

# Diffusion-Based Vocoders

2nd May 2023

Drake

## AI song featuring fake Drake and Weeknd vocals pulled from streaming services

The song, called **Heart on My Sleeve**, has been removed from TikTok, Spotify and YouTube for 'infringing content created with generative AI'



📷 For real ... Drake (right) and the Weeknd performing in London in 2014. Photograph: Jeff Barclay/REX Shutterstock

A song featuring AI-generated vocals purporting to be Drake and the Weeknd has been pulled from streaming services by Universal Music Group (UMG) after going viral over the weekend. The label condemned the song, called **Heart on My Sleeve**, for "infringing content created with generative AI".

Laura Snapes

Tue 18 Apr 2023 10:37 BST



DARK BRANDON

## Fake Biden Speeches Are the Hottest Trend in AI Voice Tech

While some bad actors are using deepfakes to sow misinformation, shitposters are just imagining a funnier version of the president

BY MILES KLEE

FEBRUARY 22, 2023



President Biden, whose voice can be easily copied with AI. PHOTO ILLUSTRATION BASED ON PHOTOGRAPH BY WIN MCNAMEE/GETTY IMAGES

# Deep Learning-Based Speech Synthesis

Generally split into two parts:

1. Text-to-speech (TTS) model:

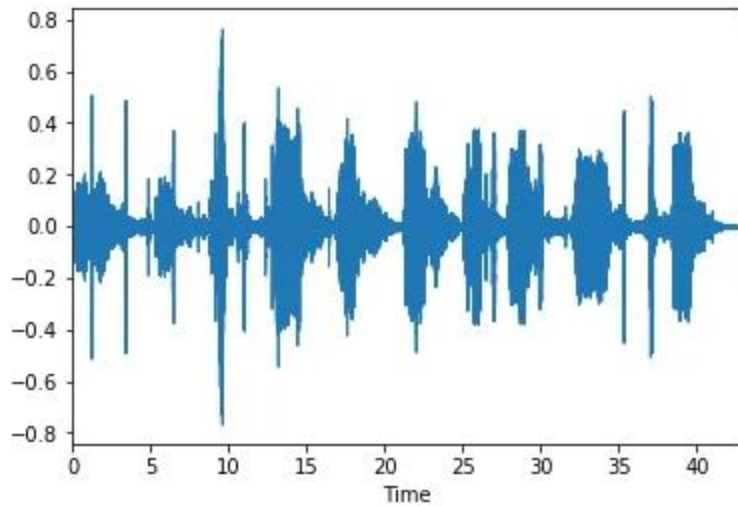
- Text  $\rightarrow$  characters/phonemes  $\rightarrow$  acoustic features (usually mel-spectrograms)

2. Vocoder

- Acoustic features  $\rightarrow$  time-domain waveform (i.e. audio)

