SIGNAL REPRESENTATIONS FOR SYNTHESIZING AUDIO TEXTURES WITH GENERATIVE ADVERSARIAL NETWORKS

Chitralekha GUPTA (chitralekha@nus.edu.sg) (0000-0003-1350-9095)[1],
Purnima KAMATH (purnima.kamath@u.nus.edu) (0000-0003-0351-6574)[1], and
Lonce WYSE (lonce.wyse@nus.edu.sg) (0000-0002-9200-1048)[1]

[1]**National University of Singapore**, Singapore

ABSTRACT

Generative Adversarial Networks (GANs) currently achieve the state-of-the-art sound synthesis quality for pitched musical instruments using a 2-channel spectrogram representation consisting of log magnitude and instantaneous frequency (the "IFSpectrogram"). Many other synthesis systems use representations derived from the magnitude spectra, and then depend on a backend component to invert the output magnitude spectrograms that generally result in audible artefacts associated with the inversion process. However, for signals that have closely-spaced frequency components such as non-pitched and other noisy sounds, training the GAN on the 2-channel IFSpectrogram representation offers no advantage over the magnitude spectra based representations. In this paper, we propose that training GANs on single-channel magnitude spectra, and using the Phase Gradient Heap Integration (PGHI) inversion algorithm is a better comprehensive approach for audio synthesis modeling of diverse signals that include pitched, non-pitched, and dynamically complex sounds. We show that this method produces higher-quality output for wideband and noisy sounds, such as pops and chirps, compared to using the IFSpectrogram. Furthermore, the sound quality for pitched sounds is comparable to using the IFSpectrogram, even while using a simpler representation with half the memory requirements.

1. INTRODUCTION

In recent years, GANs have achieved the state-of-the-art performance in neural audio synthesis, specifically for pitched musical instrument sounds [1, 2]. Engel et al. [1] showed that a progressively growing GAN [3] can outperform strong WaveNet [4] and WaveGAN [5] baselines in the task of conditional musical instrument audio generation achieving comparable audio synthesis quality and faster generation time. Nistal et al. [2] further showed that a 2-channel input representation consisting of the magnitude and the instantaneous frequency (IF) of the Short-Time Fourier Transform (STFT) achieves the best synthesis results in this framework compared to other kinds of representations, such as Mel spectrogram, MFCC, and Constant-Q Transform. The derivative of unwrapped phase of a signal with respect to time is equal to the angular difference between the frame stride and signal periodicity, and is commonly referred to as the instantaneous frequency (IF). Estimation of IF provides comprehensive information about the phase of the signal when the audio is pitched, i.e. has components that are clearly separated in frequency. Thus, a magnitude spectrogram combined with the estimated IF results in high-quality reconstruction of the signal for pitched signals such as musical instruments. In broadband and noisy short duration signals, components are not separated in frequency, and neighboring frequency bins have complex and highly interdependent amplitude and phase relationships that are necessary for reconstruction and the representation is very sensitive to IF estimation errors.

DrumGAN [6] extended the work in [2] to various drum sounds, however the authors have notably not used the IF spectrogram that produce state-of-the-art quality for pitched sounds, but instead, use spectrograms of the real and imaginary parts from the STFT directly. They also use a set of perceptually correlated features more appropriate than pitch for conditioning the percussion sounds in the target data set.

Průša et al. [7] proposed a non-iterative phase reconstruction algorithm called Phase Gradient Heap Integration (PGHI) that uses the mathematical relationship between the magnitude of Gaussian windowed STFT and the phase derivatives in time and frequency of the Fourier transform to reconstruct the phase using only the magnitude spectrogram. Marafioti et al. [8] compared three different GAN architectures, and showed that for a dataset consisting of spoken digits and piano music, the architecture using PGHI produced audio of objectively and perceptually higher quality than the other representations they compared based on an aggregate set of different signal types. A direct comparison with GanSynth [1] which was being published at about the same time was also not included in their study.

