AMP-SPACE: A LARGE-SCALE DATASET FOR FINE-GRAINED TIMBRE TRANSFORMATION

Jason Naradowsky

Preferred Networks
Tokyo, Japan
narad@preferred.jp

ABSTRACT

We release Amp-Space, a large-scale dataset of paired audio samples: a source audio signal, and an output signal, the result of a timbre transformation. The types of transformations we study are from blackbox musical tools (amplifiers, stompboxes, studio effects) traditionally used to shape the sound of guitar, bass, or synthesizer sounds. For each sample of transformed audio, the set of parameters used to create it are given. Samples are from both real and simulated devices, the latter allowing for orders of magnitude greater data than found in comparable datasets. We demonstrate potential use cases of this data by (a) pre-training a conditional WaveNet model on synthetic data and show that it reduces the number of samples necessary to digitally reproduce a real musical device, and (b) training a variational autoencoder to shape a continuous space of timbre transformations for creating new sounds through interpolation.

1. INTRODUCTION

Major advances in machine learning often accompany the development of important data resources. With state-of-the-art results increasingly being achieved by general models trained on large amounts of data[1], it is often necessary to develop new datasets to drive study into new areas. In the domain of musical instrument audio, a notable example is NSynth[1], a dataset of 305,979 short sound samples from a diverse set of instruments and pitches. The size of NSynth helped it serve as a suitable testbed for the burgeoning field of neural generative audio models, and aided in the development of temporal auto-encoders[1], disentangling factors for audio generation[2], and methods of timber transfer[3]. A newly released dataset, synth1B1[4], uses a software implementation of modular synthesis to generate an additional *billion* samples of synthesized audio, allowing for such research on an even greater scale.

However, with the exception of these two examples, to the best of our knowledge we are unaware of any other comparably large-scale datasets of musical sounds (Table 1). And while the aforementioned datasets are suitable for the study of sample-based audio, there are other types of musical sound creation for which large-scale datasets do not exist. Consider that the typical way of interacting with sound samples is through a trigger: an event occurs, typically described by a pitch and velocity, and the note develops in response to it. Naturally the instruments most associated with sample-based music creation are keyboards and drums, where the method of interaction (the press of a key or the hit of a pad) provide all the necessary conditioning information (pitch and velocity) to shape the resulting sound. But many instruments are not played in strictly in this manner, such as stringed instruments, where the player continuously shapes the sound during its production. Describing the sound of these instruments using only the pitch and velocity at the onset of the note would result in the loss of much of the nuance and expressiveness of the performance. How can we develop ML technologies for players of these instruments? As all of these instruments are controlled in different ways, we turn our attention to the attribute they all share: the production of a waveform.

		Devices	Audio (hours)
Real	*This Work*	effectors	>50
	Schmitz [5]	effectors	<5
Mixed	NSynth [1]	instruments	333
Synthetic	*This Work*	effectors	>500
	synth1B1 [4]	synthesizer	1111111

Table 1: Amp-Space (v1) dataset by the numbers. While we were not able to obtain the Schmitz dataset[5], from the data description we estimate that it would contain less than 5 hours of audio.

We introduce **Amp-Space**, a large-scale dataset of *paired* audio samples. One sample is the waveform produced by a stringed instrument (here, an electric guitar) and serves as the conditioning information the player provides in lieu of pitch and velocity. The other is the result of processing the original waveform using a black box musical tool or device. The types of devices we study are those commonly used to produce the variety of guitar sounds found in popular music. This includes devices such as amplifiers, stompboxes (fuzzes, distortions), and studio effects (compressors), which can be used to drastically sculpt the initial sound, to the extent that musicians are able to create their own signature sound despite working with essentially the same basic components. Outside of a musical context, the differences in samples are often more subtle than those found in existing datasets. Figure 1 illustrates the way in which three perceivably distinct sounds yield comparatively similar spectrograms. Amp-space is designed to be a benchmark for modeling timbre transformations which are fine-grained and proven to be musically useful.

Like NSynth, Amp-Space can be used to study instrument timbre. But while samples of a single instrument in NSynth vary only in terms of their pitch and velocity (with timbre remaining rela-

