Special Assignment in Speech and Language Processing

Embedding and retrieving watermark in generated audio

Tommaso Canova
Aalto University
tommaso.canova@aalto.fi

Luca Santagata
Aalto University
luca.santagata@aalto.fi

Abstract

This study emphasizes the importance of incorporating a watermark into generated audio content to enhance security and traceability. Four distinct encoder-decoder models were explored, drawing inspiration from the U-Net architecture (5). The presented models demonstrated the capability to embed fingerprints of lengths 3, 4, 8, and 8 bits, respectively, without compromising the original audio quality. Among the models evaluated, the most successful one, achieving a bit accuracy of 100%, was the one adept at handling 3-bit fingerprints.

1 Introduction

In the realm of AI-generated audio, incorporating watermarks is of paramount importance. Acting as a distinct signature, watermarks attribute authorship, ensuring accountability and protecting intellectual property rights. As the prevalence of AI-generated content rises, these digital fingerprints become essential for tracing authenticity and preventing unauthorized use. Moreover, in an era where deepfake threats loom large, watermarks play a pivotal role in distinguishing genuine AI-generated audio from manipulated versions, safeguarding both creators and consumers. Beyond technical considerations, the inclusion of watermarks aligns with ethical imperatives, fostering transparency and responsible use in the evolving landscape of AI applications.

In the literature, numerous generative approaches exist for producing audio given various types of input, such as textual prompts (2), Mel spectrograms (3), or short audio clips. In order to enhance the security and reliability of these models, there is a desire to incorporate watermark embedding into the generative process. Many studies, in fact, solely focus on managing the embedding and retrieval of the fingerprint using a separate module, thereby making it more susceptible to malicious attacks due to the predictability of the fingerprint and increased vulnerability. This research seeks to address this vulnerability by advocating for the integration of watermark embedding directly within the generative process, aiming to fortify the security measures associated with AI-generated audio content.

This report introduces an end-to-end architecture that includes a fingerprint encoding module for raw audio (2.3) and a decoding module (2.3). Unfortunately, the idea of embedding the fingerprint during the generative process had to be abandoned, as initial experiments indicated that the proposed decoder was unable to recover the embedded fingerprint. Further details are provided in the Appendix (5).

2 Related Works

In the realm of digital images, watermarking serves to assert ownership, guarantee authenticity, and furnish supplementary information.

The current emphasis in this domain revolves around deep-learning-based watermarking, often adhering to the prevalent *END* (Encoder-NoiseLayer-Decoder) architecture. A notable contribution to this field is *De-END* (11), which introduces a groundbreaking watermarking architecture as an evolution of the established END scheme. Departing from END, De-END adopts a decoder-driven approach, addressing potential issues related to redundant features in the encoder.

In a parallel exploration of neural network capabilities, the groundbreaking *HiDDeN* (10) framework for data hiding emerges as the first end-to-end trainable architecture for both steganography and watermarking. With three dedicated convolutional networks orchestrating the process, HiDDeN seamlessly integrates an encoder, decoder, and adversary network.

In response to concerns regarding the misuse of advanced deep generative models for generating deep fakes and disseminating misinformation, Yu et al. (6) present a method that facilitates responsible disclosure by enabling model creators to fingerprint their models, ensuring precise detection and attribution of generated samples. The approach includes the efficient generation of a varied set of models with unique fingerprints, showcasing its efficacy in deep fake detection and attribution using 128-bit fingerprints.

Karras et al. (14) introduce the *StyleGAN* architecture, a cutting-edge deep-learning solution for image generation. By incorporating the novel concept of style, StyleGAN enables precise control over generated images using style vectors. The mapping of styles is currently achieved through adaptive instance normalization (*AdaIN*), effectively weighting the contribution of each style in the generated images.

Another conditioning method named *FiLM (Feature-wise Linear Modulation)*, proposed by Perez et al (7), consists in a conditioning layer used to dynamically adjust the intermediate features of an element to enhance neural networks performances in visual reasoning tasks.

