Appendix for WAKE: Watermarking Audio with Key Enrichment

Yaoxun Xu[1], *Jianwei Yu*[2], *Hangting Chen*[2], *Zhiyong Wu*[1,3,*],
Xixin Wu[3], *Dong Yu*[2], *Rongzhi Gu*[2], *Yi Luo*[2]

[1]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]Tencent AI Lab, China
[3]The Chinese University of Hong Kong, Hong Kong SAR, China
xuyx22@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn

In the appendix, we provide a more detailed analysis of certain aspects of WAKE, as outlined below:

- 1. We discuss the importance of multiple watermark embedding and the low computational demands of our approach.
- 2. We provide a comprehensive description of the training dataset used for WAKE.
- 3. We conduct a thorough analysis of the evaluation metrics for WAKE.
- 4. We analyze the trade-off between audio quality and watermark decoding accuracy.
- 5. We examine the impact of the Predict Module on WAKE's performance.
- 6. We present a detailed analysis of audio editing operations.
- 7. We investigate the decoding of incomplete watermark audio segments by WAKE.
- 8. We analyze WAKE's performance in scenarios with multiple watermarks.

1. Discussion

1.1. The importance of multiple watermark embedding

The practice of embedding multiple watermarks holds special significance and finds extensive applications in everyday life. Firstly, multiple watermark embedding can enhance the standardization of audio processing. For instance, in an audio processing pipeline, an audio file may need to be handled by multiple individuals. To ensure that each processing step is standardized, each person needs to embed their own watermark into the audio to mark the processing stages. This allows for the identification of specific processing steps in case any issues arise. Traditional audio watermarking models, which only support the embedding and extraction of a single watermark, would be inadequate for achieving this goal.

Moreover, multiple watermark embedding can improve the robustness of audio watermarking systems. In scenarios involving watermark attacks and defenses, when an audio provider embeds a watermark to demonstrate copyright, malicious actors might inject disruptive watermarks. Since previous models only support single watermark embedding, only the last embedded watermark can be extracted, rendering the original watermark undetectable. To protect the initial watermark from being maliciously tampered with, the ability to extract specific watermarks after multiple embeddings is crucial.

Additionally, multiple watermark embedding enhances the scalability of audio watermarking systems. It allows for the embedding of watermarks of varying lengths within a single system. Traditional methods require training multiple models for different watermark lengths, which increases system complexity and maintenance difficulty. By enabling the control of multiple watermark embeddings and extractions within a single model, the system can dynamically adjust the watermark length, effectively replacing the need for multiple models. Although intuitively, only integer multiples of watermark lengths can be embedded, techniques such as controlling start and stop symbols can achieve the embedding of watermarks of arbitrary lengths.

Finally, multiple watermark embedding reduces the complexity of exploring audio watermarking systems. By gradually increasing the number of embedded watermarks, it becomes easier to explore the capacity and methods of watermark embedding. Through reasonable modeling of watermark embedding and extraction methods, the capacity of watermarks can be significantly increased.

In summary, multiple watermark embedding not only enhances the standardization and security of audio processing but also improves the flexibility and scalability of watermarking systems.

1.2. The low computational demands

The primary function of audio watermarking systems is not to enhance the auditory quality of audio but to serve as tools for anti-counterfeiting and privacy protection. Given the specialized purpose and the need for universal applicability of these systems, it is crucial to design models that are as compact and computationally efficient as possible, enabling the embedding and extraction of watermarks even under minimal computational resources.

Watermarking systems typically comprise two independent components: embedding and extraction. Traditional systems like AudioSeal utilize separate models for each function, which often leads to substantial computational inefficiencies due to the large number of parameters and the complexity of the neural networks involved. In contrast, our proposed system, WAKE, employs a reversible network architecture that utilizes a single model for both embedding and extraction processes. This integration significantly reduces the total parameter count and the computational resources required, with the parameter count amounting to only 11.77% of that required by AudioSeal.

Moreover, WAKE's approach to key management does not involve adding separate modules, which typically increase both the parameter count and the computational complexity. Instead, we employ a straightforward dot product operation to integrate the key directly into the watermark embedding and extraction processes. This method not only facilitates precise control over each digit of the key, enhancing the differentiation and feasibility between numbers, but also maintains system efficiency by

* Corresponding author.

