Towards generalizing deep-audio fake detection networks

Anonymous authors
Paper under double-blind review

Abstract

Today's generative neural networks allow the creation of high-quality synthetic speech at scale. While we welcome the creative use of this new technology, we must also recognize the risks. As synthetic speech is abused for monetary and identity theft, we require a broad set of deepfake identification tools. Furthermore, previous work reported a limited ability of deep classifiers to generalize to unseen audio generators. We study the frequency domain fingerprints of current audio generators. Building on top of the discovered frequency footprints, we train excellent lightweight detectors that generalize. We report improved results on the WaveFake dataset and an extended version. To account for the rapid progress in the field, we extend the WaveFake dataset by additionally considering samples drawn from the novel Avocodo and BigVGAN networks.

1 Introduction

The advancement of generative machine learning enables digital creativity, for example, in the form of more immersive video games and movies. However, it also creates new digital ways to lie. The technology is abused for theft (Karimi, 2023; Khatsenkova, 2023) and disinformation (Satariano & Mozur, 2023). Scammers use audio fakes to apply for remote jobs illicitly (PC-Mag, 2023). Via the telephone, cloned voices are misused in attempts to trick unsuspecting family members and stage fake kidnappings (Karimi, 2023). Problems have also surfaced on large platforms. Recently, artificially generated songs with voices from two well-known artists illicitly appeared on a large music streaming service (Guardian, 2023). The fake recordings had been created and published without the artist's consent. Consequently, we must meet the advancements in generative machine learning with deep fake detection tools that generalize well.

Our study reveals stable frequency domain artifacts for many modern speech synthesis networks. We visualize generator artifacts for all generators in the WaveFake dataset (Frank & Schönherr, 2021) generators and the Avocodo (Bak et al., 2022) and BigVGAN (Lee et al., 2023a) networks.

We challenge the commonly held belief that deep networks do not generalize well to unknown generators in the audio domain (Frank & Schönherr, 2021). Our results show that deep networks indeed generalize well. Using our dilated convolution-based model, we observe generalization to unseen generators for networks trained on Wavelet Packet Transform (WPT) and Short-Time Fourier Transform (STFT) inputs. We reproduce and improve upon synthetic media-recognition results published for the WaveFake dataset (Frank & Schönherr, 2021).

To ensure our detectors identify the newest generators, we extend the dataset proposed by Frank & Schönherr (2021) by adding two recent text-to-speech synthesis networks. We include the standard and large BigVGAN (Lee et al., 2023a) architecture as well as the Avocodo (Bak et al., 2022) network. Finally, we employ integrated gradients (Sundararajan et al., 2017) to systematically explore the behavior of our models.

To preserve the anonymity of the review process, we will share our source code and the dataset extension once the review process is complete.

2 Related work

2.1 Generative Models

The MelGAN architecture (Kumar et al., 2019) was an early Generative Adversarial Network (GAN) in the audio domain. It proposed to work with mel-scaled spectrograms as an intermediate representation. The evolution of generative models with intermediate mel-representations continues with the HiFiGAN architecture (Kong et al., 2020). Its generator contains multiple residual blocks and its training procedure minimizes the L1 distance between ground truth and generated mel-spectrograms. Parallel WaveGAN (Yamamoto et al., 2020) integrates the WaveNet (Oord et al., 2016) architecture and uses the STFT intermediate representation. Similarly, WaveGlow (Prenger et al., 2019), combines a WaveNet backbone with a flow-based Glow paradigm (Kingma & Dhariwal, 2018). The aforementioned architectures are part of the WaveFake dataset (Frank & Schönherr, 2021), which we will study in detail.

Furthermore, novel Text to Speech (TTS) systems have appeared since the publication of the WaveFake dataset. Lee et al. (2023a), for example, trained the biggest vocoder to date. Additionally, their architecture shifts to periodic activation functions. The authors report excellent generalization properties. Further, the parallelly developed Avocodo network (Bak et al., 2022) aims to reduce artifacts by removing low-frequency bias. We additionally include both architectures in our study.

2.2 Audiofake detection

The ability of generative machine learning to generate credible media samples led to an investigation into their automatic detection. Wang et al. (2020a), for example, devised generative content detectors for images and found that Convolutional Neural Network (CNN)-detectors initialized on ImageNet do allow the detection of many other CNN-based image generators, even if trained on only a single image-generator. While the deep learning community has focused its attention on the image domain (Wang et al., 2020a; Wolter et al., 2022; Huang et al., 2022; Dong et al., 2022; Li et al., 2022; Frank et al., 2020; Frank & Schönherr, 2021; Schwarz et al., 2021; Dzanic et al., 2020), audio-generation has largely been neglected so far (Frank & Schönherr, 2021).

In the audio domain, Frank & Schönherr (2021) established a baseline by introducing the WaveFake dataset. The dataset includes five different generative network architectures in nine sample sets. In addition to collecting the data, two baseline models are established. Some related work studies the ASVspoof 2019 (Lavrentyeva et al., 2019) and 2021 (Tomilov et al., 2021) challenges. Müller et al. (2022), for example, evaluate multiple randomly initialized architectures. Their experiments include a ResNet18, a transformer, and the Light Convolutional Neural Network (LCNN) architecture proposed by Lavrentyeva et al. (2017). The LCNN-architecture introduces max feature maps, which split the channel dimension in two. The resulting map contains the elementwise maximum of both halves. Jung et al. (2020) propose to detect synthetic media samples using a similar architecture that combines a CNN with a gated recurrent unit. The authors call their model RawNet2. RawNet2 works directly on raw waveforms. The model did not generalize well on the original WaveFake dataset (Frank & Schönherr, 2021).

Furthermore, Zhang et al. (2017) make use of dilated convolutional networks for environmental sound classification. While CNNs often use small filters, limiting contextual information, dilated filters appear to represent contextual information without resolution loss (Yu & Koltun, 2015).

2.3 Fingerprints of generative methods

Marra et al. (2019) propose to compute fingerprints of image generators by looking at noise residuals. Residuals are computed via $\mathbf{r}_i = \mathbf{x}_i - f(\mathbf{x}_i)$, where i loops over the inputs. In their equation, $\mathbf{x}$ denotes inputs, and $f(\cdot)$ is a suitable denoising filter. In other words, Marra et al. (2019) obtain the residual by subtracting the low-level image content $f(\mathbf{x}_i)$. Assuming residuals contain a fingerprint $\mathbf{f}$ and a random noise-component $\mathbf{n}$, that is $\mathbf{r}_i = \mathbf{f} + \mathbf{n}_i$. The fingerprint is estimated via $\hat{\mathbf{f}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{r}_i$. Wang et al. (2020a) continued along this line of work and computed additional image generator fingerprints by averaging high-pass filtered Fourier spectra. By averaging enough samples from a generator, variable parts of each signal are

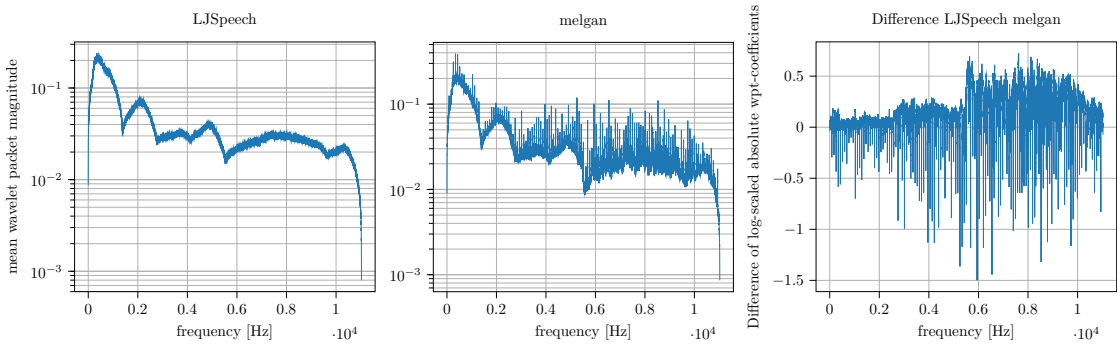

Figure 1: Mean level 14 Haar-Wavelet decomposition of original LJSpeech (left) recordings as well as synthetic versions generated by MelGAN (center). The difference between both plots is shown on the right. MelGAN (Kumar et al., 2019) displays a characteristic spike-shaped fingerprint. MelGAN produces characteristic spikes in the frequency domain.

lost, and the stable fingerprint remains. In the audio domain, Frank & Schönherr (2021) previously followed a similar logic and computed average frequency energies based on Fourier features.

2.4 Frequency analysis in audio processing

Frequency representations have a rich history in the field of audio processing. These representations are biologically motivated since the cochlea inside the human inner ear acts as a spectrum analyzer (Huang et al., 2001). Most of the literature chooses to work with the Short-Time Fourier Transform (STFT) (Tomilov et al., 2021; Palanisamy et al., 2020), or the Discrete Cosine Transform (DCT) (Sahidullah et al., 2015; Frank & Schönherr, 2021). After mapping the data to the frequency domain, the dimensionality is often reduced via a set of filter banks (Sahidullah et al., 2015). These can be linearly- or mel-spaced. Mel-spacing produces a high resolution in lower frequency ranges, where humans hear exceptionally well. Like mel-scaled STFT features, the Constant Q Transform (CQT) (Todisco et al., 2016) is perceptually motivated. The approach delivers a higher frequency resolution for lower frequencies and a higher temporal resolution for higher frequencies (Todisco et al., 2016). The process is similar to the fast wavelet transform. Wavelets have a long track record in engineering and signal processing. More recently, wavelet methods have started to appear in the artificial neural networks literature in the form of scatter-nets (Mallat, 2012; Cotter, 2020) and synthetic image detection (Wolter et al., 2022; Huang et al., 2022; Li et al., 2022). In the audio domain (Fathan et al., 2022) works with Mel-spectrogram inputs and feed features from a standard wavelet tree in parallel to a traditional CNN at multiple scales.

