2021-11-30

# Audio representations for deep learning in sound synthesis: A review

Anastasia Natsiou
*Technological University Dublin*, anastasia.natsiou@tudublin.ie

Sean O'Leary
*Technological University Dublin*, sean.oleary@tudublin.ie

# Audio representations for deep learning in sound synthesis: A review

Anastasia Natsiou
*Technological University of Dublin*
Dublin, Ireland
anastasia.natsiou@tudublin.ie

Seán O'Leary
*Technological University of Dublin*
Dublin, Ireland
sean.oleary@tudublin.ie

***Abstract*—The rise of deep learning algorithms has led many researchers to withdraw from using classic signal processing methods for sound generation. Deep learning models have achieved expressive voice synthesis, realistic sound textures, and musical notes from virtual instruments. However, the most suitable deep learning architecture is still under investigation. The choice of architecture is tightly coupled to the audio representations. A sound's original waveform can be too dense and rich for deep learning models to deal with efficiently - and complexity increases training time and computational cost. Also, it does not represent sound in the manner in which it is perceived. Therefore, in many cases, the raw audio has been transformed into a compressed and more meaningful form using upsampling, feature-extraction, or even by adopting a higher level illustration of the waveform. Furthermore, conditional on the form chosen, additional conditioning representations, different model architectures, and numerous metrics for evaluating the reconstructed sound have been investigated. This paper provides an overview of audio representations applied to sound synthesis using deep learning. Additionally, it presents the most significant methods for developing and evaluating a sound synthesis architecture using deep learning models, always depending on the audio representation.**

***Index Terms*—Sound representations, Deep learning, Generative models, Sound synthesis.**

## I. Introduction

Sound generation algorithms synthesize a time domain waveform. This waveform should be coherent and appropriate for its intended use. These waveforms can convey complex and varied information. Deep generative networks [1] have demonstrated great potential for such tasks having been used for the synthesis of a range of sounds, from pleasant pieces of music to natural speech [2]. These models discover latent representations based on the distribution of the initial data and then sample from this distribution to generate new acoustic signals with the same properties as the original ones. In many cases, the deep learning models can operate along with signal processing algorithms and enhance their expression capabilities [3] [4].

The representation of the sound embraced by the deep neural network plays a major role on the development of the algorithm. Raw time domain audio is a rich representation which leads to massive information making the network computationally expensive and therefore slow. Compressed Time-frequency representations based on spectrograms can decrease the computer power needed but the parameter detection and synthesis of the sound is usually a challenging task and the loss of information can cause significant reconstruction error [5]. Parameters extracted from state-of-the-art vocoders have also been proposed for deep neural network applications [6]. These parameters demonstrate a potential in marrying the deep generative models with statistical parametric synthesizers. Finally, contemporary investigations allow the network to determine the feature needed for the task [7]. Linguistic and acoustic features can be encoded into latent representations such as embeddings.

Apart from an overview of the audio representations existing in sound synthesis implementations, this paper additionally quotes popular schemes for conditioning a deep generative network with auxiliary data. Conditioning in generative models can control the aspects of the synthesis and lead to new samples with specific characteristics [8]. Furthermore, the paper highlights examples of deep generative models for audio generation applications. Deep neural networks have demonstrated remarkable progress in the field demonstrating impressive results. A final section discusses evaluation processes for synthesised sound. Subjective evaluation via listening tests are generally considered the most reliable measure of quality. However, multiple other metrics for assessing a generative model have been proposed converting both acoustic signals to intermediate representations of them to be examined. Consequently, audio representations assume an essential role not only as input data but also influence the network architecture, the conditioning technique as well as the evaluation process.

## II. Input Representations

In the literature, numerous audio representations have been proved beneficial for audio synthesis applications. Many times, comparisons have been conducted between different forms of the sound to reveal the most appropriate representation for a specific deep learning architecture. Raw audio and time-frequency representations usually present the first attempts in such experiments. However, recent studies also look to higher level forms that offer more meaningful description such as embeddings, or multiple sound features like the fundamental frequency, loudness and features extracted by state-of-the-art vocoders such as WORLD [9]. The table I summarizes the advantages and disadvantages of each sound representation.

## A. Waveform - Raw Audio

The term raw audio is often used to refer to a waveform encoded using pulse code modulation (PCM). This is the sampling of a continuous waveform in time and amplitude. It represents the waveform as a sequence of numbers, each number representing an amplitude sample at a chosen sampling frequency. In order for this discrete sequence of samples to capture all the necessary information, the highest frequency in the signal should adhere to the Nyquist-Shannon theorem [10]. According to this theorem, only frequencies of less than half the sampling frequency can be reproduced from the sampled signal. A typical sampling frequency for audio applications is 44.1kHz. Each real number is assigned to the approximate fixed value in a finite set of discrete numbers. The most common levels for quantization are stored in 8 bits (256 levels), 16 bits (65536 levels) and 24 bits (16.8 million levels). Therefore, a sound with a duration of one second sampled at 44.1kHz generates 44100 samples. This representations is considered extremely informative even for deep learning networks.

