OPTIMIZING AND SECURING THE MODERN WATERMARKING CHANNEL FOR IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

To comply with recent regulations requiring traceable generated content, modern watermarking has adopted multi-bit post-hoc watermarking schemes. These modern designs rest on an encoder-decoder pair implemented as deep neural networks. These models are usually treated as pure black-boxes trained end-to-end, with the noise of the watermarking channel modeled through a fixed set of geometric and valuemetric transforms applied to watermarked images. We argue that this purely empirical approach leads to unquestioned design flaws and a lack of theoretical performance guarantees. This work proposes a general theoretical model of modern post-hoc watermarking schemes grounded in a statistical analysis of the outputs of the encoder/decoder pair. We show that these deep neural networks implicitly define a watermarking channel modeled as parallel AWGN channels, with messages transmitted using BPSK modulation. This imposes a binary alphabet, greatly limiting the capacity of these watermarking systems. Another fatal flaw is their lack of a secret key, making them intrinsically insecure. We make this notion of watermarking security precise for post-hoc schemes by linking it to the possibility of estimating the secret key under a given statistical model of the decoder's output. By putting together the results from this theoretical analysis, we introduce SNW: a novel post-hoc watermarking system that significantly outperforms existing state-of-the-art baselines in terms of capacity while also providing strong security guarantees. Notably, it does not depend on a fixed codebook or binary alphabet, allowing it to reach a rate close to Shannon capacity through the use of capacity-achieving error-correcting codes.

1 INTRODUCTION

Digital media watermarking is going through a striking revival since the early 2020's, accompanying the usage growth of generative models. At the time of writing, the EU AI Act European Data Protection Supervisor (2025) has entered into full force, along with its Code of Practice AI Office (2025), making the use of marking technology mandatory for providers of generative models and services.

For this task, post-hoc watermarking has been largely favored by the industry over in-generation techniques. An instructive evidence is the SynthID report for image watermarking Gowal et al. (2025). It explicitly defines what is expected, in Google DeepMind's view, from a watermarking system – Section 2.1 – and singles out multi-bit post-hoc watermarking as the most practical approach – Section 2.2. This focus of the industry on multi-bit post schemes is also reflected in the open-source literature: Meta's SEAL family Fernandez et al. (2024); Petrov et al. (2025); Souček et al. (2025a) and Adobe's TRUSTMARK Bui et al. (2025) proposed *almost only* post-hoc multi-bit schemes[1].

Such a convergence from the industry warrants a rigorous and critical evaluation methodology for this family of watermarking systems, something which, we believe, is currently missing in the literature.

In order to introduce our argument, it is fruitful to distinguish between two categories of post-hoc schemes: classical and modern. The former, exemplified by Spread-Spectrum Cox et al. (1997) and

[1]The only exception being DISTSEAL Rebuffi et al. (2026), a seed-based in-generation scheme.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Broken-Arrows Furon & Bas (2008), rely on handcrafted, often invertible and orthonormal transforms to construct their watermarking space. The modern approach was pioneered in 2018 by the HIDDEN architecture and training recipe Zhu et al. (2018), and refined across the 2020's with the aforementioned SEAL family, TRUSTMARK and SYNTHID-IMAGE. The main idea rests on training end-to-end a pair of encoder/decoder deep neural networks (DNN). Noise in the watermarking channel is modelled with a finite set of image augmentations, applied after encoding. The pair of DNN is optimized to be as robust as possible to these augmentations, while keeping the watermarking signal imperceptible.

There is a longstanding problem in classical watermarking: one cannot be robust *at the same time* to all geometric transforms. For example, if one builds a watermarking space invariant to rotation, scaling, and translation using the Fourier-Mellin transform, one loses robustness to croppingBas et al. (2002). Solving this so-called geometric robustness problem has often relied on synchronization strategies which bring their own difficulties.

The promise of the modern approach is to solve the geometric robustness problem, *for free*. By simply adding both scaling and cropping to the set of augmentations during training, VIDEOSEAL empirically demonstrated Fernandez et al. (2024)[Appendix B.2] that robustness to both operations was indeed possible and was observed, to some extent, for every modern approach.

Despite this breakthrough, a number of recent works have been pointing out major security flaws in the design of modern watermarking schemes, putting into question the "free-lunch" claim of these schemes. These critiques relate to two main weaknesses: 1) the lack of a secret key to hide the message, making the watermarking system highly vulnerable under Kerckhoffs' principle Bas & Butora (2025); Tarhini et al. (2026) and 2) the reliance on DNN which largely opens up the attack surface related to adversarial security Jiang et al. (2023); Kassis & Hengartner (2025); Gesny & Giboulot (2026a). These works demonstrated that, for these systems, it is not only possible but simple and inexpensive to erase, copy, or forge a watermarking signal.

This also questions the value of modern schemes beyond the geometric problem: classical schemes seem to perform as well in terms of robustness against valuemetric augmentations (JPEG compression, linear filtering, saturation, . . .), yet they are far less vulnerable to both white-box and black-box attacks Gesny & Giboulot (2026a). The problem seems even more pressing given that, of all the modern schemes cited so far, only one, SYNTHID-IMAGE, *mentions* the notion of security.

Another limitation of the modern literature is methodological. Modern systems all rely on the same end-to-end training recipe, with innovations being mostly architectural. A case in point is Chunky-Seal Petrov et al. (2025): its authors are surprised that the theoretical watermarking capacity achievable for images is far from being attained by current schemes. Their answer is to train a larger model. In Section 3 we demonstrate that the inefficiencies of current schemes actually stem from implicit design choices induced by the architecture and training pipeline. We then show how to **increase the capacity of modern schemes almost threefold** without increasing the size of the encoder/decoder, nor changing their architecture, and with significantly less training compute.

The main goal of this work is to clarify and systematize the design of post-hoc watermarking systems. Notably, we want to emphasize the security aspect of watermarking, which is too often neglected in the modern literature:

- We propose a general framework for defining and evaluating each component of modern post-hoc systems – Section 2. In particular, we propose a formal distinction between watermarking security and adversarial robustness.
- We provide a theoretical model of the modern post-hoc watermarking channel as an Additive White Gaussian Channel (AWGN), with the watermarking signal transmitted using Binary Phase-Shift Keying (BPSK). We empirically validate this model on state-of-the-art watermarking systems, demonstrating at the same time fundamental limitations of this design – Section 3.
- We design a novel watermarking system, Secure Neural Watermarking (SNW), which integrates a secret key in the embedding and decoding mechanisms. It reaches a maximum capacity of 656 bits with a binary alphabet, significantly outperforming the 256 bits current state-of-the-art with the same DNN architecture, while being perfectly secure against PCA attacks – Section 4-5.

2 MULTI-BIT WATERMARKING SYSTEMS

2.1 PROBLEM FORMULATION

Cox defines multi-bit watermarking as the "reliable transmission of a message over an unreliable channel" Cox et al. (2006). A useful illustrative scenario for this problem is the attribution scenario that we adapt from Gesny & Giboulot (2026b)[Section 2]:

Watermarking attribution scenario Alice supplies an API where users can request images to be generated. In order to trace the use of her system, she asks Bob, a third-party, to provide her with a secret key k and a post-hoc watermarking system $\mathcal{W}$. From the point of view of Bob, the secret key k is now uniquely linked to Alice. She then associates a unique ID $\mathbf{m}_{\text{user}}$ with each user. Each request generates an unwatermarked image which is passed to $\mathcal{W}$ in order to watermark it. The client is served only the watermarked content. If Bob is presented with the key k and a watermarked image generated for user i, he should decode the correct message $\mathbf{m}_i$. If the image was not watermarked, or the key is incorrect, the decoded message should be random.

Our threat model is focused on spoofing attacks under Kerchoff's principle. Note that shifting to a threat model on erasure attacks requires only slight changes in the analysis, which mostly amounts to working on watermarked images instead of non-watermarked ones.

Threat model – Spoofing Attack Eve is a malicious third party who observes a collection of N pristine watermarked images created by independent users of Alice's API. Eve aims to spoof a specific user's identity: Camille's. Her identity is recorded as the message $\mathbf{m}_{\text{Camille}}$ through the codeword $\mathbf{c}_{Camille}$. Kerckhoff, a malicious colleague of Alice, provides Eve with complete access to the watermarking pipeline **except** Alice's secret key k. We further assume that Eve has access to at least one image generated by Camille – though she does not know $\mathbf{c}_{Camille}$ a priori. We assume that when decoding an image, Eve does not introduce any noise (i.e., perfect channel assumption).

2.2 MAIN DEFINITIONS

Message vs Codeword For a M-bit multi-bit system, it is important to distinguish a message $\mathbf{m} \in \{0,1\}^M$ from a codeword $\mathbf{c}$. A codeword is the object that is eventually mapped into a message using an error-correcting code – see the redundancy mechanism definition Def B.1. In our case, a codeword can be either a sequence of bits in $\{0,1\}^{M'}$ or a real vector in $\mathbb{R}^{M'}$. We impose that $M' \geq M$. We sometimes leave the alphabet of the codeword unspecified, in which case we denote it as $\mathcal{A}^{M'}$.

Post-hoc watermarking systems We adapt the general formulation from Gesny & Giboulot (2026b)[Def. B.2] to post-hoc watermarking systems $\mathcal{W}$ as:

- A set of secret keys $\mathcal{K}$
- A family of *deterministic* embedding functions $(e_k)_{k\in\mathcal{K}} : \mathbb{R}^{L_e} \times \mathcal{A}^{M'} \to \mathbb{R}^L$, embedding the watermark signal into the host content, with respect to the secret key k.
- Two forward projection functions $f_e : \mathbb{R}^D \to \mathbb{R}^{L_e}$ and $f_d : \mathbb{R}^D \to \mathbb{R}^L$, projecting the content from pixel space into watermarking space.
- A backward projection function $f_e^\dagger : \mathbb{R}^{L_e} \to \mathbb{R}^D$
- A family of decision mechanisms $(d_k)_{k\in\mathcal{K}} : \mathbb{R}^L \to \mathcal{A}^{M'}$, extracting the codeword $\mathbf{c}$ from an observation in watermarking space using the secret key k.

For the rest of the paper, we denote by $\mathcal{F}$ the distribution of cover images. Following our watermarking scenario, the cover distribution should be such that it is not biased towards a given message – see Def. B.3.

2.3 WATERMARKING SECURITY VS ADVERSARIAL ROBUSTNESS

A post-hoc watermarking system offers two main attack surfaces for Eve in our spoofing scenario: 1) unauthorized access to the watermarking channel if the secret key is stolen and 2) copying the latent vector $f_d(\mathbf{x}_{\text{wm}})$ of an image for which $\mathbf{m}$ is known using an adversarial example.

The current literature usually conflates these two approaches into a single concept of security. We argue in Appendix E that they are two very different problems, with different consequences. We thus distinguish between true watermarking security, pertaining to embedding/decoding mechanism (e_k, d), and adversarial robustness, which pertains to the decoding projection function f_d.

Definition 1 (Watermarking Security)**.** *Let $\mathcal{W}$ be a watermarking system. Let $X^{(N)}$ be a set of N independent images watermarked with a system $\mathcal{W}$. Denote $Z_i := f_d(X_i)$.*

Let $\psi_N : (\mathbb{R}^L)^N \to \mathcal{K}$ be a function that estimates a secret key from a set of watermarked observations.