3 Experiments

To better understand how synthetic media differs from actual recordings, this section looks for artifacts that neural speech generators leave behind. We follow prior work and study Fourier-based representations. Additionally, we employ the Wavelet Packet Transform (WPT).

This section recreates the experimental setup described by Frank & Schönherr (2021). We do so to allow easy comparisons. Frank & Schönherr (2021) works with the LJSpeech and JSUT data sets. Various generators are tasked to recreate LJSpeech or JSUT sentences, which allows direct comparison.

We find artifacts across the entire spectrum. Consequently, we do not apply the high-pass filter step proposed by Marra et al. (2019) for images. Instead, we identify stable patterns all recordings have in common by averaging wavelet packet coefficients for 2500 single-second recordings per audio source. Artifact plots in this paper show log-scaled mean absolute wavelet-packet or Fourier coefficients up to 11.025 kHz. All spectra are averages over 2.500 recordings.

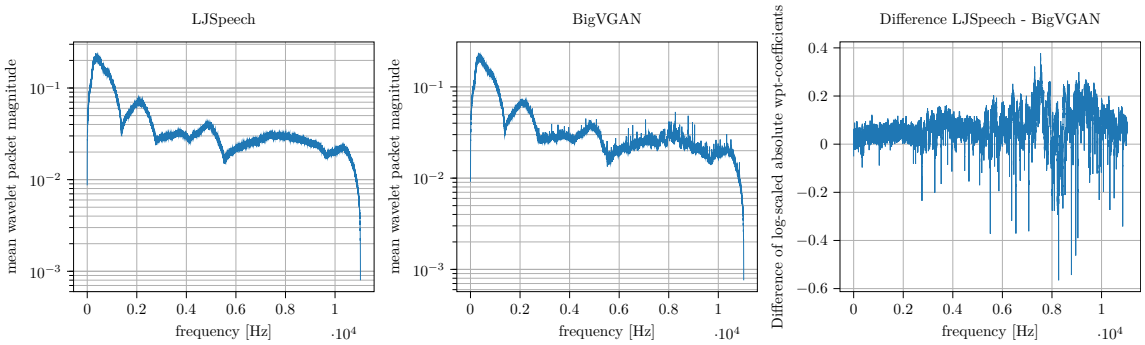

Figure 2: Mean level 14 Haar-Wavelet decomposition of original LJSpeech (left) recordings as well as synthetic versions generated by BigVGAN (center). The plot on the right shows the difference. The more recent BigVGAN produces spikes in the spectral representation. Albeit less pronounced than those of MelGAN, BigVGAN's spectral representation diverges from the original LJSpeech coefficients, especially for higher frequencies.

Figure 1 studies stable frequency domain patterns created by the MelGAN architecture. The figure shows mean absolute level 14 Haar-Wavelet packet transform coefficients for LJSpeech and MelGAN (Kumar et al., 2019) audio files. The transform reveals that MelGAN produces a spike-shaped pattern in the frequency domain. Supplementary Figure 16 confirms the spike pattern using a Fourier transform-based approach.

We repeat the same analysis for the recent BigVGAN-architecture (Lee et al., 2023a). Figure 2 presents our results. Small spikes are visible in the center figure. The pattern resembels what we saw for MelGAN.

Supplementary section 6 includes Wavelet packet and Fourier Artifact visualizations for all generative networks in this study. Furthermore, we create an audible version of the mean spectra, by transforming it back into the time domain. The supplementary material contains sound files with amplified generator artifacts. The files are exciting but not aesthetically pleasing. We recommend listening at low volumes.

3.1 Training general detectors for synthetic audio

If similar artifacts appear for all generators, we should be able to design detectors for synthetic media from unknown sources. Our detectors must be able to detect synthetic audio from generators, which had not been part of the training set. We have reason to believe this is possible for images (Wang et al., 2020b). Yet for audio Frank & Schönherr (2021) reported that standard deep detectors had low recognition rates for samples from unknown generators. To solve this problem, we devise a new Dilated Convolutional Neural Network (DCNN) architecture.

We ran our experiments on a four compute-node cluster with two AMD EPYC 7402 2.8 GHz host CPUs and four NVidia A100 Tensor Core graphics cards per host with 40 GB memory each. All experiments require four GPUs. Our experimental work builds upon Pytorch (Paszke et al., 2019) and the Pytorch-Wavelet-Toolbox (Moritz Wolter, 2021).

This experimental section studies the generalizing properties of deep fake-audio classifiers trained only on a single generator. We consider the WaveFake dataset (Frank & Schönherr, 2021) with all its generators, including the conformer-TTS samples and the Japanese language examples from the JSUT dataset.

We extend the WaveFake dataset by adding samples drawn from the Avocodo (Bak et al., 2022) and BigVGAN (Lee et al., 2023a) architectures. Due to a lack of pre-trained weights, we retrained Avocodo using the publicly available implementation from (Bak et al., 2023) commit `2999557`. We trained for 346 epochs or 563528 steps. Hyperparameters were chosen according to Bak et al. (2022) with a learning rate of 0.0002 for the discriminator and generator. After training, we used their inference script to generate additional LJSpeech samples. Following a similar procedure, we add BigVGAN (Lee et al., 2023a) generated audio samples. We employ a BigVGAN Large (L) with 112 M parameters, and it's downsized counterpart with

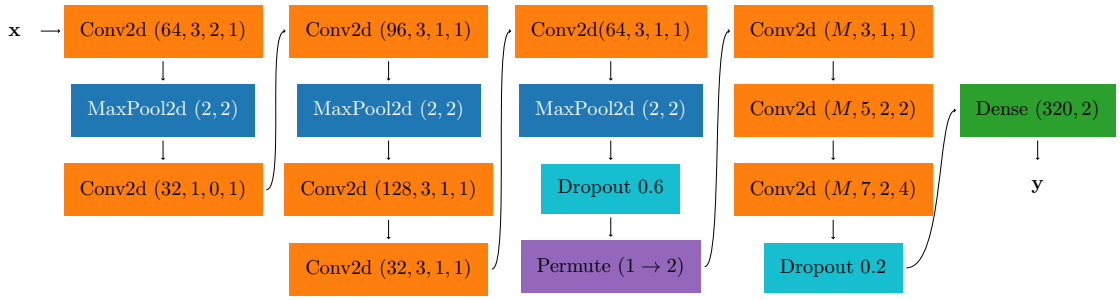

Figure 3: Structure of our Dilated Convolutional Neural Network (DCNN). The Conv2d blocks denote 2D-Convolution operations with hyperparameters (Output Channels, Kernel Size, Padding, Dilation). We always work with unit strides. Each Conv2d is preceded by a Batch Normalization Layer (Ioffe & Szegedy, 2015) and followed by a PreLU activation (Xu et al., 2015). The permutation operation permutes the first with the second dimension of the input (we consider the batch dimension to be dimension zero). M denotes the number of output channels from the convolutional layers before.

14 M weights, which we refer to as BigVGAN. We use the code from Lee et al. (2023b). Again, additional fake LJSpeech samples are generated with the authors' inference script for both models.

The Japanese language (JSUT) samples from Frank & Schönherr (2021) are downsampled from 24 kHz to 22.05 kHz to ensure a uniform sampling rate for all recordings. All samples are cut into one-second segments. All sets contain an equal amount of real and fake samples.

3.1.1 Generalizing detectors for the extended WaveFake Dataset

Frank & Schönherr (2021) include MelGAN, Parallel WaveGAN, Multi-bad MelGAN, full-band MelGAN, HiFi-GAN, and WaveGlow in their Wavefake data set. We add samples from Avocodo and BigVGAN by re-synthesizing all utterances in the LJSpeech corpus, we obtain a total of 13 100 additional audio files for Avocodo, 13 100 extra files for BigVGAN, and another 13 100 for BigVGAN Large.

Frequency information was a crucial ingredient in previous work on images (Frank et al., 2020; Wolter et al., 2022). Consequently, we explore wavelet packets and the STFT as input representations. Both capture frequency information. Our input pipeline proceeds as follows: Audio is loaded first. After batching, we transform with either the WPT or STFT. Once the transformation is complete, we always take the absolute values of the result. Generally, we compute the square and add extra labels if we don't. Finally, the result is re-scaled using the natural logarithm (ln). The WPT depends on the underlying wavelet. We study the effect of wavelets from the Daubechies, Symlet, and Coiflet families.

According to Frank & Schönherr (2021), a Gaussian Mixture Model (GMM) trained on top of Linear Frequency Cepstral Coefficient (LFCC) features performed best on the original WaveFake dataset. The GMM outperformed the deep RawNet2 proposed by Jung et al. (2020). RawNet2 processes raw and unmodified waveforms. A convolution-based encoder computes feature vectors. After the encoder, RawNet2 employs a recurrent layer to integrate information over time. Instead of relying on recurrent connections, we process contextual information via dilated convolution. Dilated convolutions enlarge the receptive field without downsampling (Yu & Koltun, 2015). We found dilated convolutions delivered improved run-time and fake detection accuracy. Furthermore, our network employs the PReLU Xu et al. (2015) activation function, which performs well in tandem with dilated convolution Zhang et al. (2017). Figure 3 depicts our architecture.

