StyleWaveGAN: Style-based synthesis of drum sounds using generative adversarial networks for higher audio quality

Antoine Lavault
Apeira Technologies
Le Creusot, France
a.lavault@apeira-technologies.fr

Axel Roebel
UMR CNRS 9912
IRCAM, Sorbonne Université
Paris, France
roebel@ircam.fr

Matthieu Voiry
Apeira Technologies
Le Creusot, France
m.voiry@apeira-technologies.fr

***Abstract*—In this paper we introduce StyleWaveGAN, a style-based drum sound generator that is a variation of StyleGAN, a state-of-the-art image generator. By conditioning StyleWaveGAN on the type of drum, we are able to synthesize waveforms faster than real-time on a GPU directly in CD quality up to a duration of 1.5s while retaining some control over the generation. We also introduce an alternative to the progressive growing of GANs and experimented on the effect of dataset balancing for generative tasks. The experiments are carried out on an augmented subset of a publicly available dataset comprised of different drums and cymbals. We evaluate against two recent drum generators, WaveGAN and NeuroDrum, demonstrating significantly improved generation quality using two quality measures: first the Frechet Audio Distance and second a perceptual test.**

***Index Terms*—Percussive Sound Synthesis, Generative Models, Creative Interfaces**

I. Introduction

Drum machines are musical devices creating percussion sounds using analog or digital signal processing [1], [2]. The characteristic sound of this synthesis process contributed to their use in the '80s and their appreciation nowadays. However, these drum machines did not provide an extensive set of controls over the generation.

Following the success of deep learning, several generative processes for percussive sounds have been proposed in the recent years, and two approaches retained our attention. [3] used a generative adversarial network (GAN) for waveform generation with a conditioning on the type of drum, generating 0.3s at 44100Hz. There is also [4], where a GAN was trained to generate STFT of drum sounds, allowing them to generate 1s at 16kHz. Both of them used the progressive growing of GANs [5].

In this paper, we build upon the same idea of conditional synthesis using discrete and continuous controls, with time-domain generation like [3] with a style-based approach (SGAN) [6], [7]. We conduct our experiments on an augmented version of the ENST-Drums [8] dataset, containing kick, snare, toms and hi-hats and comprising about 120k samples amounting to 100 hours of recordings. To evaluate the quality of the model on this dataset, we are using the Fréchet Audio Distance (FAD) [9], in an attempt to obtain a reference-free automatic evaluation of the generated samples. We also performed perceptual tests on the generated samples to measure how the generated samples are perceived by human listeners.

All in all, our goal is to create an algorithm for drum sound synthesis suitable for professional music production. In other words, we expect good output quality, real-time generation and relevant controls. We will especially compare to a few networks whose performances are summarized in table I. WaveGAN [10] is the first use of a generative adversarial network (GAN) for temporal generation of audio and NeuroDrum [11] was the first to introduce perceptual features as part of the control for drum synthesis. DrumGAN [4] uses a GAN with perceptual features as part of the control for drum synthesis and finally [3] uses a GAN for drum sound synthesis at higher sample rate but with shorter duration than the previously mentioned networks.

Reference	Sample Rate	Duration
WaveGAN [10]	16kHz	1.1s
NeuroDrum [11]	16kHz	1s
DrumGAN [4]	16kHz	1.1s
Drysdale et al. [3]	44.1kHz	0.4s
Ours	44.1kHz	1.5s

TABLE I
COMPARISON OF STATE OF THE ART NEURAL DRUM SYNTHESIZERS

II. Model

A. Generative Adversarial Networks and StyleGAN

Generative Adversarial Networks (GAN) are a family of training procedures in which a generative model (the generator) competes against a discriminative adversary (the discriminator) that learns to distinguish whether a sample is real or fake [12].

Instead of using a vanilla GAN, we are using an evolution called StyleGAN [6], [7]. StyleGAN attempts to mitigate the entangled representation when using noise as latent and input of the generator. The key idea here is to use a *style encoding*,

a vector which is obtained through a mapping network and is then used to control (through an affine transform) every layer of a synthesis network.

B. Proposed architecture

Since StyleGAN was originally used for high-quality image generation, we have to modify it for direct waveform generation. In particular, we transform 2D convolution (3×3) into 1D causal convolutions (1×9) [13], the upsampling is done with an averaging filter before each convolution block in the synthesis network, the mapping networks has 4 layers instead of 8 and the loss function is WGAN-LP [14] (see figure 1).

