is the DyCAST-CA variant: thanks to the *hard* character-level alignment provided by the tokenizer, characters and speech tokens are in one-to-one correspondence. This enables the use of a **non-autoregressive TTS** model that directly maps each input character to its corresponding speech token. Details on the model architecture, hyperparameters, and training procedure are provided in Appendix D.6.

As shown in Table 3, DyCAST achieves strong performance across naturalness, intelligibility, and speaker similarity while operating at substantially reduced frame rates. Since the TTS model is autoregressive, lower frame rates directly translate into shorter sequences, yielding a more favorable learning regime by reducing exposure bias and easing long-range dependency modeling. Notably, DyCAST-CA clearly stands out, achieving by far the best TTS performance across all metrics. This result is enabled by the non-autoregressive one-to-one architecture uniquely supported by DyCAST-CA, which eliminates sequential token prediction altogether and is particularly effective in this limited-data regime. In addition, this architecture enables extremely fast inference, as generation is performed in a single parallel pass and does not require sampling or rescoring multiple hypotheses. Overall, these results highlight the dual benefit of DyCAST: efficient, controllable, low-frame-rate tokenization that simplifies autoregressive generation, and character-aligned representations that enable state-of-the-art quality and high inference efficiency in data-limited TTS scenarios.

6 Conclusions

We introduced DyCAST, a speech codec that enables variable-frame-rate tokenization through soft character-level alignment and explicit duration modeling. DyCAST produces substantially shorter token sequences than fixed-frame-rate codecs while maintaining competitive speech resynthesis quality and downstream performance, resulting in improved efficiency and more tractable transformer-based sequence modeling. A key advantage of DyCAST is its flexibility, enabling content-adaptive frame rates and explicit control over duration, bitrate, and reconstruction quality. DyCAST offers a promising foundation for next-generation speech tokenization.

AI use statement

In this work, we used generative AI tools to assist with polishing the writing and drafting portions of the manuscript. We did not use generative AI tools to generate research ideas, experimental results, or scientific claims. All AI-assisted text was carefully reviewed, edited, and verified by the authors for accuracy, originality, and consistency with the underlying research. The authors take full responsibility for the final content of this work, including any text or other material produced with the aid of generative AI.

Reproducibility statement

To facilitate reproducibility, we provide comprehensive implementation and experimental details in the Appendix, including model architectures, training configurations, data processing, and evaluation protocols.

References

Matthew Baas, Benjamin van Niekerk, and Herman Kamper. Voice conversion with just nearest neighbors. In *Interspeech*, pp. 2053–2057, 2023.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 12449–12460, 2020.

He Bai, Tatiana Likhomanenko, Ruixiang Zhang, Zijin Gu, Zakaria Aldeneh, and Navdeep Jaitly. dMel: Speech tokenization made simple. *arXiv preprint arXiv:2407.15835*, 2024.

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 (TASLP)*, 31:2523–2533, 2023.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, 2008.

Edresson Casanova, Paarth Neekhara, Ryan Langman, Shehzeen Hussain, Subhankar Ghosh, Xuesong Yang, Ante Jukic, Jason Li, and Boris Ginsburg. NanoCodec: Towards high-quality ultra fast speech LLM inference. In *Interspeech*, pp. 5028–5032, 2025.

Heng-Jui Chang, Hongyu Gong, Changhan Wang, James Glass, and Yu-An Chung. DC-Spin: A speaker-invariant speech tokenizer for spoken language models. In *Interspeech*, pp. 5723–5727, 2025.

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.

Sanyuan Chen, Chengyi Wang, 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. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 33:705–718, 2025.

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.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Defossez. Simple and controllable music generation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 47704–47720, 2023.

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.

Santiago Cuervo, Adrian Łańcucki, Ricard Marxer, Paweł Rychlikowski, and Jan Chorowski. Variable-rate hierarchical CPC leads to acoustic unit discovery in speech. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 34995–35006, 2022.

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.

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

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.

John Garofolo, L Lamel, W Fisher, Jonathan Fiscus, D Pallett, and Nancy Dahlgren. Darpa timit acoustic-phonetic continuous speech corpus cd-rom TIMIT, 1993.

Yitian Gong, Luozhijie Jin, Ruifan Deng, Dong Zhang, Xin Zhang, Qinyuan Cheng, Zhaoye Fei, Shimin Li, and Xipeng Qiu. XY-Tokenizer: Mitigating the semantic-acoustic conflict in low-bitrate speech codecs. *arXiv preprint arXiv:2506.23325*, 2025.

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.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pp. 369–376, 2006.

Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. Recent advances in discrete speech tokens: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.

Nadav Har-Tuv, Or Tal, and Yossi Adi. PAST: Phonetic-acoustic speech tokenizer. In *Interspeech*, pp. 3509–3513, 2025.

