SpecGuard: Spectral Projection-based Advanced Invisible Watermarking

Anonymous ICCV submission

Paper ID 13406

Abstract

Watermarking embeds imperceptible patterns into images for authenticity verification. However, existing methods often lack robustness against various transformations primarily including distortions, image regeneration, and adversarial perturbation, creating real-world challenges. In this work, we introduce SpecGuard, a novel watermarking approach for robust and invisible image watermarking. Unlike prior approaches, we embed the message inside hidden convolution layers by converting from the spatial domain to the frequency domain using spectral projection of a higher frequency band that is decomposed by wavelet projection. Spectral projection employs Fast Fourier Transform approximation to transform spatial data into the frequency domain efficiently. In the encoding phase, a strength factor enhances resilience against diverse attacks, including adversarial, geometric, and regeneration-based distortions, ensuring the preservation of copyrighted information. Meanwhile, the decoder leverages Parseval's theorem to effectively learn and extract the watermark pattern, enabling accurate retrieval under challenging transformations. We evaluate the proposed SpecGuard based on the embedded watermark's invisibility, capacity, and robustness. Comprehensive experiments demonstrate the proposed SpecGuard outperforms the state-of-the-art models. To ensure reproducibility, we release the full code on ***GitHub****.*

1. Introduction

With the rapid advancement of digital media and artificial intelligence, concerns regarding image authenticity, copyright protection, and content integrity have become more challenging than ever [7, 15, 31]. Moreover, the widespread availability of the latest image manipulation tools [3, 13] enables malicious tamperers to easily forge and redistribute digital content without authorization, posing a significant threat to ownership verification [28]. This growing risk emphasizes the need for reliable techniques for secure authentication and detection of unauthorized modifications of the original information.

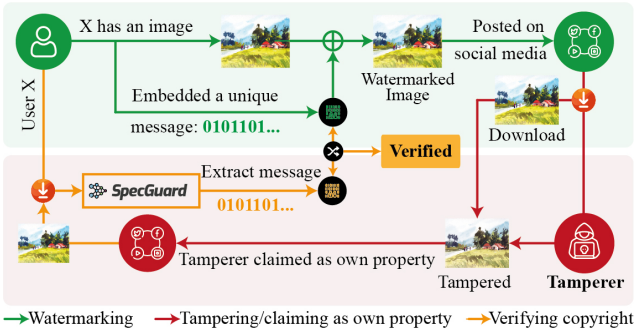

Figure 1. Image authentication using our proposed SpecGuard.

Recently, invisible watermarking has gained significant attention as a prominent defense mechanism for media authentication by embedding invisible messages into images to verify authenticity [34, 51]. In fact, invisible watermarks are preferred for preserving image quality and resisting tampering. These watermarks are unique to the creator and enable tamper verification by comparing the retrieved watermark to the original, as the high-level process is presented in Fig. 1. Traditional watermarking methods often rely on transformation techniques [17, 32]. Deep learning approaches like StegaStamp [44], Stable Signature [34], and HiDDeN [58] provide end-to-end solutions for message embedding. However, these methods often struggle with fragility in handling common image processing operations such as resizing, cropping, compression, and noise addition, which can distort or erase the embedded watermark. Additionally, the performance of watermark embedding and extraction often remains vulnerable to attacks with noise injection, blurring, contrasting, and rotation [6].

To address the aforementioned challenges, we introduce a novel robust, and invisible image watermarking method named SpecGuard. SpecGuard is designed to overcome the fundamental trade-offs [6] between imperceptibility, and robustness. Our proposed SpecGuard strategically embeds watermark information in the spectral domain, leveraging wavelet-based decomposition to distribute the watermark across high-frequency components. Unlike traditional frequency domain watermarking techniques [50, 53] that are

easily disrupted by common image manipulations, SpecGuard maintains imperceptibility while significantly improving robustness against a wide range of transformations.

Overall, our proposed SpecGuard addresses the current limitations of the previous watermarking methods by providing a robust, imperceptible watermarking technique that maintains integrity under diverse manipulations, significantly enhancing digital content security and authenticity verification. Our key contributions are as follows:

- We introduce a novel watermarking approach that embeds message bits in high-frequency spectral components via wavelet and spectral projection inside hidden convolutional layers, ensuring robustness against various transformations and adversarial attacks.
- We adapt Parseval's theorem [19] as a learnable threshold to optimize SpecGuard and spectral masking for robust watermark bit recovery under diverse transformations including distortions, regeneration, and adversarial attacks, proven through the experimental results.
- Our extensive evaluations demonstrate SpecGuard's superior bit embedding capacity and producing better invisible watermarked images, surpassing the performance of state-of-the-art (SOTA) methods.

2. Related Works

Watermarking an image has been a widely researched topic for securing the ownership and verifying authenticity of digital content [40]. Traditional watermarking techniques typically embed invisible [45] or visible [5] watermarks into images, which can later be extracted or detected to verify the content's originality. These methods can be broadly classified into spatial-domain [42, 47] and frequency-domain [11] watermarking, while some are based on combined methods [41, 54]. However, researchers recently proposed many advanced models [22, 26, 30, 43] for effective watermark removal. To face this growing challenge, researchers introduced different methods [2, 21, 27, 55, 58] as alternatives to deep learning-based encoders or decoders to produce more robust image watermarking. Furthermore, iterative models have demonstrated competitive performance [20, 33], particularly in robustness against a wide range of transformations. In addition, with the rise of generative methods, researchers used the watermark-labeled data for training to learn how to produce watermarks [9, 23]. Also, models that combine generative methods with watermarking techniques show promise in effective image watermarking [24, 29, 36]. However, such approaches face limitations such as increased computational complexity and longer processing times. These approaches are also more vulnerable to adversarial attacks that can target and distort the embedded watermark without altering the content visibly.

3. Proposed Method: SpecGuard

We introduce SpecGuard, as illustrated in Fig. 2, which involves two fundamental modules: an "Encoder" for embedding the watermark and a "Decoder" for accurately extracting the watermark detailed in the following sections.

3.1. Encoder

By targeting high-frequency components, the encoder integrates a binary message M into the cover image I. Using wavelet projection (WP) [32] and a Fast Fourier Transform (FFT)-based spectral projection (SP) [17] approximation, the message M is inserted into specific frequency bands, minimizing perceptual impact.

Wavelet Projection. We use a wavelet projection to capture frequency and spatial localization features that describe an image across different scales, as shown in Eq. (1):

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(x)\, \psi\left(\frac{x-b}{a}\right) dx, \tag{1}$$

where $a \in \mathbb{R} \setminus \{0\}$, and $b \in \mathbb{R}$ denote the scaling and translation parameters, respectively. Here, $\psi_{a,b}(x)$ represents a rescaled and translated form of the mother wavelet ψ, defined as follows:

$$\psi_{a,b}(x) = \psi\left(\frac{x-b}{a}\right) \cdot \frac{1}{\sqrt{|a|}}, \tag{2}$$

where $\frac{1}{\sqrt{|a|}}$ functions as a normalization factor, guaranteeing that the energy of the wavelet is invariant to the scaling parameter a. Minimal values of a compress the wavelet, enabling the inspection of high-frequency components, whereas greater values of a elongate the wavelet, promoting low-frequency analysis. Since each mother wavelet ψ is built with zero mean and finite energy [10], it guarantees to maintain stability as follows:

$$\int_{-\infty}^{\infty} \psi(x)\, dx = 0, \quad \int_{-\infty}^{\infty} |\psi(x)|^2\, dx < \infty, \tag{3}$$

where the wavelet projection from Eq. (1) decomposes the input into orthogonal wavelet sets using discrete scales and translations. For $2D$ inputs, the scaled and translated basis elements [1] are defined for each coordinate pair (u, v):

$$\begin{aligned} \mathbf{S}_{LL} &= \phi(u,v) = \phi(u)\phi(v), \quad & \mathbf{S}_{LH} &= \psi_H(u,v) = \psi(u)\phi(v), \\ \mathbf{S}_{HL} &= \psi_V(u,v) = \phi(u)\psi(v), \quad & \mathbf{S}_{HH} &= \psi_D(u,v) = \psi(u)\psi(v), \end{aligned} \tag{4}$$

where, H, V, and D represent the horizontal, vertical, and diagonal decomposition direction, respectively. To depict the image at different resolutions, we define scaling and wavelet functions at scale j as shown below:

$$\begin{aligned} \phi_{j,m,n}(u,v) &= 2^{j/2}\phi\left(u - \frac{m}{2^j}, v - \frac{n}{2^j}\right), \\ \psi^d_{j,m,n}(u,v) &= 2^{j/2}\psi^d\left(u - \frac{m}{2^j}, v - \frac{n}{2^j}\right), \end{aligned} \tag{5}$$

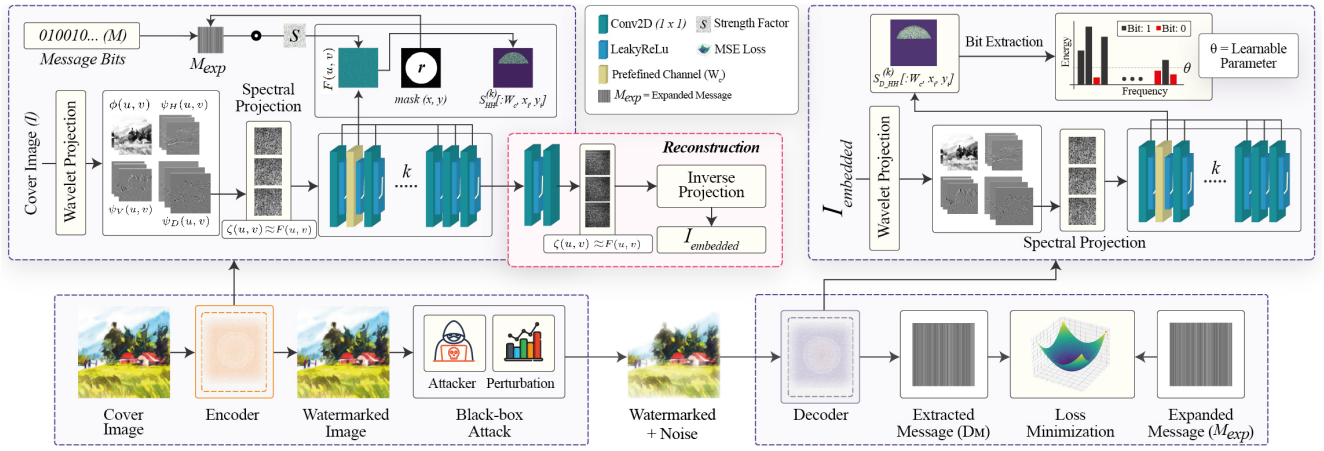

Figure 2. Architecture of the proposed SpecGuard watermarking method involves encoding a binary message M into the high-frequency band of the cover image I using wavelet and spectral projection and learning to decode the embedded message.