In order for the outcome of the deep learning model to be more effective, a pre-processing step can be used to reduce the quantization range of the raw audio. Many research approaches [11] [12] [13] apply $\mu$-law to decrease the possible values of each prediction. $\mu$-law is presented in Eq. 1 where $-1 < x < 1$ and $\mu$ equals the number of levels created after the transformation.

$$f(x) = sgn(x)\frac{ln(1+\mu|x|)}{ln(1+\mu)} \tag{1}$$

Although non-linear quantization processes such as $\mu$-law received much attention the last years, the majority of the existing papers use a normalized high resolution signal as input [14]. Finally, other applications include linear quantization of the input waveform [15] [16] and different designs for most and less significant bits [17].

## B. Spectrograms

A spectrogram is a time/frequency visual representation of sound. A spectrogram can be obtained via the Short Time Fourier Transform (STFT), where the Fourier Transform is applied to overlapping segments of the waveform. The Discrete Fourier Transform (DFT) is presented by the equation Eq. 2 for $k = 0, 1, .., N-1$ where N is the number of samples and k is the number of segments. The spectrogram uses just the absolute values of the STFT, discarding the phase. This type of spectrograms has been used in many by a variety of papers [18] [19] [20].

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\omega_k n} \tag{2}$$

Apart from the original spectrogram, deep learning architectures have also experimented with non linear spectrograms such as mel-spectrograms [21] [22] [23] [24] [25] [26] [27] or Constant-Q Transformations (CQT) [28]. The mel-spectrogram is generated by the application of perceptual filters on the DFT called mel-filter bands. The most common formula for encoding mel-filter bands is presented by Eq. 3 where $f$ is the frequency in Hertz. However, other models have captured the perceptual transformation applying a linear transformation until 1kHz and a logarithmic above this threshold.

$$mel = 2595log_{10}(1 + \frac{f}{700}) \tag{3}$$

CQT is another time-frequency representation where the frequencies are geometrically spaced. The centre frequencies of the filters are calculated from the result of the formula $\omega_k = 2^{\frac{k}{b}}\omega_0$ where $k = 1, 2, ..k_{max}$ and b is a constant number. The bandwidth of each frequency then comes as $\delta_k = \omega_{k+1} - \omega_k = \omega_k(2\frac{1}{b} - 1)$ and therefore the frequency resolution is determined by the Eq. 4 where Q is the quality factor.

$$Q = \frac{\omega_k}{\delta_k} = (2^{\frac{1}{b}} - 1)^{-1} \tag{4}$$

CQT is a representation with different frequency resolution in low and high frequencies. However, the phase part is discarded and the representation is in most of the cases irreversible. Following this argument, Velasco et al [29] proposed an invertible CQT based on nonstationary Gabor frames. Another variation of CQT is rainbowgrams. Rainbowgrams proposed by Engel et al [7] using colors to encode time derivatives of the phase.

In addition, more complicated spectrogram-based representations have also been investigated. GANSynth [30] conducted experiments with numerous spectrograms including scaled logarithmic amplitude and phase of the STFT, increased resolution of the original spectrogram or applied mel-filters. Also, they examined an Instantaneous Frequency based spectrogram where the phase of the STFT is scaled and unwraped (add $-\pi$) and then the finite difference between the frames is computed. Other applications [31] [32], also, made comparisons between raw audio and spectrogram to uncover the most functional representation for their deep learning model.

## C. Acoustic Features

Overcoming the wealth of acoustic information presenting in a sound waveform, various studies extract perceptual features from the original signal. These acoustic features can be represented by phoneme inputs [33], fundamental frequency and spectral features [34] or multiple information such as the velocity, instrument, pitch and time [35]. Other implementations have included cepstral coefficients [4] [36] or a variety of linguistic and acoustic features [37] [13]. Finally, widely recommended parameters have also been extracted by the WORLD vocoder [6] [38] [39].

## D. Embeddings

Embeddings initially introduced by Natural Language Processing (NLP) in order to convert a word or sentence into a real-valued vector. This approach assisted the process of text in deep learning applications by inserting the property of closer embeddings in vector space to encode words with similar meaning. The same approach has been adopted by sound processing to reduce the dimensionality of the signal [40] [22], enhance the timbre synthesis [3] or even generate a more interpretable representation [41] [42] to effectively extract parameters for a synthesizer. In [7] an autoencoder generates a latent representation to condition a WaveNet model while Dhariwal et al [43] implemented three separate encoders to generate vectors with different temporal resolutions.

### *E. Symbolic*

In music processing, the term symbolic refers to the use of representations such as Musical Instrument Digital Interface (MIDI) or piano rolls. MIDI is a technical standard that describes a protocol, a digital interface or the link for the simultaneous operation between multiple electronic musical instruments. A MIDI file demonstrates the notes being played in every time step. Usually this file consists of information of the instrument being played, the pitch and its velocity. MidiNet [44] is one of the most popular implementations using MIDI to generate music pieces.

