NEURAL WATERMARKING: LACK OF A SECRET KEY IS STILL LACK OF SECURITY

Hussein Tarhini, Aurélien Noirault, Jan Butora, Patrick Bas

UMR 9189 CRIStAL
Univ. Lille, CNRS, Centrale Lille
Lille, France

ABSTRACT

Neural watermarking has been on the rise as a simple tool for marking generated multimedia content in a robust way. This simplicity, however, comes at a cost. While it is easy to enforce robustness in a training loss through various data augmentations, it is currently unknown how to include the security aspect of watermarking. Consequently, these black-box schemes are easily breakable by targeted attacks. In this work, we show how to remove a watermark from a recently proposed WAM [16] image watermarking model. Consistent with Kerckhoffs's principle, we show that the absence of a secret key, combined with the requirement for robustness, creates an exploitable weakness. This manifests as near-periodic patterns in one principal color component of the image. We show that these patterns can be accurately estimated even from a single image, and erased by manipulating a single component of the image in its Fourier representation. Furthermore, we demonstrate that the same watermark can be manually injected, yielding noticeably higher image fidelity than when using WAM for watermarking. The code used in this work will be made available upon acceptance of the paper.

Index Terms— watermarking, watermark removal, watermarking security, watermark only attack

1. INTRODUCTION

With the rapid growth of deep learning applications, concerns regarding the security and integrity of digital content have become increasingly pressing. Neural image watermarking has emerged as a promising approach for protecting intellectual property and authenticity verification. In essence, watermarking involves embedding hidden information within an image such that it remains imperceptible to the human eye while still being detectable and decodable by a model.

A watermarking scheme is typically characterized by two fundamental properties: **robustness** and **security**. While robustness has received considerable attention in the research community, little to no security evaluation is typically provided. The reason for this imbalance is simple: robustness can be assessed in a largely automated manner by applying a range of transformations to watermarked data and integrating these perturbations into the training of neural networks. However, a question remains, *how do we measure security?* While this question is all but simple, it is evident that the lack of a secret key has to lead to serious security vulnerabilities. This is hardly surprising as it directly contradicts the Kerckhoffs' principle introduced already in the 19th century.

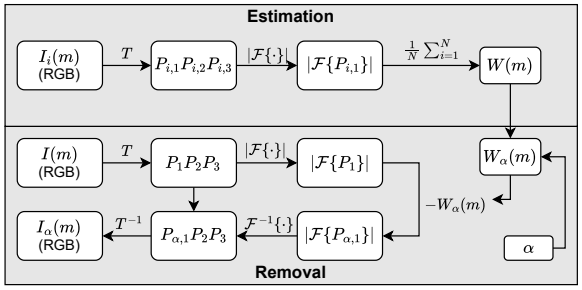

Fig. 1: The presented attack consists of first estimating the principal color component $P_{i,1}$ carrying the watermark signal, then computing the Fourier spectrum $\mathcal{F}$ to estimate peaks coding the watermark. The removal simply consists of removing the peaks from the estimated subspace and performing the inverse Fourier Transform. This attack can be conducted when the message is known or not (single-shot attack).

If several pioneering works have been published on classical watermarking security [1,3,7,8], yet many recent neural watermarking works neglect the role of secret keys. For instance, Butora et al. [5], [6] showed that Adobe's image development software embeds a keyless watermark pattern. They showed that this pattern can be effectively estimated from developing constant images and subsequently removed or injected into the image space. An audio watermarking scheme Audioseal [15], has been successfully attacked in [2] by using a naive band-stop filter between 1.0 and 1.2 kHz. Additionally, the watermark can be simulated by enhancing a single frequency at 1.1 kHz. Building on these findings, this paper turns to a neural watermarking scheme known as the Watermark Anything Model (WAM) [16].

This is motivated by the fact that even if WAM achieves state-of-the-art performance in controlled settings, its re-

silience to adversarial perturbations remains unverified. Specifically, the lack of a secret key raises critical questions regarding its security guarantees, which we aim to exploit. Establishing rigorous methodologies for evaluating the robustness and security of watermarking schemes is imperative for ensuring reliable protection of ownership and integrity in digital media.

The next section introduces the WAM model and some of its properties. Section 3 describes our experimental setup and simple attacks on the model assuming the attacker has access to the public watermarking system. In Section 4, we restrict the attacker's knowledge to a single watermarked image, and the paper is concluded in Section 5.

