A Secure Image Watermarking Framework with Statistical Guarantees via Adversarial Attacks on Secret Key Networks

Feiyu Chen[1], Wei Lin[1], Ziquan Liu[2], and Antoni B. Chan[1]

[1] Department of Computer Science, City University of Hong Kong, Hong Kong, PRC
{feiyuchen3-c@my., wlin38-c@my., abchan@}cityu.edu.hk
[2] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
ziquan.liu@qmul.ac.uk

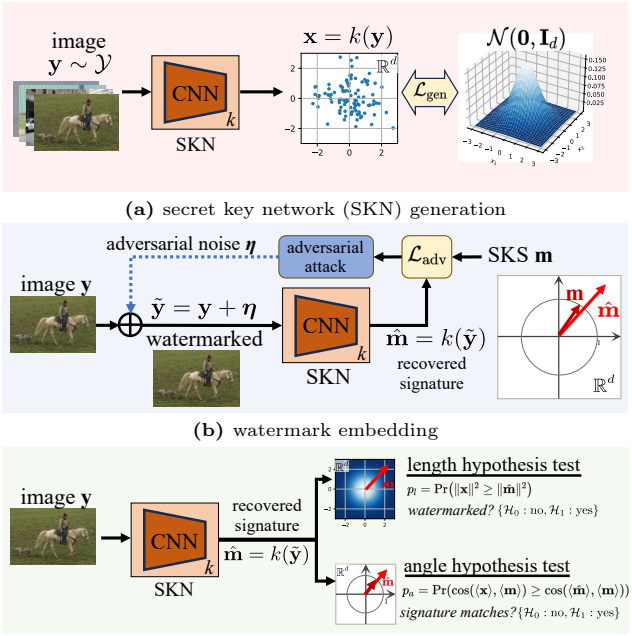
Abstract. Imperceptible watermarks are essential in safeguarding the content authenticity and the rights of creators in imagery. Recently, several leading approaches, notably zero-bit watermarking, have demonstrated impressive imperceptibility and robustness in image watermarking. However, these methods have security weaknesses, e.g., the risk of counterfeiting and the ease of erasing an existing watermark with another watermark, while also lacking a statistical guarantee regarding the detection performance. To address this issue, we propose a novel framework to train a secret key network (SKN), which serves as a non-duplicable safeguard for securing the embedded watermark. The SKN is trained so that natural images' output obeys a standard multi-variate normal distribution. To embed a watermark, we apply an adversarial attack (a modified PGD attack) on the image such that the SKN produces a secret key signature (SKS) with a longer length. We then derive two hypothesis tests to detect the presence of the watermark in an image via the SKN response magnitude and the SKS angle, which offer a statistical guarantee of the false positive rate. Our extensive empirical study demonstrates that our framework maintains robustness comparable to existing methods and excels in security and imperceptibility.

Keywords: Zero-bit watermark · Adversarial attack · Hypothesis test

1 Introduction

With the advancement of image-editing [18, 24] and generative models [30], watermarking technology has garnered increasing attention from researchers as a means to protect image rights and verify the authenticity of image sources [5, 29]. Watermarking achieves this by embedding unique identifiers into images through imperceptible modifications that do not compromise the aesthetics of images.

Most existing watermarking methods are based on either: 1) traditional methods [6, 11, 31, 43], which can provide nice theoretical guarantees on detector performance but are less secure due to their usage of *known* linear embedding



(a) secret key network (SKN) generation

(b) watermark embedding

(c) watermark detection

Fig. 1: Our secret-key watermarking framework. **(a)** the SKN is trained so that its output follows a standard multi-variate normal (SMVN) distribution given an input image distribution; **(b)** given an image, the watermark is generated using adversarial attack that makes the SKN output the desired secret key signature (SKS) with extended length; **(c)** the signature is recovered by applying the SKN to the image, and the watermark is detected using hypothesis tests derived from the assumed SMVN distribution of the SKN.

functions; or 2) deep learning methods [1, 13, 17, 23, 25, 45] that use non-linear embedding/detection functions (deep neural networks, DNNs) to improve detection performance, but do not have any detection guarantees. Furthermore, because these encoder/decoder frameworks are trained end-to-end, the mechanisms learned to embed and detect the watermark are obfuscated. However, since such DNNs could be kept secret (i.e., not publicly available), the security of the watermark is higher than traditional methods.

To simultaneously obtain a high detection rate, high invisibility, and high secrecy, our paper proposes to combine traditional and deep-learning methods – we propose a new watermarking framework that combines a statistical detection framework with a secret-key DNN that generates zero-bit watermarks using adversarial attack (see Fig. 1). Specifically, we train a DNN (a non-linear function mapping from an image to a vector) to imbue its output with known statistical properties. We denote this DNN as a *secret key network* (SKN). To watermark an image, we then use adversarial attack [32] on the SKN so that the adversarial image produces a desired *secret key signature* (SKS), a unique vector identi-

fier. Watermark detection is achieved using hypothesis tests, which leverage the statistical properties of the trained SKN, providing both statistical guarantees and interpretability of the detector (as in traditional approaches). The trained SKN is unique and kept secret, ensuring the security of the watermark, and its non-linear mapping allows for a high detection rate.