Adam (Kingma & Ba, 2015) optimizes all networks with a learning rate of 0.0004. Each training step used 128 audio samples per batch. Finally, we employ weight decay and dropout. The L2 penalty is set to 0.001 unless stated otherwise. We retrain the LFCC-GMM using the same training setup as Frank & Schönherr (2021). LCNN training is based on our code. We have to ensure generalization to unknown generators. Consequently, all detectors see *only* full-band MelGAN and real samples during training. Frank & Schönherr (2021) trained their best performing detectors exclusively on Full-band MelGAN, which motivates

Table 1: Fake detection results on the extended WaveFake dataset including JSUT. All input transforms work with 256-frequency bins. Our models are trained exclusively on samples drawn from a full-band MelGAN. We report the test set accuracy and aEER. To add additional context, we report mean test set accuracy and aEER and standard deviation over five runs for all experiments using seeds 0 to 4.

		Accuracy [%]		aEER	
Network	Input	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
DCNN (ours)	STFT	96.46	91.72 ± 2.94	0.036	0.159 ± 0.150
	db4	95.31	90.84 ± 5.59	0.079	0.133 ± 0.048
	db5	96.88	94.65 ± 1.85	0.048	0.082 ± 0.042
	db8	98.23	89.11 ± 8.80	0.097	0.132 ± 0.044
	sym4	97.73	94.31 ± 3.05	0.059	0.203 ± 0.125
	sym5	97.70	95.25 ± 3.09	0.031	0.069 ± 0.036
	sym8	98.49	96.77 ± 2.40	0.062	0.145 ± 0.078
	coif4	97.90	91.32 ± 5.41	**0.025**	0.084 ± 0.047
	coif8	98.72	97.39 ± 1.80	0.026	0.079 ± 0.047
LCNN	STFT	91.65	79.21 ± 16.55	0.083	0.169 ± 0.101
	sym5	97.46	90.12 ± 6.44	0.067	0.108 ± 0.042
GMM	LFCC (Frank & Schönherr)	–	–	0.145	–

our choice. Test accuracies and average Equal Error Rates (aEERs) are computed for test samples from all eight generators and the original audio, where we measure our detector's ability to separate real and fake.

Table 1 lists our results for the extended WaveFake dataset. We find that deep neural networks do generalize well to unseen generators. Both the LCNN and DCNN outperform the GMM proposed by Frank & Schönherr (2021). Furthermore, we observe improvements for our DCNN compared to the LCNN. In addition to the accuracy improvements, we reduce the number of optimizable parameters. For STFT and sym5-WPT inputs the DCNN produces better results on average and when considering only the best run. An LCNN has 3,312,450 parameters. Our DCNN uses only 239,015 for a sym5 input. Which leads to an additional efficiency improvement. A complete list of model parameter numbers is available in supplementary Table 8. Regarding the choice of the wavelet, we observe the average performance for the sym5 wavelet, while the coif4 input WPT leads to the best-performing network. Supplementary Table 6 lists results for all wavelets we tested.

3.1.2 The original WaveFake dataset

We continue to study the original WaveFake (Frank & Schönherr, 2021) dataset in isolation. Table 2 enumerates the performance of our classifiers on this unmodified dataset. We find a different picture in comparison to Table 1. Excluding the newer BigVGAN and Avocodo networks shifts the observed performance in favor of networks trained on STFT inputs. Please consider Table 3 to understand the root of the differences we previously observed in section 3.1.2. Table 3 reveals the performance of our classifiers when tasked to identify our most recent vocoders in isolation. Our STFT based networks struggle with identifying BigVGAN samples. The WPT based networks do much better on this task, which explains the differences between Tables 1 and 2. Further, in comparison to our DCNN the GMM from Frank & Schönherr (2021) drops in performance when evaluated on the newer generators. In summary, the DCNN improves upon the numbers presented by Frank & Schönherr (2021) for both coif4 and STFT inputs.

4 Ablation Study

This section ablates network parts and questions our design choices. We start with the network architecture. We either leave out all max pooling layers in exchange for a stride two downsampling in the layer before or leave out the Dropout layer. Table 4 lists the accuracy and aEER values we measured. The proposed DCNN

Table 2: Results on the unmodified WaveFake dataset (Frank & Schönherr, 2021). We cite baseline numbers as reported in the original paper. All frequency representations work with 256 bins. Our network outperforms the GMM-LFCC and their RawNet2 implementation for the db4, db5, sym4, sym5, coif4 as well as the coif8 wavelet, and the STFT.

		Accuracy [%]		aEER	
Network	Input	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
DCNN (ours)	STFT	99.88	97.98 ± 3.18	**0.001**	0.099 ± 0.178
	db4	94.71	90.20 ± 5.90	0.079	0.136 ± 0.050
	db5	96.36	94.39 ± 1.45	0.048	0.083 ± 0.041
	db8	98.21	89.17 ± 8.83	0.100	0.133 ± 0.046
	sym4	97.24	94.09 ± 2.26	0.057	0.200 ± 0.127
	sym5	97.60	95.57 ± 2.58	0.032	0.066 ± 0.035
	sym8	98.80	96.92 ± 1.69	0.062	0.142 ± 0.081
	coif4	98.24	92.88 ± 5.05	0.025	0.118 ± 0.088
	coif8	98.81	97.87 ± 0.92	0.026	0.121 ± 0.089
GMM	LFCC (Frank & Schönherr)	–	–	0.062	–
RawNet2	raw (Frank & Schönherr)	–	–	0.363	–

Table 3: Results on BigVGAN (Lee et al., 2023a) and Avocodo (Bak et al., 2022). When looking at the two newer generators in isolation, we observe aEERs in line with previous results for our wavelets-based classifiers. The GMM-LFCC (Frank & Schönherr, 2021) and the networks with STFT-inputs drop in performance when evaluated with the unseen BigVGAN.

		Avocodo				BigVGAN (L)			
		Accuracy [%]		aEER		Accuracy [%]		aEER	
Network	Input	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
DCNN (ours)	STFT	99.99	91.40 ± 16.49	**0.001**	0.096 ± 0.179	84.67	65.12 ± 11.82	0.182	0.365 ± 0.095
	db4	98.46	94.11 ± 3.69	0.024	0.098 ± 0.056	93.78	89.68 ± 2.92	0.079	0.136 ± 0.047
	db5	98.60	96.13 ± 2.87	0.026	0.064 ± 0.048	97.47	93.31 ± 3.09	0.052	0.097 ± 0.047
	db8	99.53	96.92 ± 2.23	0.005	0.055 ± 0.041	94.88	87.77 ± 6.81	0.101	0.159 ± 0.061
	sym4	97.85	86.58 ± 11.73	0.033	0.184 ± 0.142	93.58	86.55 ± 5.57	0.094	0.226 ± 0.108
	sym5	99.34	97.48 ± 2.21	0.010	0.046 ± 0.039	96.28	92.66 ± 4.02	0.043	0.097 ± 0.051
	sym8	99.76	91.82 ± 6.43	0.003	0.130 ± 0.095	95.08	90.85 ± 3.69	0.064	0.166 ± 0.071
	coif4	99.72	95.24 ± 7.07	0.005	0.072 ± 0.101	95.94	87.23 ± 7.40	0.047	0.173 ± 0.093
	coif8	98.70	92.31 ± 6.58	0.025	0.122 ± 0.090	96.11	93.46 ± 3.32	0.058	0.128 ± 0.083
GMM	LFCC	–	–	0.316	–	–	–	0.432	–

Table 4: DCNN fake-detection results on the extended WaveFake dataset (with JSUT) comparing a modified version of our architecture depicted in figure 3 with small modifications to motivate the use of max pooling and dropout. All input transforms work with 256-frequency bins. Our models are trained exclusively on samples drawn from a full-band MelGAN. We report the test set accuracy and aEER. To add additional context, we report mean test set accuracy and aEER and standard deviation over five runs for all experiments using seeds 0 to 4.

		Accuracy [%]		aEER	
DCNN-variant	Input	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
	sym5	97.70	95.25 ± 3.09	0.031	0.069 ± 0.036
	coif4	97.90	91.32 ± 5.41	**0.025**	0.084 ± 0.047
without Max Pooling	sym5	97.39	95.59 ± 1.73	0.056	0.082 ± 0.029
without Dropout	sym5	97.75	95.74 ± 1.41	0.049	0.062 ± 0.014
without Dropout	coif4	97.60	93.25 ± 5.28	0.029	0.073 ± 0.044

Table 5: The effect of the sign on the extended WaveFake dataset. We additionally concatenate a tensor with the sign patterns onto each input and retrain our networks.

	Accuracy [%]		aEER	
Input	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
SG-db5	87.69	83.03 ± 3.14	0.122	0.156 ± 0.024
SG-db8	91.31	85.43 ± 5.96	0.086	0.134 ± 0.045
SG-sym5	92.76	74.86 ± 13.47	0.073	0.203 ± 0.096
SG-sym8	94.64	77.77 ± 15.19	0.054	0.180 ± 0.105
SG-coif8	91.18	84.17 ± 5.40	0.087	0.147 ± 0.041

architecture is robust. It works without max-pooling or dropout. However, the best individual networks employed both, which justifies their use in our experiments.

4.1 Ablating the effect of the sign

The sign pattern is important for the WPT. Assume a wavelet packet tensor $\mathbf{P}$, which is obtained by stacking the end nodes of the packet graph. Given the pattern of minus signs $\mathbf{N}$, we could invert the rescaling since $\exp(\ln(\mathrm{abs}(\mathbf{P}))) \odot \mathbf{N} = \mathbf{P}$, if $\odot$ denotes the element wise Hadamard-product. Conserving the signs avoids information loss. For the WPT, we additionally explore two-channel inputs, where we tag on the sign pattern of the WPT. We call this case signed (SG). We argue that a lossless input is essential from a theoretical point of view. We observe no experimental benefit in Table 5. Consequently, we do not consider the sign pattern elsewhere in this study.