[1] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

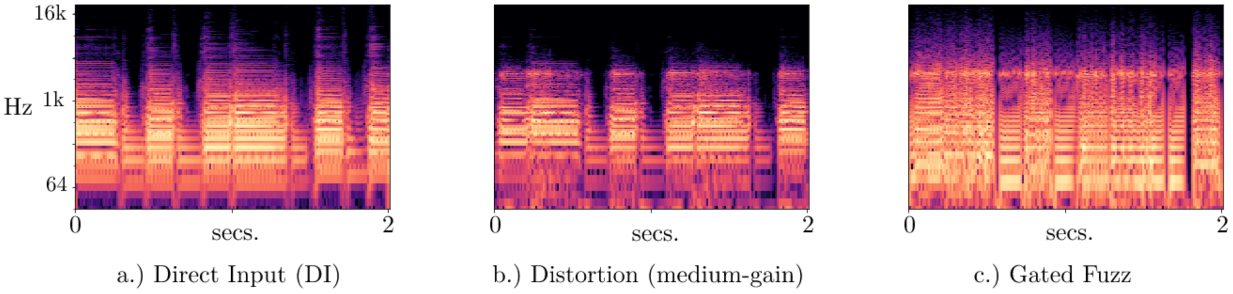

Figure 1: Spectrograms of the original instrument audio (a), and of the transformed sounds after processing with a music effect (b and c).

tively consistent otherwise), each device in our dataset is highly parameterized and capable of a wide range of sounds. This introduces problems unique to this task, namely that the combinatorial nature of device settings creates an intractably large space of sound. Even a relatively simple three-knob device would itself offer 1,000 different timbres if discretizing the values of each knob (1-10). In order to record sufficiently many samples of the devices contained in the dataset, we utilize a combination of programmatic manipulation of software device emulations, automated controls on real devices (such as motorized faders), and robotic servo manipulation on actual physical devices. This enables us to release the largest dataset of its type, containing over 500 hours of synthesized audio and 50 hours of samples recorded from physical devices (Figure 1), all in high-fidelity 44.1kHz monaural sound. We provide the settings of these devices, as well as any known variables which may affect the sound (voltage, transistor type, recording information, etc.), as an annotation with each pair of sounds.

Amp-space supports research into a variety of problems, including but not limited to:

- Accurate and/or real-time modeling of a wide variety of analog music devices, due to its abundance of real device samples.
- New interfaces for exploring the sounds of these devices, such as moving the position of a point in a projection of the vector space, due to its large size.
- Predicting device parameters and source waveform from audio samples, due its inclusion of parameter annotations.
- Generating new sounds by extrapolation, or by interpolating between samples from two different devices.

In Section 3 we describe the data collection procedure. We then present two experiments (Section 4 and 5) utilizing the unique properties of the data. First, we demonstrate the effectiveness of pre-training on synthetic data by showing that it can significantly reduce the data burden when adapting these models to real data via transfer learning. Second, we show the potential for using the data to construct a continuous space via variational auto-encoders, enabling new sound creation via interpolation in the vector space.

2. RELATED WORK

2.1. Modeling of Amplifiers and Effects

Many approaches have been proposed for modeling audio circuits. At the most fine-grained level, substantial literature exists on modeling circuit components with traditional DSP methods[6], including JFETs[7] and other parts commonly used in music device circuits. Abstracting slightly, gray box approaches model sections of circuits with blocks of either linear models (LTI filters) or function-specific non-linear models, depending on the behavior of the circuit section[8].

In our work we propose an entirely black box approach to modeling, in which the inner workings of the devices are completely abstracted away, and replaced with a single function, in this case a deep neural network. Previous work[9] has compared different neural methods of guitar amplifier model, using typical feed-forward, CNN, and LSTM networks, on a comparatively small set of recorded guitar amplifiers[10]. Other work examines the use of RNNs and WaveNets for real-time generation[11]. Our work extends on this, and in Section 4.1 we perform a parameter search over benchmark models in high-fidelity audio synthesis (WaveNets, SampleRNNs, etc.). A dilated convolutional architecture with EMD loss is the best performing system on our data, but we find that many architectures are capable of achieving similar quality when trained on large data with good hyperparameters.

2.2. Vector Spaces of Timbre and Sound

In Section 5 we utilize a subset of Amp-Space to build a continuous vector space of timber transfer sounds. While such vector spaces are an increasingly active area of research, such spaces also have a long history in audio research. Early studies of timbre relied on visualization techniques like multidimensional scaling (MDS) to arrange sounds in a vector space in according to their perceptual differences[12, 13]. In order to obtain the relative differences of points in the space, human participants were asked to score the similarity of pairs of sounds. With the advent of deep learning-based approaches it became possible to utilize the latent spaces induced by a neural model trained to perform a sound prediction task, replacing human judgements as the organizing metric. Other work has combined the two approaches, using data from human studies to regularize the latent space of neural models[14].

An important consequence of constructing spaces with predictive models is the ability to synthesize sound from points in the latent space. Dilated convolutional architectures, of the kind explored in this work, are a common choice for predicting the waveform directly[1]. Other options include recurrent networks, or predicting spectrograms[15]. A unifying theme of existing work is how sound is generated from the latent vector, often conditioning on other attributes such as pitch or velocity, and generating the entire envelope of the timbre across time (sample-based generation). Our work differs in this regard in that the model can condition on an aligned waveform during generation. Thus it is more closely related in methodology to work in speaker voice transfer[16] or singing voice conversion[17], where linguistic features or an input waveform provide temporal information rather than encoding it in the latent space. This distinction means that the latent space can be solely devoted to encoding timbral characteristics of the sound, and eschews the need to model time-sensitive dynamics. Indeed, we find that a relatively small receptive field is capable of achieving high accuracy on the task (Section 4.1.1).