Works on decoding watermark extraction have been carried on as well. One recent work presented by Fernandez et al (16) combines image watermarking and Diffusion Models to retrieve embedded binary fingerprints in generated images. Their method called Stable Signature modifies the generative networks so that the generated images have a given signature through a fixed watermark extractor. In this way a Latent Diffusion Model decoder of the image generator is fine-tuned with a binary signature provided by a watermark extractor network. After the pre-trained watermark extractor recovers the hidden signature from any generated images a statistical test is performed to determine whether the image comes from the generative model.

HiFi-GAN (3) represents a cutting-edge speech synthesis model that leverages generative adversarial networks (*GANs (4)*) to attain optimal efficiency coupled with high-fidelity audio output. Operating as a two-stage pipeline, the model first predicts a low-resolution intermediate representation from textual input and subsequently synthesizes raw waveform audio based on this intermediate representation. Notably, HiFi-GAN excels in generating raw waveforms from Mel-Spectograms, employing Residual Blocks (ResBlocks) to perform feature extraction across various temporal resolutions.

In this recent work, Juvela et al. (9) discuss advancements in neural speech synthesis, emphasizing its human-like naturalness and instant voice cloning capabilities. With the proliferation of synthetic speech content, the paper addresses the need for detection and watermarking, introducing a collaborative training scheme for synthetic speech watermarking.

A recent method to embed a watermark in audio waveform has been developed by Chen et al in (12). Their framework encodes an imperceptible 32-bit watermark within a 1-second audio snippet, offering resilience against attacks and serving as an effective identifier for synthesized voices, with potential application in audio copyright protection.

Stoller et al present *Wave-U-Net* (13), an adaptation of the well-known *U-Net* (5) model to deal with audio source separation in the time-domain. The architectures relies on upsampling and downsampling blocks to resample, combine feature maps and finally combine audio features at different time scales. In addition, the model performances have been considered comparable to the state-of-the art spectogram-based U-Net architecture given the same data.

Finally, focusing on audio features, a novel approach to deep speaker embedding using attentive statistic pooling, is introduced in (8). The method employs attention mechanism to dynamically weight statistical features (mean and variance), enhancing the model's ability to capture long-term variations in speaker characteristics more effectively.

Proposed architecture

In this section we expound upon the fundamental constituents of our proposed architectural framework, encompassing FiLM modulation, an encoder predicated on the U-Net paradigm, and a Watermark extractor. Employing a bespoke U-Net Encoder, we encode a binary watermark, colloquially referred to as a fingerprint, within a raw WAV file characterized by specific parameters, namely 16 bits per sample and a sample rate of 22 kHz. Subsequently, the generated audio facilitates the extraction of the fingerprint through the utilization of the Watermark extractor module.

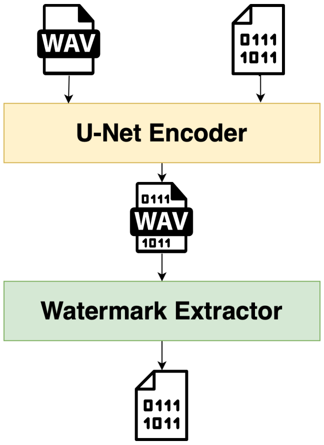

Figure 1: Watermark encoding/decoding pipeline

2.1 FiLM

Introduced as a method enhancing neural network outputs, FiLM dynamically influences intermediate features through an affine transformation based on input, as illustrated in Figure 2. This adaptive approach enables the network to selectively emphasize or suppress feature maps, enhancing performance in visual reasoning tasks. FiLM achieves this by learning functions f and h that yield γ and β as input-dependent parameters, facilitating feature-wise affine transformations for nuanced manipulation of network features.

In the method proposed by us, FiLM is strategically employed in the watermarking process to intricately condition the fingerprint based on the underlying feature maps. By incorporating FiLM into the watermarking framework, we achieve a more tailored and effective conditioning of the fingerprint, enhancing the robustness and perceptual quality of the watermark.