Table 1: *Impact of the ratio between perceptual and accuracy constraints on watermark performance. A higher w_{t2} indicates WAKE's emphasis on decoding ability.*

w_{t2}	Single Watermark			Double Watermark					
	$SNR\uparrow$	$PESQ\uparrow$	$BER_1^1\downarrow$	$SNR\uparrow$	$PESQ\uparrow$	$BER_1^1\downarrow$	$BER_2^2\downarrow$	BER_3^1	BER_3^2
1	**42.932**	**4.454**	2.55	**41.859**	**4.435**	6.03	15.24	44.11	45.21
10 (WAKE)	41.192	4.397	0.13	38.482	4.339	1.26	2.71	42.44	41.59
100	37.86	4.271	0.06	36.885	4.226	1.08	2.53	45.88	40.00
1000	33.308	3.93	0.03	31.669	3.823	1.02	2.29	39.92	33.82
10000	25.678	3.218	**0.02**	23.463	3.059	**0.77**	**1.75**	37.24	33.10

avoiding additional computational complexity. Thus, WAKE effectively embeds the key **without increasing computational demands**, aligning with our goals of efficiency and minimalism in design.

2. Dataset

Referring to WavMark, to fully consider the diversity of audio types, we have chosen four datasets: LibriSpeech, CommonVoice, FMA, and AudioSet, representing English speech, multilingual speech, music, and acoustic events, respectively. Specifically, we used the entire LibriSpeech training dataset totaling 961.05 hours, a portion of the CommonVoice training dataset amounting to 849.99 hours, the FMA dataset of 271.21 hours, and a subset of AudioSet with 1447.44 hours. Our combined dataset amounts to 3529.69 hours.

3. Evaluation metrics

3.1. Perceptual quality

We evaluate the perceptual quality of the watermarked audio using two evaluation metrics: Perceptual Evaluation of Speech Quality (PESQ) and Signal-to-Noise Ratio (SNR). Specifically, we have:

PESQ is an objective method for evaluating speech quality by comparing the original audio x with the watermarked audio x_{wm}. It provides a score in the range of -0.5 to 4.5, where higher values indicate better quality.

SNR is a measure of the quality of a signal by comparing the level of the desired signal to the level of background noise. A higher SNR indicates a cleaner signal. The SNR can be calculated as follows:

$$\text{SNR} = 10\log_{10}\left(\frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N}(x_i - x_{wm_i})^2}\right) \tag{1}$$

where N is the total number of samples in the audio signals.

3.2. Decode accuracy

To evaluate the similarity between the decoded watermark wm_{re} and the original watermark wm, we use the Bit Error Rate (BER). The BER is the ratio of the number of incorrect bits to the total number of bits transmitted. It can be calculated as follows:

$$\text{BER} = \frac{\sum_{i=1}^{M} I(wm_{re_i} \neq wm_i)}{M} \tag{2}$$

where M is the total number of bits in the watermark and $I(\cdot)$ is the indicator function that equals 1 if the condition inside the parentheses is true and 0 otherwise.

4. Audio editing operation

Referring to AudioSeal, we have included various types of audio editing operations in our experiments, as shown below:

- Up-Down Sampling (UD): The audio with a sample rate of 16,000 is first upsampled to 32,000 and then downsampled back to 16,000.
- Random Noise (RN): Gaussian noise with a standard deviation of 0.01 is added to the audio to introduce randomness.
- Pink Noise (PN): Pink noise with a standard deviation of 0.01 is incorporated into the audio to simulate the presence of background noise.
- Low-pass Filters (LF): A low-pass filter is applied to the audio, attenuating frequencies above 5,000 Hz.
- High-pass Filters (HF): A high-pass filter is applied to the audio, attenuating frequencies below 500 Hz.
- Band-pass Filters (BF): A band-pass filter is applied to the audio, selecting only the frequency range between 300 and 8,000 Hz.
- Boost Audio (BA): The intensity of the audio is increased by 20% to enhance its loudness.
- Duck Audio (DA): The intensity of the audio is decreased by 20% to reduce its loudness.
- Shush Attacks (SA): The input audio tensor is modified by setting a fraction of its indices to 0, with a proportion of 0.001.

5. Balance between audio quality and decoding accuracy

Audio quality and accurate watermark decoding are both crucial for audio watermarking models. To balance these aspects, we adjust the accuracy constraint weight w_{t2} in the training loss while keeping the perceptual constraint weight w_{t1} fixed. Results are shown in Table 1.

The experiments show that as w_{t2} increases, both SNR and PESQ decrease, reducing audio quality. Conversely, the BER for single and double watermarked scenarios drops respectively, indicating improved decoding capability. Pursuing better decoding performance leads to lower sound quality, and vice versa. Therefore, we choose w_{t2}=10 for the optimal balance between audio quality and decoding capability.