Piano roll constitutes a more dense representation of MIDI. A piece of music can be represented by a binary $N \times T$ matrix where N is the number of playable notes and T is the number of timesteps. In MuseGAN [45], Generative Adversarial Networks (GANs) have been applied for music generation using multiple-track piano-roll representation. Also, in DeepJ [46], they scaled the representation matrix between 0 and 1 to capture the note's dynamics. The most notable disadvantage of symbolic representations is that holding a note and replaying a note are demonstrated by the same representation. To differentiate these two stages, DeepJ included a second matrix called *replay* along with the original matrix *play*.

## III. Conditioning Representations

Neural networks are able to generate sound based on the statistical distribution of the training data. The more uniform the input data to the network is, the more natural outcome can be achieved. However, in cases where the amount of training data is not sufficient, additional data with similar properties can be included by applying conditioning methods. Following these techniques, the generated sound can be conditioned on specific traits such as speaker's voice [47] [27], independent pitch [3] [48] [36], linguistic features [49] [17] or latent representations [4] [45]. Instead of one-hot-embeddings, some implementations have also used a confusion matrix to capture a variation of emotions [39], while others provided supplementary positional information of each segment conditioning music to the artist or genre [43]. After training, the user is able to decide between the conditioning properties of the synthesised sound.

### *A. Additional Input*

The simplest strategy for applying conditioning to deep learning architectures is by including auxiliary input data while training. Two types of conditioning have been proposed, global and local [11] [50]. In global conditioning, additional latent representations can be appended across all training data. Global conditioning can encode speaker's voice or linguistics features. Local conditioning usually refers to supplementary timeseries with lower sampling rate than the original waveform or even mel-spectrograms, logarithmic fundamental frequency or auxiliary pitch information.

WaveNet has achieved one of the most effective strategies for conditioning deep neural networks [51]. Therefore, later sound generation schemes adopted a WaveNet network for conditioning. The majority of these works conditioned their model to spectrograms [31] [52] [49] [53] [33] [54] [28] [55] while others included linguistic features and pitch information [12] [56], phoneme encodings [6], features extracted from the STRAIGHT vocoder [57] or even MIDI representations [58].

Although it has been proven that convolutional networks are capable of effective conditioning, other architectures can use auxiliary input data as well. Recurrent neural networks such as LSTMs have been adopted conditioning as frame-level auxiliary feature vectors [59] or as one-hot representation encoding music style [46]. Autoencoders can be conditioned including additional input to the encoder [23] [36] but also as input only to the decoder [40].

### *B. Input to the Generator*

Generative Adversarial Networks (GANs) consist of two separate networks, the Generator and the Discriminator. Following the fundamental properties of GANs, the Generator converts random noise to structured data while the Discriminator endeavors to classify a signal as original or generated. For applying conditioning in GANs, the most common technique constitutes a biased input to the Generator. In sound synthesis, a well established conditioning method includes the mel-spectrogram as input to the Generator [60] [16] [19]. This way, the synthesised sound is not just a product of a specific distribution but it also obtains desirable properties. For example, it can be enforced to conditioning on predetermined instrument or voice. Furthermore, a Generator conditioned on spectrograms can also be used as a vocoder [61]. In addition to the mel-spectrogram, other implementations have been conditioned on raw audio [62], one-hot vectors to encode musical pitch [30], linguistic features [13], or latent representations to identify speaker [63].

### *C. Other*

At last, other variations of conditioning have been introduced as well. Kim et al [14] adjusted conditioning through the loss function. They estimated an auxiliary probability density using mel-spectrograms for local conditioning. Pink et al [64] applied bias terms in every layer of the convolutional network using also mel-spectrograms. Extra bias to the network has been also proposed by [65] to encode linguistic features while in [7] every layer was biased with a different linear projection of the temporal embeddings. In [38] linguistic features were added to the output of each hidden layer in

TABLE I
OVERVIEW OF SOUND REPRESENTATIONS

| Representation | Papers | Advantages | Disadvantages | Comments |
|---|---|---|---|---|
| Waveform | [11] [12] [13] [14] [15] [16] [17] | -Completely describes the waveform.<br>-Directly generates the output waveform. | -Computationally expensive.<br>-Unstructured representation that does not reflect sound perception. | Used as input or conditional representation |
| Spectrograms | [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] | -Interpretable representations that are related to sound perception.<br>-Easy to illustrate/plot. | -Typically phase is discarded meaning it is not directly invertible. | Used as input or conditional representation |
| Acoustic features | [33] [34] [35] [4] [36] [37] [13] [6] [38] [39] | -Compressed, descriptive representation of aspects of sound. | -Difficult to synthesize waveforms with long term coherence.<br>-Typically don't fully specify sounds. | Mostly used for conditioning |
| Latent representations | [3] [22] [40] [41] [42] [7] [43] | -Similar sounds lead to smaller distance in multi-dimensional space.<br>-Compressed representation. | -Losses in decoding.<br>-Can be difficult to interpret. | Mostly VAEs |
| Symbolic | [44] [45] [46] | -Meaningful description of musical content. | -Very high level description of audio. | |

the Generator while in [44] a new network was introduced by the name conditionerCNN to work along with the Generator encoding chords for melody generation. Finally, Juvela et al [37] conducted a comparative study of conditioning methods.