Our experiments assess three key factors in watermarking performance: *imperceptibility*, *robustness*, and *security*. Tab. 1 compares our method with other zero-bit DNN techniques, DNN0B [35] and SSLWM [10]. We address three potential security threats to watermarking through targeted experiments, and the results indicate that our method significantly enhances security. The security of our framework stems from each SKN's uniqueness, which enables watermarked images generated by one SKN to be undetectable by another SKN with the same architecture but different weights. Importantly, faking a private SKN is challenging because a well-trained model consists of a 100 million parameters, which are randomly initialized, while the training objective is loose (only supervising the output distribution, not specific input-output pairs), thus leading to significant deviations between the learned functions of different SKNs. Meanwhile, attackers would also need to know the specific structure of SKN. For imperceptibility, we compare the quality of watermarked images to their originals. For the same target PSNR of 32, our method surpasses others in image quality metrics, such as SSIM. In terms of robustness, our method achieves comparable detection rates to other methods when the watermarked images undergo different perturbations. Despite using adversarial attack for embedding the watermark, our method runs faster than other zero-bit DNN methods due to its smaller backbone and ability to be completely parallelized in GPU (see Tab. 1). Finally, our experiments also verify that the well-trained SKN has obtained the required normality in its output, which is important for the statistical guarantee (calibration) of the hypothesis tests, while others [10, 35] do not obtain such verification. [3]

In summary, our contributions are as follows:

1. We introduce a new framework using adversarial attack for watermarking, integrating the advantages of traditional and deep learning techniques.
2. We propose to train a secret-key network (SKN) to serve as the non-linear mapping function for watermarking images, whose outputs are imbued with known statistical properties. In the detection phase, we propose two hypothesis tests, on the length and on the angle, for detecting SKN and SKS in watermarked images. The hypothesis tests offer a statistical guarantee, as well as explainability of the detector.
3. The experiment shows that our method produces more secure and imperceptible watermarks while maintaining robustness against image distortions.

[3] Our code is available at https://github.com/FelixFeiyu/ECCV2024-AA-WM

Property	Test Case	Metric	DNN0B [36]	SSLWM [10]	Ours
Security	Random-fake	False Rate ↓	**0.08%**	12.84%	1.94%
	Model-fake	False Rate ↓	100%	100%	**4%**
	WM-Remove	Resistance Level ↑	2	3	**>5**
Imperceptibility	Image Quality	SSIM↑	0.9103	0.9026	**0.9768**
Robustness	Gaussian noise	Avg DR↑	92.20%	40.88%	**99.02%**
	Gaussian blur		81.88%	88.33%	**99.40%**
	Rotation		84.36%	**86.43%**	85.89%
	Cropping		**97.90%**	85.51%	90.58%
	JPEG		95.65%	78.12%	**99.22%**
Runtime	watermarking	s/image	1.18	7.13	0.67
	detecting		0.013	0.013	0.005

Table 1: Summary of comparisons with other zero-bit methods: assessing watermarking on security, imperceptibility, robustness, and runtime. "False Rate" is the success percentage of fake signatures (generated randomly or via the watermarking model) that match true embedded signatures of watermarked images. "Resistance Level" denotes the number of watermarks that can be recursively embedded before the detection rate of the first watermark falls below 50%. "Avg DR" is the average detection rate for each image distortion across various parameters.

2 Related Work

2.1 Watermarking Techniques

Imperceptible watermarking aims to embed unique identifiers into images and is crucial in protecting image copyrights and verifying an image's provenance [37].

Traditional methods. Most traditional methods are based in the frequency domain, e.g., leveraging the Fourier-Mellin transform [28], discrete Wavelet transform [19] or SVD-based transform [4]. Although frequency-based approaches often obtain better hiding ability and robustness, some works explore more direct approaches in the spatial domain (e.g., [43]). Compared to traditional methods, our work embeds watermarks in the spatial domain by subtly modifying the image's pixels using adversarial attacks (AA) on DNNs. The DNN essentially serves as a non-linear embedding function for the watermark, and the imperceptibility is guaranteed through AA's perturbation constraint.

Deep learning methods. Recently, convolutional neural networks (CNNs) have been applied to watermark images using end-to-end frameworks. HiDDeN [45] is an end-to-end trained CNN that uses encoder and decoder networks to embed and extract the watermark. Subsequent works enhanced robustness through training with simulated image attacks [1, 25] and 2-stage training [23].Recent works also modify generative image models to produce watermarked images [9, 38, 44]. Wen *et al.* [38] embeds a watermark by modifying each denoising step of the diffusion model. A related area is steganography, which aims to hide a secret message inside an image [2,13,17,39,42]. While these works obtain good performance and secrecy, since the trained encoder/decoder CNN pairs are unique, their watermark detectors lack statistical guarantees and interpretability due to the black-box nature of end-to-end trained CNNs. In con-

trast, our approach maintains high security due to the uniquely trained CNN, while also offering detector interpretability and statistical guarantees due to our hypothesis testing approach. In terms of architecture, previous deep learning methods use encoder/decoder CNN pairs to embed and extract the watermark. In contrast, our approach uses a single CNN as a non-linear extraction function and an adversarial attack on the CNN as the embedding function.

Zero-bit watermarking. Most of the aforementioned methods assume the hidden watermark as a message composed of words or bits. In contrast, "zero-bit" (ZB) watermarking is only concerned with detecting a watermark's presence or absence without message recovery [6, 11, 31]. Traditional methods for ZB watermarking embed a real vector (a key signature) into the image using a linear embedding function (e.g., frequency-domain transformations) and then derive theoretically optimal methods for detecting the presence/absence of the watermark, contrasting with other methods [1, 25, 45] that use a binary vector to represent a message and use a decoder to recover it. Recent works [10, 35, 36] replace the linear extracting function for ZB watermarking with a CNN pre-trained on the ImageNet image classification task, where the feature vector in the penultimate layer serves as the embedding space for the vector signature.

