

Tokenizers, TTS Models, Fundamentals Models

1 Ye, Zhen, Xinfu Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, et al. "Llasa: Scaling Train-Time and Inference-Time Compute for Llama-Based Speech Synthesis." arXiv, February 22, 2025. .

1.1 Model Architecture

1.2 Training

1.3 Novel Contributions and Validations

2 Zhang, Xin, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. "SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models." arXiv, January 23, 2024. .

2.1 Key Contributions

2.2 Key Results

3 Borsos, Zalán, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, et al. "AudioLM: A Language Modeling Approach to Audio Generation." arXiv, July 26, 2023. .

3.1 Key Innovation

3.2 Demonstrated Capabilities

3.3 Key Results

4 Zeghidour, Neil, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. "SoundStream: An End-to-End Neural Audio Codec." arXiv, July 7, 2021. .

4.1 Info

4.2 Performance Highlights

4.3 Technical Contributions

5 Hsu, Wei-Ning, 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." arXiv, June 14, 2021. .

5.1 Model

5.2 Training

5.3 Key Advantages

5.4 Performance

5.5 Key Findings from Ablations

5.6 Conclusion

6 Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations." arXiv, October 22, 2020. .



6.1 Key Innovations:

6.2 Model

6.3 Training

6.4 Results:

6.5 Key Finding:

**Ye, Zhen, Xinfu Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, et al. "Llasa: Scaling Train-Time and Inference-Time Compute for Llama-Based Speech Synthesis." arXiv, February 22, 2025.  [Llasa: Scaling Train-Time and Inference-Time Co](#)
[mpute for...](#) . **

- [Models](#)
- [Llasa Training Code](#)
- [Codec Training Code](#)
- [Inference-time Scaling Code](#)
- [Demo page](#)

Model Architecture [↗](#)

Llaza consists of two main components:

1. **Tokenizer:** Text tokenizer inherited from Llama + Speech tokenizer (Xcodec2)
2. **Transformer:** Single Transformer-based LLM initialized from Llama (1B, 3B, or 8B parameters)

Speech Tokenizer (Xcodec2) Architecture

1. Encoder System

- Semantic Encoder (Encs): Based on pre-trained Wav2Vec2-BERT
- Acoustic Encoder (Enca): Employs multiple residual convolutional blocks with Snake activation functions
- The outputs from both encoders are concatenated to form a unified feature embedding (H) that represents both semantic and acoustic aspects of the speech signal.

2. Vector Quantization Module

- Uses FSQ (Fully Separable Quantization) to convert continuous features to discrete tokens
- Key advantages of FSQ:
 - Training stability
 - High codebook usage efficiency
 - No need for explicit VQ objective terms (like codebook commitment loss)
- Xcodec2 adopts a single vector quantizer. This design choice ensures 1D causal dependency, which aligns naturally with:
 - Left-to-right autoregressive processing in LLMs
 - The inherent temporal structure of audio signals

3. Decoder System

The decoder reconstructs both semantic and acoustic information from the quantized representation:

- Semantic Decoder: (not used during inference)
 - Predicts semantic features using an l2 loss for reconstruction
 - Primarily serves as a supervisory signal during training
 - Ensures the codebook retains sufficient semantic information
- Acoustic Decoder:
 - Transformer-based architecture (replacing the ConvNeXt backbone used in earlier versions)
 - Predicts both STFT (Short-Time Fourier Transform) magnitude and phase
 - Includes an inverse STFT (iSTFT) head to convert spectral predictions back to time-domain waveforms

Training [↗](#)

Speech Tokenizer (Xcodec2) Training

1. Data: Approximately 150,000 hours of multilingual speech from:
 - Emilia dataset (English, Chinese, German, French, Japanese, Korean)
 - MLS dataset (English, French, German, Dutch, Spanish, Italian, Portuguese, Polish)
2. Configuration
 - Downsampling ratio: 320
 - Codebook size: 65,536
3. Training Details: 1.4 million training steps total
 - Random 6-second audio crops
 - Learning rate: 1×10^{-4} with 3,000-step warmup
 - Perceptual loss activated during final 0.2 million steps