2. WAM

We focus on evaluating the security of WAM, which presents a novel approach to digital watermarking by redefining the task as a pixel-level segmentation problem, as opposed to traditional methods, which make a single global decision per image.

WAM is composed of two main components: a watermark embedder and a watermark extractor, trained jointly. The embedder is responsible for injecting a binary message $m \in \{0,1\}^{32}$ into an image in an imperceptible yet robust manner, while the extractor detects and decodes the embedded message at the pixel level. The embedder leverages a variational auto-encoder backbone to encode the image into a latent space, combines it with a message embedding generated through a binary lookup table, and decodes it into a watermark signal that is added to the original image. The extractor, inspired by modern segmentation architectures like SETR [17] and Segment Anything [13], uses a Vision Transformer (ViT) backbone to localize watermarked regions and decode messages on a per-pixel basis. Together, these components enable robust, localized, and invisible watermarking, adaptable to high-resolution images and complex transformations. One of the main novelties of WAM is that it can embed potentially different messages into different parts of an image and reconstruct different embedding masks. This could complicate a potential watermark removal, but as we will see in Section 3.2, it does not.

The extractor makes a two-fold prediction. On one hand, it localizes the watermarking mask indicating the watermarked areas. On the other hand, it uses the mask to decode the 32-bit embedded messages. Two functionalities are allowed:

Localization: A specific pixel is deemed watermarked if the decoder's sigmoid is at least $\tau_l = 0.5$

Detection: The whole image is detected as watermarked if at least $\tau_d = 0.07$ of all pixels are detected as watermarked.

The embedder operates on a $h \times w$ ($= 256 \times 256$) resolution, and to watermark a higher-resolution image, the image is downsampled to produce a watermark signal, which is subsequently upsampled to the image size and added to the image. On the other hand, the decoder always resamples the images to $h \times w$ to be consistent with the embedder pre-training. For this reason, all images used in this work are of this size. We used the publicly available code[1] to implement WAM.

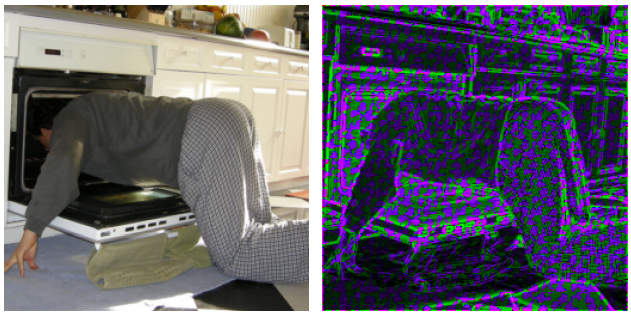

Fig. 2: Orignal image I (left) and its watermark $w(m)$ (right) - zoom in to see the periodical patterns.

3. ORACLE ATTACKS

We now describe the attacks in which the embedder and detector part of the watermarking scheme are available to the attacker. First, we will study the watermark properties and how to remove it from a given image. Next, we will show how to inject the watermark information with very little distortion.

In our experiments, we use 1000 randomly selected images from the COCO dataset [14]. All the images are resized with a bilinear kernel when entering the extractor to size $h \times w = 256 \times 256$. Along the way, we make several observations that will be important for the proposed attacks.

3.1. Security setups

A first scenario is to assume the knowledge of the embedded message m. This is trivial, as we can decode m from an image under scrutiny by simply using the WAM extractor to obtain the embedded message. It also corresponds to the *Known Message Attack* (KMA) in watermark security setups [8] where the adversary only observes contents watermarked with the same message and has no access to the embedder or the decoder.

A second scenario is to assume that the adversary does not know the embedded message. This is related to the *Watermarked contents Only Attack* (a.k.a. WOA) security setup described in [8] and it is addressed in Section 4 where we propose an attack on a single watermarked image, *i.e.* a one-shot attack.

[1] https://github.com/facebookresearch/watermark-anything

3.2. Removal attack

3.2.1. Grayscale attack

Since WAM is localizing watermarks on a pixel level, it is natural to assume that the watermark signal is encoded in the color of each pixel separately. To verify this assumption, we watermarked the 1000 selected images and converted them into grayscale with the torchvision library. As expected, the accuracy of detecting the watermarking mask dropped from 100% to 0% in all cases. Moreover, we noticed that when the extractor does not localize any mask, the decoded message is always a binary string of 32 zeros. We thus conclude that the most efficient attack removes the mask, as it consequently prevents reading any (potentially multiple) embedded secret messages. Let us now present an attack on color images.