where $d \in \{H, V, D\}$ is the wavelet function direction that serves as discrete basis elements for multi-resolution analysis, capturing details across frequency bands and spatial locations. In Eq. (6), $T_{m,n}$ denotes the intensity or pixel value of the cover image I at spatial coordinates (m, n). The discrete scaling function $W_\phi(j, u, v)$ (approximation at scale j) and the detail coefficients $W_\psi^d(j, u, v)$ for each direction are computed accordingly as follows:

$$W_\phi(j,u,v) = \frac{1}{l}\sum_{m=0}^{l-1}\sum_{n=0}^{l-1} T_{m,n}\,\phi\left(m - u\cdot 2^{-j}, n - v\cdot 2^{-j}\right),$$
$$W_\psi^d(j,u,v) = \frac{1}{l}\sum_{m=0}^{l-1}\sum_{n=0}^{l-1} T_{m,n}\,\psi^d\left(m - u\cdot 2^{-j}, n - v\cdot 2^{-j}\right), \tag{6}$$

with l as the discrete region dimension, these coefficients capture multi-scale, multi-orientation image information, forming the basis of spectral features as follows:

$$\beta_j = \bigcup_{d\in\{H,V,D\}} \left(W_\phi(j,u,v) \cup W_\psi^d(j,u,v)\right). \tag{7}$$

This feature set β_j captures key frequency and spatial details across resolutions, forming the foundation for the watermark embedding process of our SpecGuard.

Selective Frequency Band Decomposition. To refine the embedding process, we segment the data into distinct frequency bands. The decomposition level κ is determined by the image complexity, calculated as follows:

$$\kappa = \lfloor \sqrt{\log(1+N)} \rfloor, \tag{8}$$

where N denotes the total pixel count in the cover image I. And, each component β_j falls within a unique frequency band, yielding a total of $1 + 3\kappa$ distinct frequency bands as follows:

$$\beta_j = \phi_j(u,v) \cup \bigcup_{d\in\{H,V,D\}} \psi_j^d(u,v). \tag{9}$$

The components β_j, consisting of scaling functions $\phi_j(u, v)$ and wavelet functions $\psi_j^d(u, v)$, capture specific spatial frequency bands, enabling targeted high-frequency embedding. We translate the WP into disjoint intervals representing a unique frequency range to approximate the segmentation in the frequency domain:

$$\beta_j = \left\{ W_\psi^d(u,v) \mid u,v \in \left(\frac{j\cdot L}{\kappa}, \frac{(j+1)\cdot L}{\kappa}\right)\right\}, \tag{10}$$

where, L is the dimension of S_{HH}, and $W_\psi^d(u, v)$ represents wavelet values within segmented intervals. This frequency band partitioning mimics the frequency selectivity of wavelet sub-bands, enabling effective targeting of high-frequency regions for optimal embedding.

Approximation of Spectral Projection. We first apply spectral projection on the S_{HH} sub-band, transforming it into the spectral domain. Given a matrix $T(x, y)$ representing pixel intensities in S_{HH}, the spectral projection computes the spectral components $\zeta(u, v)$ as follows:

$$\zeta(u,v) = \frac{1}{L^2}\sum_x\sum_y T(x,y)\cdot \exp\left(-i\frac{2\pi}{L}\left(x\cdot u + y\cdot v\right)\right), \tag{11}$$

where L denotes the dimension of S_{HH}, $T(x, y)$ provides the intensity at each coordinate (x, y) which is equivalent to $W_\psi^d(u, v)$ in Eq. (6), i is the imaginary unit, and (u, v) are the spectral coordinates.

To approximate the spectral components using the FFT, we create a symmetrically extended version $\tilde{T}(x, y)$ of the original $N \times N$ matrix $T(x, y)$. This extension is achieved by mirroring $T(x, y)$ along its boundaries, doubling its size to $2N \times 2N$. Specifically, the original matrix occupies the top-left quadrant, with the remaining quadrants filled by reflecting $T(x, y)$ horizontally, vertically, and diagonally, respectively. This symmetric structure ensures that the FFT

yields only real values, allowing the spectral coefficients to be extracted directly from the real part of the FFT operation. Then, we apply the 2D FFT to $\tilde{T}(x, y)$ as follows:

$$F(u,v) = \frac{1}{(2N)^2} \sum_x \sum_y \tilde{T}(x,y) \cdot \exp\left(-i\frac{2\pi}{2N}(x \cdot u + y \cdot v)\right). \tag{12}$$

The SP coefficients are then approximated by taking the real part (Re) of F in the original $N \times N$ region as follows:

$$\zeta(u,v) \approx \mathrm{Re}(F(u,v)), \quad 0 \le u, v < N. \tag{13}$$

Applying Eq. (13) to the sub-bands extracted from wavelet projection in Eq. (6), we achieve a computationally efficient spectral projection by leveraging the FFT approximation on a symmetrically extended matrix, maintaining effective embedding properties within the spectral domain.

SpecGuard Embedding Process. The embedding process integrates the binary message M into the high-frequency band S_{HH} of the cover image I, enhancing robustness and imperceptibility through wavelet and spectral projection. Using the Eq. (6) and Eq. (13), the cover image I is decomposed into sub-bands S_{LL}, S_{LH}, S_{HL}, and S_{HH} within spectral domain, with S_{HH} providing high-frequency details for embedding. A variable number k of convolutional layers with a $K \times K$ kernel, followed by LeakyReLU activation, are recursively applied to S_{HH} to refine spectral features as follows:

$$S_{HH}^{(n+1)} = \mathrm{LeakyReLU}(\mathrm{Conv}_{2D}(S_{HH}^{(n)}, K)), \quad n = 1, \ldots, k. \tag{14}$$

The final output $S_{HH}^{(n+1)}$ from Eq. (14) represents the modified high-frequency band, primed for embedding.

The message M, represented as a binary vector of length l ($M \in \{0,1\}^l$), with batch size b and message length l, is reshaped and expanded across channels c to align with $S_{HH}^{(n+1)}$. This transformation ensures M_{expanded} conforms to the dimension $[b, c, l]$, where each message is structured accordingly.

To localize the embedding, we create a radial mask centered at $(c_x, c_y) = \left(\frac{h}{2}, \frac{w}{2}\right)$, where h and w represent the height and width of S_{HH}. The Euclidean distance $D(x_i, y_i)$ from the center (c_x, c_y) is computed for each coefficient (x_i, y_i). A binary mask is then generated within the predefined radius r based on the distance $D(x_i, y_i)$, such that if $D(x_i, y_i) \le r$, the mask value is set to 1, allowing embedding in the corresponding region. Otherwise, the mask value is 0, restricting embedding to areas within a specified radius r, ensuring focus on high-frequency regions.

For each coordinate (x_i, y_i) where mask (x_i, y_i) is 1 and $W_c \in c$, the embedding operation is performed as follows:

$$S_{HH}^{(n+1)}[:, W_c, x_i, y_i] \mathrel{+}= M_{\text{expanded}}[:, W_c, i] \cdot s, \tag{15}$$

where s is the strength factor controlling embedding intensity and invisibility. After embedding, the modified coefficients $S_{HH}^{(n+1)}$ undergo a final convolution and LeakyReLU using Eq. (14), by setting the value of $k = 1$ to harmonize the embedded message. Following this approach, SpecGuard embeds the message into the spectral domain in a transformed form, differing from its original input representation. By blending the message seamlessly into the spectral space based on the r, s, and W_c, it becomes inherently concealed within the domain, rendering its presence imperceptible. Without knowledge of r, s, and W_c, it becomes exceedingly challenging to localize the embedded message, further enhancing the security of the system. This transformation ensures the embedding process remains opaque to any adversarial attacker, effectively making SpecGuard a black-box system.

Reconstruction. SpecGuard encoder reconstructs the watermarked image I_{embedded} by inverse transformation restoring S_{HH} back into the spatial domain. The reconstruction process integrates the inverse wavelet projection (IWP) [32] and inverse spectral projection (ISP) [17], ensuring the embedded modifications are correctly translated into the spatial domain. To reconstruct the spatial domain image, S_{HH} is combined with the other sub-bands S_{LL}, S_{LH}, and S_{HL}. For the SP embedded in S_{HH}, the ISP is applied to reconstruct S_{HH} to spatial domain as follows:

$$S_{HH}(x,y) = \sum_{u=0}^{L-1} \sum_{v=0}^{L-1} \zeta(u,v) \cdot \exp\left(i\frac{2\pi}{L}(x \cdot u + y \cdot v)\right), \tag{16}$$

where $\zeta(u, v)$ represents spectral coefficients from the embedding process, L denotes the dimension of S_{HH}, and (x, y) are spatial coordinates. SpecGuard then reconstructs the watermarked image I_{embedded} using the IWP as follows:

$$I_{\text{embedded}}(x,y) = \mathrm{IWP}(S_{LL}, S_{LH}, S_{HL}, S_{HH}). \tag{17}$$

This process seamlessly embeds the watermark message M in the spectral domain, preserving the cover image I's integrity. The inverse transformations that are expressed in Eq. (16) and Eq. (17) fully restore visual quality, maintaining all frequency components.

