Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders

Jesse Engel* [1] **Cinjon Resnick*** [1] **Adam Roberts** [1] **Sander Dieleman** [2] **Douglas Eck** [1]
Karen Simonyan [2] **Mohammad Norouzi** [1]

Abstract

Generative models in vision have seen rapid progress due to algorithmic improvements and the availability of high-quality image datasets. In this paper, we offer contributions in both these areas to enable similar progress in audio modeling. First, we detail a powerful new WaveNet-style autoencoder model that conditions an autoregressive decoder on temporal codes learned from the raw audio waveform. Second, we introduce NSynth, a large-scale and high-quality dataset of musical notes that is an order of magnitude larger than comparable public datasets. Using NSynth, we demonstrate improved qualitative and quantitative performance of the WaveNet autoencoder over a well-tuned spectral autoencoder baseline. Finally, we show that the model learns a manifold of embeddings that allows for morphing between instruments, meaningfully interpolating in timbre to create new types of sounds that are realistic and expressive.

1. Introduction

Audio synthesis is important for a large range of applications including text-to-speech (TTS) systems and music generation. Audio generation algorithms, know as vocoders in TTS and synthesizers in music, respond to higher-level control signals to create fine-grained audio waveforms. Synthesizers have a long history of being hand-designed instruments, accepting control signals such as 'pitch', 'velocity', and filter parameters to shape the tone, timbre, and dynamics of a sound (Pinch et al., 2009). In spite of their limitations, or perhaps because of them, synthesizers have had a profound effect on the course of music and culture in the past half century (Punk, 2014).

In this paper, we outline a data-driven approach to audio synthesis. Rather than specifying a specific arrangement of oscillators or an algorithm for sample playback, such as in FM Synthesis or Granular Synthesis (Chowning, 1973; Xenakis, 1971), we show that it is possible to generate new types of expressive and realistic instrument sounds with a neural network model. Further, we show that this model can learn a semantically meaningful hidden representation that can be used as a high-level control signal for manipulating tone, timbre, and dynamics during playback.

Explicitly, our two contributions to advance the state of generative audio modeling are:

- A WaveNet-style autoencoder that learns temporal hidden codes to effectively capture longer term structure without external conditioning.
- NSynth: a large-scale dataset for exploring neural audio synthesis of musical notes.

The primary motivation for our novel autoencoder structure follows from the recent advances in autoregressive models like WaveNet (van den Oord et al., 2016a) and SampleRNN (Mehri et al., 2016). They have proven to be effective at modeling short and medium scale (∼500ms) signals, but rely on external conditioning for longer-term dependencies. Our autoencoder removes the need for that external conditioning. It consists of a WaveNet-like encoder that infers hidden embeddings distributed in time and a WaveNet decoder that uses those embeddings to effectively reconstruct the original audio. This structure allows the size of an embedding to scale with the size of the input and encode over much longer time scales.

Recent breakthroughs in generative modeling of images (Kingma & Welling, 2013; Goodfellow et al., 2014; van den Oord et al., 2016b) have been predicated on

*Equal contribution [1]Google Brain, Mountain View, California, USA. Work done while Cinjon Resnick was a member of the Google Brain Residency Program [2]DeepMind, London, England. Correspondence to: Jesse Engel <jesseengel@google.com>.

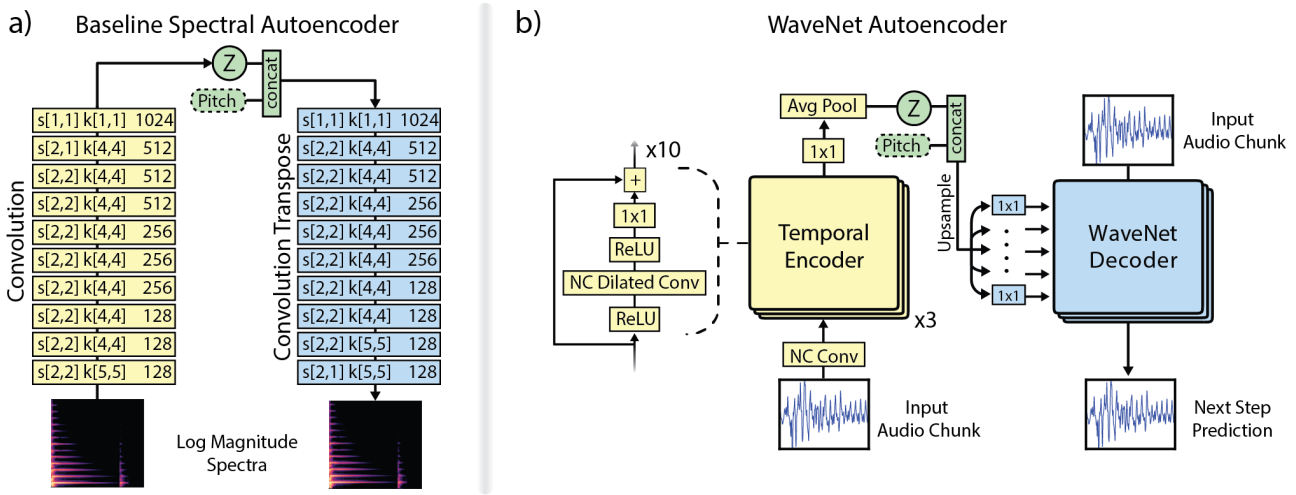

Figure 1. Models considered in this paper. For both models, we optionally condition on pitch by concatenating the hidden embedding with a one-hot pitch representation. 1*a*. Baseline spectral autoencoder: Each block represents a nonlinear 2-D convolution with stride (s), kernel size (k), and channels (#). 1*b*. The WaveNet autoencoder: Downsampling in the encoder occurs only in the average pooling layer. The embeddings are distributed in time and upsampled with nearest neighbor interpolation to the original resolution before biasing each layer of the decoder. 'NC' indicates non-causal convolution. '1x1' indicates a 1-D convolution with kernel size 1. See Section 2.1 for further details.

the availability of high-quality and large-scale datasets such as MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), CIFAR (Krizhevsky & Hinton, 2009) and ImageNet (Deng et al., 2009). While generative models are notoriously hard to evaluate (Theis et al., 2015), these datasets provide a common test bed for consistent qualitative and quantitative evaluation, such as with the use of the Inception score (Salimans et al., 2016).

We recognized the need for an audio dataset that was as approachable as those in the image domain. Audio signals found in the wild contain multi-scale dependencies that prove particularly difficult to model (Raffel, 2016; Bertin-Mahieux et al., 2011; King et al., 2008; Thickstun et al., 2016), leading many previous efforts at data-driven audio synthesis to focus on more constrained domains such as texture synthesis or training small parametric models (Sarroff & Casey, 2014; McDermott et al., 2009).

Inspired by the large, high-quality image datasets, NSynth is an order of magnitude larger than comparable public datasets (Humphrey, 2016). It consists of ~300k four-second annotated notes sampled at 16kHz from ~1k harmonic musical instruments.

