WaveFake: A Data Set to Facilitate Audio Deepfake Detection

Joel Frank
Ruhr University Bochum
Horst Görtz Institute for IT-Security
joel.frank@rub.de

Lea Schönherr
Ruhr University Bochum
Horst Görtz Institute for IT-Security
lea.schoenherr@rub.de

Abstract

Deep generative modeling has the potential to cause significant harm to society. Recognizing this threat, a magnitude of research into detecting so-called "Deepfakes" has emerged. This research most often focuses on the image domain, while studies exploring generated audio signals have— so far —been neglected. In this paper, we aim to narrow this gap. We present a novel data set, for which we collected ten sample sets from six different network architectures, spanning two languages. We analyze the frequency statistics comprehensively, discovering subtle differences between the architectures, specifically among the higher frequencies. Additionally, to facilitate further development of detection methods, we implemented three different classifiers adopted from the signal processing community to give practitioners a baseline to compare against. In a first evaluation, we already discovered significant trade-offs between the different approaches. Neural network-based approaches performed better on average, but more traditional models proved to be more robust.

1 Introduction

$243,000 were lost when criminals used a generated voice recording to impersonate the CEO of a UK company [94]. This is just one of several reports where current state-of-the-art generative modeling was used in harmful ways. Other examples include: impersonation attempts [23], influencing opposition movements [49], being used to justify military actions [28, 62], or online harassment [11]. While there is a multitude of beneficial use cases, for example, enhancing data sets for medical diagnostics [21, 25] or designing DNA to optimize protein bindings [38], finding effective ways to detect fraudulent usage is of utmost importance to society.

Detection in the image domain has received tremendous attention [56, 61, 111, 97, 102, 59, 63, 57, 20, 24]. However, the audio domain is severely lacking. In this paper, we aim at closing this gap. We start by reviewing standard signal processing techniques used for analyzing audio signals. For example, we give an introduction to spectrograms, which are commonly used as an intermediate representation for generative models [48, 75, 108, 109]. Additionally, we provide a survey of current state-of-the-art generative models.

Our main contribution is a novel data set. We collected ten sample sets from six different network architectures across two languages. This paper focuses on analyzing samples that resemble (i.e., recreate) the training distributions. This allows for one-to-one comparisons of audio clips between the different architectures. In this comparison, we find subtle differences between the generators. We also expect good performing classifiers to transfer well to other contexts since recreating the training distribution should yield the most quality samples. We test this hypothesis by also generating completely novel phrases.

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

Finally, we implement three classifiers, which we adopted from best practices in the signal processing community [83, 96], to give future researchers a baseline to compare against [1]. In a first evaluation we already discovered trade-offs between the different approaches. Neural networks performed better overall, but proved to be susceptible to changing settings. Finally, we implemented BlurIG [107] a popular attribution method/package, so practitioners can inspect their predictions when building on our results.

We summarize our main contributions as follows:

- A novel data set consisting of samples from several SOTA network architectures. Additionally, we perform a comprehensive analysis of this data set and find subtle differences between the different architectures.
- An implementation of two baseline models for future researchers to compare against. These models were evaluated in three different settings and we provide a popular attribution method to inspect the prediction.

2 Background

In this section, we provide an introduction to standard techniques used for analyzing speech audio signals. For additional material on the topic, the interested reader is referred to the excellent books by Rabiner et al. [78] or Quatieri [77]. Additionally, we provide a survey on current SOTA generative models and summarize related work.

2.1 Analyzing speech signals

(Mel) spectrograms: A spectrogram is a visual representation of the frequency information of a signal over time (cf. Section 3, Figure 2 for an example). To calculate a spectrogram for an audio signal, we first divide the waveform into *frames* (e.g., 20 ms) with an overlap (e.g., 10 ms) between two adjacent frames. We then apply a window function to avoid spectral leakage [2]. These functions (e.g., Hamming, Hann, Blackman window) are a trade-off between frequency resolution and spectral leakage. Their choice depends on the task and the signal properties, cf. Prabhu [72] for a detailed overview. The frames are then transformed individually using the *Discrete Fourier Transform* (DFT) to obtain a representation in the frequency domain $X(t,k)$. Where $t = 1, \dots, T$ is the frame index of the signal and $k = 0, \dots, K-1$ are the DFT coefficients. Finally, we calculate the squared magnitude $|X(t,k)|^2$ of the complex-valued signal to obtain our final representation—the spectrogram.

A commonly used variant is the so-called Mel spectrogram. It is motivated by studies that have shown that humans do not perceive frequencies on a linear scale. In particular, they can detect differences in lower frequencies with a higher resolution when compared to higher frequencies [117]. The Mel scale is an empirically determined non-linear transformation that approximates this relationship:

$$f_{\text{mel}} = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right), \tag{1}$$

where f is the frequency in Hz and f_{mel} the Mel-scaled frequency. To obtain the Mel spectrogram, we apply an ensemble of S triangular filters H_{mel} (we provide a visual representation in Section 8 of the supplementary material). These filters have a linear distance between the triangle mid frequencies in the Mel scale, which results in a logarithmic increasing distance of the frequencies in the frequency domain

$$X_{\text{mel}}(t,s) = \sum_{k=0}^{K-1} |X(t,k)| H_{\text{mel}}(s,k) \quad \forall\, s = 1, \dots, S, \tag{2}$$

which gives us the final Mel spectrogram. Based on it, we can compute a common feature representation for audio analysis.

[1] Our code and pretrained models can be found at `github.com/RUB-SysSec/WaveFake`

[2] Energies from one frequency leak into other frequency bins.

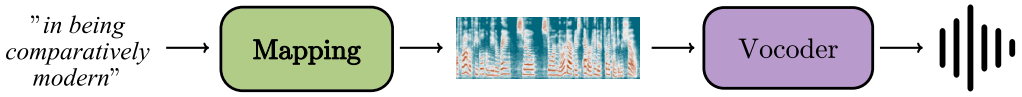


Figure 1: **A typical TTS pipeline.** One model takes a textual prompt with the desired audio transcription (we call it the "mapping" model) and outputs an intermediate representation, for example Mel spectrograms. This intermediate representation is then fed to a second model (the "vocoder") to obtain the final raw audio.

Mel Frequency Cepstral Coefficients: *Mel Frequency Cepstral Coefficients* (MFCC) are derived from a Mel-scaled spectrogram by applying a *Discrete Cosine Transform* (DCT) to the logarithm of the Mel-filtered signal

$$c(t,r) = \sum_{s=0}^{S-1} \log\left[X_{\text{mel}}(t,s)\right] \cdot \cos\left[\frac{\pi \cdot r \cdot (s+0.5)}{S}\right] \quad \forall\, r = 0,\ldots,R-1, \tag{3}$$

where R is the number of DCT coefficients.

Linear Frequency Cepstral Coefficients: We can also calculate *Linear Frequency Cepstral Coefficients* (LFCC). As the name suggests, these coefficients are derived by applying a linear filterbank (instead of a Mel filterbank) to the signal's spectrogram. This results in retaining more high-frequency information. Except for replacing the filter bank, all other steps remain the same as for MFCC features.

(Double) Delta features: MFCCs and LFCCs are often augmented by their first and second derivatives to represent the temporal structure of the input. These are referred to as delta and double delta features, respectively. In practice, these are often calculated by central difference approximation via

$$d(t) = \frac{\sum_{n=1}^{N} n \cdot \left[c\,(t+n) - c\,(t-n)\right]}{2 \cdot \sum_{n=1}^{N} n^2} \quad \forall\, t = 0,\ldots,T-1, \tag{4}$$

where $d(t)$ is the delta at time t and N is a user-defined window length for computing the delta, and c is either the MFCCs/LFCCs or the delta features (when calculating the double delta features).

2.2 Text-to-speech (TTS)

In this Section, we want to give a broad overview of different research directions for *Text-To-Speech* (TTS) models. Due to the rapid developments of the field, this is a non-exhaustive list.

While there has been some research into end-to-end models [19, 95], typical TTS models consist of a two-stage approach, represented in Figure 1. First, we enter the text sequence which we want to generate. This sequence gets mapped by some model (or feature extraction method) to a low-dimensional intermediate representation, often linguistic features [8] or Mel spectrograms [65]. Second, we use an additional model (often called vocoder) to map this intermediate representation to raw audio. We focus on the literature on vocoders since it directly connects to our work.

Today, vocoders are typically implemented by Deep Neural Networks (DNNs). The first DNN [113, 22] approaches adopted the parametric vocoders of earlier HMM-based models [114, 99, 110]. Here the DNN was used to predict the statistics of a given time frame, which are then used in traditional speech parameter generation algorithms [99]. Later variants replaced each component in traditional pipelines with neural equivalents [8, 7, 79, 80, 103, 4]. The first architectures which started using DNNs exclusively as the vocoder were auto-regressive generative models, such as WaveNet [65], WaveRNN [35], SampleRNN [60], Char2Wav [93] or Tactron 2 [90].

Due to their auto-regressive nature, these models do not leverage the parallel structure of modern hardware. There have been several attempts to circumvent this problem: One direction is to distill trained auto-regressive decoders into flow-based [42] convolutional student networks, as done by Parallel

WaveNet [65] and Clarinet [70]. Another method is to utilize direct maximum likelihood training as done by several flow-based models, for example, WaveGlow [75], FloWaveNet [39] or WaveFlow [71]. Other probabilistic approaches include those based on variational auto-encoders [66, 69] or diffusion probabilistic models [47, 15]. Another family of methods is based on Generative Adversarial Networks (GANs) [26], examples include, MelGAN [48], GAN-TTS [10], WaveGAN [18], HiFi-GAN [46], Parallel WaveGAN [108] or Multi-Band MelGAN [109].

