# TOWARDS CONTROLLABLE AUDIO TEXTURE MORPHING

*Chitralekha Gupta*<sup>\*†</sup>, *Purnima Kamath*<sup>\*†</sup>, *Yize Wei*<sup>†</sup>, *Zhuoyao Li*<sup>†</sup>, *Suranga Nanayakkara*<sup>†</sup>, *Lonce Wyse*<sup>\*‡</sup>

<sup>†</sup>National University of Singapore, Singapore <sup>‡</sup>Universitat Pompeu Fabra, Barcelona, Spain

\*equal contribution

#### ABSTRACT

In this paper, we propose a data-driven approach to train a Generative Adversarial Network (GAN) conditioned on "soft-labels" distilled from the penultimate layer of an audio classifier trained on a target set of audio texture classes. We demonstrate that interpolation between such conditions or control vectors provide smooth morphing between the generated audio textures, and show similar or better audio texture morphing capability compared to the state-of-the-art methods. The proposed approach results in a well-organized latent space that generates novel audio outputs while remaining consistent with the semantics of the conditioning parameters. This is a step towards a general data-driven approach to designing generative audio models with customized controls capable of traversing out-ofdistribution regions for novel sound synthesis.

Index Terms- audio texture, morphing, audio classifier, GAN

# 1. INTRODUCTION

Sound morphing encompasses a set of models with the goal of producing gradual transformations between sounds [1]. Sound morphing is useful in applications of sound design including music compositions, video games, and sound synthesizers [2]. Although there is a lack of consensus in the literature about the exact definition of sound morphing [1, 2], there are certain characteristics of sound morphs that are commonly agreed upon. For example, the morphing transformation between two sounds is expected to produce perceptually intermediate results that should fuse into a single perceptual source that resembles both sounds at the same time [1, 3, 2].

We focus on audio textures, a rich class of sounds in which certain parameters remain stationary over time [4] despite statistical variation within the sound. For example, the sound of wind at a certain strength or the sound of tapping at a certain rate. Sounds with specifically varying spectro-temporal envelopes such as a single footstep, speech, or music do not fall under this definition of audio textures. Automatic audio texture synthesis is an active area of research [4, 5, 6] that has applications in sound design and Foley synthesis systems [7].

Many studies have explored morphing between musical instrument timbres [2, 8, 1] or voice timbres [3, 9] using various signal processing techniques, however there have been limited studies on audio texture morphing. Morphing between two pitched musical instruments or two voiced phonemes is typically achieved through signal processing techniques such as interpolation between the coefficients of a source-filter model representation of the two sounds [3], or interpolation between the harmonic components of a sinusoidal model representations of the two sounds [2]. Such methods have an underlying requirement that the two sounds are pitched, such as musical instruments or voiced utterances, therefore applicability of such techniques to non-pitched audio texture sounds is limited. Moreover, linear interpolation between parameters may not result in perceptually linear interpolation between the sounds [1].

The goal of parametric audio texture synthesis is to generate novel sounds with descriptive parameters that match those of a target texture. McDermott et al. [4] developed a set of statistics based on a cochlear model to describe the perceptually relevant aspects of a given audio texture. Recent works [10, 5, 11] have adapted the seminal work on image style transfer [12] for audio texture synthesis, where hand-crafted statistics are replaced with Gram matrix statistics computed as the correlation between feature activations to represent style. Though this method of audio style transfer produces interesting combinations of the sounds, there is no control of semantic style or content features other than through the data examples provided.

