# Supplementary Material - 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

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

## 1 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) [11]. 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 [7]

The debate dates back to at least the 19th century where the cryptographer Auguste Kerckhoffs introduced Kerckhoffs's principle [3]. 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 [12].

A typical example is the advanced encryption standard (AES). The algorithms entire specification and inner workings can be found in the standardization [9]. 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 [1]. 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 [2], 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 [6] and Spectre [5] 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 [14]. 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.

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

Table 1: **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**.

## 2 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.

## 3 Training details

We trained Gaussian Mixture Models (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. [13]. 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 training. We use the Adam [4] 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. [13].

We resample all audio files to 16kHz and remove silence parts that are longer than two seconds. When converting the audio files to MFCC/*Linear Frequency Cepstral Coefficients* (LFCC) features, we use the parameters proposed by Sahidullah et al. [10]. 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. [13] 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 [8]. 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 [13]. The code can be found online, and we do not redistribute it.

## 4 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 1. When comparing the overall performance, i.e., the lowest average *Equal Error Rate* (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,

Table 2: **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 3: **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

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 ($\sim 0.400$; 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.

## 5 Phone simulation results

Table 2 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.

## 6 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.

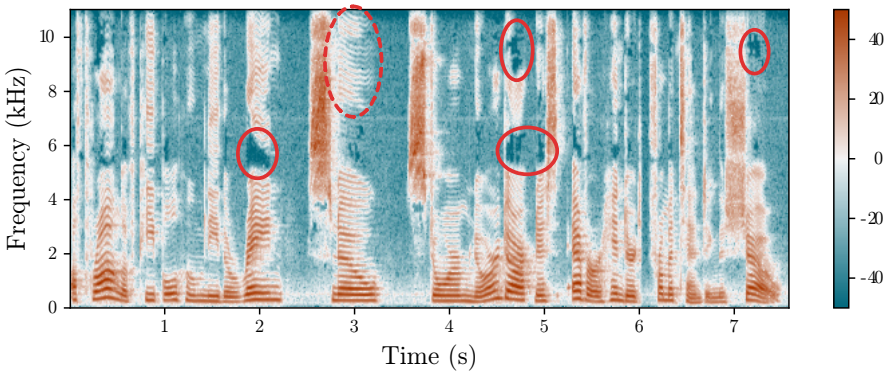



Figure 1: Original

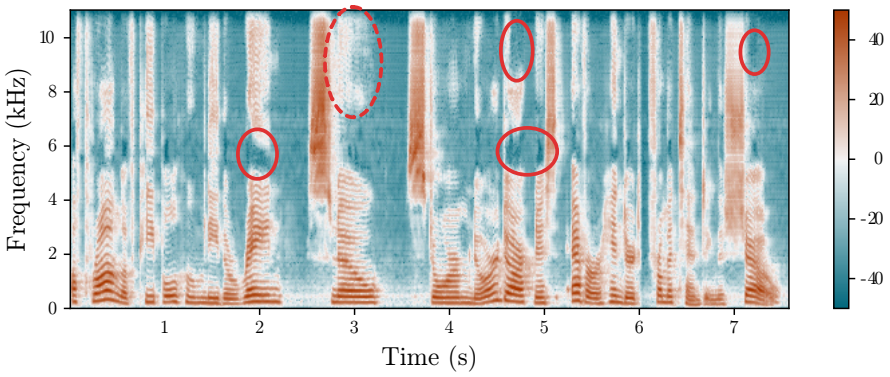



Figure 2: MelGAN

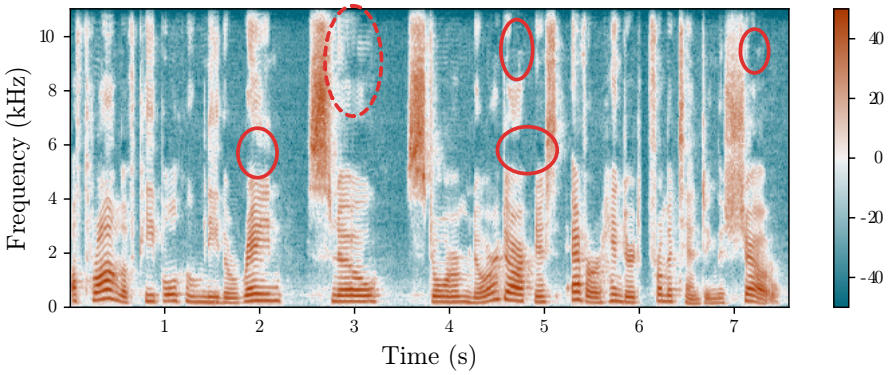



Figure 3: FB-MelGAN

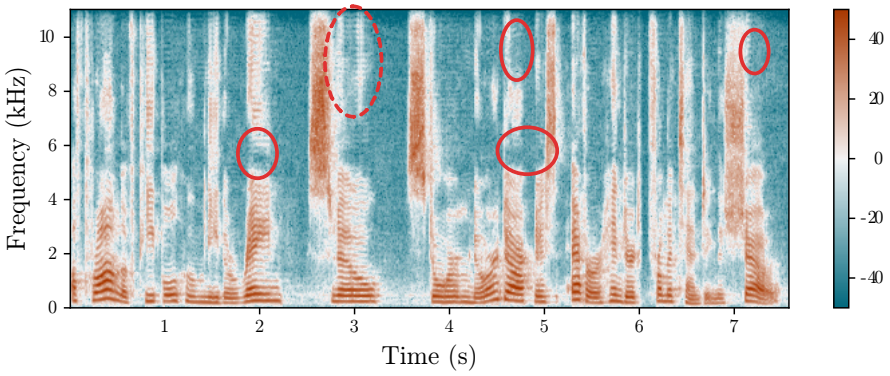



Figure 4: MB-MelGAN

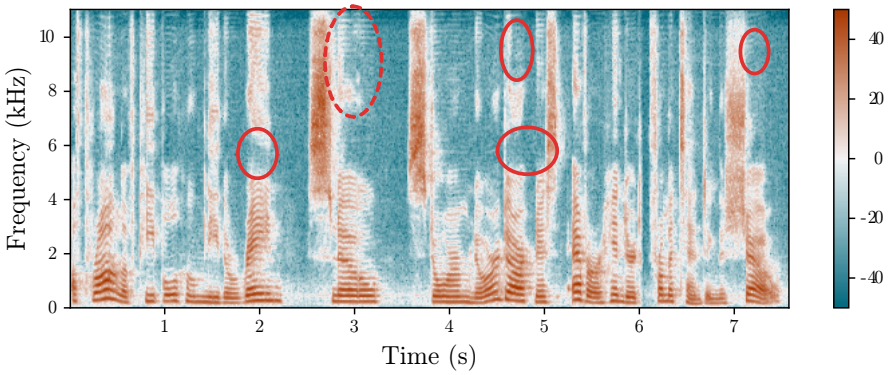



Figure 5: HiFi-GAN

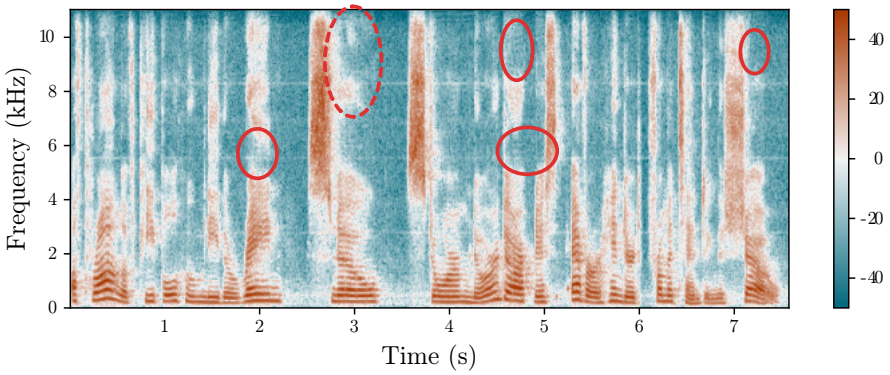



Figure 6: WaveGlow

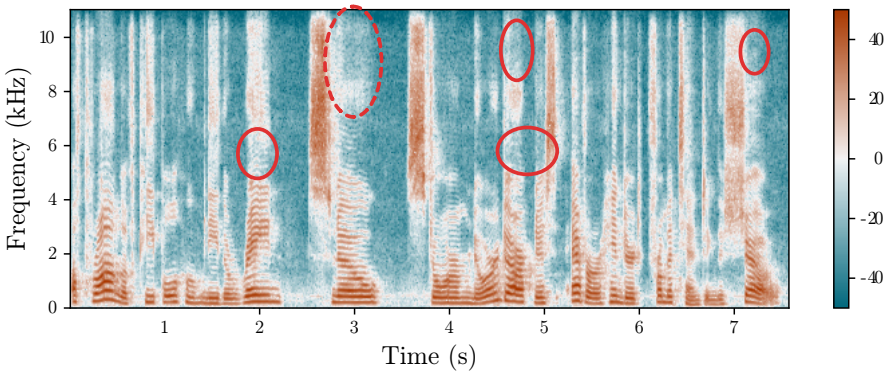



Figure 7: PWG

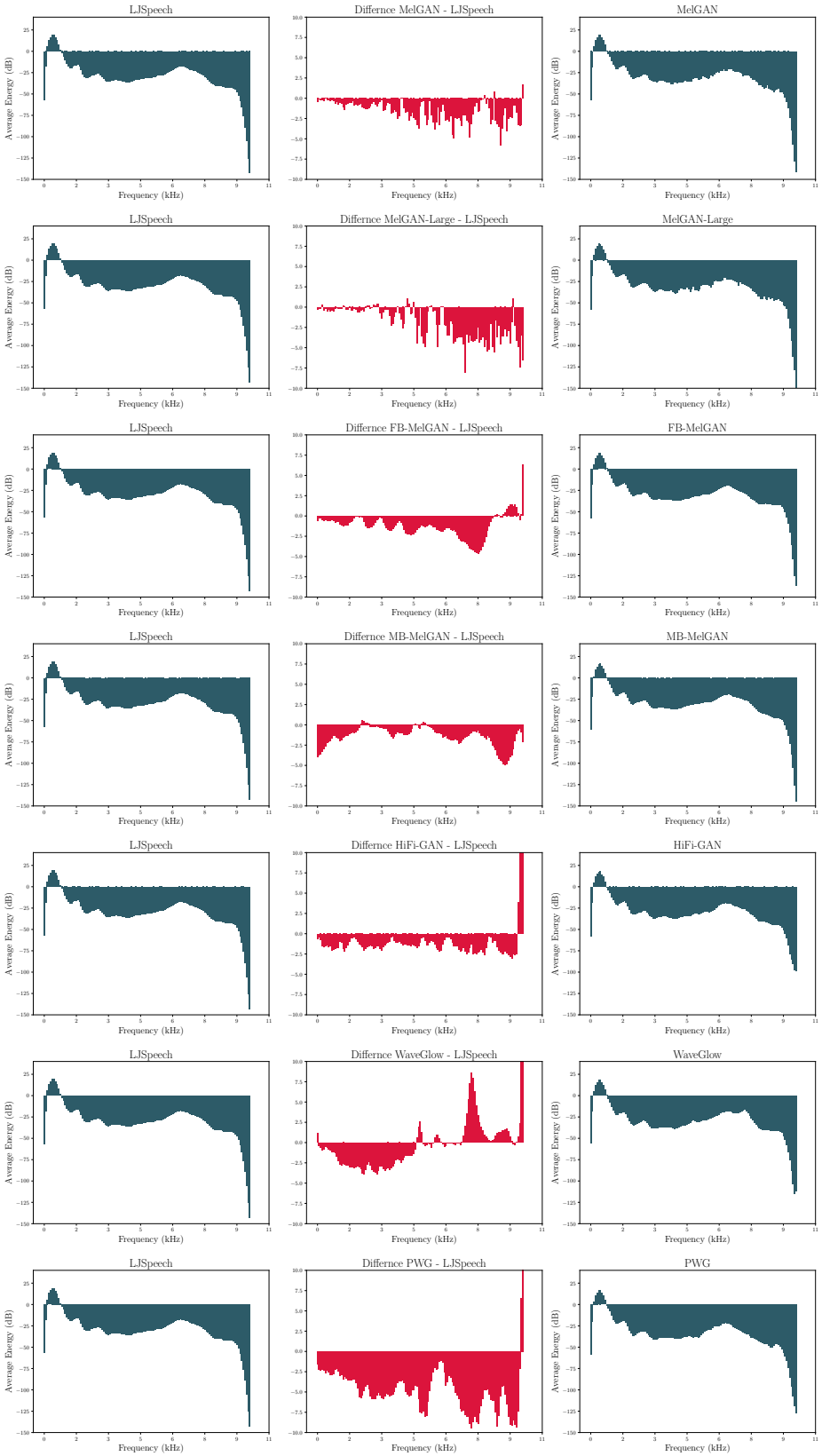
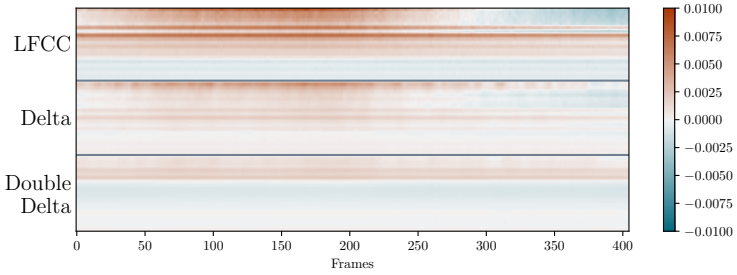