For a given codeword $\mathbf{c}$ and unwatermarked image $\mathbf{x}$, denote the probability of attack success as:

$$\mathrm{P}_{\text{atk}}(N, \mathbf{x}) := \mathbb{P}[d(f_d(\mathbf{x}_{\text{atk}}^{(N)}, k)) = \mathbf{c}], \tag{1}$$

where $\mathbf{x}_{\text{atk}}^{(N)} := f_e^{\dagger}(e_{\psi_N(Z^{(N)})}(f_e(\mathbf{x}), \mathbf{c}))$. The watermarking system $\mathcal{W}$ is said to be ηL-secure iff:

$$\mathbb{E}\left[P_{\text{atk}}(\eta L, Y)\right] \leq \mathbb{E}\left[\mathbb{P}[d(f_d(Y), K) = C]\right], \tag{2}$$

where the expectation is taken over (Y, C, K) triplets, with $\mathbf{x}$ sampled from the cover distribution $\mathcal{F}$, and (C, K) sampled uniformly from their respective set.

In words, a watermarking system is $\frac{N}{L}$-secure if, on average, providing Eve with N independent observations does not improve her spoofing success beyond random chance. We will only focus on the watermarking security aspect; adversarial robustness is a wholly different problem that needs to be tackled with adversarial machine learning tools, which is out of the scope of this paper (but see Appendix E for an evaluation of the current state of things).

3 THE MODERN POST-HOC WATERMARKING CHANNEL

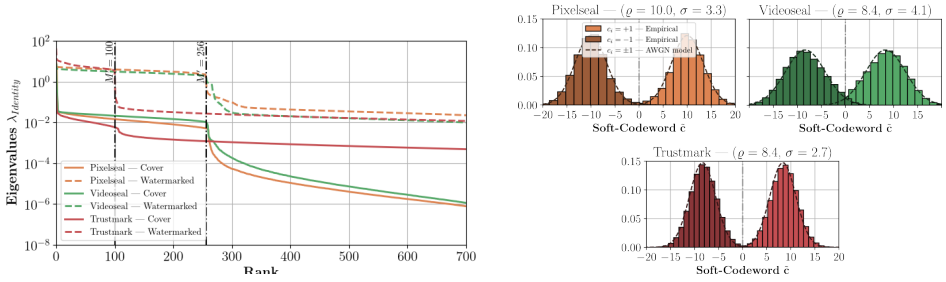


Figure 1: Empirical analysis of the decoding projection f_d and decision mechanism d of three SOTA post-hoc watermarking systems computed over 10k 1024×1024 ImageNet Deng et al. (2009) images watermarked with random messages. *(Left)* Eigenvalues of the covariance Σ_{Identity} of the latent vector $f_d(\mathbf{x}_{\text{wm}})$ (i.e. when t is the identity). *(Right)* Global distribution of the soft-codeword values $\tilde{\mathbf{c}}$ and the corresponding theoretical AWGN model with (ϱ, σ) computed empirically.

Let $\mathbf{x}$ be an unwatermarked image and $t : \mathbb{R}^D \to \mathbb{R}^D$ a transform in pixel space (which can be the identity function). The deep neural networks serving as the projection functions of modern watermarking systems all rely on the same design:

Embedder $(f_e, f_e^{\dagger}, e)$ The forward and backward embedding projections are implemented with a single auto-encoder[2]. It is composed of a downsampling stage f_e – the forward embedding projection – a bottleneck, where current systems usually apply the embedding function e, and an upsampling stage $f_e^{\dagger}$ – the backward embedding projection. The embedding function e maps the codeword

[2]The HiDDeN embedder was more complicated, but recent designs have all streamlined the embedding down to one single auto-encoder.

	Channel characteristic $\varrho\sigma^{-1}(\uparrow)$					Capacity $C_(\varrho,\sigma)$ / Empirical Capacity				
	Identity	JPEG 50	Contrast ×2	Crop 0.6	Rot. 90°	Identity	JPEG 50	Contrast ×2	Crop 0.6	Rot. 90°
PIXELSEAL	3.07	2.21	1.60	1.73	1.97	0.99 / 0.96	0.90 / 0.88	0.69 / 0.72	0.75 / 0.75	0.84 / 0.79
VIDEOSEAL	2.03	1.64	1.03	0.71	1.17	0.85 / 0.85	0.71 / 0.75	0.38 / 0.47	0.21 / 0.21	0.74 / 0.75
TRUSTMARK	3.11	2.67	1.26	0.26	0.20	0.99 / 0.98	0.96 / 0.95	0.52 / 0.58	0.03 / 0.00	0.02 / 0.00

Table 1: Channel characteristic $\varrho\sigma^{-1}$ computed empirically over 10k 1024×1024 ImageNet images watermarked with random messages and the corresponding theoretical capacity in message bits per codeword bit.

$\mathbf{c} \in \{0,1\}^{M'}$ to what we call a *steering vector* $\mathbf{v}(\mathbf{c}) : \{0,1\}^{M'} \to \mathbb{R}^{M'}$. This mapping is fixed in advance, for example, VideoSeal and PixelSeal use WAM's method Sander et al. (2025) of mapping the i-th component c_i to one of two a priori fixed number (v_i^-, v_i^+) depending on the parity of c_i. The steering vector $\mathbf{v}(\mathbf{c})$ is then mixed within the downsampling features $f_e(\mathbf{x})$. To summarize, the embedding pipeline can be written as:

$$\mathbf{w} = f_e^\dagger(e(f_e(\mathbf{x}), \mathbf{c})) \text{ where } e(\cdot, \mathbf{c}) := \mathrm{Mix}(\cdot, \mathbf{v}(\mathbf{c})) \tag{3}$$

$$\mathbf{x}_{\mathrm{wm}} = \mathbf{x} + 10^{\frac{\gamma}{20}} \sqrt{\mathbf{w}} ||\mathbf{w}||_2^{-1}, \tag{4}$$

where γ controls the PSNR of the watermarking signal and Mix is an unspecified function mixing the steering vector with the input (in practice implemented as concatenation). Before reaching the decoder, the watermarked image gets perturbed by the transform t: $\tilde{\mathbf{x}}_{\mathrm{wm}} := t(\mathbf{x}_{\mathrm{wm}})$.

Decoder f_d The decoding stage is always implemented as a convolutional neural network (CNN) acting as a bit classifier. The decoding projection itself encompasses the main CNN module and the final pooling operation that outputs a L-dimensional latent vector $\mathbf{z} = f_d(\tilde{\mathbf{x}}_{\mathrm{wm}})$. Empirically, we observe that $\mathbf{z}$ can be modeled as following a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_z, \Sigma_t)$. In practice, none of the studied systems have $L = M'$. Consequently, we must assume Σ_t to be only semi-definite, i.e the Gaussian may be degenerate.

Decision mechanism d Finally, a linear head further reduces the latent vector down to what we call the soft-codeword $\tilde{\mathbf{c}}$. The final codeword $\mathbf{c}$ is then decoded by applying the sign function: $\mathbf{c} = \mathrm{sign}(\mathbf{W}\mathbf{z} + \mathbf{b})$.

In order to understand the role of the linear head, we turn our attention to the structure of Σ_t in Figure 1 (left). As expected, the rank of these covariance matrices is never L. One can observe a "step-like" behavior within the eigenvalues: M' of them are high, while the rest quickly goes to zero. We call the former the *robust components* of the latent vector. Though it could be mistaken for estimation noise, there is indeed a small subset of non-zero eigenvalues with smaller magnitude, these we call the *brittle components*. The rest of the eigenvectors have eigenvalues of zero: these dimensions are purely *redundant*.

The fact that Σ_t is rank deficient sheds some light on role of the linear head: it performs a whitening operation, getting rid of the redundant and brittle components, as well as unbiasing the codeword estimation. We propose to model $\tilde{\mathbf{c}}$ as following a multivariate Gaussian with diagonal covariance $\mathcal{N}(\varrho_t \mathbf{c}, \boldsymbol{\sigma}_t \mathbf{I}_{M'})$ – see Figure 1 (right).

From this analysis, we claim that the watermarking channel induced by the encoder/decoder DNN pair can be modeled with M' parallel AWGN channels and a watermarking signal modulated using BPSK. We record here the well-known results about such communication systems:

Proposition 1 (Capacity of AWGN channel with BPSK)**.** *The capacity $C_{\varrho,\sigma}$ of an AWGN channel with variance σ^2 and signal transmitted with power ϱ and BPSK modulation is $C_{\varrho,\sigma} = h_2\left(1 - \Phi\left(-|\varrho|\sigma^{-1}\right)\right)$, where h_2 is the binary entropy function, and Φ the standard Gaussian c.d.f. Notably, $\forall \varrho > 0, \forall \sigma \geq 0, C_{\varrho,\sigma} \leq 1$.*

In the case of a binary alphabet, BPSK modulation is optimal for an AWGN channel in the sense that it maximizes the theoretically achievable capacity; but it limits the capacity to 1 bit/channel element. We report the (ϱ, σ) for three state-of-the-art watermarking systems and validate the predictivity of this model in Table 1.

Security of keyless systems Modern watermarking systems following this model have no security since they have no secret key. Even if we allow their linear head to be kept secret – disregarding Kerchoff's principle – they are still vulnerable to a linear estimation attack: Eve can simply watermark $L+1$ images with $L+1$ independent messages (the encoding does not depend on a secret key, nor on the linear head). She then applies the decoding projection to these images, giving her $L+1$ latent vectors $\mathbf{z}$. Since she knows $\mathbf{c}$ and $\mathbf{z}$, she can solve the well-defined system $\mathbf{W}\mathbf{z}+\mathbf{b}=\mathbf{c}$ for $\mathbf{W}$ and $\mathbf{b}$. She can then recover Camille's codeword and spoof as many images as she wishes.

4 SNW : SECURE NEURAL WATERMARKING

So far, we have outlined three main limitations of modern watermarking systems: 1) lack of a secret key, making spoofing attacks trivial, 2) inefficient use of the watermark space, with redundant latent components, and 3) a capacity limited to 1 message bit per codeword bit. To address these, we propose SNW , a secure post-hoc watermarking design that allows the use of a continuous alphabet and is designed to maximize the use of watermark space.

4.1 SECURE DECISION MECHANISM AND EMBEDDING FUNCTIONS (e_k, d)

In order for our system to be secure, we integrate a secret key into both the embedding and decision mechanisms. We define the secret key set $\mathcal{K}$ as the set of all semi-orthogonal matrices of dimension $L \times M'$, denoted $\mathbf{U}$, i.e. $\forall \mathbf{U} \in \mathcal{K}, \mathbf{U}^T\mathbf{U} = \mathbf{I}_{M'}$. Let $\mathbf{c}$ be a codeword from a *continous* alphabet $\mathbb{R}^{M'}$. Analogously to Eq.(3), we define the embedding function $e_{\mathbf{U}}$ using a normalized *steering vector* $\mathbf{v} \in \mathbb{S}^{L-1}$ constructed by distributing the watermarking energy between the secret subspace defined by $\mathbf{U}$ and its orthogonal complement:

$$e_{\mathbf{U}}(\mathbf{z}, \mathbf{c}) = \mathrm{Mix}(\mathbf{z}, \mathbf{v}) \quad ; \quad \mathbf{v} := \alpha \frac{\mathbf{U}\mathbf{c}}{\sqrt{M'}} + \sqrt{1-\alpha^2}\frac{(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}}{\|(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}\|_2}, \tag{5}$$

Where $\mathbf{q}$ is realization of a standard Gaussian random variable $\mathcal{N}(0, I_L)$ and $\alpha \in [0, 1]$. We show in Section 5 that α controls the trade-off between capacity and security.