Recently, parametrically controllable audio synthesis has been used to help organize the latent space of the GAN and Variational Autoencoder (VAE) independent of the control parameters. Luo et al. [13] learn latent distributions using VAEs to separately control the pitch and timbre of musical instrument sounds. Engel et al. [14] conditioned an autoregressive model to interpolate between musical instruments to generate new sounds. The GANSynth architecture [15, 16] used a ProgressiveGAN for controlled musical note synthesis conditioned on one-hot vector for pitch. However, such architectures are under-explored for audio textures, in part because it is difficult to label audio texture data correctly and robustly with control parameter values. Moreover, one-hot representation of the conditioning vector is nominal and sparse, which may produce unconvincing interpolations in the parameter space during generation. Continuous-valued or floating point conditioning has its own challenges, particularly if the range of parameter values is not densely sampled in the training set [17], but it is more naturally suited to the goal of generation with interpolated values.

In this paper, we propose a data-driven controllable audio texture morphing strategy with the following contributions: (a) a data-driven parameter distillation method for conditioning GAN for controlled audio texture synthesis, (b) a linear interpolation strategy for conditioning parameters that leads to controlled inter- and intra-class morphing of audio textures, (c) a systematic comparison of our method with existing methods through a set of existing and new objective metrics, (d) our code for parameter distillation through an audio classifier and for GAN training.

#### 2. CONDITIONAL GAN

In this work, we identify two types of continuous conditional parameters - *class-identity parameters* C and *intra-class parameters* P. Although they function in the same way during GAN training, class-identity parameters are derived from a classifier trained on the same dataset used to train the GAN. Intra-class parameters are the ones related to the semantics within an audio class. For example, *strength* is an intra-class parameter for the audio texture class wind. Class-identity parameters, by construction, have semantics computed from the dataset, and can be used to navigate between classes while intra-class parameters have externally imposed semantics and may or may not correlate with the class labels. We explore two strategies of multi-dimensional conditioning with intra-class and class-identity parameters: (1) two one-hot conditioning vectors, one for representing intra-class parameter, and the other for representing class-identity, called *One-hot GAN*, and (2) multi-dimensional floating point soft-labels extracted from the penultimate layer of a pre-trained audio classifier representing class-identity conditioning parameters, along with a 1-dimensional floating point intra-class conditional parameter, called *MorphGAN*.

#### 2.1. One-Hot GAN

We adopt Engel et al.'s[15] progressive-GAN with one-hot conditioning (Figure 1(a)). The intra-class parameter P has dimension qequal to the number of unique control parameter values. The classidentity parameter C has dimension r equal to the number of sound classes. To encourage the generator to use the conditional information, an *auxiliary classification* (AC-criterion) loss is added to the discriminator that learns to predict the conditional vector. The ACcriterion calculates the categorical cross entropy loss between the ground-truth conditional vector and the predicted conditional vector through the discriminator.

# 2.2. MorphGAN

Previously, DarkGAN [18] took a knowledge distillation approach and used the probabilities extracted from the output layer of an audio classifier that was trained on an external dataset (AudioSet) as a conditional vector for their GAN. However, labels determined by an external dataset may have little relevance for a specific sound model training set. This can lead to a lack of interpretable control over the generated audio. In MorphGAN, we extract soft-labels from the penultimate layer of an audio classifier that is trained on the sound model training dataset.Since these multi-dimensional soft-labels are learnt by the classifier, they capture multiple class-related aspects of the sound set. These dimensions enable interpretable control over interpolation between points in the latent space of the GAN, generating cyclostationary morphs (a sequence of audio segments each produced with different parameters) across novel in-between sounds.

MorphGAN uses a single dimensional floating point value for intra-class parameter P and x dimensional floating point soft-labels from the output of the penultimate layer of the audio classifier as the class-identity parameter C. Since each data point can have nonexclusive values for each dimension, we use binary cross entropy for auxiliary classifier loss, in which the loss is the sum of the individual binary cross-entropy computation on each dimension,

$$L = -\frac{1}{K} \sum_{k=1}^{K} [y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)]$$
(1)

where  $y_k$  is target conditioning value in the range of [0,1], and  $\hat{y}_k$  is the predicted value from the auxiliary classifier for the kth dimension. Subsequently, a sigmoid activation squashes the values in the range [0,1]. Figure 1(c) shows an overview of MorphGAN.