3.2. Decoder

As shown in Algorithm 1, SpecGuard decoding process starts by applying wavelet projection (Eq. (1)) to the watermarked image I_{embedded}, separating it into low and high-frequency bands, where the high-frequency band $S_{D_{HH}}^{high}$ contains the embedded message similar to the process in the encoding phase, particularly in Eq. (6). An approximation of the spectral projection using FFT as shown in Eq. (13) is then applied to $S_{D_{HH}}^{\text{high}}$ returning the transformed data $S_{D_{HH}}^{\text{sp}}$. Then, $S_{D_{HH}}$ is further refined through convolutional layers that captures the local features for message extraction.

To extract the message, a radial mask is created to isolate high-frequency areas within $S_{D_{HH}}$, targeting the embedded regions based on their distance from the center. The

Algorithm 1 SpecGuard decoder with wavelet, spectral projection with FFT approximation, and learnable threshold.

Input: Watermarked image I_{embedded}, learnable θ, message length l, radius r, watermark channel W_c
Output: Decoded binary message D_{M}
Procedure: Apply Wavelet Projection on I_{embedded} to obtain $S_{D_{LL}}$ (low-frequency) and $S^{\text{high}}_{D_{HH}}$ (high-frequency)
Procedure: Spectral approximation with FFT ($S^{\text{high}}_{D_{HH}}$):
  Separate even and odd indices: $v = [x_{\text{even}}, \text{reverse}(x_{\text{odd}})]$
  Compute FFT on v: $V_{\text{complex}} = \text{FFT}(v)$
  $V_{\text{real}} = V_{\text{complex}} \cdot \left[\cos\left(\frac{-\pi k}{2N}\right), \sin\left(\frac{-\pi k}{2N}\right)\right]$ // Calculate Real
  $V_{\text{real}}[0] \leftarrow \frac{V_{\text{real}}[0]}{\sqrt{N}\cdot 2}, V_{\text{real}}[1:] \leftarrow \frac{V_{\text{real}}[1:]}{\sqrt{\frac{N}{2}}\cdot 2}$ // Energy preservation
  Transpose result and repeat to obtain $S^{\text{sp}}_{D_{HH}}$
Return $S^{\text{sp}}_{D_{HH}}$
Procedure: Pass $S^{\text{sp}}_{D_{HH}}$ through sequential layers as:

$$S^{(n+1)}_{D_{HH}} = \text{LeakyReLU}\left(\text{Conv}_{2D}\left(S^{sp(n)}_{HH}, K\right)\right), n = 1, ..., k,$$

Return $S^{(n+1)}_{D_{HH}}$
Procedure: Extraction ($S^{(n+1)}_{D_{HH}}, l$):
  Set $(c_x, c_y) = \left(\frac{H}{2}, \frac{W}{2}\right)$
  Generate mask for high-frequency region within radius r
  for each coordinate (i, j) **do:**
    $D(x_i, y_i) = \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2}$ // Euclidian Distance
    if $D(x_i, y_i) \leq r$ **then**
      Set mask$[i, j] = 1$
    end if
  end for
  Extract mask: $S_{D_{HH}}[:, W_c, \text{mask}[i, j]]$
  Decode message using learnable θ:

$$D_{\text{M}}[i] = \begin{cases} 1 & \text{if Extracted}[i] > \theta \\ 0 & \text{otherwise} \end{cases}$$

  Update θ dynamically: $\theta \leftarrow \theta - \eta \cdot \frac{\partial L_{\text{dec}}}{\partial \theta}$ // Optimizes robustness
Return D_{M}

masked values are compared against a learnable threshold θ to decode each bit of the hidden message D_{M}. Here, θ serves as a threshold that adapts to the spectral patterns across the entire image, learning the distinct characteristics of the embedded watermark. From Parseval's theorem [19] ensures overall spectral and spatial energies remain equivalent, though local spectral energy distributions are altered by the watermark strength factor s .

The watermark's strength factor s ensures that the high-energy areas where the message M is embedded as "1" remain robust, experiencing a minimum distortion in such conditions. Moreover, this threshold can be optimized for better bit recovery accuracy during training. As θ learns, it recognizes that areas encoded as "1" carry higher energy and impact due to the strength factor s of Eq. (15), while areas marked as "0", softened by the LeakyReLU's minimal negative slope, have a lower intensity. Such a dynamic approach enables θ to identify and protect the embedded message M even when external disturbances occur, preserving the watermark's structure within the watermarked image I_{embedded}. And, θ effectively learns to distinguish high-energy watermark regions. Therefore, the embedded message is more recoverable under diverse attacks, and SpecGuard's decoder ensures valid watermark bit extraction. Theoretical explanation of Parseval theorem's [19] impact on message extraction is in the Supplementary.

3.3. Loss Calculation for SpecGuard

To achieve the training objective of robust and invisible watermark embedding, a composite loss function is defined with two terms: encoder loss L_{enc} as expressed in Eq. (18) and decoder loss L_{dec} as expressed in Eq. (19).

$$\min_{\theta} \mathbb{E}_{(I,M)\sim D}\, L_{\text{enc}}(I, I_{\text{embedded}}) = \|E_\theta(I, M) - I\|^2, \quad (18)$$

$$\min_{\theta} \mathbb{E}_{(I,M)\sim D}\, L_{\text{dec}}(M, D_{\text{M}}) = \|D_\theta(I_{\text{embedded}}) - M\|^2, \quad (19)$$

where $E_\theta(I, M)$ denotes the encoder output, embedding the message M into the cover image I to produce I_{embedded}. By minimizing L_{enc}, the encoder learns to embed the watermark invisibly, preserving the fidelity of the cover image. $D_\theta(I_{\text{embedded}})$ denotes the decoder's output from the watermarked image I_{embedded}. Minimizing L_{dec} enables the decoder to reliably retrieve the embedded message under varying conditions, such as noise and transformation.

The total loss L as shown in Eq. (20) used for optimizing the model combines these terms, balancing invisibility and robustness through weighted coefficients as follows:

$$\min_{\theta} L = \lambda_{\text{enc}} L_{\text{enc}} + \lambda_{\text{dec}} L_{\text{dec}}, \quad (20)$$

where λ_{enc} and λ_{dec} control the relative importance of visual fidelity and message recoverability.

4. Experimental Results

4.1. Dataset

SpecGuard is trained on the MS-COCO dataset [25], which contains 25K images. To evaluate the robustness of the watermarking methods including our SpecGuard against different types of attacks, such as distortions, regenerations, and adversarial attacks, we used three datasets: DiffusionDB [49], MS-COCO [25], and DALL·E3 [1]. Each of these datasets has a unique distribution of prompt words. We also ensured that no unethical or violent terms were included in the prompts. We randomly picked 200 images from MS-COCO [25] and applied watermark using SpecGuard for further verifying the robustness after uploading on various social media platforms and applying AI-based Photoshop Neural Filters (PNFs) [2]. The PNFs include depth

[1] https://huggingface.co/datasets/OpenDatasets/dalle-3-dataset
[2] https://www.adobe.com/products/photoshop/neural-filter.html

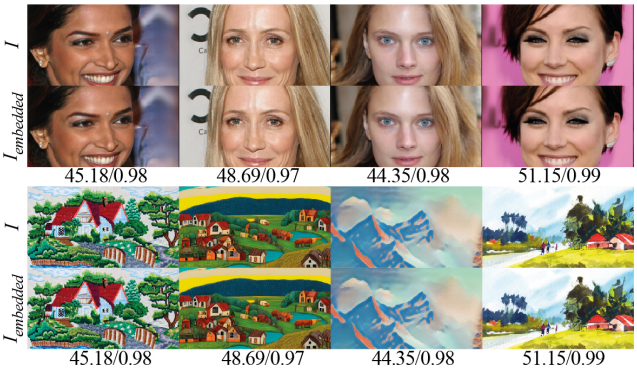

Figure 3. Some best results for cover vs watermarked images with PSNR/SSIM (↑) scores showing minimal visual degradation when watermarked using proposed SpecGuard.

blur, artistic style transfer, super zoom, JPEG artifact reduction, and colorization. For the super zoom filter, we set the 'Sharpen' and 'Noise Reduction' parameters to 15. For all other filters, we used the default settings.

4.2. Implementation

We used CUDA v11.3 and PyTorch with a batch size of 32 and the Adam optimizer on a multiple NVIDIA RTX 2080-equipped server. Mean Squared Error (MSE) and Bit Recovery Accuracy (BRA) are used for loss and accuracy calculation. We used Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Fréchet Inception Distance (FID), and MSE to evaluate perceptual quality. Our model is trained for 300 epochs, with the decoder learning rate set to 1×10^{-3}, reduced by half every 100 steps, and the encoder learning rate is set to 1×10^{-2} without scheduling. We set our watermark radius (r), strength factor (s), initial learning parameter (θ), and the number of convolutional layers (k) to 100, 20, 0.001, and 32, respectively. This setup is applied with a message bit length (BL) of 48, 64, 128, and 256. Initially, decoder loss weight (λ_{dec}) and encoder loss weight (λ_{enc}) are set to 1.0 and 0.7, respectively. For assessing the robustness of watermarking methods against diverse attacks, we inherited the experimental setups from Waves [6] and used effective metrics such as "Quality at 95% Performance (Q@0.95P)", "Quality at 70% Performance (Q@0.7P)", "Avg P" and "Avg Q." Here, Q@0.95P and Q@0.7P indicate the level of image quality degradation required for watermark detection accuracy to reach 95% and 70%, respectively. The average performance (Avg P) metric represents the mean detection accuracy across various attack strengths, while the average quality degradation (Avg Q) measures the overall impact of attacks on image quality [6, 34].

4.3. Watermark Invisibility

To evaluate the invisibility of the embedded watermark, we conducted perceptual and quantitative assessments using SpecGuard. As shown in Fig. 3, there is no noticeable perceptual degradation between the cover and watermarked images, confirming that the watermark remains imperceptible to the human eye. For a more comprehensive evaluation, we created three subsets of different image sizes ranging between 256 to 1024 with images from the MS-COCO [25] and CelebA-HQ [18] datasets and applied the SpecGuard watermarking method to compare the average PSNR values between the cover and watermarked images, as in Tab. 1.

