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

2 Method

The proposed modification to X-Codec-2.0 focuses on improving temporal efficiency and output audio fidelity with minimal architectural changes. The overall model 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 simple modification effectively halves the number of discrete tokens per second, reducing the latent rate from 50 Hz to 25 Hz, while maintaining stable training dynamics and enabling the decoder to reconstruct 24 kHz audio.

2.2 Training Strategy

All parameters except the decoder were frozen during training. The semantic encoder (frozen HuBERT) and codec encoder components were reused directly from the official X-Codec-2.0 release, ensuring that multilingual representation capability was preserved. Only the decoder was fine-tuned to adapt to the new temporal resolution and higher sampling rate.

[*] husein.zol05@gmail.com

2.3 Implementation Details

The architecture was implemented using the Hugging Face `transformers` and `torch` frameworks. A simplified version of the modification is shown below:

```
self.avg_pooler = nn.AvgPool1d(2, stride=2)
...
concat_emb = self.fc_prior(concat_emb.transpose(1, 2)).transpose(1, 2)
concat_emb = self.avg_pooler(concat_emb)
```

This pooling operation compresses the feature sequence length by half before vector quantization and decoding. The overall computation graph remains identical to the original X-Codec-2.0 pipeline.

2.4 Resulting Configuration

After modification, the model produces discrete tokens at a 25 Hz rate and reconstructs waveforms at a 24 kHz sampling rate, offering improved perceptual quality and efficiency without altering the core model design.

3 Experiments

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 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 two months.

References