2.3. Datasets of Conditional Audio

Amp-Space is not the first dataset proposed for wave-conditional modeling. Previous work on amplifier or effects modeling often included a small dataset of similar input/output waveforms for the purpose of evaluation. In addition, another dataset of paired DI and output from real guitar amplifiers was developed previously[5], consisting of samples of 5 amplifiers, across 10 stages of gain (other parameters held fixed). In comparison, Amp-Space is orders of magnitude larger both in terms of hours of audio and the diversity of timbre represented, scaling the goals of previous work to the size of the largest instrument audio datasets.

It could also be argued that synth1B1[4] is also a dataset of wave-conditional audio. Although it is not presented in such a manner and a conditioning wave is not provided, the waveform produced by the oscillators could be considered as analogous to the DIs used in this work. If the waveform of the oscillator is extracted and paired with the waveform generated by the following synthesizer modules, synth1B1 may be used as a complementary dataset for wave-conditional modeling of synth sounds.

field		**value**
DI	ID	ola_rhythm
	start	0:53
	end	1:03
DEVICE	name	5150-blockletter
	brand	Peavey
	source	real
	pre-amp	5150-blockletter
	power-amp	5150-blockletter
	loadbox	torpedo-reload-contour-5
	cabinet	none
	channel	lead
	low	0.6
	mid	0.6
	high	0.6
	gain	0.6
	master	0.3
	presence	0.5
	power-tubes	6l6-Sylvania
	power-tube-type	6L6
	bias	unknown
	watt	120W
	VAC	117V
	wall-voltage	103V
	time-delay	1.79ms

Table 2: A sample annotation. Some fields which are not pertinent to this device are omitted for clarity.

3. A DATASET OF WAVE-CONDITIONAL TIMBRE TRANSFER

The dataset collection process can be summarized as (1) collect raw guitar audio (known as *direct input* or DI) from recordings of professional musicians and choose segments to reflect a diverse set of DI sounds, (2) run the DI signal through a software simulation or physical device, recording the output signal, and (3) periodically adjust the parameters of the device, storing the parameter setting and recording as a single item in the dataset. All audio is recorded at 44.1kHz, 24bit, monaural.

DIs were obtained from professional recordings with permission of the artist, and divided into short samples, each containing a distinct music passage with no overlap with other samples. Samples were chosen to maximize diversity in pitch, dynamics, and style. The average length of a DI sample was approximately 10 seconds, and was chosen as a compromise between the brevity necessary to potentially record thousands of samples of each device, while being sufficiently long enough for a human to be able to evaluate it musically.

3.1. Large-Scale Synthetic Data

Improvements in modeling musical circuits in software have led to the production of many high-quality component-level models of amplifiers and other musical devices. In a component-level model, the circuit of the device is traced, and each component is individually measured and modeled. As a consequence, component models are time-consuming and costly to create, but the process ensures that each controllable parameter behaves similarly to how it would on the real device. We acquire several commercially available component models of musical devices (amplifiers and stompbox-style effects) to use as synthetic data generators.

Data processing for the synthetic data is done in Reaper[18], a digital audio workstation (DAW) with a full-featured API. Using Reaper's advanced scripting capabilities we automate changes to the parameters of the software device (such as gain, bass, mid, treble, presence, master, and bright/gain boost, etc.), sweeping through parameter changes after every clip. We discretize each parameter, valued 1–10. While this level of discretization still yields hundreds of thousands of training samples per device, rendering samples of a software device can be performed faster than real-time, making it tractable to produce thousands of hours of audio for each device if necessary.

3.2. Sampling Physical Devices

While software simulations of devices have improved significantly in recent years, accurately reproducing physical devices remains a challenge. We acquire a number of physical devices, including 15 popular tube amplifiers and an additional 15 solid state devices. Additionally, some devices are capable of reproducing a number of traditional circuits. Using a recording interface, a DI is sent to the device and its output is returned through the interface and recorded. If the device is an amplifier, the output signal from the device is first sent to a loadbox, in place of a traditional cabinet, which removes the cabinet, micing, and room acoustics from the dataset examples.