Ming-Hao Hsu, Liang-Hsuan Tseng, Hung yi Lee, and Zhizheng Wu. TASLA: Text-aligned speech tokens with multiple layer-aggregation. *arXiv preprint arXiv:2510.14934*, 2025.

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 Transactions on Audio, Speech, and Language Processing*, 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*, 2024a.

Xue Jiang, Xiulian Peng, Yuan Zhang, and Yan Lu. Universal speech token learning via low-bitrate neural codec and pretrained representations. *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–13, 2024b.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiang-Yang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *International Conference on Machine Learning (ICML)*, 2024.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

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.

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. In *Interspeech 2025*, pp. 4883–4887, 2025a.

Jiaqi Li, Yao Qian, Yuxuan Hu, Leying Zhang, Xiaofei Wang, Heng Lu, Manthan Thakker, Jinyu Li, Sheng Zhao, and Zhizheng Wu. FlexiCodec: A dynamic neural audio codec for low frame rates. *arXiv preprint arXiv:2510.00981*, 2025b.

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*, 18(8):1448–1461, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Yen-Ju Lu, Yashesh Gaur, Wei Zhou, Benjamin Muller, Jesus Villalba, Najim Dehak, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Srinivasan Iyer, and Duc Le. Latent speech-text transformer. *arXiv preprint arXiv:2510.06195*, 2025.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations (ICLR)*, 2024.

Shoval Messica and Yossi Adi. NAST: Noise aware speech tokenization for speech language models. In *Interspeech*, pp. 4169–4173, 2024.

Pooneh Mousavi, Luca Della Libera, Jarod Duret, Artem Ploujnikov, Cem Subakan, and Mirco Ravanelli. DASB - discrete audio and speech benchmark. *arXiv preprint arXiv:2406.14294*, 2024a.

Pooneh Mousavi, Jarod Duret, Salah Zaiem, Luca Della Libera, Artem Ploujnikov, Cem Subakan, and Mirco Ravanelli. How should we extract discrete audio tokens from self-supervised models? In *Interspeech*, pp. 2554–2558, 2024b.

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 (TMLR)*, 2025.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics (TACL)*, 13:30–52, 2025.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: an imperative style, high-performance deep learning library. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert. MLS: A large-scale multilingual dataset for speech research. In *Interspeech*, pp. 2757–2761, 2020.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research (JMLR)*, 25, 2024.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 28492–28518, 2023.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain de Langen, Cem Subakan, Peter Plantinga, Yingzhi Wang, Pooneh Mousavi, Luca Della Libera, Artem Ploujnikov, Francesco Paissan, Davide Borra, Salah Zaiem, Zeyu Zhao, Shucong Zhang, Georgios Karakasidis, Sung-Lin Yeh, Pierre Champion, Aku Rouhe, Rudolf Braun, Florian Mai, Juan Zuluaga-Gomez, Seyed Mahed Mousavi, Andreas Nautsch, Ha Nguyen, Xuechen Liu, Sangeet Sagar, Jarod Duret, Salima Mdhaffar, Gaëlle Laperrière, Mickael Rouvier, Renato De Mori, and Yannick Estève. Open-source conversational AI with SpeechBrain 1.0. *Journal of Machine Learning Research (JMLR)*, 25(333):1–11, 2024.

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.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *AAAI Conference on Artificial Intelligence*, 2019.

A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 749–752, 2001.

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.

Judith D. Singer and John B. Willett. It's about time: Using discrete-time survival analysis to study duration and the timing of events. *Journal of Educational Statistics*, 18(2):155–195, 1993.

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.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Yakun Song, Jiawei Chen, Xiaobin Zhuang, Chenpeng Du, Ziyang Ma, Jian Wu, Jian Cong, Dongya Jia, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. MagiCodec: Simple masked gaussian-injected codec for high-fidelity reconstruction and generation. *arXiv preprint arXiv:2506.00385*, 2025.

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, pp. 2125–2136, 2011.

Weiting Tan, Hirofumi Inaguma, Ning Dong, Paden D. Tomasello, and Xutai Ma. SSR: Alignment-aware modality connector for speech language models. In Elizabeth Salesky, Marcello Federico, and Antonis Anastasopoulos (eds.), *International Conference on Spoken Language Translation (IWSLT 2025)*, pp. 56–75, 2025.

Jinchuan Tian, Jiatong Shi, William Chen, Siddhant Arora, Yoshiki Masuyama, Takashi Maekaku, Yihan Wu, Junyi Peng, Shikhar Bharadwaj, Yiwen Zhao, Samuele Cornell, Yifan Peng, Xiang Yue, Chao-Han Huck Yang, Graham Neubig, and Shinji Watanabe. ESPnet-SpeechLM: An open speech language model toolkit. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies (System Demonstrations)*, pp. 116–124, 2025.