Similar to our approach, [10, 35, 36] use an adversarial attack to embed the signature into the image. However, there are three crucial differences regarding *security*, *capability*, and *detector guarantees*. First, other ZB methods are based on *known* embedding functions (either linear frequency transforms or pre-trained CNNs), which leaves them vulnerable to signature stealing or signature overwriting (since the embedding function and its inverse are known), and thus lack security (see §4.6). In contrast, we regard the DNN itself as a secret key (i.e., SKN), which enhances our framework's security. We can generate distinct SKNs based on different random seeds, and the signatures embedded with one SKN are unrecognizable by another SKN, maintaining the detectability of the original watermark even after multiple overlaps (see §4.6).[4] Second, our approach employs two signatures, the *network* SKN and the *vector* SKS, which provide two complementary methods to secretly embed information into the image via the SKN's output length and output direction. Correspondingly, we use two complementary hypothesis tests, based on length and angle, to detect the watermark. In contrast, other ZB methods only use a single vector signature and an angle hypothesis test. Third, because we train our SKN's output to adhere to a Gaussian distribution, we obtain better-calibrated detector guarantees (see §4.2) than the pre-trained CNN approaches [10, 35, 36], which can only *approximate* a Gaussian distribution by matching the 1st and 2nd moments via feature whitening.

[4] Here all discussions on security are under the assumption that the keys/identifiers for decoding watermarked images will be securely stored and processed by a hosting platform (see Supp. S8). If the platform were to leak the keys/identifiers to adversaries, then all existing methods (including ours) would lose their security.

2.2 Adversarial Attacks (AA)

AAs aim to inject subtle noise into an image in order to alter the prediction of a DNN, e.g., to produce a misclassification [21, 26, 41]. The concept of an adversary can be extended to improving the robustness of watermarks. Improving on [45], [25] used adversarial samples in the DNN's training stage to enhance the watermark's robustness against a set of image distortions. Adversarial noise is also employed defensively [15,34], safeguarding images against malicious edits by generative models. [16] leveraged adversarial training to find the optimal position and transparency of visible watermarks for copy protection. In contrast to these methods, which use AA for model training, our approach directly leverages AA *to generate the watermark as adversarial noise.*

3 Watermarking Framework

In this section, we propose a new watermarking framework that combines a statistical detection framework with a secret-key DNN and adversarial attack. As summarized in Fig. 1, our framework is composed of three stages: 1) secret key network generation; 2) watermark embedding; 3) watermark detection. In the first stage (Fig. 1a), we train a DNN as a *secret key network* (SKN) so that its output distribution is a standard multivariate normal (SMVN) distribution when given an input distribution of clean images. In the watermark embedding stage (Fig. 1b), we apply an image as the input to the SKN and use an adversarial attack on the image to create the watermarked image. We generate a *secret key signature (SKS)* as a unit vector, which serves as a unique identifier for the watermark. The goal of the adversarial attack is to make the SKN output in the same direction as the SKS, with the length extended such that it is unlikely to be a sample from the SMVN. In the watermark detection stage (Fig. 1c), we apply the SKN to the image to obtain the recovered signature, and then use two complementary hypothesis tests to detect the presence of the watermark. The first hypothesis test works on the length of the recovered signature (denoted as HT4L), testing if the vector is unlikely to be a sample from the assumed SMVN for typical images. The second hypothesis test works on the angle (HT4A), testing if the direction of the recovered signature matches the original SKS.

Note that in our framework, we have two secret keys: a well-trained CNN whose output vector should follow an SMVN distribution (SKN) and a real vector (SKS). We next describe each stage in detail.

3.1 Secret Key Network Generation

For the SKN architecture, we select ResNet18 [14] and modify its final fully-connected layer to use linear activation, thus enabling a mapping from an input image $\mathbf{y} \in \mathbb{R}^n$ to a real vector $\mathbf{x} \in \mathbb{R}^d$. Here, d represents the dimension of the watermark space (e.g., 32), and n is the size of the image. Given an input distribution of images $\mathcal{Y}$, we require that the SKN output follows an SMVN

distribution, i.e., $\mathbf{x} = k(\mathbf{y}) \sim \mathcal{N}(0, \mathbf{I}_d), \mathbf{y} \sim \mathcal{Y}$. To achieve this, the parameters $\boldsymbol{\theta}$ of SKN $k(\cdot)$ are trained to minimize the loss,

$$\mathcal{L}_{\text{gen}} = \lambda_1 \mathcal{L}_{\text{w}} + \lambda_2 \mathcal{L}_{\text{v}}, \tag{1}$$

where $\mathcal{L}_{\text{w}}$ is the Wasserstein loss between the output distribution and the SMVN, $\mathcal{L}_{\text{v}}$ is a loss on the output variances, and λ_1, λ_2 are weighting hyperparameters.

Loss $\mathcal{L}_{\text{w}}$ steers the output vector $\mathbf{x}$ to follow the desired SMVN, and is based on the Wasserstein distance [8,12,27] between two distributions (see Supp. S1),

$$\mathcal{L}_{\text{w}} = \boldsymbol{\mu}_d^T \boldsymbol{\mu}_d + \text{tr}(\boldsymbol{\Sigma}_d) + d - 2\,\text{tr}(\boldsymbol{\Sigma}_d^{\frac{1}{2}}), \tag{2}$$

where $(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$ are the mean and covariance of $k(\mathbf{y})$ for a mini-batch $\{\mathbf{y}\} \subset \mathcal{Y}$. The loss $\mathcal{L}_{\text{v}}$ improves the convergence of each dimension of $\mathbf{x}$ to unit variance,

$$\mathcal{L}_{\text{v}} = ||\,\text{diag}(\boldsymbol{\Sigma}_d) - \mathbf{1}_d||_1, \tag{3}$$

where $\text{diag}(\cdot)$ extracts the matrix diagonal, $\mathbf{1}_d$ is a vector of d ones, and $||\cdot||_1$ is the L1-norm.

We use large image datasets (e.g., MSCOCO [22]) for training the SKN. Note that $k(\cdot)$ defines a secret non-linear manifold space in which the watermark is embedded. SKNs generated with different architectures or initial seeds will result in different non-linear manifold spaces. We select ResNet18 for the SKN since it is an efficient and uncomplicated CNN (other architectures could also be used).