6. Ablation study of the Predict Module

In this study, we examine the Predict Module's impact on WAKE, with results shown in Table 2.

The results show that removing the Predict Module significantly weakens WAKE's decoding capability, leading to a notable increase in BER for both single and double-watermark scenarios. Using the Predict Module's predicted representation,

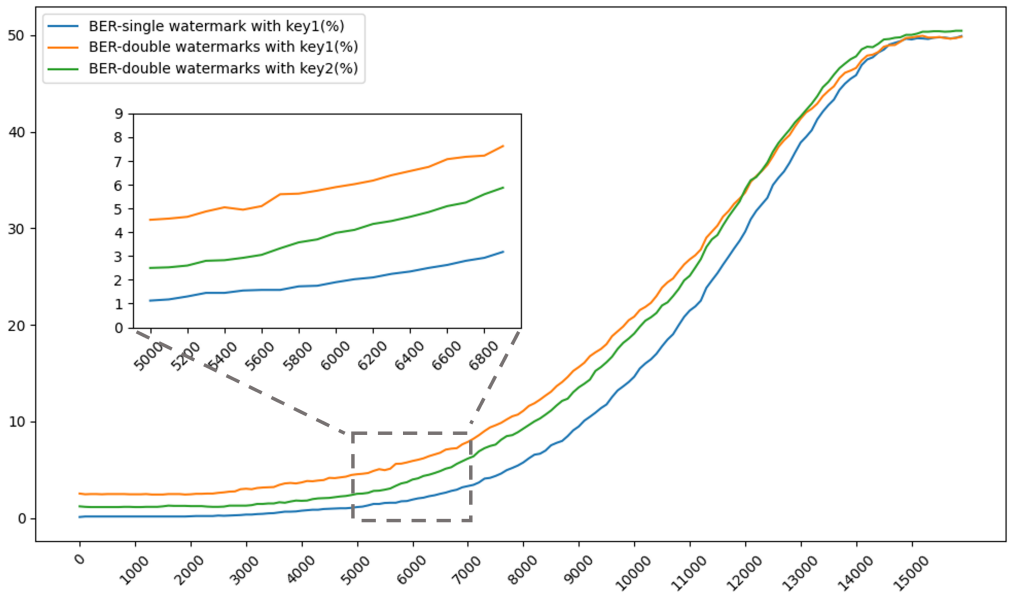

Figure 1: *The BER of watermarked audio replacing original audio with different lengths*

Table 2: *Ablation study on the effect of the Predict Module (PM) on model performance.*

	Single Watermark			Double Watermark			
	$SNR\uparrow$	$PESQ\uparrow$	$BER_1^1\downarrow$	$SNR\uparrow$	$PESQ\uparrow$	$BER_1^1\downarrow$	$BER_2^2\downarrow$
w/o PM	36.930	4.223	5.10	34.25	4.102	8.92	13.24
w/ PM	41.227	4.392	0.13	38.442	4.321	1.25	2.72

Table 3: *Comparison of the number of parameters between WAKE and the other two baselines*

Model		Parameters
AudioSeal	Encoder	14679906
	Decoder	8649138
WavMark		2488337
WAKE		2746227

closely related to the watermarked audio, instead of random sampling, greatly improves WAKE's decoding accuracy. Additionally, integrating the Predict Module significantly enhances the quality of watermarked audio. This improvement is likely due to WAKE's INN architecture, where encoding and decoding are invertible processes. Random Gaussian sampling during training complicates WAKE's convergence, affecting watermark encoding performance. More effective constraints during decoding provide better guidance for the encoding process, resulting in improved audio quality.

7. Robustness of incomplete watermarked audio decoding

Detecting the watermark embedded in an audio clip is essential since, in real-world situations, watermarks are not only embedded in one-second audio clips. Furthermore, the watermark model should be as lightweight as possible, functioning as a post-processing module for the audio and demanding minimal computational resources for processing. Table 3 below compares the number of parameters between AudioSeal, WavMark, and WAKE.

We observe that AudioSeal has significantly more parameters than WavMark and WAKE, as it employs a dedicated decoder to decode the watermarked audio. Even the decoder's parameter quantity surpasses WAKE's, typically necessitating more computational resources for processing.

To effectively assess WAKE's decoding performance's robustness against incomplete watermarked audio, we generate watermarked audio and replace the first n frames with varying lengths of original audio before decoding it with WAKE. Specifically, we choose the first n frames of the watermarked audio for the replacement experiment, incrementing the value of n from 0 to 16000 at intervals of 100. In each experiment, we substitute the first n frames of the watermarked audio with the original audio without the watermark. The experimental results are illustrated in Figure 1.