TTS Model Training

1. **Data:** 250,000 hours of mixed Mandarin Chinese and English speech from:
 - Libriheavy
 - Chinese-English subset from Emilia corpus
 - WenetSpeech4TTS
 - Internal data
 - Text maintained with original punctuation
2. **Training Configuration**
 - 3 epochs total, Batch size: 2 million tokens
 - Maximum learning rate: 5e-5
 - Cosine learning rate schedule with:
 - 3% of an epoch warmup
 - Final learning rate at 10% of peak
 - Text and speech sequences concatenated and cropped to 2048 tokens maximum

Novel Contributions and Validations [🔗](#)

1. **Unified LLM-style TTS Framework**
 - Validated that a single Transformer architecture with a speech tokenizer can achieve competitive TTS results
 - Demonstrated that the architecture simplification doesn't inherently disadvantage the system on key metrics
2. **Training-time Scaling Laws for TTS**
 - Proved that larger models (8B) significantly enhance performance on complex tasks requiring deeper semantic comprehension (emotional speech, poetry, tongue twisters)
 - Demonstrated that certain tasks (rare characters, compound nouns, foreign words) benefit more from increased data than model size
3. **Inference-time Scaling for TTS**
 - Established that inference-time computation (search strategies) can significantly improve TTS quality
 - Proved that Process Reward Models outperform Output Reward Models under the same compute budget
 - Validated a novel hybrid approach (partial PRM + ORM) that balances speaker similarity and transcription accuracy
 - Showed that smaller models with inference-time scaling can approach larger model performance on certain tasks
4. **Speech Tokenizer Design**
 - Created a unified codebook integrating both semantic and acoustic features
 - Substituted residual vector quantization with a single vector quantizer to ensure 1D causal dependency
 - Validated this approach achieves state-of-the-art performance at a token rate of 50
5. **Cross-model Performance Patterns**
 - Rejected the assumption that AR+NAR hybrid architectures are inherently superior to single Transformer designs
 - Demonstrated the primary limitation is in the acoustic reconstruction of the codec rather than in the core generative modeling

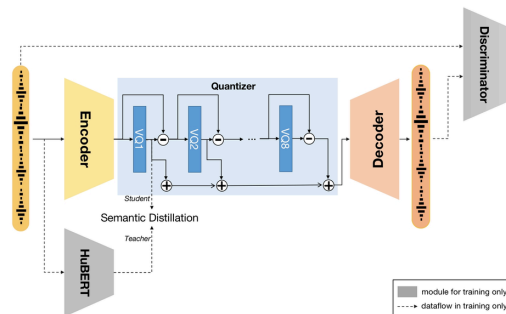
Table 1. Comparison between different codec models. Bold values indicate the best for each token rate. We use token rate instead of bitrate because, from the perspective of LLMs, it is more intuitive: dividing the speech context window length by the token rate directly gives the generated audio duration in seconds.