In this paper, we study and compare the state-of-the-art GanSynth with magnitude spectrogram+IF audio representation and reconstruction method and the PGHI method of representation and reconstruction for a systematically organized collection of audio textures such as pitched musical instruments, noisy pops, and chirps, spanning a range from pitched steady-state to broadband signals. We show

that the PGHI method of reconstruction from GAN estimates is more robust for synthetic spectrograms and estimation errors for different kinds of input signals than the state-of-the-art magnitude+IF representation. This study contributes to the development of general and efficient representations for training GANs for complex audio texture synthesis.

2. AUDIO TEXTURES AND REPRESENTATIONS

2.1 Audio Representations and Inversion Techniques

Many algorithms learn to estimate the magnitude spectrogram and then use iterative methods such as Griffin-Lim [9] to estimate the phase and reconstruct the time domain signal. However, these traditional methods of phase estimation and reconstruction are known to have perceptible artifacts in the reconstructed signal. Estimation of phase is difficult and prone to errors in part because artificial or manipulated images may not produce a real-valued time domain signal when inverted.

Another way of representing phase is with instantaneous frequency. The estimate of magnitude spectrogram and IF in frequency domain can be used to reconstruct a time domain signal by computing the unwrapped phase from the cumulative sum of IF across time axis, and computing an inverse Fourier transform. The state-of-the-art GANSynth framework [1,2] estimates this 2-channel audio representation, i.e. log magnitude and IF, or IFSpectrogram. Engel et al. hypothesized and showed that synthesized audio quality from the IFSpectrogram is robust to estimation errors for the NSynth dataset of pitched musical instrument audio while noting the importance of choosing analysis window sizes large enough to be primarily sensitive to a single frequency component. However, to the best of our knowledge, IFSpectrogram method has not been tested and compared to other representations for non-pitched and noisy sounds.

We observe that whether converting pitched instrument or noisy transient audio into IFSpectrogram representation, that resynthesizing produces a high quality audio output for both the kinds of sounds. However, if we add a small Gaussian noise to the IF channel (to simulate estimation error in IF) and then resynthesize, the perceptual quality of the pitched sounds is not affected as much as the quality of the noisy pop sounds. Audio examples of this simulation are presented in the companion website [1]. This indicates that IFSpectrogram method may not be robust to manipulated and synthetic spectrograms or estimation errors for non-pitched and noisy sounds.

For a signal composed of sinusoidal components with constant frequencies, the phase grows linearly in time for all the frequency channels that have energy in the spectrogram. For these frequency channels, the IF is constant and the local group delay (STFT phase derivative with respect to frequency) is zero. However, in case of an impulse train, the situation is reverse to that of sinusoidal components, wherein the phase derivative with respect to frequency axis will have more information than the IF as there is energy across almost all the frequency channels in the spectrogram, but the change of phase with respect to time exists only around the impulse events, and otherwise it is zero. Furthermore, for signals that have fast moving or closely spaced frequency components, IF does not capture the variability in the frequency direction.

The Phase Gradient Heap Integration (PGHI) method [7] is a non-iterative phase estimation method that exploits the mathematical relationship between the time and frequency derivatives of log magnitude spectrogram with the phase gradients in frequency and time axes respectively. To provide a brief summary here, Průša et al. [7] proved mathematically and experimentally that the derivative of phase along frequency axis $\phi_\omega(m,n)$ and, the derivative of phase along time axis $\phi_t(m,n)$ can be estimated solely from the time and frequency derivatives of log-magnitude of STFT ($\mathrm{s}_{\log_t}$, $\mathrm{s}_{\log_\omega}$) respectively computed with a Gaussian window, as [10, 11],

$$\begin{aligned}\phi_\omega(m,n) &= \frac{-\gamma}{2aM}(\mathrm{s}_{\log_t}(m,n)) \\ \phi_t(m,n) &= \frac{aM}{2\gamma}(\mathrm{s}_{\log_\omega}(m,n)) + 2\pi am/M\end{aligned} \tag{1}$$

where, M is the number of frequency channels, a is the hop size, and γ is the time-frequency ratio of Gaussian window, which is recommended to be aM/L, L being the length of the input signal in samples. Although the theory behind the non-iterative method of phase reconstruction from the STFT magnitude holds for Gaussian continuous window, Prusa et al [7] showed that the algorithm works well for a discretised truncated Gaussian window, however with the Gaussian approximation of other windows such as Hann and Hamming windows, they found significant signal degradation. Therefore in this work, we have used the truncated Gaussian window function. Redundancy between frames should be such that there is sufficient dependency between the values of the STFT to facilitate magnitude-only reconstruction. The recommended redundancy is $M/a \geq 4$ [8].