We use the same number of filters, with respect to the depth, as StyleGAN2 [7]. Just like StyleGAN2 , the synthesis network uses input/output skips and the discriminator is a residual network.

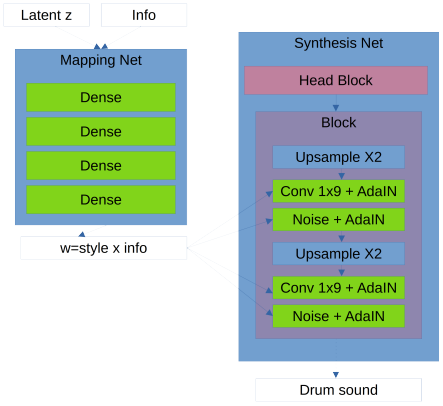

Fig. 1. StyleWaveGAN

In this work we follow [3], [11] using a temporal signal representation. Informal perceptual evaluations performed in the initial phase of this study supported our idea that the temporal representation produces better audio quality than spectral representation : we suppose it is because of the high amount of noise and the importance of the transient in the drum sounds.

C. Noise addition layers

We modified the noise addition layers of StyleGAN to make them style-dependant. We also add noise shaping (with a linear fade out) to avoid noisy tails. Having controlled noise addition is useful since some classes need more noise than other to get a good quality synthesis.

D. Output envelopes

One of the main complaints during informal perceptual tests for StyleWaveGAN was the generated sounds have an audible noisy tail which makes them easily identifiable. To avoid this pitfall, we added envelopes after the output of the network.

These envelopes where generated using the training dataset, one per type of drum. For each sample of one given type, the final envelope is the filtered mean of the analytical part of the Hilbert transform of these normalized samples. A small fade out is applied to avoid audible clicks at the end of the generated sounds.

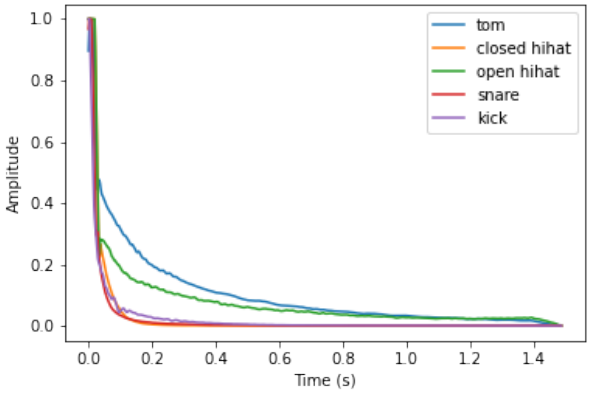

Fig. 2. Generated envelopes from the training dataset

E. Controlling the network

To control the network, we use 5 labels describing the type of drum (kick, snare, toms, closed hi-hat and open hi-hat). The labels are fed into an embedding layer which is then concatenated to the latent z (c.f figure 1) and fed to the mapping network. These labels are concatenated after the mapping network too. This is done to allow the user to control the synthesis network without the necessity of a mapping network.

In our experiments, we are using 5 labels. These labels are added to the network with a one-hot vector. We expect to have a better disentanglement between the class label during the style encoding by using this method.

F. AutoFade

While Progressive Growing of GANs [5] is used in recent papers [3], [4], we have to note it was dropped by its creators [7]. As a compromise, we introduce AutoFade. It is a ResNet architecture with a convolution path and a bypass where a learned parameter is used to fade more or less of one path. Rather than fixing a value like ResNet, we let the network choose the best value as part of the training process, without the need of training it block by block. If x and y represents the two different branches, we have:

$$\sin(\alpha)x + \cos(\alpha)y \tag{1}$$

α is independent of x or y. It makes this structure an intermediate between ResNet and Highway Networks. By using trigonometric function in equation (1), we guarantee the conservation of the standard deviation, if both inputs have equal variance. Informal listening tests showed that AutoFade was of great use in the discriminator, but did not bring any improvement when used in the generator. The Autofade feature will therefore be evaluated in the following sections, only as part of the discriminator.

III. Experimental setup

A. Dataset

We are using a subset of ENST-Drums [8], comprised of 350 samples of close miking of kicks, snares, toms and hi-hat. Since 350 elements is too low for a data-driven approach, we used an augmentation method similar to [15]. We used `SuperVP` [1] to process the original dataset. The modifications applied to the sounds consist of a gain applied to transient/attack components [16], noise components as well as independent transposition of the signal source and the spectral envelope.