3.2 Watermark Embedding

With the well-trained SKN, we use AA to create the watermark by adding imperceptible noise into an image. In our framework, the watermarking should achieve two goals simultaneously. First, the AA watermark should make the SKN produce an output vector that is unlikely to be drawn from its assumed SMVN (for clean images) - the longer the output vector, the more unlikely it is, and thus the stronger the watermark. Second, the AA should make the SKN output vector in the same direction as the SKS.

Specifically, given an image $\mathbf{y}$, the watermark is embedded using AA on $k(\mathbf{y})$, resulting in the watermarked image $\tilde{\mathbf{y}} = \mathbf{y} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is the adversarial noise. To achieve the two design goals on the SKN output $\hat{\mathbf{m}} = k(\tilde{\mathbf{y}})$ for the adversarial image, we devise a specific adversarial loss,

$$\mathcal{L}_{\text{adv}} = \lambda_3(\|\hat{\mathbf{m}}\|_2^2 - t_l)^2 + \lambda_4\left(t_a - \cos(\hat{\mathbf{m}}, \mathbf{m})\right)^2, \tag{4}$$

where the 1st term (denoted as $\mathcal{L}_{\text{len}}$) controls the length of $\hat{\mathbf{m}}$ to match a target length t_l, and the 2nd term (denoted as $\mathcal{L}_{\text{agl}}$) controls the angle between $\hat{\mathbf{m}}$ and SKS $\mathbf{m}$ to be a target cosine value t_a, and λ_3, λ_4 are the loss weights. In practice, we set the target length as $t_l = 63$ (equivalent to $p = 10^{-4}$ in the length hypothesis test), and $t_a = 1$ so that the angle between $\hat{\mathbf{m}}$ and $\mathbf{m}$ is shrunk to 0.

We use a modified version of PGD [26] as our AA for watermarking images, where the gradient value (instead of its sign) is used to update the adversarial noise. The perturbation ϵ-bound is measured with L_2-norm, which is equivalent to mean-squared error and related to PSNR. Thus, ϵ directly controls the PSNR of the watermarked image. In addition, to further improve the robustness of the watermark to image transformations, a data augmentation module (DA) from [10,35,36] is adapted into our watermarking process. See details in Supp. S3.1.

3.3 Watermark Detection

To detect the watermark in an image $\tilde{\mathbf{y}}$, the SKN is applied to the image to obtain the recovered signature $\hat{\mathbf{m}} = k(\tilde{\mathbf{y}})$. We propose two hypothesis tests focusing on length and angle metrics (denoted as HT4L and HT4A) to confirm that the watermark was generated by the SKN-SKS pair.

HT4L. Since the SKN output for clean images follows an SMVN, we devise a hypothesis test to detect the presence of a watermark produced by the SKN. The null hypothesis $\mathcal{H}_0$ is that the image does not contain a watermark (the $\hat{\mathbf{m}}$ is a sample from the SMVN), and the alternative hypothesis $\mathcal{H}_1$ is that the image contains a watermark ($\hat{\mathbf{m}}$ is unlikely to be a sample from the SMVN). Since $\mathbf{x}$ is distributed as an SMVN, the test statistic is $||\mathbf{x}||^2$ and follows a χ^2 distribution with d degrees of freedom, and the p-value of observing a test statistic as extreme as $||\hat{\mathbf{m}}||^2$, i.e., $p_l = \Pr(||\mathbf{x}||^2 \geq ||\hat{\mathbf{m}}||^2)$, can be calculated as [40]

$$p_l = 1 - \int_0^{||\hat{\mathbf{m}}||^2} \frac{1}{2^{d/2}\Gamma(d/2)} t^{d/2-1} e^{-t/2}\, dt, \tag{5}$$

where the $\Gamma(d/2)$ is the gamma function [20].

HT4A. We devise a hypothesis test that the recovered signature vector $\hat{\mathbf{m}}$ matches the pre-defined SKS $\mathbf{m}$. Here, the null hypothesis $\mathcal{H}_0$ is that $\hat{\mathbf{m}}$ and $\mathbf{m}$ do not match directions (as they are randomly sampled from the SMVN), while the alternative hypothesis $\mathcal{H}_1$ is that they have the same direction. The test statistic is the normalized vector $\langle \mathbf{x} \rangle = \mathbf{x}/||\mathbf{x}||$, which ideally follows a uniform distribution on the unit hypersphere S^{d-1}, and thus the p-value of observing $\langle \hat{\mathbf{m}} \rangle$ or better can be derived, $p_a = \Pr\left(\cos\left(\langle \mathbf{x} \rangle, \langle \mathbf{m} \rangle\right) \geq \cos\left(\langle \mathbf{x} \rangle, \langle \mathbf{m} \rangle\right)\right)$. By using multifold integration and the geometry of high-dimensional spheres, the p-value can be derived (see Supp. S2),

$$p_a = 1 - \frac{1}{\pi}\left[\theta - \cos\theta \sum_{k=1}^{\frac{d-2}{2}} \frac{(2k-2)!!}{(2k-1)!!} \sin^{(2k-1)}(\theta)\right], \tag{6}$$

where $\theta = \arccos(\langle \mathbf{m} \rangle, \langle \hat{\mathbf{m}} \rangle)$ is the angle between $\mathbf{m}, \hat{\mathbf{m}}$.

Combined hypothesis test. Combining the p-values p_l, p_a can test whether the image is watermarked by SKN k with SKS $\mathbf{m}$. Using the combination method from [33], the p-value for the combined hypothesis test is

$$p_c = p_l \cdot p_a \cdot \left(1 - \ln\left((p_l \cdot p_a)\right)\right. . \tag{7}$$

Statistical guarantees. If the calculated p-value (e.g., p_c) is less than a predetermined significance level α (usually 0.05), then we reject the null hypothesis $\mathcal{H}_0$ (watermark absent) in favor of the alternative hypothesis $\mathcal{H}_1$ (watermark present). The significance level α acts as a benchmark for decision-making and is equal to the false positive (Type I) error rate, i.e., detecting the watermark even though none is actually present. Thus by selecting α, we can obtain a watermark detector with a certain false positive rate.