# Audio Representations - Fourier Transform

Time-Domain,  $(x_n)$

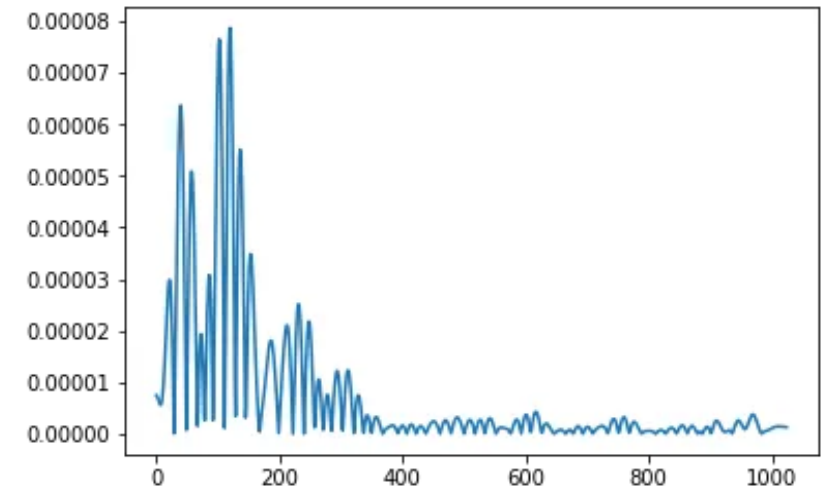


$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

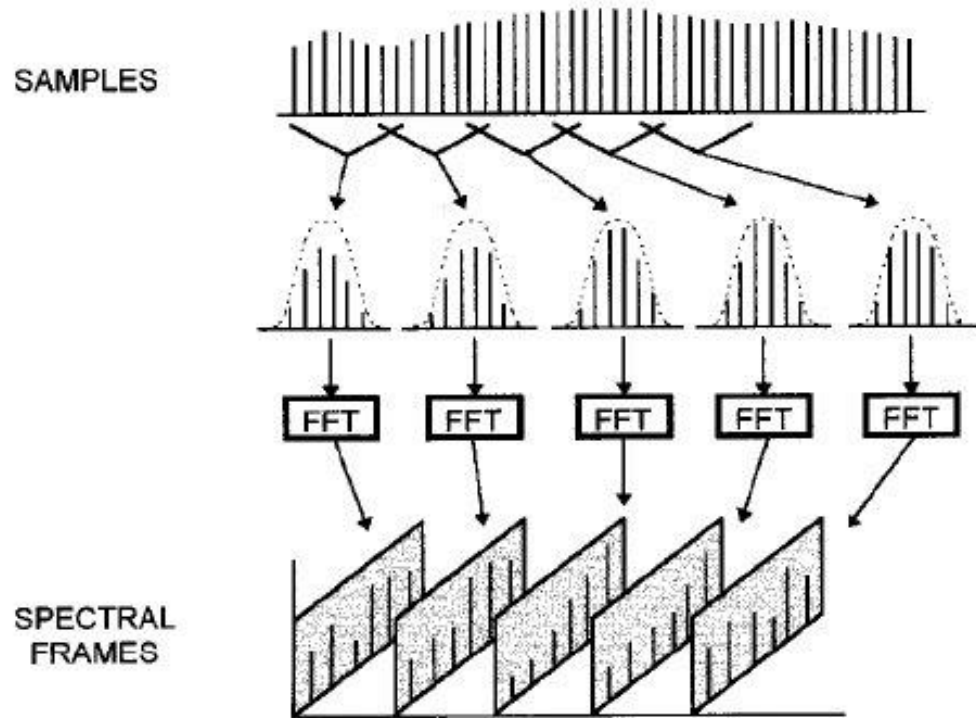
(Discrete)  
Fourier Transform



Frequency-Domain,  $(X_k)$

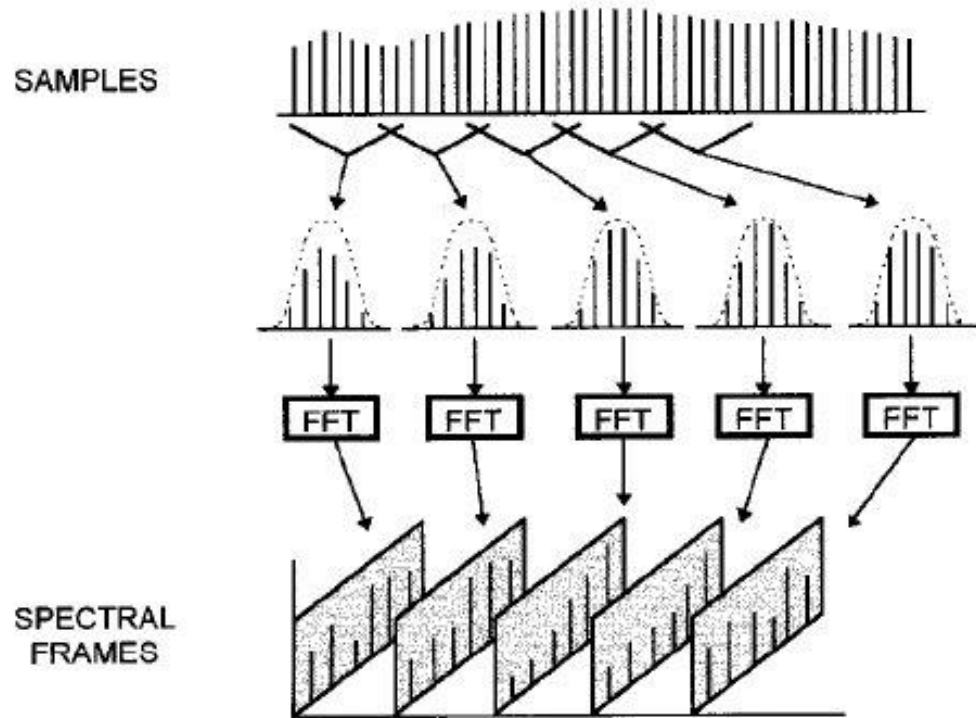


# Audio Representations – Short-Time Fourier Transform

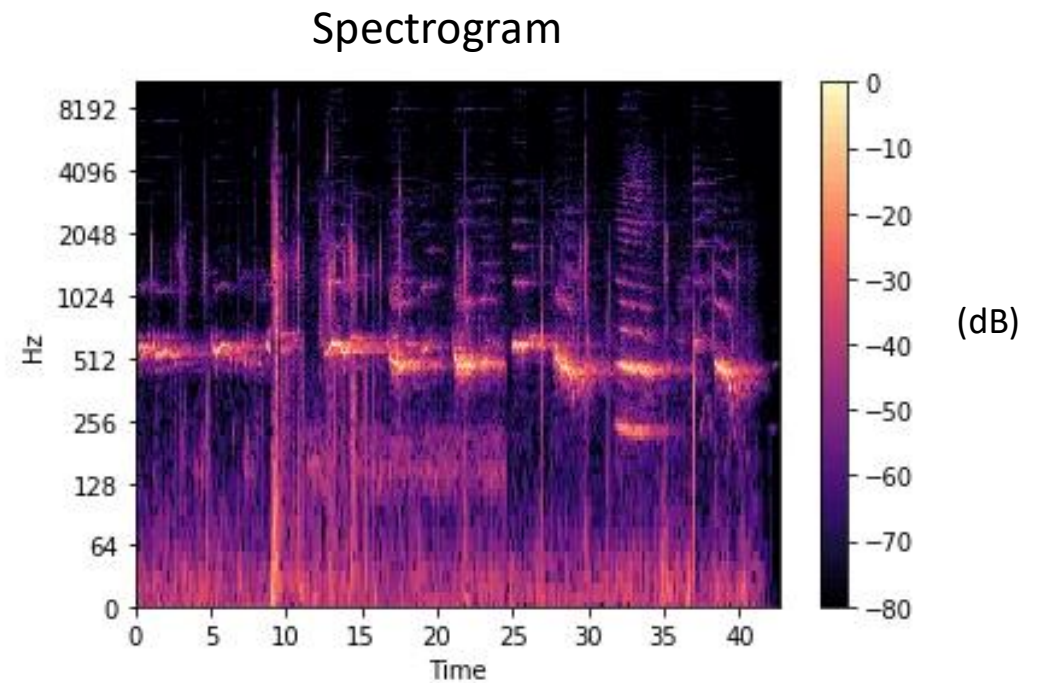


Fischman, Rajmil. 'The Phase Vocoder: Theory and Practice'.  
Organised Sound 2 (1 August 1997): 127–45.  
<https://doi.org/10.1017/S1355771897009060>.

# Audio Representations – Short-Time Fourier Transform



Fischman, Rajmil. 'The Phase Vocoder: Theory and Practice'.  
Organised Sound 2 (1 August 1997): 127–45.  
<https://doi.org/10.1017/S1355771897009060>.

