

GCT634/AI613: Musical Applications of Machine Learning

Neural Audio Synthesis



Juhan Nam

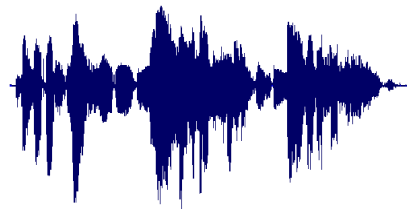
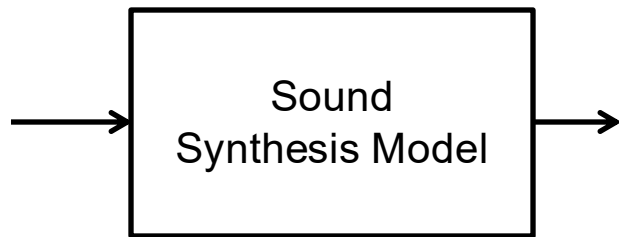
Sound Synthesis

- Generate sound from score-level input such as MIDI
 - Music Synthesizers



Note (pitch, velocity, duration)

Control (wheel, pedal, slides, LFO...)



Traditional Sound Synthesis

- Abstract sound synthesis
- Sample-based synthesis
- Physical modeling

Abstract Sound Synthesis

- Use “digital” (or band-limited) oscillators and modifiers (digital filters)
 - Highly parametric and programmable
 - Subtractive: harmonic oscillator (sawtooth, square) + filter
 - Modulation: frequency modulation
 - Distortion: sine + distortion
 - Additive: a collection of sine oscillations



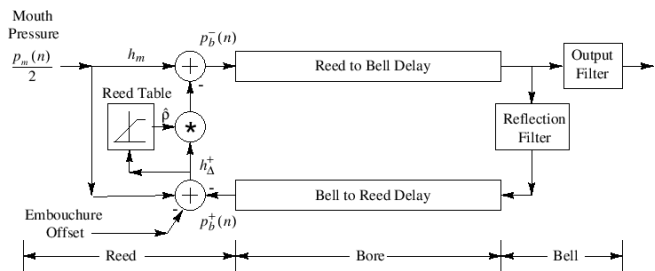
Sample-based Synthesis

- Use recorded samples
 - Require large memory
 - Natural tones but not flexible: pitch shifting by resampling
 - Granular Synthesis
 - Concatenative synthesis: e.g. singing voice synthesis



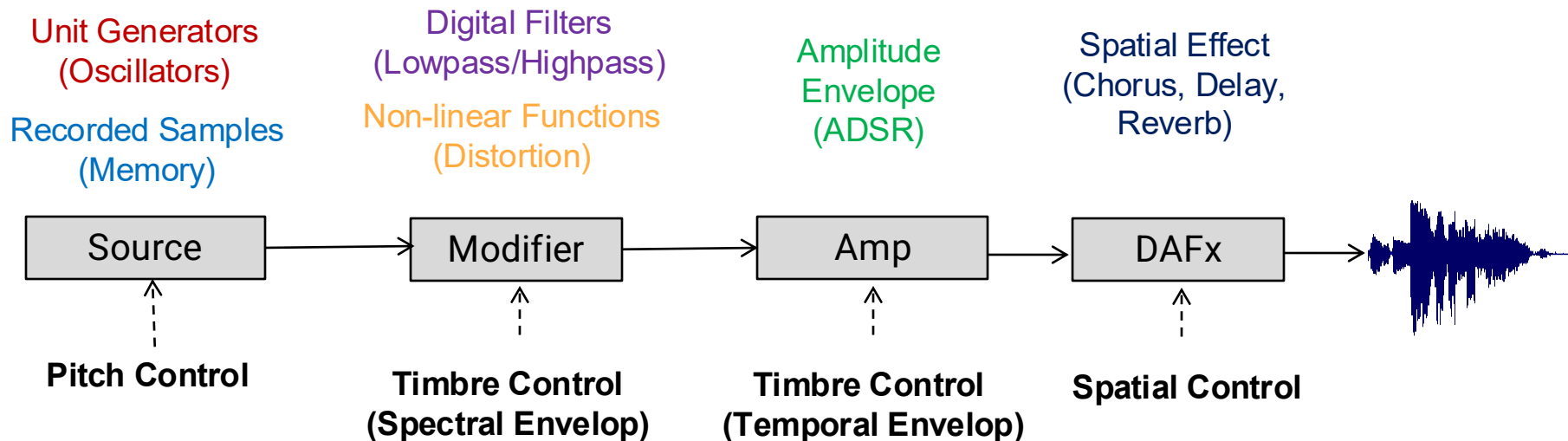
Physical Modeling Synthesis

- Imitate the physical phenomenon of vibrating objects
 - Numerical modeling of wave equations: e.g. finite difference
 - Digital waveguide: efficient model based on delaylines
 - Physically interpretable parameters



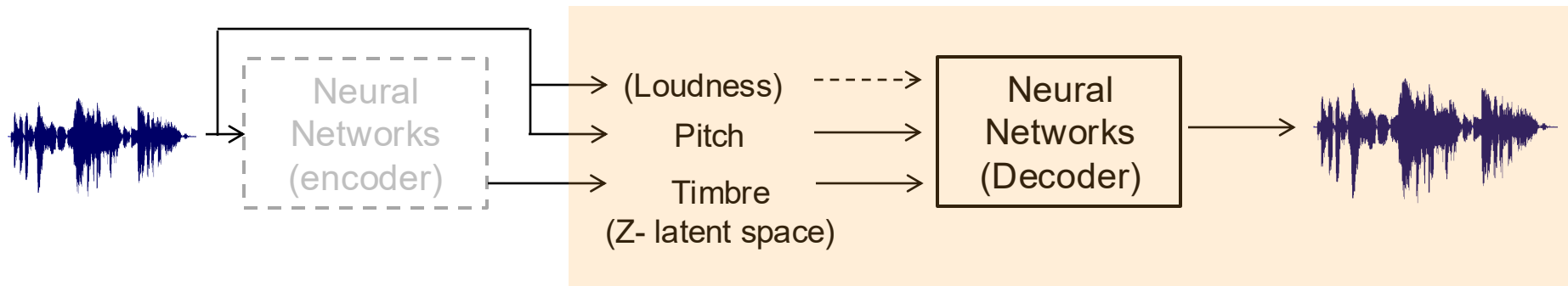
Tone Generation on Computer

- General framework



Neural Music synthesizer

- Input: pitch (note) and timbre (instrument)