4 Experiments

In this section, we demonstrate the effectiveness of the proposed approach through experiments on: SKN normality (§4.2), detection performance (§4.3), impercep-

tibility (§4.4), robustness (§4.5), and security (§4.6). We also conduct ablation studies in §4.7. A summary of our important results is presented in Tab. 1.

4.1 Experimental Setup

We use MSCOCO [22] as our training and test dataset, which is also adopted by our baselines [10, 36, 45]. MSCOCO comprises 118k training images and 5k test images. We adopt ResNet18 [14] as our SKN while changing its output layer to contain 32 neurons with linear activations. The SKN is trained on the MSCOCO training set. Training details are presented in Supp. S5.

The SKS is generated to be in the same semi-hemispherical domain in the 32-dim space as the natural response of the SKN on the given image, i.e., their angle is less than 180°. This way ensures that the AA does not need to perform drastic changes to the image, improving watermark invisibility and maintaining image corruption within the specified range. A single SKS could also be used for all images with minimal effect on detection accuracy (see Supp. S10).

For the AA, we use the modified PGD with perturbation bound with $\epsilon = 6.3 \times 10^{-4}$, corresponding to an average PSNR of 32 for the watermarked images. We measure the success detection rate (SDR) by computing the percentage of successfully detected watermarked images over all the test images (each test image is watermarked). A successful detection refers to a watermark image that obtains a p-value from the hypothesis test that is lower than the specified confidence level. We measure image quality via PSNR, SSIM, MAE, and RMSE.

We compare our model with two recent zero-bit deep-learning methods, DNN0B [36] and its self-supervised variant (SSLWM) [10]. [10] proposes both zero-bit and multi-bit methods, and here we only consider the zero-bit version for fair comparison. To calculate the SDR for DNN0B and SSLWM, we set the significance level at 0.05, aligning with commonly used statistical thresholds. We also compare with the recent DNN-based method, HiDDeN [45][5]. We follow [45] and define a successful watermark detection as having a bit error rate lower than 0.05 (a level deemed correctable in communication transmissions).

4.2 Normality of Secret Key Network

We first evaluate the normality of the SKN's output by analyzing the covariance matrix and mean vector of a batch of outputs. Quantitative and qualitative results indicate that the SKN output closely follows the desired SMVN distribution. We further verify the effectiveness of the proposed variance loss term $\mathcal{L}_{\mathrm{v}}$. Lastly, using three hypothesis tests for normality, we demonstrate that the SKN achieves normality, unlike DNN0B and SSLWM. See the results in Supp. S7.

[5] Note that [16] improves the robustness of [45] to unknown distortions. Since [16] is a closed-source project, we can only compare with its original version [45].

	watermarked images				clean images			
	mean	std	< 0.05	< 0.01	mean	std	< 0.05	< 0.01
Length	0.0123	0.0277	95.74%	73.02%	0.5227	0.2804	4.28%	1.18%
Angle	0.0109	0.0376	93.60%	85.08%	0.4958	0.2285	5.06%	0.96%
Combined	0.0011	0.0091	99.48%	98.28%	0.5140	0.2286	4.96%	1.28%

Table 2: Watermark detection performance. For both watermarked and clean images, the mean and standard deviation (std) of the detector's p-values are shown, as well as the percentage of p-values below significance level $\alpha = 0.05$ and $\alpha = 0.01$.

	PSNR ↑	SSIM ↑	MAE ↓	RMSE ↓
HiDDeN [45]	31.66	0.9531	0.0204	0.0267
DNN0B-32 [36]	32.06	0.9103	0.0195	0.0250
SSLWM-32 [10]	32.11	0.9026	0.0192	0.0248
Ours-32	**32.56**	**0.9768**	**0.0134**	**0.0237**
DNN0B-42 [36]	41.81	0.9859	0.0057	0.0081
SSLWM-42 [10]	41.81	0.9878	0.0061	0.0081
Ours-42	**42.00**	**0.9972**	**0.0038**	**0.0079**

Table 3: Comparison of watermark imperceptibility at target PSNRs of 32 and 42.

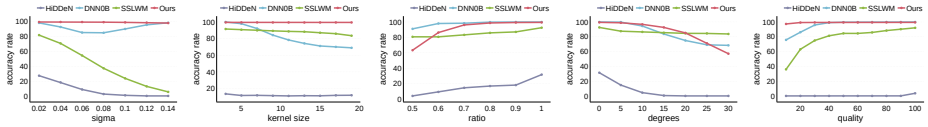
4.3 Detection Performance

We next analyze the performance of the detector's hypothesis tests. We calculate the mean and standard deviation of the three p-values (p_l, p_a, and p_c) across all watermarked images and record the percentage of p-values below significance levels of 0.05 and 0.01. Additionally, we compare these results with those from clean images to highlight the detection performance and verify our false positive error rate. The results are presented in Tab. 2, revealing successful watermarking, which is evident from the low mean p-values and nearly 100% detection rate on watermarked images. The results on clean images confirm a false positive rate that matches the desired significance level, showing the soundness of our method.

4.4 Imperceptibility Analysis

We evaluate the watermark invisibility by measuring the image quality of watermarked images against the original images. Our method can control the image quality by setting the ϵ-bound of the perturbation, while DNN0B and SSLWM can target specific PSNR/SSIM values. In contrast, HiDDeN encourages imperceptibility using a discriminator network without a preset quality target. The trained HiDDeN model obtains a PSNR of 32, and thus, for a fair comparison, we set ours, DNN0B, and SSLWM to produce the same PSNR of 32. We also evaluate at a higher PSNR of 42.