In contrast to software collection, all recording of physical devices must be done in real-time. We accomplish this through various means. For some devices, digital control of device parameters is possible (here via MIDI) and can be automated from a computer. On devices without digital control, if the knobs are attached to typical potentiometers it is sometimes possible to automate interaction with the device by attaching specially-purposed servos to the shaft of the potentiometers. In these cases, some device parameters can be manipulated from a computer via MIDI. For the remainder of the devices, the knobs are operated manually. A natural consequence of manual operation is far fewer samples of each device. In this cases we prioritize settings that sound most pleasing by our own judgement.

3.3. Annotation

Each sound sample is annotated with the label of the DI track (and corresponding start and end time) used to record it. From the device, all adjusted parameters and their values are labeled. Properties of the device type are also annotated to facilitate learning generalities of these properties. This may include the tube-type or wattage for an amplifier, or transistor type or bias for a fuzz pedal. An example annotation is provided in Table 2.

3.4. Release Notes

Due to large size and diversity of the dataset, we deviate from standard dataset release practices in two ways. First, because there are many research questions which might be pursued using this dataset, we choose not partition the data into a standard train, development, and test set. Section 4 illustrates how the dataset might be partitioned in different ways to create test sets of varying difficulty levels. Second, we release all samples recorded from real devices, but for practical purposes and because the synthetic data is essentially unbounded, we limit the amount of synthetic data samples we release. We instead make available the scripts and DIs necessary to recreate the data, or generate entirely new samples. Further details can be found on the data release page[2], together with the materials for data generation).

4. EXPERIMENT 1: SIM-TO-REAL TRANSFER

A long-standing hurdle to developing black box models of musical devices is the sheer number of different possible parameter settings on a typical device. If a human is required to change the device parameters manually then the process of recording the device may become very time-consuming. Therefore most research in this area focuses on recreating only a small number of possible sounds per device. However, results from the fields of computer vision[19] and natural language processing[20] demonstrate the effectiveness of pre-training a model on a large set of easily obtainable data before fine-tuning it on additional (typically less available) data for the desired task. We question whether the same strategy could hold true for audio generation, pre-training on the synthetic data partition from our proposed dataset.

Our task here is the same in both conditions (given an input waveform and parameter settings, predict the output waveform), but transferring from sim-to-real has been shown to be a difficult problem and may require additional modifications to work successfully[21]. However, if the model can gain some predictive understanding of how a particular control operates across many synthetic devices via pre-training, it may be able to generalize how these controls should behave on a real device from only a few samples. This would in turn reduce the burden of extensive data collection needed in order to recreate a real device.

We begin by performing an architecture search for effective models of timbre transfer on the Amp-Space Dataset. We identify the best performing model and train it on synthesized data, before fine-tuning it on limited samples from a real device, and evaluate its ability to predict the target waveform of the real device as the parameters are adjusted away from observed values.

4.1. Models

As part of the preliminary exploration of the dataset, we perform a large scale architecture and hyperparameter search on the timbre-transfer task. We define the task as follows. Input x is the clean DI signal, and output y is the transformed signal. We then quantize y into μ-law 256 bins. A contextual feature vector $c \in \mathbb{R}^{|F|}$, where $|F|$ is the size of the feature set, represent the device settings and other attributes included in the annotation fields. Knob settings are represented as real-values $[0, 1]$, while attributes are represented as one-hot vectors concatenated together length-wise.

We compare wave-to-wave variants of RNNs, LSTMs[22], Dilated LSTMs[23], SampleRNN[24], and WaveNet[25]. Within each model type, hyperparameters are searched using a flexible and highly-parallelizable hyperparameter optimizer[26], and include important characteristics of the problem, such as the model receptive field, depth, type of nonlinearity, and optimization method. For succinctness, we describe in detail only our best performing model architecture.

4.1.1. Wave2WaveNet

The best performing model in our architecture search is a dilated convolutional architecture which we refer to as Wave2WaveNet, mainly to emphasize that the expected input and output of the model differ from that of WaveNet (and to avoid confusion with later discussion which includes WaveNets), and that generation does not occur in a truly autoregressive manner as there are no dependencies on past *outputs*. As in a traditional conditional WaveNet[25], features c are fed in as an auxiliary input and concatenated to to each layer of the network. A network depth of 20, with a filter size of 3, yields a receptive field of 4093 samples. This is relatively short for a WaveNet, supporting the hypothesis that the properties of wave-conditional synthesis differ significantly from those of text-to-speech. Difficult long-distance problems of high-fidelity real audio prediction, such as phase information, are directly observable from the input waveform.

A novel finding gleaned from hyperparameter search is an improved loss function. As described previously, we discretize the target output (μ-law as used previously in WaveNet) representing it as a one-hot vector, transforming the problem into one of classification. However, as the classification task in this case is ordered, i.e. predicting a value adjacent to the target value is better than predicting one further away, we find that standard cross-entropy loss function used in WaveNet[25] is not optimal. A more appropriate loss takes into account these inter-class distances. Earth Mover's Distance (EMD), which computes the optimal transport between the predicted distribution p over classes and the target distribution $\hat{p}$, is one such example.