This method also implements a numerical integration of these phase gradients such that integration is first performed along the prominent contours of the spectrogram in order to reduce accumulation of the error, and so on. This heap integration method to estimate phase from the phase gradients helped to make the synthesis robust to estimation errors and noise [7, 10].

In this work, our goal is to investigate the quality of audio produced by a progressive GAN trained on a single channel log magnitude spectrogram and using PGHI for inversion of the estimated spectrogram to time domain signal and compare it to using the two-channel IFSpectrogram representation, for wideband, noisy, non-pitched or fast changing signals, as well as pitched instrument signals. With this framework, we propose a general approach for audio synthesis using the state-of-the-art GAN that works for a variety of different sounds.

[1] https://animatedsound.com/amt/listening_test_samples/#simulation

2.2 Audio Textures

Audio synthesis finds practical applications in creative sound design for music, film, and gaming, where creators are looking for sound effects suited to specific scenarios. Research in this field aims to learn a compact latent space of audio such that adjustments to these latent variables would help the creator search through a known space of sounds (eg. water drops and footsteps), parametrically control (eg. rate of water dripping) as well as explore new sounds in the spaces in between the known sounds [5].

Building upon generative adversarial image synthesis techniques, researchers exploring GAN techniques for neural audio synthesis have made significant progress in building frameworks for conditional as well as unconditional synthesis of a wide range of musical instrument timbres [1, 2]. These models are trained on NSynth dataset [12] that consists of notes from musical instruments across a range pitches, timbres, and volumes. Conditioning on pitch allows the network to learn natural timbre variation while providing musical control of notes for synthesis. The NSynth dataset provides a comprehensive representation of pitched sounds comprised primarily of well-separated harmonics. There has been some work on audio texture modeling for synthesis [13–15] including deep learning approaches [16], but audio textures have received considerably less attention than traditional musical sounds and speech.

Sound textures [13, 17] have more timbral variation including wideband or noisy components, such as footsteps or motors, and a wide range of temporal structure not found in pitched instruments. Furthermore, there can be very fast-varying frequency components and pitches in sounds such as water dripping, and chirps. Thus we examine the performance of controlled audio synthesis techniques on trained networks using three types of sounds - pitched instruments, noise burst pops, and frequency sweep chirps, as shown in Figure 1. In this work, we conduct experiments on pitched musical instruments and carefully controlled synthetic non-pitched and dynamic textures. More complex and natural textures are left for future study.

2.3 Conditional GAN architecture for audio synthesis

Parametrically controllable audio synthesis has also been an active field of research in recent years. Hsu et al. [18] used hierarchical variational autoencoders (VAEs) for conditional or controlled speech generation. Similarly, Luo et al. [19] learn separate latent distributions using VAEs to control the pitch and timbre of musical instrument sounds. Engel et al. [12] conditioned a WaveNet-style autoregressive model to generate musical sounds, as well as interpolate between sounds to generate new sounds. The current state-of-the-art performance in conditional synthesis of audio is the GANSynth architecture [1] which introduces a progressively growing Wasserstein GAN for controlled music synthesis and is based on the IFSpectrogram representation [2]. Thus, we adopt this architecture with IFSpectrogram representation as our baseline.

3. EXPERIMENTAL DETAILS

3.1 Audio Datasets

3.1.1 Pitched Musical Instruments

We make use of the NSynth dataset [12], that consists of approximately 300,000 single-note audios played by more than 1,000 different instruments. It contains labels for pitch, velocity, instrument type, acoustic qualities (acoustic or electronic), and more, although, for this particular work, we only make use of the pitch information as the conditional parameter. We use the same subset of this dataset as was used by Nistal et al. [2]. It contains acoustic instruments from the brass, flutes, guitars, keyboards, and mallets families, and the audio samples are trimmed from 4 to 1 seconds and only consider samples with a MIDI pitch range from 44 to 70 (103.83 - 466.16 Hz). This yields a subset of approximately 22,000 audio files with balanced instrument class distribution.