Assuming f_d can perfectly retrieve $\mathbf{v}$, the decision mechanism $d_{\mathbf{U}}$ extracts $\mathbf{c}$ simply by applying the secret rotation $\mathbf{U}$ to $\mathbf{v}$: $d_{\mathbf{U}} = \mathbf{U}^T\mathbf{v} = \mathbf{c}\alpha\sqrt{M'}^1$. Note that, by further applying the sign function after $d_{\mathbf{U}}$, we retrieve the BPSK modulation in Section 3. Yet, since $\mathbf{c}$ is real-valued, we are also free to use any off-the-shelf capacity-achieving codes for AWGN channels such as nested lattice codes Zamir (2014).

4.2 LEARNING ISOTROPIC AND ROBUST NEURAL PROJECTIONS $(f_e, f_e^\dagger, f_d)$

Our decoding projection needs to retrieve the steering vector $\mathbf{v}$ with the least amount of noise possible. However, contrary to other modern designs, we allow embedding any arbitrary vector on the $L-1$ hypersphere. Furthermore, in order to maximize capacity, we don't want the decoding function to favor certain regions of the hypersphere, i.e. we want the distribution of f_d's output to have a covariance Σ with full rank.

To achieve this, we optimize jointly $(f_e, f_e^\dagger, f_d)$ through the following objective balancing alignment, latent space isotropy, and image quality:

$$\mathcal{L} = \lambda_{\text{align}}\mathcal{L}_{\text{align}}(f_d(\mathbf{x}_{\text{wm}}), \mathbf{v}) + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}}(f_d(\mathbf{x}_{\text{wm}}), f_d(\mathbf{x})) + \lambda_{\text{qual}}\mathcal{L}_{\text{qual}}(\mathbf{x}, \mathbf{x}_{\text{wm}}). \tag{6}$$

Alignment $\mathcal{L}_{\text{align}}$ Maximizes the cosine similarity between the extracted representation $\mathbf{z}$ and the target vector $\mathbf{v}$.

Isotropy $\mathcal{L}_{\text{iso}}$ Minimizes correlations by penalizing non-zero inner products across non-watermarked images or images watermarked with different codewords. This enforces a uniformly distributed spherical latent space.

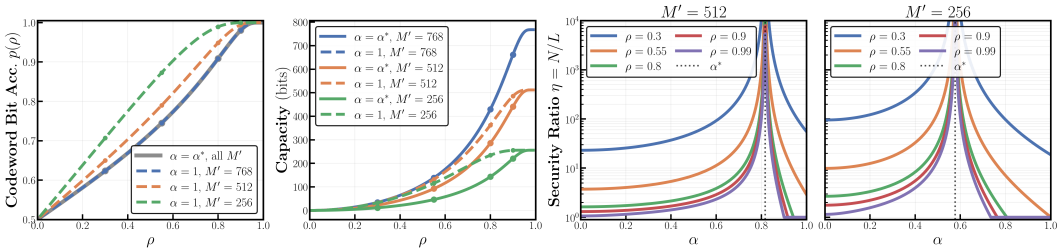


Figure 2: Theoretical bit accuracy $p(\rho)$ (left) and Shannon capacity in bits (right) as a function of the projection cosine alignment ρ. Setting $\alpha = \sqrt{M'/L}$ guarantees perfect security against PCA attacks while maintaining an invariant bit accuracy across all codeword dimensions M'. Dotted lines represent the maximum robustness setting, when $\alpha = 1$. Security ratio η required to estimate the secret subspace with a PCA attack as a function of α for different values of alignment ρ and $M' \in \{256, 512\}$.

Quality $\mathcal{L}_{\text{qual}}$ Preserves the visual fidelity of the host content by combining a constraint on the PSNR of the watermark power and an LPIPS perceptual metric Zhang et al. (2018).

We provide complete definitions of all loss components in Appendix C.2 and details on the training procedure in Appendix C.

5 SNW THEORETICAL ANALYSIS

5.1 CAPACITY

Let $\mathbf{x}_{\text{wm}}$ be an image watermarked with SNW and $t : \mathbb{R}^D \to \mathbb{R}^D$ a transform in pixel space (which can be the identity). Denote $\tilde{\mathbf{v}} := (f_d \circ t)(\mathbf{x}_{\text{wm}})$ the normalized steering vector extracted from the transformed $\mathbf{x}_{\text{wm}}$. We model $\tilde{\mathbf{v}}$ as a perturbation of the true steering vector $\mathbf{v}$:

$$\tilde{\mathbf{v}} = \rho\mathbf{v} + \sqrt{1-\rho^2}\,\mathbf{n}, \quad \text{with } \mathbf{n} \sim \mathcal{U}(\mathbb{S}^{L-2}), \quad \mathbf{n} \perp \mathbf{v}. \tag{7}$$

In practice, note that due to the decoding projection f_d imperfections, even if t is the identity function, we don't expect $\rho = 1$.

We now provide the SNW capacity when using a binary alphabet for the codeword $\mathbf{c}$:

Proposition 2 (Binary SNW capacity)**.** *Under the perturbation model defined in Equation 7, the probability $p(\rho)$ of correctly decoding a bit is given by:*

$$p(\rho) = \Phi\left(\alpha \frac{\rho}{\sqrt{1-\rho^2}} \sqrt{\frac{L}{M'}}\right), \tag{8}$$

where Φ is the standard Gaussian c.d.f. The resulting Shannon capacity is: $C_\rho = M'\,(1 - h_2(p(\rho)))$.

Proof. See Appendix D.1. □

As shown in Figure 2, maximizing decoding robustness corresponds to $\alpha = 1$, where all embedding energy is allocated to the watermark signal.

5.2 WATERMARKING SECURITY

The attacker aims to retrieve the secret matrix $\mathbf{U}$ from $N \geq L$ watermarked images with different codewords. Under our statistical model, Principal Component Analysis (PCA) can be used to do so. Eve estimates the covariance matrix $\mathbf{\Sigma}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top$. The expected eigenvalues have two possible values, each spanning subspaces of dimensions M' and $L - M'$, respectively:

$$\lambda_1 = \rho^2 \frac{\alpha^2}{M'} + \frac{1-\rho^2}{L}, \qquad \lambda_2 = \rho^2 \frac{1-\alpha^2}{L-M'} + \frac{1-\rho^2}{L}, \tag{9}$$

For un-watermarked images, the Marchenko-Pastur distribution gives the asymptotic support S_0 of empirical eigenvalues as $N \to \infty$ Vallet et al. (2012); Furon et al. (2013). Furthermore, the theory shows that one cannot distinguish between covariances if the support S_1 and S_2 of the Marchenko-Pastur distribution associated with λ_1 and λ_2 are such that $S_1 \cup S_2 \subseteq S_0$: the PCA is not able to estimate any dimension of the secret key.

Proposition 3 (SNW PCA security)**.** *A* SNW *system is ηL-secure against PCA attack, requiring at least $N = \eta L$ watermark observations to retrieve the watermark subspace, with:*

$$\frac{N}{L} = \eta = \max\left(1, \frac{\left(1 - \sqrt{\lambda_1 M'}\right)^2}{L\left(\sqrt{\lambda_1} - \frac{1}{\sqrt{L}}\right)^2}\right). \tag{10}$$

In particular, by setting α to $\alpha^* := \sqrt{\frac{M'}{L}}$ we have that $N \to \infty$ and the system is said to be perfectly secure against PCA attacks.

Proof. See Appendix D.2-D.3. □

Note that SNW is always perfectly secure when $\alpha = 1$ and $L = M'$. This is also the regime that achieves maximum capacity. In other words, when the watermark space is used efficiently – the codeword size matches watermarking space dimensions – one should allocate all the energy to the watermarking signal. However, if the codeword size is smaller than L, such as in the case of other modern systems, one must waste energy in order to "drown" the watermarking signal into random noise. We validate the complete theoretical analysis empirically in Appendix H.1.

6 Experiments

6.1 Experimental Settings

We evaluate the proposed SNW watermarking scheme against state-of-the-art recent post-hoc baselines across the capacity-quality-security trade-off.

Baselines watermarking systems We benchmark SNW against three recent neural network watermarking schemes: VideoSeal Fernandez et al. (2024) and PixelSeal Souček et al. (2025a), which embed a 256-bit binary codeword, and TrustMark Bui et al. (2025), which embeds a 100-bit binary codeword. To ensure a fair message-agnostic comparison, we whiten the output representation of all models on a set of 10^6 MFlickr images, following the procedure described in Appendix F.

SNW setup We train the SNW encoder/decoder pair on the COCO 2017 train dataset Lin et al. (2014) using the multi-stage pipeline detailed in Appendix C. To isolate the contribution of our watermarking design from architectural choice, SNW adopts the same architecture and augmentations as PixelSeal, *except for the linear head*, which is removed. Although our framework supports continuous alphabets to achieve higher transmission rates, for fairness we constrain our evaluation to a binary alphabet when comparing with baselines. Throughout this section, SNW is parameterized with $\alpha^* = 1$ at $M' = L$, guaranteeing optimal security against PCA key estimation attacks, as stated in Proposition 3.

Image and transforms dataset Evaluation is performed over 1000 1024×1024 natural images randomly sampled from the MFlickr dataset, disjoint from the COCO training set. To assess capacity under realistic channel degradations, all methods are tested against an extensive set of valuemetric and geometric transforms reported in Appendix H(Table 7).

Evaluation protocol We evaluate the performance of a watermarking system along three characteristics:

- *Capacity:* The watermarking literature usually reports message bit-accuracy without error correcting code. Such a measure makes no sense: even a watermarking system which

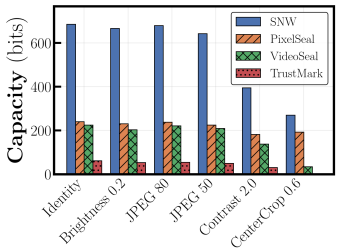


Figure 3: Comparison of recent watermarking systems in terms of capacity in bits over 1000 images. The evaluation is performed against classic image transformations.

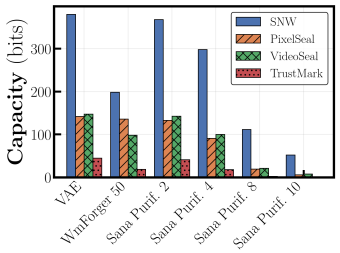


Figure 4: Comparison of recent watermarking systems in terms of capacity in bits over 1000 images. The evaluation is performed against recent watermarking erasure attacks.