We used a weighted EMD^2 loss[27], which can be computed efficiently in the 1-dimensional case as the L1 distance between

[2] https://github.com/narad/amp-space

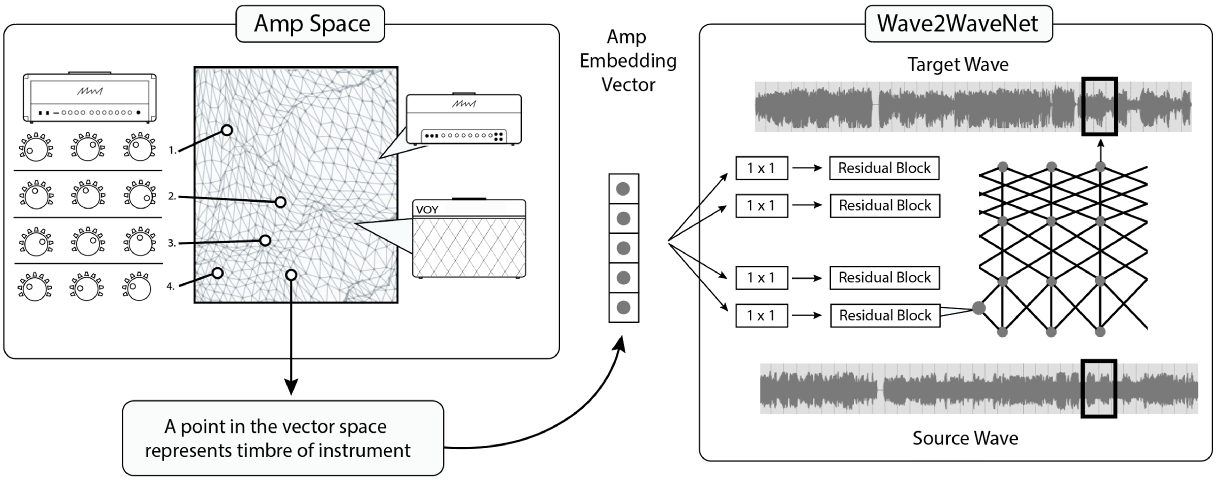

Figure 2: A hypothetical black box approach for conditional timbre-transfer of musical device sounds. Sound samples from numerous music devices and numerous settings are embedded into a single continuous vector space, and used as conditioning when generating new samples. This allows for interpolation between settings, and for the production of sounds not seen during training. In this example, a conditional Wave2WaveNet shapes the embedding space towards improving the task loss predicting the timbre transformation from DI signal to the signal recorded from the output of the device. We discuss the practicality of this design in Section 5.

the two distributions' CDFs[28]:

$$||\text{CDF}(p) - \text{CDF}(\hat{p})||_1 \tag{1}$$

As noted previously[27], this loss converges faster and is easier to optimize if the absolute value in the L1 distance is replaced with squaring. In addition, to combat the class imbalance present in the data, we weight the loss at every sample with label i by weight w_i, which is inversely proportional to the frequency of the label in the data N_i:

$$w_i \propto (\frac{1}{N_i})^\sigma \tag{2}$$

where σ is an optimization parameter.

During generation, we further leverage this loss and the properties it imposes over the predicted distribution by calculating the expected value rather than using the argmax of the distribution, which (a) improves performance, and (b) allows generation of any continuous value in the range $[-1, 1]$, which negates the theoretical minimal error which would be imposed by the discretization. These changes yield significant performance improvements over our implementations of previously proposed models[11].

4.2. Results on Gain Interpolation

For the purpose of this experiment we focus on one of the most important traditional parameters of timbre control in devices like amplifiers or distortion pedals: gain. Adjustments to gain vary the amount of clipping and compression in the output signal, but the precise nature of this effect varies significantly between devices. For pre-training we sample a random synthetic device (which has a gain control), and train on samples of the device until convergence (∼18hrs on 2 Nvidia V100 GPUs). During training device parameters are sampled randomly, while the gain parameter is swept across its full range of values. For fine-tuning, we select a real device, and fine-tune with samples of a restricted range of gain settings (each real device sample is approximately 120s). As gain is represented as a real-value in the feature vector c, it can be modified even in models which are not exposed to different gain values during training. We experiment with which gain settings are seen during fine-tuning, and what gain setting(s) are used for evaluation (Table 3). Our aim is then to test how well the fine-tuned model is able to generalize to gain settings unseen during fine-tuning on the real device, but whose general properties may have been learned to some extent during pre-training on synthetic data. We contrast this with the same Wave2WaveNet model architecture, trained solely on the fine-tuned data.