#### 3. EXPERIMENTAL SETUP

# 3.1. Datasets

We use two types of audio textures in this paper - water, and wind. **Water**: The water sound was recorded by filling a metallic bucket with water at an approximately constant rate over a duration of 30 seconds. We collected 50 audio recordings of different lengths to capture the variation between multiple fillings. The transient sounds at the beginning and end of each sound was trimmed, and then sound was divided into 11 equally spaced time points used as the starting



**Fig. 1.** System overview. GAN input features are a random noise latent vector  $Z_p$  (*p*-dim), along with either (a) One-hot vectors for intra-class parameter  $P_q$  (*q*-dim) and class-identity parameter  $C_r$  (*r*-dim), or (b) Morph-GAN with one dimensional intra-class parameter  $P_1$  but *x* dimensional soft labels for class parameter  $C_x$  from the output of the penultimate layer of a pre-trained *n*-class audio classifier.

point of a 2-second excerpt labeled with one of 11 different "fill levels" normalized to steps of 0.1 in [0,1]. Ten variations were produced from different recordings.

**Wind**: The wind sound is from the Syntex collection of synthetic datasets [19]. This texture is generated with noise passed through filters modulated with simplex noise<sup>1</sup>. A "strength" parameter controls the wind gust fluctuations defined by the bandpass filter center frequencies. The strength parameter ranges in [0,1] across steps of 0.1, and 10 variations are generated from different random seeds, resulting in 110 audio files each of 2 seconds duration.

#### 3.2. Architectures

GAN: We adapt the Nistal et al. [16] progressive-GAN implementation where generator G transforms the 1D input vector (Z+P+C)to the generated output signal over 5 progressively-grown stages and upsampling CNN blocks. The Z vector is 32-dimensional following [18]. The P and C vector dimensions are different across the two GAN variants we employ (Section 3.3). We found that training models for 120K iterations on batch-size 12 with 20k iterations for the first three stages and batches of 8 files with 30k iterations for the last two produces high quality output (Table 1). The audio representation is a magnitude spectrogram computed using the Gabor transform (window size=256, hop size=128). Inversion of the estimated spectrogram is done using phase gradient heap estimation (PGHI) [20]. PGHI is a non-iterative phase reconstruction algorithm 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. Gupta et al. [6] showed through listening tests that training the GANSynth architecture using only log-mag representation and PGHI inversion produces significantly better audio quality for wideband, non-pitched or fast changing signals. Since the audio data we use in this paper consist of such sounds, eg. water-filling, we used PGHI for reconstruction as it gives better audio quality.

Audio Classifier: The DenseNet model [21] pretrained on ImageNet [22] and fine-tuned for a specific audio dataset can achieve stateof-the-art results for audio classification. We adopted this method for audio classification to generate class soft-labels for MorphGAN. We use the pre-trained Dense Convolutional Network (DenseNet201 PyTorch library), that connects each layer to every other layer in a feed-forward fashion. DenseNet expects a 3-channel input, so a three-channel mel-spectrogram of the audio input is computed using different window sizes and hop lengths of [25ms, 10ms], [50ms,

<sup>&</sup>lt;sup>1</sup>Section 1.2 of https://animatedsound.com/ismir2022/ metrics/appendix\_dataset/index.html



**Fig. 2.** Three dimensional soft label values from the penultimate layer of the audio classifier. These are subsequently used for conditioning MorphGAN. The blue markers are water-filling sounds and the red markers are the wind sounds.

25ms], and [100ms, 50ms] on each of the channels respectively. The different window sizes and hop lengths ensure the network has different levels of information from the frequency and time domain on each channel, which was shown to perform well for audio classification [21]. The DenseNet gives a 1,920 dimensional output after which we add two linear layers sequentially of x and n dimensions respectively, where x is a selectable number of soft-labels and n is the number of audio classes. Subsequently, a sigmoid activation function squashes x values between 0 and 1 before using them as class conditional inputs for MorphGAN.