The set of parameters is shown in table II. The limits have been obtained by means of subjective evaluation of the modified sounds aiming to avoid transformations that can be perceived as unnatural by a human listener. Examples are available in the supplementary material which can be found at https://alavault.github.io/stylewavegan_eusipco/.

As a supplementary metric, the Fréchet Audio Distance between the original dataset and the augmented one is 0.62.

Process	Parameters
Remix attack	0.1, 0.3, 0.6, 1.5, 2, 3
Remix noise	0.6, 1.5, 2, 3
Transposition	0, $\pm$100, $\pm$200
Spectral envelope transposition	0, $\pm$200

TABLE II
Augmentation operations and parameters

B. Training procedure

The training procedure is the same as StyleGAN 2 [7], except that we trained the network on 2M samples. With a batch size of 10, it totals to 200k iterations.

C. Imbalanced dataset

Balancing datasets is common in classification tasks but to our knowledge, has never been done for generation tasks. As shown in table III, our augmented dataset is quite unbalanced, so to obtain a balanced dataset, we use a sampler which takes elements from sub-datasets (one per label) at random according to a uniform distribution. We call it "equal-proportion sampling", even if it is a form of oversampling method.

Element	Proportion
Kick	3%
Snare	18%
Toms	45%
Closed hi-hat	10%
Open hi-hat	22%

TABLE III
Dataset population

[1] SuperVP is available free of charge in form of a Max/MSP object at https://forum.ircam.fr/projects/detail/supervp-for-max/

D. Baseline

In our evaluation, we will use NeuroDrum [11] and WaveGAN [10] as baselines when comparing using an objective measure, and we will use demo samples provided by the author of DrumGAN [4] as a baseline for a perceptual test. These demo samples have a sample rate of 16kHz.

Because NeuroDrum [11] works with 16kHz sample rate we adapted our model to use this sample rate for this comparison. We also compared with WaveGAN [10], a model with a similar training method, using our dataset with 44.1kHz. Here we configured both networks to generate 0.3s (@44.1kHz).

Unfortunately, DrumGAN is not reproducible because of missing source code or/and missing or unknown meta parameter. We also don't have any insights on the contents of the private dataset used for training DrumGAN, even if it sounds like synthetic drums were used. DrumGAN results are provided as a comparison to one of the most recent neural drum synthesizer.

The lack of available source code and meta parameters for [3] makes it unreproducible as well. However, we know the dataset they built for this task is comprised of synthetic samples since they aimed at drum sound synthesis for electronic music production.

E. Evaluation

1) Reference-free evaluation: We chose to use the Fréchet Audio Distance (FAD) [9], a reference-free evaluation metric for audio generation algorithms using a VGGish model trained on AudioSet. We compare the embedding of the augmented database to the embedding obtained from 64k samples generated by the evaluated network. In terms of computational cost, we achieve a generation rate of 52drum sounds$/s$ on one 1080GTX with the network in full resolution (1.5s@44.1kHz).

Network	FAD
Baseline [11]	25.35
StyleWaveGAN@16kHz	**11.48**

TABLE IV
FAD comparison to NeuroDrum [11] (lower is better)

Network	FAD
Baseline [10]	13.08
StyleWaveGAN (SWG)	**7.75**
SWG + AutoFade (AF)	**6.84**

TABLE V
FAD on networks without labels (lower is better)

Network	FAD
SWG + labels	6.85
SWG + labels + AF	6.72
SWG + labels + AF + B	6.65
SWG + labels + AF + B + Envelope	**3.62**

TABLE VI
FAD on label-conditioned networks (lower is better)

Class	SWG	SWG + AF + B	SWG + AF + B + Env
Kick	8.79	11.71	**3.58**
Snare	7.87	7.53	**4.29**
Tom	8.17	8.09	**6.27**
Closed HH	10.12	6.97	**4.23**
Open HH	8.26	8.91	**4.12**

TABLE VII
INTRA-CLASS FAD FOR LABEL-CONDITIONED STYLEWAVEGAN

Data	MOS	Cymbals	Kick	Snare
Real s	4.2 ± 0.3	4.1 ± 1.1	4.1 ± 0.6	4.4 ± 0.3
Augmented	3.8 ± 0.5	3.3 ± 1.3	4.0 ± 0.5	3.9 ± 0.5
SWG	3.5 ± 0.4	3.9 ± 0.7	3.0 ± 0.7	3.6 ± 0.8
DrumGAN	2.3 ± 0.5	2.3 ± 1.3	2.8 ± 0.6	1.6 ± 0.8