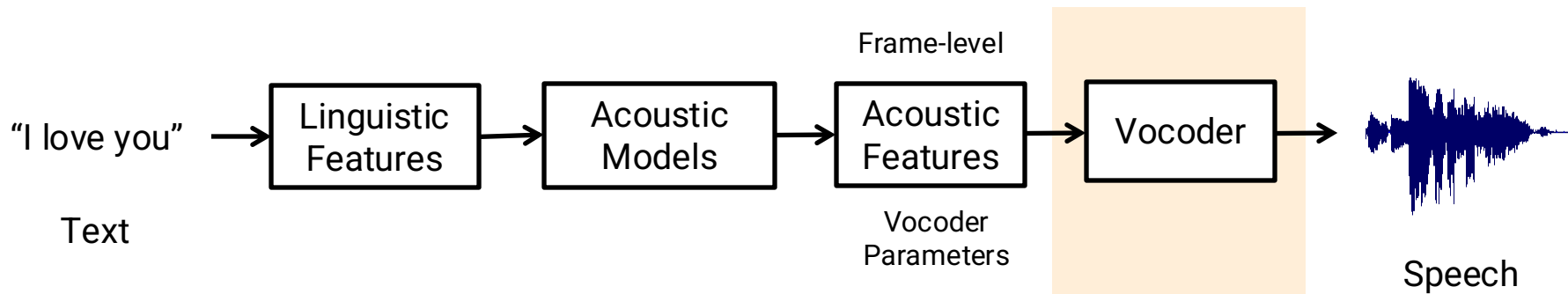


- Models

- WaveNet / NSynth
- WaveGAN
- GANSynth
- DDSP

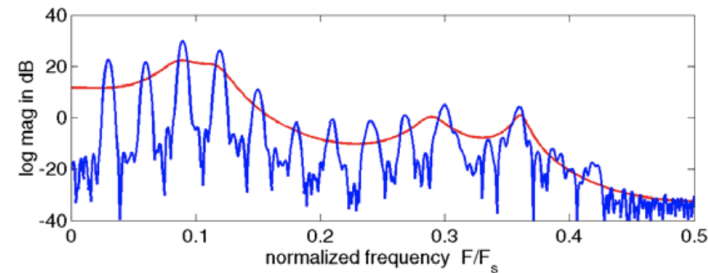
Text-To-Speech (TTS)

- Traditional TTS model
 - The acoustic model: generates frame-level acoustic features from text
 - Rule-based, HMM, RNN
 - The vocoder: synthesizes speech waveforms from the acoustic features
 - Acoustic features: F0, Voice/Unvoice, Formant (e.g., LPC)
 - WOLRD, STRAIGHT

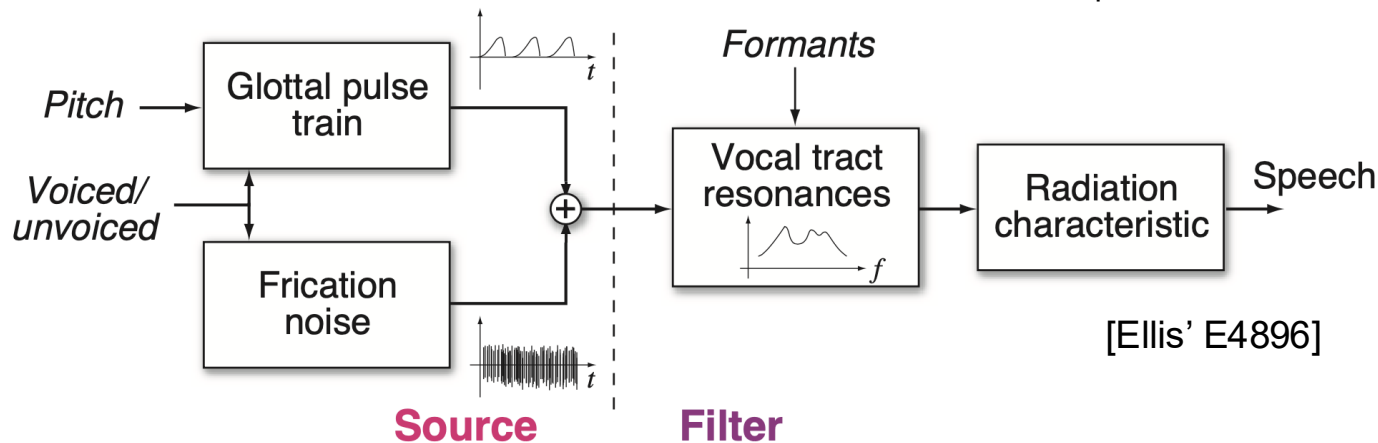


Vocoder

- Source-filter model
 - Source: oscillator or noise
 - Filters: shape the formant



Spectrum and format of a vowel sound

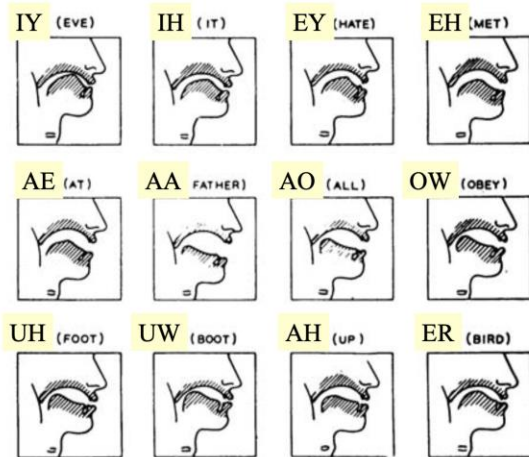


[Ellis' E4896]

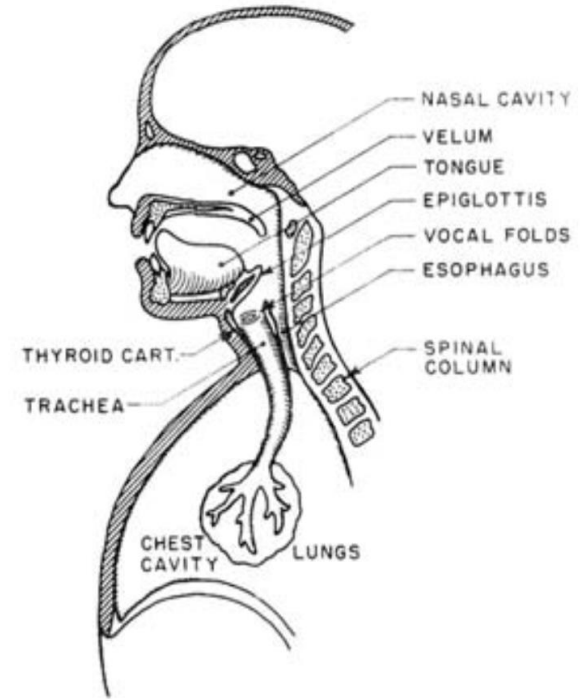
Fun example: <https://artsandculture.google.com/experiment/blob-opera/AAHWrq360NcGbw?hl=en>

Speech production

- Vocal cords (Source): oscillation of air flow
- Vocal tract (Filter): air pathway to the mouth
 - Throat + tongue + lips
 - Changes to pronounce different vowel sounds
 - Resonances at different frequencies