Method	LPIPS (↓)	η-Security (↑)
PIXELSEAL	0.0034	1.00
VIDEOSEAL	0.0025	1.00
TRUSTMARK	**0.0010**	1.01
SNW	0.0047	$+\infty$

Table 2: Comparison between watermarking systems in terms of perceptual quality and security at a fixed watermarking power of 48 dB PSNR. The perceptual quality is measured with the LPIPS.

boasts a bit-accuracy of 0.997 such as PIXELSEAL has a probability of decoding error of $1 - 0.997^{256} = 0.53$; far too high to be of any use. We argue that a practical system must use error-correcting codes (ECC). In order to stay agnostic to the specific choice of ECC, as well as to the choice of probability of decoding error, we report the total Shannon capacity (in bits), computed from the empirical codeword bit-accuracy. Note that this empirical evaluation assumes each codeword bit to be independent, which is ensured with the whitening operation in Appendix F.

- *Quality:* For a fair evaluation, we calibrate the watermark power to a fixed 48dB PSNR across all methods. We assess the perceptual quality of the watermark via the standard LPIPS metric Zhang et al. (2018).
- *Security:* Evaluated through the η-security of Definition 1 when Eve uses a PCA attack for key estimation.

6.2 RESULTS

Capacity – Figure 3-4 SNW consistently achieves higher empirical capacity than existing baselines under a binary alphabet, achieving around $2.5\times$ the capacity of PIXELSEAL for both classical valuemetric and geometric operations as well as more recent blind erasure attacks based on diffusion models. Additional results covering extended transforms set and watermark erasure attacks are provided in Appendix H(Tables 7 and 6).

Quality – Table 2 Although TrustMark achieves slightly lower perceptual distortion (at the cost of far lower capacity than all other schemes), all evaluated methods remain within the same order of magnitude.

Security – Table 2 Following the discussion in Section 3, a charitable analysis associates a security ratio $\eta = L(L+1)^{-1} \approx 1$ for keyless systems. Based on Proposition 3, under the regime $\alpha^* = 1$ for $M' = L$, SNW is perfectly secure against PCA secret key estimation.

7 CONCLUSION

The main impetus of this paper was to pull the focus away from training and architecture in modern watermarking design. Instead, we argued for a careful examination of the implicit channel of these systems. In addition to a lack of security, our analysis revealed an inefficient use of the watermarking space, yielding suboptimal capacity for a given neural network architecture. We introduced SNW, a novel post-hoc watermarking system that achieves $2.5\times$ higher capacity while offering theoretical security guarantees through the use of a secret key. Beyond the question of pure performance, we made the design general enough to accommodate more powerful codebook constructions. In particular, the use of a continuous alphabet and corresponding capacity-achieving codes would be a natural extension, allowing to reach far better capacities at almost no cost.

REPRODUCIBILITY

The model was trained on a single NVIDIA L40S GPU for 590 hours. The complete training procedure is detailed in Appendix C. We provide demonstration code to run the SNW model in the supplementary material. All datasets used in this work are detailed in Section 6.1 and Appendix H.

REFERENCES

AI Office. General-Purpose AI Code of Practice. Technical report, European Commission, July 2025. URL https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai.

P. Bas, J.-M. Chassery, and B. Macq. Geometrically invariant watermarking using feature points. *IEEE Transactions on Image Processing*, 11(9):1014–1028, September 2002. ISSN 1057-7149. doi: 10.1109/TIP.2002.801587. URL http://ieeexplore.ieee.org/document/1036050/.

Patrick Bas and Jan Butora. The AI Waterfall : A Case Study in Integrating Machine Learning and Security. In *GRETSI*, Strasbourg, France, August 2025. URL https://hal.science/hal-05011387.

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, July 2013. URL https://hal.science/hal-00836404.

Tu Bui, Shruti Agarwal, and John Collomosse. TrustMark: Robust Watermarking and Watermark Removal for Arbitrary Resolution Images. pp. 18629–18639, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Bui_TrustMark_Robust_Watermarking_and_Watermark_Removal_for_Arbitrary_Resolution_Images_ICCV_2025_paper.html.

François Cayre, Caroline Fontaine, and Teddy Furon. Watermarking security part I: theory. volume 5681, pp. 746. SPIE, January 2005. URL https://inria.hal.science/inria-00083329.

I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, December 1997. ISSN 1941-0042. doi: 10.1109/83.650120. URL https://ieeexplore.ieee.org/abstract/document/650120.

Ingemar Cox, Gwenaël Doërr, and Teddy Furon. Watermarking is not cryptography. 2006. URL https://inria.hal.science/inria-00504528.

Ingemar J Cox and Jean-Paul MG Linnartz. Public watermarks and resistance to tampering. In *International Conference on Image Processing (ICIP'97)*, pp. 26–29, 1997.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

European Data Protection Supervisor. *AI Act Regulation (EU) 2024/1689 – Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance).* Publications Office of the European Union, 2025. doi: doi/10.2804/4225375.

Pierre Fernandez, Hady Elsahar, I. Zeki Yalniz, and Alexandre Mourachko. Video Seal: Open and Efficient Video Watermarking, December 2024. URL http://arxiv.org/abs/2412.09492. arXiv:2412.09492 [cs].

Teddy Furon and Patrick Bas. Broken Arrows. *EURASIP Journal on Information Security*, 2008(1):1–13, December 2008. ISSN 1687-417X. doi: 10.1155/2008/597040. URL https://jis-eurasipjournals.springeropen.com/articles/10.1155/2008/597040.

Teddy Furon, Hervé Jégou, Laurent Amsaleg, and Benjamin Mathon. Fast and secure similarity search in high dimensional space. In *2013 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 73–78, November 2013. doi: 10.1109/WIFS.2013.6707797. URL https://ieeexplore.ieee.org/abstract/document/6707797. ISSN: 2157-4774.

Enoal Gesny and Eva Giboulot. Do Modern Post-Hoc Watermarking Methods Beat Broken-Arrows? In *Proceedings of the 2026 ACM Workshop on Information Hiding and Multimedia Security*, pp. 195–200, Firenze Italy, June 2026a. ACM. ISBN 979-8-4007-2376-6. doi: 10.1145/3785353.3815081. URL https://dl.acm.org/doi/10.1145/3785353.3815081.

Enoal Gesny and Eva Giboulot. Secure Seed-Based Multi-bit Watermarking for Diffusion Models from First Principles, May 2026b. URL http://arxiv.org/abs/2605.06153. arXiv:2605.06153 [cs.CR].

Enoal Gesny, Eva Giboulot, Teddy Furon, and Vivien Chappelier. Guidance watermarking for diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5ifzhjMCKq.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, Chris Gamble, Miklós Z. Horváth, Fabian Kaczmarczyck, Alex Kaskasoli, Aleksandar Petrov, Ilia Shumailov, Meghana Thotakuri, Olivia Wiles, Jessica Yung, Zahra Ahmed, Victor Martin, Simon Rosen, Christopher Savčak, Armin Senoner, Nidhi Vyas, and Pushmeet Kohli. SynthID-Image: Image watermarking at internet scale, October 2025. URL http://arxiv.org/abs/2510.09263. arXiv:2510.09263 [cs.CR].

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading Watermark based Detection of AI-Generated Content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1168–1181, Copenhagen Denmark, November 2023. ACM. ISBN 979-8-4007-0050-7. doi: 10.1145/3576915.3623189. URL https://dl.acm.org/doi/10.1145/3576915.3623189.

T. Kalker. Considerations on watermarking security. In *2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564)*, pp. 201–206, October 2001. doi: 10.1109/MMSP.2001.962734. URL https://ieeexplore.ieee.org/document/962734.

Andre Kassis and Urs Hengartner. UnMarker: A Universal Attack on Defensive Image Watermarking. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2602–2620, May 2025. doi: 10.1109/SP61157.2025.00005. URL http://arxiv.org/abs/2405.08363. arXiv:2405.08363 [cs.CR].

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Jean-Paul MG Linnartz and Marten Van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In *International Workshop on Information Hiding*, pp. 258–272. Springer, 1998.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

Aleksandar Petrov, Pierre Fernandez, Tomáš Souček, and Hady Elsahar. We Can Hide More Bits: The Unused Watermarking Capacity in Theory and in Practice, December 2025. URL http://arxiv.org/abs/2510.12812. arXiv:2510.12812 [cs] version: 2.

Sylvestre-Alvise Rebuffi, Tuan Tran, Valeriu Lacatusu, Pierre Fernandez, Tomáš Souček, Nikola Jovanović, Tom Sander, Hady Elsahar, and Alexandre Mourachko. Learning to Watermark in the Latent Space of Generative Models, January 2026. URL http://arxiv.org/abs/2601.16140. arXiv:2601.16140 [cs.CV] version: 1.

Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations*, volume 2025, pp. 79569–79599, 2025.

Tomas Soucek, Sylvestre-Alvise Rebuffi, Pierre Fernandez, Nikola Jovanović, Hady Elsahar, Valeriu Lacatusu, Tuan A. Tran, and Alexandre Mourachko. Transferable black-box one-shot forging of watermarks via image preference models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yb5JOOmfxA.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel Seal: Adversarial-only training for invisible image and video watermarking, December 2025a. URL http://arxiv.org/abs/2512.16874. arXiv:2512.16874 [cs].

Tomáš Souček, Sylvestre-Alvise Rebuffi, Pierre Fernandez, Nikola Jovanović, Hady Elsahar, Valeriu Lacatusu, Tuan Tran, and Alexandre Mourachko. Transferable Black-Box One-Shot Forging of Watermarks via Image Preference Models, October 2025b. URL http://arxiv.org/abs/2510.20468. arXiv:2510.20468 [cs.LG].

Hussein Tarhini, Aurélien Noirault, Jan Butora, and Patrick Bas. Neural Watermarking: Lack of a Secret Key is still Lack of Security. In *2026 IEEE International Conference on Image Processing*, Tampere, Finland, September 2026. URL https://hal.science/hal-05272053.

Pascal Vallet, Philippe Loubaton, and Xavier Mestre. Improved subspace estimation for multivariate observations of high dimension: the deterministic signals case. *IEEE Transactions on Information Theory*, 58(2):1043–1068, February 2012. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2011.2173718. URL http://arxiv.org/abs/1002.3234. arXiv:1002.3234 [cs].

Ram Zamir. *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory*. Cambridge University Press, Cambridge, 2014. ISBN 978-0-521-76698-2. doi: 10.1017/CBO9781139045520. URL https://www.cambridge.org/core/books/lattice-coding-for-signals-and-networks/23B8D22FD43CD3FD0CBE7F1CE28BAF27.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding Data With Deep Networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, volume 11219, pp. 682–697. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01266-3 978-3-030-01267-0. doi: 10.1007/978-3-030-01267-0_40. URL https://link.springer.com/10.1007/978-3-030-01267-0_40. Series Title: Lecture Notes in Computer Science.

A Notations

Spaces

- **Pixel space**: in $\mathbb{R}^D$ with cover observations denoted $\mathbf{x}$ and watermarked as $\mathbf{x}_{\text{wm}}$.