3.1.2 Noisy Pops

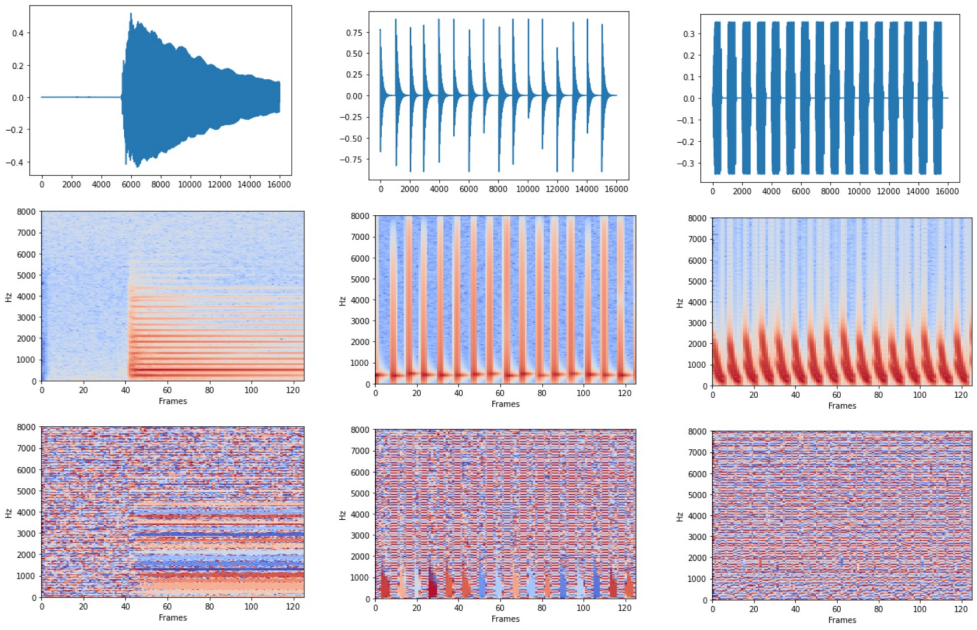
On the other end of the spectrum of sounds we tested are *pops*. A pop is a burst of noise filtered by a bandpass filter. We generated the pop textures with three parameters - rate (number of events per seconds), irregularity in the temporal distribution (using a Gaussian distribution around each evenly-spaced time value), and the center frequency of the bandpass filter. Rate ranges from 2 to 16 pops per second, center frequency ranges from 440 to 880 Hz (corresponding to midi pitch values 69 to 81), and irregularity described by a Gaussian distribution with a standard deviation ranging from 0.04 to 0.4. We generate 21 values for each of these three parameters, and five one-second long audio clips of each combination, resulting in a total of 46,305 ($21 \times 21 \times 21 \times 5$) audio files.

3.1.3 Chirps

In between the quality of the pitched sounds with relatively steady frequency components and the noisy pop sounds with sharp broadband transients are *chirps*. A chirp is a signal in which the frequency increases or decreases quickly with time. The chirps were generated with two frequency components space by an octave, and were controlled with 5 parameters - irregularity in time (like the pops), chirp rate (2 to 16 chirps per second, 9 samples), frequency sweep range in octaves indicating steepness of chirp ([-3. -1, 1, 3] where negative is descending and positive is ascending), event duration i.e. duration of each chirp in seconds (5 linearly spaced samples in [.02, .2]), and center frequency (9 linearly space samples in musical pitch space between 440 and 880 Hz). We generate 5 variations of each parameter (different due to the statistical distribution of events in time) resulting in a total of 40,500 ($5 \times 9 \times 4 \times 5 \times 9 \times 5$) audio files of 1 second each.