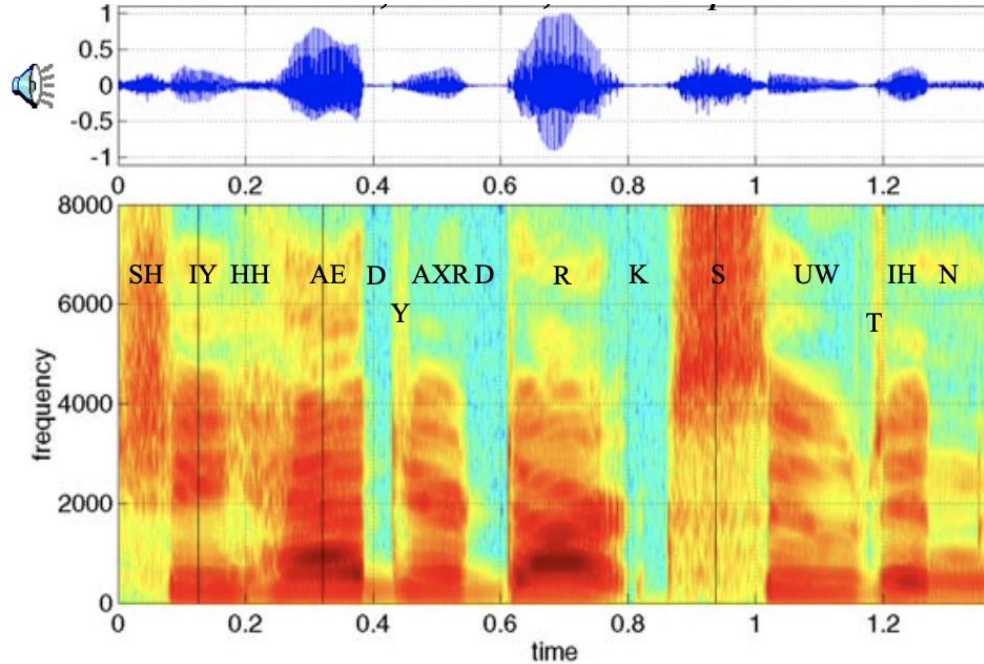
From Flanagan



From Rabiner and Juang

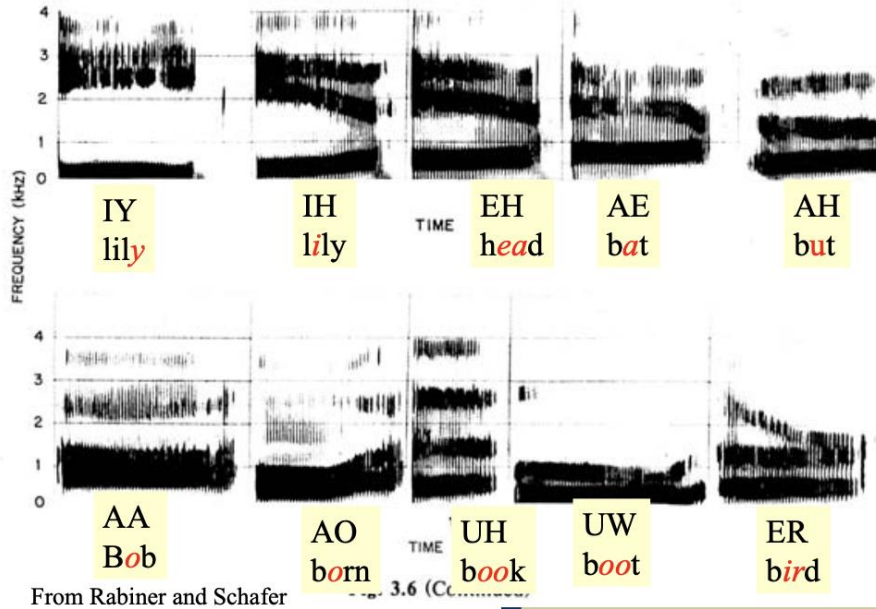
Speech Waveform

- Consonant sounds: soft and noisy waveform
- Vowel sounds: loud and periodic waveform