After introducing the models and describing the dataset, we evaluate the performance of the WaveNet autoencoder over a baseline convolutional autoencoder model trained on spectrograms. We examine the tasks of reconstruction and interpolation, and analyze the learned space of embeddings. For qualitative evaluation, download audio files for all examples mentioned in this paper **here**. Despite our best efforts to convey analysis in plots, *listening to the samples is essential to understanding this paper* and we strongly encourage the reader to listen along as they read.

2. Models

2.1. WaveNet Autoencoder

WaveNet (van den Oord et al., 2016a) is a powerful generative approach to probabilistic modeling of raw audio. In this section we describe our novel WaveNet autoencoder structure. The primary motivation for this approach is to attain consistent long-term structure without external conditioning. A secondary motivation is to use the learned encodings for applications such as meaningful audio interpolation.

Recalling the original WaveNet architecture described in (van den Oord et al., 2016a), at each step a stack of dilated convolutions predicts the next sample of audio from a fixed-size input of prior sample values. The joint probability of the audio x is factorized as a product of conditional probabilities:

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, ..., x_{N-1})$$

Unconditional generation from this model manifests as "babbling" due to the lack of longer term structure (Listen: *CAUTION, VERY LOUD!* (ex1, ex2, ex3, ex4)). However, (van den Oord et al., 2016a) showed in the context of speech that long-range structure can be enforced by conditioning on temporally aligned linguistic features.

Our autoencoder removes the need for that external conditioning. It works by taking raw audio waveform as input from which the encoder produces an embedding $Z = f(x)$. Next, we causally shift the same input and feed it into the decoder, which reproduces the input waveform. The joint probablity is now:

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, ..., x_{N-1}, f(x))$$

We could parameterize Z as a latent variable $p(Z|x)$ that we would have to marginalize over (Gulrajani et al., 2016), but in practice we have found this to be less effective. As discussed in (Chen et al., 2016), this may be due to the decoder being so powerful that it can ignore the latent variables unless they encode a much larger context that's otherwise inaccessible.

Note that the decoder could completely ignore the deterministic encoding and degenerate to a standard unconditioned WaveNet. However, because the encoding is a strong signal for the supervised output, the model learns to utilize it.

During inference, the decoder autoregressively generates a single output sample at a time conditioned on an embedding and a starting palette of zeros. The embedding can be inferred deterministically from audio or drawn from other points in the embedding space, e.g. through interpolation or analogy (White, 2016).

Figure 1b depicts the model architecture in more detail. The temporal encoder model is a 30-layer nonlinear residual network of dilated convolutions followed by 1x1 convolutions. Each convolution has 128 channels and precedes a ReLU nonlinearity. The output feed into another 1x1 convolution before downsampling with average pooling to get the encoding Z. We call it a 'temporal encoding' because the result is a sequence of hidden codes with separate dimensions for time and channel. The time resolution depends on the stride of the pooling. We tune the stride, keeping total size of the embedding constant (∼32x compression). In the trade-off between temporal resolution and embedding expressivity, we find a sweet spot at a stride of 512 (32ms) with 16 dimensions per timestep, yielding a 125x16 embedding for each NSynth note. We additionally explore models that condition on global attributes by utilizing a one-hot pitch embedding.

The WaveNet decoder model is similar to that presented in (van den Oord et al., 2016a). We condition it by biasing every layer with a different linear projection of the temporal embeddings. Since the decoder does not downsample anywhere in the network, we upsample the temporal encodings to the original audio rate with nearest neighbor interpolation. As in the original design, we quantize our input audio using 8-bit mu-law encoding and predict each output step with a softmax over the resulting 256 values.

This WaveNet autoencoder is a deep and expressive network, but has the trade-off of being limited in temporal context to the chunk-size of the training audio. While this is sufficient for consistently encoding the identity of a sound and interpolating among many sounds, achieving larger context would be better and is an area of ongoing research.

2.2. Baseline: Spectral Autoencoder

As a point of comparison, we set out to create a straightforward yet strong baseline for the our neural audio synthesis experiments. Inspired by image models (Vincent et al., 2010), we explore convolutional autoencoder structures with a bottleneck that forces the model to find a compressed representation for an entire note. Figure 1a shows a block diagram of our baseline architecture. The convolutional encoder and decoder are each 10 layers deep with 2x2 strides and 4x4 kernels. Every layer is followed by a leaky-ReLU (0.1) nonlinearity and batch normalization (Ioffe & Szegedy, 2015). The number of channels grows from 128 to 1024 before a linear fully-connected layer creates a single 1984[1] dimensional hidden vector (Z) to match that of the WaveNet autoencoder.

Given the simplicity of the architecture, we examined a range of input representations. Using the raw waveform as input with a mean-squared error (MSE) cost proved difficult to train and highlighted the inadequacy of the independent Gaussian assumption. Spectral representations such as the real and imaginary components of the Fast Fourier Transform (FFT) fared better, but suffered from low perceptual quality despite achieving low MSE cost. We found that training on the log magnitude of the power spectra, peak normalized to be between 0 and 1, correlated better with perceptual distortion.

We also explored several representations of phase, in-

[1]This size was aligned with a WaveNet autoencoder that had a pooling stride of 1024 and a 62x32 embedding.

cluding instantaneous frequency and circular normal cost functions (see Appendix), but in each case independently estimating phase and magnitude led to poor sample quality due to phase errors. We find a large improvement by estimating only the magnitude and using a well established iterative technique to reconstruct the phase (Griffin & Lim, 1984). To get the best results, we used a large FFT size (1024) relative to the hop size (256) and ran the algorithm for 1000 iterations. As a final heuristic, we weighted the MSE loss, starting at 10 for 0Hz and decreasing linearly to 1 at 4000Hz and above. At the expense of some precision in timbre, this created more phase coherence for the fundamentals of notes, where errors in the linear spectrum lead to a larger relative error in frequency.

2.3. Training

We train all models with stochastic gradient descent with an Adam optimizer (Kingma & Ba, 2014). The baseline models commonly use a learning rate of 1e-4, while the WaveNet models use a schedule, starting at 2e-4 and descending to 6e-5, 2e-5, and 6e-6 at iterations 120k, 180k, and 240k respectively. The baseline models train asynchronously for 1800k iterations with a batch size of 8. The WaveNet models train synchronously for 250k iterations with a batch size of 32.

3. The NSynth Dataset

To evaluate our WaveNet autoencoder model, we wanted an audio dataset that let us explore the learned embeddings. Musical notes are an ideal setting for this study as we hypothesize that the embeddings will capture structure such as pitch, dynamics, and timbre. While several smaller datasets currently exist (Goto et al., 2003; Romani Picas et al., 2015), deep networks train better on abundant, high-quality data, motivating the development of a new dataset.