2.3 Related Work

Several previous proposals have collected Deepfake data: The FaceForensics++ dataset [82] curated 1.8 million manipulated images and provides a benchmark for automated facial manipulation detection. Celeb-DF [54] contains high-quality face-swapping Deepfake videos of celebrities with more than 5,000 fake videos. Dolhansky et al. [17] released the Deepfake detection challenge that contains more than 100,000 videos, generated with different methods.

There exists a multitude of research into identifying GAN-generated images: Several approaches use CNNs in the image domain [56, 61, 111, 97, 102], others use statistics in the image domain [59, 63]. Another group of systems employs handcrafted features from the frequency domain: steganalysis-based features [57], spectral centroids [101] or frequency analysis [116, 20, 24, 76]. Li and Lyu [53] proposed a CNN-based Deepfake video detection framework that utilizes artefacts that are consequences of the generation process. Another strain of research combines image analysis with audio analysis. Chintha et al. [16] combined a Deepfake detection with an audio spoofing detection to identify fake videos. At the time of writing and to the best of our knowledge no work has provided an overview over Deepfake audio in isolation.

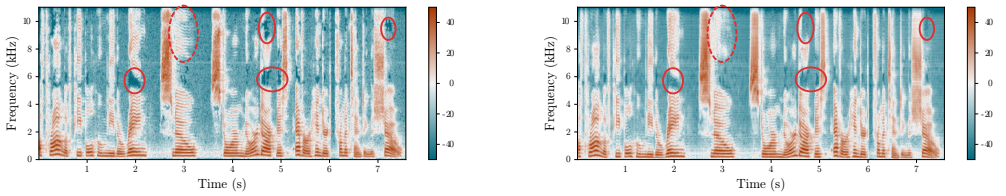
The signal processing community undertakes a related line of research. The biyearly ASVspoof challenges [106, 98, 64] promote countermeasure against spoofing attacks that aim to fool speaker verification systems via different kinds of attacks. Their benchmarking data sets include replay attacks, voice conversion, and synthesized audio files. Note that the 2021 edition of the challenge features an audio Deepfake track but does not provide specific training data. We imagine our data set to be used complementary with the training data of the challenge. At the time of writing the 2021 edition is still on-going, but evaluating the best performing models in conjunction with our data set is an interesting direction for future work. In the meantime, we adopt one of the baseline models of the ASVspoof challenge to enable a direct comparison. This bi-yearly challenge has led to several proposed models for detecting spoofing attacks, for example, CNN-based methods [100, 51, 50], ensemble methods on different feature representations [68, 36, 86] or methods which detect unusual pauses in human speech [115, 3]. Additionally, another data set is proposed by Kinnunen et al. [43]. They released a re-recorded version of the RedDots database for replay attack detection.

3 The data set

In this Section we provide an overview of our data set. It consists of 117,985 generated audio clips (16-bit PCM wav) and can be found on zenodo [3]. In total, it consists of approximately 196 hours of generated audio files. We mostly base our work on the LJSPEECH [33] data set. While TTS models often get trained on private data sets, LJSPEECH is the most common public data set among the publication listed in Section 2.2. Additionally, we consider the JSUT [92] data set, a Japanese speech corpus.

Reference data: We examine multiple networks trained on two reference data sets. First, the LJSPEECH [33] data set consisting of 13,100 short audio clips (on average 6 seconds each; roughly 24 hours total) read by a female speaker. It features passages from 7 non-fiction books, and the audio was recorded with a MacBook Pro microphone. Second, we include samples based on the JSUT [92] data set, specifically, the basic5000 corpus. This corpus consists of 5,000 sentences covering all basic kanji of the Japanese language. (4.8 seconds on average; roughly 6.7 hours total). A female native Japanese speaker performed the recordings in an anechoic room. Note that we do not redistribute the reference data. They are freely available online [33, 92].

[3] `https://zenodo.org/record/5642694` - DOI: 10.5281/zenodo.5642694



(a) Original (b) MelGAN

Figure 2: **Spectrograms for the same sample, for different generating models.** They show the frequencies of a signal, plotted over the time of a signal. Lower frequencies at the bottom, higher at the top. Best viewed in color.

Architectures: We included a range of architectures in our data set:

- **MelGAN**: We include MelGAN [48], which is one of the first GAN-based generative models for audio data. It uses a fully convolutional feed-forward network as the generator and operates on Mel spectrograms. The discriminator combines three different discriminators that operate on the original and two downsampled versions of the raw audio input. Additionally, it uses an auxiliary loss over the feature space of the three discriminators.
- **Parallel WaveGAN (PWG)**: WaveNet [65] is one of the earliest and most common architectures, We include samples from one of its variants, Parallel WaveGAN [108]. It uses the GAN training paradigm, with a non-autoregressive version of WaveNet as its generator. In a similar vein to MelGAN, it uses an auxiliary loss, but in contrast, matches the *Short-Time Fourier Transform* (STFT) of the original training sample and the generated waveform over multiple resolutions.
- **Multi-band MelGAN (MB-MelGAN)**: Incorporating more fine-grained frequency analysis, might lead to more convincing samples. We include MB-MelGAN, which computes its auxiliary (frequency-based; inspired by PWG) loss in different sub-bands. Its generator is based on a bigger version of the MelGAN generator. Still, instead of predicting the entire audio directly, the generator produces multiple sub-bands, which are then summed up to the complete audio signal.
- **Full-band MelGAN (FB-MelGAN)**: We include a variant of MB-MelGAN which generates the complete audio directly and computes its auxiliary loss (the same as PWG) over the full audio instead of its sub-bands.
- **HiFi-GAN (HiFi-GAN)**: HiFi-GAN [46] utilizes multiple sub-discriminators, each of which examines only a specific periodic part of the input waveform. Similarly, its generator is built with multiple residual blocks, each observing patterns of different lengths in parallel. Additionally, HiFi-GAN employs the feature-space-based loss from MelGAN and minimizes the L_1 distance between the Mel spectrogram of a generated waveform and a ground truth one in its loss function.
- **WaveGlow**: The training procedure might also influence the detectability of fake samples. Therefore, we include samples from WaveGlow to investigate maximum-likelihood-based methods. It is a flow-based generative model based on Glow [41], whose architecture is heavily inspired by WaveNet.

Additionally, we examine MelGAN both in a version similar to the original publication, which we denote as MelGAN, and in a larger version with a bigger receptive field, MelGAN (L)arge. This version is similar to the one used by FB-MelGAN, allowing for a one-to-one comparison. Finally, we also obtain samples from a complete TTS-pipeline. We use a conformer [27] to map novel phrases (i.e., not part of LJSPEECH) to Mel spectrograms. Then we use a fine-tuned PWG model (trained on LJSPEECH) to obtain the final audio. We call this data set TTS. In total, we sample ten different data sets, seven based on LJSPEECH (MelGAN, MelGAN (L), FB-MelGAN, HiFi-GAN, WaveGlow, PWG, TTS) and two based on JSUT (MB-MelGAN, PWG).

Sampling procedure: For WaveGlow we utilize the official implementation [74] (commit 8afb643) in conjunction with the official pre-trained network on PyTorch Hub [73]. HiFi-GAN also offers a public repository with pretrained models [45]. We use a popular implementation available on GitHub [29] (commit 12c677e) for the remaining networks. When sampling the data set, we first extract Mel spectrograms from the original audio files, using the pre-processing scripts of the

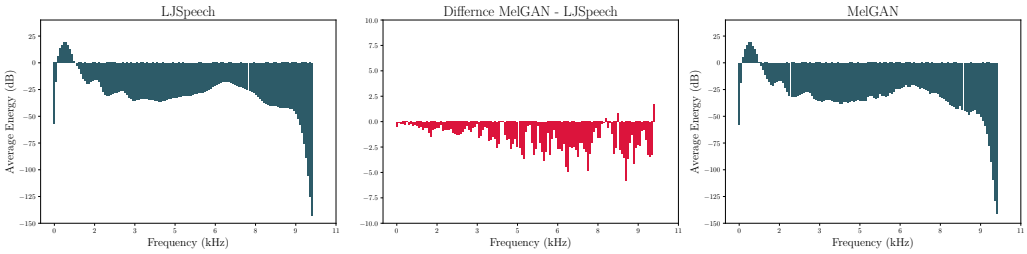


Figure 3: **Average energy per frequency bin.** We show the average energy per frequency bin in dB. Additionally, we plot the difference to the original data (LJSPEECH).

Table 1: **Basic statistics for all LJSPEECH-based data sets.** We report the average pitch frequency and its standard deviation as well as the average spectral centroid.

	LJSPEECH	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG
Avg. Pitch	137.61	133.51	130.94	135.22	133.519	133.60	135.80	131.018
Std. Pitch	49.64	47.00	46.15	48.90	48.19	48.09	47.38	47.08
Avg. Centroid	2367.79	2414.81	2355.59	2362.22	2414.81	2374.51	2422.63	2348.31

corresponding repositories. We then feed these Mel spectrograms to the respective models to obtain the data set. Intuitively, the networks are asked to "recreate" the original data sets. For sampling the full TTS results, we use the ESPnet project [104, 30, 32, 52]. To make sure the generated phrases do not overlap with the training set, we downloaded the common voices data set [6] and extracted 16,283 phrases from it.

