JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

MusicalDL: Choral Voice Synthesizer

William David, Sophia Doerr, David Samson

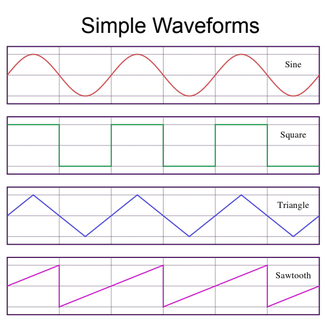
Motivation

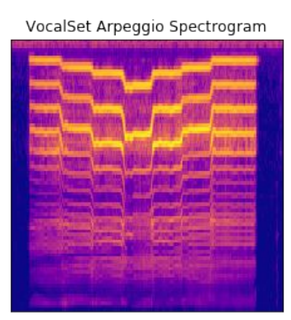
In the field of music, composers make use of previewing tools to hear what their music will sound like as they create it. For choral composers specifically, these tools are inadequate as they cannot capture both the complex timbre of the human voice, as well as its unique variability, namely the capacity to produce words.

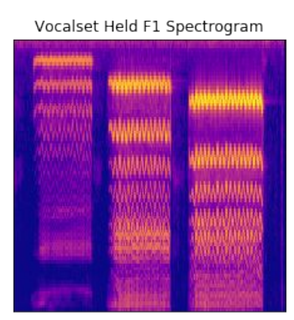
With this motivation in mind, the goal of the project is to utilize deep learning techniques to develop a realistic sounding voice synthesizer that can sing
We decide to explore two deep learning approaches:
1) Conditional WaveNet Vocoder network
2) GANSynth network

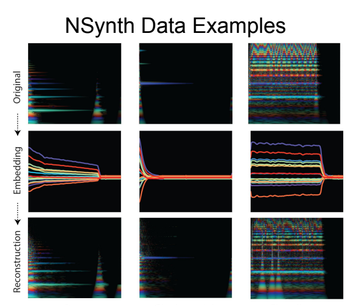
Data

Basic Waveforms: Sine, square, triangle, and sawtooth waves generated to train WaveNet initially
NSynth (Neural Synthesizer): Dataset of ~300,000 samples of annotated music notes from ~100 instruments
VocalSet: Dataset of 10.1 hours of professional singing; 9 male and 11 female vocalists
Self-supervision: Melodia pitch detector was used to label VocalSet with pitch for network conditioning









WaveNet Vocoder

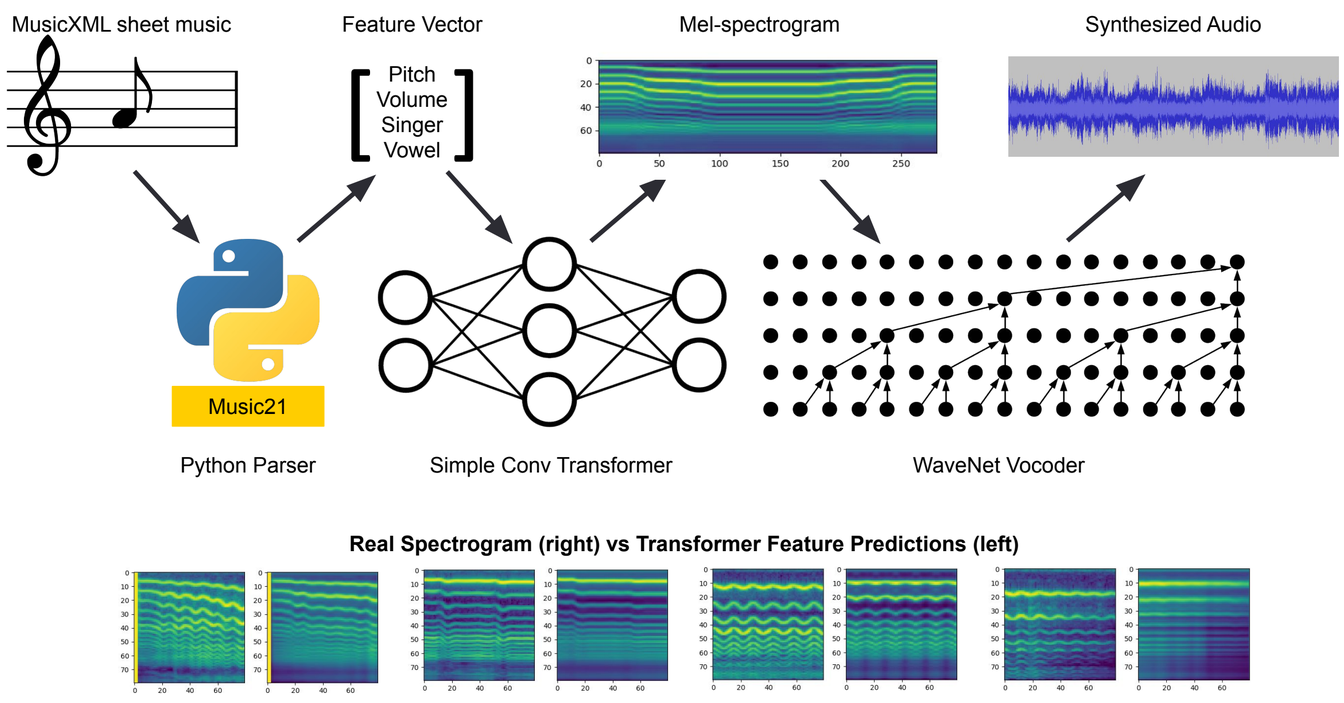
Two WaveNet approaches are commonly utilized:
1) Direct feature to waveform generation
2) Feature to mel-spectrogram via a separate transformer and then WaveNet as a vocoder to generate final output waveform