2.2 U-Net encoder

Taking into consideration the architectural framework proposed by (13), we present a bespoke U-Net architecture designed for the embedding of a watermark—a binary string with f bits—into a WAV audio file. The model comprises a series of downsampling and upsampling blocks, wherein feature maps at distinct temporal resolutions are integrated with feature maps of the watermark utilizing FiLM modules. The comprehensive architecture is illustrated in Figure 4.

Primarily, the audio's feature map is derived through a convolutional layer, paralleled by a similar process for the watermark. Subsequently, both outcomes are mixed via a FiLM module. A subsequent convolutional operation is applied to the module's output, maintaining an equivalent number of channels. Following this, the original watermark's feature map is regenerated using the previously described method, and both outputs are then directed back through another FiLM module. The resulting output from the second FiLM module is transmitted to the subsequent upsampling block while concurrently being concatenated with the input of the corresponding downsampling block at the corresponding level.

Upon reaching the final upsampling level, the ultimate upsampled output is passed through a Convolutional Transpose module and concatenated with the corresponding output from the preceding

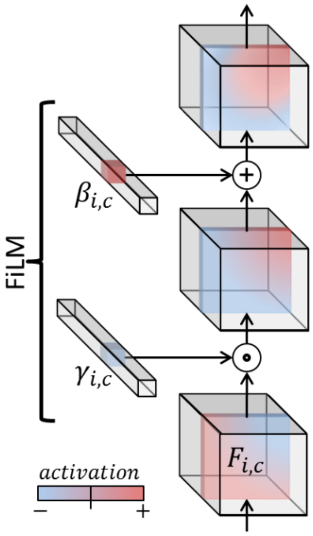

Figure 2: A single FiLM layer for a CNN. The dot signifies a Hadamard product. Various combinations of γ and β can modulate individual feature maps in a variety of ways.

upsampling block. Concerning the downsampling block, the concatenated output undergoes further processing following the same methodology elucidated in the upsampling blocks, recognizing a progressive reduction in the number of channels for the feature maps. Ultimately, following the second modulation within the last downsampling block, an additional convolutional layer is applied to recover the fingerprint. Figure 3 illustrates the fundamental structure of both an upsampling and a downsampling block.

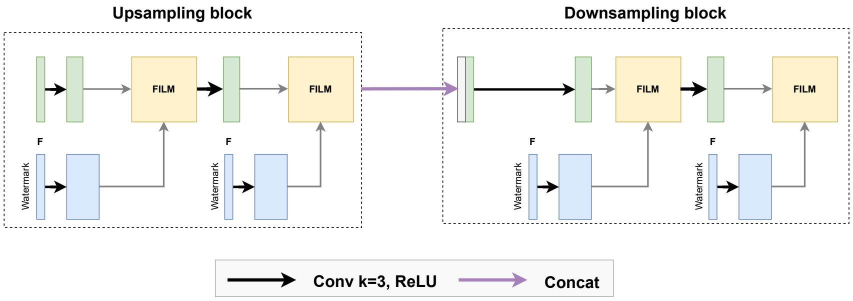

Figure 3: U-Net Encoder Upsampling and Downsampling blocks

2.3 Watermark extractor

The Watermark extractor module consists in a series of Convolutional layers in order to extract a binary fingerprint from the raw audio. First of all, the audio is upsampled to *upsampling_channels* (reffered as *u*) using a transposed 1D convolution. Following this adjustment, a sequence of 1D convolutional layers with weight normalization is applied, progressively decreasing the channel dimensions. This sequence continues until reaching the desired *fingerprint_size* (referred as *f*).