Differences between the architectures: We perform an analysis of the differences between the architectures. First, by plotting the spectrograms of an audio file in Figure 2 (LJSPEECH 008-0217; all data sets can be found in Section 6 of the supplementary material). Generally, all architectures produce spectrograms different from the original. The networks seem to struggle with the absence of information generally (solid circles in Figure 2a) as well as with higher frequencies, especially for vocals (dashed circle). Additionally, MelGAN and WaveGlow seem to cause a repeating horizontal pattern. The remaining networks (all using an auxiliary loss in the frequency domain) do not seem to exhibit this behavior. However, they still produce apparent differences. Note that these differences are visible when plotting the audio but generally inaudible when listening to the samples.

Second, we perform a prosody analysis of each data set. We collect 10.000 samples from LJSPEECH and the corresponding sample from each of our architectures. For each data set, we compute the fundamental frequency (pitch) by using normalized cross-correlation and median smoothing [31] in the range $50 - 500$Hz. Additionally, we compute the center of mass of the frequency spectrum by the mean of the frequencies weighted by their magnitudes (the so-called spectral centroid). The results can be found in Table 1 and confirm our visual observations. While all architectures come close to the original, none can approximate it perfectly. Generally, all vocoders produce a lower and less varied pitch.

The spectral centroid results are varied. To investigate further, we perform an additional, more fine-grained analysis by plotting a histogram of the energy contained in each frequency bin. Furthermore, we plot the relative difference to the original data, i.e., the difference weighted by the initial (LJSPEECH) energy. The plots can be found in Figure 3. For brevity, we only show the MelGAN comparison here, the other histograms can be found in Section 6 of the supplementary material. However, all analyses had similar results. The histograms' overall shape is identical, but the generated samples exhibit apparent differences, especially in the higher frequencies.

Licensing: The LJSPEECHdata set is in the public domain. The JSUTcorpus is licensed by CC-BY-SA 4.0, noting that redistribution is only permitted in some instances. We contacted the author, who saw no conflict in distributing our fake samples, as long as it's for research purposes. Thus, we also released our data set under a CC-BY-SA 4.0 license.

Ethical considerations: Our data set consists of phrases from non-fiction books (LJSPEECH) and everyday conversational Japanese (JSUT), which are already available online. The same is true

for all models used to generate this data set. One might wonder if releasing research into detecting Deepfakes might negatively affect the detection "arms race". This discussion has a long-standing history in the security community, and the general conclusion is that withholding research is hurtful. We provide a more in-depth discussion of this topic from the perspective of security researchers in the supplementary material (Section 1).

4 Providing a baseline

We base our experiment on the ASVspoofchallenge [98] introduced in Section 2.3. The challenge aims to promote research into detecting (audio) spoofing attacks and speaker verification. As a point of reference, the challenge offers two baseline models (*Gaussian Mixture Model* (GMM) and RawNet2 [34, 96]). We adopt these models and the metric used by the challenge to compare the two domains.

4.1 Experiments

We start by training seven different classifiers, one for each vocoder in our data set (MelGAN, MelGAN (L), FB-MelGAN, MB-MelGAN, HiFi-GAN, PWG and WaveGlow). For training our classifiers, we exclusively use the data sets based on LJSPEECH. We use the JSUT (different speaker, language, and recording setup) and TTS (same speaker, completely novel phrases) data sets for accessing the classifiers ability to generalize to an unknown setting. We train six additional models in a leave-one-out setting to control if the models picked up on vocoder-specific characteristics. Finally, we simulate a phone recording to emulate a real-world fraud attempt.

For each classifier, we evaluate the performance on all vocoders over a hold-out set of 20% of the data. We use the *Equal Error Rate* (EER) as our evaluation metric. The ASVspoof challenge also uses this metric. It is defined as the point on the ROC curve, where false acceptance rate and false rejection rate are equal and is commonly used to assess the performance of binary classifications tasks like biometric security systems [85]. The best possible value is 0.0 (no wrong predictions), worst 1.0 (everything wrong), guessing is 0.5. Additionally, we compute *average Equal Error Rate* (aEER) over all test sets.

When training GMM models, we follow Sahidullah et al. [83] and train two GMMs per data set, one fitting the real distribution (the original LJSPEECH data set) and one fitting the generated audio samples (the respective vocoder-samples from our data set). In addition to the LFCC features used by Sahidullah et al. [83], we evaluate MFCC features. To classify a given sample, we calculate its likelihood $\Lambda(\mathbf{X})$ via

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X}|\theta_n) - \log p(\mathbf{X}|\theta_s), \tag{5}$$

where $\mathbf{X}$ are the input features (namely MFCC or LFCC) and θ_n and θ_s are the GMM model parameter of the real and the generated audio distributions, respectively. The out-of-distribution models are exclusively trained on LFCC features, since we found them to strictly outperform the MFCC features (cf., Section 2).

Additionally, we train RawNet2 [34] instances to investigate a neural alternative. RawNet2 is a CNN-GRU hybrid model which extracts a speaker embedding directly from raw audio. When used to perform speaker verification (or Deepfake detection), a fully connected layer is trained on top of this embedding to make the final decision [96]. Details on all training setups can be found in the supplementary material (Section 3).

Single training set: In a first experiment, we evaluate the performance when training on a single data set. For the GMM experiments, we only present the LFCC results since we found them to outperform the MFCC features strictly. LFCC features contain a significantly higher amount of high-frequency components. We hypothesize that these are meaningful for achieving good overall performance. Similar patterns were observed in the image domain [24], implying that methods might transfer between the two. The results are presented in Table 2. The rows show the respective training sets and the columns the different test sets. Gray values indicate that the same generative model is used for the training of the GMM classifier as for the test set.

When training on a single data set, we observe that FB-MelGAN serves as the best prior for all other data sets, achieving the lowest average EER (0.062). Intuitively this makes sense. FB-MelGAN

Table 2: **Equal Error Rate (EER) of the baseline classifier on different subset (LFCC).** We train a new GMM model for each data set and compute the EER as well as the **aEER**.

	LJSPEECH								JSUT		
Training Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG	TTS	MB-MelGAN	PWG	**aEER**
MelGAN	0.148	**0.094**	0.155	0.153	0.168	0.189	0.109	0.023	0.384	0.533	0.215
MelGAN (L)	**0.119**	0.044	0.176	0.132	0.150	0.245	0.115	**0.012**	0.406	0.607	0.222
MB-MelGAN	0.359	0.316	0.002	0.124	0.083	0.007	**0.017**	0.021	0.017	0.051	0.108
FB-MelGAN	0.197	0.133	0.030	0.025	**0.034**	0.037	0.019	0.025	0.026	0.058	**0.062**
HiFi-GAN	0.255	0.193	0.034	**0.050**	0.029	0.035	0.020	0.018	0.057	0.123	0.089
PWG	0.402	0.374	0.008	0.161	0.100	0.001	**0.017**	0.042	**0.014**	**0.042**	0.124
WaveGlow	0.287	0.237	**0.015**	0.066	0.041	**0.008**	0.003	0.015	0.031	0.075	0.085

When the distribution is part of the training set we highlight it in gray. For other results, we highlight the best results per column in **bold**.

Table 3: **Equal Error Rate (EER) of the RawNet2 classifier.** We train a single RawNet2 model per data set and compute the EER as well as the **aEER**.

	LJSPEECH								JSUT		
Training Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG	TTS	MB-MelGAN	PWG	**aEER**
MelGAN	0.001	**0.001**	0.485	0.509	0.525	0.497	0.407	0.356	0.113	0.089	0.292
MelGAN (L)	**0.008**	0.000	0.511	0.490	0.486	0.369	0.446	0.265	**0.009**	**0.003**	**0.258**
MB-MelGAN	0.118	0.371	0.003	0.282	**0.216**	0.302	**0.002**	0.522	0.922	0.997	0.357
FB-MelGAN	0.161	0.239	0.122	0.082	0.304	**0.259**	0.130	0.391	0.974	0.994	0.363
HiFi-GAN	0.174	0.437	0.242	0.364	0.023	0.359	0.057	0.098	0.499	0.719	0.319
PWG	0.052	0.358	0.261	**0.234**	0.324	0.000	0.006	**0.001**	0.984	0.999	0.358
WaveGlow	0.086	0.379	**0.07**9	0.361	0.226	0.316	0.001	0.250	0.409	0.786	0.294

When the distribution is part of the training set we highlight it in gray. For other results, we highlight the best results per column in **bold**.

uses the same architecture as MelGAN (L), while deploying a similar auxiliary loss as PWG or MB-MelGAN. Generally, we can see a clear divide between MelGAN/MelGAN (L) and the other networks, which we will explore in Section 4.2.

When examining completely novel data (JSUT), all classifier drop in performance. However, PWG, WaveGlow, HiFi-GAN, FB-MelGAN and, MB-MelGAN still serve as a good prior, implying that the generating architectures exhibit common patterns which can be recognized for different training data sets and speakers. Again, a similar pattern was also observed in the image domain [102]. The TTS data set is one of the easiest data sets. This undermines our belief that data that recreates the training set is harder to classify correctly. Interestingly the PWG classifier does not generalize well to the TTS data set. Remember that while we use completely novel phrases, the vocoder for this data set is a PWG network trained on LJSPEECH. This might imply that our models overfit their specific training set.