The experimental results indicate that, in both single and double watermark scenarios, the decoding performance deteriorates as the proportion of replaced watermarked audio increases. However, we find that as the beginning portion of the watermarked audio is gradually replaced with the original audio, the decoding performance does not change significantly, regardless of whether it is a single or double watermark embedding scenario. Replacing 6,000 frames, or 37.5%, does not have a notable impact, which is a considerable improvement compared to WavMark, which could only guarantee no significant effect when replacing 10%. This means we can use a larger sliding window for sliding detection, greatly enhancing the retrieval efficiency. After balancing efficiency and accuracy, we can choose a sliding window of 6,000 frames as the frame shift for watermark extraction from the audio.

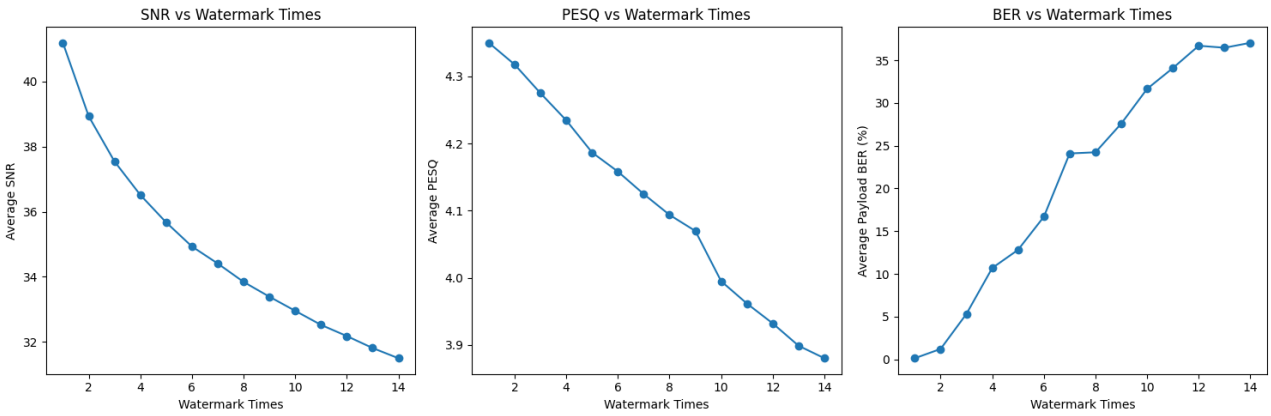

Figure 2: *Experimental results of different watermark embedding times*

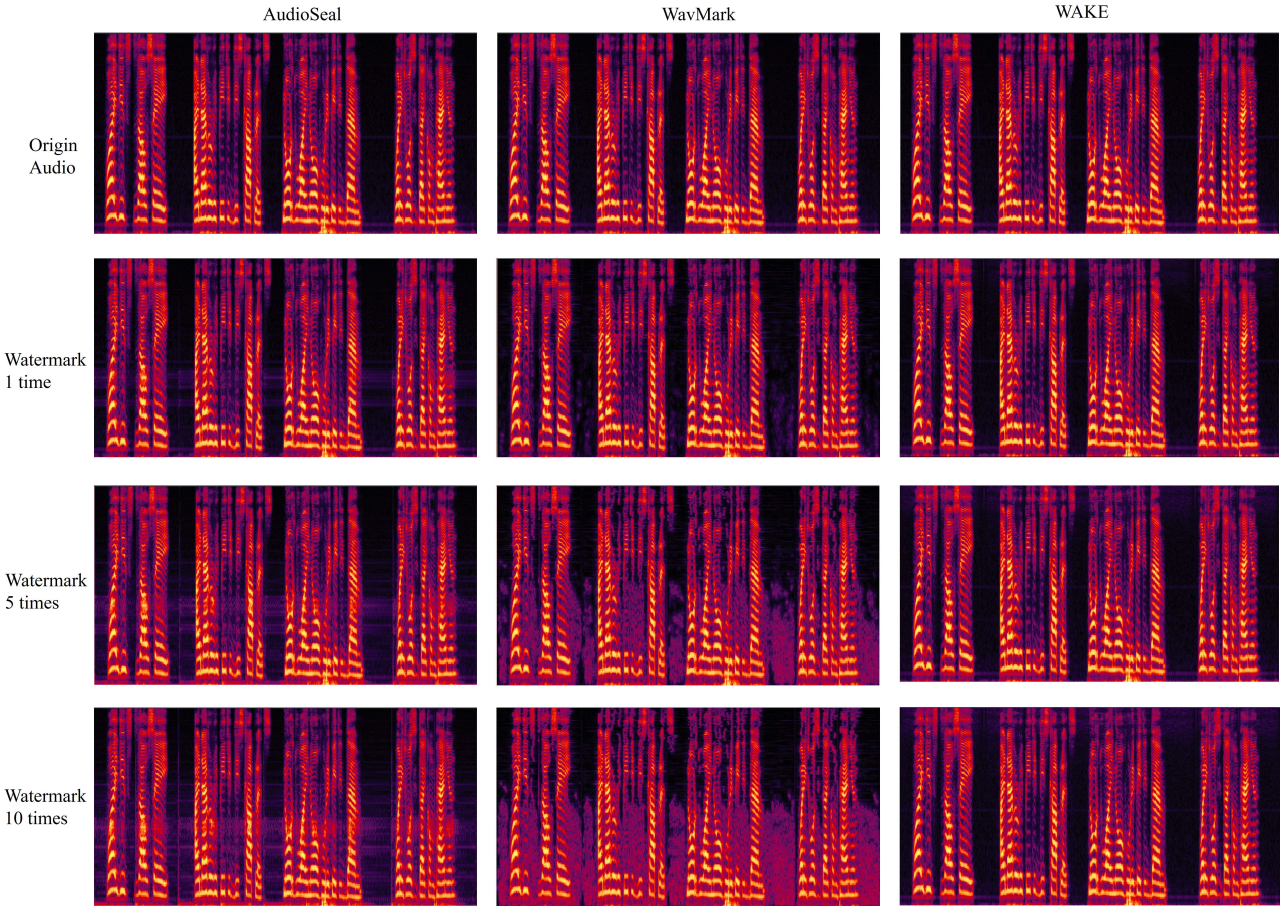

Figure 3: *Comparison of watermarked audio spectrograms with different times of watermark embeddings*

8. Multiple Watermark Embedding

To better assess WAKE's decoding performance with more than two watermarks, we conduct a series of experiments involving multiple audio watermark embeddings. In each n-times watermark embedding experiment, we first generate n distinct key-watermark pairs and then sequentially embed these watermarks into the audio using their corresponding keys, creating an audio file containing n watermarks. We then decode this audio using the key corresponding to the first embedded watermark, yielding the decoded watermark. To comprehensively evaluate the effectiveness, we compute the PESQ and SNR values between the audio with n embedded watermarks and the original audio, as well as the BER between the decoded and the first embedded watermark. We test results from embedding the watermark once up to 15 times, with results presented in Figure 2.

As shown in the figure, from the audio quality perspective,