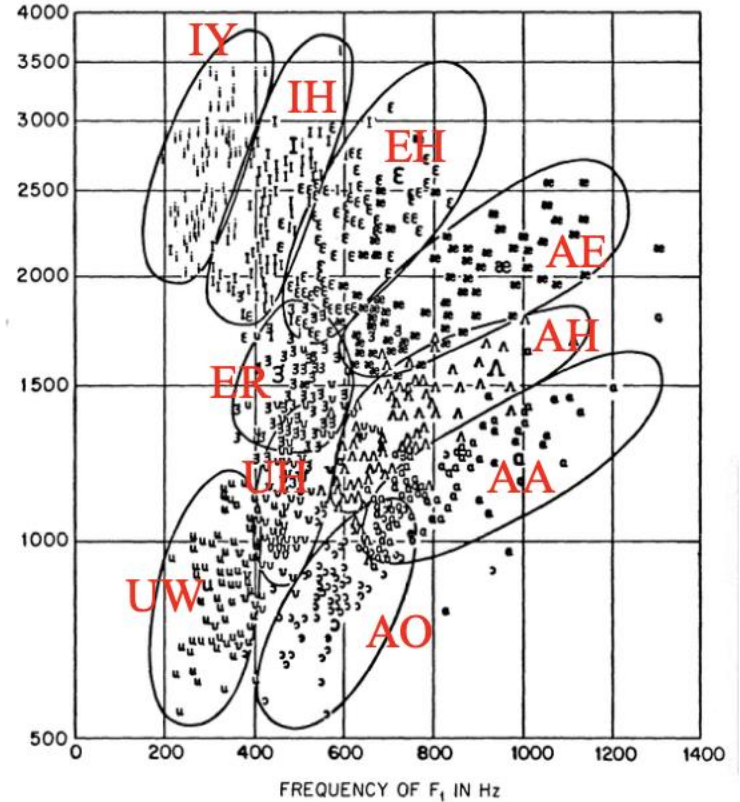


Speech Waveform

- Vowel and Formant

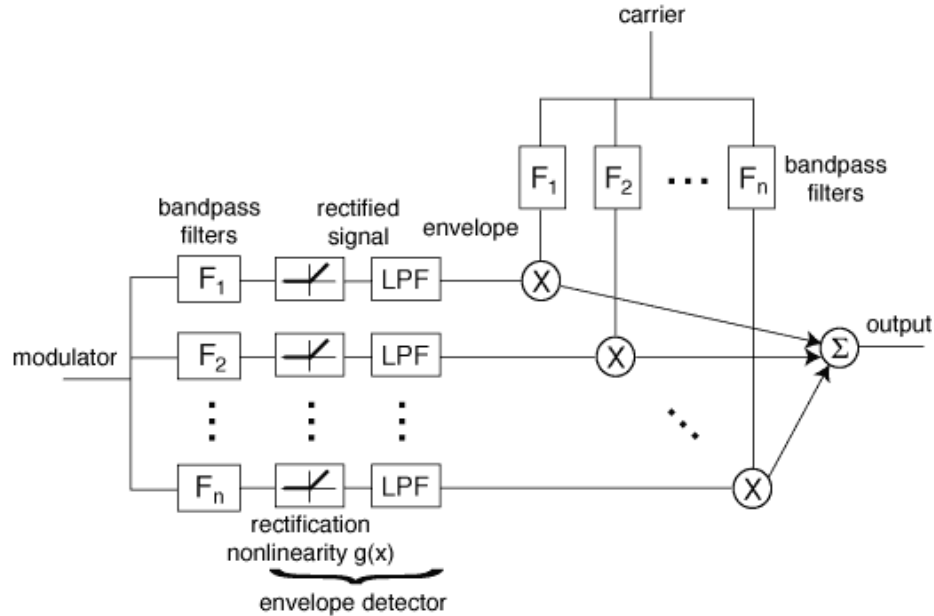


From Rabiner and Schafer



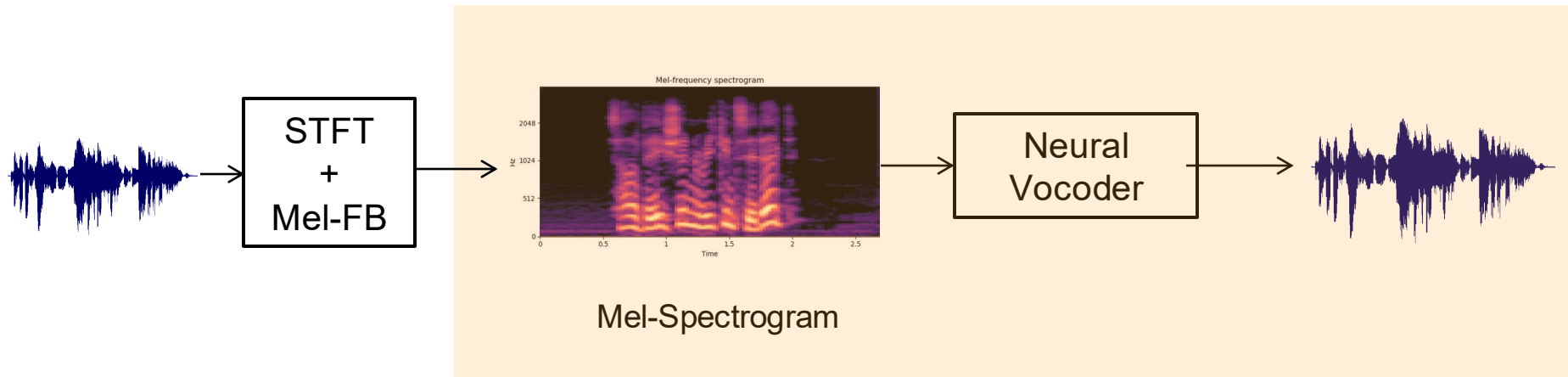
Channel Vocoder

- Extract formants using a filterbank and use them to modulate a wideband carrier signal