4.2 Attribution with Integrated Gradients

Figure 4 illustrates attribution using integrated gradients as proposed by Sundararajan et al. (2017) on the sym8-WPT input-network evaluated on the extended WaveFake test-dataset. We distinguish between the regions of interest within the input, averaged across 2500 samples of real audio samples (left plot) and 2500 samples of fake audio samples (middle), in order to identify areas of significance for the artificial neural network. This distinction is also made by comparing the averaged attribution across fake and real audio samples (right plot). Similarly, we repeat the same experiment for the STFT version of the DCNN. Figure 5 plots the result.

We observe that, for both transformations, the high-frequency components play a significant role in the classification process. Both transformations exhibit an inverse character in terms of attribution between real and fake audio samples. In Figures 1 and 2, we saw differences across the spectrum, while higher frequencies

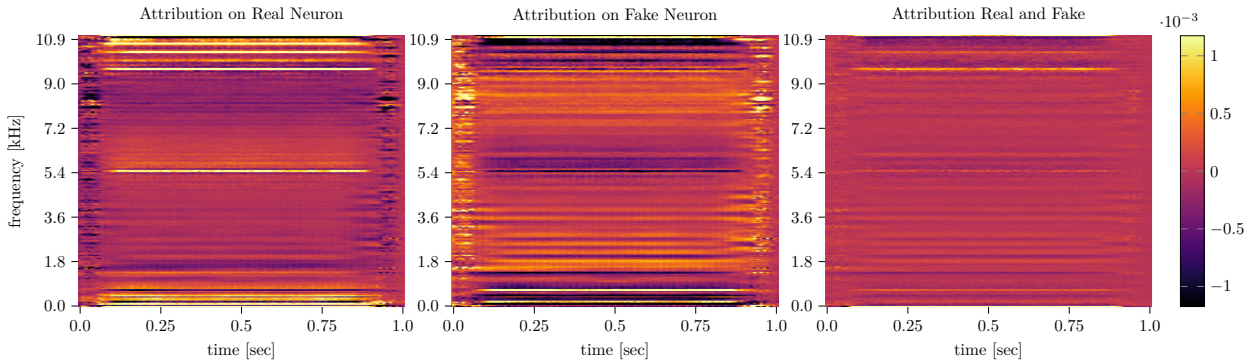

Figure 4: Attribution using integrated gradients on the-sym8-WPT-based DCNN-classifier evaluated over 2500 real audio samples from LJSpeech (left), 2500 fake audio samples from our extended version of the WaveFake dataset (middle) and averaged over both real and fake audios (right). We observe high values in the high-frequency domain as well as an inverse character between real and fake audio files. This WPT based CNN shows clearly distinguishable interest regions for the model in several frequency bins.

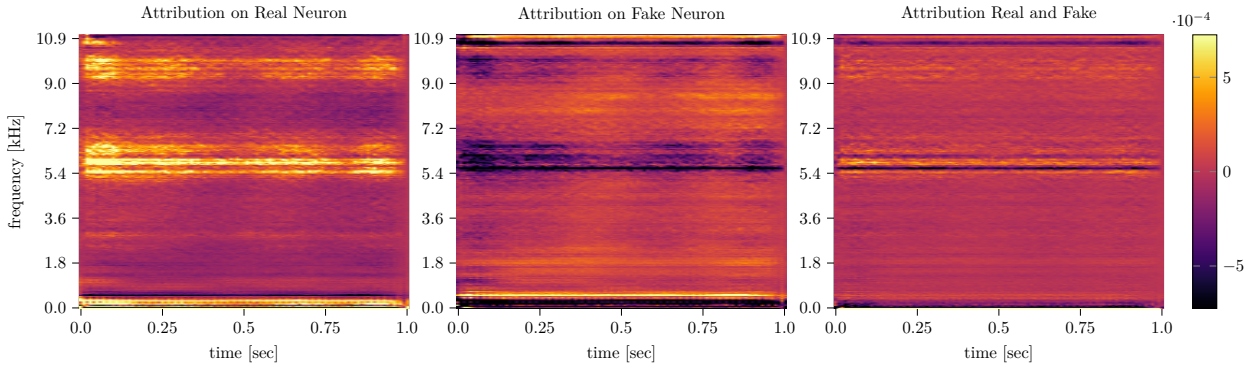

Figure 5: Attribution using integrated gradients (Sundararajan et al., 2017) on the STFT-DCNN-classifier for 2500 real audio samples from LJSpeech (left), 2500 fake audio samples from our extended version of the WaveFake dataset (middle) and averaged over both real and fake audios (right). We observe moderately high values in the high-frequency domain as well as an inverse character between real and fake audios similar to what we saw in Figure 4.

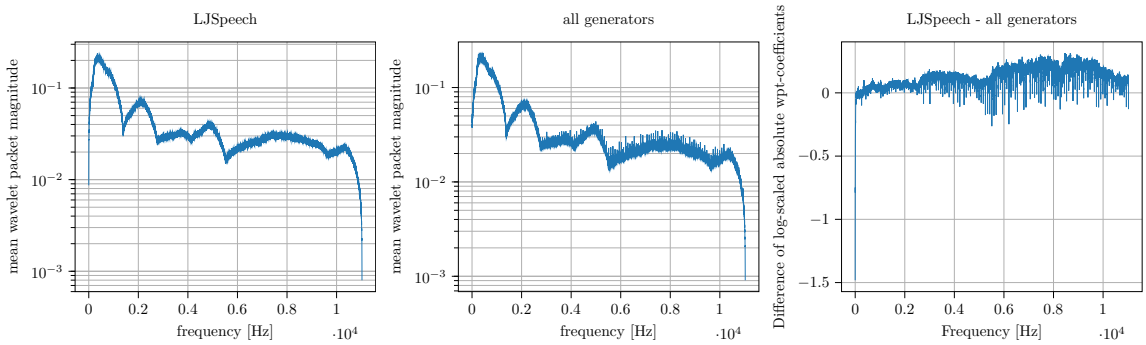

Figure 6: Mean level 14 Haar-Wavelet decomposition of original LJSpeech (left) recordings as well as synthetic versions generated by all generators in the extended WaveFake dataset Frank & Schönherr (2021) (center). The difference between both plots is shown on the right.

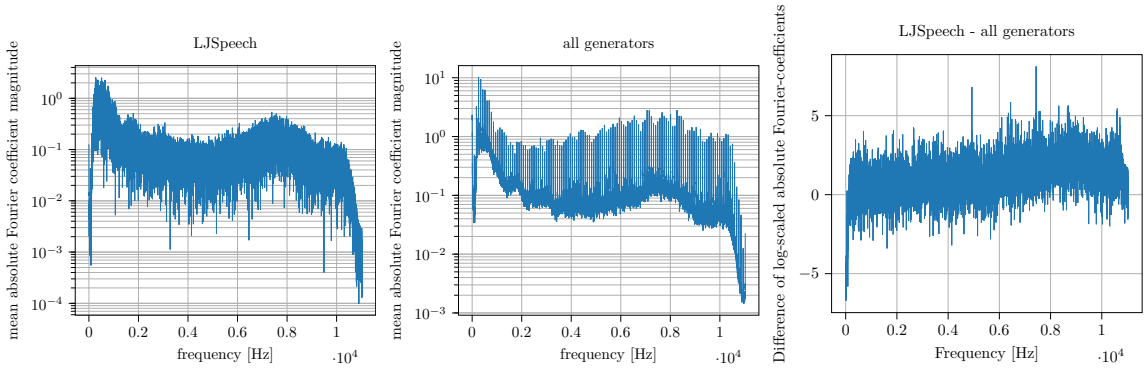

Figure 7: Mean absolute Fourier coefficient visualizations of original LJSpeech (left) recordings as well as synthetic versions generated by all generators in the extended WaveFake dataset Frank & Schönherr (2021) (center). The difference between both plots is shown on the right.

tended to differ more. Figures 4 and 5 reflect this pattern. Generally, all frequencies impact the result, but frequencies above 4kHz matter more. Figures 6 and 7 allow us to sanity check the integrated gradient plots. The integrated gradient indicated impoact across the spectrum, with higher influence at higher frequencies. The mean spectra differ more at higher frequencies, which validates our previous observation.

5 Conclusion

This paper investigated the WaveFake dataset and its extended version with two additional recent GAN models. Frank & Schönherr (2021) observed poor generalization for deep neural network-based fake detectors. In response, the authors proposed using traditional methods to mitigate overfitting to specific audio generators. We found stable patterns in the averaged spectra for all generators. Building upon this observation, DCNN-based classifier demonstrated robust generalization performance across various representations in the two-dimensional frequency domain. This observation remained consistent regardless of the chosen representation method. Moreover, our attribution analysis highlighted the significance of high-frequency components. It revealed distinct attribution patterns between real and fake audio samples, underscoring the interpretability and potential insights provided by the investigated transformations.

References

Taejun Bak, Junmo Lee, Hanbin Bae, Jinhyeok Yang, Jae-Sung Bae, and Young-Sun Joo. Avocodo: Generative adversarial network for artifact-free vocoder. *arXiv preprint arXiv:2206.13404, accepted for publication at the 37th AAAI conference on artificial intelligence (to appear)*, 2022.

Taejun Bak, Junmo Lee, Hanbin Bae, Jinhyeok Yang, Jae-Sung Bae, and Young-Sun Joo. Github-repository - avocodo: Generative adversarial network for artifact-free vocoder. `https://github.com/ncsoft/avocodo/commit/2999557bbd040a6f3eb6f7006a317d89537b78cd`, 2023. Accessed: 2023-05-17.

Fergal Cotter. *Uses of complex wavelets in deep convolutional neural networks*. PhD thesis, University of Cambridge, 2020.

Shichao Dong, Jin Wang, Jiajun Liang, Haoqiang Fan, and Renhe Ji. Explaining deepfake detection by analysing image matching. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV*, pp. 18–35. Springer, 2022.

Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33:3022–3032, 2020.

Abderrahim Fathan, Jahangir Alam, and Woohyun Kang. Multiresolution decomposition analysis via wavelet transforms for audio deepfake detection. In *Speech and Computer: 24th International Conference, SPECOM 2022, Gurugram, India, November 14–16, 2022, Proceedings*, pp. 188–200. Springer, 2022.

Joel Frank and Lea Schönherr. Wavefake: A data set to facilitate audio deepfake detection. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract-round2.html`.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pp. 3247–3258. PMLR, 2020.

Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

Guardian. Ai song featuring fake drake and weeknd vocals pulled from streaming services. `https://www.theguardian.com/music/2023/apr/18/ai-song-featuring-fake-drake-and-weeknd-vocals-pulled-from-streaming-services`, 2023. Accessed: 2023-05-11.

Wei Huang, Michelangelo Valsecchi, and Michael Multerer. Anisotropic multiresolution analyses for deep fake detection. *arXiv preprint arXiv:2210.14874*, 2022.

Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, Frankreich, 07–09 Juli*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Juli 2015. PMLR.

Arne Jensen and Anders la Cour-Harbo. *Ripples in mathematics: the discrete wavelet transform*. Springer Science & Business Media, 2001.

Jee-weon Jung, Seung-bin Kim, Hye-jin Shim, Ju-ho Kim, and Ha-Jin Yu. Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms. In Helen Meng, Bo Xu, and Thomas Fang Zheng (eds.), *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pp. 1496–1500. ISCA, 2020. doi: 10.21437/Interspeech.2020-1011. URL `https://doi.org/10.21437/Interspeech.2020-1011`.

Faith Karimi. mom, these bad men have me: She believes scammers cloned her daughters voice in a fake kidnapping. https://edition.cnn.com/2023/04/29/us/ai-scam-calls-kidnapping-cec/index.html, 2023. Accessed: 2023-05-11.

Sophia Khatsenkova. Audio deepfake scams: Criminals are using ai to sound like family and people are falling for it. https://www.euronews.com/next/2023/03/25/audio-deepfake-scams-criminals-are-using-ai-to-sound-like-family-and-people-are-falling-fo, 2023. Accessed: 2023-05-11.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32, 2019.

Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. Audio Replay Attack Detection with Deep Learning Frameworks. In *Proc. Interspeech 2017*, pp. 82–86, 2017. doi: 10.21437/Interspeech.2017-360.

Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov. STC antispoofing systems for the asvspoof2019 challenge. In Gernot Kubin and Zdravko Kacic (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 1033–1037. ISCA, 2019. doi: 10.21437/Interspeech.2019-1768. URL https://doi.org/10.21437/Interspeech.2019-1768.

Sang Gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sung-Hoon Yoon. Bigvgan: A universal neural vocoder with large-scale training. *International Conference on Learning Representations (ICLR) 2023*, 2023a.

Sang Gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. Github-repository - bigvgan: A universal neural vocoder with large-scale training. https://github.com/nvidia/bigvgan, 2023b. Accessed: 2023-05-17.

Jiaming Li, Hongtao Xie, Lingyun Yu, and Yongdong Zhang. Wavelet-enhanced weakly supervised local feature learning for face forgery detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1299–1308, 2022.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10): 1331–1398, 2012.

Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 506–511. IEEE, 2019.

Moritz Wolter. *Frequency Domain Methods in Recurrent Neural Networks for Sequential Data Processing.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, July 2021. URL https://hdl.handle.net/20.500.11811/9245.

Nicolas M. Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? In Hanseok Ko and John H. L. Hansen (eds.), *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22*

September 2022, pp. 2783–2787. ISCA, 2022. doi: 10.21437/Interspeech.2022-108. URL https://doi.org/10.21437/Interspeech.2022-108.

Aaron van den Oord, 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 preprint arXiv:1609.03499*, 2016.

Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking cnn models for audio classification. *arXiv preprint arXiv:2007.11154*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

PC-Mag. Fbi: Scammers are interviewing for remote jobs using deepfake tech. https://www.pcmag.com/news/fbi-scammers-are-interviewing-for-remote-jobs-using-deepfake-tech, 2023. Accessed: 2023-05-11.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621. IEEE, 2019.

Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A comparison of features for synthetic speech detection. 2015.

Adam Satariano and Paul Mozur. The people onscreen are fake. the disinformation is real. https://www.nytimes.com/2023/02/07/technology/artificial-intelligence-training-deepfake.html, 2023. Accessed: 2023-05-11.

Katja Schwarz, Yiyi Liao, and Andreas Geiger. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34:18126–18136, 2021.

Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017.

Massimiliano Todisco, Héctor Delgado, and Nicholas WD Evans. A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients. In *Odyssey*, volume 2016, pp. 283–290, 2016.

Anton Tomilov, Aleksei Svishchev, Marina Volkova, Artem Chirkovskiy, Alexander Kondratev, and Galina Lavrentyeva. STC Antispoofing Systems for the ASVspoof2021 Challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 61–67, 2021. doi: 10.21437/ASVSPOOF.2021-10.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 8692–8701. Computer Vision Foundation / IEEE, 2020a. doi: 10.1109/CVPR42600.2020.00872. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_CNN-Generated_Images_Are_Surprisingly_Easy_to_Spot..._for_Now_CVPR_2020_paper.html.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704, 2020b.

Moritz Wolter, Felix Blanke, Raoul Heese, and Jochen Garcke. Wavelet-packets for deepfake image analysis and detection. *Machine Learning*, Special Issue of the ECML PKDD 2022 Journal Track:1–33, August 2022. ISSN 0885-6125. doi: https://doi.org/10.1007/s10994-022-06225-5. URL https://rdcu.be/cUIRt.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015. doi: 10.48550/arXiv.1505.00853.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203. IEEE, 2020.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 492–498. IEEE, 2021.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122, published as a conference paper at ICLR 2016*, 2015. doi: 10.48550/arXiv.1511.07122.

Xiaohu Zhang, Yuexian Zou, and Wei Shi. Dilated convolution neural network with leakyrelu for environmental sound classification. In *2017 22nd international conference on digital signal processing (DSP)*, pp. 1–5. IEEE, 2017.

6 Supplementary

Acronyms

aEER average Equal Error Rate

CNN Convolutional Neural Network

CQT Constant Q Transform

DCNN Dilated Convolutional Neural Network

DCT Discrete Cosine Transform

FFT Fast Fourier Transform

FWT Fast Wavelet Transform

GAN Generative Adversarial Network

GMM Gaussian Mixture Model

LCNN Light Convolutional Neural Network

LFCC Linear Frequency Cepstral Coefficient

ln natural logarithm

STFT Short-Time Fourier Transform

TTS Text to Speech

WPT Wavelet Packet Transform

6.1 Additional Fourier-Artifact plots

Fourier Transform based energy difference plots previously also appeared in Frank & Schönherr (2021). We confirm their results and show Fourier-Transform-based plots again to allow easy comparison with the wavelet packet approach.

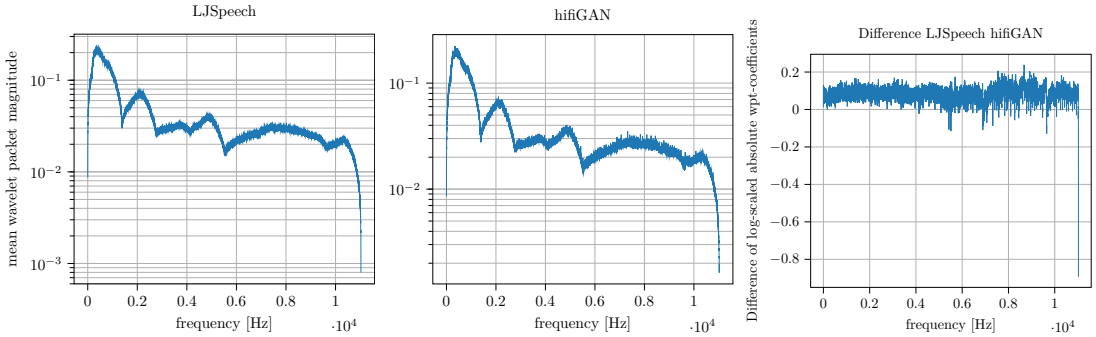

Figure 8: Mean level 14 Haar-Wavelet coefficients of LJSpeech (left) recordings as well as versions generated by the Hifi-GAN-architecture (Kong et al., 2020). We see some spikes for the Hifi-GAN-architecture (Kong et al., 2020) as well, but MelGAN produces more prominent spikes. The difference plot reveals that HifiGAN produces smaller coefficients on average than the true distribution.

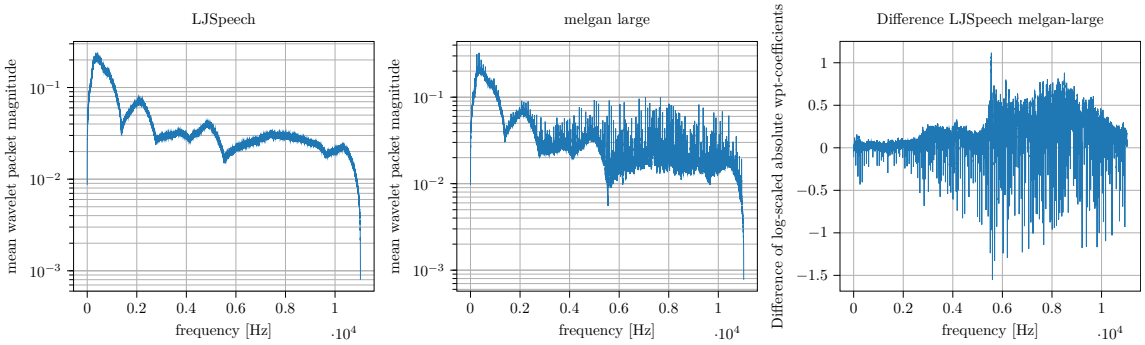

Figure 9: Mean absolute wavelet packet plots for the melgan-large and multi-band melgan generators alongside mean coefficients from the original data set. We see melgan-like spikes for the large melgan architecture.

