TF-EnCodec: High Fidelity Neural Audio Compression using Time-Frequency Representation

Nathanaël Perraudin

Student project - Fall 2023

Figure 1: Two representations of the same sound. Left: waveform. Right: STFT magnitude and phase derivatives. From the magnitude, recovering the time signal is often difficult. We believe that reconstructing the phase derivatives in a first step can significantly improve the quality of the recovered waveform or reduce the computation burden.

1 Abstract

Despite impressive recent advances with neural networks [11, 6, 5, 8], generating the waveform in audio synthesis remains a challenge in machine learning. As a result, recent works [7, 3] typically use specific encoders such as EnCodec [4] to synthesize sound and avoid generating the waveform. Training audio encoders, however, is challenging due to the oscillatory and multiscale nature of sound, making it data and computation-heavy. In this project, we investigate an alternative technique to build an encoder that requires less memory and computation.

Instead of generating the waveform directly, we propose generating a Short Time Fourier Transform (STFT) representation of the waveform. The STFT magnitude is a much more stable sound representation that does not oscillate in time. However, previous attempts to produce high-quality audio using the STFT often failed because of the difficulty to recover the phase from the STFT magnitude. Therefore, additionally to the magnitude, we suggest
producing the phase gradients \[1,2\] with the decoder. Then, the waveform can be recovered using phase integration and STFT inversion \[10\]. Since EnCodec uses a discriminator based on the STFT, we believe the computation will be significantly reduced compared to the original EnCodec, making it possible to train with a smaller dataset and less compute while still achieving similar quality.

### 2 Details of the project

Concretely, the main goal of this project is to build an audio encoder similar to \[4\] that works with the STFT magnitude and phase gradient representation. Since the code for EnCodec \[4\] is available on Github\[1\], we will start from this code and modify it to work with the STFT magnitude and phase gradient.

### 3 Student tasks

**Stage 1: Setting up data pipeline**
- Implement a data pipeline for the STFT magnitude and phase gradient
- Test the pipeline to ensure inversion is working

**Stage 2: Adaptation of EnCodec**
- Adapt the EnCodec codebase to work with the STFT magnitude and phase gradient
- Compare the efficiency with the original EnCodec

**Stage 3: Large Scale Training (if possible)**
- Build a large-scale audio dataset
- Train the model on the large dataset

**Stage 4: Reporting**
- Write a report
- Perform a project presentation

### 4 Additional information

**What will you learn?**  Numerical Analysis, Deep Learning, Generative Modeling, PyTorch, WandB, Pytorch Lightning, Time-Frequency Analysis, Audio Synthesis.

**Requirements:**  Machine Learning fundamentals, linear algebra, *good Python level*, experience with Git.

\[1\]https://github.com/facebookresearch/encodec
Is a publication possible? Yes, if the project goes well.

What should I read if I want to know more? You can start with TifGAN [4, 10, 9].

References