Liang-Hsuan Tseng, Yi-Chang Chen, Kuan-Yi Lee, Da-Shan Shiu, and Hung yi Lee. TASTE: Text-aligned speech tokenization and embedding for spoken language modeling. *arXiv preprint arXiv:2504.07053*, 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.

S. Van Kuyk, W. B. Kleijn, and R. C. Hendriks. On the information rate of speech communication. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5625–5629, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6000–6010, 2017.

Hankun Wang, Yiwei Guo, Chongtian Shao, Bohan Li, Xie Chen, and Kai Yu. CodecSlime: Temporal redundancy compression of neural speech codec via dynamic frame rate. *arXiv preprint arXiv:2506.21074*, 2025a.

Yuancheng Wang, Dekun Chen, Xueyao Zhang, Junan Zhang, Jiaqi Li, and Zhizheng Wu. TaDiCodec: Text-aware diffusion speech tokenizer for speech language modeling. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2025b.

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.

Ziqian Wang, Xinfa Zhu, Zihan Zhang, YuanJun Lv, Ning Jiang, Guoqing Zhao, and Lei Xie. SELM: Speech enhancement using discrete tokens and language models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11561–11565, 2024.

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.

Haibin Wu, Naoyuki Kanda, Sefik Emre Eskimez, and Jinyu Li. TS3-Codec: Transformer-based simple streaming single codec. In *Interspeech*, pp. 604–608, 2025.

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.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, Hongzhan Lin, Jianyi Chen, Xingjian Du, Liumeng Xue, Yunlin Chen, Zhifei Li, Lei Xie, Qiuqiang Kong, Yike Guo, and Wei Xue. Llasa: Scaling train-time and inference-time compute for Llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025.

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 (TASLP)*, 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, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

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.

Hanglei Zhang, Yiwei Guo, Zhihan Li, Xiang Hao, Xie Chen, and Kai Yu. Unlocking temporal flexibility: Neural speech codec with variable frame rate. In *Interspeech*, pp. 5003–5007, 2025.

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.

Rui-Chen Zheng, Wenrui Liu, Hui-Peng Du, Qinglin Zhang, Chong Deng, Qian Chen, Wen Wang, Yang Ai, and Zhen-Hua Ling. Say more with less: Variable-frame-rate speech tokenization via adaptive clustering and implicit duration coding. In *AAAI Conference on Artificial Intelligence*, 2026.

Youqiang Zheng, Weiping Tu, Yueteng Kang, Jie Chen, Yike Zhang, Li Xiao, Yuhong Yang, and Long Ma. FreeCodec: A disentangled neural speech codec with fewer tokens. In *Interspeech*, pp. 4878–4882, 2025.

A DATASETS

The following datasets were used in this work:

- **LibriSpeech** (Panayotov et al., 2015) is a large-scale corpus of English read speech derived from audiobooks in the LibriVox project. It contains approximately 1000 hours of speech sampled at 16 kHz, with predefined training, validation, and test splits. License: CC BY 4.0.
- **LibriTTS** (Zen et al., 2019) is a corpus designed for text-to-speech research, constructed from the same source as LibriSpeech. It consists of 585 hours of transcribed speech with predefined training, validation, and test splits. License: CC BY 4.0.
- **MLS** (Pratap et al., 2020) is an extension of LibriSpeech to multiple languages, including English, German, Dutch, French, Spanish, Italian, Portuguese and Polish. It provides approximately 44,500 hours of transcribed English speech and about 6000 hours from other languages. License: CC BY 4.0.
- **VoiceBank** (Valentini-Botinhao et al., 2016) is a dataset primarily used for speech enhancement, including 11,572 utterances from 28 speakers in the training set (noise at 0 dB, 5 dB, 10 dB, and 15 dB), and 872 utterances from 2 unseen speakers in the test set (noise at 2.5 dB, 7.5 dB, 12.5 dB, and 17.5 dB). License: CC BY 4.0.
- **LibriMix** (Cosentino et al., 2020) is a dataset for speech separation and enhancement, created by mixing LibriSpeech utterances with noise from the WHAM! (Wichern et al., 2019) corpus. It provides mixtures of two or three speakers at different signal-to-noise ratios. License: MIT.
- **VCTK** (Yamagishi et al., 2017) is a corpus of English speech recordings from 110 speakers with various accents. It is widely used for speaker adaptation, text-to-speech, and voice conversion tasks. License: CC BY 4.0.
- **IEMOCAP** (Busso et al., 2008) is a dataset designed for emotion recognition, consisting of scripted and improvised dialogues performed by 10 actors. It includes audio, video, and textual transcriptions with emotion labels such as happiness, sadness, and anger. License: https://sail.usc.edu/iemocap/iemocap_release.htm.