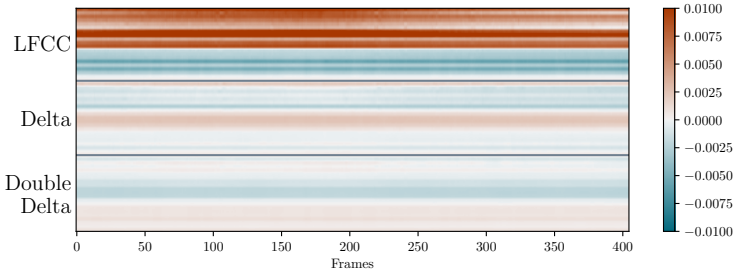

Figure 8: **Average energy per frequency bin.** We show the average energy per frequency bin in dB. Additionally, we plot the difference to the original data (LJSPEECH).

# 7 Attribution

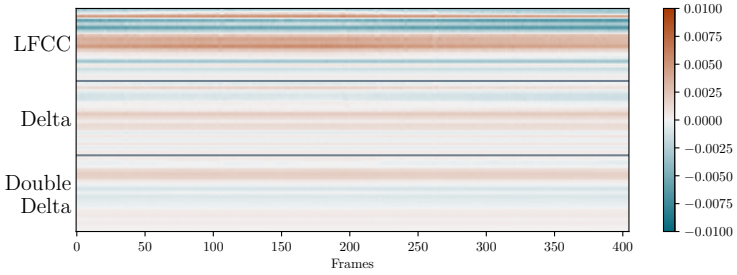
These are the full-size version of the attribution plots used in Section 4.3. Note the spread out the attention of the MelGAN classifier, the transition to narrow band attribution, and the balance of the classifier trained on FB-MelGAN.



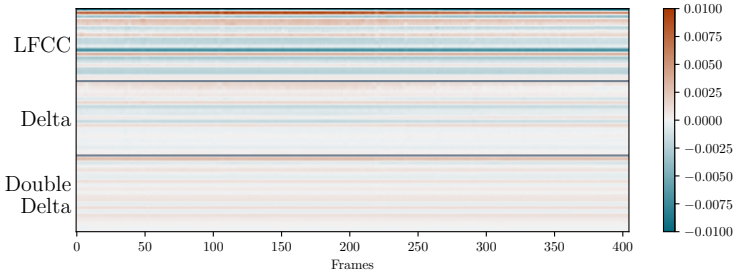

(a) MelGAN (L)




(b) FB-MelGAN




(c) MB-MelGAN




(d) PWG

## 8 Filterbanks

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

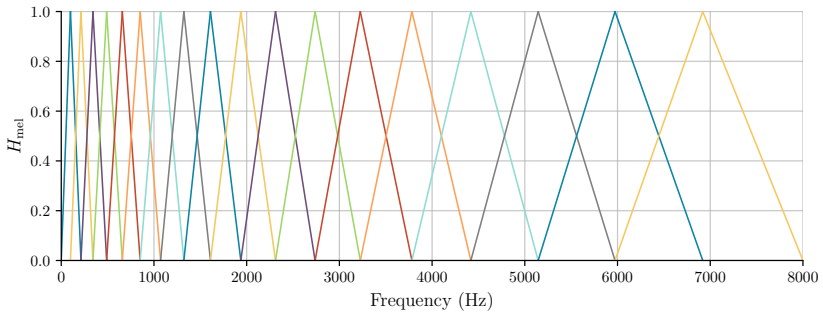



Figure 10: Mel filterbank

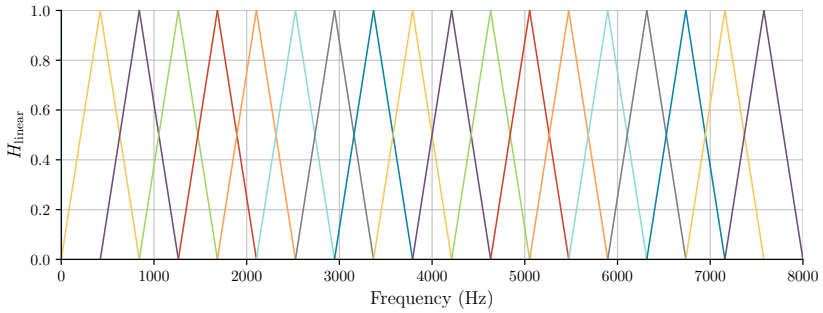



Figure 11: Linear filterbank

## References

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

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

[8] 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.

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

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

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

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

[13] 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.

[14] 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 3, 2021.