## IV. METHODS

During recent years, deep learning models have significantly contributed to research on sound generation. Using a variety of deep learning algorithms, multiple representations have been applied. The most common architectures include autoregressive methods, variational autoencoders (VAE), adversarial networks and normalising flows. However, many approaches can fall in more than one category.

### A. Autoregressive

Autoregressive models define a category of generative models where every new sample in a sequence of data depends on previous samples. Autoregressive deep neural networks can be represented by architectures that demonstrate this continuation implicitely or explicitely. Conventional methods that implicitely indicate a time-related manner are the recurrent neural networks. These models are able to recall previous data dynamically using complex hidden state. SampleRNN [15] is one well established research work that applies hierarchical recurrent neural networks such as GRU and LSTM on different temporal resolutions for sound synthesis. In order to illustrate the temporal behaviour of the network, Mehri et al conducted experiments to test the model's memory by injecting one second of silence between two random sequential samples. Other significant papers on autoregressive models using recurrent neural networks are WaveRNN [17], MelNet [27] or LPCNet [4]. In WaveRNN, they introduced a method for reducing the sampling time by using a batch of short sequences instead of a unique long sequence while maintaining high quality sound.

Generative models where the synthesis of the sequential samples follows a conditional probability distribution like the one in Eq.5 are able to explicitly demonstrate temporal dependencies.

$$p(X) = \prod_{t-1}^{T} p(x_t|x_1, ..., x_{t-1}) \tag{5}$$

WaveNet [11] presents the most influential architecture of explicit autoregressive models. The probability distribution can be imitated by a stack of convolutional layers. However, to improve efficiency, the sequential data passes through a stack of dilated causal convolutional layers where the input data are masked to skip some dependencies. Following a similar scheme, FFTNet [48] takes advantage of convolutional networks mimicking the FFT algorithm while upsampling the input data. However, to clear up the confusion, convolutional networks do not always lead to autoregressive models [20].

Autoregressive models have been initially proposed for sequential data. Therefore, in sound synthesis, raw audio is ordinarily used as the input representation. However, many auxiliary representations have been applied conditioning the audio generation on a variety of properties. More details about conditioning techniques for autoregressive models have been already presented in Section III.

Since autoregressive models can be applied on sequential data, they are well established in sound generation related topics. Autoregressive models are easy to train and they can manipulate data in real time. Furthermore, convolutional-based models can be trained in parallel. Nevertheless, although these models can be paralleled during training, the generation is sequential and therefore slow. Synthesised data are affected only by previous samples, providing half way dependencies. Finally, the generation can be consistent to specific properties for a definite number of samples and the outcome often lacks global structure.

## *B. Normalizing Flow*

Normalizing flows constitute a family of generative models consisting of multiple simple distributions for transforming input data to latent representations. A sequence of simple, invertible and computationally inexpensive mappings $z \backsim p(z)$ can model a reversible complex one. This complex transformation is presented in Eq. 6 and the inverse can be achieved by repeatedly changing the variables as shown in Eq. 7. The mapping functions, then, can be parametrised by a deep neural network.

$$x = f_0 \circ f_1 \circ ... \circ f_k(z) \tag{6}$$

$$z = f_k^{-1} \circ f_{k-1}^{-1} \circ ... \circ f_0^{-1}(x) \tag{7}$$

WaveGlow [21], a flow-based generative network, can synthesise sound from its mel-spectrogram. By applying an Affine Coupling Layer and a 1x1 Invertible Convolution, the model aims to maximise the likelihood of the training data. The implementation has been proposed by NVIDIA and it is able to generate sound in real time. Insightful alternatives have also been proposed on normalising flows by using only a single loss function, without any auxiliary loss terms [14] or by applying dilated 2-D convolutional layers [64].

Finally, in order to reduce the number of repeated iterations needed by normalising flows, they have been merged with autoregressive methods. This architecture manages to increase the performance of autoregressive models since the sampling can be processed in parallel. Using Inverse Autoregressive Flows (IAF), Oord et al increased the efficiency of WaveNet [12]. Their implementation follows a "probability density distillation" where a pre-trained WaveNet model is used as a teacher and scores the samples a WaveNet student outputs. This way, the student can be trained in accordance with the distribution of the teacher. A similar approach has been adopted by ClariNet [31], where a Gaussian inverse autoregressive flow is applied on WaveNet to train a text-to-wave neural architecture.

## *C. Adversarial Learning*

Unlike the Inverse Autoregressive Flows where a pre-trained teacher network assist a student model, in adversarial learning, two neural networks match against each other in a two-player minimax game. The fundamental architecture of Generative Adversarial Networks (GANs) is based on two models, the Generator (G) and the Discriminator (D). The Generator maps a latent representation to the data space. In a vanilla GAN, the Generator maps random noise to a desirable representation. For sound synthesis this representation could be raw audio or spectrogram. This desired representation, original or generated, is used as input to the Discriminator which is trained to distinguish between real and fake data. The maximum benefit from GANs is acquired when the Generator produces perfect data and the Discriminator is not able to differentiate between real and fake data.