B BASELINES

We compare DyCAST against a diverse set of fixed-frame-rate neural audio codecs covering general audio, speech-focused, single- and multi-codebook, acoustic, semantic, and hybrid (additional details are provided in Table 4):

- **EnCodec** (Défossez et al., 2023): A general-purpose neural audio codec supporting both causal and non-causal operation, trained on large-scale heterogeneous audio data and widely used as a strong baseline for speech and audio compression.
- **DAC** (Kumar et al., 2023): A high-quality non-causal audio codec optimized for perceptual reconstruction, trained on diverse speech and music datasets and commonly used in downstream speech tasks.
- **WavLM6-KM** (Wang et al., 2024): A speech codec obtained by clsutering WavLM representations from the 6-th transfomer layer.
- **SpeechTokenizer** (Zhang et al., 2024): A multi-codebook speech tokenizer designed to disentangle content and speaker information across codebooks, primarily trained on LibriSpeech.
- **SemantiCodec** (Liu et al., 2024): A general audio codec trained on large-scale multimodal datasets, explicitly targeting semantic preservation across speech and non-speech audio.
- **Mimi** (Défossez et al., 2024): A large-scale causal speech tokenizer trained on millions of hours of speech, designed for streaming and speech language modeling applications.
- **WavTokenizer** (Ji et al., 2025): A unified audio tokenizer trained on mixed speech and music data, supporting multilingual and multi-domain speech modeling.
- **BigCodec** (Xin et al., 2024): A speech-focused codec trained on LibriSpeech, emphasizing efficient discrete representations for downstream speech generation and modeling tasks.

Table 4: Baseline codecs.

Codec	Training Datasets	Hours	Multilingual	Audio Domain	Checkpoint	License
EnCodec (Défossez et al., 2023)	DNS, CommonVoice, AudioSet, FSD50K, Jamendo	17k+	Yes	General	encodec_24khz	MIT
DAC (Kumar et al., 2023)	DAPS, DNS, CommonVoice, VCTK, MUSDB, Jamendo	10k+	Yes	General	weights_16khz.pth	MIT
WavLM6-KM (Wang et al., 2024)	Subset of LibriSpeech (in addition to Libri-Light, GigaSpeech, and VoxPopuli English for WavLM pretraining)	460 (+ 94k)	No	Speech	discrete-wavlm-codec	Apache 2.0
SpeechTokenizer (Zhang et al., 2024)	LibriSpeech	960	No	Speech	speechtokenizer_hubert_avg	Apache 2.0
SemantiCodec (Liu et al., 2024)	GigaSpeech, subset of OpenSLR, Million Song Dataset, MedleyDB, MUSDB18, AudioSet, WavCaps, VGGSound	20k+	Yes	General	semanticodec_tokenrate_50	MIT
Mimi (Défossez et al., 2024)	Predominantly English speech (in addition to Libri-Light, GigaSpeech, and VoxPopuli English for WavLM pretraining)	7M (+ 94k)	Likely	Speech	mimi	CC BY 4.0
WavTokenizer (Ji et al., 2025)	LibriTTS, VCTK, subset of CommonVoice, subset of AudioSet, Jamendo, MUSDB	8k	Yes	General	WavTokenizer-large-unify-40token	MIT
BigCodec (Xin et al., 2024)	LibriSpeech	960	No	Speech	bigcodec.pt	MIT
Stable Codec (Parker et al., 2025)	Libri-Light, MLS English	105k	No	Speech	stable-codec-speech-16k	StabilityAI
FocalCodec (Della Libera et al., 2025)	LibriTTS (in addition to Libri-Light, GigaSpeech, and VoxPopuli English for WavLM pretraining)	585 (+ 94k)	No	Speech	focalcodec_50hz	Apache 2.0

- **Stable Codec** (Parker et al., 2025): A transformer-based neural speech codec trained on large-scale speech corpora, supporting optional causal inference.
- **FocalCodec** (Della Libera et al., 2025): A low-bitrate speech codec based on focal modulation and binary spherical quantization, serving as the closest architectural baseline to DyCAST.

C Hyperparameters and Training Details

We summarize the main hyperparameters and training settings used for all DyCAST components. All learnable modules are trained on LibriTTS (Zen et al., 2019) (585 hours from 2456 speakers) using full utterances and identical optimizer settings, except for the decoder, which requires a different training setup. For all other modules, training is performed using the AdamW (Loshchilov & Hutter, 2019) optimizer with an initial learning rate of 0.0005, $\beta_1 = 0.8$, $\beta_2 = 0.99$, and a weight decay of 0.01. The learning rate is reduced by a factor of 0.9 when the validation loss does not improve by at least 0.0025. Gradients are clipped to a maximum L_2 norm of 5, and training stops when the validation loss fails to decrease for several consecutive epochs.

Character Aligner. As a character aligner, we use MMS (Pratap et al., 2024), a 1B-parameter multilingual wav2vec2.0 ASR model pretrained on approximately 500k hours of speech across more than 1,000 languages and equipped with a CTC-based character prediction head. The model is kept frozen throughout training and is used to obtain character-level durations via simple `argmax` decoding. While more sophisticated alignment strategies such as Viterbi decoding (Graves et al., 2006) are possible, we adopt `argmax` decoding for its simplicity and efficiency.