- **Latent space**: in $\mathbb{R}^L$ during decoding with observations denoted $\mathbf{z}$ and steering vector denoted as $\mathbf{v}$. In $\mathbb{R}^{L_e}$ during encoding.
- **Soft-codeword space**: in $\mathbb{R}^{M'}$ with observations denoted as $\tilde{\mathbf{c}}$.

Distributions We use the standard notation for the standard Gaussian p.d.f ϕ and its c.d.f Φ. Other distributions are usually referred to with calligraphic letters ($\mathcal{N}$).

Other common notation

- $\mathbb{S}^{L-1}$: hypersphere embedded in a L-dimensionnal real space
- $\mathcal{A}^{M'}$: either the binary alphabet $\{0,1\}^{M'}$ or the continuous alphabet $\mathbb{R}^{M'}$.
- $\mathcal{F}$ always refers to the cover distribution – see Def B.3.
- Vectors: vectors $\mathbf{v}$ use lowercase boldface letters.
- **Functions**: functions f use lowercase regular letters.
- **Constant and random variables**: both constant and r.v. use uppercase, regular Latin letters such as C and X. Constants take letters from the start of the alphabet, whereas random variables take letters from the end. Random variables following multivariate distributions are not bolded.

Post hoc watermarking system We refer to a SNW system as an error-corrected watermarking system, where:

- The set of secret keys $\mathcal{K}$, is given by the set of $L \times M'$ matrices with columns summing to 1. An element of this set is denoted as $\mathbf{U}$.
- The decoding projection function f_d is any function from $\mathbb{R}^D$ to $\mathbb{R}^L$.
- The encoding projection function $(f_e, f_e^\dagger)$ are any function from $\mathbb{R}^D$ to $\mathbb{R}^{L_e}$ and from $\mathbb{R}^{L_e}$ to $\mathbb{R}^{L_e}$. Note that L_e need not agree with L.

We assume the redundancy mechanism works at Shannon's capacity. A message is denoted as $\mathbf{m} \in \{0,1\}^M$ and its representative codeword as $\mathbf{c} \in \mathcal{A}^{M'}$.

For convenience, we use a slightly modified version of the sign function defined as:

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{else} \end{cases} \tag{11}$$

Importantly, note that $\text{sign}(0) = -1$.

B Formal Definitions

We herein make precise the distinction between a codeword and a message. This leads to a definition of an error-corrected watermarking system.

Definition B.1 (Redundancy mechanism)**.** *A redundancy mechanism is composed of an encoder and a decoder $(c_{\text{enc}}, c_{\text{dec}})$. The encoder c_{enc} is a bijection between messages and a subset $\mathcal{C} \in \mathcal{A}^{M'}$ called the codebook. For each codeword $\mathbf{c}$ in the codebook, the decoder c_{dec} defines an equivalence class $[\mathbf{c}] = \{\tilde{\mathbf{c}} \in \mathcal{A}^{M'} : c_{\text{dec}}(\tilde{\mathbf{c}}) = \mathbf{c}\}$ and where $c_{\text{dec}} : \mathcal{A}^{M'} \to \mathcal{A}^{M'}$. We abuse notation and make no distinction between the equivalence class $[\mathbf{c}]$ and its representative codeword $\mathbf{c}$.*

Definition B.2 (Error-corrected watermarking system)**.** *A M-bit watermarking system $\mathcal{W}$ equipped with a redundancy mechanism $(c_{\text{enc}}, c_{\text{dec}})$ replaces its embedding functions $(e_k)_{k\in\mathcal{K}}$ by $e_k^{(c)} e_k \circ c_{\text{enc}}$. Its decision mechanism is replaced by a function $d_k^{(c)} c_{\text{enc}}^{-1} \circ c_{\text{dec}} \circ d_k$ where $d_k : \mathbb{R}^L \to \mathcal{A}^{M'}$. We call $d_k^{(c)}$ the error-corrected decision mechanism; we still call d_k the decision mechanism.*

We also characterize implicitly the fact that a watermarking system should not favor a message over another when it receives a cover image with the notion of cover distribution:

Definition B.3 (Cover distribution with binary alphabet)**.** *Let $\mathcal{F}$ be a probability distribution with support in $\mathbb{R}^D$. It is said to be a cover distribution for a M-bit watermarking system $\mathcal{W}$ with a binary alphabet (i.e $\mathcal{A}^{M'} \equiv \{0,1\}^{M'}$) iff, for all secret keys $k \in \mathcal{K}$:*

$$d_k\left(f(X)\right) \sim \mathcal{B}^M\left(\frac{1}{2}\right), \tag{12}$$

where $X \sim \mathcal{F}$ and $\mathcal{B}^M$ is a M-dimensional Bernoulli distribution.

C Training details

In this section, we provide additional details on the training of the projection function of SNW. We do not claim that the procedure presented here is the unique or optimal training method to achieve the objectives introduced in Section 4. The model has been trained on the COCO 2017 train set.

C.1 Architecture

We adopt the neural network architecture proposed by PixelSeal. We remove the classification linear head to train only the ConvNeXt projection and the embedder. While PixelSeal embeds a binary message of dimension $M' = 256$, we modify the input of the U-Net embedder to accept a continuous random unit vector of dimension $L = 768$. Keeping the PixelSeal backbone intact is essential to demonstrate that, with an identical architecture, their end-to-end training pipeline was suboptimal in terms of both capacity and security.

C.2 Losses

We detail below the losses used during the training of the encoder-projection model. As stated in Section 4, the projection aims to:

- Accurately embed and extract an arbitrary target unit vector,
- Ensure isotropy of the extracted representations across distinct codewords and non-watermarked images.

Alignment losses $\mathcal{L}_{\text{align}}$

- $\mathcal{L}_{\text{align}}^{\text{all}}$ Enforces that each spatial vector of the unpooled feature map aligns with the target vector $\mathbf{v}$:

$$\mathcal{L}_{\text{align}}^{\text{all}} = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} \left(1 - \langle \bar{\mathbf{v}}_{i,j}, \mathbf{v} \rangle\right), \tag{13}$$

  where $\bar{\mathbf{v}}_{i,j} = \tilde{\mathbf{v}}_{i,j} / \|\tilde{\mathbf{v}}_{i,j}\|_2$.

- $\mathcal{L}_{\text{align}}^{\text{pool}}$ Enforces that the pooled representation aligns with the target vector $\mathbf{v}$:

$$\mathcal{L}_{\text{align}}^{\text{pool}} = 1 - \langle \tilde{\mathbf{v}}, \mathbf{v} \rangle. \tag{14}$$

Isotropy losses $\mathcal{L}_{\text{iso}}$

- $\mathcal{L}_{\text{iso}}^{\text{same}}$ Penalizes correlations between representations of the same host image $\mathbf{x}^{(i)}$ encoded with two independent targets $\mathbf{v}_1$ and $\mathbf{v}_2$:

$$\mathcal{L}_{\text{iso}}^{\text{same}} = \frac{1}{B} \sum_{i=1}^{B} \langle \tilde{\mathbf{v}}_1^{(i)}, \tilde{\mathbf{v}}_2^{(i)} \rangle^2. \tag{15}$$

- $\mathcal{L}_{\text{iso}}^{\text{cross}}$ Penalizes correlations between representations of different host images $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ encoded with independent targets $\mathbf{v}_1$ and $\mathbf{v}_2$:

$$\mathcal{L}_{\text{iso}}^{\text{cross}} = \frac{1}{B(B-1)} \sum_{i \neq j} \langle \tilde{\mathbf{v}}_1^{(i)}, \tilde{\mathbf{v}}_2^{(j)} \rangle^2. \tag{16}$$

- $\mathcal{L}_{\text{iso}}^{\text{host}}$ Penalizes correlations between representations of unwatermarked host images $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$:

$$\mathcal{L}_{\text{iso}}^{\text{host}} = \frac{1}{B(B-1)} \sum_{i \neq j} \langle \tilde{\mathbf{v}}^{(i)}, \tilde{\mathbf{v}}^{(j)} \rangle^2. \tag{17}$$

Quality losses $\mathcal{L}_{\text{qual}}$

- $\mathcal{L}_{\text{qual}}^{\text{PSNR}}$ Hinge loss on the PSNR bounded by a target budget to ensure watermarking power:

$$\mathcal{L}_{\text{qual}}^{\text{PSNR}} = \text{ReLU}\big(\tau(t) - \text{PSNR}(\mathbf{x}, \mathbf{x}_w)\big), \tag{18}$$

  where $\tau(t)$ is the target PSNR at step t and $\mathbf{x}_w$ is the host image $\mathbf{x}$ watermarked with codeword $\mathbf{c}$.
- $\mathcal{L}_{\text{qual}}^{\text{LPIPS}}$ LPIPS loss to ensure perceptual quality:

$$\mathcal{L}_{\text{qual}}^{\text{LPIPS}} = \text{LPIPS}(\mathbf{x}, \mathbf{x}_w). \tag{19}$$

Anti-spoofing loss $\mathcal{L}_{\text{spoof}}$

- $\mathcal{L}_{\text{spoof}}$ Enforces that a watermark residual $\mathbf{w}_1^{(i)} = \mathbf{x}_{w_1}^{(i)} - \mathbf{x}^{(i)}$ yields an orthogonal representation when transferred onto an unrelated host image $\mathbf{x}^{(j)}$:

$$\mathcal{L}_{\text{spoof}} = \frac{1}{B} \sum_{i=1}^{B} \langle f(\mathbf{x}^{(j)} + \mathbf{w}_1^{(i)}), \mathbf{v}_1 \rangle^2, \tag{20}$$

  where $j = (i \bmod B) + 1$, and $\mathbf{v}_1$ is the target vector embedded into $\mathbf{x}^{(i)}$. See Appendix G for further details on the residual spoofing attack.

C.3 Training Procedure

The training pipeline is divided into three stages involving different combinations of losses, as summarized in Table 3.

Losses	Stage 1	Stage 2	Stage 3
$\mathcal{L}_{\text{align}}^{\text{all}}$	✓	✗	✗
$\mathcal{L}_{\text{align}}^{\text{pool}}$	✗	✓	✓
$\mathcal{L}_{\text{iso}}^{\text{same}}$	✓	✓	✓
$\mathcal{L}_{\text{iso}}^{\text{cross}}$	✗	✓	✓
$\mathcal{L}_{\text{iso}}^{\text{host}}$	✗	✓	✓
$\mathcal{L}_{\text{qual}}^{\text{PSNR}}$	✓	✓	✓
$\mathcal{L}_{\text{qual}}^{\text{LPIPS}}$	✗	✓	✓
$\mathcal{L}_{\text{spoof}}$	✗	✗	✓

Table 3: Active losses per training stage.

Stage 1: Find the signal The first stage of training aims to train the model to encode and extract a watermark signal when there are no transforms, while being isotropic. To find this signal and the isotropy, the model needs to remove the host content to keep only the signal. This is why, during the first step, the loss function is:

$$\mathcal{L}_1 = \lambda_{\text{align}}^{\text{all}} \mathcal{L}_{\text{align}}^{\text{all}} + \lambda_{\text{iso}}^{\text{same}} \mathcal{L}_{\text{iso}}^{\text{same}} + \lambda_{\text{qual}}^{\text{PSNR}} \mathcal{L}_{\text{qual}}^{\text{PSNR}}, \tag{21}$$

with a target PSNR scaled from 0 dB to 42 dB. $\mathcal{L}_{\text{align}}^{\text{all}}$, $\mathcal{L}_{\text{iso}}^{\text{same}}$, and $\mathcal{L}_{\text{qual}}^{\text{PSNR}}$ are respectively described by Equations 13, 15, and 18. We specifically use the align loss on all components of the representation before the mean pooling to enforce the model to find a signal. To provide robustness, we progressively add transforms to the augmentation layer. We use the same set of transforms used by PixelSeal Souček et al. (2025a).