3.2.2. Watermark color component

In order to remove the watermark from an image, we have to first understand the nature of the watermark. We randomly select an image I and note $I(m)$ its watermarked version with message m. We will denote the watermark signal as their difference $w(m) = I(m) - I$.[2] We can observe in Figure 2 several periodic patterns that emerge in the watermark. We asked ourselves if this periodicity is a global phenomenon or if it differs from image to image. To verify this, we took several pairs of watermark signals from different images and studied their cross-correlation. We indeed observed distinct peaks in the cross-correlation of all three color channels, although their locations vary for different pairs of images. This means that while the pattern is present in each image, its exact spatial placement varies slightly due to image content. Nevertheless, the periodic nature of the watermark should manifest as distinct peaks in the frequency domain. We can therefore isolate these peaks (and thus the watermarking signal) in the translation-invariant frequency domain. To avoid inspecting the spectrum of all three color channels separately, we look at the watermark's behavior across these channels.

We examined the correlation between the RGB channels for several watermarked images with the same message m, and observed that they are correlated with relative contributions approximately in the ratio $(1, -1, 2)$. To decorrelate the color channels and isolate the dominant signal, we apply Principal Component Analysis (PCA) to the watermark signals. We generated $N = 20$ random uniform images $I_i \in [0, \dots, 255]^{h \times w \times 3}$, $i \in \{1, \dots, N\}$, and computed their watermarked signals $I_i(m)$. We then applied a PCA to find an orthonormal color transformation $T \in \mathbb{R}^{3\times3}$. In this transformed color space $P_1P_2P_3$, we observed that 99.60% of the signal variance is contained in the first principal component P_1, with an associated eigenvector $T_{1,*} = (0.390, -0.409, 0.825)$. This effectively means that the watermark is entirely contained in this color component, and its removal should be possible in this component only. Embedding a watermark in a different space is quite common to hide its presence, however, the embedding space is typically conditioned on a secret key [11] for security reasons. It is therefore interesting to see a neural system simulating such behavior.

[2]The watermark is also influenced by the image content, a dependency we omit for simplicity.

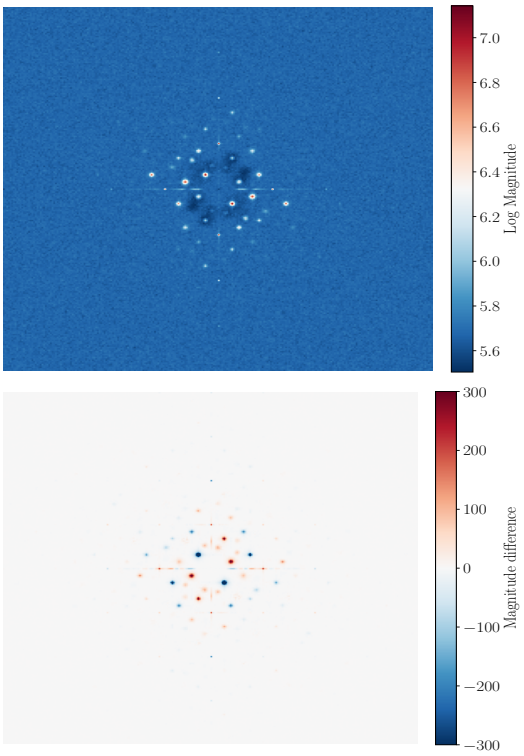

Fig. 3: Top: Average magnitude $W(m)$ of the watermarked image spectrum in the watermark color component found by PCA. The average is taken over 500 watermarked random uniform color images. Bottom: Difference between two average magnitudes $W(m_1)$ and $W(m_2)$ with $m_1 \neq m_2$. The values are clipped for better visualization.

We can now inspect the watermarking effect in a single color channel given by the vector $T_{1,*}$. Let P_1 be the principal color component after the transformation T, which we shall call the *watermark color component*. We generated $N = 500$ random uniform images I_i with constant message m as in the previous paragraph, watermarked them to obtain $I_i(m)$, and transformed to the watermark color component via T. This can be seen as a small simplification of the KMA scenario, since in this case the estimation of the watermark is less noisy than when having content.