### 3.3. Models

**One-Hot GAN:** P is 11 dimensional to represent the 11 discrete values (10 equally-spaced intervals across the range) of the control parameter for the two textures, and C is 2 dimensional. Note that P is a dual serving intra-class parameter, representing fill level for water and strength for wind.

**MorphGAN**: *P* is a 1D floating point dual serving intra-class parameter for the two textures, discretized to 11 values in [0,1], while *C* is a 3 dimensional soft-label extracted from the penultimate layer of the audio classifier, values between [0,1]. An 80/20% split was used for training and validation (val accuracy=100%). The entire water-wind dataset was then passed through this trained classifier to extract the soft-labels from the penultimate layer. Figure 2 shows the 3 dimensional soft labels that were learnt by the audio classifier, color-coded with the audio texture class. It is evident that this 3D vector has a wide range of values while also being able to represent the two classes, thus we hypothesize that this 3D vector, when used for class conditioning MorphGAN, will offer more flexibility and control for inter-class morphing than using 1D class vectors.

**Baseline:** As a baseline for comparison, we use Zynaptic's MORPH2.0<sup>2</sup>, a commercial real-time plug-in for structural audio morphing. We chose their "classic" (vocoder-like) interpolation algorithm that uses signal processing to model the timbral shape for every time frame of the two audio inputs, and then interpolates between these models, transforming one sound into the other. Other open-source toolkits, such as sound morphing toolbox [8] fail for non-pitched audio textures such as water-filling as they employ matching of harmonics in the sound, thus restricting their use in our experiments.

We explore two kinds of morphs, inter-class morphs, that interpolate between two points in class-identity conditioning dimensions, and intra-class morphs, that interpolate between two points in intra-class conditioning dimensions. In the Morph2.0 baseline, we compute interpolations between two sounds from the original data, where for inter-class interpolations, the two sounds belong to water and wind classes, and for intra-class interpolations, the two

<sup>2</sup>https://www.zynaptiq.com/morph/morph-overview/

sounds belong to the same class but two extreme intra-class parameter (fill-level/strength) values. The interpolation is along the morph axis (cross-fade axis=0) of the interface.

#### 4. RESULTS

Audio examples and the evaluation code to generate the metrics are available on our webpage.  $^{3}$ .

# 4.1. Audio Quality

We use the Fréchet Audio Distance (FAD) [23] metric (distance between the distributions of the embeddings of real and synthesized audio data extracted from a pre-trained VGGish model) to evaluate audio synthesis quality as it has been shown to be consistent with human judgements [23, 16, 6]. We compute the FAD for the wind and water sounds generated by the two GAN models, as well as the original one-hot model with latent vector size and training iterations same as in [16, 6], as shown in Table 1. We use the training data as the reference distribution and generate 10 variations per condition from each GAN as the test distributions. Our one-hot GAN has reduced dimensions for Z and fewer training iterations but shows similar performance as the original one-hot model. This lightweight architecture has the advantage of reduced training time, and is more suitable for the limited but targeted range of sounds in our dataset. Overall, MorphGAN performs better than the One-Hot GANs.

**Table 1**. FAD between generated distribution and real distribution.  $\downarrow$  indicates smaller is better.

| Architecture     | Details                     | FAD-<br>water(↓) | FAD-<br>wind(↓) |
|------------------|-----------------------------|------------------|-----------------|
| Original one-hot | 128-D Z, one-hot $P,C$ ,    | 5.83             | 1.25            |
| [16, 6]          | 1.2M training iterations    |                  |                 |
| One-Hot GAN      | 32-D Z, one-hot $P,C$ ,     | 5.04             | 1.21            |
| (Reduced Z dims) | 120K training iterations    |                  |                 |
| MorphGAN         | 32-D Z, 1-D FP P, 3-D FP    | 3.13             | 0.87            |
|                  | C, 120K training iterations |                  |                 |

