A STUDY OF CONTROL METHODS FOR PERCUSSIVE SOUND SYNTHESIS BASED ON GANS

António Ramires *

Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
firstname.lastname@upf.edu

Jordan Juras

Native Instruments GmbH.
Berlin, Germany
firstname.lastname@native-instruments.de

Julian D. Parker

Native Instruments GmbH.
Berlin, Germany
firstname.lastname@native-instruments.de

Xavier Serra

Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
firstname.lastname@upf.edu

ABSTRACT

The process of creating drum sounds has seen significant evolution in the past decades. The development of analogue drum synthesizers, such as the TR-808, and modern sound design tools in Digital Audio Workstations led to a variety of drum timbres that defined entire musical genres. Recently, drum synthesis research has been revived with a new focus on training generative neural networks to create drum sounds. Different interfaces have previously been proposed to control the generative process, from low-level latent space navigation to high-level semantic feature parameterisation, but no comprehensive analysis has been presented to evaluate how each approach relates to the creative process. We aim to evaluate how different interfaces support creative control over drum generation by conducting a user study based on the Creative Support Index. We experiment with both a supervised method that decodes semantic latent space directions and an unsupervised Closed-Form Factorization approach from computer vision literature to parameterise the generation process and demonstrate that the latter is the preferred means to control a drum synthesizer based on the StyleGAN2 network architecture.

1. INTRODUCTION

Sound synthesis techniques for percussion sounds have evolved significantly throughout the last decades, with many techniques being associated with and even defining entire Electronic Music genres. In early times, timbral characteristics of percussion were obtained by modifying the acoustic properties of the instruments themselves – for instance, the use of different shell configurations or materials for constructing drums, or different shapes and alloys for cymbals. Analogue synthesis paved the way for creating and designing percussive sounds electronically. Drum machines such as the Roland TR-808 generated sounds by combining synthesised tones with white noise, which provided novel drum timbres [1]. These drum machines became a staple in a variety of music genres including Hip Hop, House and Techno. More recently, with the development of music-making software such as Kick2[1], SubLab[2] and the default drum synthesizers available in Digital Audio Workstations, digital drum sound creation became common practice for music makers of all backgrounds, enabling advanced digital signal processing techniques to be applied to drum sound design.

Recent advances in Deep Learning introduced novel methodologies for synthesising data. Instead of relying on experts for designing systems to generate specific kinds of data, these methodologies are data-driven: the algorithms learn how to represent the distribution of data on which they are trained. Architectures such as Autoregressive Networks [2, 3, 4], Variational Autoencoders (VAEs) [5], and Generative Adversarial Networks (GANs) [6] have all been proven to generate high-quality results in a variety of domains, from images of human faces to musical audio. Besides achieving the best synthesis quality in several domains, recent work has shown that GANs can even capture high-level semantic concepts [7] in the latent space dimensions driving the networks. Nevertheless, determining the correct latent space feature to manipulate to achieve a specific variation in the data space can be cumbersome – especially when dealing with high-dimensional latent spaces. To overcome this issue, new techniques have been proposed to find directions in the latent space that correspond to semantic concepts, either in a supervised [8] or unsupervised manner [9].

Recently, research into percussive sound creation using generative deep learning models has been receiving increased attention. Both DrumGAN [10] and Adversarial Synthesis of Drum Sounds (ASDS) [11, 12] employed the GAN training paradigm for this task. Each proposed its own methodology for controlling the synthesis, conditioning on timbral features, and the drum class respectively. However, neither of these studies evaluated user preference between their respective approaches to controlling the generation process.

The main goal of our research is to evaluate 3 different methodologies for navigating the latent space of a GAN trained to generate drum sounds. To this end, we adapted StyleGANv2 [13]

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Nº 765068, MIP-Frontiers.

[1] sonicacademy.com/products/kick-2
[2] futureaudioworkshop.com/product/sublab/

to accommodate the dimensionality of the time-frequency representation of musical audio and trained the network on a private dataset of drum sounds. Both a supervised [8] and unsupervised [9] method were applied to the trained network to find perceptually salient directions in the latent space. Finally, a user study based on the Creativity Support Index [14] was employed to compare these two latent navigation approaches against a simplistic approach to latent space vector manipulation.