Fine-Tune Gain	**Test Gain**	**MSE (e-03)**	
		Baseline	Pretrain
5	5	0.05	0.005
5	$\{4, 6\}$	0.33	0.023
5	$\{3, 7\}$	0.81	0.017
5	$\{2, 8\}$	2.10	0.064
$\{3, 7\}$	5	0.08	0.011
$\{2, 8\}$	5	0.15	0.004

Table 3: Gain Interpolation Accuracy. Fine-tune Gain is the set of gain settings sampled the target device recording and given to the model for additional training. Test Gain is the set of gain values which the model must infer and generate new audio.

Table 3 shows the results. First we compare (top row) a model pre-trained on synthetic data and then fine-tuned on samples from a single gain setting (gain=5), against a baseline model trained solely on the fine-tuned data. Both models are then tasked with converting new DI samples to match the timbre of the fine-tuned data. We measure an improvement in mean squared error (MSE) when using the pre-trained model, but neither model makes errors which are easily perceivable. This finding supports the results of previous work which indicate that dilated convolutional networks are capable of high accuracy on similar supervised timbre transfer tasks[10, 11]. However, it is interesting to note that pre-training does appear to provide some advantage even in the purely supervised scenario when data is limited. The Wave2WaveNet is a highly parameterized model, and it is possible that pre-training provides a better initialization of the model weights in comparison to the baseline model, which must learn these weights from scratch.

As the model is asked to generalize further (Table 3, rows 2-4), the pre-trained model performs significantly better in all scenarios. By the point where the model is tasked with generalizing to gain settings ≥ 2 points further from the original gain=5 setting observed in the fine-tuning data, the performance of the baseline model has decreased by nearly two orders of magnitude. The difficulty of the task also has an effect on the pre-trained model, but only to the extent that it performs comparably to the baseline model in the fully supervised setting. We believe this is an important result as it establishes that large-scale pre-training on synthetic device audio is beneficial to digitally recreating real devices, and demonstrates the potential for pre-trained black box models to infer missing parameter values.

5. EXPERIMENT 2: A CONTINUOUS SOUNDSPACE

In Section 4 we experimented with pre-training techniques for the dilated convolutional architectures commonly used for audio generation in the waveform domain. By adjusting the real-valued features in the conditional Wave2WaveNet model, we were able to interpolate (or extrapolate) sounds not seen during training. However, the parameter space implied by the conditioning features may be arranged in a way which is disjoint, and there is no explicit pressure for interpolations in these values to produce realistic audio. To examine this issue further we perform another experiment where we interpolate between two amplifier models.

Consider a collection of audio samples from two amplifiers A and B, denoted Y_a and Y_b, and corresponding to the input DI X, which is the same 4-second clip of audio for all samples in this dataset partition. The first model for comparison is the same Wave2WaveNet model where the device type feature is $[0, 1]$ for amplifier A and $[1, 0]$ for amplifier B. We train on the provided data and predict an interpolation of amplifier A and B using the conditional device type feature values $[0.5, 0.5]$, analogous to the approach used in Section 4, while holding other parameters fixed.

We compare this to a variational autoencoder (VAE) model that explicitly regularizes the latent space. Traditional autoencoders are a type of neural model consisting of an encoder and decoder, which aim to learn a compressed representation of the input while being trained to accurately construct the input. VAEs extend AEs by assuming the data is generated by a directed graphical model $P(z|X)$, and is approximated by the network. We use the typical isotropic multivariate Gaussian prior, and optimize the negative log-likelihood of the reconstruction with regularization which encourages the latent space to be more continuous.

In order to train the VAE efficiently, we opt to utilize Mel spectrogram representations (with 2048 FFT bins, 256 Mel filters, and a hop length of 256). Mel spectrograms have been utilized in many timbre-transfer models, or in circumstances where timbral characteristics of a sound sample (for instance, a speaker) are used elsewhere in the model. Using spectrograms in this scenario requires that all data samples be aligned (as y is used as input and output; the DI waveform is ignored), whereas the Wave2WaveNet need only aligned input/output for each instance, and thus it is worth noting that this method is not suitable for generalizing to timbre transfer on new DI waveforms. However, it allows us to utilize a simpler model architecture. The VAE encoder consists of four layers of 1D convolutions and pooling operations (the decoder contains analogous 1D deconvolutions), predicting the mean and variance of the latent distribution (3-dimensional) from an encoded representation of size 512 for each factor. The spectrograms predicted by the model are converted to waveforms for evaluation via the Griffin-Lim method.