This trend can also be seen in the RawNet2 results, which overall perform worse than the GMM models. They seem to overfit their respective training architecture, which prevents them from generalizing to other data sets. This explains the good performance of the MelGAN/MelGAN (L) models and the PWG/TTS models, since these pairs share generator architectures. Additionally, we can note that the MelGAN/MelGAN (L) models serve as a good prior for generalizing to the JSUT data sets.

Leave-one-out: We explore this hypothesis by running a leave-one-out experiment. Results can be found in Table 4. Overall the results improve on the aEER ($0.062 \rightarrow 0.058$). Also, the generalization results to a novel setting (JSUT) increase significantly. However, FB-MelGAN seems to be an essential ingredient for good performance on the JSUT data since not training on it hurts performance significantly. Additionally, the MelGAN and MelGAN (L) data sets still prove to be a challenge, even when included in the training set.

The results are similar for RawNet2 (Table 5). When trained on multiple distributions, the networks can successfully generalize, even surpassing the best aEER (0.04). However, some models still overfit to the training data, making generalization to JSUT impossible (MelGAN (L), MB-MelGAN, WaveGlow). Additionally, the better average performance is traded off with worse performance on the training data. For example, the best performing average model has a 13% false acceptance/false rejection rate. This would be unacceptable in a real-life setting.

Simulating a phone call: Finally, we return to our motivating example and examine how well our models generalize to a (simulated) real-life setting. We emulate a phone recording for the three test

Table 4: **Equal Error Rate (EER) for the GMM classifier in an out-of-distribution setting.** We train a new GMM model for each but one distribution on LFCC features.

	LJSPEECH								JSUT		
Training Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG	TTS	MB-MelGAN	PWG	**aEER**
MelGAN	0.220	0.146	0.009	0.051	0.040	0.016	0.009	0.006	0.023	0.067	0.065
MelGAN (L)	0.231	0.164	0.010	0.045	0.040	0.014	0.012	0.009	0.013	**0.043**	0.064
MB-MelGAN	0.187	0.117	0.013	0.043	0.039	0.018	0.010	**0.002**	0.058	0.141	0.069
FB-MelGAN	0.191	0.116	0.013	0.058	0.053	0.022	0.013	0.003	0.084	0.220	0.086
HiFi-GAN	0.192	0.119	0.011	0.050	0.047	0.015	0.012	0.004	**0.020**	0.061	**0.058**
PWG	0.176	0.105	0.014	0.044	0.042	0.034	0.013	0.005	0.033	0.101	0.062
WaveGlow	0.191	0.114	0.013	0.049	0.045	0.021	0.015	0.008	0.031	0.078	0.062

When the distribution is part of the training set we highlight it in gray. For other results, we highlight the best results per column in **bold**.

Table 5: **Equal Error Rate (EER) of the RawNet2 classifier in an out-of-distribution setting.** We train a single RawNet2 model on all but one distribution and compute the EER as well as the **aEER**.

	LJSPEECH								JSUT		
Training Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG	TTS	MB-MelGAN	PWG	**aEER**
MelGAN	0.008	0.005	0.023	0.137	0.098	0.076	0.011	0.019	**0.000**	**0.000**	**0.040**
MelGAN (L)	0.005	0.046	0.009	0.048	0.050	0.024	0.004	0.020	0.985	0.996	0.241
MB-MelGAN	0.013	0.039	0.037	0.102	0.060	0.055	0.005	0.089	0.860	0.758	0.214
FB-MelGAN	0.013	0.023	0.032	0.216	0.058	0.054	0.011	0.026	0.092	0.088	0.065
HiFi-GAN	0.006	0.009	0.031	0.113	0.196	0.065	0.010	0.044	0.001	0.001	0.048
PWG	0.005	0.004	0.026	0.108	0.088	0.209	0.011	0.044	0.047	0.123	0.069
WaveGlow	0.001	0.006	0.008	0.038	0.046	0.010	0.001	**0.005**	0.828	0.904	0.205

When the distribution is part of the training set we highlight it in gray. For other results, we highlight the best results per column in **bold**.

data set (both JSUT data sets and the full TTS-pipeline) and evaluate the out-of-distribution models on them. The GMM classifiers work exceptionally well, even surpassing the performance in the out-of-distribution setting. The highest EER we could detect was 0.003, all other results were lower or separated the data perfectly (a full table can be found in Section 5 of the supplementary material). The results are flipped for the RawNet2 models. While in the clean setting, the (some) models classify the data almost perfectly, under the phone simulation, the error rate shoots up significantly.

While we only examined a simulated setting, we take this as the first evidence that our data set can be used to extrapolate classifier performance to the real world. Overall, we can conclude that these first results are encouraging, but there is still much room for improvement.

4.2 Attribution

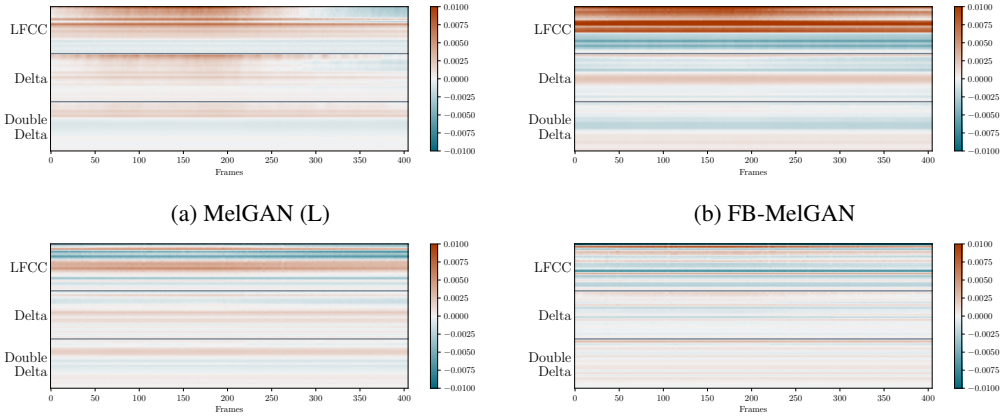
Finally, we want to investigate which parts of the audio signal influence the prediction. To this end, we implemented BlurIG [107], a popular attribution method. We inspect the attribution of four classifiers for the audio clip used in Section 3. The results are displayed in Figure 4.

Overall, we can see a shift from very broad attention, spread somewhat evenly across all three feature representations (MelGAN (L)), to more narrow-focused attention across very specific filters (PWG). MelGAN (L) and FB-MelGAN classifiers operate (mostly) on the higher frequencies, while MB-MelGAN and PWG also rely on low frequencies for the detection. These observations confirm our suspicion about the MFCC features. They mask higher frequencies, needed for classifying MelGAN (L) and FB-MelGAN, while over-representing lower frequencies, which still leads to a good performance on the MB-MelGAN and PWG data sets. This also explains the performance of FB-MelGAN, which strikes a balance between all necessary features. The spread-out attribution might also explain the poor in-distribution performance of the classifiers trained on the MelGAN variants since the classifier needs to focus on a broader range of features.

5 Discussion

In this paper, we took the first step towards research into audio Deepfakes. While we hope our data set proves useful for future practitioners, there are several limitations to our work:

Evaluating on realistic data: The difficulties of obtaining realistic data set have been a long-standing problem in the security community [91]. Often benign data is readily available, but data



(a) MelGAN (L)

(b) FB-MelGAN

(c) MB-MelGAN

(d) PWG

Figure 4: **Attribution of the different models on a real audio sample.** We show the LFCC, delta, and, double delta features. The plot can be read similarly to the spectrogram plots, i.e., features computed over lower frequencies are at the bottom of their respective section, features over higher frequencies are at the top. Best viewed in color.

used in malicious contexts is hard to come by. That leaves us with estimating real-world performance on proxy data. We argue that in our case, we might have good odds that results transfer. Currently, images generated by off-the-shelf neural networks are used in malicious attempts [11]. We expect the number of audio Deepfakes to increase as well.

Variety of the data: We specifically choose to focus on the LJSPEECH corpus since it is commonly used for training generative audio models. That allows a one-to-one comparison. However, it only contains recordings by one speaker. We can make some observations about generalization by comparing against the JSUT and TTS data sets, but a broader analysis focusing on different scenarios would be ideal.

Adversarial examples and perturbations: Deepfake-image detectors have already been shown to be vulnerable against adversarial examples [12]. There also exists a myriad of adversarial attacks against automatic speech recognition [13, 87, 112, 87, 5, 88, 2] (Abdullah et al. [1] provide a survey). We have looked at phone recordings, but classifiers should report their robustness against these attacks and other common perturbations (noise, room responses, over-the-air settings, etc.) as part of their evaluation. In this work, we focused on providing the first steps towards audio Deepfake detection.

6 Conclusion

This paper presents a starting point for researchers who want to investigate generated audio signals. We started by presenting a broad overview of signal processing techniques and common feature representations as well as a survey of the current TTS landscape. We then moved on to introduce our main contribution, a novel data set, with samples from six different state-of-the-art architectures across two languages. We discovered subtle differences between the different models by plotting the frequency spectrum, especially among the higher frequencies. Following up, we conducted a prosody analysis and investigated each data set's average energy per frequency. This analysis confirmed our previous findings, revealing that while all models come close to correctly approximating the training data, we can still detect differences unique to each model. To provide a baseline for future practitioners, we trained several baseline models. We evaluated their performance across the different data sets and multiple settings. Specifically, we trained GMM and neural network-based solutions. While we found the neural networks to perform better overall, the GMM classifiers proved to be more robust, which might give them an advantage in real-life settings. Finally, we inspected the different classifiers using an attribution method. We found that lower frequencies cannot be neglected while high-frequency information proved indispensable.