2. GENERATIVE MODELS FOR DRUM SOUND SYNTHESIS

The earliest work on drum sound synthesis using Deep Learning is the Neural Drum Machine [15]. It coupled a Conditional Wasserstein Auto Encoder [16] trained on the magnitude component of the spectrogram of percussive sounds together with a Multi-Head Convolutional Neural Network for reconstructing the audio from the spectral representation. Principal Component Analysis (PCA) was used on the low-dimensional representation learned by the autoencoder to select the 3 most influential of the 64 embedding dimensions. These were provided to the user as a control interface. However, these user-controllable parameters are abstract and were not shown to be perceptually or semantically meaningful to the goal of parameterising the generation process. The data used for training this network was a proprietary dataset of drum sounds, which could not be shared, making the work non-reproducible. In Neurodrum [17], a feed-forward neural network using a Wave-U-Net architecture [18] was conditioned on the AudioCommons timbral characteristics[3] to synthesize drum sounds. These features were identified from the 7 most relevant search queries used in Freesound[19] and were calculated by combining existing high-level features. Some of these features are brightness, boominess and sharpness. The conditioning features allowed for reliable and intuitive control of the sound generation process for music makers while taking advantage of the fast generation characteristic of the Wave-U-Net. The main shortcoming of this approach was the audio quality of the generated data – which was far from the quality found in professional drum samples. A different control methodology for generating drums was presented in CRASH [20]. In this work, diffusion models and a conditional U-NET are used to enable interpolation on the noise of the latent space to produce sounds "in between" drum classes.

Two approaches based on GANs have been recently proposed: DrumGAN [10] and ASDS [11]. In ASDS, the authors trained a conditional Wasserstein GAN that learns to generate waveforms of drum sounds in high-resolution (44.1kHz). A label of the desired drum sound was used as a conditioning signal (for example 'kick', 'snare', or 'cymbal'), and generation control was based on the corresponding 3-dimensional embedding space learned by the network. Despite the network's high-resolution output, some audio artefacts were still present in the generated audio. DrumGAN, on the other end, employed the same conditioning scheme as Neurodrum – high-level timbral features, but on a Progressive Growing GAN [21]. This network was trained on the spectrograms and was able to generate both the real and the imaginary components of this representation. This permitted the use of the Inverse Short Term Fourier Transform to return output spectrogram data to the audio domain. The resulting sounds were of very high quality, despite the use of a 16kHz sampling rate. However, the coherence between the input values in the control signal and the resulting analysis of the output was lower than that achieved by the Wave-U-Net approach.

Despite the significant research effort in this area, no studies have yet compared the various approaches made to controlling the synthesis process itself. In this work, we compare an unsupervised approach – where synthesis control is determined based on the learned weights of the network itself, against a supervised approach – where features determined from data generated by the trained network are used to infer synthesis control. Since existing pre-trained GAN networks rely on private data and are trained with conditioning features, we train a new StyleGAN2-based model for this study.

3. STYLEGAN2

As its name implies, StyleGAN2 is a flavour of GAN – a neural network architecture that exploits adversarially training independent generator (G) and discriminator (D) networks. The input to the former are one-dimensional vectors $\mathbf{z}$ from a latent space Z - an i.i.d. Gaussian probability distribution, while the latter is input with one-, or multi-dimensional, vectors of data $\mathbf{x}$ from the data space X, which represents all real data instances. The generator is tasked with learning a mapping between $p_z(z)$, and $p_{data}(x)$, the probability distribution of all training data samples. The discriminator is a classifier that ideally scores real data examples (training data) with a score of $D(X) = 1$, and generated data examples with a score of $D(G(Z)) = 0$. Thus, the discriminator wishes to maximize the probability of assigning the correct label to both training data and generated data. For training data, it is trained to maximise the expected value over all instances in X:

$$\mathbb{E}_{X\sim p_{\text{data}}}[logD(X)] \tag{1}$$

Likewise for generated data, the discriminator is trained to maximise the expected value over all generated fake instances $G(Z)$:

$$\mathbb{E}_{Z\sim p_z}[log(1 - D(G(Z))] \tag{2}$$

while the generator works to minimize Equation 2. Adversarial training amounts to a *two-player minimax* game between the generator and discriminator networks:

$$\begin{aligned}\min_G \max_D V(D,G) := \mathbb{E}_{X\sim p_{\text{data}}}[\log D(X)] + \\ \mathbb{E}_{Z\sim p_z}[\log(1 - D(G(Z))]\end{aligned} \tag{3}$$

StyleGAN2 is part of a lineage of generative models developed by the NVIDIA research team. Initially motivated by stabilising the training process for GANs, Karras et al. proposed ProGAN – a new model architecture and corresponding training procedure that 'progressivly' trained layers of a deep convolutional GAN against different downsampled resolutions of its training data [21].