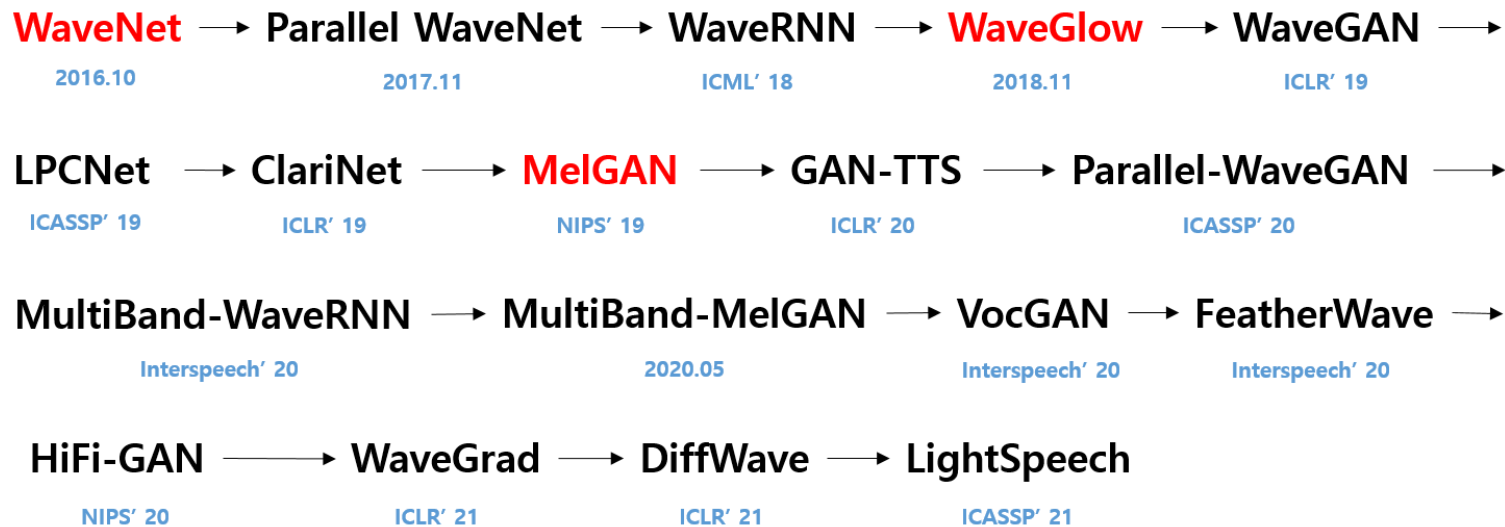


Neural Vocoder

- Input: mel-spectrogram, compressed speech vectors

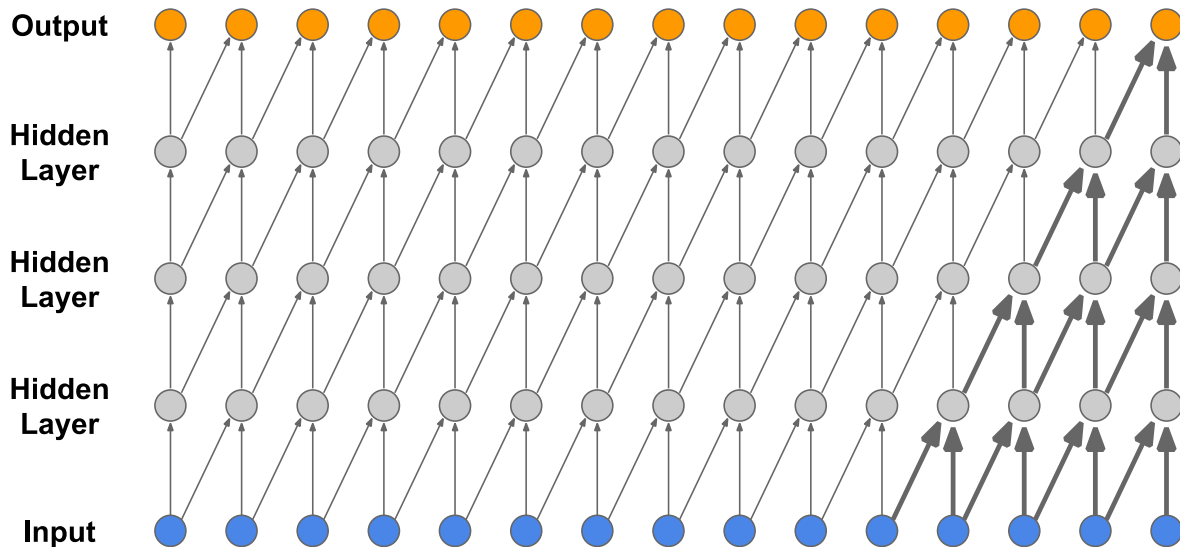


Neural Vocoder



WaveNet

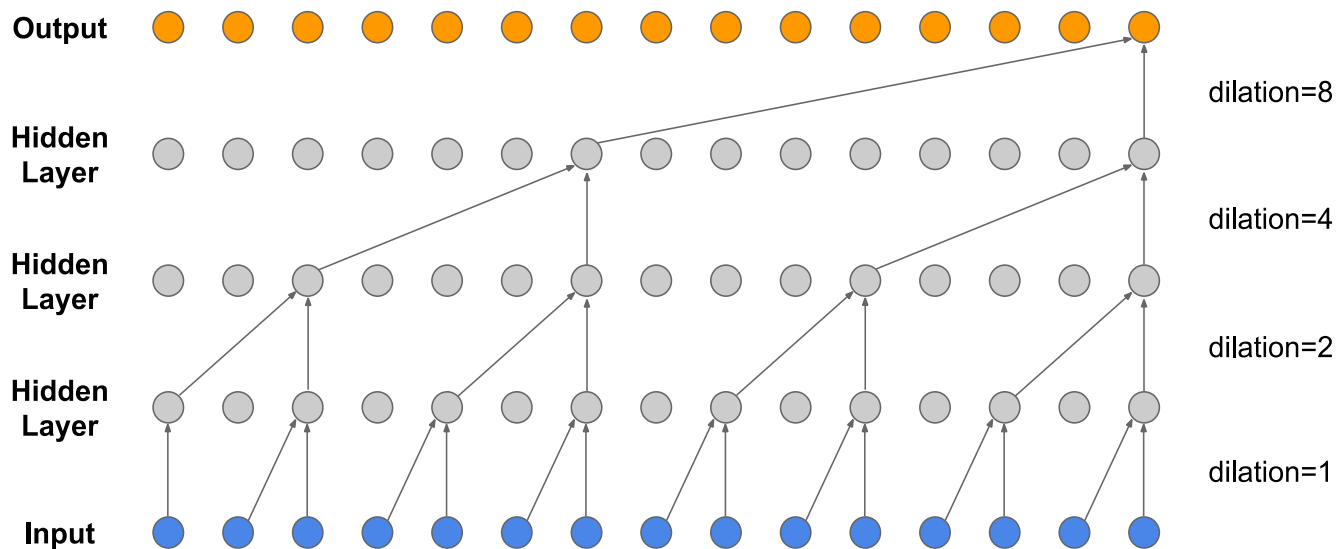
- CNN with causal prediction (no pooling)
 - Note that CNN is faster than RNN to train the model
 - But, it requires many layers to increase the receptive field



WaveNet

- Dilated Convolution

- Dilation increases the receptive field: learn long-term dependency better
- A reminiscent of FFT



WaveNet Architecture

- Multi-class classification with the softmax output
 - Choose one of 256 possible sample values with “ μ -law” 8-bit quantization
 - Better reconstruction quality than linear quantization
 - Gated activation units: better results than ReLU

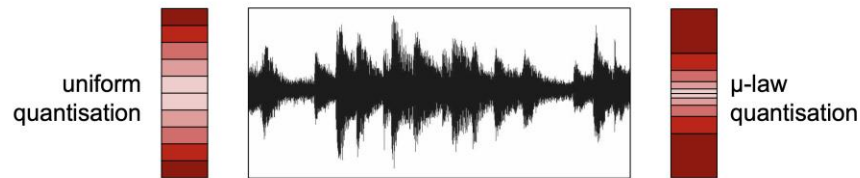
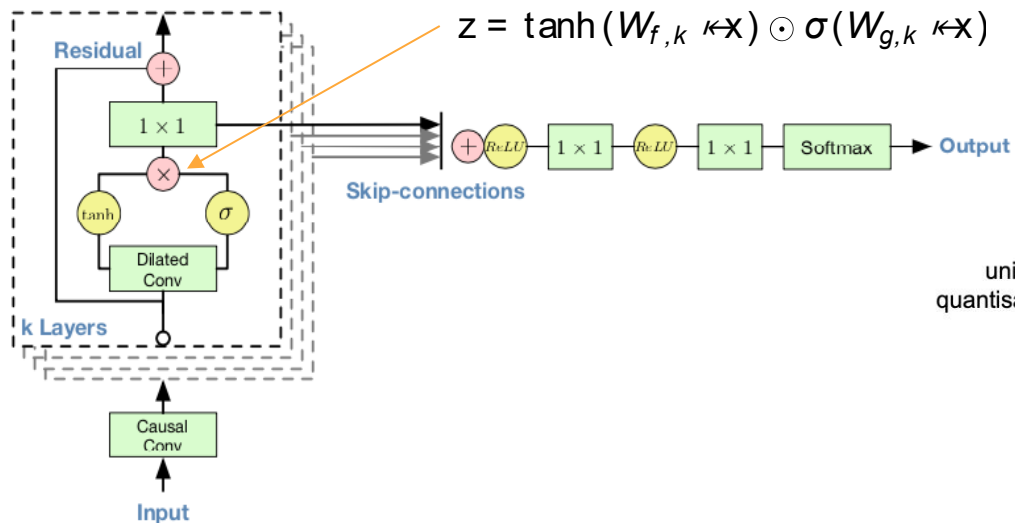


Image source: Sander Dieleman

Conditional WaveNet


- Adding condition \mathbf{h}

$$p(\mathbf{x} | \mathbf{h}) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1}, \mathbf{h}).$$

- Global condition: e.g., speaker (speaker-dependent generation)

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k}^T \mathbf{h}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k}^T \mathbf{h})$$

- Local condition: linguistic features (text-to-speech generation)
 - Upsample them to have a frame-rate sequence using transposed convolution