Stage 2: Mean pooling of the signal The second stage of the training consists of transferring the signal from the components of the representation to the pooled representation while improving the perceptual quality. With this step, we also want to improve the isotropy. To do so, we use the following loss function:

$$\mathcal{L}_2 = \lambda_{\text{align}}^{\text{pool}}\mathcal{L}_{\text{align}}^{\text{pool}} + \lambda_{\text{iso}}^{\text{same}}\mathcal{L}_{\text{iso}}^{\text{same}} + \lambda_{\text{iso}}^{\text{cross}}\mathcal{L}_{\text{iso}}^{\text{cross}} + \lambda_{\text{iso}}^{\text{host}}\mathcal{L}_{\text{iso}}^{\text{host}} + \lambda_{\text{qual}}^{\text{PSNR}}\mathcal{L}_{\text{qual}}^{\text{PSNR}} + \lambda_{\text{qual}}^{\text{LPIPS}}\mathcal{L}_{\text{qual}}^{\text{LPIPS}}. \quad (22)$$

Concerning the alignment, $\mathcal{L}_{\text{align}}^{\text{pool}}$ is used to train the model such that the pooled representation obtained is aligned with the target vector $\mathbf{v}$. Regarding the perceptual quality, we add the LPIPS loss presented in 19, as well as a JND attenuation at the encoding stage as proposed by PixelSeal **?**. We also continue to enforce isotropy using the three losses $\mathcal{L}_{\text{iso}}^{\text{same}}$, $\mathcal{L}_{\text{iso}}^{\text{cross}}$, and $\mathcal{L}_{\text{iso}}^{\text{host}}$ that respectively request isotropy between the same image watermarked with different codewords, between different images watermarked with different codewords, and between different non-watermarked images.

Stage 3: Anti-spoofing fine-tuning To address the residual transfer attack, we add a fine-tuning stage with a few steps to train the model such that the detectability of a residual depends on the content of the image. The loss used is:

$$\mathcal{L}_3 = \mathcal{L}_2 + \lambda_{\text{spoof}}\mathcal{L}_{\text{spoof}}. \quad (23)$$

More details about the anti-spoofing loss function and the residual transfer attack are provided in the Appendix G.

D PROOFS

D.1 PROPOSITION 2

Proposition 4 (Binary SNW capacity)**.** *Under the perturbation model defined in Equation 7, the probability $p(\rho)$ of correctly decoding a bit is given by:*

$$p(\rho) = \Phi\left(\alpha\frac{\rho}{\sqrt{1-\rho^2}}\sqrt{\frac{L}{M'}}\right), \quad (24)$$

where Φ is the standard Gaussian c.d.f. The resulting Shannon capacity is: $C_\rho = M'\left(1 - h_2(p(\rho))\right)$.

Proof. Let $\mathbf{v} \in \mathbb{S}^{L-1}$ be the target unit direction encoded in the host content $\mathbf{x}$ for codeword $\mathbf{c} \in \{-1,1\}^{M'}$ under the secret key matrix $\mathbf{U} \in \mathbb{R}^{L\times M'}$. As defined in Equation 5, $\mathbf{v}$ decomposes into the watermark subspace and its orthogonal complement:

$$\mathbf{v} = \alpha\frac{\mathbf{U}\mathbf{c}}{\sqrt{M'}} + \sqrt{1-\alpha^2}\frac{(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}}{\|(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}\|_2}, \quad (25)$$

where $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_{M'}$, $\|\mathbf{c}\|_2 = \sqrt{M'}$, and $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L)$. By orthogonality, $\mathbf{U}^\top(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top) = \mathbf{0}$, which yields:

$$\mathbf{U}^\top\mathbf{v} = \alpha\frac{\mathbf{c}}{\sqrt{M'}}. \quad (26)$$

We model the extracted normalized latent representation $\tilde{\mathbf{v}} \in \mathbb{S}^{L-1}$ as a perturbed $\mathbf{v}$ by a isotropic orthogonal noise component:

$$\tilde{\mathbf{v}} = \rho\mathbf{v} + \sqrt{1-\rho^2}\mathbf{n}, \quad (27)$$

where $\mathbf{n} \sim \mathcal{U}(\mathbb{S}^{L-2})$ is uniformly distributed on the unit sphere of the hyperplane orthogonal to $\mathbf{v}$ ($\|\mathbf{n}\|_2 = 1$, $\langle\mathbf{n}, \mathbf{v}\rangle = 0$).

Projecting $\tilde{\mathbf{v}}$ onto the secret projection matrix $\mathbf{U}$ gives:

$$\mathbf{U}^\top\tilde{\mathbf{v}} = \rho\mathbf{U}^\top\mathbf{v} + \sqrt{1-\rho^2}\mathbf{U}^\top\mathbf{n} \quad (28)$$

$$= \rho\alpha\frac{\mathbf{c}}{\sqrt{M'}} + \sqrt{1-\rho^2}\mathbf{U}^\top\mathbf{n} \quad (29)$$

Asymptotically, we have:

$$(\mathbf{U}^\top \mathbf{n})_j \sim \mathcal{N}\left(0, \frac{1}{L}\right) \tag{30}$$

with

$$\mathbb{E}[(\mathbf{U}^\top \mathbf{n})_j] = 0 \tag{31}$$

$$\mathrm{Var}[(\mathbf{U}^\top \mathbf{n})_j] = \frac{1}{L}. \tag{32}$$

$$\tag{33}$$

We examine the decision on the j-th bit. Let $S_j = \alpha \frac{\rho}{\sqrt{M'}} c_j$ denote the projected signal component and $B_j = \sqrt{1-\rho^2}(\mathbf{U}^\top \mathbf{n})_j$ denote the projected noise component. The noise term follows:

$$B_j \sim \mathcal{N}\left(0, \sigma_B^2\right), \quad \text{with } \sigma_B = \sqrt{\frac{1-\rho^2}{L}}. \tag{34}$$

Without loss of generality, let's consider $c_j = +1$. The probability of success is given by:

$$\mathbb{P}\left(S_j + B_j > 0\right) = \mathbb{P}\left(\alpha \frac{\rho}{\sqrt{M'}} + B_j > 0\right) \tag{35}$$

Consequently, applying the standard normal CDF:

$$p(\rho) = \Phi\left(\alpha \frac{\rho}{\sqrt{1-\rho^2}} \sqrt{\frac{L}{M'}}\right). \tag{36}$$

□

D.2 Proposition 3

Proposition 5 (SNW PCA security)**.** *A* SNW *system is ηL-secure against PCA attack, requiring at least $N = \eta L$ watermark observations to retrieve the watermark subspace, with:*

$$\frac{N}{L} = \eta = \max\left(1, \frac{\left(1 - \sqrt{\lambda_1 M'}\right)^2}{L\left(\sqrt{\lambda_1} - \frac{1}{\sqrt{L}}\right)^2}\right). \tag{37}$$

In particular, by setting α to $\alpha^* := \sqrt{\frac{M'}{L}}$ we have that $N \to \infty$ and the system is said to be perfectly secure against PCA attacks.

Proof. Let $\mathbf{v} \in \mathbb{S}^{L-1}$ be the directional embedding vector defined in Equation 5:

$$\mathbf{v} = \alpha \frac{\mathbf{U}\mathbf{c}}{\sqrt{M'}} + \sqrt{1-\alpha^2}\,\mathbf{q}', \quad \text{with} \quad \mathbf{q}' = \frac{(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}}{\|(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)\mathbf{q}\|_2}, \tag{38}$$

where $\mathbf{U} \in \mathbb{R}^{L \times M'}$ satisfies $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_{M'}$, $\mathbf{c} \in \{-1, 1\}^{M'}$, and $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L)$.

We can compute the following covariance matrix:

$$\mathbb{E}[\mathbf{q}'\mathbf{q}'^\top] = \frac{1}{L - M'}(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top), \tag{39}$$

$$\mathbb{E}[\mathbf{c}\mathbf{c}^\top] = I_{M'}, \tag{40}$$

$$\mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \frac{\alpha^2}{M'}\mathbf{U}\mathbf{U}^\top + \frac{1-\alpha^2}{L - M'}(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top). \tag{41}$$

Under the latent perturbation model $\tilde{\mathbf{v}} = \rho\mathbf{v} + \sqrt{1-\rho^2}\,\mathbf{n}$ with isotropic noise $\mathbf{n} \sim \mathcal{U}(\mathbb{S}^{L-2})$ independent of $\mathbf{v}$ (where $\mathbb{E}[\mathbf{n}\mathbf{n}^\top] = \frac{1}{L}\mathbf{I}_L$), the population covariance matrix of the watermarked latents is:

$$\mathbf{\Sigma} = \mathbb{E}[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^\top] = \rho^2\mathbb{E}[\mathbf{v}\mathbf{v}^\top] + (1-\rho^2)\mathbb{E}[\mathbf{n}\mathbf{n}^\top] \tag{42}$$

$$= \rho^2\frac{\alpha^2}{M'}\mathbf{U}\mathbf{U}^\top + \rho^2\frac{1-\alpha^2}{L-M'}(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top) + \frac{1-\rho^2}{L}\mathbf{I}_L. \tag{43}$$

Because $\mathbf{U}\mathbf{U}^\top$ and $(\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top)$ define mutually orthogonal projection operators, $\mathbf{\Sigma}$ is diagonalized in the basis of $\mathbf{U}$ and its orthogonal complement. It exhibits two distinct population eigenvalues:

$$\lambda_1 = \rho^2\frac{\alpha^2}{M'} + \frac{1-\rho^2}{L} \quad (\text{ for } M' \text{ dimensions}), \tag{44}$$

$$\lambda_2 = \rho^2\frac{1-\alpha^2}{L-M'} + \frac{1-\rho^2}{L} \quad (\text{ for } L-M' \text{ dimensions}). \tag{45}$$

According to the Marchenko-Pastur distribution of eigenvalues for random matrix theory, the support of the non-watermarked latent space is:

$$\mathcal{S}_0 = \left[\lambda_0\left(1-\sqrt{\frac{L}{N}}\right)^2, \lambda_0\left(1+\sqrt{\frac{L}{N}}\right)^2\right] \tag{46}$$

where N is the number of observations and $\lambda_0 = \frac{1}{L}$.

The support of the eigenvalues of the covariance matrix from the watermarked latent is:

$$\mathcal{S}_1 = \left[\lambda_1\left(1-\sqrt{\frac{M'}{N}}\right)^2, \lambda_1\left(1+\sqrt{\frac{M'}{N}}\right)^2\right] \tag{47}$$

We consider a model to be secure against PCA for $N = \eta L$ observations if $\mathcal{S}_1 \subseteq \mathcal{S}_0$.