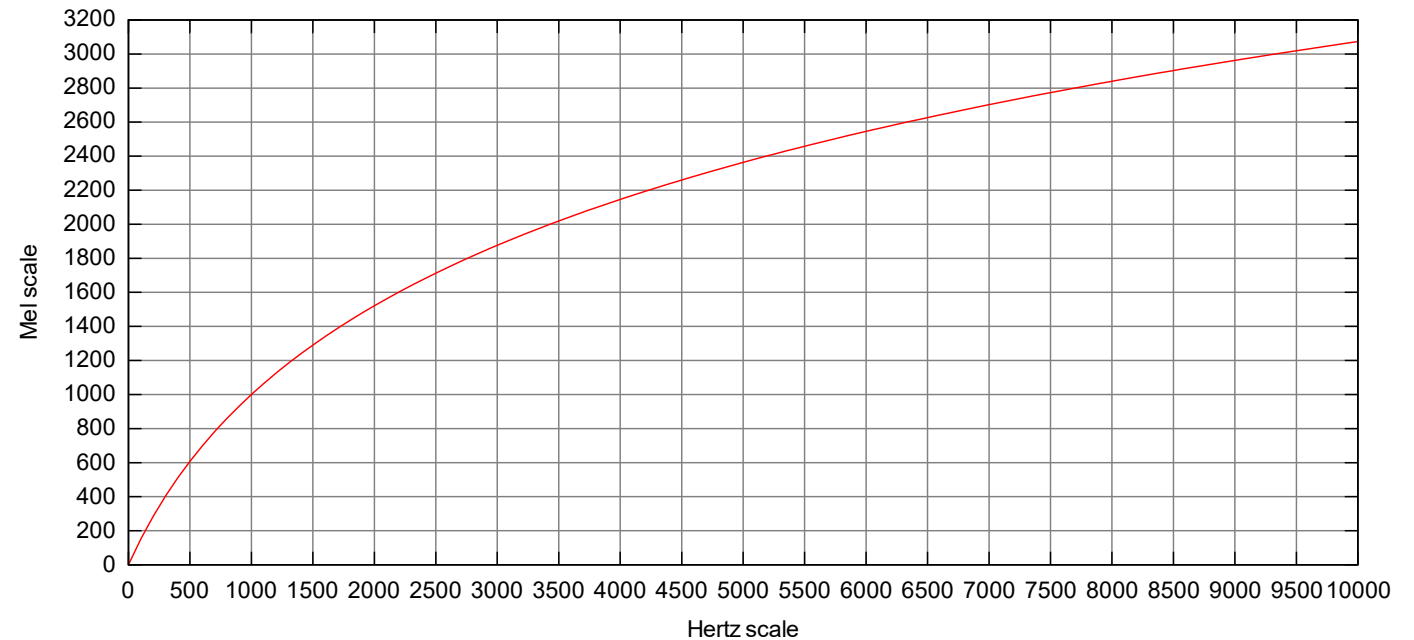


<https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0>

# Mel-Spectrogram

- Convert frequencies from Hz to the log-based mel scale.
- Better captures aural information as humans process it.
- More focus is given to differences between low frequencies than between high frequencies.

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right)$$



By Krishna Vedala - This W3C-unspecified plot was created with Gnuplot., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3775197>

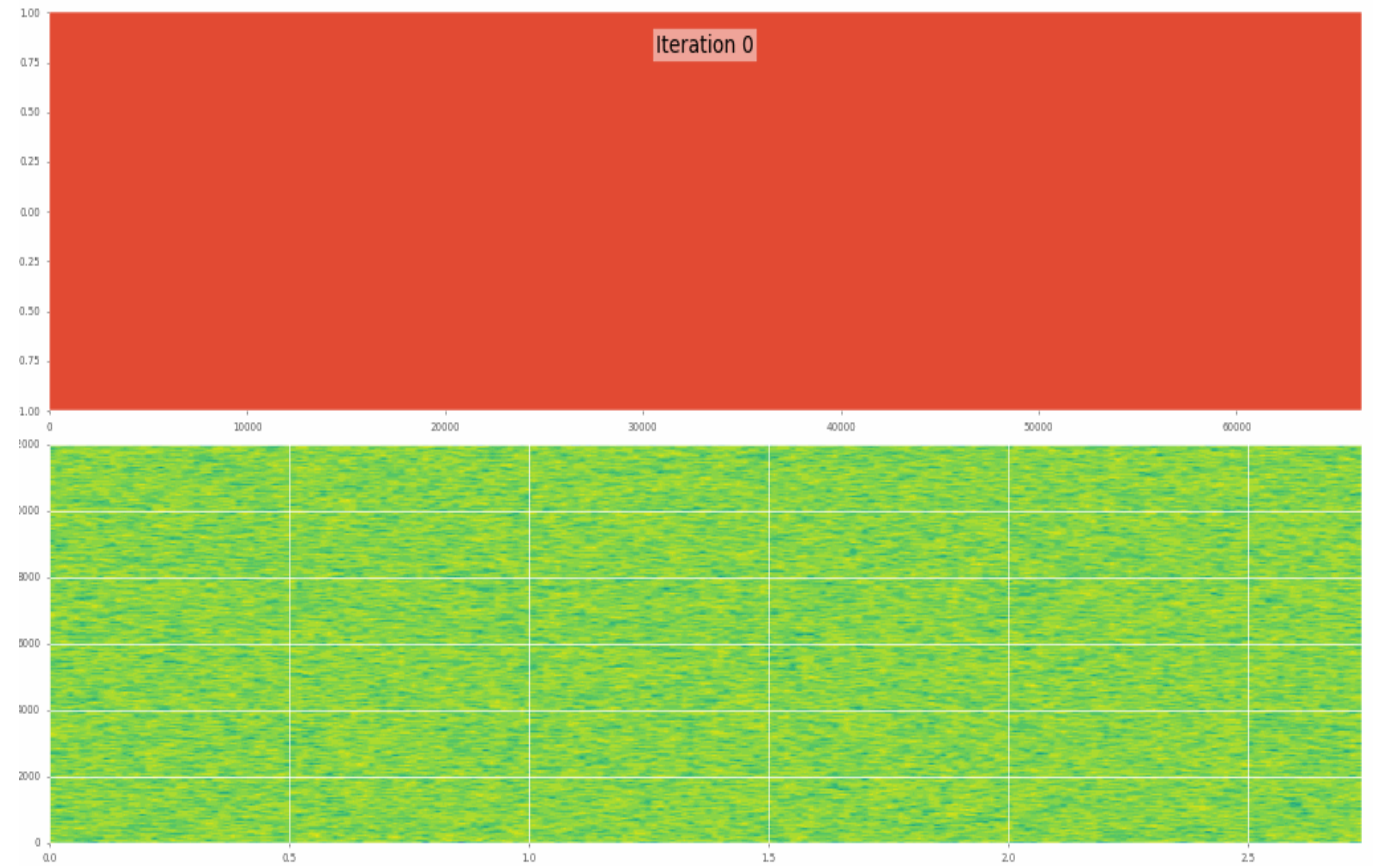
# Vocoders

- Vocoders are used to convert from (mel-) spectrograms to waveforms.
- This is non-trivial for a few reasons:
  - Finite resolution of spectrograms
  - Finite sample-rate of desired waveform
  - Information lost in windowing process of STFT (especially if windowing filters are used)
  - Fourier transform outputs are complex-valued, but we only plot the power (i.e. modulus) in a spectrogram—the phase (i.e. argument) of each frequency bin is lost.