3.1. A Dataset of Musical Notes

NSynth consists of 306 043 musical notes, each with a unique pitch, timbre, and envelope. For 1006 instruments from commercial sample libraries, we generated four second, monophonic 16kHz audio snippets, referred to as notes, by ranging over every pitch of a standard MIDI piano (21-108) as well as five different velocities[2] (25, 50, 75, 100, 127). The note was held for the first three seconds and allowed to decay for the final second. Some instruments are not capable of producing all 88 pitches in this range, resulting in an average of 65.4 pitches per instrument. Furthermore, the commercial sample packs occasionally contain duplicate sounds across multiple velocities, leaving an average of 4.75 unique velocities per pitch.

[2]MIDI velocity is similar to volume control and they have a direct relationship. For physical intuition, higher velocity corresponds to pressing a piano key harder.

3.2. Annotations

We also annotated each of the notes with three additional pieces of information based on a combination of human evaluation and heuristic algorithms:

- Source: The method of sound production for the note's instrument. This can be one of 'acoustic' or 'electronic' for instruments that were recorded from acoustic or electronic instruments, respectively, or 'synthetic' for synthesized instruments.
- Family: The high-level family of which the note's instrument is a member. Each instrument is a member of exactly one family. See Appendix for the complete list.
- Qualities: Sonic qualities of the note. See Appendix for the complete list of classes and their co-occurrences. Each note is annotated with zero or more qualities.

3.2.1. Availability

The full NSynth dataset is available for download at `https://magenta.tensorflow.org/datasets/nsynth` as TFRecord files split into training and holdout sets. Each note is represented by a serialized TensorFlow Example protocol buffer containing the note and annotations. Details of the format can be found in the README.

4. Evaluation

We evaluate and analyze our models on the tasks of note reconstruction, instrument interpolation, and pitch interpolation.

Audio is notoriously hard to represent visually. Magnitude spectrograms capture many aspects of a signal for analytics, but two spectrograms that appear very similar to the eye can correspond to audio that sound drastically different due to phase differences. We have included supplemental audio examples of every plot and *encourage the reader to listen along as they read.*

That said, in our analysis we present examples as plots of the constant-q transform (CQT) (Brown, 1991), which is useful because it is shift invariant to changes in the fundamental frequency. In this way, the struc-

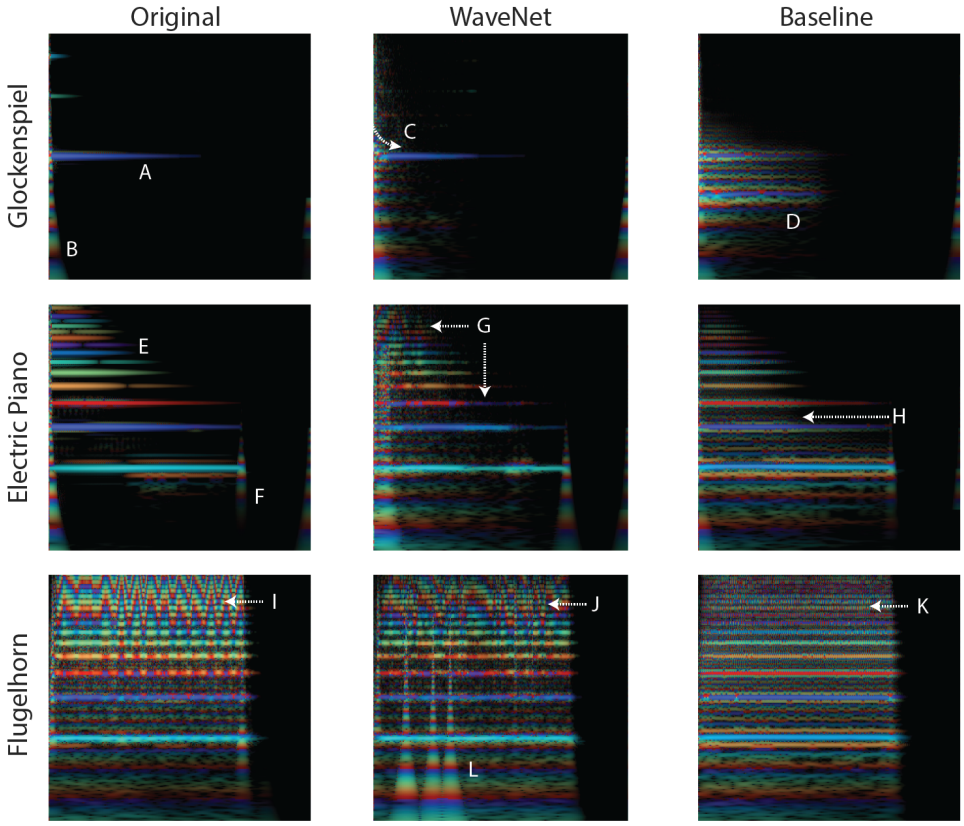

Figure 2. Reconstructions of notes from three different instruments. Each note is displayed as a "Rainbowgram", a CQT spectrogram with intensity of lines proportional to the log magnitude of the power spectrum and color given by the derivative of the phase. Time is on the horizontal axis and frequency on the vertical axis. See Section 4.1 for details. (Listen: Glockenspiel (O, W, B), Electric Piano (O, W, B), Flugelhorn (O, W, B))

ture and envelope of the overtone series (higher harmonics) determines the dynamics and timbre of a note, regardless of its base frequency. However, due to the logarithmic binning of frequencies, transient noise-like impulses appear as rainbow "pyramidal spikes" rather than straight broadband lines. We display CQTs with a pitch range of 24-96 (C2-C8), hop size of 256, 40 bins per octave, and a filter scale of 0.8.

As phase plays such an essential part in sample quality, we have attempted to show both magnitude and phase on the same plot. The intensity of lines is proportional to the log magnitude of the power spectrum while the color is given by the derivative of the unrolled phase ('instantaneous frequency') (Boashash, 1992). We display the derivative of the phase because it creates a solid continuous line for a harmonic of a consistent frequency. We can understand this because if the instantaneous frequency of a harmonic (f_{harm}) and an FFT bin (f_{bin}) are not exactly equal, each timestep will introduce a constant phase shift, $\Delta\phi = (f_{bin} - f_{harm})\frac{hopsize}{samplerate}$. We affectionately refer to these instantaneous frequency colored spectrograms as "Rainbowgrams" due to their tendency to form rainbows as the instantaneous frequencies modulate up and down.

4.1. Reconstruction

Figure 2 displays rainbowgrams for notes from 3 different instruments in the holdout set, where the original notes are on the first column and the model reconstructions are on the second and third columns. Each note has a similar structure with some noise on onset, a fundamental frequency with a series of harmonics, and a decay. For all the WaveNet models, there is a slight built-in distortion due to the compression of the mu-law encoding. It is a minor effect for many samples, but is more pronounced for lower frequencies. Using different representations without this distortion is an ongoing area of research.