For quantitative evaluation, we further compare the performance of SpecGuard with the SOTA pre-processing and post-processing watermarking methods. As presented in Tab. 2, SpecGuard achieves the highest PSNR of 42.89 when the bit length was 128. Additionally, it attains the highest SSIM of 0.99 at a BL of 128 and 256 among all compared methods, indicating minimal visual distortion. Additionally, SpecGuard achieved the lowest FID of 17.0 and the highest BRA of 0.99, ensuring strong

Metrics	256 × 256		512 × 512		1024 × 1024	
	CelebA-HQ	MS-COCO	CelebA-HQ	MS-COCO	CelebA-HQ	MS-COCO
PSNR↑	40.361	40.320	44.651	44.680	48.170	48.081
SSIM↑	0.9889	0.9888	0.9927	0.9927	0.9937	0.9936
FID↓	16.451	16.690	16.972	17.020	17.446	16.955
MSE↓	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001

Table 1. Perceptual quality evaluation for watermarked images using SpecGuard across resolutions and datasets.

	Methods	Venue	BL	PSNR↑	SSIM↑	FID↓	BRA↑
Pre-processing methods	Tree-Ring [50]	NeurIPS'23	64	32.33	0.91	17.7	0.98
			128	32.10	0.90	17.8	0.96
			256	31.85	0.89	17.9	0.94
	Stable Signature [34]	ICCV'23	64	30.00	0.89	19.6	0.98
			128	29.80	0.88	19.7	0.96
			256	29.50	0.87	19.8	0.96
	Yang et al. [53]	CVPR'24	64	31.45	0.90	18.2	0.98
			128	31.20	0.89	18.3	0.93
			256	30.95	0.88	18.4	0.89
	SleeperMark [48]	CVPR'25	64	31.80	0.92	18.0	0.97
			128	31.60	0.91	18.1	0.93
			256	31.35	0.90	18.2	0.87
Post-processing methods	HiDDeN [58]	ECCV'18	64	32.01	0.88	19.7	0.98
			128	31.80	0.87	19.8	0.85
			256	31.50	0.86	19.9	0.82
	StegaStamp [44]	CVPR'20	64	28.50	0.91	17.9	0.99
			128	28.20	0.90	18.0	0.98
			256	28.00	0.89	18.1	0.94
	MBRS [16]	ACM MM'21	64	38.20	0.96	17.9	0.98
			128	37.90	0.95	18.0	0.96
			256	37.50	0.94	18.2	0.94
	FIN [8]	AAAI'23	64	36.70	0.95	18.3	0.97
			128	36.40	0.94	18.4	0.96
			256	36.10	0.93	18.5	0.96
	MuST [46]	AAAI'24	64	41.20	0.97	17.5	0.98
			128	40.90	0.96	17.6	0.93
			256	40.50	0.95	17.8	0.90
	EditGuard [56]	CVPR'24	64	41.56	0.97	17.8	0.98
			128	41.30	0.96	17.9	0.97
			256	40.90	0.95	18.0	0.97
	SpecGuard	**Ours**	64	42.59	0.98	17.2	0.99
			128	**42.89**	**0.99**	**17.0**	**0.99**
			256	40.86	**0.99**	17.6	0.98

**BL: Bit Length, BRA: Bit Recovery Accuracy*

Table 2. Comparison of SOTA pre-processing and post-processing watermarking methods with SpecGuard without attacks.

Attack Type		Tree-Ring [50]				Stable Signature [34]				StegaStamp [44]				SpecGuard (Ours)			
		Q@0.95P	Q@0.7P	Avg P	Avg Q	Q@0.95P	Q@0.7P	Avg P	Avg Q	Q@0.95P	Q@0.7P	Avg P	Avg Q	Q@0.95P	Q@0.7P	Avg P	Avg Q
Distortions	Rotation	0.464	0.521	0.375	0.648	0.624	0.702	0.594	0.650	0.423	0.498	0.357	0.616	0.863	0.863	0.687	0.653
	Crop	0.592	0.592	0.332	0.463	inf	inf	0.995	0.461	0.602	0.602	0.540	0.451	0.812	0.812	0.998	0.742
	Bright	inf	inf	inf	0.304	inf	inf	0.998	0.305	inf	inf	0.998	0.317	inf	inf	0.998	0.466
	Contrast	inf	inf	0.998	0.243	inf	inf	0.998	0.243	inf	inf	0.998	0.231	inf	inf	0.998	0.556
	Blur	0.861	1.112	0.563	1.221	– inf	– inf	0.000	1.204	0.848	0.962	0.414	1.000	0.921	inf	1.000	1.452
	Noise	0.548	inf	0.980	0.395	0.402	0.520	0.870	0.390	inf	inf	1.000	0.360	inf	inf	0.999	0.568
	JPEG	0.499	0.499	0.929	0.284	0.485	0.485	0.793	0.284	inf	inf	0.998	0.263	inf	inf	1.000	0.495
	Geo	0.525	0.593	0.277	0.768	0.850	inf	0.937	0.767	0.663	0.693	0.396	0.733	0.869	0.869	0.865	0.623
	Deg	0.620	inf	0.892	0.694	0.206	0.369	0.300	0.679	0.826	0.975	0.852	0.664	0.895	1.141	0.915	0.749
	Combine	0.539	0.751	0.403	0.908	0.538	0.691	0.334	0.900	0.945	1.101	0.795	0.870	0.979	1.256	0.911	0.952
Regeneration	Regen-Diff	– inf	0.307	0.612	0.323	– inf	– inf	0.001	0.300	0.331	inf	0.943	0.327	inf	inf	0.982	0.477
	Regen-DiffP	inf	0.307	0.601	0.327	– inf	– inf	0.001	0.303	0.333	inf	0.940	0.329	inf	inf	0.982	0.562
	Regen-VAE	0.578	0.578	0.832	0.348	0.545	0.545	0.516	0.339	inf	inf	1.000	0.343	inf	inf	0.995	0.521
	Regen-KLVAE	inf	inf	0.990	0.233 6	– inf	0.176	0.217	0.206	inf	inf	1.000	0.240	inf	inf	0.990	0.492
	Rinse-2xDiff	– inf	0.333	0.510	0.357	– inf	– inf	0.001	0.332	0.391	inf	0.941	0.366	inf	inf	0.993	0.561
	Rinse-4xDiff	– inf	0.355	0.443	0.466	– inf	– inf	0.000	0.438	0.388	inf	0.909	0.477	inf	inf	0.992	0.533
Adversarial	AdvEmbG-KLVAE8	– inf	0.164	0.448	0.253	inf	inf	0.998	0.249	inf	inf	1.000	0.232	inf	inf	1.000	0.456
	AdvEmbB-RN18	0.241	inf	0.953	0.218	inf	inf	0.999	0.212	inf	inf	1.000	0.196	inf	inf	1.000	0.467
	AdvEmbB-CLIP	0.541	inf	0.932	0.549	inf	inf	0.999	0.541	inf	inf	1.000	0.488	inf	inf	1.000	0.436
	AdvEmbB-KLVAE16	0.195	inf	0.888	0.238	inf	inf	0.997	0.233	inf	inf	1.000	0.206	inf	inf	1.000	0.482
	AdvEmbB-SdxlVAE	0.222	inf	0.934	0.221	inf	inf	0.998	0.219	inf	inf	1.000	0.204	inf	inf	1.000	0.492
	AdvCls-UnWM&WM	– inf	0.102	0.499	0.145	inf	inf	0.999	0.101	inf	inf	1.000	0.101	inf	inf	1.000	0.497
	AdvCls-Real&WM	inf	inf	1.000	0.047	inf	inf	0.998	0.092	inf	inf	1.000	0.106	inf	inf	1.000	0.427
	AdvCls-WM1&WM2	– inf	0.101	0.492	0.139	inf	inf	0.999	0.084	inf	inf	1.000	0.129	inf	inf	1.000	0.441

Table 3. Robustness comparison various across attacks using Q@0.95P(↑), Q@0.7P(↑), Avg P(↑) and Avg Q(↑). Here, 'inf' denotes that no attack was sufficient to degrade performance below the threshold, indicating strong robustness, whereas '-inf' signifies that even the weakest attack caused detection to fall below the threshold, reflecting weak robustness.

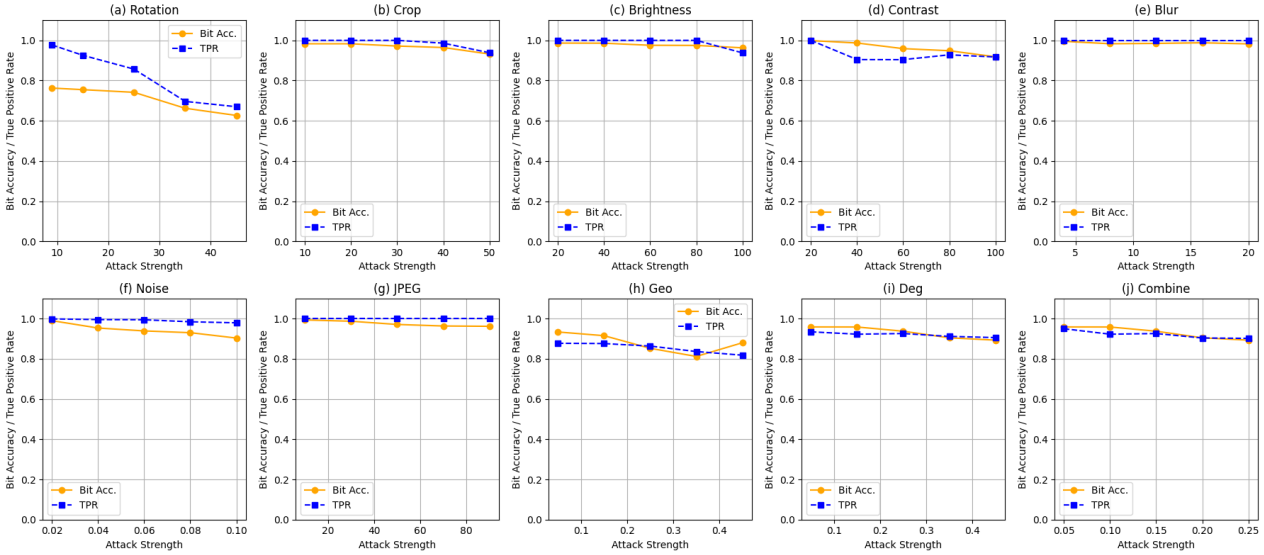

Figure 4. Robustness validation of our proposed SpecGuard under different distortion attacks, including geometric transformations: Geo (rotation, cropping), photometric modifications (brightness, contrast), and degradations: Deg (blur, noise, JPEG compression).

robustness while maintaining imperceptibility. Overall, our results demonstrate that SpecGuard outperforms both pre-processing and post-processing watermarking methods, achieving superior imperceptibility and robustness.