Layer	**Input Channels**	**Output Channels**	**Kernel Size**	**Dilation**	**Stride**
0	1	*u*	1	1	1
1	*u*	*u // 8*	3	2	1
2	*u // 8*	*u // 8*	5	1	1
3	*u // 8*	*u // 16*	5	4	1
4	*u // 16*	*u // 32*	5	8	1
5	*u // 32*	*u // 64*	3	16	1
6	*u // 64*	*f*	3	1	1

Table 1: Watermark extractor architecture

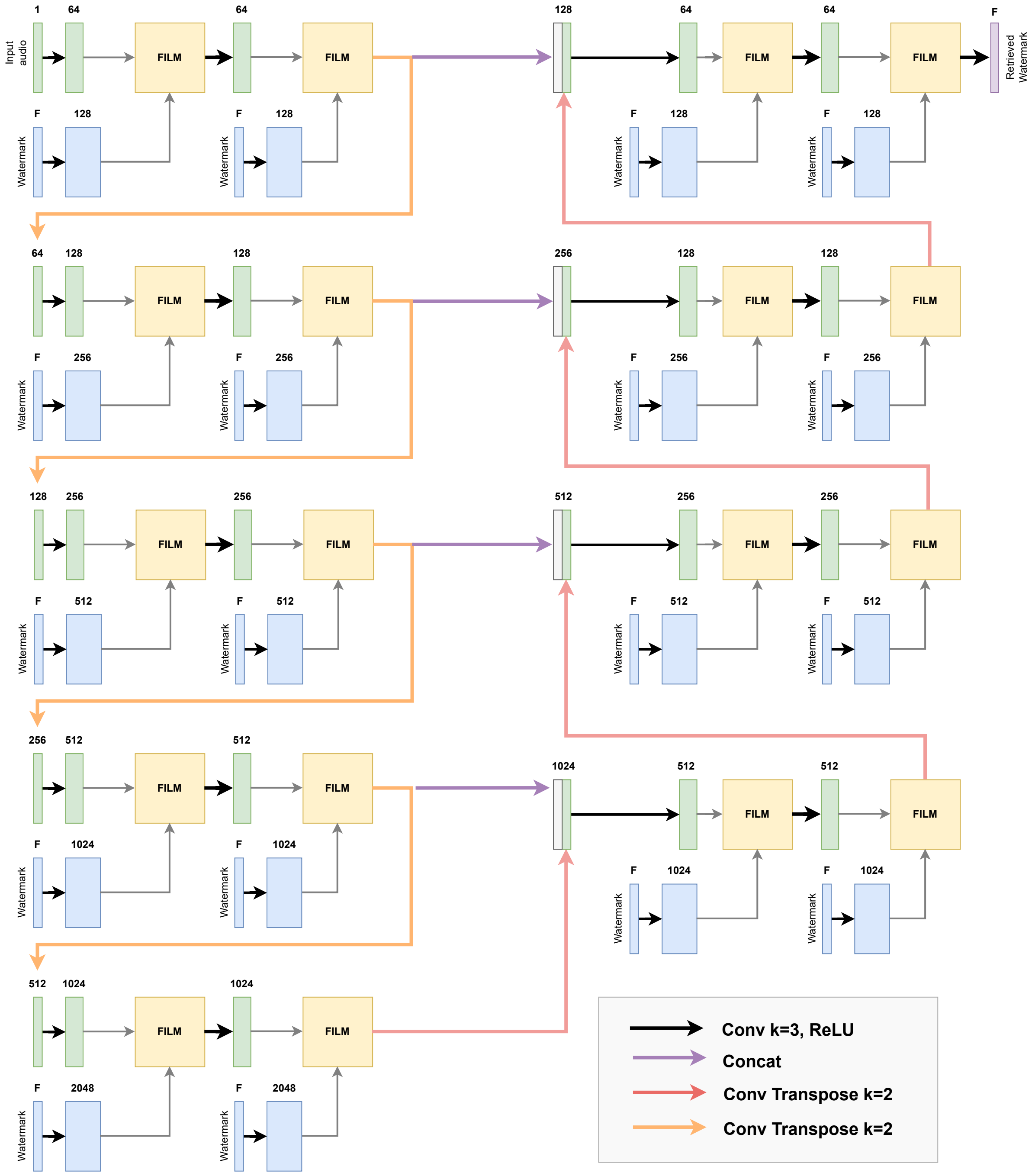

Figure 4: U-Net Encoder architecture

3 Experiments

3.1 Dataset

The model has been trained on the *LJ Speech Dataset* (17). This is a public domain speech dataset consisting of 13100 short audio clips of a single speaker reading passages from 7 non-fiction books. A transcription is provided for each clip. Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours. Each audio is in WAV format with a 16-bit sample resolution and a sample rate of 22 kHz.