TABLE VIII
MOS ON DIFFERENT SETS OF SOUNDS, GLOBAL AND PER-LABEL (1 IS LOWEST, 5 IS HIGHEST)

2) perceptual testing: In our perceptual evaluation framework, we are evaluating the quality of generation among 4 sets of sounds : the original dataset, the sounds of the augmented dataset that are generated with the most extreme setting of the transformation parameters in table II, sounds from StyleWaveGAN at full resolution(1.5s at 44.1kHz) and sounds from DrumGAN (courtesy of Javier Nistal). The comparison of these 4 sets is motivated as follows: first, results obtained for the most extreme examples of the augmented dataset will provide a lower bound for our model, and second the evaluation of DrumGAN synthesis on their private dataset establishes a baseline which is proven to be better than NeuroDrum [11] in terms of FAD and claiming to be comparable to [10] in terms of perceived quality.

Instead of NeuroDrum or WaveGAN, we could use DrumGAN [4] or Drysdale et al. [3] as our perceptual test baseline. Both are unreproducible due to lack of available source code and metaparamaters. Drysdale et al. uses drum type conditioning when DrumGAN only has perceptual feature conditioning. However, we were able to get samples with drum type conditioning for DrumGAN through personal communication with the main author of [4]. Their performance differ, since DrumGAN can generate samples of 1.1s at 16kHZ, while Drysdale et al. can generate samples of 0.4s at 44.1kHz. We will see in the following discussion that the test participants indicate that the decay time is important to evaluate the realness of drum sounds, which gives an advantage to DrumGAN with its longer samples even if the sample rate is lower. Also, both of these solutions do not provide much insights about their training datasets. We know that Drysdale et al. focus on sample-based electronic music (EM), as described in their article. Samples used in EM are inherently synthetic and are built to sound different from real drums. Since our perceptual testing aims to evaluate how close the synthesis sounds like a real drum, having synthetic samples in the set will always be evaluated as worse.

Given the limitations discussed above, we selected DrumGAN as our baseline in our perceptual test. Its role is to represent one of the models of the state of the art.

The mean opinion score (MOS) is calculated as the average of the score given by the test participants. Part of the samples generated with StyleWaveGAN and used for the test are available in the supplementary material here.

Due to its low-risk nature, this evaluation didn't need an ethic approval from the host organization.

IV. EXPERIMENTAL RESULTS

A. Impact of our contributions

The first result we have is that we improved in terms of FAD (tables IV and V) when comparing to NeuroDrum and WaveGAN. We can also see from table V that using AutoFade in the discriminator helped at getting a better generation.

The results with dataset balancing are mitigated. It improved the supervised generation, as seen on table VI. However, without the label conditioning, using it didn't bring any decrease in the FAD : since it makes the training and evaluation dataset different (in proportions), the learned distribution differs, impacting negatively the FAD.

Envelopes on the output are the greatest contribution to the quality of generation in terms of FAD, almost halving it when comparing to StyleWaveGAN without the output envelope.

B. Results of perceptual testing

9 people took part on the test. The total Mean Opinion Score (MOS), with their confidence interval at 95%, is shown in table VIII as well as more detailed per-class results. Even if the number of participant is low, most of them (5 out 9) are audio professionals. Each of them were presented with 24 samples to evaluate along a scale going from "1-Poor" to "5-Real drum". Data was randomly picked among the original and augmented dataset as well as samples generated by StyleWaveGAN with a fixed label.

The score of the augmented samples is slightly lower than the real samples. This indicates that the extreme cases of our augmentation strategy are a bit too extreme. Here less extreme augmentation parameters with more intermediate values should be selected for future work. The main problem that can be found against the augmented samples comes from the pitch changes made by the augmentation process. The pitch change affect negatively the attack of the sound, making them sounding less natural than the real data. While the change is minor, it is sufficiently present to be perceived and graded worse than a real sample. Note however that these extreme parameter combinations remain rather rare in the full set of augmented sounds.

Comparing StyleWaveGAN to DrumGAN we can conclude that StyleWaveGAN trained on augmented data produces results that are perceived either similarly close (kick) or significantly closer (snare, cymbals) to real drums than DrumGAN trained on a drum dataset obtained from sources that are not further detailed in [4]. We conclude that the StyleWaveGAN model trained on augmented data achieves state of the art performance for drum synthesis.