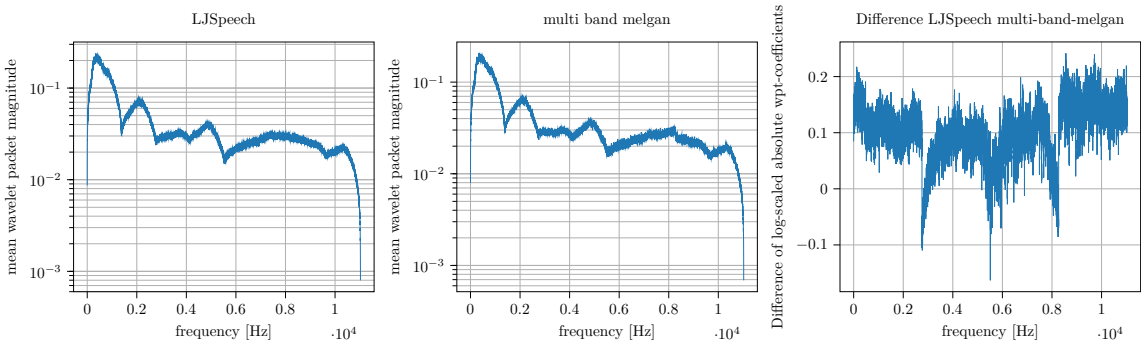

Figure 10: The multiband melgan's (Yang et al., 2021) mean absolute coefficients reveal a discontinuity between 8 and 10kHz.

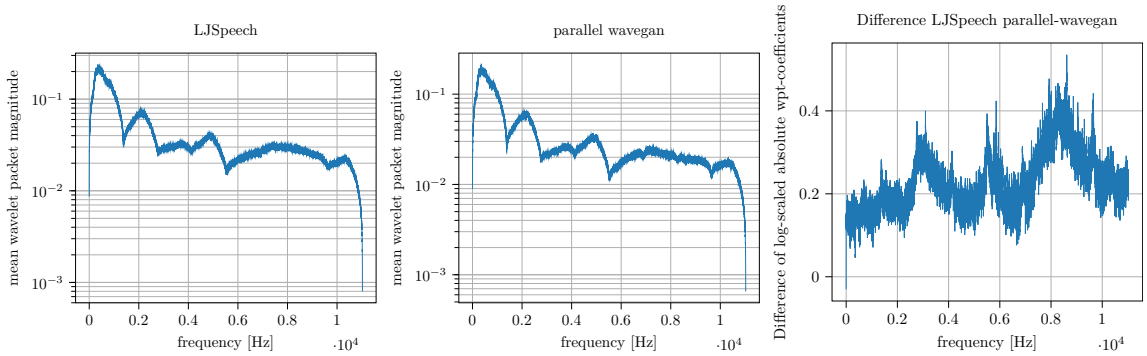

Figure 11: Parallel Wavegan (Yamamoto et al., 2020)

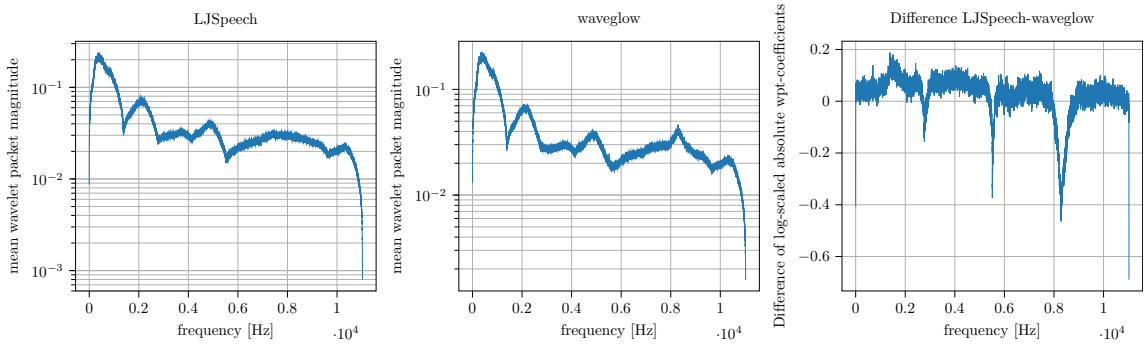

Figure 12: WaveGlow (Prenger et al., 2019)

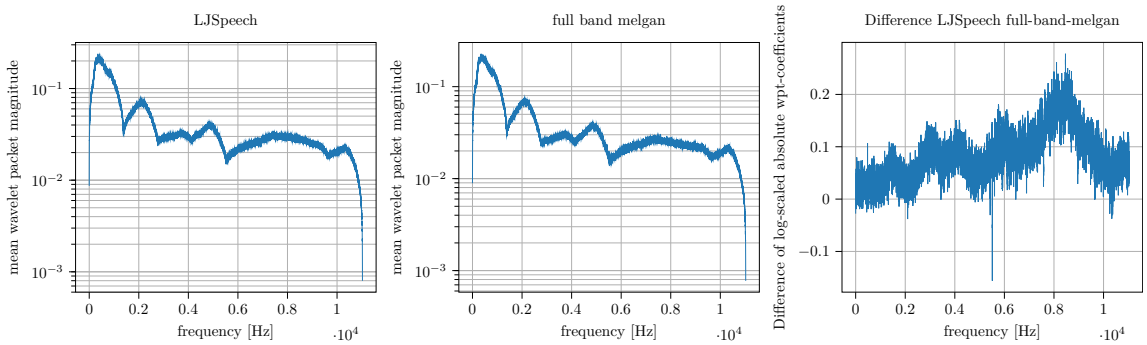

Figure 13: Full-band MelGAN plots.

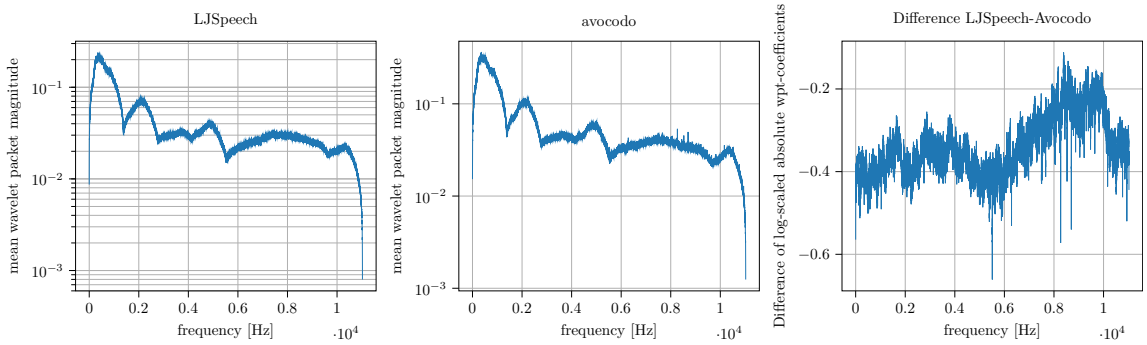

Figure 14: Avocodo plots.

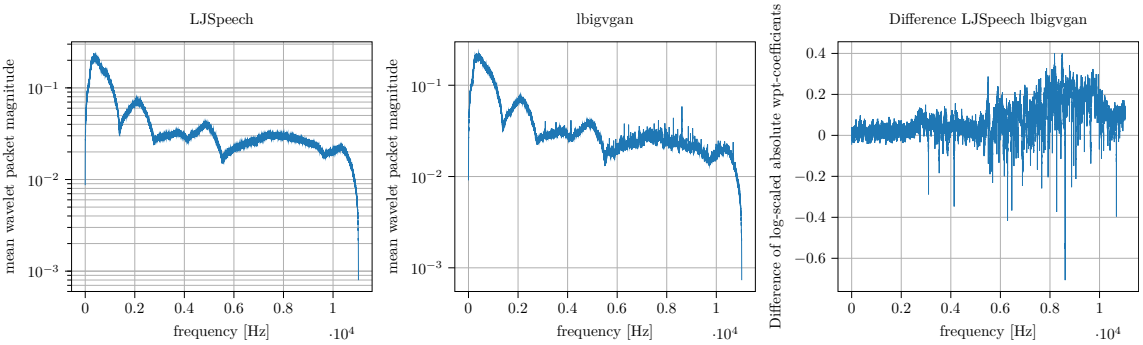

Figure 15: BigVGAN Large plots.

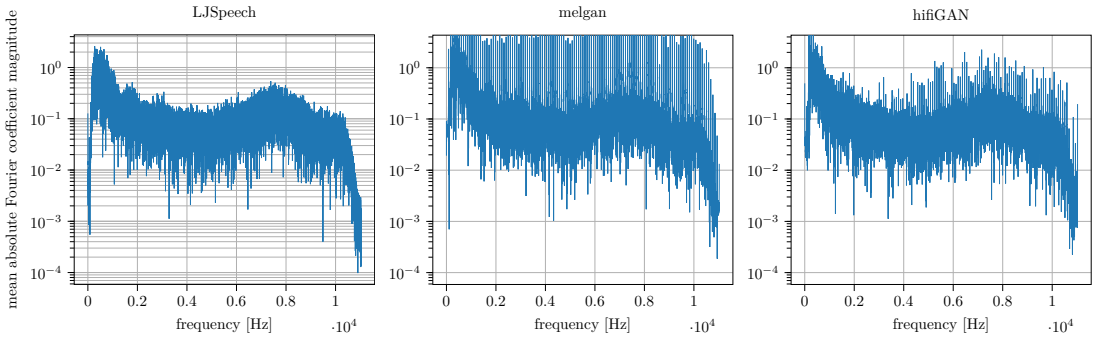

Figure 16: Mean absolute Fourier coefficient visualizations. Some vocoders leave distinct patterns in the frequency domain. Compared to the plot computed using the original LJSpeech recording in the top left, we see distinct spike patterns for the MelGAN and HiFiGAN vocoders. The real audio samples (left) come from the LJSpeech corpus. LJSpeech also served as a training dataset for both generators. Not all generators display patterns like these.