If $\lambda_1 < \lambda_0$:

$$\lambda_1\left(1-\sqrt{\frac{M'}{N}}\right)^2 = \lambda_0\left(1-\sqrt{\frac{L}{N}}\right)^2 \tag{48}$$

$$\sqrt{\lambda_1} - \sqrt{\lambda_0} = \frac{\sqrt{\lambda_1 M'} - \sqrt{\lambda_0 L}}{\sqrt{N}} \tag{49}$$

$$\frac{N}{L} = \eta = \frac{\left(\sqrt{\lambda_1 M'} - 1\right)^2}{L\left(\sqrt{\lambda_1} - \frac{1}{\sqrt{L}}\right)^2} \tag{50}$$

If $\lambda_1 > \lambda_0$, the result is the same and the proof is analogous.

Finally, since forming a full-rank empirical covariance matrix in $\mathbb{R}^L$ requires at least $N \geq L$ independent observations, we obtain the security threshold:

$$\eta = \max\left(1, \frac{\left(1-\sqrt{\lambda_1 M'}\right)^2}{L\left(\sqrt{\lambda_1} - \frac{1}{\sqrt{L}}\right)^2}\right). \tag{51}$$

□

D.3 PROPOSITION 3 (PERFECT SECURITY)

Proposition 6 (SNW perfect PCA security)**.** *The perfect security regime of SNW against PCA estimation is achieved when:*

$$\alpha^* = \sqrt{\frac{M'}{L}} \tag{52}$$

Proof. Consider the population covariance matrix $\mathbf{\Sigma} = \mathbb{E}[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^\top]$ derived in Equation 43:

$$\mathbf{\Sigma} = \rho^2 \frac{\alpha^2}{M'} \mathbf{U}\mathbf{U}^\top + \rho^2 \frac{1-\alpha^2}{L-M'} (\mathbf{I}_L - \mathbf{U}\mathbf{U}^\top) + \frac{1-\rho^2}{L} \mathbf{I}_L. \tag{53}$$

The population spectrum is characterized by the two eigenvalues λ_1 on the M'-dimensional watermark subspace and λ_2 on the $(L-M')$-dimensional complementary subspace:

$$\lambda_1 = \rho^2 \frac{\alpha^2}{M'} + \frac{1-\rho^2}{L}, \qquad \lambda_2 = \rho^2 \frac{1-\alpha^2}{L-M'} + \frac{1-\rho^2}{L}. \tag{54}$$

Perfect security against PCA is reached when the population covariance matrix is strictly isotropic, which occurs if $\lambda_1 = \lambda_2$:

$$\rho^2 \frac{(\alpha^*)^2}{M'} + \frac{1-\rho^2}{L} = \rho^2 \frac{1-(\alpha^*)^2}{L-M'} + \frac{1-\rho^2}{L} \tag{55}$$

$$\iff \quad \alpha^* = \sqrt{\frac{M'}{L}}. \tag{56}$$

□

E WATERMARKING SECURITY AND ADVERSARIAL ROBUSTNESS

The concept of security in the modern post-hoc literature is nebulous. It is seldom discussed in papers dedicated to novel architectures. Adversarial robustness is discussed only as a mean to improve image quality (HiDDeN's strategy Zhu et al. (2018) or decoding performance (PIXELSEAL's adversarial training Souček et al. (2025a)). TRUSTMARK Bui et al. (2025) mentions neither security nor adversarial robustness in the whole paper. To the best of our knowledge SYNTHID-IMAGE is the *only* work of this type to mention and discuss watermarking security specifically Gowal et al. (2025)[Section 6].

Again, the SYNTHID-IMAGE report is instructive. It conflates genuine watermarking security (watermark forgery, removal, and secret extraction) with adversarial robustness and, more surprisingly, with model extraction vulnerabilities. The confusion is especially interesting, since the authors seem to imply that adversarial machine learning is the main tool for attacking a watermarking system.

Similarly, the literature specialized in the security of watermarking does not differentiate security when talking about

- Signal estimation and forgery based on classical watermarking analysis and signal processing Tarhini et al. (2026); Bas & Butora (2025).
- Spoofing and erasure attacks using adversarial machine learning techniques Gesny & Giboulot (2026a); Souček et al. (2025b); Jiang et al. (2023).

The first category is what would truly be called watermarking security in the classic literature. Citing the reference definition of Kalker Kalker (2001), the goal of these attacks is to obtain:

> [...] unauthorized [...] access to the raw watermarking channel. In other words, watermark security refers to the inability of unauthorized users to remove, detect and estimate, write, or modify the raw watermarking bits. In particular, watermark security is not concerned with the semantics of the watermarking bits, but solely with the physical presence of the watermarking bits.

The PCA attack studied in this paper falls into this category: estimating the key grants complete access to the watermarking channel, in the sense that any arbitrary image can now be spoofed with any codeword. This definition of security has a long history, culminating in the refined definitions of *equivocation* Cayre et al. (2005) and *effective key length* Bas & Furon (2013), with our definition being a simplification of the combination of the two.

This is in contrast to adversarial machine learning, which can be leveraged against a specific part of the watermarking system – the decoding projection. When they were not based on DNNs, such attacks were called "oracle attacks" or "sensitivity attacks" in the classical literature – for example, see Broken-Arrows' "snake traps" Furon & Bas (2008)[Section 5.2], or work from the early 1990's Cox & Linnartz (1997); Linnartz & Van Dijk (1998). They do allow to perform erasure and spoofing attacks. But they do not, by themselves, fully compromise the system. Their output is a single perturbation, not always transferable (across images, systems, models, etc...), that allows a specific operation (erasure, spoofing, ...

E.1 Adversarial robustness and Lipschitz networks

Adversarial robustness assesses how much the decoding projection function f against an adversarial pixel perturbation. In our scenario, such perturbations are crafted to spoof a watermark signal:

Definition 2 (Adversarial robustness). *Let $\Psi : \mathbb{R}^D \to \mathbb{R}^D$ be an attack such that:*

$$d_k(f_d(\Psi(\mathbf{x}))) = \mathbf{c}, \forall \mathbf{x} \in \mathbb{R}^D, \forall k \in \mathcal{K}, \forall \mathbf{c} \in \mathcal{A}^{M'} \tag{57}$$

We say that a (decoding) projection f_d is ϵ-adversarial-robust against Ψ iff :

$$\mathbb{E}\left[||\Psi(X) - X||_2\right] \leq \epsilon \tag{58}$$

where the expectation is taken over (X, C, K) triplets, with X sampled from the cover distribution $\mathcal{F}$, and (C, K) sampled uniformly from their respective set.

In theory, we are not required to evaluate the adversarial robustness of a projection function empirically. We can leverage the fact that DNNs are often considered to be Lipschitz functions. Formally, if a projection function f_d is L_{f_d}-Lipschitz with respect to the ℓ_2 norm, for any perturbation $\boldsymbol{\epsilon}$:

$$\|f_d(\mathbf{x} + \boldsymbol{\epsilon}) - f_d(\mathbf{x})\|_2 \leq L_{f_d} \|\boldsymbol{\epsilon}\|_2. \tag{59}$$

Consequently, inducing a target latent displacement $\|f_d(\mathbf{x}+\boldsymbol{\epsilon}) - f_d(\mathbf{x})\|_2$ required to cross a decision boundary necessitates a pixel-space distortion strictly lower-bounded by:

$$\|\boldsymbol{\epsilon}\|_2 \geq \frac{\|f_d(\mathbf{x} + \boldsymbol{\epsilon}) - f_d(\mathbf{x})\|_2}{L_{f_d}}. \tag{60}$$

When L_{f_d} is large, this theoretical lower bound vanishes, allowing imperceptible pixel noise to displace latents across the decision boundary.

E.2 Lipschitz estimation

Computing the global Lipschitz constant of a deep neural network is computationally intractable. We evaluate local stability by estimating the mean local Lipschitz constant across 100 randomly sampled images from the ImageNet dataset Deng et al. (2009) for each projection function f.

The local Lipschitz constant $L_{f_d}(\mathbf{x}_0)$ of f_d at a point $\mathbf{x}_0$ corresponds to the spectral norm of its Jacobian matrix $J_f(\mathbf{x}_0)$, which bounds the first-order sensitivity to local perturbations:

$$L_{f_d}(\mathbf{x}_0) = \|J_f(\mathbf{x}_0)\|_2 = \sigma_{\max}\left(J_{f_d}(\mathbf{x}_0)\right), \tag{61}$$

where $\sigma_{\max}$ is the largest singular value.

We estimate $\sigma_{\max}$ using the power iteration algorithm proposed by Miyato et al. (2018) because materializing the full Jacobian matrix is too computationally expensive for high-dimensional inputs such as images.

We report estimates of the Lipschitz constant for the watermarking systems studied in this paper in Table 4. For comparison, we also add Broken-Arrows Furon & Bas (2008) as a classical watermarking system that guarantees $L_f = 1$ by design through its Discrete Wavelet Transform (DWT), ensuring Euclidean distance conservation. In contrast, deep neural network projection functions are trained without Lipschitz constraints, leading to elevated Lipschitz constants and vulnerability to adversarial watermark erasure Gesny & Giboulot (2026a).

WM	Projection function	Lipschitz constant L_f
PIXELSEAL	ConvNeXT	1079
VIDEOSEAL	ConvNeXT	595
TRUSTMARK	ResNet	120
Broken-Arrows	DWT	1

Table 4: Comparison of empirical estimates of the Lipschitz constant L_f of the projection function across schemes. Classical transforms guarantee distance preservation ($L_f = 1$), whereas unconstrained neural projection functions exhibit large empirical Lipschitz constants, exposing them to low-distortion removal. The estimation is performed over 100 ImageNet images.

F WHITENING

Deep watermarking detectors yield biased and correlated output logits Gesny et al. (2026)[Appendix C.1.1]. To ensure a fair comparison across the methods, we follow the whitening procedure established in Gesny et al. (2026)[Appendix C.1.1]. For completeness, we summarize the methodology below.

For each detector ϕ, we compute the logits vector over $n = 10^6$ natural images from the MFlickr dataset. We first compute the empirical bias:

$$\mathbf{b}_\phi = \frac{1}{n}\sum_{i=1}^{n} \phi\left(\mathbf{x}^{(i)}\right), \tag{62}$$

and the covariance matrix:

$$\mathbf{\Sigma}_\phi = \frac{1}{n-1}\sum_{i=1}^{n} \left(\phi\left(\mathbf{x}^{(i)}\right) - \mathbf{b}_\phi\right)\left(\phi\left(\mathbf{x}^{(i)}\right) - \mathbf{b}_\phi\right)^\top. \tag{63}$$

Since $\mathbf{\Sigma}_\phi$ is symmetric positive semi-definite, we compute its eigendecomposition:

$$\mathbf{\Sigma}_\phi = \mathbf{V}_\phi \mathbf{\Lambda}_\phi \mathbf{V}_\phi^\top, \tag{64}$$

where $\mathbf{\Lambda}_\phi = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ contains the eigenvalues and $\mathbf{V}_\phi$ the corresponding orthonormal eigenvectors.

