# Improving X-Codec-2.0 for Multi-Lingual Speech: 25 Hz Latent Rate and 24 kHz Sampling

**Husein Zolkepli**[*]

## Abstract

X-Codec-2.0 has shown strong performance in neural audio compression and multilingual speech modeling, operating at a 50 Hz latent rate and a 16 kHz sampling rate using frozen HuBERT features. While effective, this configuration limits efficiency and audio fidelity. In this work, we explore a simple yet effective modification: introducing additional pooling and increasing the decoder hop size. This reduces the latent rate from 50 Hz to 25 Hz and simultaneously raises the output sampling rate from 16 kHz to 24 kHz, improving temporal efficiency and perceptual quality without altering the core architecture. Evaluated on the multilingual Common Voice 17 test set, the proposed configuration achieves a 0.5 MOS improvement over the original X-Codec-2.0 baseline. The source code, checkpoints and generation comparison released at xcodec2-25tps.github.io.

## 1 Introduction

Recent advances in neural audio tokenization have enabled end-to-end speech modeling through discrete latent representations. Among these, X-Codec-2.0 has emerged as a powerful multilingual codec model, combining a HuBERT-based semantic encoder and a transformer-based codec architecture to produce high-quality, language-agnostic audio tokens. Its design allows large language models (LLMs) and autoregressive decoders to handle speech as a discrete sequence prediction task.

However, X-Codec-2.0 operates at a 50 Hz latent frame rate with a 16 kHz target sampling rate. While effective, this configuration limits temporal resolution and upper-frequency fidelity, producing slightly muffled high-frequency content and larger sequence lengths for generation models. Moreover, as multilingual datasets expand to include more diverse acoustic and expressive conditions, the fixed 50 Hz resolution may underutilize the model's potential to capture fine-grained speech variation.

In this work, we propose a simple yet effective modification to X-Codec-2.0 that improves both efficiency and perceptual quality. By increasing the hop size to 960 samples and introducing a lightweight pooling layer before quantization, the codec operates at a reduced 25 Hz latent rate while simultaneously increasing the audio sampling rate to 24 kHz. All encoder components are kept frozen, and only the decoder is fine-tuned under this new temporal configuration. We find that this adjustment produces higher-quality reconstructions without additional parameters or training complexity.

Empirically, the proposed model achieves a +0.5 improvement in mean opinion score (MOS) as estimated by UTMOSv2 on the multilingual Common Voice 22 test set. The improvement is consistent across languages and demonstrates better high-frequency reconstruction and overall perceptual clarity. Our work shows that small architectural refinements, particularly hop-size and pooling adjustments can meaningfully improve codec quality without re-training from scratch.

[*] husein.zol05@gmail.com

Technical Report.

## 2 Method

The proposed modification to X-Codec-2.0 focuses on improving temporal efficiency and output audio fidelity through minimal architectural changes. The overall structure remains consistent with the original design, consisting of a frozen semantic encoder (HuBERT-based), a transformer codec encoder, and a vocoder-style decoder.

### 2.1 Temporal Pooling and Hop Size Adjustment

In the original X-Codec-2.0 configuration, the encoder operates at a latent rate of 50 Hz with a 16 kHz sampling rate, corresponding to a hop size of 320 samples. To achieve a lower latent rate and higher waveform sampling rate, we increased the hop size to 960 samples and introduced an additional average pooling layer:

$$\text{AvgPool1d}(k = 2,\ \text{stride} = 2).$$

This modification reduces the latent rate from 50 Hz to 25 Hz, halving the number of discrete tokens per second while maintaining temporal coherence. The pooling operation is applied before vector quantization, compressing the feature sequence by a factor of two.