3.2 Training details

To train our model, we extended the work from the Hifi-gan repository (3). We processed the dataset audio in batches of 16 items, where each audio was cut to a specific predefined length (in our case, 8192 samples). Additionally, for each audio, we extracted the corresponding Mel Spectrogram.

We utilized two loss functions to train the encoder and watermark extractor. Specifically, we employed an *L1-distance* between the Mel Spectrograms of the watermarked audio and the original audio to prevent distortions or quality loss in the fingerprint addition. The second loss was *Binary Cross Entropy* (BCE) between the original fingerprint and the one extracted by the watermark extractor. We used two ADAM optimizers, adjusting the following parameters: learning rate = 10^{-4}, weight decay = 10^{-2}, $B1 = 0.9$, $B2 = 0.999$. The two models were trained using an *NVIDIA RTX A2000 GPU* with 12 GB of memory, for 50 epochs, for an approximately total of 20.5k iterations.

We conducted various experiments by adjusting parameters related to the fingerprint length and upsampling channels, denoted as f and u, respectively. The configurations for each experiment are detailed in Table 2.

Model	***upsampling channels***	***fingerprint size***
1	256	3
2	256	4
3	256	8
4	512	8

Table 2: Experiment Configurations

4 Results

4.1 Metrics

To validate the obtained results, the following metrics were employed: Bit accuracy, Mel Spectrogram distance, and Mean Opinion Score (MOS) obtained through Perceptual Evaluation of Speech Quality (PESQ) loss.

It is noteworthy that a Mel spectrogram is a spectro-temporal representation of an audio signal, created by applying the Fast Fourier Transform (FFT) to short-time frames of the signal. The resulting spectrum is then converted into the Mel frequency scale, capturing the non-linear perceptual characteristics of human hearing.

The Perceptual Evaluation of Speech Quality (PESQ) is intricately connected to the Mean Opinion Score (MOS) in the realm of speech quality assessment. MOS represents a subjective measure obtained through human listeners who rate the overall quality of a speech signal on a numerical scale. This scale typically ranges from 1 to 5, where higher values correspond to better perceived quality. PESQ, on the other hand, is an objective measure designed to emulate human perceptual judgments of speech quality. It operates by comparing the synthesized or processed speech signal to the original, considering factors such as distortion, noise, and intelligibility. The outcome is a numerical score on a scale ranging from -0.5 to 4.5, where higher scores signify superior perceived speech quality. PESQ predicts subjective MOS scores based on the comparison of the processed speech files with the reference speech files

Each of the four models proposed in Table 2 has been assessed using the following metrics: Mel Loss, Bit Accuracy, and Perceptual Evaluation of Speech Quality (PESQ). The summarized results are presented in Table 3. Figure 5 provides a visualization of the samples within the validation set across the feature space for each model. From Model 1, it can be observed that for all samples in the validation set, the bit accuracy of the extracted fingerprint was consistently 1. This, however, is not the case for the other models, where the samples are more dispersed along the bit accuracy axis, particularly as the number of bits increases. For the other two metrics, it is evident that across all models, the results exhibit significant variance, resulting in greater sparsity in the feature space.