While each rainbowgram matches the general contour of the original note, we can hear a pronounced difference in sample quality that we can ascribe to certain

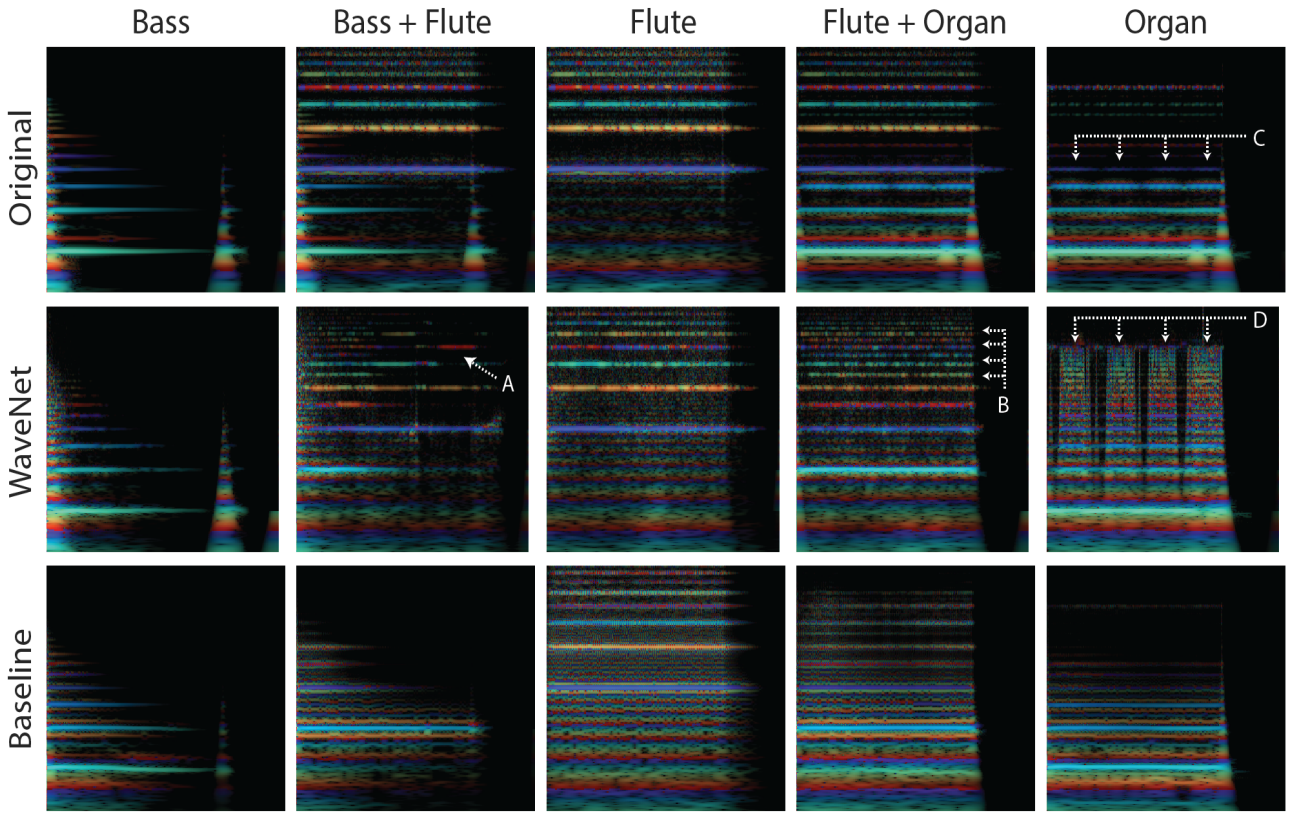

Figure 3. Rainbowgrams of linear interpolations between three different notes from instruments in the holdout set. For the original rainbowgrams, the raw audio is linearly mixed. For the models, samples are generated from linear interpolations in embedding space. See Section 4.2 for details.(Listen: Original (B, BF, F, FO, O, OB), WaveNet (B, BF, F, FO, O, OB), Baseline (B, BF, F, FO, O, OB))

features. For the Glockenspiel, we can see that the WaveNet autoencoder reproduces the magnitude and phase of the fundamental (solid blue stripe, (A)), and also the noise on attack (vertical rainbow spike (B)). There is a slight error in the fundamental as it starts a little high and quickly descends to the correct pitch (C). In contrast, the baseline has a more percussive, multitonal sound, similar to a bell or gong. The fundamental is still present, but so are other frequencies, and the phases estimated from the Griffin-Lim procedure are noisy as indicated by the blurred horizontal rainbow texture (D).

The electric piano has a more clearly defined harmonic series (the horizontal rainbow solid lines, (E)) and a noise on the beginning and end of the note (vertical rainbow spikes, (F)). Listening to the sound, we hear that it is slightly distorted, which promotes these upper harmonics. Both the WaveNet autoencoder and the baseline produce rainbowgrams with similar shapes to the original, but with different types of phase artifacts. The WaveNet model has sufficient phase structure to model the distortion, but has a slight wavering of the instantaneous frequency of some harmonics, as seen in the color change in harmonic stripes (G). In contrast, the baseline lacks the structure in phase to maintain the punchy character of the original note, and produces a duller sound that is slightly out of tune. This is represented in the less brightly colored harmonics due to phase noise (H).

The flugelhorn displays perhaps the starkest difference between the two models. The sound combines rich harmonics (many lines), non-tonal wind and lip noise (background color), and vibrato - oscillation of pitch that results in a corresponding rainbow of color in all of the harmonics. While the WaveNet autoencoder does not replicate the exact trace of the vibrato (I), it creates a very similar rainbowgram with oscillations in the instantaneous frequency at all levels synced across the harmonics (J). This results in a rich and natural sounding reconstruction with all three aspects of the original sound. The baseline, by comparison, is unable to model such structure. It creates a more or less

correct harmonic series, but the phase has lots of random perturbations. Visually this shows up as colors which are faded and speckled with rainbow noise (K), which contrasts with the bright colors of the original and WaveNet examples. Acoustically, this manifests as an unappealing buzzing sound laid over an inexpressive and consistent series of harmonics. The WaveNet model also produces a few inaudible discontinuities visually evidenced by the vertical rainbow spikes (L).

4.1.1. Quantitative Comparison

Inspired by the use of the Inception Score for images (Salimans et al., 2016), we train a multi-task classification network to perform a quantitative comparison of the model reconstructions by predicting pitch and quality labels on the NSynth dataset (details in the Appendix). The network configuration is the same as the baseline encoder and testing is done on reconstructions of a randomly chosen subset of 4096 examples from the held-out set.

Table 1. Classification accuracy of a deep nonlinear pitch and quality classifier on reconstructions of a test set.

	Pitch	Quality
Original Audio	91.6%	90.1%
WaveNet Recon	79.6%	88.9%
Baseline Recon	46.9%	85.2%

The results in Table 1 confirm our qualititive observation that the WaveNet reconstructions are of superior quality. The classifier is ∼70% more successful at extracting pitch from the reconstructed WaveNet samples than the baseline and several points higher for predicting quality information, giving an accuracy roughly equal to the original audio.