# Neural Vocoder Approaches

- RNNs (e.g. with LSTM [1])
- GANs (e.g. WaveGAN [2])
- Autoregressive models (e.g. WaveNet [3])
  - Good, but slow!
- Diffusion models (e.g. DiffWave [4] & WaveGrad [5])
  - Non-autoregressive and therefore faster to sample from (we hope)



<https://wavegrad.github.io>

# Diffusion Models

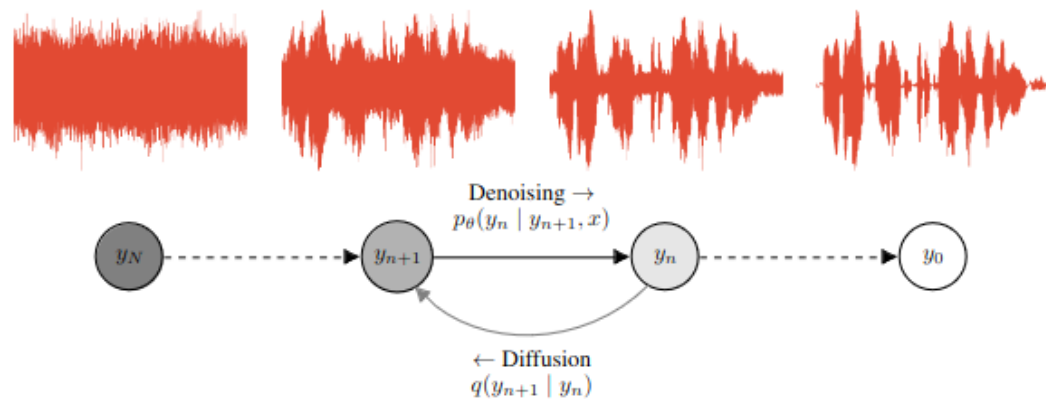


Figure 2: WaveGrad directed graphical model for training, conditioned on iteration index.  $q(y_{n+1} | y_n)$  iteratively adds Gaussian noise to the signal starting from the waveform  $y_0$ .  $q(y_{n+1} | y_0)$  is the noise distribution used for training. The inference denoising process progressively removes noise, starting from Gaussian noise  $y_N$ , akin to Langevin dynamics. Adapted from [Ho et al. \(2020\)](#).

# Diffusion Models

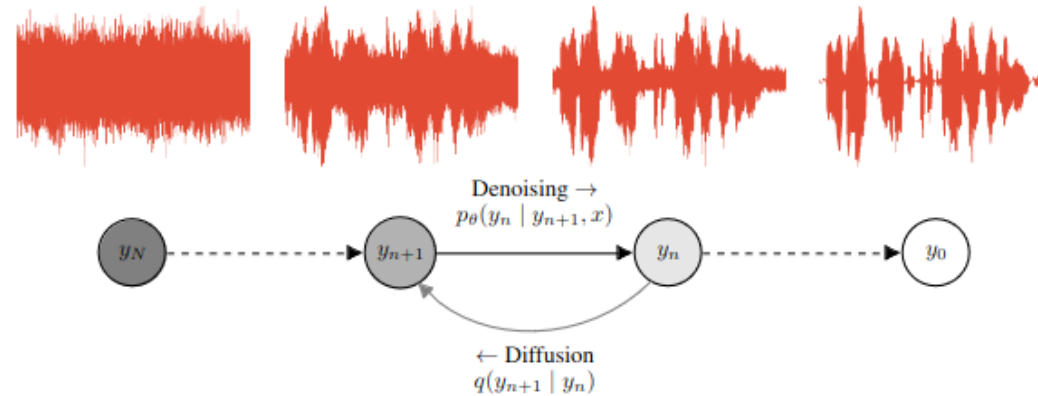


Figure 2: WaveGrad directed graphical model for training, conditioned on iteration index.  $q(y_{n+1} | y_n)$  iteratively adds Gaussian noise to the signal starting from the waveform  $y_0$ .  $q(y_{n+1} | y_0)$  is the noise distribution used for training. The inference denoising process progressively removes noise, starting from Gaussian noise  $y_N$ , akin to Langevin dynamics. Adapted from [Ho et al. \(2020\)](#).

The diffusion process is defined through the Markov chain:

$$q(y_{1:N} | y_0) := \prod_{n=1}^N q(y_n | y_{n-1}),$$

which adds Gaussian noise at each iteration, for some fixed constant noise schedule  $\beta_1, \dots, \beta_N$ , via  $q(y_n | y_{n-1}) := \mathcal{N}\left(y_n; \sqrt{(1 - \beta_n)} y_{n-1}, \beta_n I\right)$

# Diffusion Models

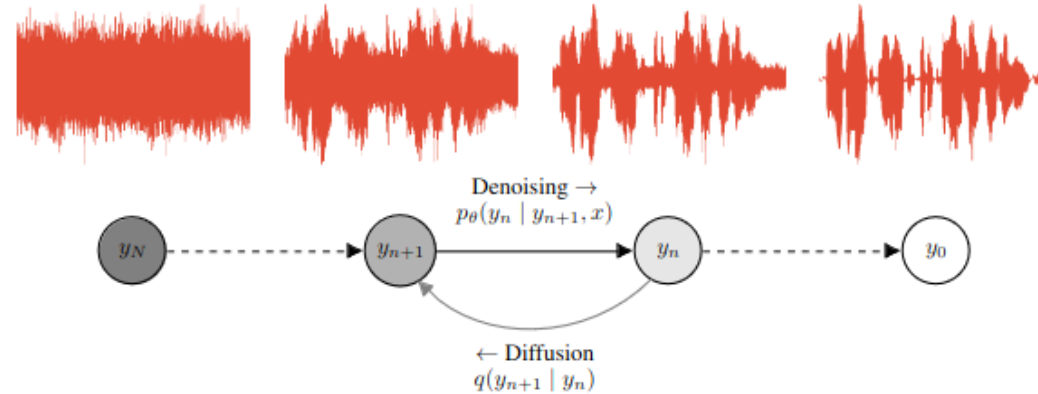


Figure 2: WaveGrad directed graphical model for training, conditioned on iteration index.  $q(y_{n+1} | y_n)$  iteratively adds Gaussian noise to the signal starting from the waveform  $y_0$ .  $q(y_{n+1} | y_0)$  is the noise distribution used for training. The inference denoising process progressively removes noise, starting from Gaussian noise  $y_N$ , akin to Langevin dynamics. Adapted from [Ho et al. \(2020\)](#).

We can write the output of the diffusion (forward) process at any time step  $n$  as:

$$y_n = \sqrt{\bar{\alpha}_n} y_0 + \sqrt{(1 - \bar{\alpha}_n)} \epsilon$$

where  $\epsilon \sim \mathcal{N}(0, I)$ ,  $\alpha_n := 1 - \beta_n$  and  $\bar{\alpha}_n := \prod_{s=1}^n \alpha_s$ .