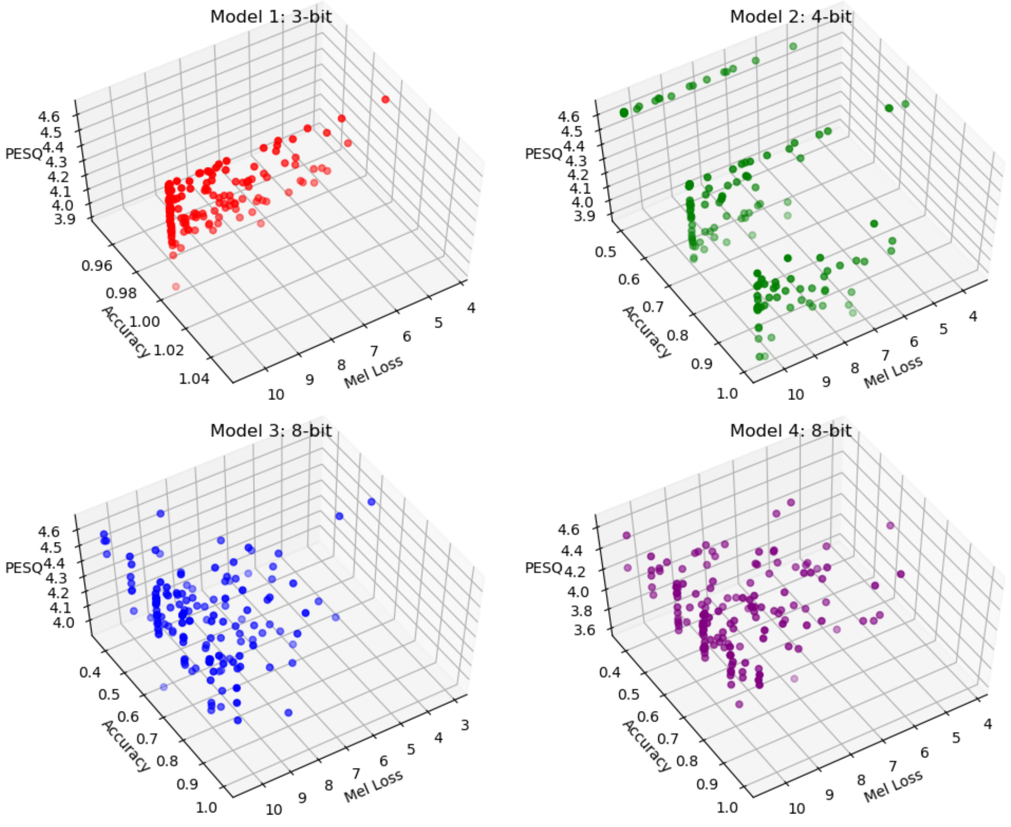

Figure 5: Models metrics compared

4.2 Mel Distance Loss

As evident from the table, the values of Mel Loss and PESQ remain relatively consistent across the various models. Notably, it is essential to observe that bit accuracy decreases with the increase in fingerprint size, in line with the complexity of the retrieval operation.

The first considered loss was computed by comparing the Mel spectrograms of the audio with fingerprint to those of the original audio, utilizing the L1 distance between the two. This representation offers a detailed portrayal of the audio signal's frequency content over time, particularly beneficial in audio and speech processing applications, serving as a basis for tasks like feature extraction and analysis.

As observed from the obtained results, the value of this loss remains relatively constant across various experiments. This stability is attributed to the fact that the architecture of the U-Net Encoder remains unaltered, in contrast to the Watermark extractor.

4.3 PESQ (MOS)

From the previously cited experiments, it is evident that the Mean Opinion Score (MOS) value predicted by the PESQ measure consistently hovers around 4.42. When juxtaposed with the MOS value scale (as reported in Table 4), this corresponds to an audio quality classification of "Good." This substantiates the effectiveness of watermark encoding within the original audio using the U-Net encoder, as it does not compromise the quality of the audio signal.