4.2. Interpolation in Timbre and Dynamics

Given the limited factors of variation in the dataset, we know that a successful embedding space (Z) should span the range of timbre and dynamics in its reconstructions. In Figure 3, we show reconstructions from linear interpolations (0.5:0.5) in the Z space among three different instruments and additionally compare these to interpolations in the original audio space. The latter are simple super-positions of the individual instruments' rainbowgrams. This is perceptually equivalent to the two instruments being played at the same time.

In contrast, we find that the generative models fuse aspects of the instruments. As we saw in Section 4.1, the WaveNet autoencoder models the data much more realistically than the baseline, so it is no surprise that it also learns a manifold of codes that yield more perceptually interesting reconstructions.

For example, in the interpolated note between the bass and flute (Figure 3, column 2), we can hear and see that both the baseline and WaveNet models blend the harmonic structure of the two instruments while imposing the amplitude envelope of the bass note onto the upper harmonics of the flute note. However, the WaveNet model goes beyond this to create a dynamic mixing of the overtones in time, even jumping to a higher harmonic at the end of the note (A). This sound captures expressive aspects of the timbre and dynamics of both the bass and flute, but is distinctly separate from either original note. This contrasts with the interpolation in audio space, where the dynamics and timbre of the two notes is independent. The baseline model also introduces phase distortion similar to those in the reconstructions of the bass and flute.

We see this phenomenon again in the interpolation between flute and organ (Figure 3, column 4). Both models also seem to create new harmonic structure, rather than just overlay the original harmonics. The WaveNet model adds additional harmonics as well as a sub-harmonic to the original flute note, all while preserving phase relationships (B). The resulting sound has the breathiness of a flute, with the upper frequency modulation of an organ. By contrast, the lack of phase structure in the baseline leads to a new harmonic yet dull sound lacking a unique character.

The WaveNet model additionally has a tendency to exaggerate amplitude modulation behavior, while the baseline suppresses it. If we examine the original organ sound (Figure 3, column 5), we can see a subtle modulation signified by the blue harmonics periodically fading to black (C). The baseline model misses this behavior completely as it is washed out. Conversely, the WaveNet model amplifies the behavior, adding in new harmonics not present in the original note and modulating all the harmonics. This is seen in the figure by four vertical black stripes that align with the four modulations of the original signal (D).

4.3. Entanglement of Pitch and Timbre

By conditioning on pitch during training, we hypothesize that we should be able to generate multiple pitches from a single Z vector that preserve the identity of timbre and dynamics. Our initial attempts were unsuccessful, as it seems our models had learned to ignore the conditioning variable. We investigate this further with classification and correlation studies.

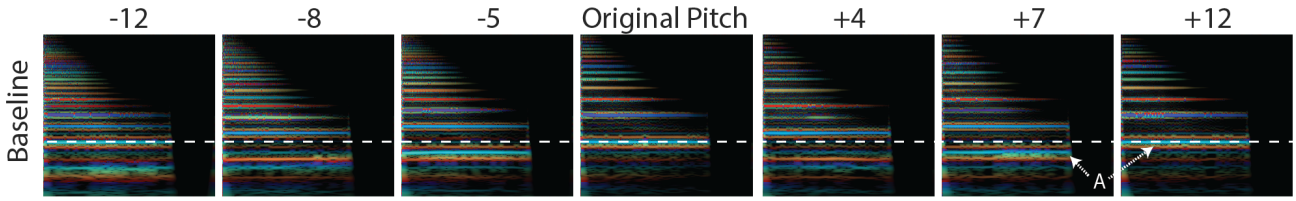

Figure 4. Conditioning on pitch. These rainbowgrams are reconstructions of a single electric piano note from the holdout set. They were synthesized with the baseline model (128 hidden dimensions). By holding Z constant and conditioning on different pitches, we can play two octaves of a C major chord from a single embedding. The original pitch (MIDI C60) is dashed in white for comparison. See Section 4.3.1 for details. (Listen: -12, -8, -5, 0, +4, +7, +12)

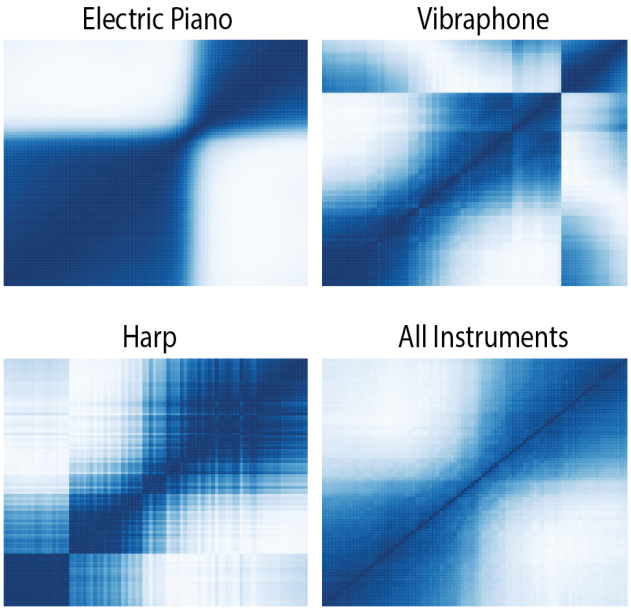

Figure 5. Correlation of embeddings across pitch for three different instruments and the average across all instruments. These embeddings were taken from a WaveNet model trained without pitch conditioning.

Table 2. Classification accuracy (in percentage) of a linear pitch classifier trained on learned embeddings. The decoupling of pitch and embedding becomes more pronounced at smaller embedding sizes as shown by the larger relative decrease in classification accuracy.

	Z Size	No Pitch Cond.	Pitch Cond.	Relative Change
WaveNet	1984	58.1	40.5	**-30.4**
Baseline	1984	63.8	55.2	**-13.5**
Baseline	1024	57.4	42.1	-26.7
Baseline	512	63.2	21.8	-65.5
Baseline	256	57.7	21.0	-63.6
Baseline	128	58.2	21.2	-63.6
Baseline	64	59.8	15.2	**-74.6**

4.3.1. Pitch Classification from Z

One way to study the entanglement of pitch and Z is to consider the pitch classification accuracy from embeddings. If training with pitch conditioning disentangles the representation of pitch and timbre, then we would expect a linear pitch classifier trained on the embeddings to drop in accuracy. To test this, we train a series of baseline autoencoder models with different embedding sizes, both with and without pitch conditioning. For each model, we then train a logistic regression pitch classifier on its embeddings and test on a random sample of 4096 held-out embeddings.

The first two rows of Table 2 demonstrate that the baseline and WaveNet models decrease in classification accuracy by 13-30% when adding pitch conditioning during training. This is indicative a reduced presence of pitch information in the latent code and thus a decoupling of pitch and timbre information. Further, as the total embedding size decreases below 512, the accuracy drop becomes much more pronounced, reaching a 75% relative decrease. This is likely due to the greater expressivity of larger embeddings, where there is less to be gained from utilizing the pitch conditioning. However, as the embedding size decreases, so too does reconstruction quality. This is more pronounced for the WaveNet models, which have farther to fall in terms of sample quality.