3.2 GAN architecture

We used the progressively growing Wasserstein GAN architecture [1, 2] which consists of a generator G and a discriminator D, where the input to G is a random vector z



(a) Pitched Instrument (Piano) (b) Pops (c) Chirps

Figure 1. Examples of (a) a pitch instrument (piano), (b) Noise burst or pops, and (c) Frequency sweeps or chirps, with their respective audio waveform (top row), log magnitude spectrogram (middle row), and instantaneous frequency of unwrapped phase (bottom row) plots. The audio examples presented are 1 second long at 16 kHz sampling rate. Spectrogram computation is with window size 512 and hop size 128 samples.

with 128 components from a spherical Gaussian distribution along with a one-hot conditional vector c_{in}. Separate models were trained for each data set with the only difference being the dimension of the one-hot pitch vector (27, 13, and 9 for NSynth, pops, and chirps, resp.) For each dataset, we train two models as shown in Figure 2. Model A uses a 2-channel audio representation consisting of the log magnitude spectrogram and IF (Figure 2(a)) computed from Short Time Fourier Transform (STFT) with Hanning window, and Model B uses a single-channel log magnitude of Gabor transform (i.e. STFT with Gaussian window) audio representation (Figure 2(b)). During generation, Model A's estimated IFSpectrogram is inverted to a real time domain signal using Librosa's inverse STFT which uses Griffin-Lim iterative algorithm for synthesis initialized by the estimated phase from IF. For model B, we use phase gradient heap integration (PGHI) [7] [2] for reconstruction of the audio signal from the log magnitude. It reconstructs the phase only for the positive frequency coefficients and enforces conjugate symmetry to the negative frequency coefficients in order to guarantee a real-valued time domain signal.

The generator's architecture consists of a Format block and a stack of Scale blocks. The Format block turns the 1D input vector z + one-hot conditional c_{in}, with 128 + x dimensions (where x could be 27, 13, or 9) into a 4D convolutional input consisting of [batch size, 128, w_0, h_0], where w_0 and h_0 are the sizes of each dimension at the

[2] https://github.com/andimarafioti/tifresi

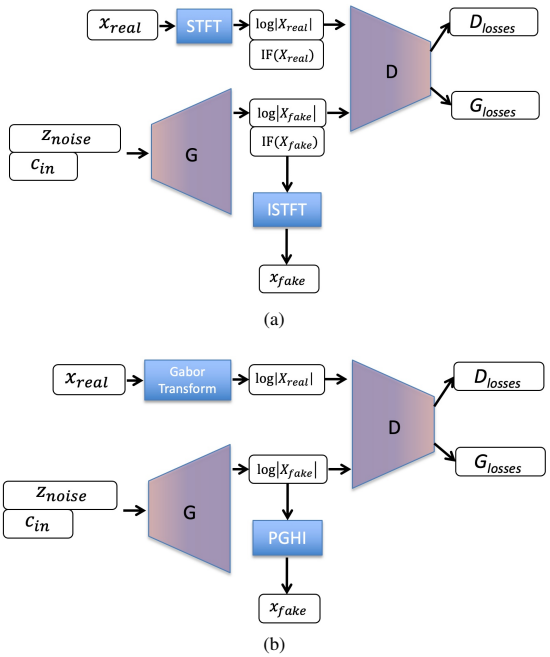


Figure 2. GAN block diagram with (a) IF, and (b) PGHI. z_{noise} is the 128 dimensional latent vector, c_{in} is the conditional parameter one-hot vector. G is the generator, D is the discriminator.

input of the scale block.

The scale blocks are a stack of convolutional and box-upsampling blocks that transform the convolutional input to the generated output signal progressively in 5 stages. The discriminator D is composed of convolutional and down sampling blocks, mirroring the configuration of the generator. D estimates the Wasserstein distance between the real and generated distributions. For more details, please refer to [2] [3] . Our code that implements the GAN architecture with IF as well as PGHI methods (an extended version of Nistal et al.'s code) is available here [4] .

3.2.1 Training

Training is divided into 5 stages, wherein each stage a new layer, generating a higher-resolution output, is added to the existing stack, which is the essence of the progressive-GAN [1, 3]. The gradual blending in of the new layers with a blending parameter *alpha* ensures minimum possible perturbation effects as well as stable training. We train all the models for 1.2M iterations on batches of 8 samples: 200k iterations in each of the first three phases and 300k in the last two. Adam optimization method is employed.