### 2.2 Decoder Weight Interpolation

Changing the hop size also alters the dimensionality of the decoder's output layer. Rather than discarding the pretrained decoder weights, we applied one-dimensional linear interpolation to the output projection parameters of the generator head:

$$w'_i = (1 - \alpha_i)\, w_{\lfloor x_i \rfloor} + \alpha_i\, w_{\lceil x_i \rceil}, \quad x_i = \frac{L-1}{L'-1}\, i, \quad \alpha_i = x_i - \lfloor x_i \rfloor, \quad i = 0, \dots, L' - 1, \quad (1)$$

Both the output weight and bias were interpolated to match the new hop size (960 samples). This procedure allows the decoder to retain the spectral characteristics of the original pretrained model while adapting smoothly to the new resolution. Although this interpolation strategy has not been empirically validated in isolation, it provided a straightforward way to transfer pretrained weights without retraining from scratch. We did not conduct ablation experiments with randomly initialized parameters, so it remains unclear whether interpolation contributes to faster convergence or improved stability during fine-tuning.

### 2.3 Training Strategy

All model parameters were frozen except for the decoder. The semantic encoder (frozen HuBERT) and codec encoder were reused directly from the pretrained X-Codec-2.0 checkpoint. The decoder was fine-tuned to accommodate the new hop size and temporal pooling. The resulting model generates 25 Hz discrete tokens and reconstructs 24 kHz audio with improved perceptual quality.

### 2.4 Training Configuration

All hyperparameters were kept consistent with the original X-Codec-2.0 implementation to ensure comparability. The model was trained on two NVIDIA RTX 3090 Ti GPUs using BF16 mixed precision for 3 million steps using a batch size of 20 for each device. Both the generator and discriminator were optimized with the Adam optimizer ($\beta_1 = 0.8, \beta_2 = 0.9$) and a cyclic learning rate schedule with warmup and decay phases:

$$\text{max\_lr} = 1.0 \times 10^{-4}, \quad \text{min\_lr} = 2.0 \times 10^{-5}.$$

The loss function followed the same multi-objective formulation as X-Codec-2.0, combining mel-spectrogram, adversarial, and semantic losses except for feature matching:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{mel}}\mathcal{L}_{\text{mel}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}}.$$

Empirically, the following coefficients were retained from the original configuration: $\lambda_{\text{mel}} = 15$, $\lambda_{\text{adv}} = 1$, $\lambda_{\text{sem}} = 5$.

To align with the new sampling rate, the `MultiResolutionMelSpectrogramLoss` was configured for a 24 kHz sample rate. This ensures that spectral loss computation matches the decoder's output frequency resolution. Validation was performed every 4000 steps on 2560 held-out samples. Gradient clipping was set to 1.0 for both generator and discriminator to maintain stability. The overall setup reproduces the X-Codec-2.0 training dynamics while isolating the effect of the proposed hop-size and pooling modifications.

# 3 Experiments

**Datasets.** We train our model on a large-scale multilingual corpus totaling approximately 16,000 hours of speech.

The dataset is constructed by combining publicly available TTS and expressive speech datasets from Hugging Face, covering over 20 languages, including English, Mandarin, Malay, Japanese, Korean, Arabic, Hindi, Tamil, Bengali, and several European languages. Each audio sample is uniformly resampled to 24 kHz and cropped into randomly selected 5-second segments. The data mixture includes both neutral and expressive speaking styles, although most samples are clean studio-quality recordings. No text transcriptions are used during training.

**Model Initialization.** We initialize from the official X-Codec-2.0 checkpoint provided by HKUSTAudio. All encoder components (HuBERT-based semantic encoder and codec encoder) are frozen. Only the decoder is fine-tuned under the modified hop size (960 samples) and temporal rate (25 Hz). The decoder head weights and biases are linearly interpolated from the original X-Codec-2.0 configuration to match the new output resolution.

**Training Configuration.** Training is conducted on two NVIDIA RTX 3090 Ti GPUs using mixed BF16 precision for approximately two months. We use the same hyperparameters and optimization strategy as X-Codec-2.0, including identical loss weightings and scheduler configurations. The `MultiResolutionMelSpectrogramLoss` is updated for a 24 kHz target sample rate to align with the new output resolution. Batch accumulation and gradient clipping follow the original setup.