Model	Token Rate	Codebook Size	Codebook Layer	Frame Rate	WER ↓	STOI ↑	PESQ-WB ↑	PESQ-NB ↑	SPK-SIM ↑	UT-MOS ↑
Ground Truth	-	-	-	-	1.96	1.00	4.64	4.55	1.00	4.09
DAC	600	1024	12	50	2.00	0.95	4.01	4.15	0.95	4.00
Encodec	600	1024	8	75	2.15	0.94	2.77	3.18	0.89	3.09
Encodec	150	1024	2	75	4.90	0.85	1.56	1.94	0.60	1.58
DAC	100	1024	2	50	13.27	0.73	1.13	1.40	0.32	1.29
SpeechTokenizer	100	1024	2	50	3.92	0.77	1.25	1.59	0.36	2.28
Mimi	100	2048	8	12.5	2.96	0.91	2.25	2.80	0.73	3.56
X-codec	100	1024	2	50	2.49	0.86	2.33	2.88	0.72	4.21
BigCodec	80	8192	1	80	2.76	0.93	2.68	3.27	0.84	4.11
WavTokenizer	75	4096	1	75	3.98	0.90	2.13	2.63	0.65	3.79
Mimi	75	2048	6	12.5	3.61	0.89	1.99	2.51	0.65	3.38
Encodec	75	1024	1	75	28.92	0.77	1.23	1.48	0.25	1.25
DAC	50	1024	1	50	74.55	0.62	1.06	1.20	0.08	1.25
SpeechTokenizer	50	1024	1	50	5.01	0.64	1.14	1.30	0.17	1.27
Mini	50	2048	4	12.5	4.89	0.85	1.64	2.09	0.50	3.03
StableCodec	50	15625	2	25	5.12	0.91	2.24	2.91	0.62	4.23
SemantiCodec	50	32768/8192	2	25	6.89	0.84	1.66	2.18	0.58	2.71
X-codec	50	1024	1	50	3.42	0.83	1.84	2.38	0.52	4.05
WavTokenizer	40	4096	1	40	11.20	0.85	1.62	2.06	0.48	3.57
X-codec2 (ours)	50	65536	1	50	2.47	0.92	2.43	3.04	0.82	4.13

Zhang, Xin, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. “SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models.” arXiv, January 23, 2024.

✗ [SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models](#) . [🔗](#)

This paper addresses a key limitation in speech language models: current approaches rely on either semantic tokens (good for content but poor for audio quality) or acoustic tokens (good for quality but less accurate for content), or use both in complex multi-stage systems that suffer from error propagation.



Key Contributions [🔗](#)

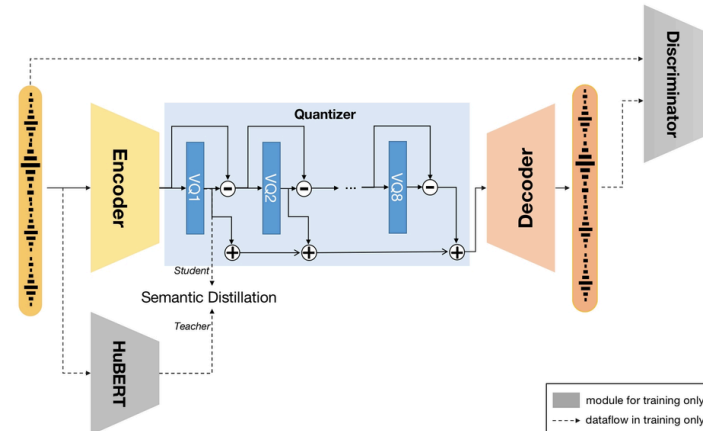
1. **SLMTokBench**: The first benchmark to evaluate how well speech tokens perform for language modeling, measuring both text alignment and information preservation.
2. **SpeechTokenizer**: A unified speech tokenizer that combines the strengths of semantic and acoustic tokens through hierarchical information disentanglement:
 - a. Uses an Encoder-Decoder architecture with residual vector quantization (RVQ)
 - b. First RVQ layer captures content information (guided by HuBERT as a semantic teacher)
 - c. Subsequent RVQ layers capture paralinguistic information (timbre, prosody, etc.)
3. Unified Speech Language Model (USLM)**: Built on SpeechTokenizer, combining:
 - a. An autoregressive (AR) model for content modeling
 - b. A non-autoregressive (NAR) model for acoustic details

Key Results [🔗](#)

- Speech Reconstruction: SpeechTokenizer achieves better content preservation (lower WER) than EnCodec while maintaining comparable audio quality.
- Information Disentanglement: Successfully separates content from speaker information across layers, as demonstrated through visualization and one-shot voice conversion experiments.

- Zero-Shot TTS: USLM outperforms VALL-E in both content accuracy and speaker similarity.

The research proves that properly designed speech tokens with hierarchical disentanglement can overcome the limitations of existing approaches, offering a more elegant and effective solution for speech language modeling.