Acknowledgments and Disclosure of Funding

We would like to thank our colleagues Thorsten Eisenhofer, Thorsten Holz, Dorothea Kolossa, and our anonymous reviewers for their valuable feedback and fruitful discussions. Additionally, we would like to thank Tomoki Hayashi, Hemlata Tak, and the WaveGlow team for their excellent repositories. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2092 CASA – 390781972.

References

[1] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[2] Hojjat Aghakhani, Thorsten Eisenhofer, Lea Schönherr, Dorothea Kolossa, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. VENOMAVE: Clean-Label Poisoning Against Speech Recognition. *Computing Research Repository (CoRR)*, abs/2010.10682, 2020.

[3] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. Void: A fast and light voice liveness detection system. In *USENIX Security Symposium*, 2020.

[4] Yang Ai and Zhen-Hua Ling. A Neural Vocoder With Hierarchical Generation of Amplitude and Phase Spectra for Statistical Parametric Speech Synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[5] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. Did you hear that? Adversarial Examples Against Automatic Speech Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[6] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common Voice: A Massively-Multilingual Speech Corpus. In *Language Resources and Evaluation Conference*, 2020.

[7] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep Voice 2: Multi-Speaker Neural Text-to-Speech. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep Voice: Real-Time Neural Text-to-Speech. In *International Conference on Machine Learning (ICML)*, 2017.

[9] Elaine Barker et al. Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms. *NIST special publication*, 2016.

[10] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C Cobo, and Karen Simonyan. High Fidelity Speech Synthesis with Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] Matt Burgess. Telegram Still Hasn't Removed an AI Bot That's Abusing Women. *Wired*, 2020.

[12] Nicholas Carlini and Hany Farid. Evading DeepFake-Image Detectors with White-and Black-Box Attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] Nicholas Carlini and David Wagner. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. In *IEEE Deep Learning and Security Workshop (DLS)*, 2018.

[14] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On Evaluating Adversarial Robustness. *Computing Research Repository (CoRR)*, abs/1902.06705, 2019.

[15] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating Gradients for Waveform Generation. In *International Conference on Learning Representations (ICLR)*, 2020.

[16] Akash Chintha, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. Recurrent Convolutional Structures for Audio Spoof and Video DeepFake Detection. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[17] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The DeepFake Detection Challenge (DFDC) Dataset, 2020.

[18] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial Audio Synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] Jeff Donahue, Sander Dieleman, Mikołaj Bińkowski, Erich Elsen, and Karen Simonyan. End-to-End Adversarial Text-to-Speech. In *International Conference on Learning Representations (ICLR)*, 2021.

[20] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-Valued (Medical) Time Series Generation with Recurrent Conditional GANs. In *International Conference on Learning Representations (ICLR)*, 2018.

[22] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong. TTS Synthesis with Bidirectional LSTM Based Recurrent Neural Networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[23] Lorenzo Franceschi-Bicchierai. Listen to This Deepfake Audio Impersonating a CEO in Brazen Fraud Attempt. *Motherboard*, 2020.

[24] 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 (ICML)*, 2020.

[25] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. GAN-Based synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification. *Neurocomputing*, 2018.

[26] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[27] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-Augmented Transformer for Speech Recognition. In *Proceedings of Interspeech (INTERSPEECH)*, 2020.

[28] Karen Hao. The Biggest Threat of Deepfakes isn't the Deepfakes Themselves. *MIT Technology Review*, 2019.

[29] Tomoki Hayashi. Parallel WaveGAN (+ MelGAN & Multi-band MelGAN) implementation with Pytorch. `https://github.com/kan-bayashi/ParallelWaveGAN`, 2020.

[30] Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[31] Thomas Huang, GJTGY Yang, and Greory Tang. A fast two-dimensional median filtering algorithm. *IEEE transactions on acoustics, speech, and signal processing*, 27(1):13–18, 1979.

[32] Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[33] Keith Ito and Linda Johnson. The LJ Speech Dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[34] 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. *Proceedings of Interspeech (INTERSPEECH)*, 2020.

[35] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient Neural Audio Synthesis. In *International Conference on Machine Learning (ICML)*, 2018.

[36] Madhu R Kamble, Hemlata Tak, and Hemant A Patil. Effectiveness of Speech Demodulation-Based Features for Replay Detection. In *Proceedings of Interspeech (INTERSPEECH)*, 2018.

[37] Auguste Kerckhoffs. *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883.

[38] Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing DNA with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

[39] Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. FloWaveNet: A Generative Flow for Raw Audio. In *International Conference on Machine Learning (ICML)*, 2019.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[41] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[42] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. In *International Conference on Learning Representations (ICLR) - Workshop track*, 2016.

[43] Tomi Kinnunen, Md Sahidullah, Mauro Falcone, Luca Costantini, Rosa González Hautamäki, Dennis Thomsen, Achintya Sarkar, Zheng-Hua Tan, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Ville Hautamäki, and Kong Aik Lee. RedDots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[44] Paul Kocher, Jann Horn, Anders Fogh, , Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. In *40th IEEE Symposium on Security and Privacy (S&P'19)*, 2019.

[45] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. `https://github.com/jik876/hifi-gan`, 2020.

[46] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[47] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *International Conference on Learning Representations (ICLR)*, 2021.

[48] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[49] The Atlantic Council's Digital Forensic Research Lab. Inauthentic Instagram accounts with synthetic faces target Navalny protests. *Medium*, 2021.

[50] Cheng-I Lai, Alberto Abad, Korin Richmond, Junichi Yamagishi, Najim Dehak, and Simon King. Attentive Filtering Networks for Audio Replay Attack Detection. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[51] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. Audio Replay Attack Detection with Deep Learning Frameworks. In *Proceedings of Interspeech (INTERSPEECH)*, 2017.

[52] Chenda Li, Jing Shi, Wangyou Zhang, Aswin Shanmugam Subramanian, Xuankai Chang, Naoyuki Kamo, Moto Hira, Tomoki Hayashi, Christoph Boeddeker, Zhuo Chen, and Shinji Watanabe. ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[53] Yuezun Li and Siwei Lyu. Exposing DeepFake Videos by Detecting Face Warping Artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[54] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A New Dataset for DeepFake Forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[55] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown: Reading kernel memory from user space. In *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[56] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of GAN-Generated Fake Images over Social Networks. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018.

[57] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs Leave Artificial Gingerprints? In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019.

[58] Bill McCarty. The honeynet arms race. *IEEE Security & Privacy*, 2003.

[59] Scott McCloskey and Michael Albright. Detecting GAN-Generated Imagery Using Color Cues. *arXiv preprint arXiv:1812.08247*, 2018.

[60] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. *arXiv preprint arXiv:1612.07837*, 2016.

[61] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake Faces Identification via Convolutional Neural Network. In *ACM Workshop on Information Hiding and Multimedia Security*, 2018.

[62] Peter Mwai. Tigray conflict: The fake UN diplomat and other misleading stories. *BBC Reality Check*, 2021.

[63] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting GAN Generated Fake Images Using Co-Occurrence Matrices. *Electronic Imaging*, 2019.

[64] Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi H. Kinnunen, Ville Vestman, Massimiliano Todisco, Héctor Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee. ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021.

[65] 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.

[66] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[67] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[68] Hemant A Patil, Madhu R Kamble, Tanvina B Patel, and Meet H Soni. Novel Variable Length Teager Energy Separation Based Instantaneous Frequency Features for Replay Detection. In *Proceedings of Interspeech (INTERSPEECH)*, 2017.

[69] Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao. Non-Autoregressive Neural Text-to-Speech. In *International Conference on Machine Learning (ICML)*, 2020.

[70] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel Wave Generation in End-to-End Text-to-Speech. In *International Conference on Learning Representations (ICLR)*, 2019.

[71] Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. WaveFlow: A Compact Flow-based Model for Raw Audio. In *International Conference on Learning Representations (ICLR)*, 2020.

[72] KM Muraleedhara Prabhu. *Window Functions and their Applications in Signal Processing*. Taylor & Francis, 2014.

[73] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: a Flow-based Generative Network for Speech Synthesis. `https://pytorch.org/hub/nvidia_deeplearningexamples_waveglow/`, 2018.

[74] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: a Flow-based Generative Network for Speech Synthesis. `https://github.com/NVIDIA/waveglow`, 2018.

[75] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A Flow-based Generative Network for Speech Synthesis. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[76] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues. In *European Conference on Computer Vision (ECCV)*, 2020.

[77] Thomas Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. Pearson Education India, 2006.

[78] Lawrence Rabiner, Bernard Gold, and CK Yuen. *Theory and Application of Digital Signal Processing*. Prentice-Hall, 2016.

[79] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. FastSpeech: Fast, Robust and Controllable Text to Speech. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[80] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. *arXiv preprint arXiv:2006.04558*, 2020.

[81] Vincent Rijmen and Joan Daemen. Advanced Encryption Standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, 2001.

[82] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to Detect Manipulated Facial Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[83] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A Comparison of Features for Synthetic Speech Detection. In *Proceedings of Interspeech (INTERSPEECH)*, 2015.

[84] Karen Scarfone, Wayne Jansen, Miles Tracy, et al. Guide to General Server Security. *NIST Special Publication*, 2008.

[85] Dirk Scheuermann, Scarlet Schwiderski-Grosche, and Bruno Struif. *Usability of Biometrics in Relation to Electronic Signatures*. GMD-Forschungszentrum Informationstechnik Sankt Augustin, 2000.