# Diffusion Models

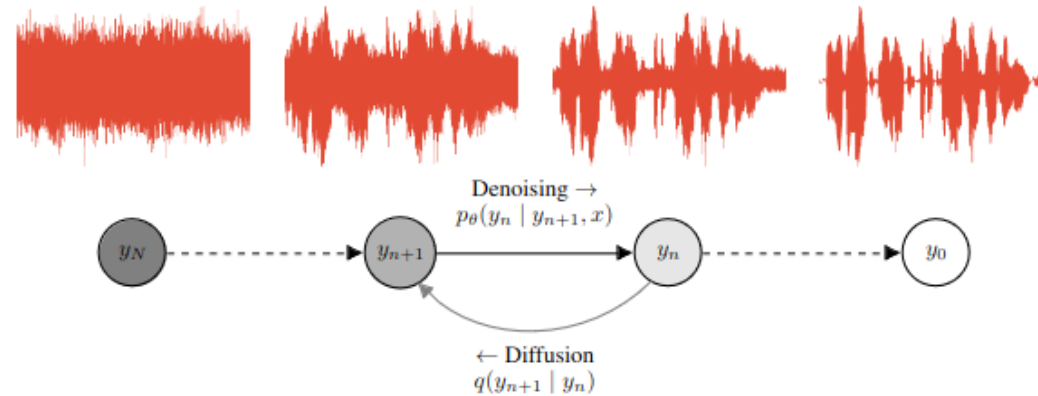


Figure 2: WaveGrad directed graphical model for training, conditioned on iteration index.  $q(y_{n+1} | y_n)$  iteratively adds Gaussian noise to the signal starting from the waveform  $y_0$ .  $q(y_{n+1} | y_0)$  is the noise distribution used for training. The inference denoising process progressively removes noise, starting from Gaussian noise  $y_N$ , akin to Langevin dynamics. Adapted from [Ho et al. \(2020\)](#).

In the denoising (backwards) process  $p_\theta(x_{t-1} | x_t)$ , we use a neural net to predict some  $\epsilon_\theta$  that can be used to repeatedly denoise samples (with the first sample originally taken from a standard Gaussian distribution:  $y_N \sim \mathcal{N}(0, I)$ ).

# Diffusion Models

We train this neural net  $\epsilon_\theta$  by optimising the ELBO, which can be written as follows (using derivations in [6]):

**Proposition 1.** (*Ho et al., 2020*) Suppose a series of fixed schedule  $\{\beta_t\}_{t=1}^T$  are given. Let  $\epsilon \sim \mathcal{N}(0, I)$  and  $x_0 \sim q_{\text{data}}$ . Then, under the parameterization in Eq. (5), we have

$$-\text{ELBO} = c + \sum_{t=1}^T \kappa_t \mathbb{E}_{x_0, \epsilon} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|_2^2 \quad (6)$$

for some constants  $c$  and  $\kappa_t$ , where  $\kappa_t = \frac{\beta_t}{2\alpha_t(1-\bar{\alpha}_{t-1})}$  for  $t > 1$ , and  $\kappa_1 = \frac{1}{2\alpha_1}$ .

# WaveGrad Training

---

**Algorithm 1** Training. WaveGrad directly conditions on the continuous noise level  $\sqrt{\bar{\alpha}}$ .  $l$  is from a predefined noise schedule.

---

- 1: **repeat**
  - 2:    $y_0 \sim q(y_0)$
  - 3:    $s \sim \text{Uniform}(\{1, \dots, S\})$
  - 4:    $\sqrt{\bar{\alpha}} \sim \text{Uniform}(l_{s-1}, l_s)$
  - 5:    $\epsilon \sim \mathcal{N}(0, I)$
  - 6:   Take gradient descent step on  
           $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}} y_0 + \sqrt{1 - \bar{\alpha}} \epsilon, x, \sqrt{\bar{\alpha}})\|_1$
  - 7: **until** converged
-

# WaveGrad Training

- The authors found improved performance by making a few alterations:
  - Training with a continuous hierarchical noise schedule
  - Optimising an L1 norm rather than an L2 norm

---

**Algorithm 1** Training. WaveGrad directly conditions on the continuous noise level  $\sqrt{\bar{\alpha}}$ .  $l$  is from a predefined noise schedule.

---

```
1: repeat
2:    $y_0 \sim q(y_0)$ 
3:    $s \sim \text{Uniform}(\{1, \dots, S\})$ 
4:    $\sqrt{\bar{\alpha}} \sim \text{Uniform}(l_{s-1}, l_s)$ 
5:    $\epsilon \sim \mathcal{N}(0, I)$ 
6:   Take gradient descent step on
        $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}} y_0 + \sqrt{1 - \bar{\alpha}} \epsilon, x, \sqrt{\bar{\alpha}})\|_1$ 
7: until converged
```

---



# WaveGrad Training

- The authors found improved performance by making a few alterations:
  - Training with a continuous hierarchical noise schedule
  - Optimising an L1 norm rather than an L2 norm
- We condition on acoustic information  $x$  (e.g. Mel spectrogram) and also provide the neural net with information of the time/noise schedule.

---

**Algorithm 1** Training. WaveGrad directly conditions on the continuous noise level  $\sqrt{\bar{\alpha}}$ .  $l$  is from a predefined noise schedule.

---

```
1: repeat
2:    $y_0 \sim q(y_0)$ 
3:    $s \sim \text{Uniform}(\{1, \dots, S\})$ 
4:    $\sqrt{\bar{\alpha}} \sim \text{Uniform}(l_{s-1}, l_s)$ 
5:    $\epsilon \sim \mathcal{N}(0, I)$ 
6:   Take gradient descent step on
        $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}} y_0 + \sqrt{1 - \bar{\alpha}} \epsilon, x, \sqrt{\bar{\alpha}})\|_1$ 
7: until converged
```

---

# WaveGrad Sampling

- Sampling denoises pure Gaussian noise.
- In step 4 we're subtracting the noise  $\epsilon_\theta(y_n, x, \sqrt{\bar{\alpha}_n})$  from our previous sample  $y_n$ , along with some scaling based on the noise schedule.
- We add some extra noise  $\sigma_n z$  in all steps other than the final (when  $n=1$ ).