StyleGAN radically revised the deep convolutional GAN architecture, by redefining the functional relationship between latent space vectors and the generator network. [22]: (i) As a means to disentangle possible non-linear subspaces within the normally distributed latent space $\mathbf{z} \in Z$, a learned intermediate latent space, or *style space*, $\mathbf{w} \in W$, was introduced. Connected to the latent space by a non-linear mapping, $f : Z \rightarrow W$, the style space doesn't have to support sampling according to any fixed distribution; (ii) Instead of feeding the latent vector directly to the generator network like in ProGAN, StyleGAN fed the generator with

[3] github.com/AudioCommons/timbral_models

a fixed seed and applied the style vectors in $\mathbf{w}$ across each layer of the generator through an affine transformation. This effectively applied the affine-transformed style to each level of resolution in the network, influencing coarse features at lower resolution layers, and fine-grained features at higher resolution layers.

Improving on ProGAN's unsatisfactory performance in generating stochastic image features (hair, background foliage, pores, etc.), StyleGAN further introduced noise to each resolution layer of the generator, scaled by a learned weight. While the network generated state-of-the-art images, artefacts from the training procedure – notably shift-invariance, were left as open questions for future work.

StyleGAN2 introduces several improvements on the original StyleGAN architecture [13]. To address 'blob'-like artefacts that were common in generated StyleGAN images, StyleGAN2 replaces inter-layer normalisation (Adaptive Instance Normalisation (AdaIN), which independently normalises both the mean and variance between adjacent convolution layers) with what Karras et al. refer to as weight demodulation. In general, the goal of inter-layer normalisation is to remove the statistics of the applied style vector, $\mathbf{w}$, from the output feature map. However, by normalising both the mean and variance between layers, information discovered by the network about the magnitudes of the features relative to each other is potentially destroyed – which is speculated as a culprit for the 'blob'-like artefacts. Weight demodulation is proposed as a 'weaker' means to normalise than AdaIN (and respectively Pixel-wise normalisation in ProGAN), since it is based on statistical assumptions of the signal passing through the layer rather than the actual contents of the feature map – which thus preserves relative magnitude information between layers.

Furthermore, the StyleGAN2 network is no longer trained progressively: it was shown that with a large enough training dataset, the gradient updates applied to the network during training are roughly in line with how ProGAN trains – ie. early training focuses on lower resolution layers, and progressively fine-tunes higher resolution layers; and without the risk of shift-invariant image generation.

The many innovations of StyleGAN2 led us to believe that it would be a well-suited architecture to synthesize drum sounds. Notably, given the many different drum classes present in our dataset, the disentangled nature of the network architecture's *style space* presents the potential for latent conditioning suitable for coherent interpolation between drum classes. Furthermore, the trainable stochastic noise components fed to each network layer are well suited for the task of generating the noise components common to drum sounds – from kick drum transients to sustained hihats and cymbals.

4. CONTROLLING THE GENERATION

To allow a degree of control over the synthesis process in generative models, several approaches have been proposed. Previous research on percussive sound generation used conditioning features which, based on an external conditional signal c, force the model to learn the conditional probability $p(x|c)$. The chosen conditioning signal c can vary in terms of how much information it contains, from low information signals like class labels (e.g. kick, snare or cymbal), to very rich conditioning signals like the envelope of the drum sound.

In this paper we compare supervised and unsupervised approaches for finding latent directions in the learned latent space of GANs. While conditioning can be used as a control for the generation process, we want to create a fair comparison between approaches. We therefore focus on approaches that do not require constraining the network during training and can be applied directly to a pre-trained model. Our goal is to find latent directions $n \in \mathbb{R}^d$, with some interpretable meaning, which allow the modification of a sound $G(z)$ with latent code z to a new sound $G(z') = G(z + \alpha n)$ where α represents the amount of modification.