Quantitative and qualitative comparisons are presented in Tab. 3 and Supp.S9. At PSNR 32, our model achieves higher SSIM scores than other methods, even without SSIM optimization as in SSLWM. Visually, our method minimizes textural distortions, which are produced by SSLWM and DNN0B, while maintaining nearly 100% watermark detection accuracy at a significance level of 0.05. Even



(a) Noise **(b)** Blur **(c)** Cropping **(d)** Rotation **(e)** JPEG

Fig. 2: Robustness of watermarking methods to various distortions: (a) Gaussian noise, (b) Gaussian blur, (c) cropping, (d) rotation, and (e) JPEG compression. For each type of distortion, we vary its parameters and calculate the success detection rate.

at the higher PSNR of 42, our method's image quality SSIM remains superior, with results visually indistinguishable from the original images.

4.5 Robustness Analysis

We selected image perturbations commonly used for testing watermark robustness in $[1, 2, 10, 25, 36]$: Gaussian noise, Gaussian blur, rotation, cropping, and JPEG compression. For each perturbation, we measure the robustness of the model using SDR. Fig. 2 presents the comparative results, which are also summarized in Tab. 1. Our method achieves comparable performance to other approaches and is more effective than others against Gaussian noise, Gaussian blur, and JPEG compression. However, for rotation and cropping perturbation, our method exhibits a declining trend in performance as the distortion factor intensifies. We attribute this to differences in the spatial distribution of modified pixels in the watermarked images. We hypothesize that methods like SSLWM and DNN0B distribute the content distortion across the entire image, which leads to advantages when rotating or cropping the watermarked image. In contrast, our method focuses on specific areas, which reduces its robustness if key parts are cropped out or repositioned. This advantage of SSLWM and DNN0B likely comes from the rotation and cropping data-augmentation used when pretraining their DNNs on ImageNet image classification, which could possibly be adopted for training our SKNs.

Finally, we use InstructPix2Pix [3] to edit watermarked images. Compared with DNN0B and SSLWM whose detection accuracies drop to 0%, our method is more robust, maintaining a detection accuracy of 40% after image editing.

4.6 Security Analysis

In this section, we test the security of our watermarking framework against other zero-bit methods. We imagine a scenario involving two users: Alice, the owner of an image, and Bob, a would-be thief, attempting to claim ownership of Alice's image. In this scenario, Bob tries the following three methods. We assume that a host platform can store and process the keys/identifiers securely (see Supp. S8 for discussion), and Bob cannot attack the platform and steal them directly.

	DNN0B [36]	SSLWM [10]	Ours
Case 1: randomly generated	**0.08**%	12.84%	1.94%
Case 2: fake model generated	100.00%	100.00%	**4.00**%

Table 4: Comparison of the security of signature vector generation. The percentage of fake signatures that incorrectly pass the authority check.

Case 1: Bob generates a fake SKS randomly in hopes of matching the SKS in Alice's watermarked image. To determine the viability of this case, we randomly produced two sets of signatures, one for watermarking test images and the other as fake signatures to match these watermarked images. The results in Tab. 4 (1st row) indicate that our method is more secure than SSLWM. Although DNN0B almost approaches a zero false detection rate, it performs the worst in the watermark removal test of Case 3 below.

Case 2: Bob attempts to steal Alice's signature by examining the DNN's output on a watermarked image (assuming that Bob knows the watermarking framework). Since DNN0B and SSLWM use models pre-trained from other tasks, Bob can easily obtain their DNNs and thus recover the signature vector $\hat{\mathbf{m}}$ embedded by DNN0B/SSLWM by running the pre-trained DNN on the watermarked image. Bob can then use the recovered $\hat{\mathbf{m}}$ as his own signature, easily passing the authority check. However, in our framework, Alice's SKN is kept secret from Bob, and thus he cannot use it to recover the signature from Alice's watermarked image. He can only resort to training a *new* SKN, which will likely be different from Alice's SKN due to different random initial seeds. The experimental results in Tab. 4 (2nd row) also support this conclusion: watermarked images produced using DNN0B and SSLWM methods are easily matched with fake signatures extracted using the pre-trained DNN, whereas our method exhibits high security in this scenario since Alice's SKN cannot be replicated by Bob.

An extreme case, only applicable to our proposed framework, is that Bob has Alice's SKS, and he tries to train a new SKN to steal the ownership of Alice's watermarked images. After training the new SKN, he uses it to detect watermarks from those images watermarked by Alice's SKN. In this case, the SDR of Bob's SKN drops to 5% when using the combined p-value, and drops to 4.14% and 4.92% when using HT4A and HT4L.

This result suggests that the SKN plays a critical role in watermarking security because it is both a secret key for the length metric and guarantees the uniqueness of the angle metric.

Case 3: Bob tries to remove Alice's watermark from an image by adding his own watermark to the image. For each watermarking method, we embed a watermark signature into an image, acting as Alice's watermark. Then, we generate four different signatures and recursively embed them into the watermarked image, acting as Bob's attack. The final image contains 5 watermarks, one for Alice and 4 for Bob. In each iteration, we check whether Alice's watermark can still be detected. For DNN0B and SSLWM, the same pre-trained DNN is used, and different signatures are embedded. For our method, we test two versions: 1) the

No. Watermarks	1	2	3	4	5
DNN0B [36]	98.50%	0.00%	0.00%	0.00%	0.00%
SSLWM [10]	92.62%	60.50%	27.00%	19.50%	12.00%
Ours(S)	99.50%	99.00%	98.00%	99.70%	99.60%
Ours(S+N)	99.50%	98.60%	96.10%	89.70%	89.70%

Table 5: Robustness test against embedding of multiple overlapping signatures. The detection rate of the original watermark (1) after recursively embedding new watermarks (2 to 5) to the image. Ours(S) means the SKS varies in each iteration and the SKN is the same, and Ours(S+N) means both the SKS and SKN vary in each iteration.