---

**Algorithm 2** Sampling. WaveGrad generates samples following a gradient-based sampler similar to Langevin dynamics.

---

```
1:  $y_N \sim \mathcal{N}(0, I)$ 
2: for  $n = N, \dots, 1$  do
3:    $z \sim \mathcal{N}(0, I)$ 
4:    $y_{n-1} = \frac{\left(y_n - \frac{1-\alpha_n}{\sqrt{1-\bar{\alpha}_n}} \epsilon_\theta(y_n, x, \sqrt{\bar{\alpha}_n})\right)}{\sqrt{\alpha_n}}$ 
5:   if  $n > 1$ ,  $y_{n-1} = y_{n-1} + \sigma_n z$ 
6: end for
7: return  $y_0$ 
```

---

# WaveGrad Architecture

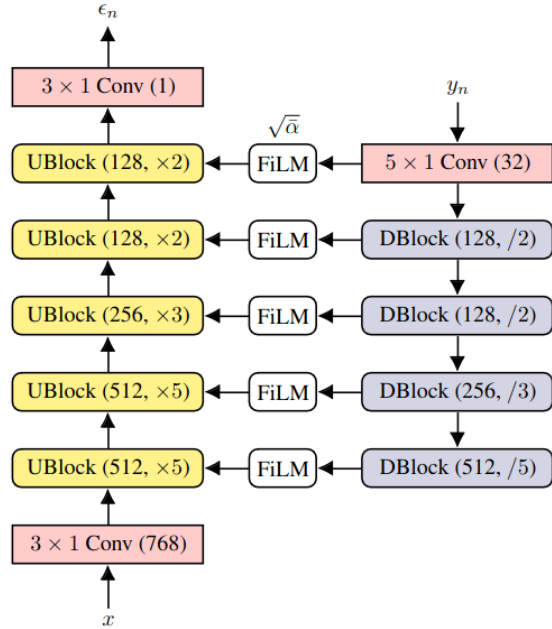






Figure 3: WaveGrad network architecture. The inputs consists of the mel-spectrogram conditioning signal  $x$ , the noisy waveform generated from the previous iteration  $y_n$ , and the noise level  $\sqrt{\bar{\alpha}}$ . The model produces  $\epsilon_n$  at each iteration, which can be interpreted as the direction to update  $y_n$ .

- Downsampling and upsampling blocks help to share sample information across time.
- The feature-wise linear modulation (FiLM) modules also use positional encodings of information from the noise schedule via  $\sqrt{\bar{\alpha}_n}$ .

# WaveGrad Samples

- Conditioned on ground-truth mel-spectrogram
  1. Ground truth audio 
  2. With 1000 denoising steps 
  3. With 25 denoising steps 
  4. WaveFlow (autoregressive model) 

# WaveGrad Results

Table 1: Mean opinion scores (MOS) of various models and their confidence intervals. All models except WaveRNN are non-autoregressive. WaveGrad, Parallel WaveGAN, MelGAN, and Multi-band MelGAN were conditioned on the mel-spectrograms computed from ground truth audio during training. WaveRNN and GAN-TTS used predicted features for training.

Model	MOS ( $\uparrow$ )
WaveRNN	$4.49 \pm 0.04$
Parallel WaveGAN	$3.92 \pm 0.05$
MelGAN	$3.95 \pm 0.06$
Multi-band MelGAN	$4.10 \pm 0.05$
GAN-TTS	$4.34 \pm 0.04$
WaveGrad	
Base (6 iterations, continuous noise levels)	$4.41 \pm 0.03$
Base (1,000 iterations, discrete indices)	$4.47 \pm 0.04$
Large (1,000 iterations, discrete indices)	$4.51 \pm 0.04$
Ground Truth	$4.58 \pm 0.05$

# DiffWave Architecture

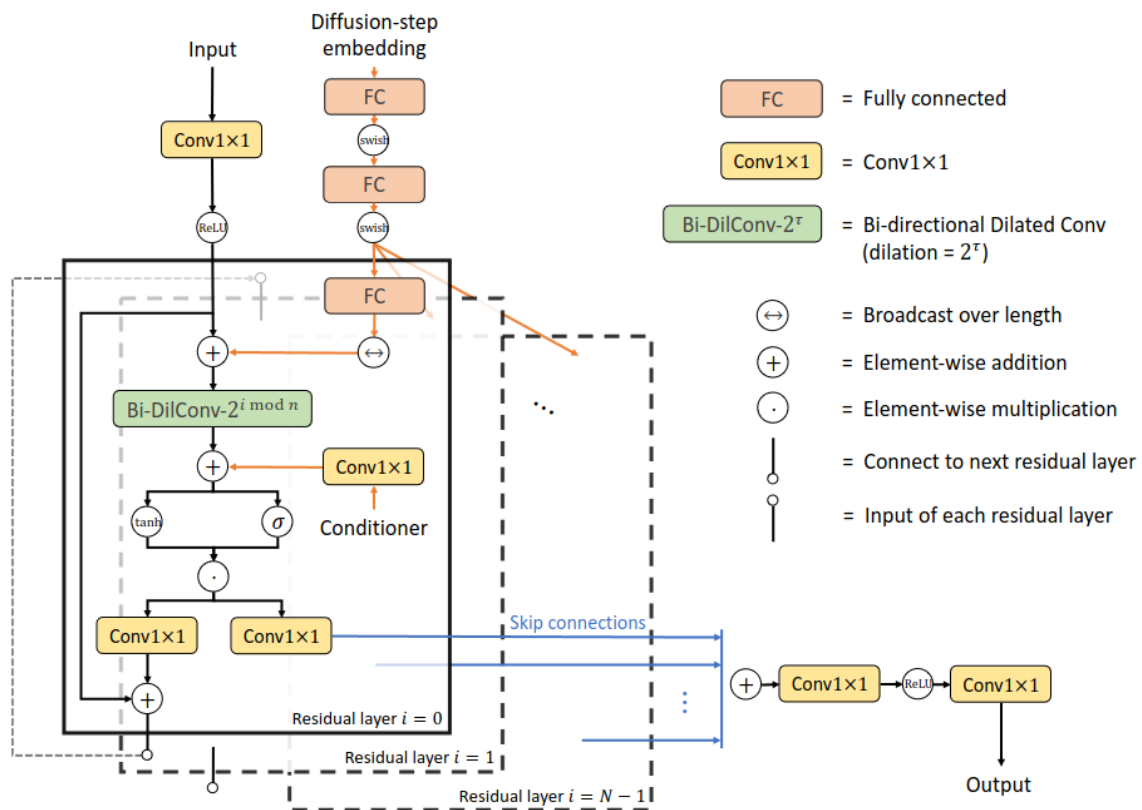


Figure 2: The network architecture of DiffWave in modeling  $\epsilon_\theta : \mathbb{R}^L \times \mathbb{N} \rightarrow \mathbb{R}^L$ .