both SNR and PESQ exhibit a downward trend as the number of embedded watermarks increases. As more watermark information is embedded, it becomes increasingly likely to interfere with the original audio and listening experience. However, we find that after multiple embeddings, SNR and PESQ still maintain optimistic values, showcasing WAKE's excellent watermark encoding capability. Regarding decoding performance, we observe that as the number of watermark embeddings increases, the decoding ability gradually declines. The more watermarks embedded, the more challenging it becomes to detect earlier watermarks. After four times of watermark embeddings, WAKE's decoding performance deteriorates significantly, making it difficult to accurately detect the watermark. Nevertheless, WAKE's watermark encoding capacity remains far greater than that of the current audio watermark models with the largest embedding capacity.

Additionally, to better showcase WAKE's performance in maintaining audio quality with multiple watermark embeddings, we use both WAKE and the two baselines to conduct experiments with multiple watermark embeddings. Specifically, we display the spectrograms of an audio sample after being embedded with 1, 5, and 10 distinct watermarks by three different models, as illustrated in Figure 3.

We can clearly observe from the figure that as the number of embedded watermarks increases, the spectrograms of the watermarked audio created by AudioSeal and WavMark diverge more significantly from the original audio spectrogram. However, the spectrogram of the watermarked audio generated by WAKE remains largely consistent with the original, demonstrating WAKE's robustness to multiple audio watermarks. Furthermore, we note distinct differences in the mid-to-low frequency range between the original audio and the watermarked audio generated by WavMark and AudioSeal. This suggests that WavMark and AudioSeal tend to embed watermarks in the mid-to-low frequency range of the audio. The differences between the original audio and the watermarked audio generated by WAKE are concentrated in the high frequencies. Since the human ear is more sensitive to mid-to-low frequencies (1000-3000Hz), the watermarked audio generated by WAKE is least perceptible to human hearing. This further highlights WAKE's powerful ability to maintain a high level of auditory imperceptibility in watermark audio.