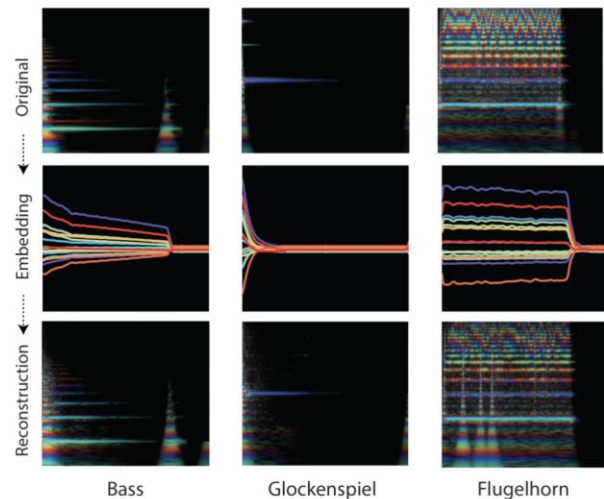
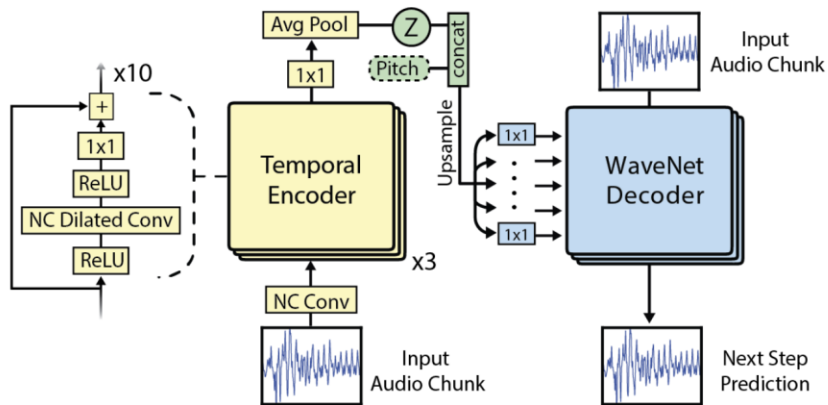
$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y}) \quad \mathbf{y} = f(\mathbf{h})$$


Audio Generation Examples

- Speech synthesis and piano sound synthesis
 - <https://deepmind.com/blog/wavenet-generative-model-raw-audio/>

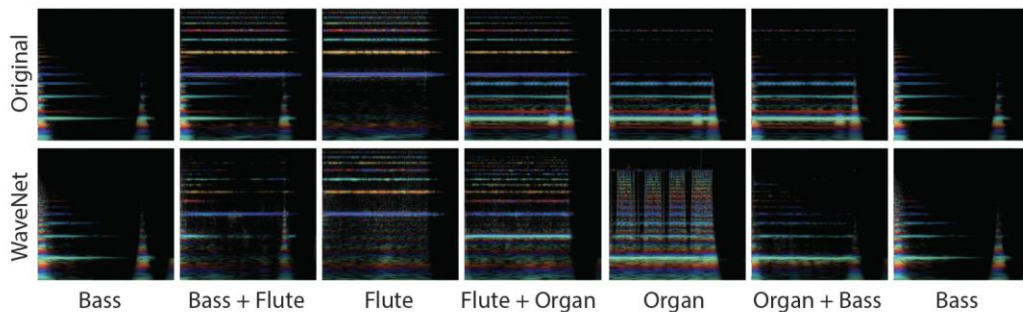
NSynth

- Neural audio synthesis using a WaveNet-style auto-encoder
 - Encoder: compress the timbre context into a latent vector z
 - Temporal embedding of 16 dimensions for every 512 samples
 - Decoder: conditioned by the latent vector z + pitch embedding



NSynth

- Timbre change
 - Linear interpolation between two instruments on the latent vector
 - Latent vector modulation



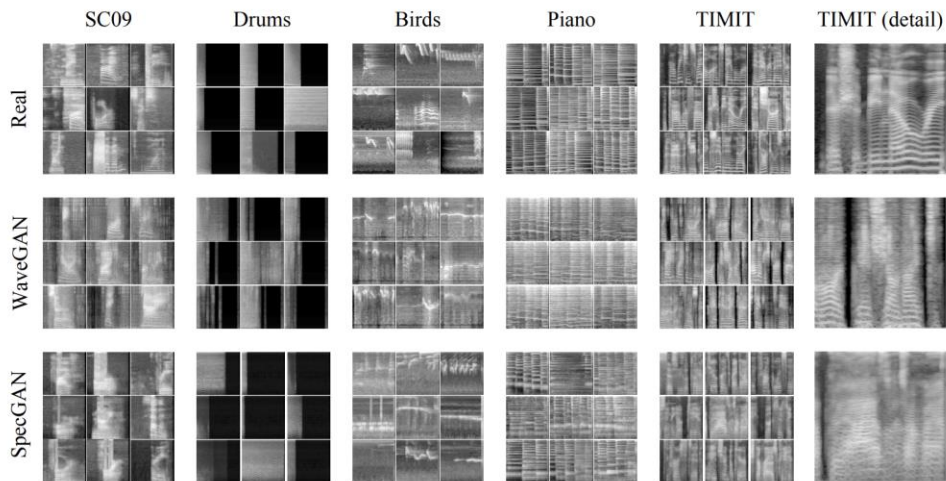
- Demos
 - <https://magenta.tensorflow.org/nsynth>
 - <https://magenta.tensorflow.org/nsynth-instrument>

Limitations of WaveNet

- Generation speed is very slow
 - Predicting a single sample at a time in an autoregressive way is very slow
- Lack global latent structure
 - Use a time-distributed latent vector instead of a single latent vector

WaveGAN

- Generate one second of raw waveforms using GAN
 - Compare the proposed model with the spectrogram-generating GANs
 - Use a modified DCGAN built with transposed convolution layers
 - Flatten 2D 5x5 filters into 1D 25 filters, increase the striding size: 2 → 4
 - Wasserstein loss