Encoder. For the encoder, we use the pretrained WavLM-large[3] model (Chen et al., 2022). In particular, we extract representations from the 6-th transformer layer, which provide semantically rich features while retaining fine-grained acoustic information (Della Libera et al., 2025; Baas et al., 2023). The encoder is kept frozen throughout training.

Compressor. The compressor takes as input 1024-dimensional WavLM features and processes them through 3 focal downscaling blocks (Della Libera et al., 2025) while preserving the original temporal resolution (50 Hz), each with a hidden dimension of 1024. Each block uses 2 focal levels, a window size of 14, a focal factor of 4, and a layer scale initialization of 0.0001. A final projection maps the 1024-dimensional hidden states to 32-dimensional latent representations.

Chunker. Within the chunker, the boundary predictor adopts the same architecture as the compressor. A final binary classification head maps the 1024-dimensional hidden states to boundary logits. The dynamic downsampling operation itself is implemented as a simple frame-selection and does not introduce additional trainable parameters.

Quantizer. The scalar spherical quantizer discretizes the continuous 32-dimensional latent representations using 4 levels per dimension, resulting in an implicit codebook of size $|\mathcal{C}| = 4^{32}$. In

[3]https://github.com/microsoft/unilm/tree/master/wavlm

practice, it is infeasible to enumerate all codes explicitly; instead, we represent tokens by returning per-dimension indices in $\{0, 1, 2, 3\}$. The quantizer has no trainable parameters, and we set the entropy loss weight to 0.1.

Dechunker. Within the dechunker, the duration predictor follows the same architectural design but processes 32-dimensional pooled quantized latents. The model predicts a free mean parameter (a scalar value) together with a shared, learnable scalar dispersion parameter α. Dynamic upsampling is implemented by frame repetition and introduces no additional parameters; the length regularization strength is set to $\lambda = 0.05$.

Decompressor. The decompressor mirrors the compressor architecture, replacing focal downscaling blocks with focal upscaling blocks to reconstruct 1024-dimensional continuous representations from the quantized latent codes.

Decoder. The Vocos (Siuzdak, 2024) decoder operates on 1024-dimensional WavLM features and processes them through 20 ConvNeXt blocks with a hidden dimension of 1024, a feed-forward dimension of 2048, a kernel size of 7, and padding of 3. For the STFT, we use an FFT size of 1024 samples and a hop length of 320. The feature-matching loss is computed using 80-dimensional log-Mel spectrograms with the same STFT configuration. As discriminators, we employ the multi-scale and multi-period discriminators from Kong et al. (2020), along with an additional multi-resolution discriminator from Kumar et al. (2023). We further incorporate a speaker consistency loss from Casanova et al. (2025), using WavLM-base-SV (Chen et al., 2022) as a pretrained speaker embedding extractor. Training is conducted on LibriTTS `train-clean-100` using audio chunks of 7040 samples and a batch size of 16. We use the AdamW optimizer with an initial learning rate of 0.0002, β_1 of 0.8, β_2 of 0.99, and a weight decay of 0.01. The learning rate follows an exponential decay schedule with a factor of 0.999. Training continues until perceived audio quality saturates, which occurs after approximately 4M steps.

D Downstream Evaluation

In this section, we outline the downstream tasks used to evaluate the proposed codec and describe the corresponding experimental setups. Unless otherwise stated, all downstream models are trained using the AdamW optimizer (Loshchilov & Hutter, 2019) with a batch size of 16, an initial learning rate of 0.0001, β_1 = 0.8, β_2 = 0.99, weight decay of 0.01, and dropout of 0.1. The learning rate is reduced by a factor of 0.9 if validation loss does not improve within a margin of 0.0025. Gradients are clipped to a maximum L_2 norm of 0.01. Training stops if validation loss does not decrease for several consecutive epochs.

D.1 Speech Resynthesis (SR)

This task evaluates the ability of the codec to reconstruct high-quality speech from discrete tokens while preserving naturalness, speaker identity, and intelligibility. For English, we use the **LibriSpeech** (Panayotov et al., 2015) `test-clean` split. For multilingual evaluation, we use the dataset constructed by Della Libera et al. (2025), which was built by selecting 100 utterances from each of the seven non-English languages in **MLS** (Pratap et al., 2020) (Dutch, French, German, Italian, Polish, Portuguese, and Spanish) for a total of 700 utterances[4]. To assess robustness to environmental noise, we additionally evaluate on the test sets of **VoiceBank** (Valentini-Botinhao et al., 2016) and **Libri1Mix**, constructed by mixing clean utterances from the first speaker of LibriMix (Cosentino et al., 2020) with noise from WHAM! (Wichern et al., 2019). Naturalness is measured using **UTMOS** (Saeki et al., 2022) for clean speech and **DNSMOS** (Reddy et al., 2022) for noisy speech. Intelligibility is evaluated using differential word error rate (**dWER**) (Wang et al., 2021), computed from transcriptions obtained with Whisper-small (Radford et al., 2023)[5]. Speaker fidelity is assessed via cosine similarity (**Sim**) between speaker embeddings extracted using WavLM-base-SV (Chen et al., 2022)[6]. We do not report signal-level metrics such as SNR,