Model	Upsampling Channels	Fingerprint Size	Mel Loss (L1 loss)	Bit Accuracy (%)	PESQ (MOS)
1	256	3	9.32	100.00	4.44
2	256	4	9.31	79.83	4.41
3	256	8	9.34	68.42	4.42
4	512	8	9.32	70.92	4.42

Table 3: Models evaluation metrics

MOS	Audio quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 4: Mean opinion score (MOS)

4.4 Bit accuracy

Ultimately, the metric providing a more distinct differentiation between models is Bit Accuracy. Notably, an observable trend emerges wherein the retrieval capacity of the decoder diminishes with the escalation of fingerprint bits. The optimal model, demonstrating a noteworthy capability, manages 3-bit fingerprints with unparalleled accuracy of 100%. Noteworthy distinctions surface between Model 4 and Model 3, wherein the former exhibits marginally superior performance. This enhancement is ascribed to an augmentation in the number of upsampling channels. Specifically, in Model 3, considering 256 as the count of upsampling channels, an incongruity arises in the last convolutional layer of the decoder (Layer 6 in Table 1). Here, the input channel count falls short of the output channel count (4 vs. 8), resulting in the abatement of the "bottleneck" effect. Consequently, empirical validation indicates that the augmentation of upsampling channels in Model 4 corresponds to a nuanced amelioration in retrieval performance.

Conclusion

To address the challenge of audio-based deepfakes, a custom architecture has been proposed, encompassing a U-Net-inspired encoder and a convolutional watermark decoder.

Results from conducted experiments reveal that all proposed models successfully embedded the fingerprint into audio without compromising its quality. Hence, the employed Feature Wise Modulation-based technique for embedding the fingerprint can be considered valid and reliable. Additionally, PESQ-MOS values exceeding 4 substantiate the accurate fusion of the fingerprint with the audio, preserving its quality.

Regrettably, a constraint arises concerning the number of bits utilized for the watermark, as only the model capable of handling a 3-bit fingerprint achieved a 100 % bit accuracy.

To further enhance the model's performance, several strategies can be considered. Firstly, augmenting the number of upsampling and downsampling blocks can enrich the model's complexity. The incorporation of an attention mechanism in the decoder has the potential to improve the model's ability to focus on relevant features, thereby enhancing embedding precision. Conducting a grid search on model hyperparameters represents a systematic approach to identify the optimal combination of parameters that maximizes model performance. This methodology facilitates efficient experimentation across diverse configurations, guiding the selection of the most suitable hyperparameters. Lastly, exploring alternative loss functions, including the consideration of aligning identical audio samples with different fingerprints in the embedding space, as proposed by Yu et al (6)., could prove fruitful. This approach may contribute to optimizing the fusion of fingerprints with audio, further strengthening the model's robustness.

References

[1] Greshler, G., Shaham, T., & Michaeli, T. (2021). Catch-a-waveform: Learning to generate audio from a single short example. Advances in Neural Information Processing Systems, 34, 20916-20928.

[2] Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., ... & Adi, Y. (2022). Audiogen: Textually guided audio generation. arXiv preprint arXiv:2209.15352.

[3] Kong, J., Kim, J., & Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. Advances in Neural Information Processing Systems, 33, 17022-17033.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[6] Yu, N., Skripniuk, V., Chen, D., Davis, L., & Fritz, M. (2020). Responsible disclosure of generative models using scalable fingerprinting. arXiv preprint arXiv:2012.08726.

[7] Perez, E., Strub, F., De Vries, H., Dumoulin, V., & Courville, A. (2018, April). Film: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[8] Okabe, K., Koshinaka, T., & Shinoda, K. (2018). Attentive statistics pooling for deep speaker embedding. arXiv preprint arXiv:1803.10963.

[9] Juvela, L., & Wang, X. (2023). Collaborative Watermarking for Adversarial Speech Synthesis. arXiv preprint arXiv:2309.15224.

[10] Zhu, J., Kaplan, R., Johnson, J., & Fei-Fei, L. (2018). Hidden: Hiding data with deep networks. In Proceedings of the European conference on computer vision (ECCV) (pp. 657-672).

[11] Fang, H., Jia, Z., Qiu, Y., Zhang, J., Zhang, W., & Chang, E. C. (2022). De-END: Decoder-driven Watermarking Network. arXiv preprint arXiv:2206.13032.