Unsupervised methods rely on applying dimensionality reduction to the trained latent space to find the directions n that correspond to the most significant change in the output. Early experiments [23] relied on generating data from points in the latent space and posterior application of PCA to discover the directions. In this work, we use SeFa [9], a closed-form factorization method that does not require sampling and can learn directions directly from the weights of the trained model. The paper shows that given any latent code z, and the weight matrix A, the edit operation $G(z')$ can be achieved by adding the term αAn on the projected code. Therefore, A contains all information related to the output variation. The basis for finding the latent directions in SeFa is to solve the optimization problem:

$$n = \underset{n \in \mathbb{R}^d : n^T n = 1}{\arg\max} ||An||_2^2 \tag{4}$$

where $||.||^2$ denotes the l_2 norm. The solutions for this problem are shown to be the eigenvectors of $A^T A$ with the largest eigenvalues. This method showed remarkable performance when applied to the pre-trained StyleGAN [22] model for generating faces, being able to identify the directions corresponding to pose, presence of glasses, gender and amount of smiling, in a more disentangled manner than PCA.

Supervised methods for discovering latent space directions require an annotation procedure on synthesised data to train classifiers in the latent space. In InterfaceGAN [8], it is assumed that, for binary characteristics, there is a hyperplane in the latent space which separates positive and negative examples. To find the hyperplane, a large amount of data needs to be generated from the trained network to be later classified using classification algorithms. The authors experimented with classifiers for the pose, smile, age, gender and eyeglasses to get positive and negative labels for the generated data and used Support Vector Machines (SVMs) to then find the dividing latent space hyperplane for each characteristic. The direction n that encodes a characteristic to modify is therefore a normal vector of the discovered hyperplane, which passes by, z, the latent code we want to modify. It is also shown that the larger the magnitude of the modification α is, the more affected the sample is according to the encoded direction – despite n being found through a binary classification hyperplane. This algorithm also has shown remarkable performance when editing faces in terms of pose, smile and age.

5. METHODOLOGY

5.1. Dataset

The network was trained on the entire corpus of one-shot drum samples included with Native Instrument Maschine Expansion releases. Table 1 shows the distribution of sample counts for each drum class in the dataset. These samples were all created by the Native Instruments sound design department throughout the last

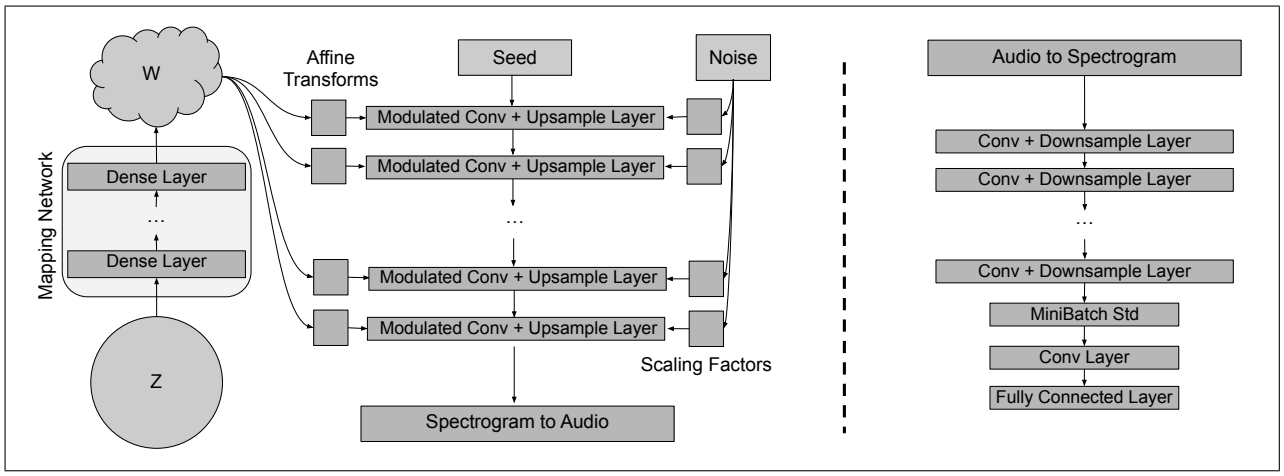

Figure 1: StylGAN2 Generator (left) and Discriminator (right) architectures.

decade. As a result, there is an inherent consistency across the dataset in terms of quality, onset locations, and pre-normalised sample volumes.