# DiffWave Architecture

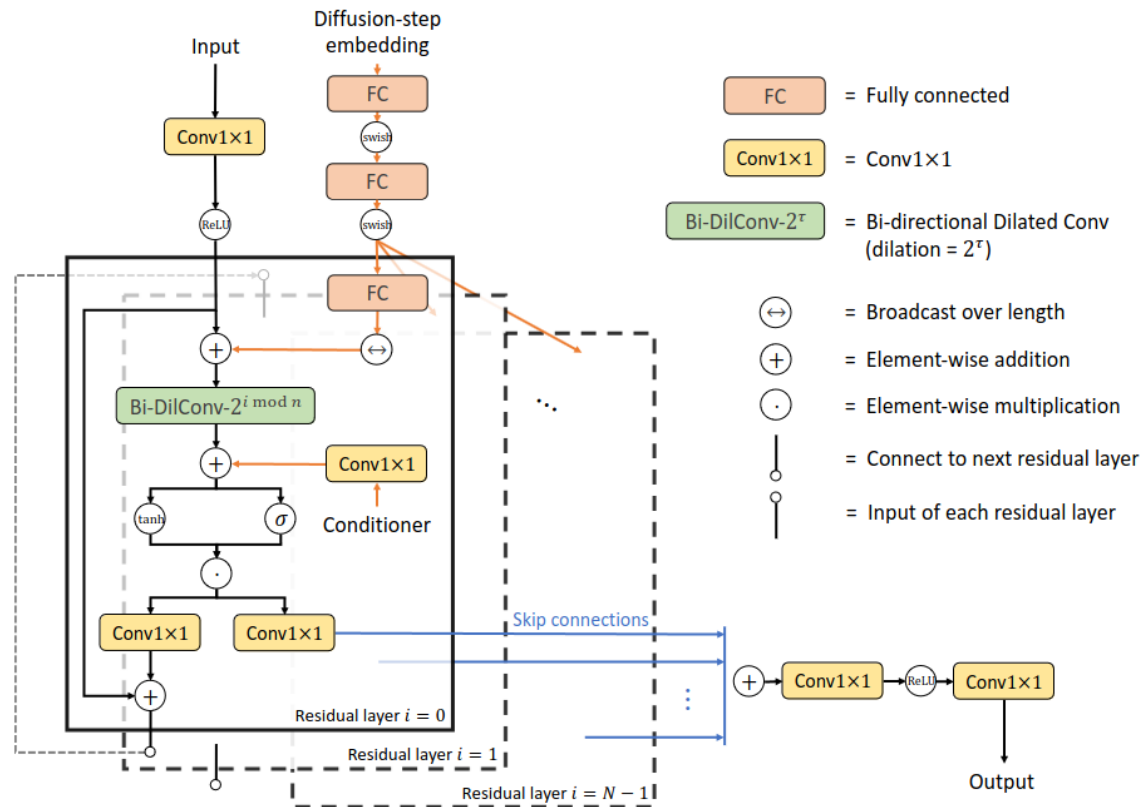
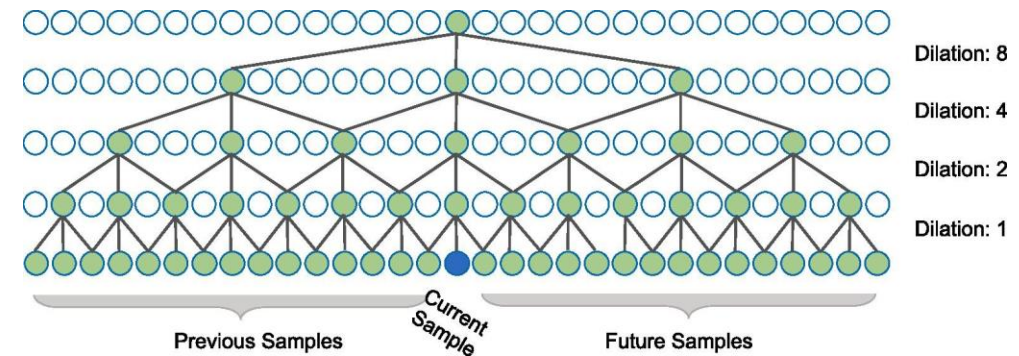


Figure 2: The network architecture of DiffWave in modeling  $\epsilon_\theta : \mathbb{R}^L \times \mathbb{N} \rightarrow \mathbb{R}^L$ .




Bidirectional-dilated convolution stacks:

- Serve similar purpose to WaveGrad's upsampling/downsampling blocks—sharing sample information over time



Jia, Ziyue, Linfeng Yang, Zhenrong Zhang, Hui Liu, and Fannie Kong. 'Sequence to Point Learning Based on Bidirectional Dilated Residual Network for Non-Intrusive Load Monitoring'. International Journal of Electrical Power & Energy Systems 129 (1 July 2021): 106837. <https://doi.org/10.1016/j.ijepes.2021.106837>.

# DiffWave Conditioned on Mel-Spectrogram

1. Ground truth 
2. DiffWave 
3. WaveGlow (autoregressive model) 



# DiffWave Conditioned on Mel-Spectrogram

Table 1: The model hyperparameters, model footprint, and 5-scale Mean Opinion Score (MOS) with 95% confidence intervals for WaveNet, ClariNet, WaveFlow, WaveGlow and the proposed DiffWave on the **neural vocoding** task.  $\uparrow$  means the number is the higher the better, and  $\downarrow$  means the number is the lower the better.

Model	$T$	$T_{\text{infer}}$	layers	res. channels	#param( $\downarrow$ )	MOS( $\uparrow$ )
WaveNet	—	—	30	128	4.57M	<b>4.43</b> $\pm$ 0.10
ClariNet	—	—	60	64	2.17M	4.27 $\pm$ 0.09
WaveGlow	—	—	96	256	87.88M	4.33 $\pm$ 0.12
WaveFlow	—	—	64	64	5.91M	4.30 $\pm$ 0.11
WaveFlow	—	—	64	128	22.25M	4.40 $\pm$ 0.07
DiffWave <sub>BASE</sub>	20	20	30	64	2.64M	4.31 $\pm$ 0.09
DiffWave <sub>BASE</sub>	40	40	30	64	2.64M	4.35 $\pm$ 0.10
DiffWave <sub>BASE</sub>	50	50	30	64	2.64M	<b>4.38</b> $\pm$ 0.08
DiffWave <sub>LARGE</sub>	200	200	30	128	6.91M	<b>4.44</b> $\pm$ 0.07
DiffWave <sub>BASE</sub> (Fast)	50	6	30	64	2.64M	4.37 $\pm$ 0.07
DiffWave <sub>LARGE</sub> (Fast)	200	6	30	128	6.91M	4.42 $\pm$ 0.09
Ground-truth	—	—	—	—	—	4.52 $\pm$ 0.06

# DiffWave Conditioned on Class Labels

- Trained on dataset of digits spoken aloud in English by a variety of speakers (a subset of the Speech Commands dataset [7]).
- Conditioned on class label (digit).