[12] Chen, G., Wu, Y., Liu, S., Liu, T., Du, X., & Wei, F. (2023). WavMark: Watermarking for Audio Generation. arXiv preprint arXiv:2308.12770.

[13] Stoller, D., Ewert, S., & Dixon, S. (2018). Wave-u-net: A multi-scale neural network for end-to-end audio source separation. arXiv preprint arXiv:1806.03185.

[14] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

[15] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8110-8119).

[16] Fernandez, P., Couairon, G., Jégou, H., Douze, M., & Furon, T. (2023). The stable signature: Rooting watermarks in latent diffusion models. arXiv preprint arXiv:2303.15435.

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

5 Apendix

- As an initial approach, we attempted to integrate the FiLM method with the HiFi Gan architecture to generate watermarked audio. For watermark decoding, we initially used a simple Multi-Layer Perceptron (MLP) and later a series of convolutional neural networks. Despite the embedding not causing distortions in audio quality, the fingerprint retrieval process did not yield satisfactory results, even in the simplest case with 3 bits. We observed a stagnation in the reconstruction loss of the fingerprint from the early training iterations. Furthermore, it was not possible to confirm whether the high quality of the generated audio was a result of the model, due to loss stagnation, focusing solely on audio generation rather than the actual embedding of the fingerprint.
- Due to the complications with the above generative model, we changed the paradigm by no longer using the Mel Spectrogram of the audio as input but rather the audio itself. In this case, we employed an MLP architecture for both the encoder and the decoder. However, the main limitation of this model is attributed to the fact that this architecture only accepts inputs of fixed length, necessitating operations such as cropping or padding each time.
- From the experiments, it emerged that there is no difference in the training phase between using a unique fingerprint for each element of the batch and using the same fingerprint for all elements of the batch.
- Subsequently, for the fingerprint embedding, we utilized a simple concatenation rather than the FiLM technique, noting promising results in the retrieval process. For this reason, we approached the combination of the FiLM method with the architecture of the U-Net, where concatenation proves to be a key step.
- The proposed architecture 1 is the one from which we obtained better results. Other architectures, with more or less similar outcomes, have been proposed and are discussed in the Tables 5, 6, 7.

Amount of project credits

The project was allocated a total of **170 hours**, corresponding to a total of **6 credits (ECTS)**.

Layer	**Input Channels**	**Output Channels**	**Kernel Size**	**Dilation**	**Stride**
0	1	*u*	1	1	1
1	*u*	*u // 4*	3	2	1
2	*u // 4*	*u // 8*	5	2	1
3	*u // 8*	*u // 16*	5	4	1
4	*u // 16*	*u // 32*	5	8	1
5	*u // 32*	*u // 64*	3	16	1
6	*u // 64*	*f*	3	1	1

Table 5: First alternative Watermark extractor architecture

Layer	**Input Channels**	**Output Channels**	**Kernel Size**	**Dilation**	**Stride**
0	1	*u*	1	1	1
1	*u*	*u // 8*	3	2	1
2	*u // 8*	*u // 8*	5	1	1
3	*u // 8*	*u // 16*	5	4	1
4	*u // 16*	*u // 32*	5	8	1
5	*u // 32*	*u // 64*	3	16	1
6	*u // 64*	*f*	3	1	1

Table 6: Second alternative Watermark extractor architecture

Layer	**Input Channels**	**Output Channels**	**Kernel Size**	**Dilation**	**Stride**
0	1	*u*	5	1	1
1	*u*	*u // 8*	5	2	2
2	*u//8*	*u // 8*	3	1	1
3	*u // 8*	*u // 16*	5	2	2
4	*u // 16*	*u // 16*	3	1	1
5	*u // 16*	*u // 32*	5	8	2
6	*u // 32*	*u // 32*	3	1	1
7	*u // 32*	*u // 64*	5	16	2
8	*u // 64*	*u // 64*	3	1	1
9	*u // 64*	*f*	3	1	1

Table 7: Third alternative Watermark extractor architecture