Drum Class	Count	Drum Class	Count
Claps	1547	Combo	91
Cymbal	1270	Hand Drum	99
HiHat	4645	Kick	4025
Mallet Drum	5	Metallic	73
Percussion	3365	Shaker	1122
Snare	3332	Tambourine	3
Tom	1017	Wooden	36
Total	**20630**		

Table 1: Distribution of drum classes in the training dataset.

5.2. Data Pre-processing

As our network was trained on a two-dimensional spectrogram representation of our audio data sampled at 16kHz, data pre-processing was implemented as follows. (i) Audio samples were resampled to 16kHz and zero-padded to 16k samples, representing one second of audio data. (ii) A logarithmic 'fade-out' was added to the last 30% of each audio vector. (iii) Audio vectors were normalised to a floating-point range of [-1.0, 1.0]. (iv) Audio vectors were converted to complex spectrograms, using the following parameters: hop size of 512 samples, window size of 2048 samples, and an FFT size of 2048 samples. (v) Complex valued spectrogram reshaped into a 2 channel feature map of real and imaginary components per channel. (vi) Finally, the DC component of the 2 channel spectrogram representation was removed.

5.3. Model and Training

The default implementation of StyleGAN2 provided by NVIDIA was used, with some adaptations made to it to work with spectrograms of shape 1024×32: (i) The network was modified to handle rectangular shapes instead of only square data. (ii) The resolution of the smaller feature map in the generator was set to 64×2, which is doubled every layer. (iii) The network was adapted to handle only 2 channels, instead of the 1 or 3 channels commonly used for image generation. We use 5 synthesis blocks in the generator comprising a Modulated Convolutional and an Upsampling layer. On the discriminator 5 blocks comprising of a Convolutional and a Downsample layer are used. An overview of the complete network is presented in Figure 1.

The network was trained on a virtual Google Cloud Platform machine, using PyTorch's GPU library *Distributed Data Parallel* to train across 4 NVIDIA Tesla T4 GPUs. The latent space and style space dimensions were both set to 512, and the learning rate was set to $2\mathrm{e}{-3}$. A batch size of 8 examples was used. Although the stated GAN training objective is to arrive at an equilibrium point, where the discriminator outputs similar scores for real and generated data, in practice, the quality of data output from the network is typically maximised before the equilibrium is reached. A Frechet Audio Distance analysis was used to determine which training epoch corresponds to the highest quality and most diverse audio output [24]. Epoch 243, which corresponds to the network being exposed to 5,012,000 spectrograms, generated a minimum FAD score of 2.689. The code used to train and create the model, as well as audio examples for the reader to assess the audio quality are available on the accompanying website.[4]

5.4. User Interface

We want to evaluate how our drum generation model can help foster creativity when assisting music makers in creating drum sounds. To this end, we created a graphical user interface that allows generating random sounds, navigating the Z latent space, and also modifying sounds according to the directions learned by the SeFa and InterfaceGAN algorithms. The interface can be seen in Figures 2 and 3.

The first panel on the left of the user interface in Figure 2 represents the Z latent space. This is the first interface we evaluate. The user can set the value for each of the 512 dimensions by either drawing the vector with the computer mouse, or randomly seeding each latent dimension, and generate the corresponding audio output.

[4] aframires.github.io/stylegan2-ada-pytorch/

Feature	Boominess	Brightness	Depth	Hardness	Roughness	Sharpness	Warmth
Val. Acc.	100%	99.2%	99.2%	97.5%	100%	100%	100%
Test Acc.	70.9%	67.6%	72.7%	72.0%	52.3%	66.5%	81.8%

Table 2: SVM accuracy when separating positive and negative examples in InterfaceGAN.

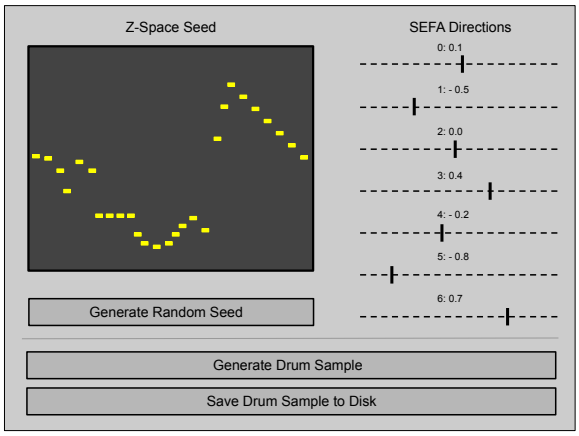

Figure 2: Graphical User Interface with SeFa directions.