From a more technical point of view, the Discriminator is trained using only the distribution of the original data. Its purpose is to maximise the probability of identifying real and generated data. On the other hand, the Generator is trained through the Discriminator. Information about the original distribution of the dataset are concealed from it and its aim is to minimise the error of the Discriminator. This minimax game can be summarised by the Eq. 8.

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x\backsim p_{data}(x)}[\log D(x)] \\ +\mathbb{E}_{z\backsim p_z(z)}[\log(1 - D(G(z)))]\end{aligned} \tag{8}$$

On the field of sound generation a variety of implementations have been proposed using numerous representations. In [30], spectrograms were generated using upsampling convolutions for fast generation while in [32], they investigated whether waveform or spectrograms are more effective for GANs applying the Wasserstein loss function. In Parallel WaveGAN [60], a teacher-student scheme was adopted using non-autoregressive WaveNet in order to improve WaveGAN's efficiency. Yamamoto et al [16] applied GANs using a IAF generator optimised by a probability density distillation algorithm. Also, in GAN-TTS [13], they examined an ensemble of Discriminators to generate acoustic features using the Hinge loss function along with [61] [63]. Lastly, GANs have also been applied in a variety of applications such as text-to-speech applications [19], speech synthesis [66] [67], speech enhancement [62] or symbolic music generation [45].

## *D. Variational Autoencoders*

An autoencoder is one of the fundamental deep learning architectures consisting of two separate networks, an encoder and a decoder. The encoder compresses the input data into a latent representation while the decoder synthesises data from the learned latent space. The original scheme of an autoencoder was initially created for dimensionality reduction purposes. Although theoretically the decoder bear some resemblance to the generator of GANs, the model is not well qualified for the synthesis of new examples. The network endeavors to reconstruct the original input, therefore it lacks of expressiveness.

To use autoencoders as generative models, variational autoencoders have been proposed [68]. In this architecture, the encoder first models a latent distribution and then the network samples from the distribution to generate latent examples. The success of the variational autoencoders is mostly based on the Kullback–Leibler (KL) divergence used as a loss function. The encoder introduces a new distribution $q(z|X)$ to estimate $p(z|X)$ as much as possible by minimising the KL divergence. The complete loss function is demonstrated in the Eq. 9 where the first term (called *reconstruction loss*) is applied on the final layer and the second term (called *regularization loss*) adjusts the latent layer.

$$\mathcal{L} = \mathbb{E}_{z\backsim q(z|X)}[\log p(X|z)] - D_{KL}[q(z|X)||p(z)] \tag{9}$$

Many variations of VAE have been applied for sound generation topics. In [3] they used VAE with feedforward networks and an additive synthesiser to reproduce monophonic musical notes. In [69] and [40] they applied convolutional layers while in [36] a Variational Parametric Synthesiser was proposed using a conditional VAE.

A modification of variational autoencoders proposed for music synthesis is VQ-VAE [43]. In this approach, the network is trained to encode the input data into a sequence of discrete tokens. Jukebox introduces this method to flatten the data and process it using autoregressive Transformers.

## V. Evaluation

Although in the last decade generative models presented significant improvements, a definitive evaluation process still remains an open question. Many mathematical metrics have been proposed for perceptually evaluating the generative sound and usually a transformation to another audio representation have been adopted. However, despite the numerous attempts, none of these metrics are as reliable as the subjective evalution of human listeners.

### A. Perceptual Evaluation

Human evaluation usually accounts for the mean opinion score between a group of listeners. To conduct the study, many researchers used crowdMOS [70], a user-friendly toolkit for performing listening evaluations. As well as the mean opinion score, a confidence interval is also been computed. Furthermore, in order to attract an accountable number of subjects with specific characteristics, Amazon Mechanical Turk has been widely used. In many cases, raters have been asked to pass a hearing test [21], keep headphones on [61] [11], or only native speakers for evaluating speech have been asked [60] [61] [54].

In these mean opinion score tests, subjects have been asked to rate a sound in a five-point Likert scale in terms of pleasantness [21], naturalness [11] [13] [63], sound quality [17] or speaker diversity [32]. In addition, subjects have been requested to express a preference between sounds of two generative models hearing the same pitch [30] or speech [17] [11]. Finally, for evaluating WaveGAN [32], humans listened to digits between one to ten and were asked to indicate which number they heard.

### B. Number of Statistically-Different Bins

The Number of Statistically-Different Bins (NDB) is a metric for unconditional generative models in order to estimate the diversity of the synthesised examples. Clustering techniques are applied on the training data creating cells of similar properties. Then, the same algorithm tries to categorise the generated data into the cells. If a generated example does not belong to a predefined cluster, then the generated sound is statistically significantly different.