1. WaveNet 

2. DiffWave 

# DiffWave Conditioned on Class Labels

Table 3: The automatic evaluation metrics (Accuracy, FID-class, IS, mIS), and 5-scale MOS with 95% confidence intervals for WaveNet and DiffWave on the **class-conditional** generation task.

Model	Accuracy( $\uparrow$ )	FID-class( $\downarrow$ )	IS( $\uparrow$ )	mIS( $\uparrow$ )	MOS( $\uparrow$ )
WaveNet-128	56.20%	$7.876 \pm 2.469$	3.29	15.8	$1.46 \pm 0.30$
WaveNet-256	60.70%	$6.954 \pm 2.114$	3.46	18.9	$1.58 \pm 0.36$
DiffWave	91.20%	$1.113 \pm 0.569$	6.63	117.4	<b><math>3.50 \pm 0.31</math></b>
DiffWave (deep & thin)	<b>94.00%</b>	<b><math>0.932 \pm 0.450</math></b>	<b>6.92</b>	<b>133.8</b>	$3.44 \pm 0.36$
Trainset	99.06%	$0.000 \pm 0.000$	8.48	281.4	—
Testset	98.76%	$0.044 \pm 0.016$	8.47	275.2	$3.72 \pm 0.28$

- Accuracy: classification accuracy of a ResNeXT classifier [8].
- FID (Fréchet Inception Distance):
  - 2-Wasserstein distance between 1024-long vectors produced by ResNeXT for training set and set of generated samples.
- IS (Inception Score):
  - "measures both quality and diversity of generated samples, and favors generated samples that can be clearly determined by the classifier".
- mIS (modified Inception Score):
  - "measures the within-class diversity of samples in addition to IS"

# DiffWave Unconditional Generation

- Trained on same subset of Speech Commands [7] as before but with no conditioning information.

1. WaveNet 

2. WaveGAN 

3. DiffWave 

# DiffWave Unconditional Generation

Table 2: The automatic evaluation metrics (FID, IS, mIS, AM, and NDB/ $K$ ), and 5-scale MOS with 95% confidence intervals for WaveNet, WaveGAN, and DiffWave on the **unconditional** generation task.  $\uparrow$  means the number is the higher the better, and  $\downarrow$  means the number is the lower the better.




Model	FID( $\downarrow$ )	IS( $\uparrow$ )	mIS( $\uparrow$ )	AM( $\downarrow$ )	NDB/ $K$ ( $\downarrow$ )	MOS( $\uparrow$ )
WaveNet-128	3.279	2.54	7.6	1.368	0.86	$1.34 \pm 0.29$
WaveNet-256	2.947	2.84	10.0	1.260	0.86	$1.43 \pm 0.30$
WaveGAN	1.349	4.53	36.6	0.796	0.78	$2.03 \pm 0.33$
DiffWave	<b>1.287</b>	<b>5.30</b>	<b>59.4</b>	<b>0.636</b>	<b>0.74</b>	<b><math>3.39 \pm 0.32</math></b>
Trainset	0.000	8.48	281.4	0.164	0.00	—
Testset	0.011	8.47	275.2	0.166	0.10	$3.72 \pm 0.28$

- AM (Activation Maximization) Score:
  - "takes into consideration the marginal label distribution of training data compared to IS"
- NDB (Number of Statistically-Different Bins):
  - "measures diversity of generated samples"
  - (K = number of K-means clusters involved in measuring the diversity of samples)




# DiffWave Interpolation

- Interpolate between two samples (of the same class) at  $t=50$  (i.e. with 50 more denoising steps left).

## Example 1

- Sample 1 
- Sample 2 
- Interpolation (50/50) 

## Example 2

- Sample 1 
- Sample 2 
- Interpolation (50/50) 

# Conclusion

- Diffusion models present versatile and lightweight neural vocoders.
- Having a large number of denoising steps can take a lot of time.
  - Solution: 'DiffWave(Fast)' parameterisation of the diffusion model that obtains good samples after just  $T=6$  denoising steps.
- Similar work has since been done on music synthesis, e.g. Google's "Multi-Instrument Music Synthesis With Spectrogram Diffusion" [9].
  - Conditioning on spectrograms and MIDI (using musical notation).

# References

- [1] Zen, Heiga, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemysław Szczepaniak. 'Fast, Compact, and High Quality LSTM-RNN Based Statistical Parametric Speech Synthesizers for Mobile Devices'. arXiv, 22 June 2016. <http://arxiv.org/abs/1606.06061>.
- [2] Oord, Aaron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 'WaveNet: A Generative Model for Raw Audio'. arXiv, 19 September 2016. <http://arxiv.org/abs/1609.03499>.
- [3] Kong, Zhifeng, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 'DiffWave: A Versatile Diffusion Model for Audio Synthesis'. arXiv, 30 March 2021. <http://arxiv.org/abs/2009.09761>.
- [4] Donahue, Chris, Julian McAuley, and Miller Puckette. 'Adversarial Audio Synthesis'. arXiv, 8 February 2019. <http://arxiv.org/abs/1802.04208>.
- [5] Chen, Nanxin, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. 'WaveGrad: Estimating Gradients for Waveform Generation'. arXiv, 9 October 2020. <http://arxiv.org/abs/2009.00713>.
- [6] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 'Denoising Diffusion Probabilistic Models'. arXiv, 16 December 2020. <http://arxiv.org/abs/2006.11239>.
- [7] Jia, Ziyue, Linfeng Yang, Zhenrong Zhang, Hui Liu, and Fannie Kong. 'Sequence to Point Learning Based on Bidirectional Dilated Residual Network for Non-Intrusive Load Monitoring'. *International Journal of Electrical Power & Energy Systems* 129 (1 July 2021): 106837. <https://doi.org/10.1016/j.ijepes.2021.106837>.
- [8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500, 2017.
- [9] Hawthorne, Curtis, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. 'Multi-Instrument Music Synthesis with Spectrogram Diffusion'. arXiv, 12 December 2022. <http://arxiv.org/abs/2206.05408>.