### 4.2. Intra-class Morphing

We quantify interpolation smoothness of the intra-class morphed sounds by adopting the parameter sensitivity metric from [24]. This sensitivity metric evaluates the linearity of change in the perceptual distance of an interpolated sound as the intra-class control parameter P is varied from its lowest to its highest value linearly. This linearity of change is quantified using the Pearson's correlation coefficient. Amongst the perceptual distance measures discussed in [24], we use Gram Matrix (GM) loss and FAD because these measures showed high correlation with human perception for the audio textures in this study. Table 2 shows that MorphGAN is able to produce more perceptually linear intra-class morphs than One-Hot GAN and Morph2.

**Table 2**. Intra-class morphing. ↑ indicates larger values are better.

|              | 1 0 1 | 2                                |        |  |
|--------------|-------|----------------------------------|--------|--|
| Architecture |       | Control Parameter Sensitivity(↑) |        |  |
|              |       | w/ GM Loss                       | w/ FAD |  |
| MorphGAN     | Wind  | 0.97                             | 0.98   |  |
|              | Water | 0.99                             | 0.95   |  |
| One-Hot GAN  | Wind  | 0.80                             | 0.75   |  |
|              | Water | 0.47                             | 0.74   |  |
| Morph2       | Wind  | 0.77                             | 0.19   |  |
|              | Water | 0.90                             | 0.64   |  |

#### 4.3. Inter-class Morphing

To evaluate inter-class morphing, we analyse the effectiveness of the algorithms to (1) linearly/smoothly morph between classes, and (2) their ability to generalize to out-of-distribution (OoD) points in the class parameter space C.

For morph smoothness, we adapt the parameter sensitivity metric outlined in the previous section by measuring the GM Loss and FAD between class interpolated samples. Table 3 shows that MorphGAN is able to produce smoother linear interpolations between classes than the other methods.

<sup>&</sup>lt;sup>3</sup>https://animatedsound.com/research/morphgan\_icassp2023/

To the best of our knowledge, there is no standard evaluation technique to test OoD generalizability during morphing. We thus develop two additional metrics: Distribution Closeness and Distribution Centeredness of the samples generated using OoD class parameter values in comparison with the training data. Specifically, we choose the out-of-distribution value k = 0.5 for the three class dimensions of MorphGAN (center of the cube in Figure 2), for the two class dimensions of One-Hot GAN, and for the morph axis (center) for Morph2.0. For each algorithm we measure the FAD between the distribution of samples generated from the center point k and the distribution of samples from each of the two classes. We term the mean of the two distances as Distribution Closeness to indicate the algorithms' ability to produce sounds related to the training data at this OoD point. Further, we refer to the difference between the two distances as Distribution Centeredness to indicate the skew of the center point towards any one class. Table 3 shows that the OoD center point of the class parameter space C of MorphGAN is perceptually closer and centered between both the classes and thus can generate more perceptually meaningful and novel morphs in the neighborhood of that location compared to One-Hot or Morph2.

Table 3. Inter-class morphing. ↑ indicates larger values are better.

| Architecture | Class Parameter Sensitivity(↑) |        | Distribution           | Distribution                    |
|--------------|--------------------------------|--------|------------------------|---------------------------------|
| Arcinecture  | w/ GM Loss                     | w/ FAD | Closeness( $\psi$ ) ee | Centercuness( <sub>\phi</sub> ) |
| MorphGAN     | 0.96                           | 0.90   | 8.03                   | 6.00                            |
| One-Hot      | 0.96                           | 0.76   | 14.05                  | 13.70                           |
| GAN          |                                |        |                        |                                 |
| Morph2       | 0.84                           | 0.25   | 16.15                  | 8.70                            |

Qualitatively, the One-Hot morph samples exhibit a stickiness towards one class, and towards the center of the interpolation, there is a sudden transition to the second class, resulting in an abrupt interpolation. In MorphGAN interpolations, the frequency components of the wind class gradually modify and merge with the frequency components of the water class which corresponds to the perception of a smooth morph (Figure 3 and webpage).