[4] https://zenodo.org/records/14791114

[5] https://huggingface.co/openai/whisper-small

[6] https://huggingface.co/microsoft/wavlm-base-sv

PESQ (Rix et al., 2001), or STOI (Taal et al., 2011), as they correlate poorly with perceived reconstruction quality (Parker et al., 2025; Wang et al., 2024; Della Libera et al., 2025), and we avoid stronger ASR models to prevent masking reconstruction artifacts.

D.2 Voice Conversion (VC)

This task evaluates the ability of the codec to disentangle speaker identity from linguistic content despite its single-codebook design. Voice conversion is performed by converting speech from a source speaker to a target speaker using reference speech from the target speaker. For single-codebook baselines, including DyCAST, we follow the k-nearest neighbors (kNN) approach of Baas et al. (2023). Specifically, each frame in the reconstructed feature sequence (immediately before the decoder) is replaced by the average of the $k = 4$ nearest neighbors (measured by cosine similarity) from continuous features extracted from the target speaker reference utterance. For multi-codebook baselines, we follow the procedure of Zhang et al. (2024). Both source and reference speech are tokenized, and voice conversion is performed by concatenating the first codebook tokens from the source with the second-to-last codebook tokens from the reference, before decoding. If sequence lengths differ, the reference sequence is truncated or circularly padded as needed. Effective disentanglement of content and speaker information across codebooks is expected to yield strong voice conversion performance. We conduct voice conversion experiments on **VCTK** (Yamagishi et al., 2017), which contains parallel utterances from multiple speakers. For each sample, we randomly select (i) an utterance from a source speaker, (ii) the corresponding parallel utterance from a target speaker, and (iii) a different utterance from the same target speaker to serve as the reference. Among candidate reference utterances, we select the longest to minimize padding. Repeating this procedure for each speaker and each of the approximately 24 parallel utterances yields a total of 2,521 evaluation samples. Performance is evaluated using the same metrics as in speech resynthesis: **UTMOS**, **dWER**, and **Sim**.

D.3 Automatic Speech Recognition (ASR)

This task evaluates the quality of the learned discrete representations through a probing setup, where a shallow model is trained on top of fixed codec tokens to assess their linguistic content. We use **LibriSpeech** (Panayotov et al., 2015) `train-clean-100` and `train-clean-360` for training, `dev-clean` for validation, and `test-clean` for evaluation. The ASR probe consists of a 2-layer bidirectional LSTM with 512-dimensional hidden states, followed by a CTC (Graves et al., 2006) prediction head. The model is trained to predict either characters or byte-pair encoding (BPE) units, using BPE vocabularies of size 1000, except for Mimi, which achieves its best performance with a vocabulary size of 500. When a codec employs multiple codebooks, we compute a learned weighted sum of the embeddings from each codebook, following (Chen et al., 2022). The embedding layer is initialized using the discrete embeddings from the codec quantizer. Performance is reported in terms of word error rate (**WER**).

D.4 Speaker Identification (SI)

This task probes whether the learned discrete representations encode speaker identity through a probing setup, where a shallow model is trained on top of fixed codec tokens to assess their acoustic content. We use **LibriSpeech** (Panayotov et al., 2015), grouping utterances from `train-clean-100` and `train-clean-360` by speaker ID. The data are randomly split into training, validation, and test sets with a ratio of 80% / 10% / 10%. The SI probe closely mirrors the ASR setup. A 2-layer bidirectional LSTM with 512-dimensional hidden states processes the token sequence, and the output is aggregated using statistics pooling, followed by a cross-entropy classification head. When multiple codebooks are present, embeddings are combined via a learned weighted sum, as in ASR. Performance is reported in terms of speaker error rate (**ER**).

D.5 Speech Emotion Recognition (SER)

This task probes whether the learned representations capture paralinguistic information related to emotion. We again adopt a probing setup with a shallow classifier trained on top of frozen codec tokens. We use the **IEMOCAP** dataset (Busso et al., 2008), focusing on four emotion classes: sadness, happiness, anger, and neutral. Sessions 1–4 are used for training, session 5F for validation,

and session 5M for testing. The SER setup is identical to SI, with the only difference being the number of output classes in the classification head. Performance is reported in terms of emotion error rate (**ER**).