[86] Lea Schönherr, Steffen Zeiler, and Dorothea Kolossa. Spoofing Detection via Simultaneous Verification of Audio-Visual Synchronicity and Transcription. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.

[87] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. In *Symposium on Network and Distributed System Security (NDSS)*, 2019.

[88] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems. In *Annual Computer Security Applications Conference (ACSAC)*, 2020.

[89] Claude E Shannon. Communication Theory of Secrecy Systems. *The Bell system technical journal*, 1949.

[90] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[91] Robin Sommer and Vern Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *IEEE Symposium on Security and Privacy (S&P)*, 2010.

[92] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. JSUT Corpus: Free Large-Scale Japanese Speech Corpus for End-to-End Speech Synthesis. *arXiv preprint arXiv:1711.00354*, 2017.

[93] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-End Speech Synthesis. In *International Conference on Learning Representations (ICLR) Workshop Track*, 2017.

[94] Catherine Stupp. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case. *The Wall Street Journal*, 2019.

[95] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop. In *International Conference on Learning Representations (ICLR)*, 2017.

[96] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-End anti-spoofing with RawNet2. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[97] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. GAN is a Friend or Foe? A Framework to Detect Various Fake Face Images. In *ACM/SIGAPP Symposium on Applied Computing*, 2019.

[98] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Hector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. *Computing Research Repository (CoRR)*, abs/1904.05441, 2019.

[99] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000.

[100] Francis Tom, Mohit Jain, and Prasenjit Dey. End-To-End Audio Replay Attack Detection Using Deep Convolutional Networks with Attention. In *Proceedings of Interspeech (INTERSPEECH)*, 2018.

[101] Rafael Valle, Wilson Cai, and Anish Doshi. TequilaGAN: How to Easily Identify GAN Samples. *arXiv preprint arXiv:1807.04919*, 2018.

[102] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[103] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[104] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech (INTERSPEECH)*, 2018.

[105] Tim Willis. Project Zero Policy and Disclosure: 2020 Edition, 2020. `https://googleprojectzero.blogspot.com/2020/01/policy-and-disclosure-2020-edition.html`, as of November 23, 2021.

[106] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Héctor Delgado. ASVspoof: The Automatic Speaker Verification Spoofing and Countermeasures Challenge. *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[107] Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in Scale and Space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[108] 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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[109] 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 *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[110] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[111] Ning Yu, Larry S Davis, and Mario Fritz. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[112] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A. Gunter. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition. In *USENIX Security Symposium*, 2018.

[113] Heiga Ze, Andrew Senior, and Mike Schuster. Statistical Parametric Speech Synthesis Using Deep Neural Networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[114] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical Parametric Speech Synthesis. *Speech Communication*, 2009.

[115] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. Voicelive: A Phoneme Localization Based Liveness Detection for Voice Authentication on Smartphones. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[116] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and Simulating Artifacts in GAN Fake Images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019.

[117] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics: Facts and Models*. Springer, Heidelberg, Germany, 2 edition, 2007.

Supplementary Material

In this supplementary material, we provide an extended discussion on security research, a note on licensing generative models, our training details, the results on MFCC data, and full-size spectrogram and attribution plots. Additionally, we provide a visual representation of the filterbanks.

A A note on releasing security research

One might wonder if releasing research into detecting Deepfakes might negatively affect the detection "arms race". That is a long-standing debate in the security community. The overall consensus is that "security through obscurity" does not work. This is often echoed in best security practices, for example, published by the National Institute of Standards and Technology (NIST) [84]. Intuitively, withholding information from the research community is more harmful since attackers will eventually adapt to any defense one deploys anyway. Thus, contributing to the invention of new systems is more helpful in an ever-changing environment [58]

The debate dates back to at least the 19th century where the cryptographer Auguste Kerckhoffs introduced Kerckhoffs's principle [37]. The principle states that an encryption scheme should still work if an adversary knows everything about the system but a secret passphrase. Similar thought would later be formulated by Claude Shannon [89].

A typical example is the advanced encryption standard (AES). The algorithms entire specification and inner workings can be found in the standardization [81]. Yet, it is considered unbreakable as long as the password used for the encryption is not revealed. AES is also the only algorithm used to encrypt US government documents [9]. The principle also found adoption in the machine learning community, where adversarial defense papers are now advised to evaluate against so-called white box attackers [14], i.e., attackers which know the inner workings of the system and actively try to avoid it.

While complete openness is not possible, the greater security community has adopted similar practices. For example, so-called attack papers are regularly published at security venues. The underlying motivation being, that before one can protect systems, one has to understand how to attack them. Prominent examples are the Meltdown [55] and Spectre [44] vulnerabilities which showed that certain instructions in CPUs could be used for unauthorized access.

Similar patterns are also used in the industry. Google's project zero team regularly analyses and finds critical vulnerabilities in commonly used software. Their standard practice is to inform the vendor and work with them to help fix the vulnerability. However, after a hard deadline of 90 days, the details of the vulnerability will be released to the public [105]. The effects are two-fold. First, the deadline encourages faster patch development by the vendor. Second, the techniques used can be studied to prevent similar vulnerabilities in the future.

B A note on licensing

During the collection of our data set, we ran into an interesting question to which we could not find a satisfying answer. We generated samples that are intrinsically designed to be as close as possible to the original data set. So, when distributing these samples (or the models that generated them), it is unclear whether the original license still applies. The data is obviously not the original data. Yet, it sounds remarkably like it. To the best of our knowledge, this question has not been addressed by the machine learning or legal community.

C Training details

We trained GMMs using gradient descent for ten epochs, with a batch size of 128, minimizing the negative log-likelihood of the data distribution. We use 128 mixture components and learn the diagonal covariance matrix of each distribution. We double the number of components to 256 for the leave-one-out experiments to compensate for the more difficult task. When training RawNet2 models we use the model configuration proposed by Tak et al. [96]. We minimize the binary cross-entropy using gradient descent, a batch size of 128, and training for ten epochs. During training we measure the validation accuracy over a hold-out set and restore the best performing model at the end of the

Table 6: **Equal Error Rate (EER) of the baseline classifier on different subset (MFCC).** We train a new GMM model for each training set and use the log-likelihood ratio to score every sample. For each data set we compute the EER, best possible result is 0.0, worst is 1.0, the lower the better. Additionally, we compute the average EER (aEER) over all sets.

	LJSPEECH								JSUT		
Training Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG	TTS	MB-MelGAN	PWG	**aEER**
MelGAN	0.332	**0.309**	0.476	0.439	0.458	0.513	0.388	0.143	0.077	0.074	0.341
MelGAN (L)	**0.295**	0.177	0.437	0.440	0.447	0.515	0.358	**0.092**	0.146	0.176	0.332
MB-MelGAN	0.481	0.466	0.025	0.371	0.318	**0.069**	0.144	0.346	0.184	0.259	0.257
FB-MelGAN	0.434	0.423	0.313	0.270	0.351	0.324	0.281	0.340	0.405	0.434	0.360
HiFi-GAN	0.468	0.458	0.313	0.386	0.252	0.288	0.256	0.285	0.225	0.253	0.322
PWG	0.503	0.508	**0.092**	0.417	0.359	0.014	**0.190**	0.427	**0.035**	**0.053**	0.241
WaveGlow	0.437	0.421	0.120	**0.334**	**0.277**	0.112	0.053	0.194	0.067	0.105	**0.214**

When the distribution is part of the training set we highlight it in gray. For other results, we highlight the best results per column in **bold**.

training. We use the Adam [40] optimizer with an initial learning rate of 0.001 when training GMM models and 0.0001 when training RawNet2. Additionally, we utilize weight decay (0.0001) when training RawNet2, following Tak et al. [96].

We resample all audio files to 16kHz and remove silence parts that are longer than two seconds. When converting the audio files to MFCC/LFCC features, we use the parameters proposed by Sahidullah et al. [83]. We extract 20 LFCC/MFCC features and compute delta-/double-delta-features, cf. Section 2. When training directly on raw audio, we also resample and remove silence from the audio. Otherwise, we follow Tak et al. [96] and either pad or trim the data to 4s.

We trained all our models on a machine running Ubuntu 18.04.5 LTS, with a AMD Ryzen 7 3700X 8-Core Processor, a GeForce RTX 2080Ti, and 64GB of RAM. The implementation of our models was performed in PyTorch 1.8.1, using the torchaudio extension in version 0.8.1 [67]. Training a model for ten epochs on 10,000 audio samples takes roughly half an hour. We do not implement the RawNet2 models but instead utilize an open-source version provided by the authors [96]. The code can be found online, and we do not redistribute it.

D MFCC results

Since MFCC features are commonly used for, e.g., automatic speech recognition, we also evaluated them. However, we found them to be strictly outperformed by LFCC features. The results are display in Table 6. When comparing the overall performance, i.e., the lowest average EER (aEER), we can observe that PWG (0.241), MB-MelGAN (0.257), and, WaveGlow (0.214) serve as the best priors for the entire data set. However, they all perform significantly worse on the MelGAN, the MelGAN (L) data sets. This trend is reversed for MelGAN and MelGAN (L), where they achieve the best results on each other (0.295 and 0.309, respectively) and dropping performance on other data sets (~ 0.400; up to 0.515 on WaveGlow). FB-MelGAN does not perform particularly well on any data set.