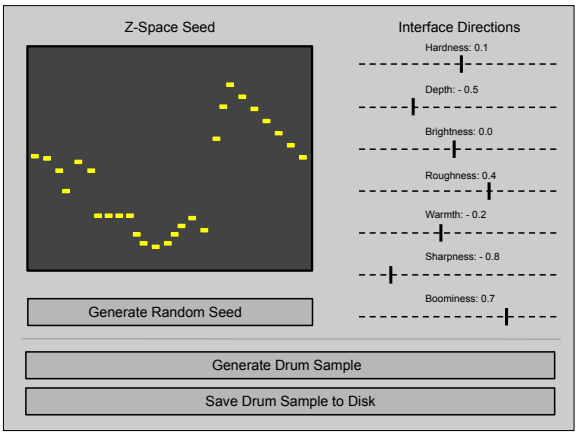

Figure 3: Graphical User Interface with InterfaceGAN directions.

The second interface and control methodology we want to evaluate are the 7 most significant directions returned by the unsupervised SeFa algorithm when applied to our trained model. The choice of 7 as the number of directions was to have the same number of control parameters as the timbral features used for InterfaceGAN. Similarly to the best-performing approach in the original SeFa paper [9], we use the latent semantic factorization algorithm on the W space. These parameters are exposed as 7 horizontal faders, as shown in Figure 2, and are labelled and displayed in decreasing order of importance.

The last interface to evaluate is the supervised directions produced by InterfaceGAN, shown in Figure 3. InterfaceGAN requires the generation of examples from the trained network, as well as posterior manual or automatic annotation. To this end, we generate 10000 percussion sounds from our network and annotate them with the descriptors computed from AudioCommons timbral models [25]. This set of 7 descriptors were obtained from analysing recurrent query terms related to timbral characteristics used for searching Freesound [19]. These features are hardness, depth, brightness, roughness, boominess, warmth and sharpness, and are calculated through regression models implemented in the AudioCommmons Extractor[5]. These exact descriptors have been previously used as conditioning features for controlling drum synthesis in previous work [17, 10].

To obtain the desired directions in InterfaceGAN, we use the latent embeddings in the W space, as these show higher classification accuracy in the SVM training stage [26]. For the SVM training, we used 280 training examples, 120 for validation, and the test set comprised the remaining 9600 samples. With this amount of data, the supervised algorithm was able to achieve a separation boundary which was able to separate negative and positive elements with decent accuracy as shown in Table 2. The parameters for the training, validation and test split were the ones used in the original InterfaceGAN article [26]. The high validation accuracy, accompanied with lower values for test accuracy might indicate that this split might not be the best, as there are a lot of test examples and the training data is fairly limited.

We provide examples of manipulating each of the 7 features from SeFa and InterfaceGAN in the accompanying website[4].

5.5. Evaluation

Ultimately, evaluating user control over the generation of drum sounds focuses on the extent to which a user can express creativity. While designing a user study for evaluating the different approaches to parameterising the StyleGAN2-based drum synthesiser, we determined that the Creative Support Index (CSI) is the most relevant tool. The CSI is a psychometric survey designed to evaluate the extent to which a 'creative support system' can assist a user engaged in creative work – in this case synthesising drums. The CSI measures six dimensions of creativity support: Exploration, Expressiveness, Immersion, Enjoyment, Results Worth Effort, and Collaboration. It allows researchers to evaluate how well a tool supports creative work overall and can pinpoint weaknesses in the various dimensions listed above. Table 3 provides an overview of example statements with which the test subjects are asked to rate from 'Highly Agree' (10) to 'Highly Disagree' (0), while they evaluate each of the three approaches to parameterisa-

[5] https://github.com/AudioCommons/ac-audio-extractor

Dimension	Statement Example
Exploration	*"It was easy for me to explore many different ideas, options, designs, or outcomes, using this system or tool."*
Immersion	*"My attention was fully tuned to the activity, and I forgot about the system or tool that I was using."*
Results Worth Effort	*"What I was able to produce was worth the effort I had to exert to produce it."*

Table 3: CSI example statements for 3 creativity support dimensions.

tion explored in the study: directly manipulating Z-Space, SeFA latent directions, and InterfaceGAN latent directions.