The projection matrix is then defined as:

$$\mathbf{W}_\phi = \mathbf{\Lambda}_\phi^{-\frac{1}{2}}. \tag{65}$$

Finally, the whitened detector output is given by:

$$\phi_w(\mathbf{x}) = \mathbf{W}_\phi\left(\phi\left(\mathbf{x}\right) - \mathbf{b}_\phi\right). \tag{66}$$

G RESIDUAL TRANSFER ATTACKS

A critical requirement for watermarking is the content-dependency: the watermark signal must be tied to the host image, preventing an adversary from spoofing the watermark by transplanting the residual from a watermarked content. Let $\mathbf{x}_i^{(1)} = \mathbf{x}_i + \boldsymbol{\delta}_i^{(1)}$ be the watermarked version of host image $\mathbf{x}_i$, where $\boldsymbol{\delta}_i^{(1)}$ is the residual watermark embedded in the pixel space.

Method	Rate R_σ / $R_\sigma \times M'$	
	Identity	Residual Transfer
PixelSeal	0.910 / 232.9	0.003 / 0.8
VideoSeal	0.848 / 217.2	0.002 / 0.6
Trustmark	0.506 / 50.6	0.059 / 5.9
SNW (w/o anti-spoofing)	0.883 / 678.0	0.113 / 86.5
SNW (w anti-spoofing)	0.888 / 682.2	0.000 / 0.2

Table 5: Bit accuracy of the methods against residual transfer attacks. The results are computed over 200 MFlickr 1024×1024 images.

While vanilla SNW exhibits residual leakage across hosts, we fix this by fine-tuning the model using the following loss:

$$\mathcal{L}_{\text{spoof}} = \frac{1}{B}\sum_{i=1}^{B}\langle f(\mathbf{x}^{(j)} + \mathbf{w}_1^{(i)}), \mathbf{v}_1\rangle^2, \tag{67}$$

where $j = i + 1 \mod B$, and $\mathbf{v}_1$ is the target vector of image $\mathbf{x}^{(i)}$.

As shown in Table 5, SNW is vulnerable to residual transfer attacks. TrustMark is also vulnerable, whereas PixelSeal and VideoSeal are robust. Fine-tuning SNW using the anti-spoofing loss 67 suppresses transferability while preserving decoding capacities.

H Additional results

H.1 SNW geometric analysis

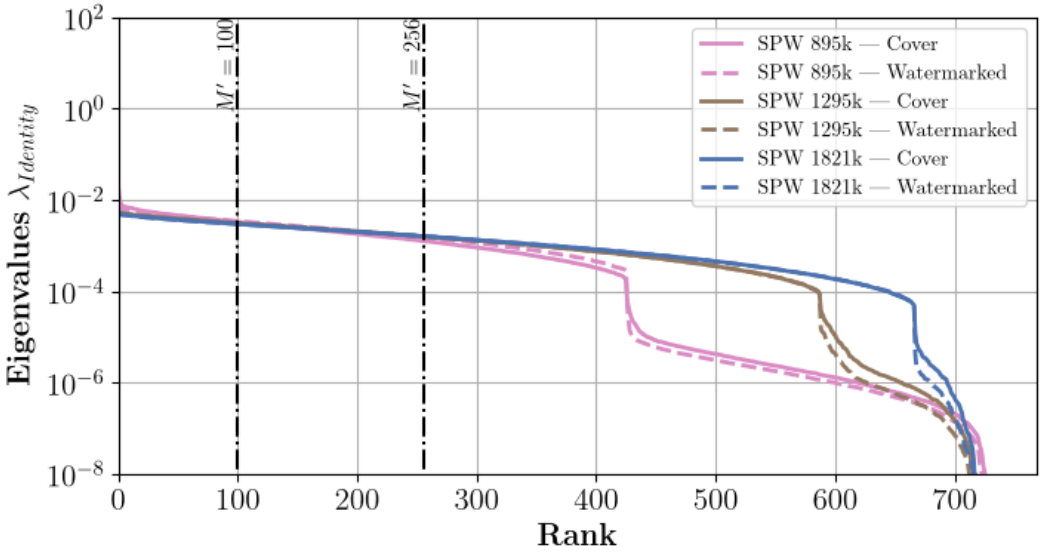


Figure 5: Eigenvalues of the covariance Σ_{Identity} of the latent vector $f_d(\mathbf{x}_{\text{wm}})$ (i.e. when t is the identity) for different checkpoint of SNW. The number in the legend correspond to the number of training step.

We replicate the eigenvalue study in Section 3 for different checkpoints of SNW in Figure 5. One can observe our training pipeline design to have succeeded for two reasons:

1. As the number of steps increase, we ”fill up” the number of robust components, i.e the non-zero eigenvalues.

2. The eigenvalues between cover and watermarked content cannot be distinguished. This validates our security analysis: by setting $\alpha = 1$ for SNW, one cannot recover the secret key through any PCA or other second-order attack: indeed one cannot distinguish between the covariance of a cover and of a watermarked image.

H.2 Recent erasing attacks

In this section, we compare the capacity of modern post-hoc watermarking schemes against recent erasing attacks. The erasing attacks studied are:

- **VAE Purification** Nie et al. (2022) Encode and decode the watermarked image with a frozen VAE.
- **DiffPure** Nie et al. (2022) Invert the last t steps of diffusion and regenerate them.
- **WM Forger** Soucek et al. (2025) Use a preference model to perform a gradient ascent on the image.

Method	Identity	VAE	Sana (2 steps)	Sana (4 steps)	Sana (8 steps)
PixelSeal	0.939 / 240.5	0.555 / 142.0	0.519 / 132.8	0.357 / 91.3	0.076 / 19.4
VideoSeal	0.884 / 226.2	0.576 / 147.6	0.557 / 142.6	0.392 / 100.4	0.084 / 21.5
TrustMark	0.632 / 63.2	0.453 / 45.3	0.415 / 41.5	0.184 / 18.4	0.023 / 2.3
SNW	0.892 / **684.7**	0.495 / **380.3**	0.479 / **368.1**	0.388 / **298.3**	0.145 / **111.5**
Method	Sana (10 steps)	Sana (20 steps)	WM Forger 50 steps	WM Forger 100 steps	
PixelSeal	0.025 / 6.4	0.003 / 0.8	0.533 / 136.3	0.280 / 71.8	
VideoSeal	0.030 / 7.8	0.003 / 0.9	0.385 / 98.5	0.191 / 48.8	
TrustMark	0.0152 / 1.5	0.007 / 0.7	0.187 / 18.7	0.078 / 7.8	
SNW	0.068 / **52.2**	0.001 / 0.8	0.312 / **239.5**	0.139 / **106.5**	

Table 6: Capacity of the watermarking systems against recent watermarking erasure attacks. The results are computed over 200 MFlickr 1024×1024 images.

H.3 Detailed results

In this section, we provide the capacity of modern post-hoc watermarking methods against a full benchmark of classic transformations.

H.4 Examples

In this section, we provide a qualitative example of images watermarked with SNW.

Method	Rate R_σ / $R_\sigma \times M'$				
	Identity	**Brightness** $+0.2$	**Contrast** $\times 2$	**JPEG QF** $= 80$	**JPEG QF** $= 50$
PixelSeal	0.939 / 240.5	0.899 / 230.1	0.709 / 181.6	0.928 / 237.5	0.876 / 224.4
VideoSeal	0.878 / 224.7	0.793 / 203.1	0.539 / 138.1	0.863 / 221.0	0.816 / 209.0
TrustMark	0.611 / 61.1	0.530 / 53.0	0.309 / 30.9	0.545 / 54.5	0.491 / 49.1
SNW	0.983 / **686.0**	0.868 / **666.3**	0.514 / **394.6**	0.884 / **679.0**	0.837 / **642.5**
	Gaussian Blur $3 \times 3, \sigma = 1$	**Rotation** $90°$	**Horizontal Flip**	**Hue** 0.5	**Saturation** 1.5
PixelSeal	0.939 / 240.3	0.863 / 221.0	0.940 / 240.6	0.736 / 188.5	0.928 / 237.6
VideoSeal	0.877 / 224.5	0.821 / 210.1	0.866 / 221.6	0.652 / 166.8	0.858 / 219.5
TrustMark	0.610 / 61.0	0.007 / 0.7	0.612 / 61.2	0.389 / 38.9	0.573 / 57.3
SNW	0.892 / **685.1**	0.633 / **486.5**	0.830 / **637.7**	0.706 / **542.6**	0.878 / **674.5**
	Crop 90%	**Crop** 80%	**Crop** 70%	**Crop** 60%	**Crop** 50%
PixelSeal	0.910 / 232.9	0.890 / 227.7	0.845 / 216.4	0.752 / 192.5	0.561 / **143.7**
VideoSeal	0.768 / 196.7	0.636 / 162.9	0.389 / 99.6	0.131 / 33.6	0.019 / 4.8
TrustMark	0.694 / 69.4	0.538 / 53.8	0.016 / 1.6	0.008 / 0.8	0.007 / 0.7
SNW	0.811 / **623.2**	0.725 / **556.6**	0.567 / **435.6**	0.351 / **269.6**	0.133 / 102.0
	Resize 0.5	**Median Filter** 3×3			
PixelSeal	0.937 / 239.9	0.938 / 240.1			
VideoSeal	0.876 / 224.2	0.877 / 224.4			
TrustMark	0.610 / 61.0	0.611 / 61.1			
SNW	0.891 / **684.1**	0.892 / **685.0**			

Table 7: Capacity of the watermarking systems against an extensive set of classic image transformations. The results are computed over 1000 MFlickr 1024×1024 images.

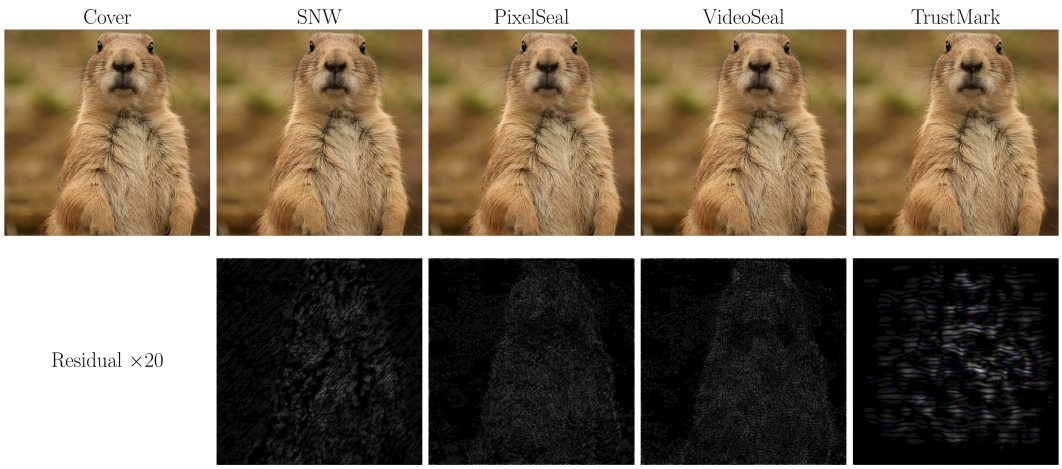


Figure 6: Example of an image watermarked with different methods and associated residuals for a fixed watermark power of 48 dB PSNR.