The similarities between PWG and WaveGlow are intuitive. The WaveGlow architecture is heavily inspired by WaveNet (the generator network of PWG). Yet, the best results for both PWG (0.092) and WaveGlow (0.069) are obtained by the models trained on MB-MelGAN and FB-MelGAN. We hypothesize that the auxiliary loss forces them to generate samples more in line with WaveGlow and PWG. Surprisingly neither FB-MelGAN nor MB-MelGAN, generalize to the MelGAN (L) data or MB-MelGAN data sets, despite using similar generator architectures.

E Phone simulation results

Table 7 presents the results of the phone simulation experiment. We evaluate the models from the out-of-distribution evaluation. The columns represent the left-out-set for the corresponding model and measure the performance on our three test sets.

Table 7: **Equal Error Rate (EER) for the phone recording simulation (LFCC).** We use the models from the out-of-distribution experiments.

Test Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG
TTS	0.000	0.000	0.001	0.000	0.006	0.000	0.000
JSUT MB-MelGAN	0.001	0.002	0.003	0.001	0.003	0.001	0.000
JSUT PWG	0.003	0.002	0.002	0.002	0.003	0.003	0.001

The columns represent the left-out-set during training

Table 8: **Equal Error Rate (EER) for the phone recording simulation (RawNet2).** We use the models from the out-of-distribution experiments.

Test Set	MelGAN	MelGAN (L)	FB-MelGAN	MB-MelGAN	HiFi-GAN	WaveGlow	PWG
TTS	0.144	0.549	0.357	0.201	0.180	0.330	0.201
JSUT MB-MelGAN	0.065	0.898	0.842	0.028	0.915	0.911	0.028
JSUT PWG	0.159	0.835	0.740	0.008	0.937	0.932	0.008

The columns represent the left-out-set during training

F Spectrograms

Here we plot the spectrograms of an audio file (LJSPEECH 008-0217) for the training data and the different generative networks. Notice the differences, especially in the higher frequencies and the horizontal artifacts produced by MelGAN and WaveGlow.