SKN remains the same, and different SKS are used in each iteration (equivalent to Alice overwriting her own watermark with her *secret* SKN); and 2) both the SKN and SKS are changed in each iteration (equivalent to Bob overwriting Alice's watermark with new SKNs).

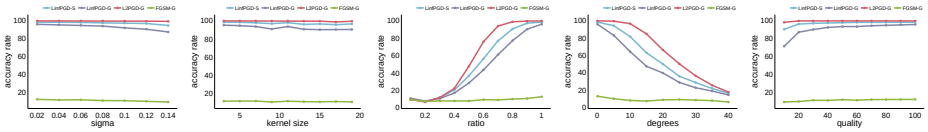
The results are presented in Tab. 5. Initially, when the first signature is introduced to the original image, it is nearly 100% detectable by all three methods. However, adding a second signature led DNN0B to eliminate the first watermark entirely, dropping its detectability to 0%. While SSLWM's method partially preserves the first signature after the second signature is added, the detectability drops significantly in subsequent iterations. In contrast, our method consistently sustains a high detection rate for a watermark, even after 4 additional watermarks are embedded. Moreover, our method demonstrates robust performance even when altering both the SKN and SKS.

4.7 Ablation Studies

We conduct ablation studies on a few key components of our framework. See Supp. S10-S13 for additional ablation studies on SKS generation, generalization to unseen datasets [3, 7], runtime, and target length.

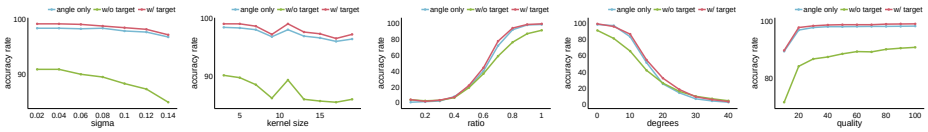
Effect of adversarial attack. Here we consider different versions of AA for embedding the watermark, with results in Fig. 3. The common PGD attack uses the sign of the gradient and the L_∞ norm (denoted as LinfPGD-S). To enhance robustness for watermarking, we replaced the sign of the gradient with its actual value (LinfPGD-G). Furthermore, employing the L_2 norm for the perturbation constraint further improves performance (L2PGD-G). In contrast to these multiple iteration adversarial attacks like PGD, the single-step attack FGSM, modified to use the gradient value (FGSM-G), exhibits the poorest performance.

Effect of adversarial loss. Our adversarial loss $\mathcal{L}_{\text{adv}}$ uses target values of $t_l = 63$ and $t_a = 1$ for the length and cosine terms. We can also define an adversarial loss without a length target value, which aims to increase the length of the output vector. The robustness comparing with and without the target length are shown in Fig. 4. Using the target length has better performance. We hypothesize that the competition between length and angle terms for modifying pixel values within the limited ϵ-bound necessitates setting target values so that once the target length is reached, the remaining capacity is used for adjusting the



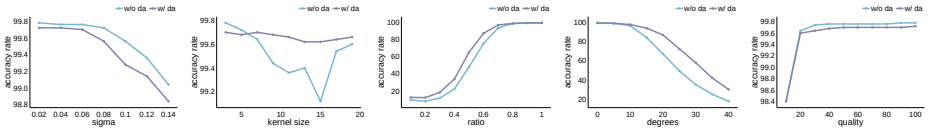
(a) Noise **(b)** Blur **(c)** Cropping **(d)** Rotation **(e)** JPEG

Fig. 3: Effect of using different AA on watermark robustness to image distortions. "G" and "S" indicate using the direct gradient value or its sign, respectively.



(a) Noise **(b)** Blur **(c)** Cropping **(d)** Rotation **(e)** JPEG

Fig. 4: Effect of watermarking with and without target length values in adversarial loss $\mathcal{L}_{\text{adv}}$, and only using the angle metric.



(a) Noise **(b)** Blur **(c)** Cropping **(d)** Rotation **(e)** JPEG

Fig. 5: Effect of data augmentation module on watermarking.

output direction. Furthermore, experiments using only the angle metric in $\mathcal{L}_{\text{adv}}$ (corresponding HT4A) show relying solely on angle is marginally less effective.

Effect of data augmentation. To enhance the robustness of the watermark, we also introduce data augmentation operations on the image during the watermarking process, as used in [10, 36]. Specifically, in the iterative process of watermarking, we randomly perform data augmentation (rotation and cropping) on the image and then recalculate $\mathcal{L}_{\text{adv}}$. Fig. 5 shows that data augmentation can significantly improve the detection rate of watermarks.

5 Conclusion

In this paper, we propose a novel watermarking framework that leverages secret key networks with specific statistical properties. We employ adversarial attacks to embed watermarks into images and deploy hypothesis tests to detect these watermarks with statistical guarantees. To ensure a higher level of watermark security, in addition to using a secret key signature (SKS), we also introduce a secret key network (SKN), which effectively makes the DNN as a watermarking key. We hypothesize three potential scenarios that could threaten watermark security and confirm that our methodology exceeds our baseline security measures.

Acknowledgments

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 11211624).