Next, we compute the average magnitude spectrum $W(m)$:

$$W(m) = \frac{1}{N} \sum_{i=1}^{N} \left| \mathcal{F}\left\{P_{i,1}\right\} \right|, \tag{1}$$

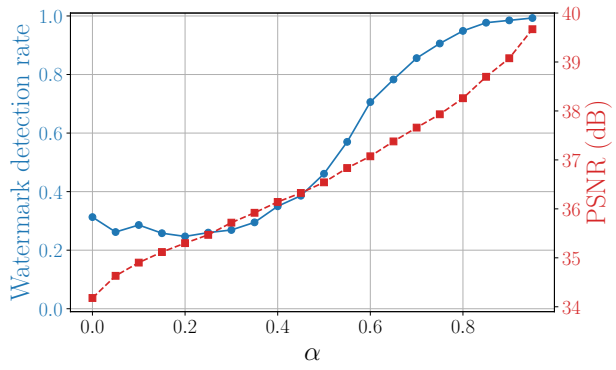

Fig. 4: **Watermark Removal**: watermark detection rate (↓) and PSNR w.r.t. the watermarked image as a function of α.

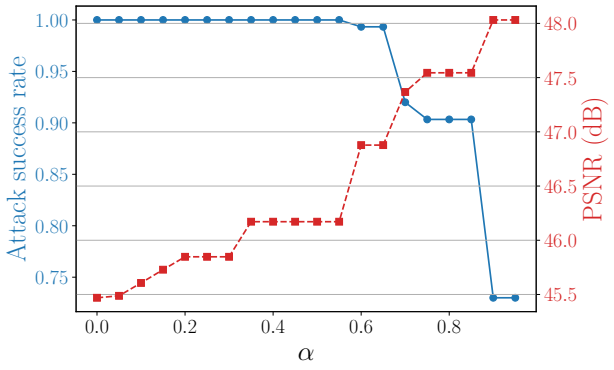

Fig. 5: **Watermark injection**: watermark detection rate (↑) and PSNR w.r.t. the original image as a function of α.

where $|\mathcal{F}\{\cdot\}|$ represents the magnitude of the Fourier transform. As expected, Figure 3 shows distinct gaussian-like blobs in the average magnitude $W(m)$, which we will aim to remove in the following section. In the same figure, we can observe that these blobs change based on the message m.

Note that while the aforementioned eigenvector closely matches the hypothesized ratio $(1, -1, 2)$, this holds for one specific message m among all 2^{32} possibilities. Yet, we observed that the first eigenvector varies only slightly across different messages. This difference is nevertheless quite substantial for a correct watermark removal described in the next subsection.

3.2.3. Frequency removal

To remove the watermark, the average magnitude $W(m)$ is first estimated as explained in (1). Given a watermarked image $I(m)$, we first compute its projection to the watermark color component and its spectrum magnitude $M(I) = |\mathcal{F}\{P_1\}|$.

We now want to subtract the average magnitude $W(m)$, but we will first threshold $W(m)$ in order to find a good compromise between the attack success rate and the PSNR between the watermarked and attacked images. We introduce a parameter $\alpha \in (0, 1)$ controlling how much of the lowest values of $W(m)$ we discard. In particular, we denote $W_\alpha(m)$ as $W(m)$ with its α lowest values set to 0. The attacked image magnitude is then $M_\alpha(I) = M(I) - W_\alpha(m)$. To reconstruct the image into the RGB domain, we replace the magnitude $M(I)$ with $M_\alpha(I)$, compute the inverse Fourier transform to obtain the attacked watermark color component.

At this point, we copy the other two (intact) components P_2, P_3 to obtain a color image and finally transform into RGB using the inverse color transform T^{-1}. The diagram of the described attack is given in Figure 1. The attack results are shown in Figure 4, where we observe that the best attack success rate is around $\alpha = 0.2$, with only 25% of images still being detected as watermarked and average PSNR above 35 dB, suggesting very good quality of the attacked images.

3.3. Copy Attack

We now propose a method for injecting a watermark into a given image. As previously, we assume that the message m we want to embed is known. We take a constant image $I_c \in \mathbb{R}^{h \times w \times 3}$ with an arbitrary value, e.g., $c = 128$, and generate its watermarked version $I_c(m)$. We compute the watermark $w(m) = I_c(m) - I_c$ and denote $w_\alpha(m)$ the watermark signal with α lowest absolute values set to 0. To watermark an image I, we simply add the watermark signal $I_\alpha(m) = I + w_\alpha(m)$ (with appropriate rounding and clipping to the $[0, \dots, 255]$ dynamic range). As we can see in Figure 5, adding only 40% of the most significant watermark values still correctly watermarks all images while preserving the average PSNR of 46.2 dB. Moreover, using only 5% of the strongest watermark values still successfully embeds the watermark in 75% of cases.