To examine this objectively, we sample a path in the class identity parameter C between a wind and a water sound at 11 points, and for each, we generate 20 audio files for random values of Z. The generated audio files are passed back through the classifier and we plot box plots of the output class node0 (water class) values for One-Hot and MorphGAN (Figure 4). Both One-Hot and MorphGAN show consistent outputs towards the class end-points (small std dev in the boxes). However, for class values in between the end points for which neither the classifier nor the GAN were trained, One-Hot shows smaller spread than MorphGAN. This indicates that MorphGAN produces novel morphing sounds with characteristics distinct from the classifier training data whereas the One-Hot tends to stick to one or the other of the two classes. This reinforces our qualitative observation about the same and limits the exploration of sounds in between classes using the One-Hot representation.

### 4.4. Semantic exploration of inter-class morphing

To analyse the semantic control of the three class parameter dimensions C of MorphGAN, we varied each dimension from 0 to 1 in steps of 0.1, while keeping all other dimensions constant at 0.5, and fixing a random Z vector (Figure 5 (top)). Qualitatively, we can describe variation in the first C dimension 0 as taking the texture from



Fig. 3. Concatenated 2s audio outputs from (a) One-Hot GAN, and (b) MorphGAN as the class parameter C interpolates between values for wind to water in 11 steps while keeping P fixed.



**Fig. 4.** Output activation values for node0 (water) of classifier for audio generated from (a) One-Hot GAN, and (b) MorphGAN. The Y-axis is the node0 (water) output from the classifier, and the X-axis is the class parameter interpolated from water to wind.

a gurgly-wind sound to a wind-like sound. Lower parameter values also contain higher frequency components from water sounds. Dimension 1 variation takes the texture from a windy whooshing sound to a more watery swish-like sound, where the higher values of this dimension introduce the higher frequency components but at lower amplitudes. Dimension 2 variation moves the texture between water-like and wind-like sounds.

To gain objective insight, we passed these generated audio files back through the audio classifier and plotted the value of the two output node (water and wind) values in Figure 5 (bottom). The classification node output values show a trend reinforcing what we hear. For example, dimension 0 variation initially shows a somewhat ambiguous class pattern which changes to predominantly windy. Dimension 1 is smoothly varying, but changes quality while remaining water-like. Dimension 2 (given the values of the other dimension) takes the sound from a fairly clear water to a clear wind sound. These controls afford a perceptual variety of paths between any two endpoints with access to novel sounds that can be explored creatively for example, to create a wind amplitude pattern modulating a water sound, or to add a bubbly quality to a a wind sound.



**Fig. 5.** Spectrogram (top) of concatenated audio outputs of 2s and corresponding audio classifier node activations (bottom) as class parameter (a) dimension 0, (b) dimension 1, and (c) dimension 2 are varied from 0 to 1 at steps of 0.1. Other dimensions are fixed.

### 5. CONCLUSION

In this work, we show that class parameters derived from an audio classifier trained on target data is effective for producing convincing morphs between different audio textures. We demonstrate that these class conditional parameters also provide multiple interpretable control dimensions for morphing between two sounds along different paths. We also show that the class parameters consistently produce the intended class across the latent Z space. In the future, improving the consistency of arbitrary control parameters along with wider range of audio textures need to be explored. Future work will also include perceptual listening tests using both audio experts as well a novice listeners and include think-aloud studies to comprehensively analyze the effect of deep learning algorithms over existing baselines. This work is a step towards building data-driven controllable audio texture morphing frameworks.