References

1. Ahmadi, M., Norouzi, A., Karimi, N., Samavi, S., Emami, A.: ReDMark: Framework for residual diffusion watermarking based on deep networks. Expert Systems with Applications **146**, 113157 (May 2020)
2. Baluja, S.: Hiding Images in Plain Sight: Deep Steganography. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
4. Chang, C.C., Tsai, P., Lin, C.C.: SVD-based digital image watermarking scheme. Pattern Recognition Letters **26**(10), 1577–1586 (2005)
5. Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P.: Digital image steganography: Survey and analysis of current methods. Signal processing **90**(3), 727–752 (2010)
6. Comesana, P., Merhav, N., Barni, M.: Asymptotically optimum universal watermark embedding and detection in the high-snr regime. IEEE Transactions on Information Theory p. 2804–2815 (Jun 2010)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
8. Dowson, D., Landau, B.: The fréchet distance between multivariate normal distributions. Journal of multivariate analysis **12**(3), 450–455 (1982)
9. Fei, J., Xia, Z., Tondi, B., Barni, M.: Supervised gan watermarking for intellectual property protection. In: 2022 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (2022)
10. Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., Douze, M.: Watermarking images in self-supervised latent spaces. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3054–3058 (2022)
11. Furon, T., Bas, P.: Broken arrows. EURASIP Journal on Information Security p. 597040 (Jan 2008)
12. Givens, C.R., Shortt, R.M.: A class of wasserstein metrics for probability distributions. Michigan Mathematical Journal **31**(2), 231–240 (1984)
13. Guan, Z., Jing, J., Deng, X., Xu, M., Jiang, L., Zhang, Z., Li, Y.: DeepMIH: Deep Invertible Network for Multiple Image Hiding. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 372–390 (Jan 2023)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Huang, H., Wang, Y., Chen, Z., Zhang, Y., Li, Y., Tang, Z., Chu, W., Chen, J., Lin, W., Ma, K.K.: CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes. Proceedings of the AAAI Conference on Artificial Intelligence **36**(1), 989–997 (Jun 2022)
16. Jia, X., Wei, X., Cao, X., Han, X.: Adv-watermark: A Novel Watermark Perturbation for Adversarial Examples. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1579–1587 (2020)

17. Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z.: HiNet: deep image hiding by invertible network. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4733–4742 (2021)
18. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6007–6017 (2023)
19. Kundur, D., Hatzinakos, D.: Digital watermarking using multiresolution wavelet decomposition. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181). vol. 5, pp. 2969–2972 vol.5 (May 1998)
20. Lanczos, C.: A precision approximation of the gamma function. Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis **1**(1), 86–96 (1964)
21. Li, Y., Cheng, M., Hsieh, C.J., Lee, T.C.: A review of adversarial attack and defense for classification methods. The American Statistician **76**(4), 329–345 (2022)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755 (2014)
23. Liu, Y., Guo, M., Zhang, J., Zhu, Y., Xie, X.: A Novel Two-stage Separable Deep Learning Framework for Practical Blind Watermarking. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 1509–1517 (2019)
24. Liu, Y., Ke, Z., Liu, F., Zhao, N., Lau, R.W.: Diff-plugin: Revitalizing details for diffusion-based low-level tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4197–4208 (2024)
25. Luo, X., Zhan, R., Chang, H., Yang, F., Milanfar, P.: Distortion Agnostic Deep Watermarking. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13545–13554. Seattle, WA, USA (Jun 2020)
26. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: (PGD) Towards Deep Learning Models Resistant to Adversarial Attacks (Sep 2019)
27. Olkin, I., Pukelsheim, F.: The distance between two random vectors with given dispersion matrices. Linear Algebra and its Applications **48**, 257–263 (1982)
28. O'Ruanaidh, J., Pun, T.: Rotation, scale and translation invariant digital image watermarking. In: Proceedings of International Conference on Image Processing. vol. 1, pp. 536–539 vol.1 (Oct 1997)
29. Potdar, V.M., Han, S., Chang, E.: A survey of digital image watermarking techniques. In: INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005. pp. 709–716 (2005)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
31. Sabbag, E., Merhav, N.: Optimal watermark embedding and detection strategies under limited detection resources. In: 2006 IEEE International Symposium on Information Theory (Jul 2006)
32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
33. Theiler, J.: Combining statistical tests by multiplying p-values. Astrophysics and Radiation Measurements Group, NIS-2 (2004)

34. Van Le, T., Phung, H., Nguyen, T.H., Dao, Q., Tran, N., Tran, A.: Anti-DreamBooth: Protecting users from personalized text-to-image synthesis (Mar 2023)
35. Vukotic, V., Chappelier, V., Furon, T.: Are deep neural networks good for blind image watermarking. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS) (Dec 2018)
36. Vukotić, V., Chappelier, V., Furon, T.: Are classification deep neural networks good for blind image watermarking? Entropy **22**(2), 198 (2020)
37. Wan, W., Wang, J., Zhang, Y., Li, J., Yu, H., Sun, J.: A comprehensive survey on robust image watermarking. Neurocomputing **488**, 226–247 (Jun 2022)
38. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust. arXiv preprint arXiv:2305.20030 (2023)
39. Wengrowski, E., Dana, K.: Light Field Messaging With Deep Photographic Steganography. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1515–1524 (Jun 2019)
40. Wilson, E.B., Hilferty, M.M.: The distribution of chi-square. Proceedings of the National Academy of Sciences **17**(12), 684–688 (1931)
41. Xu, H., Ma, Y., Liu, H.C., Deb, D., Liu, H., Tang, J.L., Jain, A.K.: Adversarial attacks and defenses in images, graphs and text: A review. International Journal of Automation and Computing **17**, 151–178 (2020)
42. Xu, Y., Mou, C., Hu, Y., Xie, J., Zhang, J.: Robust Invertible Image Steganography. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7865–7874 (Jun 2022)
43. Yeung, M., Mintzer, F.: An invisible watermarking technique for image verification. In: Proceedings of International Conference on Image Processing. vol. 2, pp. 680–683 vol.2 (Oct 1997)
44. Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.M., Lin, M.: A Recipe for Watermarking Diffusion Models (Mar 2023)
45. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: HiDDeN: Hiding Data With Deep Networks. In: Computer Vision – ECCV 2018,Lecture Notes in Computer Science, pp. 682–697. Springer (Jan 2018)