A simpler attack is to simply use the WAM embedder to inject the watermark into the image I, although we observed, somewhat surprisingly, that this reduces the PSNR w.r.t. the original image compared to the proposed injection attack to approximately 36.9 dB.

It is worth mentioning that since the model is publicly available, we can technically use gradient-based attacks. This is perhaps a more straightforward and common way of attacking, yet we do not consider it in this work to showcase that the model gradients are not necessary for successful attacks and that an oracle access to the watermarking system only is needed [9]. Moreover, our approach provides a deeper insight and understanding of the inner mechanisms of the system.

4. SINGLE-SHOT ATTACKS

In contrast to the oracle-based attacks discussed previously, we now examine a more restrictive scenario in which an adversary has access only to observed watermarked contents (WOA). This setting precludes the attacker from querying the watermark embedder. We demonstrate that, even with a sin-

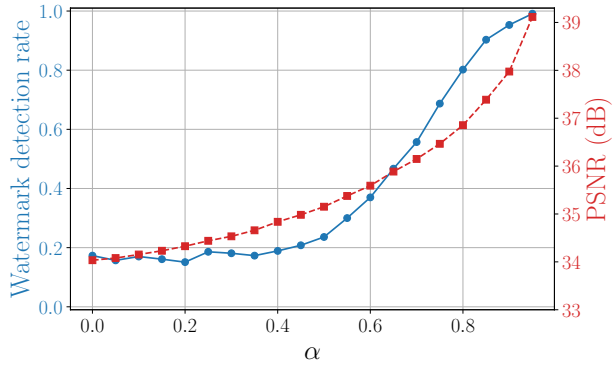

Fig. 6: **Single-shot watermark Removal**: watermark detection rate (↓) and PSNR w.r.t. the watermarked image as a function of α.

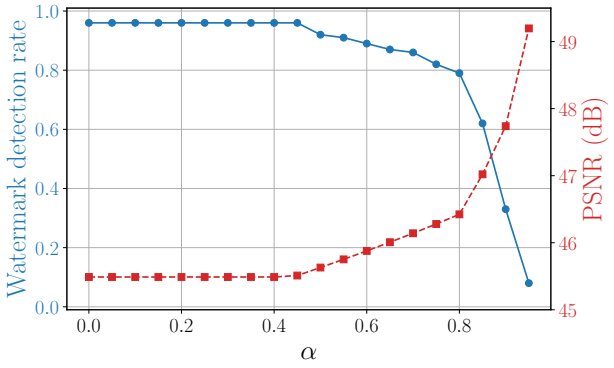

Fig. 7: **Single-shot watermark injection**: watermark detection rate (↑) and PSNR w.r.t. the original image as a function of α.

gle observed watermarked image, the embedded watermark can be recovered, removed, or copied to another image.

4.1. Watermark Estimation from a Residual Image

Direct estimation of the watermark from raw image pixels is impeded by strong inter-channel correlations among the RGB color components. To suppress these dependencies, we first compute a noise residual

$$r = I(m) - \mathcal{D}(I(m)), \tag{2}$$

where $I(m)$ denotes an observed watermarked image and $\mathcal{D}(\cdot)$ is a denoising operator. For our experiments, we employ the Wiener-filter variant proposed in [4], with its default parameters, a window size of four, a stride of two, and the FFT and interpolation weights set to 0.05.

The following attacks assume that all watermarked images contain the same payload m, which mirrors practical scenarios in which the watermark carries an immutable tag, such as a user identifier.

4.2. Removal Attack

For a single residual r, we fit a PCA model to its pixel values, yielding three orthogonal color components $P_{r,1}, P_{r,2}$, and $P_{r,3}$. Just as previously, the magnitude spectrum of the first component is computed as $R(m) = |\mathcal{F}\{P_{r,1}\}|$. Given a target image that contains the same message m, we remove the watermark by subtracting the estimated magnitude $R(m)$ from the first component of its Fourier representation, following the same procedure outlined in Section 3.2. The only difference is that the PCA transformation is fitted on the residual r rather than on the raw pixels of randomly generated noise.