4.4. Capacity

To evaluate embedding capacity, we examined SpecGuard across different bit lengths and compared it with SOTA watermarking methods. Our experiments with 64, 128, and 256 bits demonstrate SpecGuard's high capacity of bit embedding while maintaining perceptual quality and robustness, as shown in Tab. 2. Notably, it achieves a PSNR of 42.89, the highest among all methods, along with the highest BRA of 0.99 and the lowest FID of 17.0 at 128 bits, ensuring minimal visual impact. This adaptability to different bit lengths without quality loss makes SpecGuard ideal

Modules	PSNR/SSIM↑	BRA↑	Modules	PSNR/SSIM↑	BRA↑
WP(L_1)	40.51/0.96	0.92	**WP(L_1)+SP$_{FA}$**	**42.89/0.99**	**0.99**
WP(L_2)	38.15/0.93	0.87	WP(L_2)+SP$_{FA}$	36.25/0.92	0.89
Attacks	**PSNR/SSIM↑**	**BRA↑**	**Attacks**	**PSNR/SSIM↑**	**BRA↑**
Rotate (45°)	12.15/21.31	0.82	Rotate (90°)	11.15/19.31	0.65
Blur (0.3)	35.01/0.95	0.98	Blur (0.6)	30.11/0.91	0.98
Geo (0.3)	12.08/0.50	0.93	Geo (0.6)	10.25/0.45	0.86

***WP:** Wavelet Projection, **SP:** Spectral Projection, FA: FFT Approximation*

Table 4. Ablation studies on the proposed SpecGuard for across various configurations, setting $M = 128$, $r = 100$, and $s = 20$.

Platform	PSNR/SSIM↑	BRA↑	PS Filters	PSNR/SSIM↑	BRA↑
Facebook	48.56/0.97	0.97	Depth Blur	25.25/0.89	0.85
LinkedIn	47.55/0.97	0.96	StyleT.	25.12/0.84	0.85
Instagram	48.56/0.98	0.98	**Super Zoom**	**36.15/0.88**	**0.95**
WhatsApp	42.10/0.96	0.97	JPEG Artifacts	31.01/0.85	0.94
X (Twitter)	**49.25/1.00**	**0.99**	Colorize	23.15/0.82	0.92

Table 5. Evaluation of SpecGuard's robustness across Photoshop filters and while uploaded on different social media platforms.

for applications requiring flexible watermark sizes. Unlike StegaStamp and HiDDeN, which suffer reduced BRA for higher message bits, SpecGuard consistently extracts bits across all tested lengths. SpecGuard's theoretical watermark capacity is provided in the Supplementary.

4.5. Robustness

We evaluate watermarking robustness by analyzing detection performance against a range of diverse and challenging real-world attacks. Results demonstrate the strong robustness of SpecGuard across various attacks. For example, as presented in Tab. 3, against geometric distortions such as cropping and rotation, SpecGuard achieved an Avg P of 0.998 and 0.687, respectively. Similarly, across the combined distortion-based attacks, SpecGuard achieves an overall Avg P of 0.911 and Avg Q of 0.952, ensuring minimal quality loss while maintaining high detection accuracy. Notably, the high values of Q@0.95P and Q@0.7P indicate that SpecGuard can sustain reliable detection at strict performance thresholds, even under aggressive perturbations. Unlike prior methods that struggle with extreme transformations, SpecGuard shows remarkable robustness against regeneration-based attacks like Rinse-2xDiff [4] (an image is noised then denoised by Stable Diffusion v1.4 two times with strength as a number of timesteps, 20-100) and Regen-VAE [4], maintaining high Avg P. Similarly, under adversarial attacks, SpecGuard consistently secures watermark detectability, outperforming existing techniques across all tested scenarios. These results establish SpecGuard as a highly robust watermarking approach capable of preserving image integrity even under severe distortions and adversarial manipulations, ensuring watermark reliability across diverse attack types. More details about how the attacks are performed are provided in supplementary material. Further, our results in Fig. 4 highlight the strong robustness of SpecGuard against various distortion attacks compared to other SOTA watermarking methods.

Social Platforms and Photoshop Filters. SpecGuard's robustness when images are shared across social media platforms and subjected to common Photoshop Neural Filters (PNFs) is shown in Tab. 5. SpecGuard consistently maintains high PSNR and SSIM values, with BRA values close to 0.99 on platforms such as X (formally Twitter), Instagram, and Facebook. Also, it shown strong resilience to various PNFs, such as Super Zoom and JPEG Artifacts achieving BRA of 0.95 and 0.94. The PSNR, SSIM, and BRA values are expected to decrease with the severity of image manipulation, as increased manipulation leads to loss of image authenticity. For example, as we applied 60% style transfer the PSNR and BRA decreased to 25.12 and 0.85. Similarly, the depth blur which excessively reduces the image clarity also causes the decrease of BRA to 0.85.

4.6. Ablation Study

We examined the impact of wavelet projection (WP) at different levels (L_1 and L_2) and its combination with spectral projection (SP) using FFT approximation (FA) in Tab. 4. As observed, the WP(L_1) + SP$_{FA}$ configuration achieved the highest PSNR and SSIM values of 42.89 and 0.99, respectively, and BRA of 0.99, indicating improved watermark invisibility and robustness. In contrast, using WP alone at either L_1 or L_2 resulted in lower BRA, with values of 0.92 and 0.87, respectively, demonstrating that the combined WP + SP$_{FA}$ approach significantly enhances performance. We also evaluated the robustness of SpecGuard under strong adversarial attacks identified in Tab. 4, such as rotation, blur, and geometric transformations. The results indicate that higher levels of attack severity, such as 90° rotation, lead to a more significant drop in PSNR, SSIM, and BRA, with values dropping to 11.15, 19.31, and 0.65, respectively. Despite this, the model shows relatively high resilience under moderate attack intensities, such as 45° rotation and low levels of blur and geometric distortion, achieving BRA values as high as 0.93 under geometric transformations at the 0.3 thresholds. More ablations on SpecGuard are in the supplementary.

5. Conclusion

We propose SpecGuard, a novel invisible watermarking method that ensures secure and robust information concealment. Unlike traditional approaches, SpecGuard remains highly resilient against diverse distortions, adversarial attacks, and regeneration-based transformations. Experimental results demonstrate its superior bit recovery accuracy of 99% maintaining high PSNR. By outperforming SOTA watermarking methods in both detection reliability and imperceptibility, SpecGuard establishes a new benchmark for watermarking under real-world constraints.

SpecGuard: Spectral Projection-based Advanced Invisible Watermarking

Supplementary Material

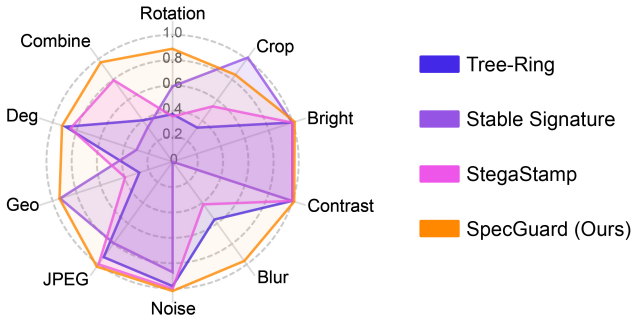

Figure 5. Comparison of SOTA watermarking methods in terms of average TPR@0.1%FPR (90% of watermarked images are correctly detected at 0.1% false positive rate) under different attacks.

6. Summary of Notations

To ensure clarity in understanding SpecGuard's mathematical formulation, we summarize the key notations used throughout the methodology (Sec. 3) of the main paper. The complete set of notations is presented in Tab. 6.

7. Impact of Parseval's Theorem in Message Extraction

To achieve robust and efficient decoding as detailed in Sec. 3.2 of the main paper, SpecGuard leverages Parseval's theorem [19], a fundamental principle in signal processing, which establishes energy equivalence between spatial and spectral domains. Formally, Parseval's theorem is defined as follows:

$$\sum_{x,y} |I(x,y)|^2 = \sum_{u,v} |\zeta(u,v)|^2, \tag{21}$$

where $I(x,y)$ denotes spatial-domain pixel intensities, and $\zeta(u,v)$ represent their corresponding spectral-domain coefficients.