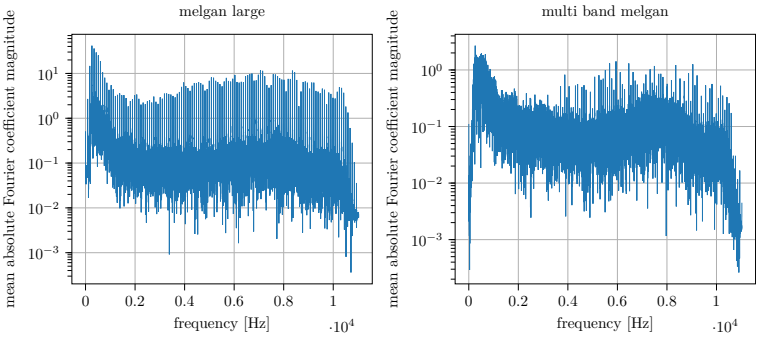

Figure 17: Fingerprint plots for MelGAN large and multi-band MelGAN generators. We observe clear spike-like patterns in the generated audio samples compared to the real LJSpeech samples.

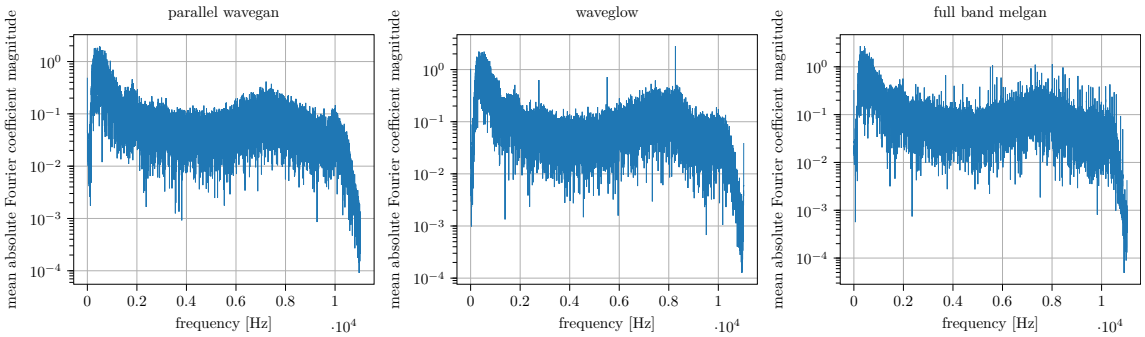

Figure 18: Fingerprint plots for the parallel WaveGAN, WaveGlow, and full-band MelGAN architectures. Especially for full-band MelGAN, we observe a clear spike-like pattern in the generated audio samples compared to the real LJSpeech samples.

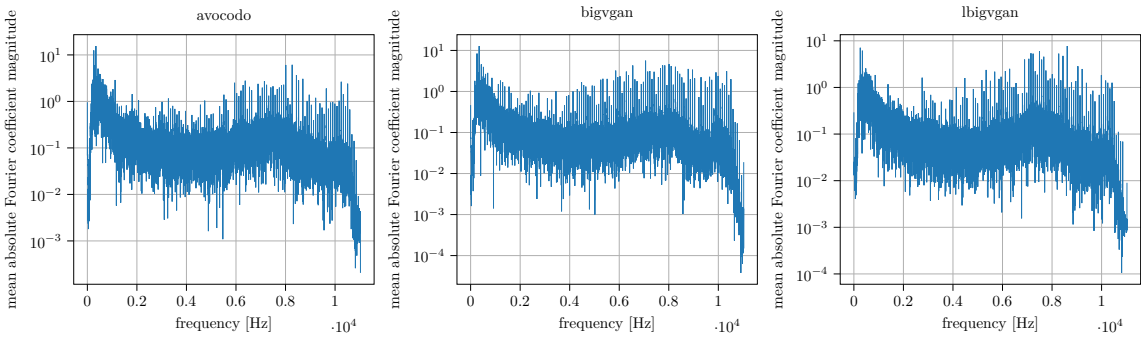

Figure 19: Fingerprint plots for the Avocodo, BigVGAN and Large BigVGAN architectures. Similarly to older generators, we observe spike-like patterns in the fingerprint plots of newer generators as well.

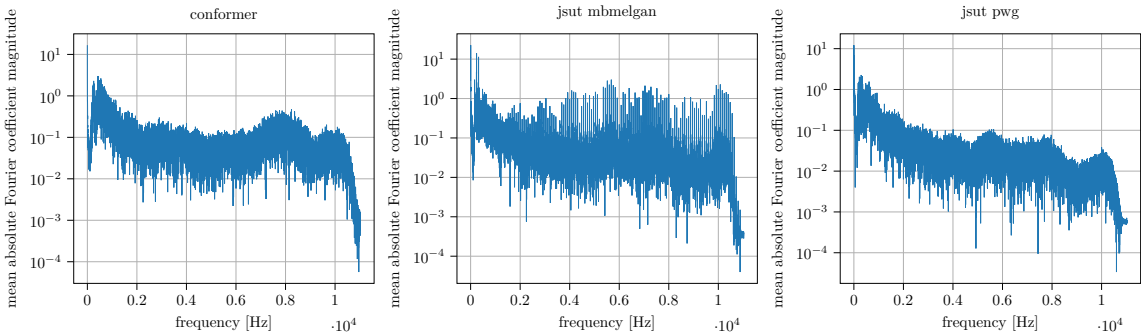

Figure 20: Fingerprint plots for the Conformer, JSUT multi-band MelGAN, and JSUT parallel WaveGAN architectures.

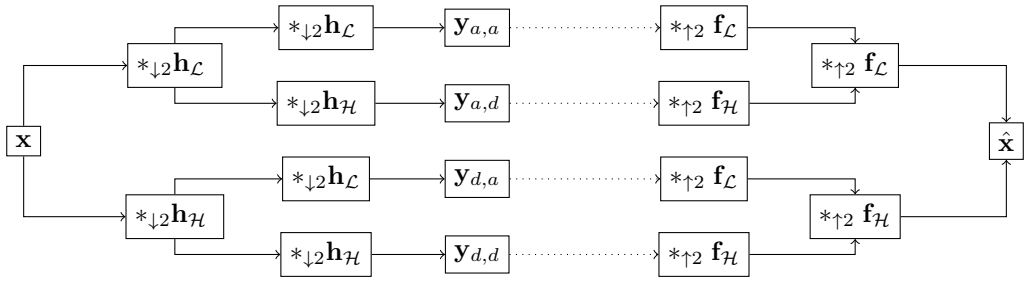

Figure 21: Visualisation of the wavelet packet transform. The figure uses $\mathbf{h}$ to denote analysis filters, $\mathbf{f}$ for synthesis filters, and stars $*$ for the convolution operation. The 0 subindex denotes lowpass- the 1 index denotes highpass-filters. $\downarrow 2$ denotes subsampling with a stride of two. $\uparrow 2$ denotes a transposed convolution with an upsampling effect and stride two. The notation follows Strang & Nguyen (1996). The $\mathbf{b}$s denote the packet coefficients. The operation is invertible. Reconstruction produces an input-reconstruction $\hat{\mathbf{x}}$.

6.2 The Fast Wavelet Transform (FWT)

The FWT works with two filter pairs. An analysis pair $\mathbf{h}_{\mathcal{L}}$, $\mathbf{h}_{\mathcal{H}}$ and a synthesis pair $\mathbf{f}_{\mathcal{L}}$, $\mathbf{f}_{\mathcal{H}}$. The analysis transform relies on convolutions with a stride of two. The first level of the transform computes

$$\mathbf{f}_{\mathcal{L}} *_{\downarrow 2} \mathbf{x} = \sum_k \mathbf{h}_{\mathcal{L}}[k]\mathbf{x}[n-k] = \mathbf{y}_a[n], \tag{1}$$

$$\mathbf{f}_{\mathcal{H}} *_{\downarrow 2} \mathbf{x} = \sum_k \mathbf{h}_{\mathcal{H}}[k]\mathbf{x}[n-k] = \mathbf{y}_d[n]. \tag{2}$$

Index k runs from 1 up to the filter length. Valid filter positions are indexed by n. $*_{\downarrow 2}$ symbolizes convolution with a stride of two. Padding is required to ensure every location is covered. Approximation coefficients have an a subindex. Detail coefficients use d. For the next level we compute $\mathbf{y}_{aa} = \mathbf{f}_{\mathcal{L}} *_{\downarrow 2} \mathbf{y}_a$ and $\mathbf{y}_{ad} = \mathbf{f}_{\mathcal{H}} *_{\downarrow 2} \mathbf{y}_a$. The process continues by recursively filtering the approximation coefficients. Introducing convolution matrices $\mathbf{H}$ allows writing the analysis transform in matrix form Strang & Nguyen (1996),

$$\mathbf{A} = \dots \left(\begin{array}{c|c} \begin{matrix}\mathbf{H}_{\mathcal{L}} \\ \mathbf{H}_{\mathcal{H}}\end{matrix} & \\ \hline & \mathbf{I} \end{array}\right) \begin{pmatrix}\mathbf{H}_{\mathcal{L}} \\ \mathbf{H}_{\mathcal{H}}\end{pmatrix}. \tag{3}$$

the matrix structure illustrates how the process decomposes multiple scales. $\mathbf{H}_{\mathcal{L}}$ and $\mathbf{H}_{\mathcal{H}}$ denotes high-pass and low-pass convolution matrices, respectively. Evaluating the matrix-product yields $\mathbf{A}$, the analysis matrix. In the factor matrices, the identity in the lower block grows with every scale. Therefore the process speeds up with every scale. The process is invertible. We could construct a synthesis matrix $\mathbf{S}$ using transposed convolution matrices to undo the transform. See Strang & Nguyen (1996) or Jensen & la Cour-Harbo (2001) for excellent in-depth discussions of the FWT.