As a proof of principle, we find that for a baseline model with an embedding size of 128, we are able to successfully balance reconstruction quality and response to conditioning. Figure 4 demonstrates two octaves of a C major chord created from a single embedding of an electric piano note, but conditioned on different pitches. The resulting harmonic structure of the original note is only partially preserved across the range. As we shift the pitch upwards, a sub-harmonic emerges (A) such that the pitch +12 note is similar to

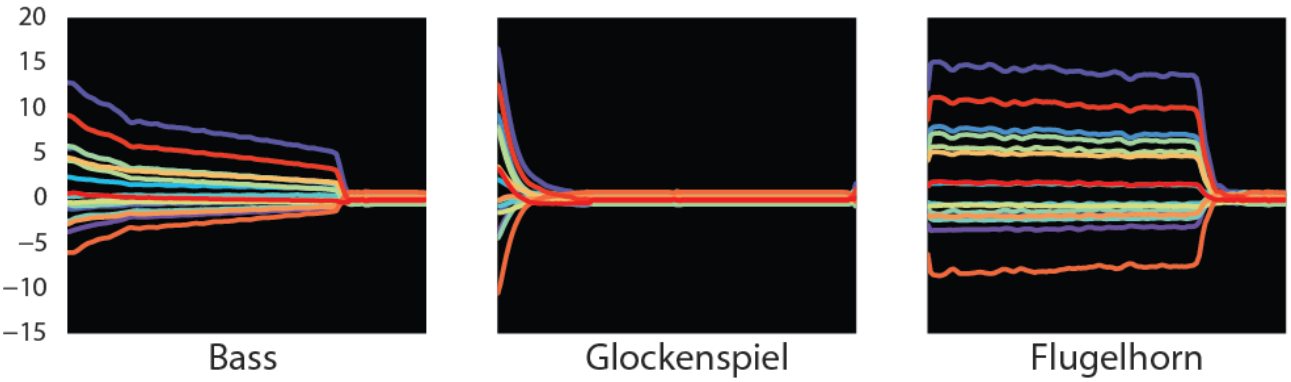

Figure 6. Temporal embeddings for three different instruments. The different colors represent the 16 different dimensions of the embeddings for 125 timesteps (each 32ms). Note that the embedding have a contour similar to the magnitude contour of the original note and decay close to zero when there is no sound. With this in mind, they can be thought of as a "driving function" for a nonlinear oscillator / infinite impulse response filter.

the original except that the harmonics of the octave are accentuated in amplitude. This aligns with our pitch classification results, where we find that pitches are most commonly confused with those one octave away (see Appendix). These errors can account for as much as 20% absolute classification error.

4.3.2. Z Correlation across Pitch

We can gain further insight into the relationship between timbre and pitch by examining the correlation of WaveNet embeddings among pitches for a given instrument. Figure 5 shows correlations for several instruments across their entire 88 note range at velocity 127. We see that each instrument has a unique partitioning into two or more registers over which notes of different pitches have similar embeddings. Even the average over all instruments shows a broad distinction between high and low registers. On reflection, this is unsurprising as the timbre and dynamics of an instrument can vary dramatically across its range.

4.4. Generalization of Temporal Encodings

The WaveNet autoencoder model has some unique properties that allow it to generalize to situations not in the dataset. Since the model learns embeddings that bias an autoregressive decoder, they effectively act as a "driving function" for a nonlinear oscillator / infinite impulse response filter. This is made clear by Figure 6, where the embeddings follow a magnitude contour similar to that of the rainbowgrams of their corresponding sounds in Figures 2 and 3.

Further, much like a spectrogram, the embeddings only capture a local context. This lets them generalize in time. The model has only ever seen single notes with sound that lasts for up to three seconds, and yet Figure 7 demonstrates that it can successfully reconstruct both a whole series of notes, as well as notes played for longer than three seconds. While the WaveNet autoencoder adds more harmonics to the original timbre of the organ instrument, it follows the fundamental frequency as it plays up two octaves of a C major arpeggio, back down a G dominant arrpeggio, and holds for several seconds on the base note. The fact that it has never seen a transition between two notes is clear, as the fundamental frequency actually glissandos smoothly between new notes.

5. Conclusion and Future Directions

In this paper, we have introduced a WaveNet autoencoder model that captures long term structure without the need for external conditioning and demonstrated its effectiveness on the new NSynth dataset for generative modeling of audio.

The WaveNet autoencoder that we describe is a powerful representation for which there remain multiple avenues of exploration. It builds upon the fine-grained local understanding of the original WaveNet work and provides access to a useful hidden space. However, due to memory constraints, it is unable to fully capture global context. Overcoming this limitation is an important open problem.

NSynth was inspired by image recognition datasets that have been core to recent progress in deep learning. Similar to how many image datasets focus on a single object per example, NSynth hones in on a