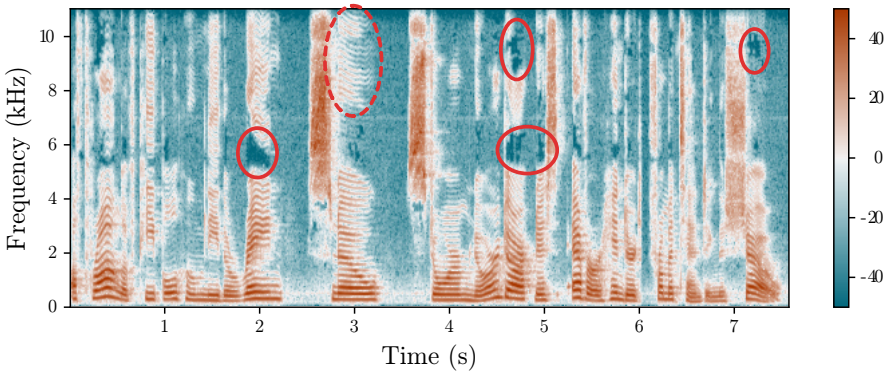


Figure 5: Original

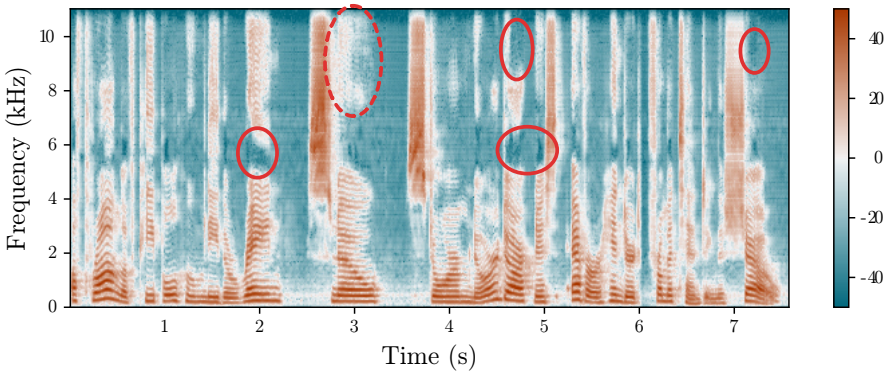


Figure 6: MelGAN

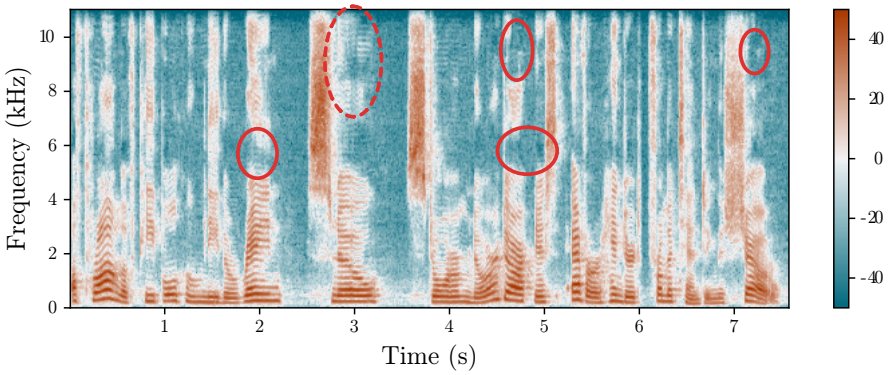


Figure 7: FB-MelGAN

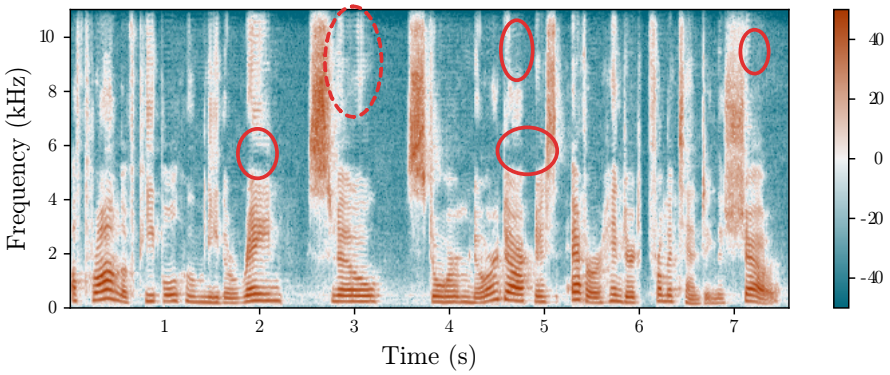


Figure 8: MB-MelGAN

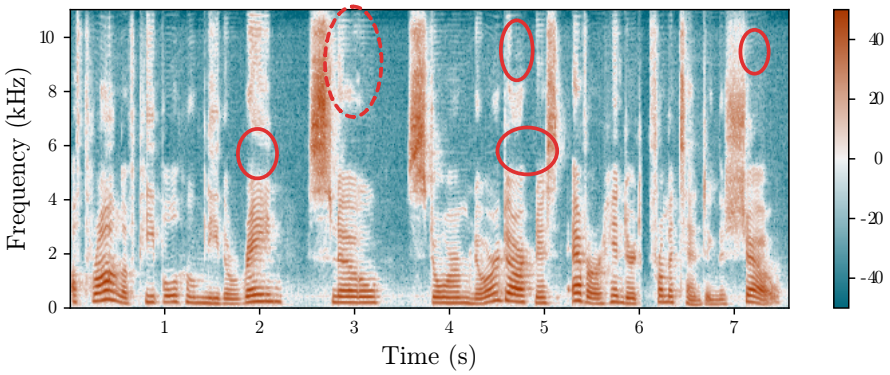


Figure 9: HiFi-GAN

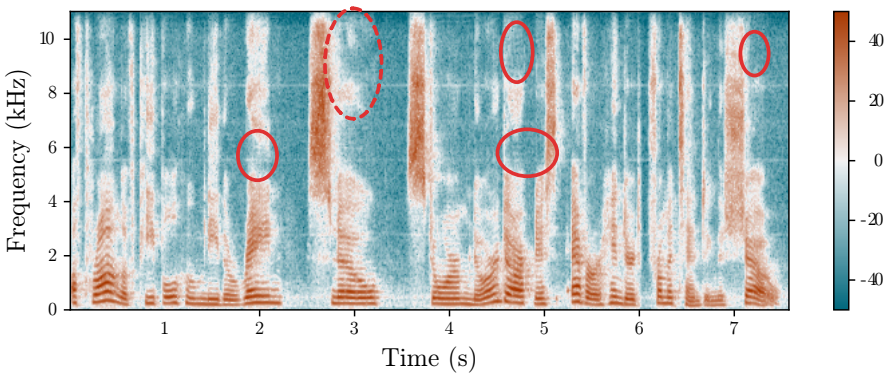


Figure 10: WaveGlow

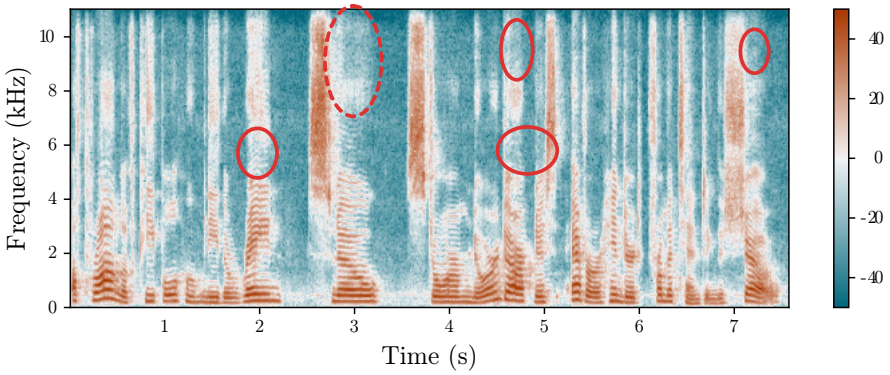


Figure 11: PWG

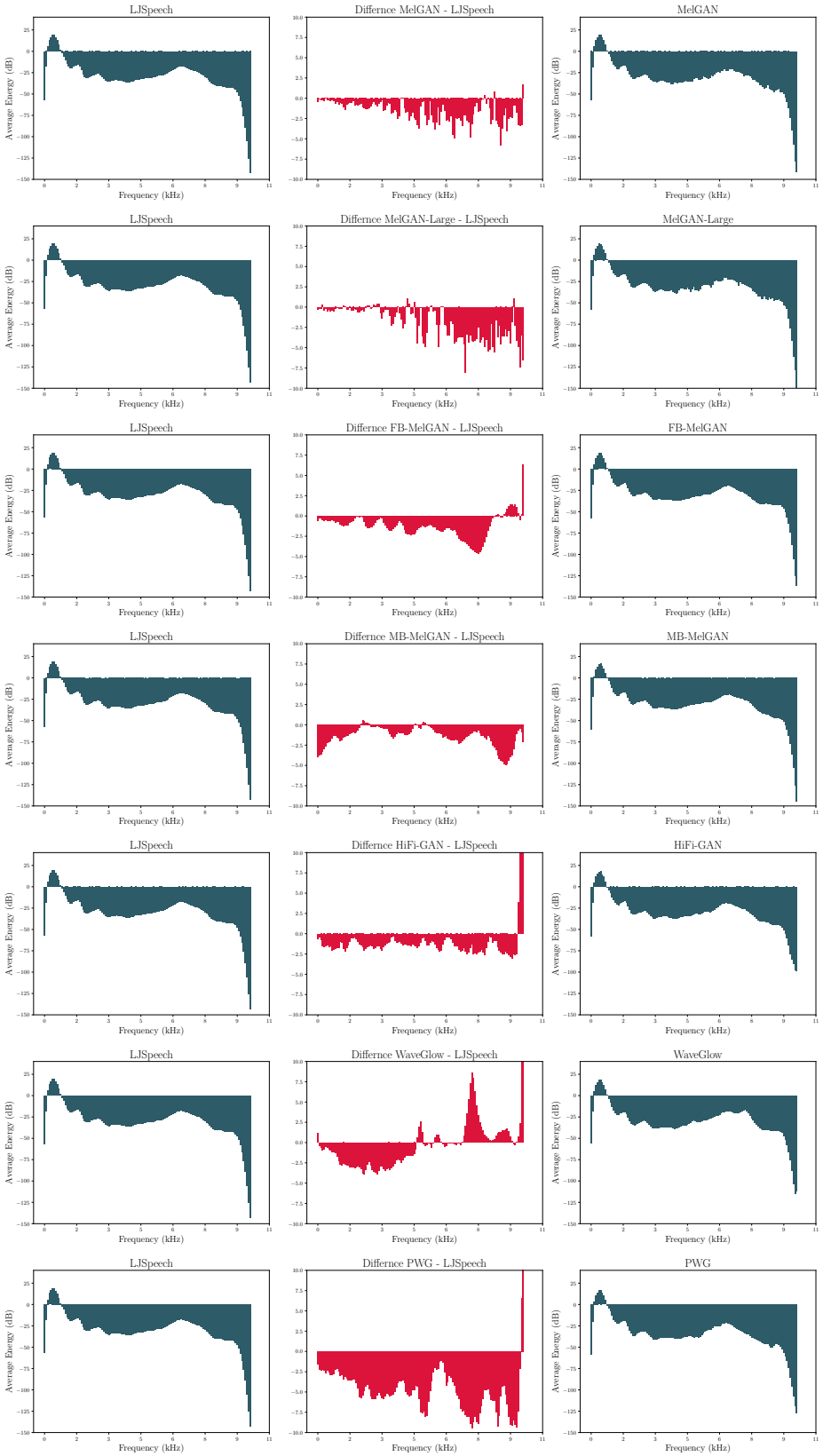
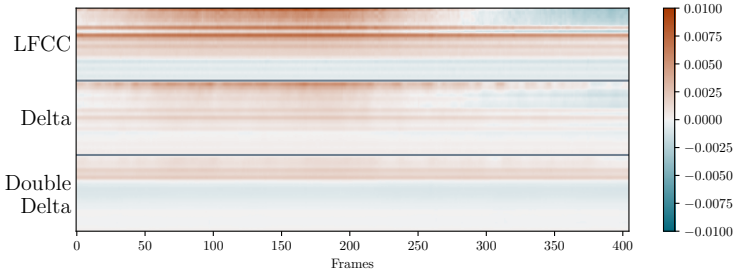


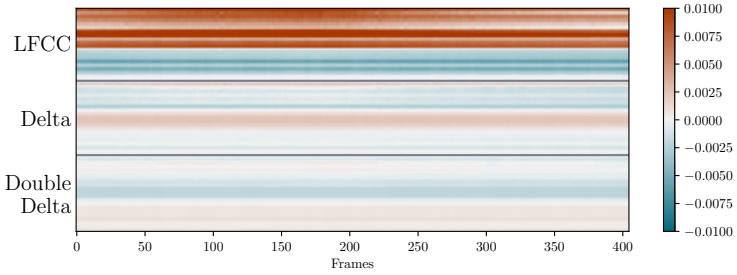
Figure 12: **Average energy per frequency bin.** We show the average energy per frequency bin in dB. Additionally, we plot the difference to the original data (LJSPEECH).

G Attribution

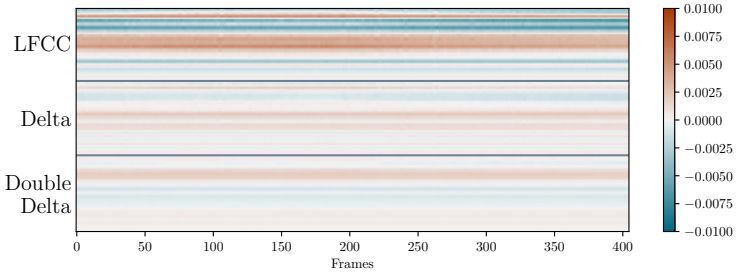
These are the full-size version of the attribution plots used in Section 4.3. Note the spread out the attention of the MelGAN classifier, the transition to narrow band attribution, and the balance of the classifier trained on FB-MelGAN.



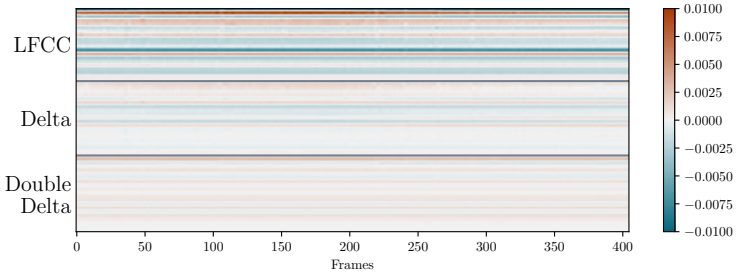
(a) MelGAN (L)



(b) FB-MelGAN



(c) MB-MelGAN



(d) PWG

H Filterbanks

Here we show a visual representation of the triangular filterbanks used to compute the MFCC and LFCC features.

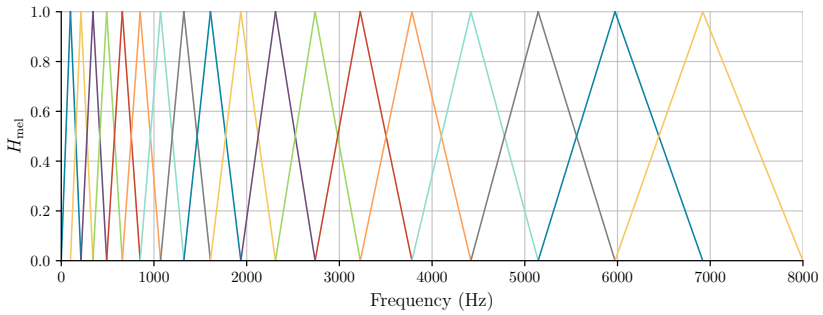


Figure 14: Mel filterbank

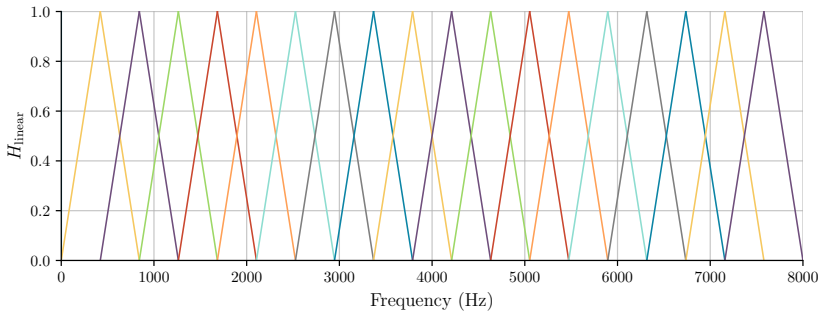


Figure 15: Linear filterbank