#### 6. REFERENCES

- [1] Marcelo Caetano, *Morphing isolated quasi-harmonic acoustic musical instrument sounds guided by perceptually motivated features*, Ph.D. thesis, Paris 6, 2011.
- [2] Savvas Kazazis, Philippe Depalle, and Stephen McAdams, "Sound morphing by audio descriptors and parameter interpolation," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16). Brno, Czech Republic*, 2016.
- [3] Malcolm Slaney, Michele Covell, and Bud Lassiter, "Automatic audio morphing," in 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. IEEE, 1996, vol. 2, pp. 1001–1004.
- [4] Josh H McDermott and Eero P Simoncelli, "Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis," *Neuron*, vol. 71, no. 5, pp. 926–940, 2011.
- [5] Joseph M Antognini, Matt Hoffman, and Ron J Weiss, "Audio texture synthesis with random neural networks: Improving diversity and quality," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3587–3591.
- [6] Chitralekha Gupta, Purnima Kamath, and Lonce Wyse, "Signal representations for synthesizing audio textures with generative adversarial networks," in *Sound and Music Computing* (*SMC*), 2021.
- [7] Keunwoo Choi, Sangshin Oh, Minsung Kang, and Brian McFee, "A proposal for Foley sound synthesis challenge," *arXiv preprint arXiv:2207.10760*, 2022.
- [8] Marcelo Caetano, "Morphing musical instrument sounds with the sinusoidal model in the sound morphing toolbox," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2019, pp. 481–503.
- [9] Tony Ezzat, Ethan Meyers, James Glass, and Tomaso Poggio, "Morphing spectral envelopes using audio flow," in *Ninth European Conference on Speech Communication and Technology*, 2005.
- [10] Dmitry Ulyanov and Vadim Lebedev, "Audio texture synthesis and style transfer," [Blog post]. Available from: https://dmitryulyanov. github. io/audio-texture-synthesis-andstyle-transfer/[accessed 16 Jan 2022], 2016.
- [11] Hugo Caracalla and Axel Roebel, "Sound texture synthesis using RI spectrograms," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*). IEEE, 2020, pp. 416–420.
- [12] Leon Gatys, Alexander S Ecker, and Matthias Bethge, "Texture synthesis using Convolutional Neural Networks," Advances in neural information processing systems, vol. 28, pp. 262–270, 2015.

- [13] Yin-Jyun Luo, Kat Agres, and Dorien 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.
- [14] 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 *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.
- [15] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.
- [16] Javier Nistal, Stefan Lattner, and Gael Richard, "Comparing representations for audio synthesis using generative adversarial networks," in 2020 28th European Signal Processing Conference (EUSIPCO). IEEE, 2021, pp. 161–165.
- [17] Xin Ding, Yongwei Wang, Zuheng Xu, William J Welch, and Z Jane Wang, "Ccgan: Continuous conditional generative adversarial networks for image generation," in *International Conference on Learning Representations*, 2020.
- [18] Javier Nistal, Stefan Lattner, and Gaël Richard, "Darkgan: Exploiting knowledge distillation for comprehensible audio synthesis with GANs," *arXiv preprint arXiv:2108.01216*, 2021.
- [19] Lonce Wyse and Prashanth Thattai Ravikumar, "Syntex: parametric audio texture datasets for conditional training of instrumental interfaces.," *International Conference on New Interfaces for Musical Expression*, 4 2022, https://nime.pubpub.org/pub/0n157935.
- [20] Zdeněk Průša, Peter Balazs, and Peter Lempel 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.
- [21] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao, "Rethinking CNN models for audio classification," arXiv preprint arXiv:2007.11154, 2020.
- [22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
- [23] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, "Fréchet Audio Distance: A reference-free metric for evaluating music enhancement algorithms.," in *IN-TERSPEECH*, 2019, pp. 2350–2354.
- [24] Chitralekha Gupta, Yize Wei, Zequn Gong, Purnima Kamath, Zhuoyao Li, and Lonce Wyse, "Parameter sensitivity of deepfeature based evaluation metrics for audio textures," *arXiv* preprint arXiv:2208.10743, 2022.