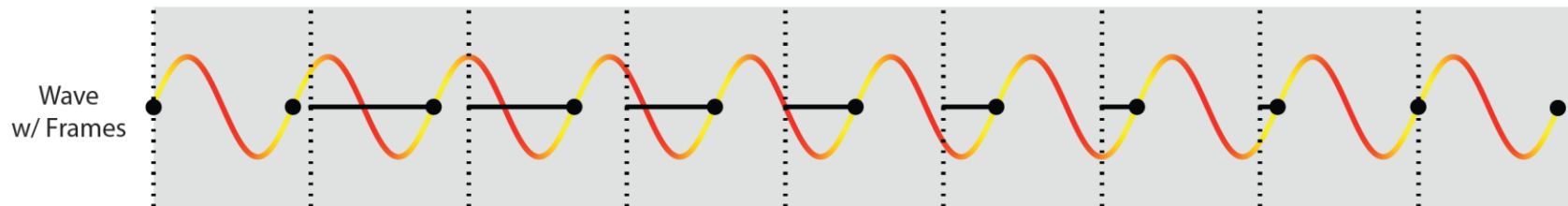


https://chrisdonahue.com/wavegan_examples/

<https://chrisdonahue.com/wavegan/>

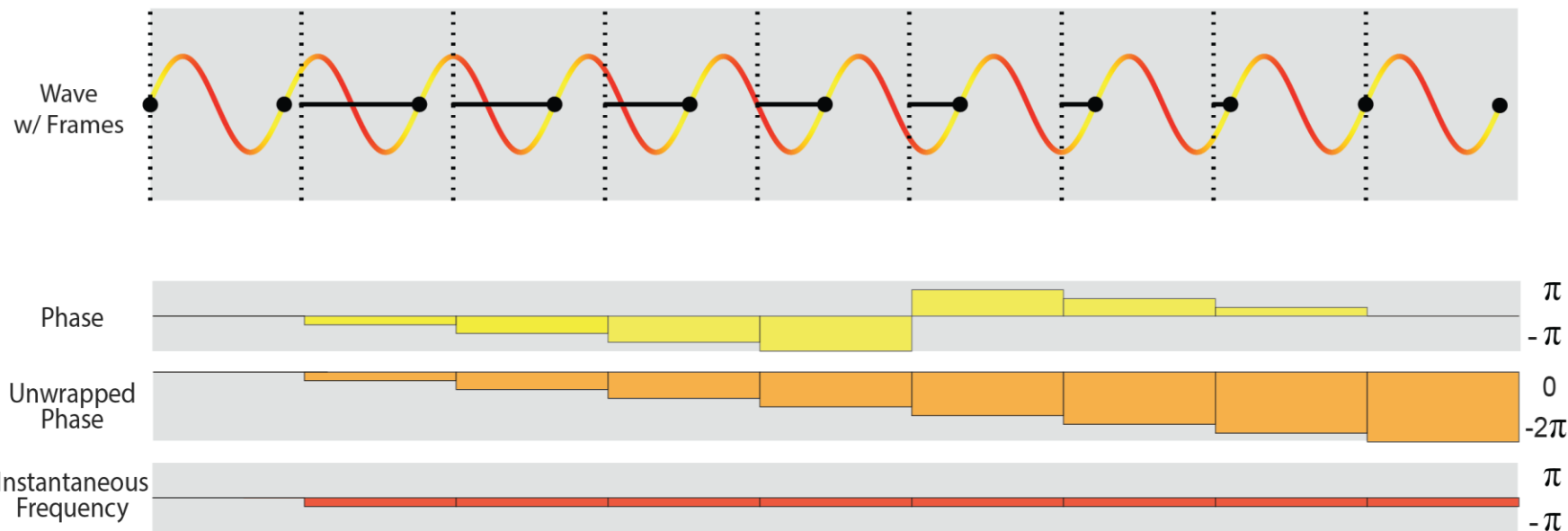
Issues in WaveGAN

- A limitation in generating locally-coherent waveforms
 - Upsampling convolution struggles with phase alignment for highly periodic signals
 - The filters in the upsampling conv layers should learn all phase variations within the frame



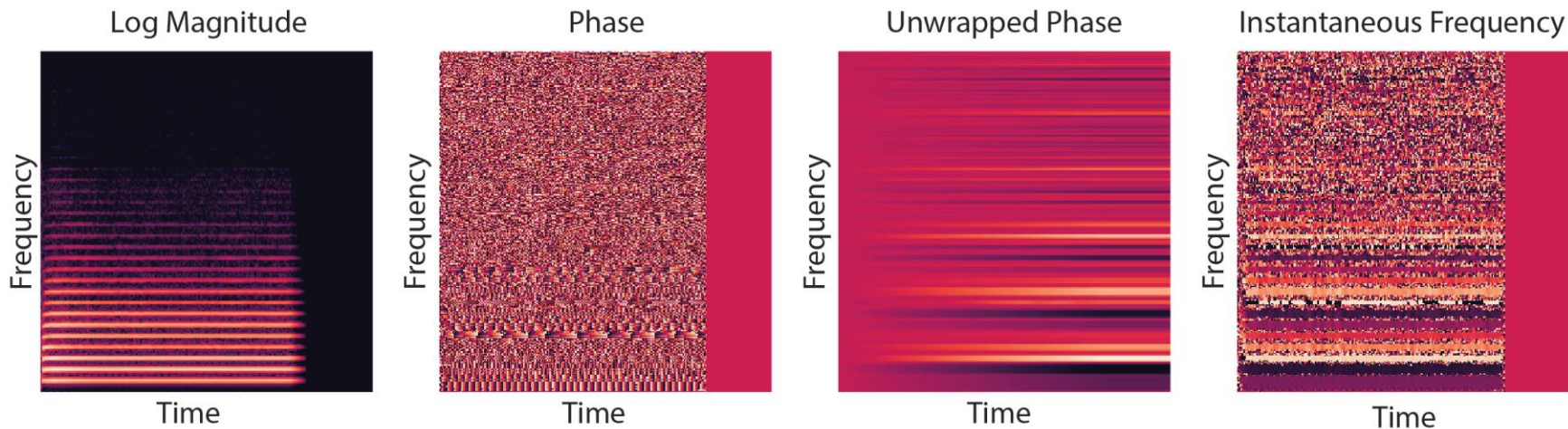
GANSynth

- Generate both spectrogram and instantaneous frequency
 - Use progressive GAN: generate from low resolution to high resolution



GANSynth

- Instantaneous frequency has consistent lines reflecting the coherent periodicity of the underlying sound

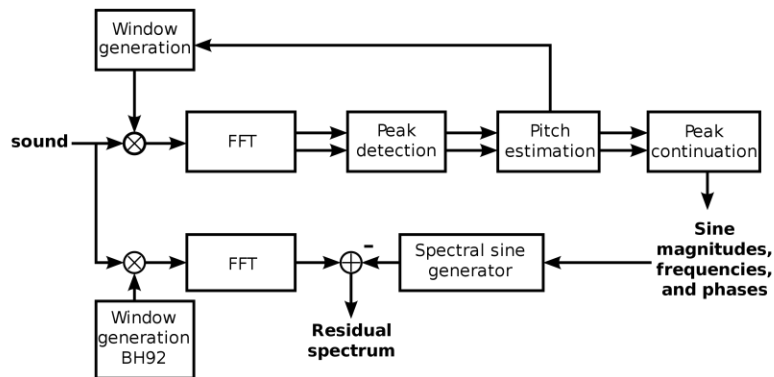


GANSynth

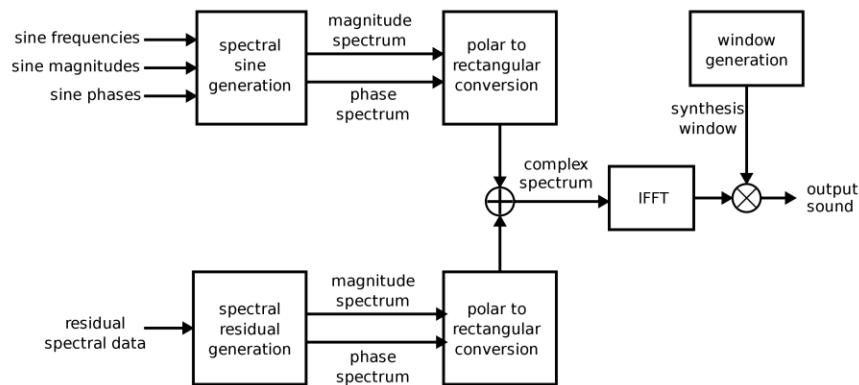
- Generation speed is significantly faster than real-time on a GPU and 50000 times faster than a standard WaveNet
- Increasing the frequency resolution of STFT improved the performance
- Demo:
 - <https://magenta.tensorflow.org/gansynth>