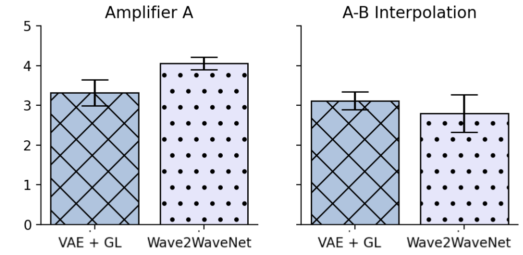

Figure 3: Mean opinion scores for perceived quality of modeling amplifier A, and modeling the interpolation of amplifier A and B in identical settings.

5.1. Results

We perform a small user study, soliciting the participation of 10 musicians from an online guitar community (average experience with music creation was approximately 8 years). We present four waveforms in random order over an online form. Participants were asked to score the quality of each on a scale from 1 to 5, unaware of the method which generated them. The results are presented as mean opinion score (MoS) in Fig. 3. Unsurprisingly, the Wave2WaveNet outperforms the spectrogram-based VAE model on modeling timbre transfer in the supervised setting. Griffin-Lim is known to perform poorly on predicted spectrograms, and it is possible that some of this performance drop could be mitigated by training a WaveNet-based vocoder specifically for this domain.

In the interpolation setting, the Wave2WaveNet model quality suffers significantly, while the strength of the VAE model at interpolating between sounds becomes apparent. Neither VAE-based result is of exceptionally high quality (lower than Wave2WaveNet on in-domain data), but it is more consistent across both scenarios. While this is only a preliminary finding, it is supported by other work on variational timbre spaces (Section 2.2) and hints at the notion that a single latent space could represent encodings of multiple timbre transformations, where the types of transformations common in musical devices could explored and created by operations in a vector space.

6. DISCUSSION

6.1. Future Directions

We have shown the usefulness of this dataset in two separate scenarios. Other lines of research which may benefit from such data are discussed below.

Remaining challenges for modeling musical devices. In terms of accuracy, we notice all models evaluated in this work have difficulties in some scenarios. In particular, gated fuzzes seem especially problematic. Notably the bias of the transistors in these circuits is set precisely for them to function like a threshold, which can create a sound sometimes described as “choppy” or as a “sputter”. Such sounds are uncommon but represent a challenging goal to address in future models. In order to aid researchers in identifying remaining challenges, we provide a table of the performance of our model on individual device categories on our data release page.

Novel interfaces for music creation. A main motivation in developing this work is to enable new interfaces for controlling traditional music device sounds. Existing device controls are inspired more by their role in the circuit than by the user’s goal, which makes their precise meaning differ from one circuit to another. For instance, a “mid” control for adjusting middle frequencies may be defined in an entirely differ manner from one device to the next, both in terms of what frequencies are affected, its Q-function, and the effect on other knobs or frequencies (interactive EQ). Such inconsistencies increase the learning curve to effectively operate these devices and craft the desired sounds. We showed how one model, a VAE, could interpolate between observed sounds to create new ones, but gaining a better understanding of this latent space is left for future work. As more data is used to train the model, an interesting question is how sound is arranged in the latent space and if the salient dimensions in its organization reflect common device controls, or perhaps other more abstract characteristics which are important to musicians. Similarly, if a user finds a sound in the latent space, the model can teach the user how to achieve it on the device via nearest neighbor search.

Timbre and equalization. A choice was made in this work to model each sound sample strictly as a black box function of its parameters and input waveform. However, a great deal of the sonic diversity found in this dataset is the direct result of changes in the EQ parameters on the devices. This raises the question of whether equalization is better handled as a pre-processing step, where the transformation from clean signal to modeled output signal first adjusts for the EQ of the target sound, then applies the timbre transfer function, thus factorizing EQ out of the latent space.

Circuit design in the Latent space. In early work in distributed semantics with neural language models[29], algebraic operations in the vector space were shown to capture meaningful relationships. For instance, the vector operations *king - man + woman* predict approximately the vector for *queen*. Could the same types of regularities arise in vector spaces of sounds? If so, new devices could be designed by movements in the vector space. The Amp-Space dataset provides two important characteristics for supporting research on this topic: (1) sound samples are annotated with individual components whose contributions to the overall sound may be learned by the model (e.g., power tube type in a vacuum tube amplifier), and (2) sound samples taken at various places along the signal chain. For instance, multiple samples might be taken from amplifiers with similar poweramps, isolating the effect of the preamplifier in the chain. If an understanding of the sonic properties of each major component can be isolated, they may also be interchanged, and new circuits defined in this manner.

6.2. Shortcomings

We present Amp-Space as a useful resource for the study of music effects and fine-grained timbre transformations, but it is by no means complete. The space of music device timbre is far too vast to incapsulate in a single fixed resource, and for practical purposes some decisions were made that may ultimately warrant new samples. For instance, the use of a small fixed set of DIs could not cover the scope of possible techniques and frequencies, and could limit modeling accuracy in situations which differ significantly from the training data. The importance of diversity in source waveforms in device modeling is not yet well understood. The use of a single loadbox, a single interface, and other possible confounding factors of dataset creation currently pose no issues, but may need to be remedied in the future. However, a guiding principle of this work is that the generalization power of large neural models and big data will help to ensure that existing data and resources are still useful even if future versions are adapted to address new concerns.