In SpecGuard, watermark embedding modifies selected spectral coefficients, introducing subtle local energy variations. The embedding process employs a strength factor s, adjusting spectral energy differences as follows:

$$\zeta_{\text{embedded}}(u,v) = \zeta(u,v) + s \cdot W(u,v), \tag{22}$$

where $\zeta_{\text{embedded}}(u,v)$ denotes modified coefficients and $W(u,v)$ is the spectral-domain watermark signal. Although local energy distribution is altered, the overall signal energy remains constant as guaranteed by Parseval's theorem as follows:

$$\sum_{x,y} |I_(x,y)|^2 = \sum_{u,v} |\zeta_{\text{embedded}}(u,v)|^2. \tag{23}$$

During decoding, these local spectral energy variations, preserved due to total energy constancy, allow stable watermark extraction. Specifically, the decoder computes spectral projections via FFT approximation to isolate embedded spectral energy patterns as follows:

$$S^{\text{sp}}_{D_{HH}} = \text{SpectralProjectionFFT}(S^{\text{high}}_{D_{HH}}). \tag{24}$$

The decoder subsequently employs a dynamically optimized threshold θ to differentiate watermark signals from noise as follows:

$$D_M[i] = \begin{cases} 1 & \text{if } S^{\text{sp}}_{D_{HH}}[i] > \theta, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

The adaptive threshold θ is optimized via gradient descent during training, adapting to spectral energy distributions as follows:

$$\theta \leftarrow \theta - \eta \cdot \frac{\partial L_{\text{dec}}}{\partial \theta}, \tag{26}$$

where L_{dec} is the decoding loss, and η is the learning rate. Thus, Parseval's theorem critically supports SpecGuard by preserving total spectral energy, enabling stable differentiation of watermark bits and reliable decoding even under diverse real-world image distortions and adversarial attacks.

8. Mathematical Proof

8.1. Proof for S_{HH} Band of Wavelet Projection.

Here we presented a proof of one of the wavelet projections S_{HH} from Eq. (4) based on the Eq. (6) of the main paper.

$$\psi^D_j(u) = 2^{j/2}\psi^D(2^j u), \quad \text{//1D wavelet}$$

$$\psi^D_{j,m}(u) = 2^{j/2}\psi^D(2^j u - m), \quad \text{//Translation}$$

$$\psi^D_{j,m,n}(u,v) = 2^{j/2}\psi^D(2^j u - m)\cdot\psi^D(2^j v - n), \quad \text{//2D wavelet}$$

$$S_{HH}(j,m,n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(u,v)\cdot \psi^D_{j,m,n}(u,v)\,du\,dv, \quad \text{//Projection}$$

$$S_{HH}(j,m,n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(u,v)\cdot\left[2^{j/2}\psi^D(2^j u - m)\cdot\psi^D(2^j v - n)\right]\,du\,dv, \quad \text{//Substitution}$$

Notation	Description
I	Cover image
I_{embedded}	Watermarked image
M	Watermark message
c	Number of channels (e.g., RGB has $c = 3$)
H, W	Height and width of the image
$W(a, b)$	Wavelet transform of signal $f(x)$
a, b	Scaling and translation parameters in wavelet transform
ψ	Mother wavelet function
$d)$	Direction of each wavelet components derived from ψ
$\phi(u, v), \psi_H(u, v), \psi_V(u, v), \psi_D(u, v)$	Every directional scaling and wavelet basis components
$S_{LL}, S_{LH}, S_{HL}, S_{HH}$	Wavelet sub-bands (low and high frequency components)
β_j	Feature set capturing frequency and spatial details
κ	Decomposition level determined by image complexity
$T(x, y)$	Pixel intensity in high-frequency sub-band S_{HH}
$\zeta(u, v)$	Spectral projection coefficients
s	Strength factor controlling embedding intensity
(c_x, c_y)	Center coordinates of the image
$D(x_i, y_i)$	Euclidean distance from the center
r	Radius of embedding region
W_c	Selected watermark channel for embedding
θ	Learnable threshold for watermark extraction
$F(u, v)$	2D Fast Fourier Transform (FFT) of the extended signal
$L_{\text{enc}}, L_{\text{dec}}$	Encoder and decoder loss functions

Table 6. Description of the notations we used in the Sec. 3 (main paper) to describe our proposed SpecGuard.

$$S_{HH}(j, m, n) = \sum_{p=0}^{l-1} \sum_{q=0}^{l-1} T_{m,n} \cdot \psi^D(2^j u - m) \cdot \psi^D(2^j v - n), \quad \text{//Discretization}$$

$$W_\psi^d(j, u, v) = \frac{1}{l} \sum_{m=0}^{l-1} \sum_{n=0}^{l-1} T_{m,n} \cdot \psi^D\left(m - u \cdot 2^{-j}, n - v \cdot 2^{-j}\right), \quad \text{//Normalized}$$

8.2. Maximum Theoretical Watermark Capacity

To determine the maximum theoretical watermark capacity of SpecGuard, we analyze the SpecGuard's embedding pipeline, which integrates wavelet projection and spectral projection. The capacity derivation considers three key stages: 'wavelet projection,' 'spectral projection,' and 'watermark distribution,' with each stage affecting the number of available coefficients for embedding.

Impact of Wavelet Projection. SpecGuard applies wavelet projection at decomposition level L, dividing the image into sub-bands. The watermark is embedded in the high-frequency sub-band, which retains fine image details and ensures robustness against low-frequency distortions. The spatial dimensions of the wavelet sub-band are reduced by a factor of 2^L along both height and width, resulting in a down-sampling effect.

The number of available coefficients after wavelet de-

Activation Function	Radius (r)	PSNR↑	SSIM↑	BRA↑
ReLU	r(50)	39.54	0.93	0.97
	r(75)	38.64	0.91	0.93
	r(100)	37.96	0.91	0.95
Tanh	r(50)	37.18	0.89	0.82
	r(75)	35.33	0.85	0.78
	r(100)	37.66	0.90	0.80
LeakyReLU	r(50)	39.77	0.96	0.98
	r(75)	40.28	0.97	0.98
	r(100)	**42.89**	**0.99**	**0.99**

Table 7. Performance evaluation of SpecGuard for different radius size and activation functions while the Strenth Factor is 20.

Activation Function	Strength Factor (s)	PSNR↑	SSIM↑	BRA↑
LeakyReLU	$s(5)$	40.79	0.98	0.97
LeakyReLU	$s(10)$	39.51	0.96	0.97
LeakyReLU	$s(15)$	38.14	0.95	0.99
LeakyReLU	**$s(20)$**	**42.89**	**0.99**	**0.99**

Table 8. Impact of Strength Factor for the best combination of the activation function (LeakyReLU) and radius $r(100)$.

composition is as follows:

$$N_{\text{WP}} = \frac{H \times W}{4^L}, \tag{27}$$

where H and W are the image height and width, respectively. Including all image channels c, the total number of wavelet coefficients available for embedding is as follows:

$$N_{\text{WP,total}} = \frac{H \times W \times c}{4^L}. \tag{28}$$

Thus, increasing the decomposition level L reduces the available spatial coefficients exponentially, limiting embedding capacity.

Impact of Spectral Project. SpecGuard employs spectral projection using FFT to distribute the watermark in the spectral domain. The spectral coefficients are selectively utilized based on an adaptive mask that prioritizes mid-to-high-frequency components while avoiding low frequencies (which contain most perceptual information) and extremely high frequencies (which are prone to compression loss).

The fraction of spectral coefficients selected for watermarking is denoted as f_{spectral} where spectral coefficients are used in between 20% and 50% as follows:

$$0.2 \leq f_{\text{spectral}} \leq 0.5. \tag{29}$$

After spectral projection following Eq. (28), the number of coefficients available for embedding is as follows:

$$N_{\text{SP}} = f_{\text{spectral}} \times N_{\text{WP,total}} = f_{\text{spectral}} \times \frac{H \times W \times c}{4^L}. \tag{30}$$

A higher f_{spectral} increases embedding capacity but may reduce robustness to compression and noise, while a lower f_{spectral} focuses on the most resilient coefficients but limits capacity.

Watermark Distribution and Final Capacity. The watermark is distributed across the selected spectral coefficients f_{spectral} using a weighting scheme, where each coefficient can embed multiple bits. Let N_b represent the number of watermark bits per selected coefficient f_{spectral}. The total embedded bits are then as follows:

$$C_{\text{total}} = N_b \times N_{\text{SP}}. \tag{31}$$

Substituting N_{SP}, the final maximum theoretical watermark capacity of SpecGuard is as follows:

$$C_{\max}(H, W, c, L, f_{\text{spectral}}, N_b) = \frac{H \times W \times c}{4^L} \times f_{\text{spectral}} \times N_b. \tag{32}$$

The watermark capacity scales proportionally with the image dimensions $H \times W$ and the number of channels c, ensuring that larger images provide greater embedding space. However, higher wavelet decomposition levels L reduce the available capacity exponentially due to the 4^L down-sampling effect. The fraction of spectral coefficients selected for embedding, denoted as f_{spectral}, controls how much of the frequency domain is utilized, balancing capacity and robustness. Additionally, the bit depth N_b determines the number of bits embedded per coefficient, directly influencing the total watermark payload.

Thus, SpecGuard achieves a flexible balance between capacity and robustness by leveraging adaptive spectral selection and wavelet decomposition, ensuring resilience under various transformations and attacks.

9. Impact of Hyperparamters

The performance of SpecGuard is influenced by several key hyperparameters, including the activation function, radius size (r), and strength factor (s). Each parameter plays a vital role in balancing the trade-off between perceptual quality, robustness, and watermark recovery accuracy. In addition to the ablation studies shown in Section 4.5 in the main paper, here we analyze the effect of the hyperparameters individually by conducting experiments under controlled conditions and report the findings in Tab. 7 and Tab. 8. All the experiments presented here were conducted using a 128-bit watermark message.

9.1. Activation Function and Radius

Table 7 highlights the performance of SpecGuard with various activation functions, including ReLU [12], Tanh [38], and LeakyReLU [52], while keeping the strength factor s fixed at 20. Among these, LeakyReLU outperforms others in terms of PSNR, SSIM, and bit recovery accuracy values across different radius sizes. Notably, with a radius r of 100, LeakyReLU achieves a PSNR and SSIM of 42.89 and 0.99, respectively, with a bit recovery accuracy of 0.99. Overall, the results indicate the effectiveness of LeakyReLU for robust and invisible watermarking compared to ReLU and Tanh. While testing with different r, such as 50 and 75, we observed a slightly lower perceptual quality and bit recovery accuracy. Therefore, we propose the SpecGuard with a combination of LeakyReLU, r of 100 and s of 20.