As a final step in the evaluation, test subjects are asked to complete a 'paired-dimension comparison', which assesses how each subject values (or weights) each of the dimensions of creativity support already evaluated in the rating scale section. With these weights, the CSI score is determined by:

$$
\begin{aligned}
&CSI_{score} = \\
&\quad (\text{CollaborationRating} \times \text{CollaborationWeigh} + \\
&\quad \text{EnjoymentRating} \times \text{EnjoymentWeight} + \\
&\quad \text{ExplorationRating} \times \text{ExplorationWeight} + \\
&\quad \text{ExpressivenessRating} \times \text{ExpressivenessWeight} + \\
&\quad \text{ImmersionRating} \times \text{ImmersionWeight} + \\
&\quad \text{ResultsWorthEffortRating} \times \\
&\qquad \text{ResultsWorthEffortWeight}) \ / \ 3.0
\end{aligned}
$$

6. RESULTS AND DISCUSSION

The evaluation was completed by 14 participants with various degrees of music production knowledge, from no music experience to professional music producer.

The results for the CSI evaluation are presented in Table 4. SeFa was the preferred interface by the participants, followed by the Z-space and the InterfaceGAN. SeFa has a clear preference with a margin of 4.14 in relation to the second best performing method. The results for InterfaceGAN and Z-Space are fairly similar, with a difference between the two of just 0.67. The low preference for InterfaceGAN could also be due to the low test-score obtained when finding the directions. However, when exploring this parameter space, the directions seemed to correspond to the desired attributes. Generally, participants reported having fun and were positively surprised by the ease of generating percussion sounds with each of the three techniques. Participants also commented on having enjoyed the exploratory process.

Z-Space	SeFa	InterfaceGAN
62.85 ± 11.91	$\mathbf{66.99 \pm 11.38}$	62.18 ± 11.40

Table 4: CSI scores for the 3 latent space navigation schemes under test.

Given the limited number of participants and their diverse backgrounds, the CSI scores unfortunately bear large confidence intervals and, therefore, these results cannot be said to be statistically significant. However, the trend towards favoring Sefa parameterisation in this exploratory study was further echoed in anecdotes from participants during the debrief.

By interacting with the SeFa latent directions, it was reported that they were clearly related to specific concepts in the data space. While the first two parameters controlled the drum class and the amount of noise content respectively, the following controls controlled finer characteristics such as the decay time, depth, and boominess. The last controls labelled 5 and 6 did not impart any consistent variation in sounds generated across different latent samples.

Furthermore, participants reported an interesting user experience while interacting with the SeFa controls: If they wished to create kick drums, they could simply tweak the SeFA sliders (likely the first two sliders influencing drum class characteristics) to produce a kick sample for the currently chosen latent vector in the Z-Space. Then, subsequent randomly seeded latent vectors generated kick drums with different timbral characteristics. The same was found for hi-hats, toms, and snares, but less easily for other percussion types. This is likely attributed to the former having the highest representation in the training dataset.

On the other hand, a few participants characterised some InterfaceGAN parameters as redundant and not orthogonal to each other. Some participants reported issues regarding a lack of consistency from one seed to the next and not understanding the semantic concepts behind the parameters. Although having labelled directions could be desired for some experienced participants, testers valued the potential for exploring new timbres using SeFa without the need for music production knowledge – for example, the terminology employed by the InterfaceGAN UI.

In Table 5, we present the accumulated participant weights resulting from the 'paired-dimension comparison' in the CSI study.

CSI Weight	Value
Collaboration	0.64
Enjoyment	2.43
Exploration	**3.64**
Expressiveness	**3.36**
Immersion	2.00
Results Worth Effort	**3.21**

Table 5: CSI weights for each dimension.

From these results, it can be seen that the participants mentioned Exploration, Expressiveness and Results Worth Effort as the most important dimensions of creativity support for generating drum sounds. The high importance for Exploration could be the reason as to why SeFa scores highly in the CSI scale, as this system allows a controllable but serendipitous exploration of the latent space. Enjoyment and Immersion were still important but not as significant as the previously mentioned ones. Collaboration was the least significant dimension with a weight of almost 0.