GANSynth [30] used $k$-means to map the log spectrogram of the generated sound into $k = 50$ Voronoi cells. As well as Mean Opinion Score and the Number of Statistically-Different Bins, GANSynth also used Inception Score, Pitch Accuracy and Pitch Entropy and Frechet Inception Distance for evaluation purposes. The rest of the metrics will be analysed in the following sections. A similar set of evaluation metrics has also been adopted by [50] including NDB.

### C. Inception Score

The Inception Score (IS) is another perceptual metric which correlates with human evaluation and is mostly adopted by GANs. For the Inception Score, a pre-trained Inception classifier is applied to the output of the generative model. In order to measure the diversity of the synthesised data, the IS calculates the KL divergence between the model scores $P(y|x)$ and the marginal distribution $P(y)$ as it can be expressed by the Eq.10 for every possible class [71]. The IS is maximised when the generated examples belong to only one class and every class is predicted equally often.

$$IS = exp(\mathbb{E}_x D_{KL}(P(y|x)||P(y))) \tag{10}$$

In [30] and [7], a pitch classifier is trained on spectrograms of the NSynth dataset while in WaveGAN [32], the classifier is trained on normalised log mel-spectrograms having zero mean and unit variance. Finally, metrics like Pitch Accuracy and Pitch Entropy or a nearest neighbour technique have been adopted by GANSynth and WaveGAN respectively to further evaluate the efficiency of their Inception Score. Finally, in [50] they also applied a modified inception score.

### D. Distances-based measurements

This evaluation category includes metrics that measure the distance between representations of the original data and the distribution of the generated examples. Binkowski et al. proposed two distance-based metrics, the Fréchet DeepSpeech Distance (FDSD) and the Kernel DeepSpeech Distance (KDSD) [13] for evaluating their text-to-speech model. The two metrics make use of the the Fréchet distance and the Maximum Mean Discrepancy respectively on audio features extracted by a speech recognition model.

The Fréchet or 2-Wasserstein distance has been proposed by other research papers as well. Engel et al [30] applied the Fréchet Inception Distance on features extracted by a pitch classifier while Kilgour et al [72] used this distance to measure the intensity of a distortion in generated sound. However, although many researchers report successful results using 2-Wasserstein, Donahue et al [63] reported that a similar evaluation metric did not produce a desirable outcome in their experiments.

Distances-based measurements have also been investigated individually by separate parameter estimations. In [3] distances between the generated loudness and fundamental frequency of synthesised and training data are used.

### E. Spectral Convergence

The Spectral Convergence expresses the mean difference between the original and the generated spectrogram. It has been applied by [43] [20] [57] [40] in order to evaluate their synthesised music. The Spectral Convergence can be expressed by the Eq.11 which is also identified as the minimization process of the Griffin-Lim algorithm.

$$SC = \sqrt{\frac{\sum_{n,m} |S(n,m) - \widetilde{S}(n,m)|^2}{\sum_{n,m} S(n,m)}} \quad (11)$$

### *F. Log Likelihood*

A final evaluation metric includes a Negative Log Likelihood (NLL) [17] [15] and an objective Conditional Log Likelihood (CLL) [14] usually measured in bits per audio sample.

## VI. Conclusion

The choice of audio representation is one of the most significant factors in the development of deep learning models for sound synthesis. Numerous representations have been proposed by previous researchers focusing on different properties. Raw audio is a direct representation demanding notable memory and computational cost. It is also not considered for evaluating purposes since different waveforms can perceptually produce the same sound. Spectrograms can overcome some of the disadvantages of raw audio and have been considered as an alternative for training as well as for evaluation. However, reconstructing the original sound from its spectrogram is a challenging task since it may produce sound suffering from distortions and lack of phase coherence. Recently, other audio representations have received much attention such as latent representations, embeddings and acoustic features but they all require a powerful decoder. The choice of audio representation is still very much dependent on the application.

## VII. Acknowledgments

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission'.

## References

[1] H. Gm, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Computer Science Review*, vol. 38, p. 100285, Nov. 2020.

[2] M. Huzaifah and L. Wyse, "Deep generative models for musical audio synthesis," *arXiv:2006.06426 [cs, eess, stat]*, Nov. 2020. arXiv: 2006.06426.

[3] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," *arXiv:2001.04643 [cs, eess, stat]*, Jan. 2020. arXiv: 2001.04643.

[4] J.-M. Valin and J. Skoglund, "LPCNET: Improving Neural Speech Synthesis through Linear Prediction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 5891–5895, IEEE, May 2019.

[5] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A Comparison of Recent Neural Vocoders for Speech Signal Reconstruction," in *10th ISCA Speech Synthesis Workshop*, pp. 7–12, ISCA, Sept. 2019.

[6] M. Blaauw and J. Bonada, "A Neural Parametric Singing Synthesizer," *arXiv:1704.03809 [cs]*, Aug. 2017. arXiv: 1704.03809.

[7] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," *arXiv:1704.01279 [cs]*, Apr. 2017. arXiv: 1704.01279.