9.2. Strength Factor

Table 8 investigates the impact of the strength factor (s) using the best combination of LeakyReLU and radius $r(100)$.

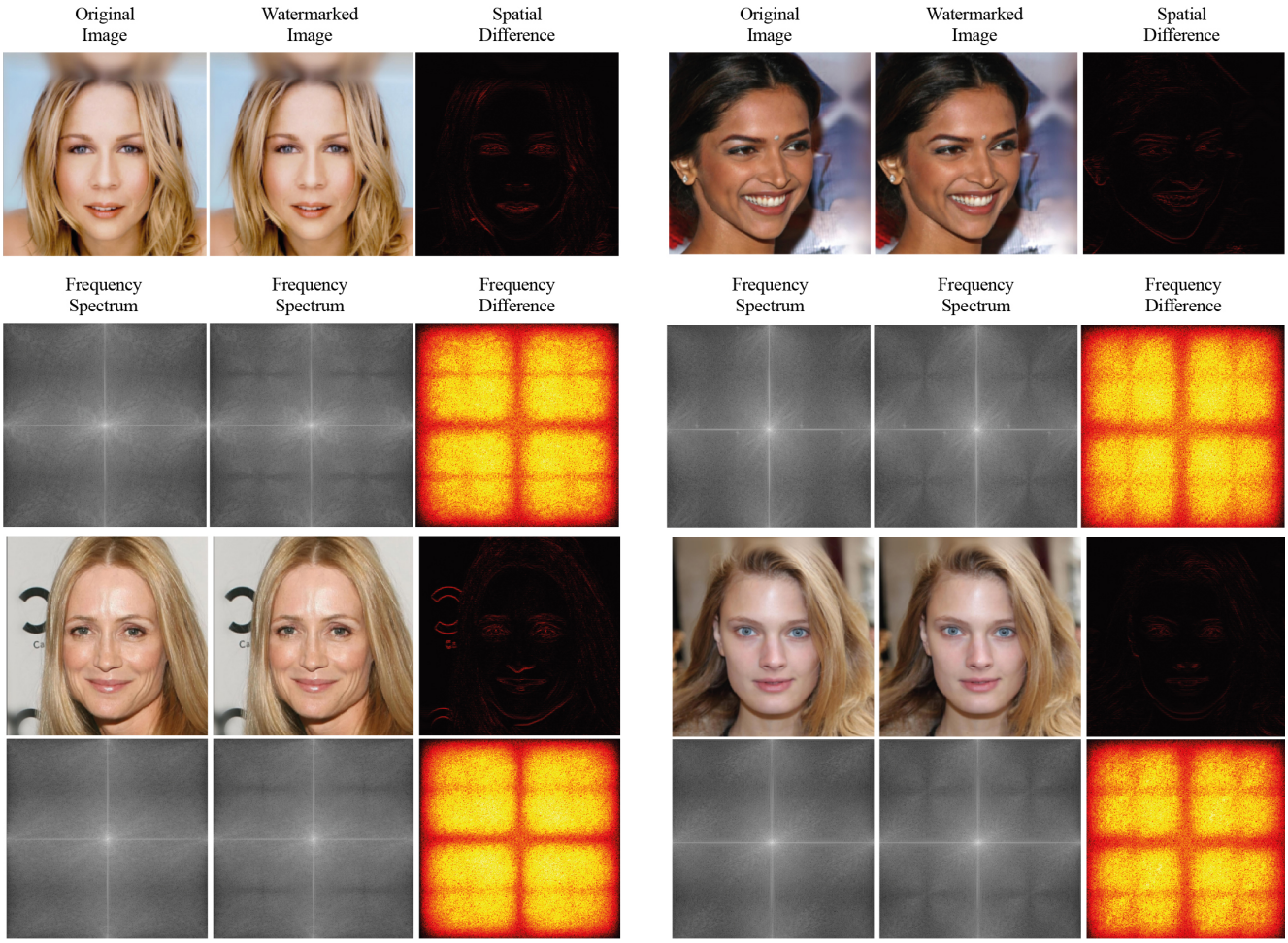

Figure 6. Visualization of the watermarking process using SpecGuard. The first row shows the original image, the watermarked image, and their spatial difference. The spatial difference highlights the minimal perceptual change between the original and watermarked images, ensuring imperceptibility. The second row presents the frequency spectrum of the original and watermarked images, along with their frequency difference, emphasizing the subtle embedding of the watermark in the high-frequency components. The comparison confirms that SpecGuard achieves invisible watermarking while maintaining robust frequency-domain characteristics for effective bit recovery.

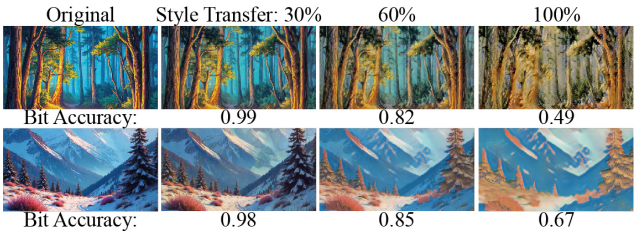

Figure 7. Effect of style transfer severity on bit recovery accuracy. As style intensity increases, bit accuracy decreases, showing the impact of major transformations.

A strength factor of $s(20)$ achieves optimal performance with a PSNR/SSIM of 42.89/0.99 and a BRA of 0.99. Increasing s beyond 20 reduces PSNR and SSIM values, indicating diminished perceptual quality, while lower strength factors compromise robustness. Therefore, $s(20)$ effec-

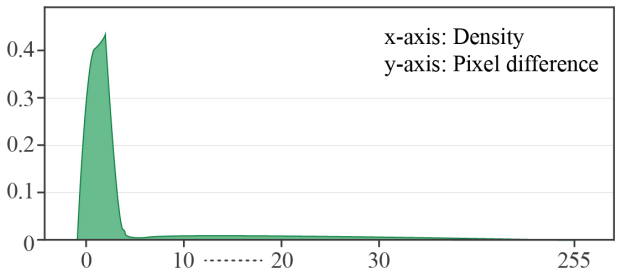

Figure 8. Pixel difference distribution between the original and watermarked images. The x-axis represents the pixel intensity difference, and the y-axis indicates the density. Most pixel differences remain close to zero, highlighting SpecGuard's minimal perceptual loss and superior imperceptibility of the embedded watermark.

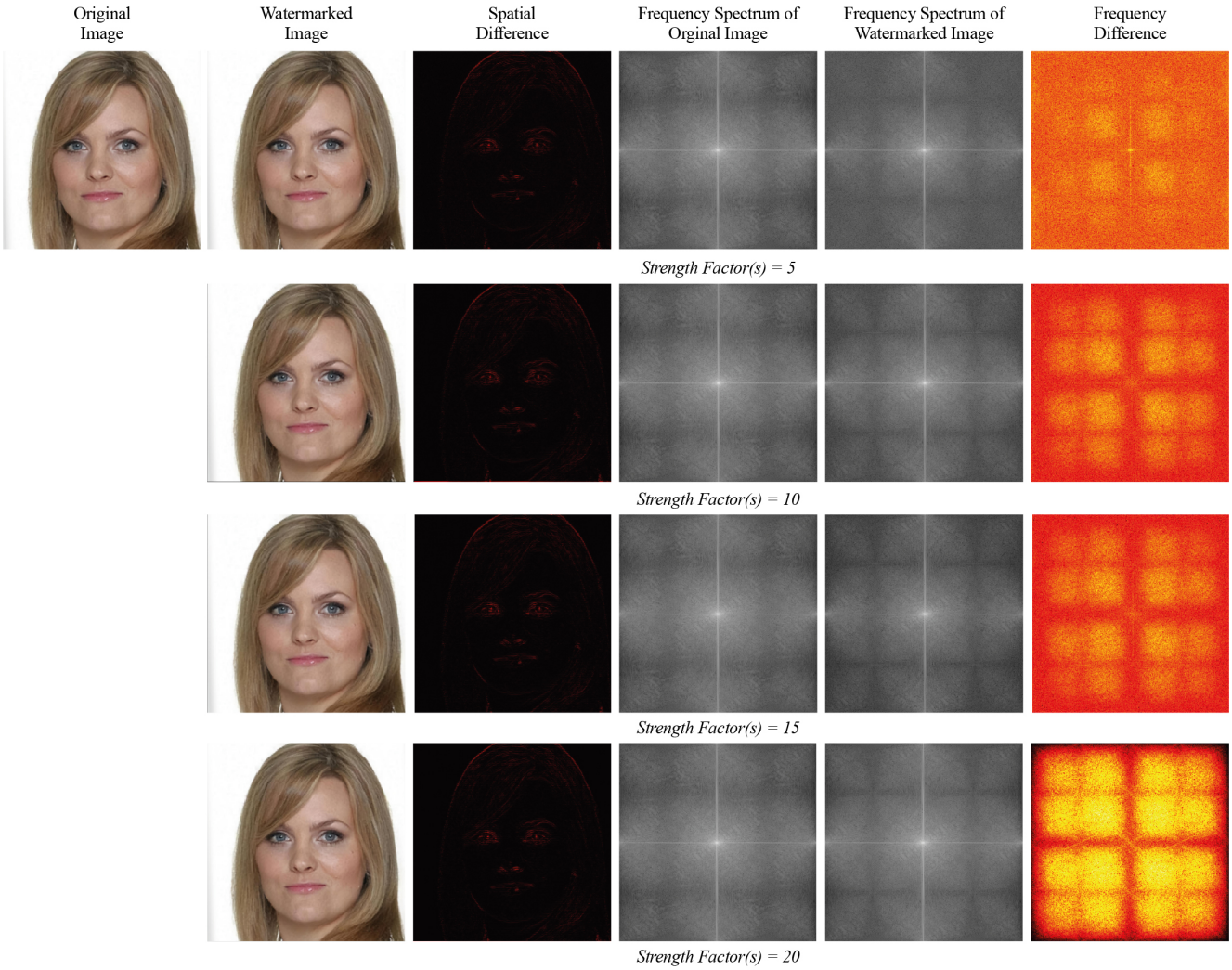

Figure 9. Visualization of the watermarking process using SpecGuard for different strength factors (s). The first row illustrates the original image, the watermarked image, and their spatial difference for $s = 5$, followed by the frequency spectra of the original and watermarked images and their frequency difference. The subsequent rows demonstrate the impact of increasing the strength factor ($s = 10, 15, 20$) on the frequency difference, highlighting the progressive embedding intensity. Higher strength factors increase the visibility in the frequency domain while maintaining imperceptibility in the spatial domain, ensuring robust watermarking without compromising image quality.

tively balances robustness and visual quality as also shown in Fig. 6.