Traditional Analysis-Resynthesis

- Spectral modeling synthesis (SMS)
 - Encoder: decompose the sound into sinusoids and a residual (sine + noise) using STFT and pitch estimation
 - Decoder: synthesis the original source using sinusoidal oscillators and filtered noise



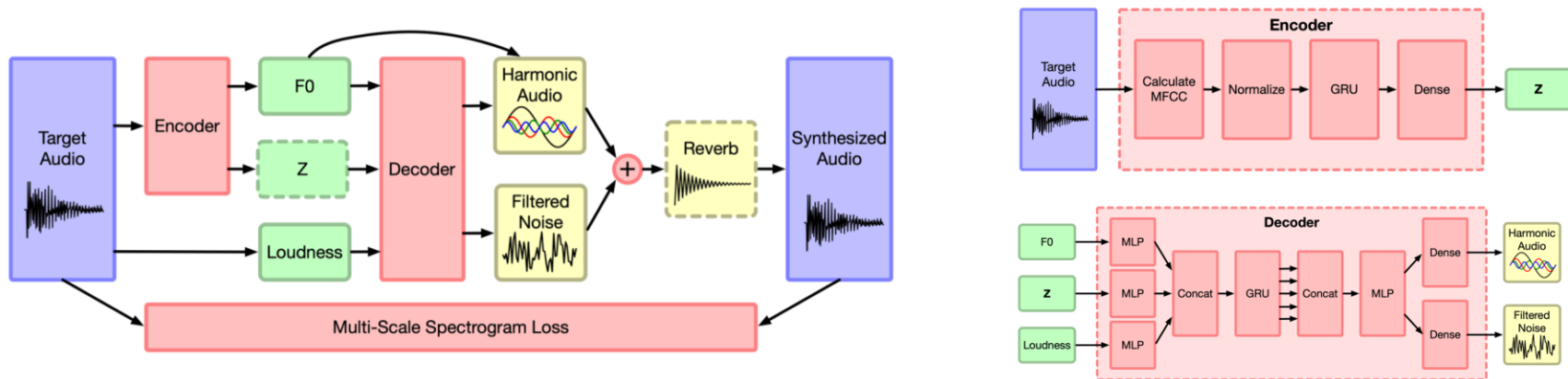
Spectral Analysis



Sine + Noise Synthesis

DDSP: Differentiable Digital Signal Processing

- A hybrid model of neural network and spectral modeling synthesis
 - Pitch (F0): use a pretrained pitch estimation model for F0 estimation
 - Loudness: an average of A-weighted power spectrum
 - Timber (z): a combination of MFCC, GRU and dense layer
 - **Trainable but partially deterministic model**



DDSP: Differentiable Digital Signal Processing

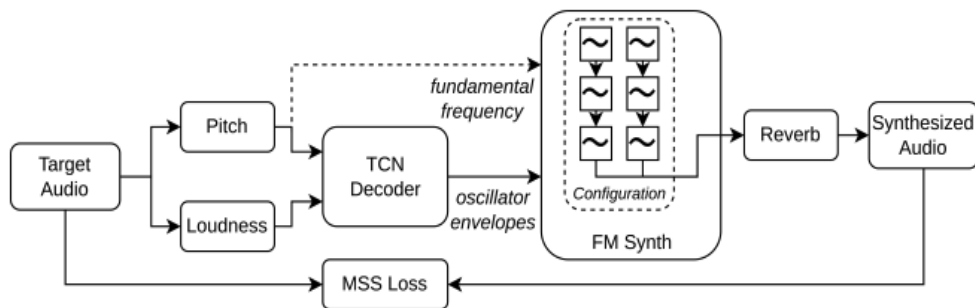
- Multi-scale spectrogram loss
 - Use L1 difference with both linear and log scales: $L_i = \|S_i - \hat{S}_i\|_1 + \|\log S_i - \log \hat{S}_i\|_1$
 - Summed for different window sizes: $L_{\text{reconstruction}} = \sum_i L_i$
 - 64, 128, 256, 512, 1024, 2048 samples
- Demos
 - <https://storage.googleapis.com/ddsp/index.html>
- Tone transfer
 - <https://sites.research.google/tonetransfer/about>
 - <https://www.aisongcontest.com/participants-2022/yaboi-hanoi>

DDSP-VST

DDSP captures the small nuances of your playing

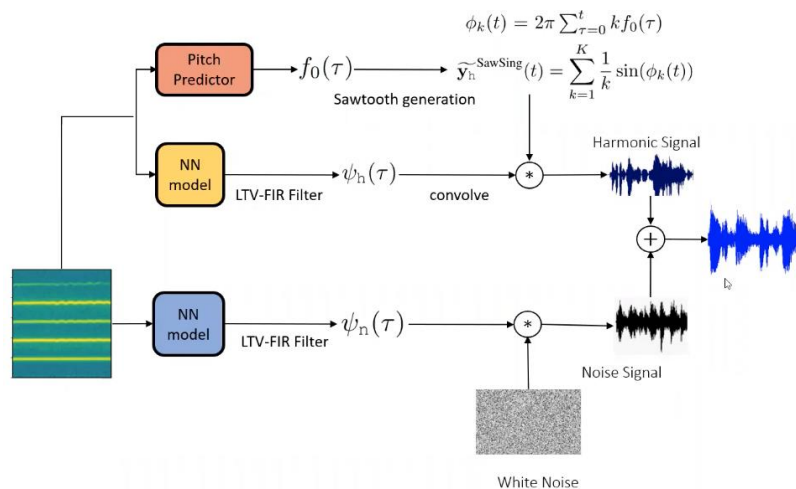
<https://magenta.tensorflow.org/ddsp-vst>

Neural Versions of Traditional Sound Synthesis Techniques



DDX7: Differential FM Synthesis

<https://fcaspe.github.io/ddx7/>



SawSing: Differential Subtractive Synthesis

<https://ddspvocoder.github.io/ismir-demo/>

Differentiable Wavetable Synthesis
Neural Granular Sound Synthesis
DDS-Piano