We tried multiple FFT sizes 512, 1024, and 2048 to compute the time-frequency representations, that correspond to window sizes of 32 ms, 64 ms, and 128 ms respectively for a signal sampled at 16 kHz. For transient sounds, a window size with higher time resolution is needed, i.e. a shorter window and indeed we empirically found that FFT size of 512 serves well for both the transient pop sounds and the steady pitched sounds. Therefore, in the experiments presented in this paper, we use FFT size of 512. We tested the effect of redundancy between frames in reconstruction, thus we trained two models, with hop sizes 64 and 128, i.e. 87.5% and 75% overlap between consecutive frames. We train two types of models IF and PGHI, for three kinds of audio textures, NSynth, pop, and chirp, for each of the two hop sizes. All of the models took about 2.5 to 3 days to train on an Nvidia Tesla V100-32GB GPU.

3.3 Evaluation Metrics

Evaluation of generative models is challenging, especially when the goal is to generate perceptually realistic audio that may not be exactly same as any real audio in the dataset. Previously, the inception score has been used as the objective measure that evaluates the performance of a model for a classification task such as pitch or instrument inception score [1, 2]. However, in this work, we are comparing signal representations and synthesis techniques, while the GAN architecture remains the same. Since the variety of sounds with respect to classification is not expected to change. Indeed, Nistal et al [2] noted that inception models are not robust to the particular artifacts of the representations they were comparing, and therefore, it is not a very reliable measure of the overall generation quality.

Marafioti et al. [8] developed an interesting *consistency* measure that estimates how close a magnitude spectrogram is to the frequency transform of a real audio signal. However, it is not obvious how it could be used to compare representations that include explicit phase representations. Also, the perceptual quality of the generated audio signal depends on other factors as well. For example, a real-valued time domain signal of poor perceptual quality will have a perfectly consistent magnitude spectrogram.

In this work, we performed listening tests for subjectively evaluating the quality of the generated sounds, as well as computed Fréchet Audio Distance (FAD) [20] as the objective evaluation metric.

3.3.1 Human Evaluation

To construct stimuli for listening experiments, three points in the latent space are randomly chosen to generate three audio signals of 1 second each per pitch class per trained model, which were then stitched together with a 0.5 second silence before each of the 3 segments) resulting in a 4.5 seconds duration audio clips that were presented in the listening test. This provided variability within each clip so that the listeners focus on the sound quality of the clips and not on the instrument type or the rate of pops and chirps. For reference, a similar set of audio clips was prepared from the original or real audio data set as well.

The listening test was conducted by recruiting twenty participants via Amazon's Mechanical Turk (AMT) website. In each assessment task, the participants were asked to listen first to the reference, then to the two synthesized audio clips, randomly ordered, and then to select the one they felt was the closest in sound quality to the reference clip, or if they were similar. Our task instructions were simplified for the participants and included text like "Although the synthetic clips may sound quite different from the original, you will need to select a clip whose sound quality is most similar to the sound quality of the original". The two audio clips belonged to either IF or PGHI reconstruction techniques for a hop size of 64 or 128 for each comparison. Only same type of sounds were compared, i.e. NSynth_IF to NSynth_PGHI, pop_IF to pop_PGHI etc. Moreover, the two clips being compared had the same pitch or center frequency. 20 random pitches from the NSynth dataset, 13 pitches from pops, and 9 pitches from chirps were selected to build a sample size of 84 comparison trials (42 comparisons each for hop 64 and 128 reconstructions respectively) and overall 1,680 ratings were collected. The trials were loaded into AMT in a random sequence and were completed by participants within 2 hours. The participants were compensated at the rate of US$ 0.02 per comparison trial.

3.3.2 Fréchet Audio Distance

The Frechet Audio Distance (FAD) [20] [5] is the distance between the statistics (mean and covariance) of real and fake data computed from an embedding layer of the pre-trained VGGish model. The embedding layer is considered

[3] https://github.com/SonyCSLParis/Comparing-Representations-for-Audio-Synthesis-using-GANs

[4] https://github.com/lonce/sonyGanFork

[5] https://github.com/google-research/google-research/tree/master/frechet_audio_distance

to be a continuous multivariate Gaussian, where the mean and covariance are estimated for real and fake data, and the FAD between these is calculated as:

$$FAD = ||\mu_r - \mu_g||^2 + tr(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}) \quad (2)$$

where μ_r, Σ_r and μ_g, Σ_g are the mean and covariances of real and fake probability distributions, respectively. Lower FAD means smaller distances between synthetic and real data distributions. The VGGish model is trained on 8M Youtube music videos with 3K classes. The FAD metric has been tested successfully specifically for the purpose of reference-free evaluation metric for enhancement algorithms. FAD performs well in terms of robustness against noise, computational efficiency, and consistency with human judgments, and has been used by Nistal et al. [2]. FAD has been found to have a high correlation (0.52) with human perceptual judgment compared to other measures such as signal-to-distortion ratio, cosine distance or magnitude-L2 distance [20].