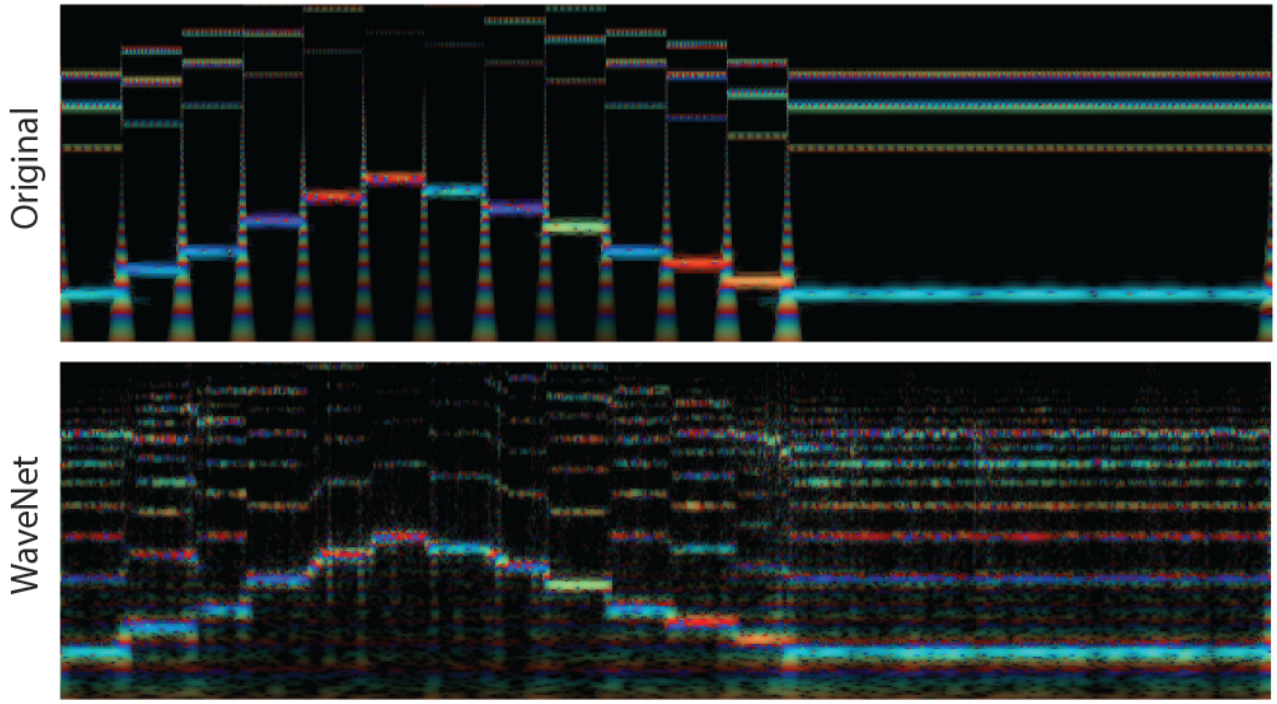

Figure 7. Rainbowgrams of a series of notes reconstructed by the WaveNet autoencoder. The model was never trained on more than one note at a time or on clips longer than four seconds, but it does a fair job of reconstructing this ten-second long scale. (Listen: Original, Reconstruction)

single note. Indeed, much modern music production employs such a factorization, using MIDI for note sequences and software synthesizers for timbre. Note-to-note dependencies can be partly restored by passing sequence-level timbre and dynamics information to the note-level synthesizer. While not perfect, this factorization is based on the physics of many instruments and is surprisingly effective.

We encourage the broader community to use NSynth as a benchmark and entry point into audio machine learning. We also view NSynth as a building block for future datasets and envision a high-quality multi-note dataset for tasks like generation and transcription that involve learning complex language-like dependencies.

References

Bertin-Mahieux, Thierry, Ellis, Daniel PW, Whitman, Brian, and Lamere, Paul. The million song dataset. In *ISMIR*, volume 2, pp. 10, 2011.

Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

Boashash, Boualem. Estimating and interpreting the instantaneous frequency of a signal. i. fundamentals. *Proceedings of the IEEE*, 80(4):520–538, 1992.

Brown, Judith C. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

Chen, Xi, Kingma, Diederik P., Salimans, Tim, Duan, Yan, Dhariwal, Prafulla, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Variational lossy autoencoder. *CoRR*, abs/1611.02731, 2016. URL http://arxiv.org/abs/1611.02731.

Chowning, John M. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the audio engineering society*, 21(7):526–534, 1973.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goto, Masataka, Hashiguchi, Hiroki, Nishimura, Takuichi, and Oka, Ryuichi. Rwc music database: Music genre database and musical instrument sound database. 2003.

Griffin, Daniel and Lim, Jae. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

Gulrajani, Ishaan, Kumar, Kundan, Ahmed, Faruk, Taiga, Adrien Ali, Visin, Francesco, Vázquez, David, and Courville, Aaron C. Pixelvae: A latent variable model for natural images. *CoRR*, abs/1611.05013, 2016. URL http://arxiv.org/abs/1611.05013.

Humphrey, Eric J. Minst, a collection of musical sound datasets, 2016. URL https://github.com/ejhumphrey/minst-dataset/.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

King, Simon, Clark, Robert AJ, Mayo, Catherine, and Karaiskos, Vasilis. The blizzard challenge 2008. 2008.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist database of handwritten digits, 1998.

McDermott, Josh H, Oxenham, Andrew J, and Simoncelli, Eero P. Sound texture synthesis via filter statistics. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, pp. 297–300. IEEE, 2009.

Mehri, Soroush, Kumar, Kundan, Gulrajani, Ishaan, Kumar, Rithesh, Jain, Shubham, Sotelo, Jose, Courville, Aaron C., and Bengio, Yoshua. Samplernn: An unconditional end-to-end neural audio generation model. *CoRR*, abs/1612.07837, 2016. URL http://arxiv.org/abs/1612.07837.

Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Pinch, Trevor J, Trocco, Frank, and Pinch, TJ. *Analog days: The invention and impact of the Moog synthesizer*. Harvard University Press, 2009.

Punk, Daft. Giorgio by morodor, 2014. URL https://www.youtube.com/watch?v=zhl-Cs1-sG4.

Raffel, Colin. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, COLUMBIA UNIVERSITY, 2016.

Romani Picas, Oriol, Parra Rodriguez, Hector, Dabiri, Dara, Tokuda, Hiroshi, Hariya, Wataru, Oishi, Koji, and Serra, Xavier. A real-time system for measuring sound goodness in instrumental sounds. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

Salimans, Tim, Goodfellow, Ian J., Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

Sarroff, Andy M and Casey, Michael A. Musical audio synthesis using autoencoding neural nets. In *ICMC*, 2014.

Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Thickstun, John, Harchaoui, Zaid, and Kakade, Sham. Learning features of music from scratch. In *preprint, https://arxiv.org/abs/1611.09827*, 2016.

van den Oord, Aäron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew W., and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016a. URL http://arxiv.org/abs/1609.03499.

van den Oord, Aäron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016b. URL http://arxiv.org/abs/1601.06759.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine.

Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

White, Tom. Sampling generative networks: Notes on a few effective techniques. *CoRR*, abs/1609.04468, 2016. URL http://arxiv.org/abs/1609.04468.

Xenakis, Iannis. Formalized music. 1971.

Appendices

A. Phase Representation for the Baseline Model

We explored several audio representations for our baseline model. Each representation uses an MSE cost and always includes the magnitude of the STFT spectrogram. We found that training on the peak-normalized log magnitude of the power spectra correlated better with perceptual distortion. When using phase in teh objective, we regress on the phase angle. We can assume a circular normal distribution (Bishop, 2006) for the phase with a log likelihood loss proportional to $cos(\pi * (x - \hat{x}))$. Figure 8 shows CQT spectrograms of reconstructions of a trumpet sound from models trained on each input representation. We also include audio of each reconstruction, which is essential listening to hear the improvement of the perceptual weighting.

B. Description of Quality Tags

We provide quality annotations for the 10 different note qualities described below. None of the tags are mutually exclusive by definition except for Bright and Dark. However, it is possible for a note to be neither Bright nor Dark.

- **Bright**: A large amount of high frequency content and strong upper harmonics.
- **Dark**: A distinct lack of high frequency content, giving a muted and bassy sound. Also sometimes described as 'Warm'.
- **Distortion**: Waveshaping that produces a distinctive crunchy sound and presence of many harmonics. Sometimes paired with non-harmonic noise.
- **Fast Decay**: Amplitude envelope of all harmonics decays substantially before the 'note-off' point at 3 seconds.
- **Long Release**: Amplitude envelope decays slowly after the 'note-off' point, sometimes still present at the end of the sample at 4 seconds.
- **Multiphonic**: Presence of overtone frequencies related to more than one fundamental frequency.
- **Non-Linear Envelope**: Modulation of the sound with a distinct envelope behavior different than the monotonic decrease of the note. Can also include filter envelopes as well as dynamic envelopes.
- **Percussive**: A loud non-harmonic sound at note onset.
- **Reverb**: Room acoustics that were not able to be removed from the original sample.
- **Tempo-Synced**: Rhythmic modulation of the sound to a fixed tempo.