6.3 Wavelet-packets

Related work (Frank et al., 2020; Wolter et al., 2022; Frank & Schönherr, 2021; Schwarz et al., 2021; Dzanic et al., 2020) highlights the importance of high-frequency bands for detecting gan-generated images. Consequently, expanding only the low-frequency part of the discrete wavelet-transform will not suffice. Instead, we employ the wavelet packet transform (Jensen & la Cour-Harbo, 2001). To ensure a fine-grained frequency resolution, we expand the complete tree. This approach is also known as the Walsh-form (Strang & Nguyen, 1996) of the transform. Figure 21 illustrates the idea. The main idea is to recursively expand the tree in the low- and high-pass direction. We index tree nodes with the filter sequences. a denotes a low-pass or approximation, d a high-pass or detail step. Via full expansion, we improve the resolution in the relevant high-frequency part of the tree compared to the standard FWT. The full tree linearly subdivides the spectrum from zero until the Nyquist frequency at half the sampling rate.

Not all filters are suitable choices for the analysis and synthesis pairs. Appropriate choices must obey the perfect reconstruction and anti-aliasing conditions Strang & Nguyen (1996). We found smaller Daubechies wavelets to be a good starting point. Symlets are a variant of the Daubechies-Wavelet family. Evaluating these symmetric cousins is usually the next step when exploring wavelet choices.

6.4 The Short-Time Fourier Transform (STFT)

By comparison, the STFT achieves localization in time via a sliding window. A sliding window $\mathbf{w}$ segments the input prior to Fast Fourier Transform (FFT). Only the part where the window is non-zero is visible. Or formally Griffin & Lim (1984)

$$\mathbf{X}[\omega, Sm] = \mathcal{F}(\mathbf{w}[Sm - l]\mathbf{x}[l]) = \sum_{l=-\infty}^{\infty} \mathbf{w}[Sm - l]\mathbf{x}[l]e^{-j\omega l}. \tag{4}$$

Here $\mathcal{F}$ denotes the classic discrete FFT. S denotes the sampling period. We select specific windows with m, while ω denotes different frequencies in each window. In other words, after multiplying slices of $\mathbf{x}$ with $\mathbf{w}$, the result is Fourier-transformed.

6.5 Network architecture

We list the total number of optimizable parameters in Table 8. We find a small increase in parameters in the signed scenario.

Table 6: Complete source identification results on the extended WaveFake Frank & Schönherr (2021) dataset (with JSUT).

		Accuracy [%]		aEER	
	Method	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
DCNN	STFT	96.46	91.72 ± 2.94	0.036	0.159 ± 0.150
	haar	62.27	-	0.343	-
	db2	93.06	74.38 ± 11.01	0.194	0.279 ± 0.059
	db3	93.07	84.46 ± 12.18	0.098	0.159 ± 0.072
	db4	95.31	90.84 ± 5.59	0.079	0.133 ± 0.048
	db5	96.88	94.65 ± 1.85	0.048	0.082 ± 0.042
	db6	97.01	89.58 ± 10.95	0.038	0.215 ± 0.146
	db7	97.52	92.78 ± 7.60	0.040	0.112 ± 0.081
	db8	98.23	89.11 ± 8.80	0.097	0.132 ± 0.044
	db9	98.19	94.72 ± 3.60	0.050	0.095 ± 0.034
	db10	98.08	90.56 ± 8.15	0.043	0.124 ± 0.049
	sym2	93.06	74.38 ± 11.01	0.194	0.279 ± 0.059
	sym3	93.07	84.46 ± 12.18	0.098	0.159 ± 0.072
	sym4	97.73	94.31 ± 3.05	0.059	0.203 ± 0.125
	sym5	97.70	95.25 ± 3.09	0.031	0.069 ± 0.036
	sym6	97.84	97.42 ± 0.39	0.048	0.172 ± 0.094
	sym7	97.47	92.27 ± 7.75	0.067	0.172 ± 0.089
	sym8	98.49	96.77 ± 2.40	0.062	0.145 ± 0.078
	sym9	98.67	95.85 ± 2.78	0.060	0.112 ± 0.064
	sym10	98.39	86.65 ± 18.35	0.028	0.148 ± 0.113
	coif2	96.43	87.86 ± 9.78	0.100	0.180 ± 0.089
	coif3	98.02	93.34 ± 4.95	0.042	0.085 ± 0.036
	coif4	97.90	91.32 ± 5.41	**0.025**	0.084 ± 0.047
	coif5	98.45	93.51 ± 3.44	0.057	0.221 ± 0.175
	coif6	98.72	89.92 ± 14.99	0.035	0.115 ± 0.095
	coif7	98.63	85.64 ± 13.14	0.045	0.128 ± 0.097
	coif8	98.72	97.39 ± 1.80	0.026	0.079 ± 0.047
	coif9	98.58	95.64 ± 3.92	0.051	0.106 ± 0.044
	coif10	98.38	95.55 ± 2.75	0.039	0.067 ± 0.025
LCNN	STFT	91.65	79.21 ± 16.55	0.083	0.169 ± 0.101
LCNN	sym5	97.46	90.12 ± 6.44	0.067	0.108 ± 0.042
GMM	LFCC (Frank & Schönherr)	–	–	0.145	–

Table 7: All our results on the original WaveFake dataset Frank & Schönherr (2021).

		WaveFake Frank & Schönherr (2021)			
		Accuracy [%]		aEER	
	Method	max	$\mu \pm \sigma$	min	$\mu \pm \sigma$
DCNN	STFT	99.88	97.98 ± 3.18	**0.001**	0.099 ± 0.178
	haar	68.64	-	0.301	-
	db2	93.39	80.59 ± 8.34	0.132	0.237 ± 0.076
	db3	93.35	85.96 ± 11.08	0.084	0.152 ± 0.070
	db4	94.71	90.20 ± 5.90	0.079	0.136 ± 0.050
	db5	96.36	94.39 ± 1.45	0.048	0.083 ± 0.041
	db6	96.15	88.96 ± 10.83	0.044	0.216 ± 0.146
	db7	97.81	92.65 ± 7.80	0.042	0.113 ± 0.083
	db8	98.21	89.17 ± 8.83	0.100	0.133 ± 0.046
	db9	97.70	94.51 ± 3.46	0.050	0.096 ± 0.034
	db10	97.81	90.79 ± 7.77	0.047	0.123 ± 0.048
	sym4	97.24	94.09 ± 2.26	0.057	0.200 ± 0.127
	sym5	97.60	95.57 ± 2.58	0.032	0.066 ± 0.035
	sym6	97.98	97.02 ± 0.81	0.047	0.172 ± 0.095
	sym7	97.16	92.01 ± 7.65	0.066	0.173 ± 0.090
	sym8	98.80	96.92 ± 1.69	0.062	0.142 ± 0.081
	sym9	98.61	96.34 ± 2.12	0.060	0.107 ± 0.067
	sym10	98.82	87.88 ± 17.77	0.028	0.142 ± 0.115
	coif2	96.23	87.69 ± 9.59	0.101	0.181 ± 0.090
	coif3	97.66	93.80 ± 4.07	0.043	0.081 ± 0.031
	coif4	98.24	92.30 ± 4.80	0.025	0.078 ± 0.043
	coif5	98.48	93.29 ± 3.33	0.056	0.217 ± 0.178
	coif6	98.82	89.83 ± 15.31	0.035	0.116 ± 0.099
	coif7	99.11	86.35 ± 14.23	0.028	0.123 ± 0.108
	coif8	98.81	97.69 ± 1.26	0.026	0.077 ± 0.048
	coif9	98.60	96.15 ± 3.25	0.042	0.101 ± 0.047
	coif10	98.47	96.40 ± 2.40	0.027	0.060 ± 0.028
GMM	LFCC (Frank & Schönherr)	–	–	0.062	–
RawNet2	raw (Frank & Schönherr)	–	–	0.363	–

Table 8: Number of model parameters and filter lengths for all trained deep fake detectors. The STFT-based and WPT-based networks have the same amount of optimizable parameters and the signed-log (SG) networks use a similar amount as well. For the STFT we use a hop length of 220 samples between the windows to achieve the same input dimensions as the WPT transformed audios.

	Model	Freq. Bins	Filter length	Model Parameters
DCNN	STFT	256	–	239,015
	haar	256	2	239,015
	db2	256	4	237,097
	db3	256	6	237,097
	db4 & sym4	256	8	237,097
	db5 & sym5	256	10	239,015
	db6 & sym6	256	12	239,015
	coif2	256	12	239,015
	db7 & sym7	256	14	239,015
	db8 & sym8	256	16	239,015
	db9 & sym9	256	18	241,099
	coif3	256	18	241,099
	db10 & sym10	256	20	241,099
	coif4	256	24	241,099
	sym14	256	28	243,349
	coif5	256	30	243,349
	coif6	256	36	245,765
	coif7	256	42	248,347
	coif8	256	48	248,347
	coif9	256	54	251,095
	coif10	256	60	254,009
LCNN	sym5	256	10	3,312,450