4. RESULTS AND DISCUSSION

Qualitatively it is observed that with the IF method, the sharp transients of the pop sounds get smeared in time, whereas PGHI method produces clear and sharp transients. This temporal smearing effect is also observed in the short duration chirps generated from the IF method. This smearing effect arises from the inability of IF to provide robust information about phase when the signal contains closely spaced wideband frequency components. For NSynth data, however, the two methods sounded approximately equal in quality. Examples of the synthesised audio presented for listening tests are here [6], and visual analysis of the generated spectrograms are provided here [7].

Figure 3 (a) and (b) show results from the listening test for reconstructions using hop sizes 64 and 128 respectively. For both hop sizes, participants rated PGHI reconstructions to be significantly better than IF for pop sounds, where they rated in favour of PGHI 80.79% and 73.15% for hop sizes 128 and 64 respectively. This result clearly shows that PGHI with GAN produces perceptually higher quality audio for noisy signals. For chirp sounds, participants rated PGHI somewhat better than IF. But for NSynth pitched instrument sounds, PGHI and IF are similarly rated for both hop lengths. Furthermore, we observe that hop size 64 shows a clearer distinction in preference between IF and PGHI for nsynth and chirp sounds, than hop size 128. This indicates that a higher redundancy in the spectrogram representation may help in better reconstruction with PGHI method than IF method. However, comparison between the two hop sizes for the same method has shown mixed responses for the different datasets, which means that redundancy of more than 4 may not have a significant impact on the reconstructed audio quality of one method.This systematic study suggests that PGHI with GAN produces audio quality perceived as roughly equal to the state-of-the-art IF method for pitched sounds, but significantly higher as the complexity of the signal increases.

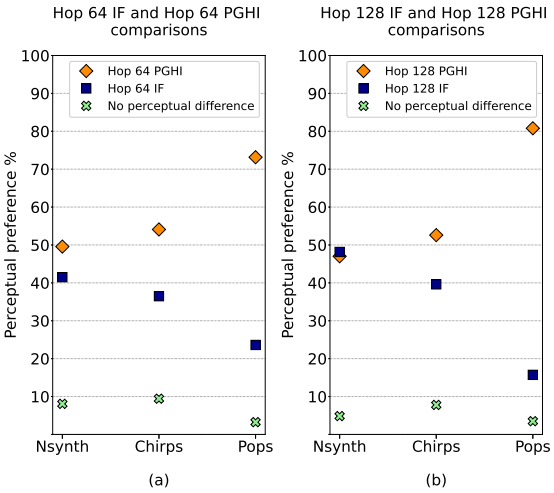


Figure 3. Results from listening tests for comparing IF and PGHI reconstructions from GAN using hop lengths of (a) 64 and (b) 128 respectively. Across both hop lengths, PGHI reconstructions of noise bursts or pops were rated to be significantly better than IF. For chirps, PGHI reconstructions were rated to be slightly better than IF and for pitched instruments PGHI reconstructions were rated almost similar to IF.

To evaluate objectively, we computed the FAD metric, as shown in Table 1. We observe that PGHI method generated audio that consistently shows a smaller distance from reference audio compared to that generated from IF method, although unlike the perceptual ratings, the two representations are closer for chirps than the other two signal types. While this objective measure is broadly in line with the higher ratings for the PGHI method, the systematic disagreement between the user and objective measures across pitched and chirp sounds demonstrate that there is more work to be done to find an objective measure that correlates with human judgements of quality.