C. Details of Pitch and Quality Classifier

We train a multi-task classification model to do pitch and quality tag classification on the entire NSynth dataset. We use the the encoder structure from the baseline model with the exception that there is no bottleneck (see Figure 10). We use a softmax-crossentropy loss for the pitch labels as they are mutually exclusive and a sigmoid-crossentropy loss for the quality tags as they are not. Note that since the architecture uses only magnitude spectra, it cannot take advantage of the improved phase coherence of the WaveNet samples.

Table 3. Instrument annotations. Instruments are labeled with both a source and a family. The source denotes how each instrument's notes are generated: acoustic instrument, electronic instrument, or by software synthesis. The family denotes a high-level class for each instrument.

	Source			
Family	Acoust	Electr	Synth	**Total**
Bass	200	8387	60 368	68 955
Brass	13 760	70	0	13 830
Flute	6572	70	2816	9458
Guitar	13 343	16 805	5275	35 423
Keyboard	8508	42 709	3838	55 055
Mallet	27 722	5581	1763	35 066
Organ	176	36 401	0	36 577
Reed	14 262	76	528	14 866
String	20 510	84	0	20 594
Synth Lead	0	0	5501	5501
Vocal	3925	140	6688	10 753
Total	108 978	110 224	86 777	306 043

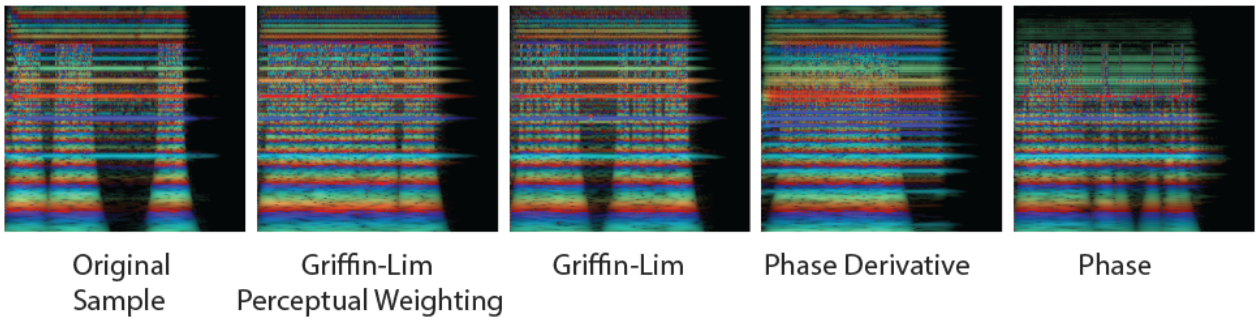

Figure 8. Reconstructions from baseline models trained with different phase representations. For Griffin-Lim, only the magnitude is modeled, and an 1000 iterations of an iterative technique is used to estimate the phase. (Listen: Original, Griffin-Lim Perceptual Weighting, Griffin-Lim, Phase Derivative, Phase)

Table 4. Co-occurrence probabilities and marginal frequencies of quality annotations. Both are presented as percentages.

	Quality	Bright	Dark	Distortion	Fast Decay	Long Release	Multiphonic	Nonlinear Env	Percussive	Reverb	Tempo-Synced
Co-occurrence	Dark	0.0									
	Distortion	25.9	2.5								
	Fast Decay	10.0	7.5	8.1							
	Long Release	9.0	5.2	9.8	0.0						
	Multiphonic	6.0	1.5	5.4	2.8	6.9					
	Nonlinear Env	8.5	1.4	6.6	2.1	6.7	8.6				
	Percussive	6.2	5.1	3.0	52.0	0.8	2.4	0.9			
	Reverb	6.6	8.9	0.3	13.0	13.7	0.7	3.5	12.4		
	Tempo-Synced	2.4	1.8	5.2	0.4	6.4	9.3	2.3	1.5	0.0	
	Frequency	13.5	11.0	17.0	14.7	8.5	3.4	3.2	10.2	16.8	1.8

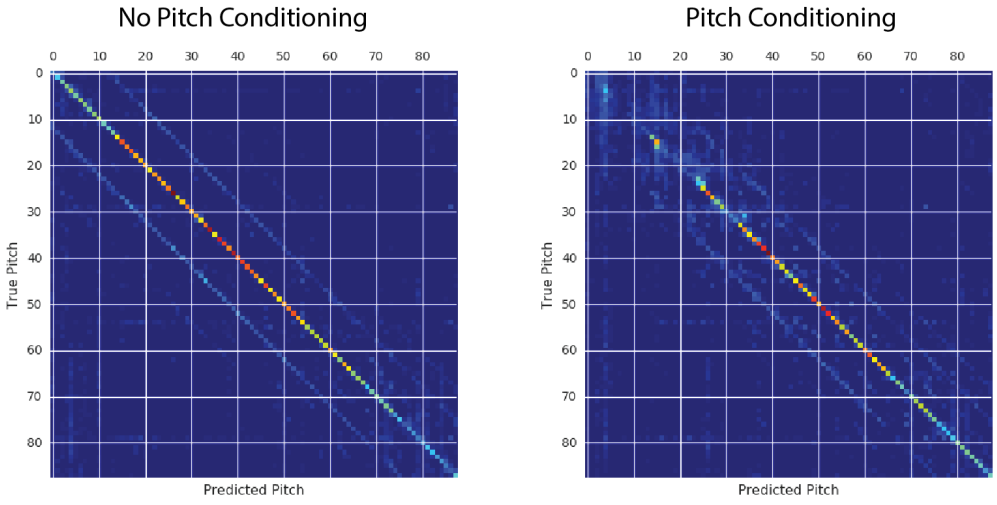

Figure 9. Confusion matrix for linear pitch classification model trained on embeddings from a WaveNet autoencoder. The prodominant error is predicting the wrong octave (being off by 12 tones). Training with pitch conditioning reduces the classifer accuracy.

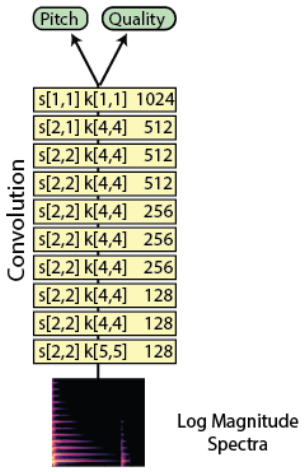

Figure 10. Model architecture for pitch and quality classification. Like the baseline encoder, each convolution layer is followed by batch normalization and a Leaky-ReLU (0.1 off-slope).