First approach typically requires much more data to succeed due to increased complexity in task

Experiments:

First approach: Direct generation failed during proof of concept with basic waveforms. Most likely due to not enough network capacity.

Second approach: Similar WaveNet architecture combined with a simple convolutional transformer model. This approach succeeded in principle, demonstrating the ability to synthesize singers voices. Main issues are transformer failure to adequately model pitch (singers are out of tune), and general noise mostly due to minimal training time for the WaveNet



GANSynth

Generative Adversarial Networks (GANs) are successful at:
1) Global latent control
2) Parallel sampling

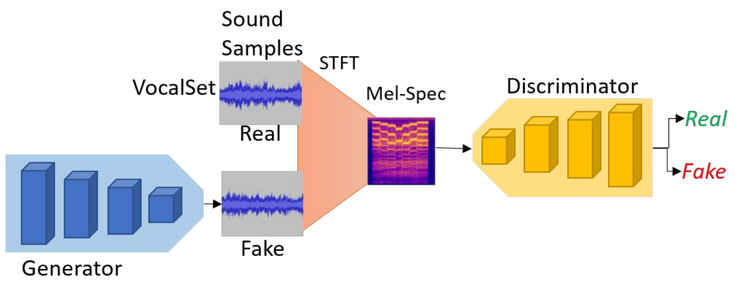
Instantaneous Angular Frequency: Prevents complexities of phase precession in learning high fidelity audio

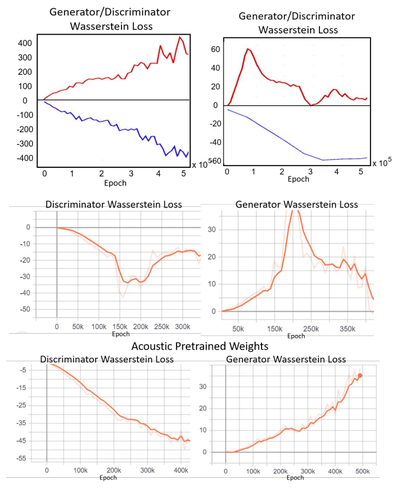
Experiments:

Pretrained vs not pretrained: Pretraining on Nsynth acoustic dataset was compared to no pretraining for the model
Learning rate: $l = \{5 \times 10^{-3}, 5 \times 10^{-4}, 5 \times 10^{-5}, 8 \times 10^{-5}, 5 \times 10^{-6}\}$
Batch Size: $b_s = \{16, 30, 61\}$
Progressive vs. non-progressive training

GANsynth optimization proved tricky in all cases. Wasserstein loss was used however, the discriminator in all cases became too strong. More tuning is needed to determine proper hyperparameters for the dataset. Generated examples show poor pitch and timbre.

GANSynth Architecture





INTUITIVE SURGICAL®