[8] R. Manzelli, V. Thakkar, A. Siahkamari, and B. Kulis, "Conditioning Deep Generative Raw Audio Models for Structured Automatic Music," *arXiv:1806.09905 [cs, eess, stat]*, June 2018. arXiv: 1806.09905.

[9] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[10] C. E. Shannont, "Communication in the Presence of Noise," *PROCEEDINGS OF THE I.R.E.*, p. 12, 1949.

[11] A. v. d. 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:1609.03499 [cs]*, Sept. 2016. arXiv: 1609.03499.

[12] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," *arXiv:1711.10433 [cs]*, Nov. 2017. arXiv: 1711.10433.

[13] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High Fidelity Speech Synthesis with Adversarial Networks," *arXiv:1909.11646 [cs, eess]*, Sept. 2019. arXiv: 1909.11646.

[14] S. Kim, S.-g. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet : A Generative Flow for Raw Audio," *arXiv:1811.02155 [cs, eess]*, May 2019. arXiv: 1811.02155.

[15] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," *arXiv:1612.07837 [cs]*, Feb. 2017. arXiv: 1612.07837.

[16] R. Yamamoto, E. Song, and J.-M. Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *arXiv:1904.04472 [cs, eess]*, Aug. 2019. arXiv: 1904.04472.

[17] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," *arXiv:1802.08435 [cs, eess]*, June 2018. arXiv: 1802.08435.

[18] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," *arXiv:1703.10135 [cs]*, Apr. 2017. arXiv: 1703.10135.

[19] P. Neekhara, C. Donahue, M. Puckette, S. Dubnov, and J. McAuley, "Expediting TTS Synthesis with Adversarial Vocoding," *arXiv:1904.07944 [cs, eess]*, July 2019. arXiv: 1904.07944.

[20] S. O. Arik, H. Jun, and G. Diamos, "Fast Spectrogram Inversion using Multi-head Convolutional Neural Networks," *IEEE Signal Processing Letters*, vol. 26, pp. 94–98, Jan. 2019. arXiv: 1808.06719.

[21] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," *arXiv:1811.00002 [cs, eess, stat]*, Oct. 2018. arXiv: 1811.00002.

[22] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-Autoregressive Neural Text-to-Speech," *arXiv:1905.08459 [cs, eess]*, June 2020. arXiv: 1905.08459.

[23] C. Aouameur, P. Esling, and G. Hadjeres, "Neural Drum Machine : An Interactive System for Real-time Synthesis of Drum Sounds," *arXiv:1907.02637 [cs, eess]*, Nov. 2019. arXiv: 1907.02637.

[24] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech," *arXiv:1905.09263 [cs, eess]*, Nov. 2019. arXiv: 1905.09263.

[25] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," *arXiv:2006.04558 [cs, eess]*, Mar. 2021. arXiv: 2006.04558.

[26] R. Liu, B. Sisman, F. Bao, G. Gao, and H. Li, "WaveTTS: Tacotron-based TTS with Joint Time-Frequency Domain Loss," *arXiv:2002.00417 [cs, eess]*, Apr. 2020. arXiv: 2002.00417.

[27] S. Vasquez and M. Lewis, "MelNet: A Generative Model for Audio in the Frequency Domain," *arXiv:1906.01083 [cs, eess, stat]*, June 2019. arXiv: 1906.01083.

[28] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical

Timbre Transfer," *arXiv:1811.09620 [cs, eess, stat]*, May 2019. arXiv: 1811.09620.

[29] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "CONSTRUCTING AN INVERTIBLE CONSTANT-Q TRANSFORM WITH NON-STATIONARY GABOR FRAMES," p. 8, 2011.

[30] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSYNTH: ADVERSARIAL NEURAL AUDIO SYNTHESIS," p. 17, 2019.

[31] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech," p. 13.

[32] C. Donahue, J. McAuley, and M. Puckette, "ADVERSARIAL AUDIO SYNTHESIS," p. 16, 2019.

[33] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis," p. 10.

[34] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *arXiv:1904.12088 [cs, eess, stat]*, Nov. 2019. arXiv: 1904.12088.

[35] A. Defossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, "SING: Symbol-to-Instrument Neural Generator," p. 11.

[36] K. Subramani, P. Rao, and A. D'Hooge, "Vapar Synth - A Variational Parametric Model for Audio Synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Barcelona, Spain), pp. 796–800, IEEE, May 2020.

[37] L. Juvela, B. Bollepalli, V. Tsiaras, and P. Alku, "GlotNet—A Raw Waveform Model for the Glottal Excitation in Statistical Parametric Speech Synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1019–1030, June 2019.

[38] S. Yang, L. Xie, X. Chen, X. Lou, X. Zhu, D. Huang, and H. Li, "Statistical Parametric Speech Synthesis Using Generative Adversarial Networks Under A Multi-task Learning Framework," *arXiv:1707.01670 [cs]*, July 2017. arXiv: 1707.01670.