Figure 6 illustrates that the removal performance closely matches that of the oracle-based attack. At a watermark detection rate of 20% with $\alpha = 0.4$, the resulting images exhibit a PSNR of approximately 35 dB.

4.3. Copy Attack

To demonstrate that the recovered watermark can be reinjected into a new image, we first remove the watermark from the observed image $I(m)$ using $\alpha = 0$, yielding a pristine non-watermarked image I_0. The pixel-domain estimate of the watermark is then obtained as $\delta(m) = I(m) - I_0$. Because direct addition of $\delta(m)$ often introduces perceptible artefacts, we scale the estimate by a factor λ. Through a small grid-search, we find $\lambda = 0.03$ provides a satisfactory trade-off between fidelity and watermark detectability. The injected image is thus defined as $I_\alpha(m) = I + \lambda\delta_\alpha(m)$, where $\delta_\alpha(m)$ denotes the estimate with its α lowest absolute values set to zero. Figure 7 shows the resulting performance.

Compared with the oracle-based copy attack, the single-shot injection is marginally noisier: at 100% detection, we obtain a PSNR of 45.6 dB, versus 46.2 dB in the oracle scenario. Nonetheless, both attacks achieve comparable performance even without access to the watermarking system.

5. CONCLUSIONS

Similarly to [2] in the audio domain, this paper shows that if the designer of the watermarking system does not take into account a secret parameter but asks for both robustness and localisation, the watermarking system may adopt a conservative strategy such as embedding periodical patterns that greatly degrade the security of the scheme. Consequently, powerful yet easy attacks can efficiently attack the watermarking system even with a single observed watermarked image. Note also that for watermarking schemes where the security is increased by the use of a secret direction in the latent space, such as [10], the relatively small dimension of the latent space also makes copy removal or copy attacks possible [12]. Future works may consider the design of loss functions capturing the notion of security.

6. ACKNOWLEDGEMENTS

This work received funding from the French Defense & Innovation Agency. This work was also supported by a French government grant managed by the Agence Nationale de la Recherche under the France 2030 program, reference ANR-22-PECY-0011.

7. REFERENCES

[1] Mauro Barni, Franco Bartolini, and Teddy Furon. A general framework for robust watermarking security. *Signal Processing*, 83(10):2069–2084, 2003.

[2] Patrick Bas and Jan Butora. The AI waterfall: Case study in integrating machine learning and security. In *XXXème Colloque Francophone de Traitement du Signal et des Images*, Strasbourg, France, August 25–29 2025.

[3] Patrick Bas and Teddy Furon. A new measure of watermarking security: The effective key length. *IEEE Transactions on Information Forensics and Security*, 8(8):1306–1317, 2013.

[4] Clément Bled and François Pitié. Pushing the limits of the wiener filter in image denoising. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2590–2594, 2023.

[5] Jan Butora and Patrick Bas. The adobe hidden feature and its impact on sensor attribution. In *Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '24, page 143–148, New York, NY, USA, 2024. Association for Computing Machinery.

[6] Jan Butora and Patrick Bas. Detection of the adobe pattern. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 730–734, 2024.

[7] Francois Cayre and Patrick Bas. Kerckhoffs-based embedding security classes for woa data hiding. *IEEE Transactions on Information Forensics and Security*, 3(1):1–15, 2008.

[8] François Cayre, Caroline Fontaine, and Teddy Furon. Watermarking security: theory and practice. *IEEE Transactions on signal processing*, 53(10):3976–3987, 2005.

[9] Pedro Comesana, Luis Pérez-Freire, and Fernando Pérez-González. Blind newton sensitivity attack. *IEE Proceedings-Information Security*, 153(3):115–125, 2006.

[10] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3054–3058. IEEE, 2022.

[11] Teddy Furon and Patrick Bas. Broken arrows. *EURASIP Journal on Information Security*, 2008(1):597040, 2008.

[12] Vitaliy Kinakh, Brian Pulfer, Yury Belousov, Pierre Fernandez, Teddy Furon, and Slava Voloshynovskiy. Evaluation of security of ml-based watermarking: Copy and removal attacks. In *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2024.

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[15] Robin San Roman, Pierre Fernandez, Hady Elsahar, Alexandre Défossez, Teddy Furon, and Tuan Tran. Proactive detection of voice cloning with localized watermarking. In *Proceedings of the 41st International Conference on Machine Learning*, pages 43180–43196, 2024.

[16] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations (ICLR)*, 2025.

[17] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6881–6890, June 2021.