- Content preservation is evaluated by computing the WER through transcribing the resynthesized speech using the Whisper en-medium model (Radford et al., 2023).
- Timbre preservation is evaluated by utilizing WavLM-TDNN (Chen et al., 2022) to calculate speaker similarity between the synthesized and groundtruth speech.

Borsos, Zalán, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, et al. “AudioLM: A Language Modeling Approach to Audio Generation.” arXiv, July 26, 2023. [✗ AudioLM: a Language Modeling Approach to Audio Generation](#) . [🔗](#)

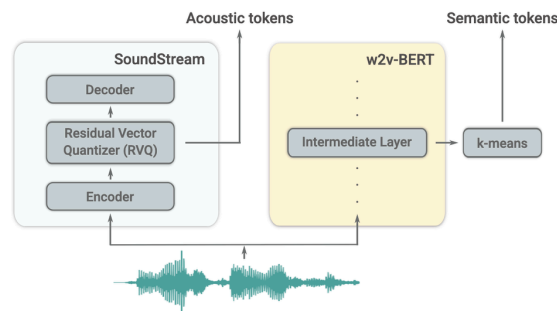


Fig. 1. Overview of the tokenizers used in AudioLM. The acoustic tokens are produced by SoundStream [16] and enable high-quality audio synthesis. The semantic tokens are derived from representations produced by an intermediate layer of w2v-BERT [17] and enable long-term structural coherence.

Key Innovation [🔗](#)

AudioLM introduces a hierarchical approach to audio generation that achieves both long-term coherence and high audio quality by combining semantic tokens for structure with acoustic tokens for fidelity.

1. Hybrid Tokenization:

- Semantic tokens from w2v-BERT: Capture linguistic/musical content and long-term structure
- Acoustic tokens from SoundStream: Capture audio details for high-quality synthesis

2. Three-Stage Generation Process:

- Semantic modeling: Generate high-level structure
- Coarse acoustic modeling: Generate basic acoustic properties

- Fine acoustic modeling: Add detailed acoustic information

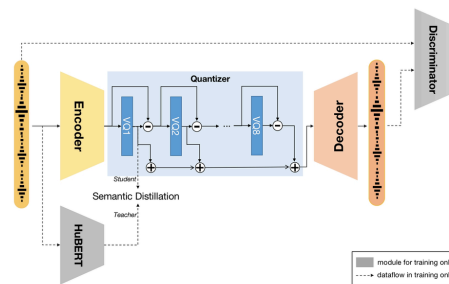
Demonstrated Capabilities [🔗](#)

- Speech Generation: Creates syntactically and semantically coherent speech without text supervision
- Speech Continuation: Continues 3-second prompts while maintaining speaker identity, prosody, and recording conditions
- Piano Continuation: Generates music consistent with a prompt's melody, harmony, and rhythm

Key Results [🔗](#)

- Human evaluators couldn't reliably distinguish AudioLM continuations from real speech
- State-of-the-art performance on lexical (sWUGGY) and syntactic (sBLIMP) understanding tests
- Successful separation of content (semantic tokens) from speaker identity (acoustic tokens)
- Applicable beyond speech, with strong results in piano music generation

AudioLM represents a significant advancement in generative audio models with potential applications across speech, music, and other audio domains.



Zeghidour, Neil, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi.
“SoundStream: An End-to-End Neural Audio Codec.” arXiv, July 7, 2021. [🔗 SoundStream: An End-to-End Neural Audio Codec](#) . [🔗](#)

SoundStream is a groundbreaking neural audio codec that outperforms traditional codecs at significantly lower bitrates across diverse audio types including speech, music, and general audio.