**Evaluation.** For evaluation, we benchmark against the original X-Codec-2.0 model (16 kHz, 50 Hz latent rate). All evaluations are performed using the UTMOSv2 model-based perceptual quality metric on the multilingual Common Voice 22 test set.

## 3.1 Evaluation with UTMOSv2 Across Checkpoints

.

# 4 Results

.

# 5 Limitation

While the proposed modification improves the perceptual quality and efficiency of X-Codec-2.0, several limitations remain. First, the training data used in this study is primarily drawn from the Common Voice dataset, which is relatively clean and contains limited variation in background noise, speaker style, and expressiveness. As a result, the model does not generalize well to unseen languages or expressive speech domains such as animated or emotional voices. Continued fine-tuning with more diverse and expressive data is expected to alleviate this issue.

Second, all evaluations were conducted using the UTMOSv2 metric, which estimates mean opinion scores from audio features. While convenient for large-scale automatic evaluation, UTMOSv2 does not fully capture human perceptual preferences.

Finally, no downstream applications have been explored. With a vocabulary size of 65,536 and a 25 Hz token rate, each discrete token represents a larger amount of information. This increases the effective prediction difficulty for autoregressive models that use these tokens, which may lead to higher perplexity.

## 6 Future Work

While this work focuses on a simple temporal modification introducing additional pooling and increasing the encoder hop size to reduce the latent rate from 50 Hz to 25 Hz in X-Codec-2.0 several directions remain to deepen the understanding and extend the capability of discrete audio representation models.

First, the semantic encoder in X-Codec-2.0, derived from a frozen HuBERT model, is known to capture cross-lingual acoustic and phonetic information effectively. However, its interaction with temporal resolution remains underexplored. Future work could systematically vary the latent frame rate (e.g., from 10 Hz to 100 Hz) and evaluate its impact on perceptual quality, token predictability, and downstream generative modeling. Such experiments would help characterize the encoder's temporal sensitivity and identify the optimal trade-off between information density and reconstruction quality. In particular, understanding how much semantic content is preserved when the latent rate decreases could inform the design of more compact and efficient tokenizers for both speech and general audio.

Second, reducing the latent rate introduces a natural compression bottleneck, potentially limiting the decoder's ability to recover fine-grained acoustic details. This raises an important design question regarding the balance between encoder compression strength and decoder capacity. Increasing the decoder depth, widening its receptive field, or incorporating attention-based upsampling mechanisms may help compensate for lost temporal detail when operating at lower frame rates. A systematic analysis of this trade-off to evaluate how decoder complexity scales with latent rate could provide new insights into optimal model scaling laws for discrete audio codecs.

Third, extending the experiments to larger and more diverse datasets, including noisy or expressive speech (e.g., emotional, conversational, or singing voice data), would reveal the model's robustness and generalization ability. The current results, based on Common Voice 16, primarily reflect performance on clean and monotonic speech. Future work could investigate whether adjusting the latent frame rate interacts differently with noisy or highly variable input conditions.

Finally, downstream evaluation remains an open area. One promising direction is to assess how the discrete tokens perform in text-to-speech (TTS) or speech-language modeling pipelines, especially under large vocabulary sizes (e.g., 65,536 tokens) and shorter temporal sequences. Analyzing token-level perplexity, attention behavior, and reconstruction fidelity in LLM-based TTS finetuning could help determine whether lower frame rates hinder or help generative quality. Integrating such codecs with speech foundation models may also reveal new ways to unify speech and text modeling under a single discrete token interface.

## 7 Acknowledgement

Special thanks to my wife for her patience and for not getting too upset about the electricity bill caused by running two RTX 3090 Ti GPUs around the clock for 45 days.

## References