The performance of the system, given all other settings are the same (training steps, architecture, etc), is better using the PGHI method than the IFSpectrogram method. Convergence during learning, especially in stage 5 of the progressive GAN differed between the two representations depending on the signal. The IF method representation converged better for NSynth, while PGHI representation converged slightly better for the other signals. However, in all cases, the quality of the synthesised audio was better (Figure 3) using the PGHI method.

5. CONCLUSIONS

We present a general method of audio synthesis using GAN that produces high quality audio output for a wide variety of sounds, pitched instruments as well as non-pitched

[6] https://animatedsound.com/amt/listening_test_samples/#examples
[7] https://animatedsound.com/amt/listening_test_samples/#analysis

Audio Texture	Hop Size	IF	PGHI
Pitched Instruments	128	1.500	1.001
Pitched Instruments	64	1.583	0.924
Pops	128	1.783	0.305
Pops	64	1.866	0.295
Chirps	128	1.395	1.031
Chirps	64	1.269	0.747

Table 1. FAD results of different GAN models with IF and PGHI. A lower FAD means smaller distances between synthetic and real data distributions.

and noisy pop and chirp sounds. We show that IFSpectrogram representation that currently produces the state-of-the-art performance with GAN for pitched instruments is not a robust representation for non-pitched and noisy sounds. Moreover, through subjective and objective measures, we show that integrating the PGHI representation and reconstruction technique in the GAN framework provides a reasonable solution to this problem, as it generates better audio quality for noisy pops and chirps than when using the IFSpectrogram method, and produces similar audio quality for pitched instruments. Audio examples generated from our experiments are available here [8] , and our code implementation is available here [9] .

A potential direction of improvement of the PGHI technique is to use the phase estimates from PGHI as a *warm-start* for other iterative phase reconstruction algorithms such as LeGLA, as shown by Prusa et al. [7]. Another possibility is to include different explicit representations of phase information in training that might outperform magnitude-only reconstruction with PGHI. Marafioti [8] used a representation with frequency derivatives for training which did not perform as well as the magnitude PGHI reconstruction method, but indicates the potential that this direction has to offer.

The method of training a GAN as a data-driven approach to designing parametrically controlled synthesizers holds a lot of promise for creative applications such sound design and music. A signal-independent representation for training the networks is an important step towards the universality and usability of this approach.

Acknowledgments

This research is supported by a Singapore MOE Tier 2 grant MOE2018-T2-2-127, and by an NVIDIA Corporation Academic Programs GPU equipment grant.

6. REFERENCES

[1] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[2] J. Nistal, S. Lattner, and G. Richard, "Comparing representations for audio synthesis using generative adversarial networks," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 161–165.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[4] 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," *arXiv preprint arXiv:1609.03499*, 2016.

[5] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[6] J. Nistal, S. Lattner, and G. Richard, "Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," *arXiv preprint arXiv:2008.12073*, 2020.

[7] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from stft magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[8] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4352–4362.

[9] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[10] Z. Průša, "The phase retrieval toolbox," in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.

[11] Z. Průša and P. Rajmic, "Toward high-quality real-time signal reconstruction from stft magnitude," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 892–896, 2017.

[12] 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 *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[13] N. Saint-Arnaud and K. Popat, "Analysis and synthesis of sound textures," in *in Readings in Computational Auditory Scene Analysis*. Citeseer, 1995.

[14] D. Schwarz, "State of the art in sound texture synthesis," in *Digital audio effects (DAFx)*, 2011, pp. 221–232.

[8] https://animatedsound.com/amt/listening_test_samples/

[9] https://github.com/lonce/sonyGanFork

[15] J. H. McDermott, A. J. Oxenham, and E. P. Simoncelli, "Sound texture synthesis via filter statistics," in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2009, pp. 297–300.

[16] J. M. Antognini, M. Hoffman, and R. J. Weiss, "Audio texture synthesis with random neural networks: Improving diversity and quality," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3587–3591.

[17] L. Wyse and M. Huzaifah, "Deep learning models for generating audio textures," in *Proceedings of the 2020 Joint Conference on Music Creativity*, Stockholm, Sweden, October 19-23 2020.

[18] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *International Conference on Learning Representations (ICLR)*, 2019.

[19] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," *International Society of Music Information Retrieval (ISMIR)*, 2019.

[20] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.