D.6 TEXT-TO-SPEECH (TTS)

This task evaluates the suitability of discrete speech representations for speech generation from text. We use **LibriSpeech** `train-clean-100` and `train-clean-360` for training, `dev-clean` for validation, and `test-clean` for evaluation. The `test-clean` split contains a small fraction of long utterances (approximately 4%) exceeding 20 seconds, which are largely absent from the training data; to reduce this train-test mismatch, we remove these long utterances from the test set. The model input consists of character-level text tokens, while the target sequence comprises speech tokens extracted from the corresponding utterances. We consider two TTS model variants. An **autoregressive** model is used for all codecs except DyCAST-CA, while a **non-autoregressive** model is used exclusively for DyCAST-CA to exploit the availability of character-aligned tokens. Performance is evaluated using **UTMOS**, **dWER**, and **Sim**.

Autoregressive TTS. The autoregressive model is a Llama 3 decoder (Grattafiori et al., 2024) with 12 layers, 4 attention heads, 1 key-value head, a model dimension of 512, a feed-forward dimension of 2048, and a base RoPE frequency of 10,000. Speaker identity is provided by extracting embeddings from the target utterance using WavLM-base (Chen et al., 2022), fine-tuned for speaker verification. The pooled speaker embedding is prepended to the character embeddings to condition the model on speaker identity. The embedding layer is initialized using the discrete embeddings from the codec quantizer. Training is performed using next-token prediction, where the input sequence consists of the pooled speaker embedding, character embeddings, and speech token embeddings. The cross-entropy loss is computed only over speech tokens, while character and speaker embeddings are excluded from the loss. If the codec has multiple codebooks, we flatten tokens across the codebook dimension, as this approach has been shown to provide the best downstream performance given a sufficiently high computational budget (Copet et al., 2023). At inference time, we use top-p sampling with $\beta_1 = 0.9$ and a temperature of 1.0. Following Tian et al. (2025); Della Libera et al. (2025), we generate five samples per utterance and select the one with the lowest word error rate relative to the input text, computed using Whisper-small (Radford et al., 2023).

Non-autoregressive TTS. The non-autoregressive model employs the same Llama 3 architecture, where causal self-attention is replaced with bidirectional self-attention, while all other components remain unchanged. The pooled speaker embedding is replicated for the number of character positions and summed with the hidden representations at each layer to condition the model on speaker identity. Training is performed using per-character target prediction with a cross-entropy loss at each step. At inference time, decoding is performed using `argmax`.

E IMPLEMENTATION AND HARDWARE

Software for the experimental evaluation was implemented in Python using PyTorch (Paszke et al., 2019) and SpeechBrain (Ravanelli et al., 2021; 2024). Each model is trained on a single NVIDIA H100 GPU with 80 GB of memory.

F ADDITIONAL RESULTS

F.1 ALIGNMENT QUALITY AND BOUNDARY ANALYSIS

We evaluate the quality of MMS-based alignments and the learned boundary predictor on the TIMIT (Garofolo et al., 1993) test set using gold word-level timestamps.

CTC Alignment With Ground-Truth Transcripts. We first assess Viterbi alignment using ground-truth transcripts on all 1,680 utterances in the TIMIT test set (14,553 words). The alignments exhibit a consistent bias: predicted word starts are delayed by +62.5 ms on median, while predicted word ends are advanced by -32.5 ms. The median absolute error is 62.5 ms for word

starts and 38.8 ms for word ends. The median intersection-over-union (IoU) between predicted and ground-truth word spans is 0.633. Additionally, 68.0% of word starts and 87.9% of word ends fall within 80 ms of the ground-truth boundaries. Given the 50 Hz frame rate of MMS (20 ms resolution), these errors correspond to only a few frames.

CTC Alignment With Predicted Transcripts. We repeat the analysis using greedy `argmax` decoding with transcripts predicted by MMS, corresponding to the setting used in DyCAST. In this setup, MMS predicts 14,555 words, of which 13,249 match the ground truth after normalization, corresponding to 91.0% precision and recall. The timing characteristics remain consistent: the median start error is +64.3 ms and the median end error is -33.8 ms, indicating that predicted boundaries tend to lie slightly inside the true word span. The median absolute error is 64.4 ms for word starts and 35.6 ms for word ends, with a median IoU of 0.618. Furthermore, 66.5% of word starts and 88.9% of word ends fall within 80 ms of the ground-truth boundaries, confirming that alignment quality remains stable under realistic inference conditions.