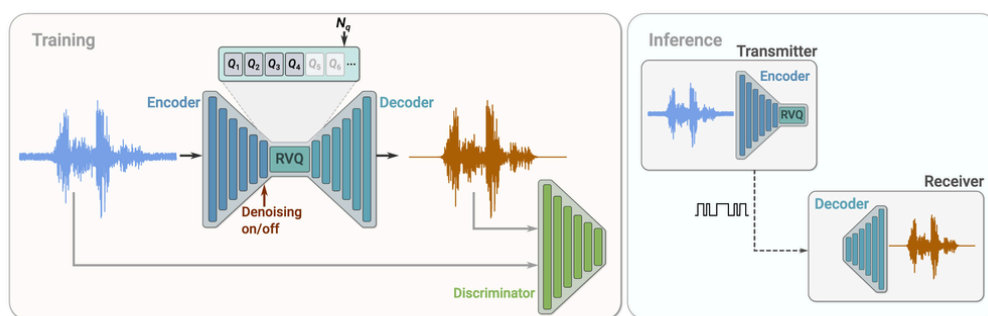


Fig. 2: *SoundStream* model architecture. A convolutional encoder produces a latent representation of the input audio samples, which is quantized using a variable number n_q of residual vector quantizers (RVQ). During training, the model parameters are optimized using a combination of reconstruction and adversarial losses. An optional conditioning input can be used to indicate whether background noise has to be removed from the audio. When deploying the model, the encoder and quantizer on a transmitter client send the compressed bitstream to a receiver client that can then decode the audio signal.

Info [🔗](#)

1. **Architecture:** Consists of a fully convolutional encoder/decoder network and a residual vector quantizer trained jointly end-to-end.
2. **Residual Vector Quantizer:** Enables efficient compression by cascading multiple vector quantizer layers that progressively refine embeddings.

3. **Bitrate Scalability:** Employs "quantizer dropout" during training to create a single model that operates across bitrates (3-18 kbps) with minimal quality loss.
4. **Joint Compression and Enhancement:** Implements Feature-wise Linear Modulation (FiLM) layers to optionally perform denoising without additional latency.

Performance Highlights [🔗](#)

- At 3 kbps, outperforms Opus at 12 kbps and approaches EVS at 9.6 kbps
- Maintains consistent quality across content types (clean speech, noisy speech, music)
- Runs in real-time on smartphone CPUs with low latency
- Encoder-side denoising offers quality improvements and additional bitrate savings

Technical Contributions [🔗](#)

- Demonstrated that learning the encoder (rather than using fixed features) significantly improves quality
- Introduced the first implementation of residual vector quantization in end-to-end neural networks
- Achieved a flexible balance between computational complexity and coding efficiency
- Proved that joint compression and enhancement nearly matches the performance of two separate models while halving computational cost
- SoundStream represents a significant advance in neural audio coding, offering superior compression efficiency and flexibility compared to established codecs, while enabling additional functionality like denoising without compromising performance.

Hsu, Wei-Ning, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed. "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units." arXiv, June 14, 2021. [🔗 HuBERT: Self-Supervised Speech Representation Learning by Masked...](#)

HuBERT (Hidden unit BERT) introduces a novel approach to self-supervised speech representation learning that combines offline clustering with masked prediction. The model predicts discrete cluster assignments of masked speech segments, forcing it to learn both acoustic features and linguistic patterns from continuous audio input.

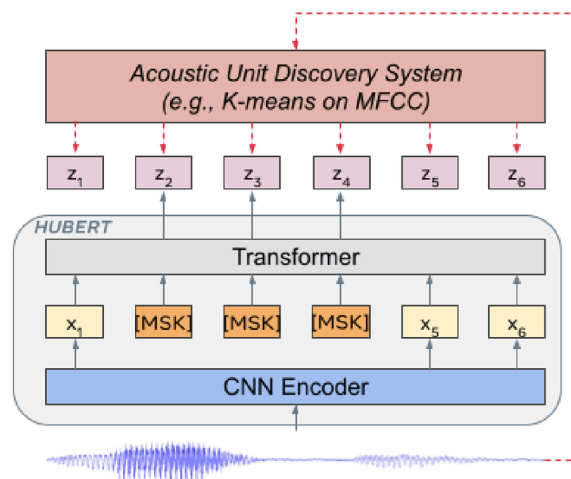


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Model [🔗](#)

Three-component architecture:

1. CNN Encoder: Processes raw waveform into feature sequences

2. Transformer: Processes masked and unmasked features
3. Projection layer: Maps to cluster assignment predictions

Training [↗](#)