We now discuss the StyleWaveGAN results in details. Given StyleWaveGAN was trained on the full dataset of augmented samples, a perfect model should produce results in between the test results of the real data and the extreme examples of the augmented data. We note that StyleWaveGAN achieves this performance only for the cymbals. Snare and kick synthesis remain less natural. A discussion with the participants of the perceptual tests reveals the following problems: for the kick drum sounds, the SWG model does not produce the characteristics long tail of the resonances and is also missing some energy in the frequency band below 100Hz. For snare drum synthesis the main problem appears to be the fact that SWG creates hybrids of sounds generated with sticks, mallets and brushes. Concatenating for example an attack of a snare sound obtained with a stick with a decay of a snare sound obtained with a brush creates fair sounding but unrealistic samples. These problems with kick and snare sounds indicate that the current implementation of the discriminator is not sufficient and further investigation will be required to improve the discriminator loss such that it avoids these perceptual problems.

V. Conclusion and future work

In this paper, we presented a new method for drum synthesis using StyleWaveGAN, an adaptation of a state of the art image generator. The proposed method has explicit controls on drum type to give some basic controllability.

We have shown the proposed style-based synthesis achieves a significantly reduced FAD compared to recent DNN based drum synthesis methods [10], [11]. perceptual tests also show that our network performs quite well in terms of perceived quality when comparing to processed and real data as well as DrumGAN [4]. To the best of our knowledge the proposed DNN is the first achieving drum synthesis with 44.1kHz sample rate (for sounds with a duration of 1.5s) with an inference speed more than 50 times faster than real-time on a consumer GPU: in 1 second we can generate 50 sounds 1.5 s long at described sample rate.

In terms of future work we will continue to work on the sound quality and additional controls for velocity as well as high-level control using perceptually relevant audio descriptors.

References

[1] G. Reid, "Practical Snare Drum Synthesis." [Online]. Available: https://www.soundonsound.com/techniques/practical-snare-drum-synthesis

[2] ——, "Practical Cymbal Synthesis." [Online]. Available: https://www.soundonsound.com/techniques/practical-cymbal-synthesis

[3] J. Drysdale, J.; Tomczak, M.; Hockman, "Adversarial Synthesis of Drum Sounds," *Proceedings ofthe 23rd International Conference on Digital Audio Effects (DAFx2020),*, no. September, pp. 24–30, 2020.

[4] J. Nistal, S. Lattner, and G. Richard, "DrumGAN: Synthesis of Drum Sounds With Timbral Feature Conditioning Using Generative Adversarial Networks," 2020. [Online]. Available: http://arxiv.org/abs/2008.12073

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," pp. 1–26, 2017. [Online]. Available: http://arxiv.org/abs/1710.10196

[6] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," 2018. [Online]. Available: http://arxiv.org/abs/1812.04948

[7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," dec 2019. [Online]. Available: http://arxiv.org/abs/1912.04958

[8] O. Gillet and G. Richard, "ENST-Drums: An extensive audio-visual database for drum signals processing," *ISMIR 2006 - 7th International Conference on Music Information Retrieval*, pp. 156–159, 2006.

[9] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 2350–2354, 2019.

[10] A. Dessein, N. Papadakis, and J. L. Rouas, "Regularized optimal transport and the rot mover's distance," vol. 19, oct 2018, pp. 1–53. [Online]. Available: http://arxiv.org/abs/1610.06447

[11] A. Ramires, P. Chandna, X. Favory, E. Gomez, and X. Serra, "Neural Percussive Synthesis Parameterised by High-Level Timbral Features," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May. Institute of Electrical and Electronics Engineers Inc., nov 2020, pp. 786–790. [Online]. Available: http://arxiv.org/abs/1911.11853

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," pp. 1–9, 2014. [Online]. Available: http://arxiv.org/abs/1406.2661

[13] 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," sep 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[14] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of Wasserstein GANs," sep 2017. [Online]. Available: http://arxiv.org/abs/1709.08894

[15] C. Jacques and A. Roebel, "Data Augmentation for Drum Transcription with Convolutional Neural Networks," 2019. [Online]. Available: http://arxiv.org/abs/1903.01416

[16] A. Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFx03)*, 2003, pp. 344–349. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161124