Boundary Predictor vs Character Aligner. We further evaluate the agreement between the learned boundary predictor (greedy decoding, $\texttt{min_gap} = 1$, corresponding to DyCAST-BP1) and the CTC-derived segmentation used for supervision. On the TIMIT test set, the predictor generates $1.23\times$ more segments than the teacher, indicating mild over-segmentation. Boundary agreement is strong: 86.3% of teacher boundaries are recovered within 20 ms, 92.7% within 40 ms, and 95.1% within 60 ms. Conversely, 75.1% of predicted boundaries match a teacher boundary within 40 ms, and more than 99.2% of teacher boundaries are covered within 40 ms. The median distance between matched boundaries is 20 ms (approximately one frame), and no systematic timing bias is observed (median signed error of 0 ms). At the segment level, agreement remains reasonable, with a median best-IoU of 0.5 between predicted and teacher segments.

Summary. Overall, MMS provides reasonably accurate word-level temporal localization, despite a consistent inward bias (delayed starts and early ends). The learned boundary predictor closely reproduces the CTC-derived segmentation, with only limited additional over-segmentation. While not suitable for fine-grained phonetic alignment, the resulting boundaries are sufficiently accurate for approximate segmentation and dynamic chunking, which are the primary use cases considered in this work.

F.2 Decoding Mode

Table 5 analyzes the impact of different decoding modes on LibriSpeech speech resynthesis. Overall, explicitly providing duration information consistently improves intelligibility, while naturalness and speaker similarity remain largely stable across decoding modes. For all DyCAST variants, *tokens + durations* yields the lowest dWER, confirming that explicit duration supervision enables more accurate temporal reconstruction and reduces linguistic distortions. This effect is particularly pronounced for predicted-duration variants, where dWER increases moderately as duration information is removed, highlighting the importance of duration modeling at low token rates. In contrast, *tokens only* decoding achieves comparable or slightly higher UTMOS across all variants, indicating that naturalness is largely preserved even without explicit timing cues. However, this comes at the cost of degraded intelligibility, especially for aggressive compression regimes, where the absence of duration constraints leads to temporal misalignment and increased recognition errors. The intermediate *tokens + utterance length* mode offers a favorable trade-off, recovering most of the intelligibility gains of full duration decoding while requiring only a global length constraint. Notably, this mode performs robustly across all variants and is used as the default setting in our experiments unless otherwise stated. Overall, these results show that DyCAST enables flexible decoding strategies with explicit control over the trade-off between timing accuracy, intelligibility, and decoding complexity, further emphasizing the practical advantages of variable-rate tokenization with explicit duration modeling.

F.3 Visualization of DyCAST Chunk Boundaries

Figure 2 provides qualitative spectrogram visualizations of DyCAST chunk boundaries for the four variants considered in this work: DyCAST-CA, DyCAST-BP1, DyCAST-BP3, and DyCAST-BP5.

Table 5: Effect of different decoding modes on LibriSpeech resynthesis.

Codec	Decoding Mode	UTMOS ↑	dWER ↓	Sim ↑
DyCAST-CA	Tokens + Durations	3.91	**3.01**	**97.6**
	Tokens + Utterance Length	4.06	3.22	97.5
	Tokens only	**4.09**	3.31	**97.6**
DyCAST-BP1	Tokens + Durations	**4.00**	**2.28**	**97.8**
	Tokens + Utterance Length	3.99	3.63	97.7
	Tokens Only	3.99	4.59	97.7
DyCAST-BP3	Tokens + Durations	4.00	**2.95**	**97.6**
	Tokens + Utterance Length	4.01	4.60	97.4
	Tokens Only	**4.02**	5.08	97.4
DyCAST-BP5	Tokens + Durations	4.00	**5.21**	**96.8**
	Tokens + Utterance Length	4.03	8.50	96.6
	Tokens Only	**4.04**	8.41	96.6

DyCAST-CA relies on boundaries provided by the character aligner, while the remaining variants use the boundary predictor with `min_gap` set to 1, 3, and 5, respectively, resulting in average frame rates of approximately 14 Hz, 17 Hz, 9 Hz, and 6 Hz. As the frame rate decreases, chunk boundaries become less frequent and more widely spaced, reflecting longer token durations and coarser temporal resolution.

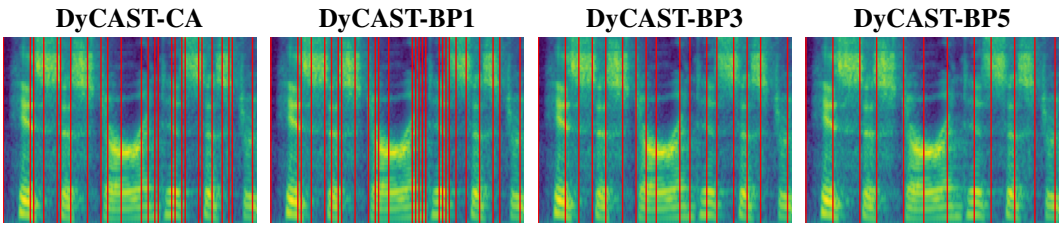

Figure 2: DyCAST chunk boundaries at average frame rates of approximately 14 Hz, 17 Hz, 9 Hz, and 6 Hz (from left to right).