- Initial targets: K-means clustering on MFCC features
- Iterative refinement: Each iteration uses features from previous model to create better clusters
- Masked prediction: Model predicts cluster assignments for masked regions only
- Model sizes: BASE (95M parameters), LARGE (317M), X-LARGE (964M)

Key Advantages [↗](#)

1. No reliance on linguistic resources: Works with raw speech without transcriptions
2. Iterative improvement: Representation quality increases with each iteration
3. Robust to low-quality targets: Masked prediction approach works even with simple k-means clustering
4. Scalable: Performance improves with more data and larger models
5. Simplicity: Direct predictive loss avoids complexities of contrastive approaches

Performance [↗](#)

- Low-resource scenarios**: Achieves impressive results with minimal labeled data
- With just 10 minutes of labeled data: 4.6%/6.8% WER on test-clean/test-other
- High-resource scenarios: Competitive with or outperforming state-of-the-art methods
- Scaling benefits: X-LARGE model shows up to 19% and 13% relative WER improvement

Key Findings from Ablations [↗](#)

1. Predicting only masked frames is crucial when using low-quality clusters
2. Cluster ensembles improve performance
3. Optimal masking level is 8% of frames
4. Fewer clusters (100-500) better capture broad phonetic concepts than many fine-grained clusters

Conclusion [↗](#)

- HuBERT demonstrates that combining clustering for target generation with masked prediction creates a powerful approach for self-supervised speech representation learning.
- The method's ability to work well with limited labeled data makes it particularly valuable for industrial applications and low-resource languages.

Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations." arXiv, October 22, 2020. [✖ wav2vec 2.0: A Framework for Self-Supervised Learning of Speech... . ↗](#)

This paper introduces wav2vec 2.0, a self-supervised approach for speech recognition that achieves breakthrough performance, especially in low-resource scenarios.

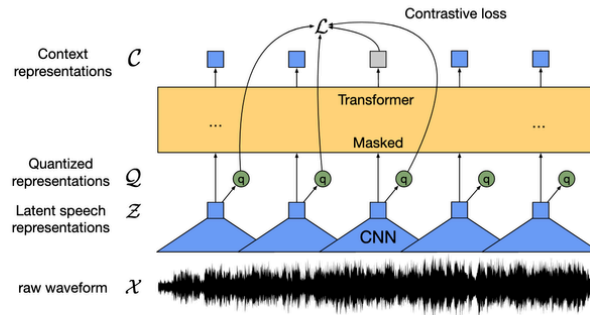


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

Key Innovations: [🔗](#)

1. Self-supervised learning framework that masks latent speech representations and uses a contrastive task over quantized speech units
2. End-to-end architecture combining feature extraction, quantization, and contextualization in a single training objective
3. Joint learning of discrete speech units and contextualized representations

Model [🔗](#)

1. **Feature Encoder:** Multi-layer CNN processes raw audio into latent representations
2. **Quantization Module:** Discretizes representations using Gumbel softmax for differentiable learning
3. **Transformer:** Builds contextualized representations from latent features
4. **Masking Strategy:** Masks spans of time steps (49% of input) to force contextual learning

Training [🔗](#)

1. **Pre-training:** Uses contrastive loss and diversity loss on unlabeled speech data
2. **Fine-tuning:** Adds linear projection layer and trains with CTC loss on labeled data

Results: [🔗](#)

- With just 10 minutes of labeled data: Achieves 4.8/8.2 WER on Librispeech
- With 100 hours of labeled data: Outperforms previous SOTA by 45%/42% relative reduction
- On full Librispeech: Achieves 1.8/3.3 WER on clean/other test sets
- Sets new ****state-of-the-art** on TIMIT phoneme recognition

Key Finding: [🔗](#)

- The combination of continuous inputs with quantized targets performs best, allowing the model to maintain detailed contextual information while learning generalized speech patterns.
- This approach demonstrates that effective speech recognition is feasible with extremely limited labeled data through powerful self-supervised pre-training.