[39] G. E. Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Deep Encoder-Decoder Models for Unsupervised Learning of Controllable Speech Synthesis," *arXiv:1807.11470 [cs, eess, stat]*, Sept. 2018. arXiv: 1807.11470.

[40] A. Bitton, P. Esling, and T. Harada, "Neural Granular Sound Synthesis," *arXiv:2008.01393 [cs, eess]*, Aug. 2020. arXiv: 2008.01393.

[41] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics," *arXiv:1805.08501 [cs, eess]*, Oct. 2018. arXiv: 1805.08501.

[42] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Universal audio synthesizer control with normalizing flows," *arXiv:1907.00971 [cs, eess, stat]*, July 2019. arXiv: 1907.00971.

[43] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A Generative Model for Music," *arXiv:2005.00341 [cs, eess, stat]*, Apr. 2020. arXiv: 2005.00341.

[44] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," *arXiv:1703.10847 [cs]*, July 2017. arXiv: 1703.10847.

[45] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," *arXiv:1709.06298 [cs, eess, stat]*, Nov. 2017. arXiv: 1709.06298.

[46] H. H. Mao, T. Shin, and G. W. Cottrell, "DeepJ: Style-Specific Music Generation," *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 377–382, Jan. 2018. arXiv: 1801.00887.

[47] Y. Zhao, S. Takaki, H.-T. Luong, J. Yamagishi, D. Saito, and N. Minematsu, "Wasserstein GAN and Waveform Loss-Based Acoustic Model Training for Multi-Speaker Text-to-Speech Synthesis Systems Using a WaveNet Vocoder," *IEEE Access*, vol. 6, pp. 60478–60488, 2018.

[48] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "Fftnet: A Real-Time Speaker-Dependent Neural Vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Calgary, AB), pp. 2251–2255, IEEE, Apr. 2018.

[49] S. Arik, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-Speaker Neural Text-to-Speech," *arXiv:1705.08947 [cs]*, Sept. 2017. arXiv: 1705.08947.

[50] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A Versatile Diffusion Model for Audio Synthesis," *arXiv:2009.09761 [cs, eess, stat]*, Mar. 2021. arXiv: 2009.09761.

[51] J. Boilard, P. Gournay, and R. Lefebvre, "A Literature Review of WaveNet: Theory, Application and Optimization," p. 17.

[52] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," *arXiv:1712.05884 [cs]*, Feb. 2018. arXiv: 1712.05884.

[53] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," *arXiv:1710.07654 [cs, eess]*, Feb. 2018. arXiv: 1710.07654.

[54] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural Speech Synthesis with Transformer Network," *arXiv:1809.08895 [cs]*, Jan. 2019. arXiv: 1809.08895.

[55] M. Angrick, C. Herff, E. Mugler, M. C. Tate, M. W. Slutzky, D. J. Krusienski, and T. Schultz, "Speech synthesis from ECoG using densely connected 3D convolutional neural networks," *Journal of Neural Engineering*, vol. 16, p. 036019, June 2019.

[56] S. O. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time Neural Text-to-Speech," p. 10.

[57] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-Dependent WaveNet Vocoder," in *Interspeech 2017*, pp. 1118–1122, ISCA, Aug. 2017.

[58] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," *arXiv:1810.12247 [cs, eess, stat]*, Jan. 2019. arXiv: 1810.12247.

[59] Z.-H. Ling, Y. Ai, Y. Gu, and L.-R. Dai, "Waveform Modeling and Generation Using Hierarchical Recurrent Neural Networks for Speech Bandwidth Extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 883–894, May 2018.

[60] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *arXiv:1910.11480 [cs, eess]*, Feb. 2020. arXiv: 1910.11480.

[61] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," *arXiv:1910.06711 [cs, eess]*, Dec. 2019. arXiv: 1910.06711.

[62] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network," *arXiv:1703.09452 [cs]*, June 2017. arXiv: 1703.09452.

[63] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-End Adversarial Text-to-Speech," *arXiv:2006.03575 [cs, eess]*, Mar. 2021. arXiv: 2006.03575.

[64] W. Ping, K. Peng, K. Zhao, and Z. Song, "WaveFlow: A Compact Flow-based Model for Raw Audio," p. 11.

[65] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (South Brisbane, Queensland, Australia), pp. 4225–4229, IEEE, Apr. 2015.

[66] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in *2018 26th European Signal Processing Conference (EUSIPCO)*, (Rome), pp. 2514–2518, IEEE, Sept. 2018.

[67] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 84–96, Jan. 2018.

[68] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014. arXiv: 1312.6114.

[69] A. Pandey and D. Wang, "A New Framework for CNN-Based Speech Enhancement in the Time Domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1179–1188, July 2019.

[70] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "crowdMOS: An Approach for Crowdsourcing Mean Opinion Score Studies," p. 4.

[71] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *arXiv:1606.03498 [cs]*, June 2016. arXiv: 1606.03498.

[72] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms," *arXiv:1812.08466 [cs, eess]*, Jan. 2019. arXiv: 1812.08466.