7. CONCLUSIONS

In this work, we evaluated three methodologies for designing and editing drum sounds using Generative Adversarial Networks. To this end, a StyleGAN2 network was adapted to work with audio data and trained on a large private collection of drum sounds. We adapted two methodologies that showed promising results in controlling the generation of images – SeFa and InterfaceGAN – to our use case. We compared these approaches against the unconstrained navigation of the latent space of the network through a user test based on the Creativity Support Index. Our user study found that the unsupervised approach SeFa performed better for creative engagement with the StyleGAN2 network and we described the advantages and disadvantages of each interface.

Avenues for future work include the research and development of characteristics and classifiers better suited for the task of drum synthesis, to further improve the supervised approach InterfaceGAN. Redoing the experiment in a more specific scenario (e.g. replicating a drum sound or exploring drum sounds to fit a composition) could lead to a more confident result. Furthermore, creative possibilities of the StyleGAN2 network such as style mixing and adjusting magnitudes of noise at each resolution layer of the network could be included in the latent direction analyses explored in this paper, to further enrich the quality of the resulting parameterisation.

8. REFERENCES

[1] Zainab Hasnain, "How the Roland TR-808 revolutionized music," Apr 2017.

[2] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016, p. 125.

[3] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 1068–1077.

[4] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. of the 5th International Conference on Learning Representations*, 2017.

[5] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

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

[7] Ceyuan Yang, Yujun Shen, and Bolei Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *International Journal of Computer Vision*, 2020.

[8] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou, "Interpreting the latent space of GANs for semantic face editing," in *CVPR*, 2020.

[9] Yujun Shen and Bolei Zhou, "Closed-form factorization of latent semantics in gans," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. 2021, pp. 1532–1540, Computer Vision Foundation / IEEE.

[10] Javier Nistal, S Lattner, and G Richard, "DrumGAN: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, 2020.

[11] Jake Drysdale, Maciej Tomczak, and Jason Hockman, "Adversarial synthesis of drum sounds," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2020.

[12] Jake Drysdale, Maciej Tomczak, and Jason Hockman, "Style-based drum synthesis with GAN inversion," in *Extended Abstracts for the Late-Breaking Demo Sessions of the 22nd International Society for Music Information Retrieval (ISMIR) Conference.*, 2021.

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of StyleGAN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. 2020, pp. 8107–8116, Computer Vision Foundation / IEEE.

[14] Erin Cherry and Celine Latulipe, "Quantifying the creativity support of digital tools through the creativity support index," *ACM Trans. Comput.-Hum. Interact.*, vol. 21, no. 4, jun 2014.

[15] Cyran Aouameur, Philippe Esling, and Gaëtan Hadjeres, "Neural Drum Machine : An interactive system for real-time synthesis," 2019.

[16] Ilya O. Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf, "Wasserstein auto-encoders," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[17] António Ramires, Pritish Chandna, Xavier Favory, Emilia Gómez, and Xavier Serra, "Neural percussive synthesis parameterised by high-level timbral features," in *Proc. of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 786–790.

[18] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, 2018, pp. 334–340.

[19] Frederic Font, Gerard Roma, and Xavier Serra, "Freesound technical demo," in *ACM International Conference on Multimedia (MM'13)*, Barcelona, Spain, 2013, ACM, pp. 411–412, ACM.

[20] Simon Rouard and Gaëtan Hadjeres, "CRASH: raw audio score-based generative modeling for controllable high-resolution drum sound synthesis," in *Proceedings of the 22nd*

International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021, Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy, Eds., 2021, pp. 579–585.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018, OpenReview.net.

[22] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 4401–4410, Computer Vision Foundation / IEEE.

[23] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris, "GANSpace: Discovering interpretable GAN controls," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, Eds., 2020.

[24] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, Gernot Kubin and Zdravko Kacic, Eds. 2019, pp. 2350–2354, ISCA.

[25] Andy Pearce, Tim Brookes, and Russell Mason, "Timbral attributes for sound effect library searching," in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*, Jun 2017.

[26] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou, "Interfacegan: Interpreting the disentangled face representation learned by GANs," *TPAMI*, 2020.