7. CONCLUSIONS

In this paper we present Amp-Space, a new large-scale dataset for the study of fine-grained music timbre transformations. In comparison to other music audio datasets it is a unique resource due its combination of unprecedented size, its focus on *conditional* transformations, and its close connection to the tools of professional musicians. Importantly, the way we define and partition the synthetic component of this dataset is only one such example of how it might be constructed. We release all data generation code, enabling researchers to instantiate the data in different ways to address the specific needs of each study.

We present two experiments in which we leverage the size of the dataset, first to pre-train conditional generative models of audio, and then to construct a rich embedding space of conditional sound transformations. The former approach solves a problem of practical importance, demonstrating that synthetic data and the pre-train/fine-tune paradigm can be utilized effectively when replicating a physical device from limited data. This may enable black box models to be constructed more efficiently than hand-engineered component models. We hope this dataset facilitates further research into these areas.

8. ACKNOWLEDGMENTS

We are greatly indebted to Philip Korsunsky for work on the ReaScripts and baseline models, and Ola Englund and Lorcan Ward for contributing DI audio. Thanks to everypixel and users przembek10 and Collectif Intro of thenounproject for some of the artwork used in this paper. Many thanks also to to Daiki Higurashi, Brian Vogel, Steven Cosgrove, Fabrice Matulic, and the anonymous reviewers for helpful comments on drafts of this work.

9. REFERENCES

[1] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1068–1077.

[2] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[3] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[4] J. Turian, J. Shier, G. Tzanetakis, K. McNally, and M. Henry, "One billion audio sounds from GPU-enabled modular synthesis," in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx)*, 2021.

[5] T. Schmitz and J.-J. Embrechts, "Introducing a dataset of guitar amplifier sounds for nonlinear emulation benchmarking," in *Proceedings of the Audio Engineering Society Convention 145*, Oct 2018.

[6] M. Holters and U. Zölzer, "Automatic decomposition of nonlinear equation systems in audio effect circuit simulation," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx)*, Vienna, 2017.

[7] L. Köper and M. Holters, "Taming the red llama—modeling a cmos-based overdrive circuit," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx)*, Vienna, 2020.

[8] F. Eichas and U. Zölzer, "Gray-box modeling of guitar amplifiers," *Journal of the Audio Engineering Society*, vol. 66, no. 12, pp. 1006–1015, December 2018.

[9] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, 2020.

[10] T. Schmitz and J.-J. Embrechts, "Objective and subjective comparison of several machine learning techniques applied for the real-time emulation of the guitar amplifier nonlinear behavior," in *Proceedings of the Audio Engineering Society Convention 146*, Mar 2019.

[11] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-Time Guitar Amplifier Emulation with Deep Learning," in *Appl. Sci*, 2020, vol. 10, pp. 251–256.

[12] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *Acoustical Society of America Journal*, vol. 63, no. 5, pp. 1493–1500, May 1978.

[13] J. A. Burgoyne and S. McAdams, "A meta-analysis of timbre perception using nonlinear extensions to CLASCAL," in *Computer Music Modeling and Retrieval. Sense of Sounds*, Berlin, Heidelberg, 2008, pp. 181–202, Springer Berlin Heidelberg.

[14] P. Esling, A. Chemla–Romeu-Santos, and A. Bitton, "Generative timbre spaces with variational audio synthesis," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx)*, Vienna, 2018.

[15] J. Kim, R. Bittner, A. Kumar, and J. Bello, "Neural music synthesis for flexible timbre control," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2019.

[16] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proceedings of Interspeech*, 2017, pp. 1118–1122.

[17] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, "PitchNet: Unsupervised singing voice conversion with pitch adversarial network," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[18] I. Cockos, "Reaper," `https://www.reaper.fm/`, 2006–2020.

[19] J. Deng, R. Socher, L. Fei-Fei, W. Dong, K. Li, and L.-J. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 06 2009, vol. 00, pp. 248–255.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186.

[21] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2017)*, 2017.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," in *Advances in Neural Information Processing Systems 30*, pp. 77–87. Curran Associates, Inc., 2017.

[24] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Arxiv*, 2016.

[26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[27] D. S. L. Hou, C.P. Yu, "Squared earth mover's distance loss for training deep neural networks on ordered-classes," in *NeurIPS workshop on Learning on Distributions, Functions, Graphs and Groups*, 2017.

[28] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: some insights from statistics," in *IEEE ICCV*, 2001, vol. 2, pp. 251–256.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.