Figure 9 further demonstrates the effect of different strength factors ($s = 5, 10, 15, 20$) on the watermark embedding process. The first row showcases the original image, the watermarked image, and their spatial difference, highlighting the imperceptibility of the watermark in the spatial domain. The subsequent rows compare the frequency spectrum of the original and watermarked images, as well as the frequency difference, illustrating how increased strength factors enhance the visibility of the watermark in the frequency domain while maintaining imperceptibility in the spatial domain. Illustrate the robustness and adaptability of the proposed SpecGuard model in embedding and retaining watermark information under varying conditions.

10. Description of Benchmarking Attacks

To comprehensively evaluate watermark robustness, we benchmark performance against a diverse set of attacks, including distortions, regeneration, and adversarial manipulations. These attacks, derived from prior benchmarking efforts [6], assess the stability of watermarks under real-world transformations. The results are presented in Tab. 3 (main paper) and the details of the attacks are in Tab. 9, comparing multiple state-of-the-art (SOTA) methods such as Tree-

Attack Name	Description	Parameters
	Distortion Attacks	
Rotation	Rotates an image by a specified angle to test watermark robustness against geometric transformations.	Angle: 9° to 45° clockwise
Crop	Crops a portion of the image and resizes it back, simulating common editing.	Crop Ratio: 10% to 50%
Bright	Adjusts image brightness to test watermark stability under illumination changes.	Brightness Increase: 20% to 100%
Contrast	Modifies image contrast to simulate lighting variations.	Contrast Increase: 20% to 100%
Blur	Applies a low-pass filter to smooth the image, reducing high-frequency details.	Kernel Size: 4 to 20 pixels
Noise	Introduces random pixel fluctuations to simulate compression noise and low-quality rendering.	Std. Deviation: 0.02 to 0.1
JPEG	Compresses the image using JPEG encoding, reducing quality and adding artifacts.	Quality Score: 90 to 10
Geo	Combination of geometric distortion attacks, including rotation, crop, applied uniformly to assess cumulative effects.	Strength: Geo(x): Rotation: $9° + x \times (45° - 9°)$, Crop: $10\% + x \times (50\% - 10\%)$
Deg	Combination of degradation attacks, integrating blur, noise, and JPEG to simulate complex real-world distortions.	Strength: Deg(x): Blur: $4 + x \times (20 - 4)$, Noise: $0.02 + x \times (0.1 - 0.02)$, JPEG: $90 - x \times (90 - 10)$
	Regeneration Attacks	
Regen-Diff	Passes an image through a diffusion model to reconstruct a similar but altered version.	Denoising Steps: 40 to 200
Regen-DiffP	A prompted version of diffusion-based regeneration, leveraging text guidance to refine results.	Denoising Steps: 40 to 200 with Prompt
Regen-VAE	Uses a variational autoencoder to encode and decode an image, affecting watermark integrity.	Quality Level: 1 to 7
Regen-KLVAE	Uses a KL-regularized autoencoder to compress and reconstruct an image, weakening watermark signals.	Bottleneck Sizes: 4, 8, 16, 32
Rinse-2xDiff	Applies a two-stage diffusion regeneration, progressively altering the image over multiple steps.	Timesteps: 20 to 100 per diffusion
Rinse-4xDiff	Performs four cycles of diffusion-based image reconstruction, aggressively erasing watermark traces.	Timesteps: 10 to 50 per diffusion
	Adversarial Attacks	
AdvEmbG-KLVAE8	Embeds adversarial perturbations using a grey-box VAE-based attack to reduce detection accuracy.	KL-VAE Encoding, ϵ = 2/255 to 8/255, PGD Iterations = 100, Step Size = $0.01 \times \epsilon$
AdvEmbB-RN18	Uses a pre-trained ResNet18 model to introduce adversarial noise and affect watermark recognition.	ℓ_∞ Perturbation: 2/255 to 8/255, PGD Iterations = 50, Step Size = $0.01 \times \epsilon$
AdvEmbB-CLIP	Attacks the CLIP image encoder to introduce embedding shifts that disrupt watermark decoding.	ℓ_2 Perturbation Norm = 2.5, PGD Iterations = 50, Learning Rate = 0.001
AdvEmbB-KLVAE16	Uses an alternative KL-VAE model to introduce structured perturbations into the embedding process.	KL-VAE Embedding, Latent Size = 16, ℓ_∞ Perturbation = 4/255
AdvEmbB-SdxlVAE	Attacks Stable Diffusion XL's VAE encoder to alter latent representations and remove watermarks.	Targeted VAE Perturbation, Diffusion Steps = 100, ℓ_2 Perturbation = 3.0
AdvCls-UnWM&WM	Trains a surrogate detector on watermarked and non-watermarked images to bypass watermark detection.	Dataset Size = 3000 Images (1500 Per Class), ResNet-18, Learning Rate = 0.001, Batch Size = 128
AdvCls-Real&WM	Trains an adversarial classifier using real and watermarked images to classify watermark presence.	Dataset Size = 15,000 Images (7500 Per Class), Adam Optimizer, Learning Rate = 0.0005, Batch Size = 128, Epochs = 10
AdvCls-WM1&WM2	Exploits watermark signal variations between different users to remove or alter hidden information.	Two Sets of Watermarked Images, Model = Vision Transformer (ViT), PGD Attack, Perturbation Strength = 6/255

Table 9. Overview of attack types, their mechanisms, and key parameters based on the prior study [6] that we also utilized in our study.

Ring [50], Stable Signature [34], and StegaStamp [44]. The attacks are categorized as follows:

10.1. Distortion Attacks

These include standard image-processing transformations that alter the spatial or color properties of images. We consider rotation (9° to 45°) where images are rotated at varying degrees to test watermark stability. Resized cropping (10% to 50%) removes portions of an image and resizes the remaining content, mimicking common real-world editing. Random erasing (5% to 25%) replaces regions with gray pixels, simulating object removal. Brightness adjustments (20% to 100%) and contrast modifications (20% to 100%) simulate lighting variations. Gaussian blur (4 to 20 pixels) applies low-pass filtering, while Gaussian noise (0.02 to 0.1 standard deviation) adds random pixel fluctuations, simulating compression noise [6].

10.2. Regeneration Attacks

These attacks leverage generative models such as diffusion and variational autoencoders (VAEs) to reconstruct images while suppressing embedded watermarks. We evaluate single regeneration attacks including Regen-Diff (diffusion-based reconstruction), Regen-DiffP (perceptually optimized diffusion), Regen-VAE (autoencoder-based reconstruction), and Regen-KLVAE (KL-regularized VAE reconstruction). Additionally, multi-step regeneration attacks such as Rinse-2xDiff and Rinse-4xDiff involve iterative diffusion processes designed to further erase watermark traces [39, 57].

10.3. Adversarial Attacks

These attacks attempt to deceive watermark detectors through embedding perturbations or surrogate model training. Grey-box embedding attacks (AdvEmbG-KLVAE8) perturb watermarks while preserving image content. Black-box embedding attacks (AdvEmbB-RN18, AdvEmbB-CLIP, AdvEmbB-KLVAE16, AdvEmbB-SdxlVAE) introduce noise during watermark embedding to decrease detection confidence. Adversarial classifiers (AdvCls-UnWM&WM, AdvCls-Real&WM, AdvCls-WM1&WM2) use learned classifiers to distinguish watermarked images and remove hidden signals [14, 35, 37, 39].

Overall, our evaluation framework ensures a rigorous assessment of watermark robustness under various real-world transformations and adversarial strategies.

References

[1] Edward H Adelson, Eero Simoncelli, and Rajesh Hingorani. Orthogonal pyramid transforms for image coding. In *Visual Communications and image processing II*, pages 50–58. SPIE, 1987. 2

[2] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020. 2

[3] Aashutosh AV, Srijan Das, Abhijit Das, et al. Latent flow diffusion for deepfake video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3781–3790, 2024. 1

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 8

[5] Tali Dekel, Michael Rubinstein, Ce Liu, and William T Freeman. On the effectiveness of visible watermarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2146–2154, 2017. 2

[6] Mucong Ding, Tahseen Rabbani, Bang An, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, et al. Waves: Benchmarking the robustness of image watermarks. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024. 1, 6, 5, 7

[7] Hubert Etienne. The future of online trust (and why deepfake is advancing it). *AI and Ethics*, 1(4):553–562, 2021. 1

[8] Han Fang, Yupeng Qiu, Kejiang Chen, Jiyi Zhang, Weiming Zhang, and Ee-Chien Chang. Flow-based robust watermarking with invertible noise layer for black-box distortions. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5054–5061, 2023. 6

[9] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised gan watermarking for intellectual property protection. In *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2022. 2

[10] S. A. Fulling. The local geometric asymptotics of continuum eigenfunction expansions. ii. one-dimensional systems. *SIAM Journal on Mathematical Analysis*, 14(4):605–623, 1983. 2

[11] Mahdieh Ghazvini, Elham Mohamadi Hachrood, and Mojdeh Mirzadi. An improved image watermarking method in frequency domain. *Journal of Applied Security Research*, 12 (2):260–275, 2017. 2

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011. 3

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[15] K Jayashre and M Amsaprabhaa. Safeguarding media integrity: A hybrid optimized deep feature fusion based deepfake detection in videos. *Computers & Security*, 142:103860, 2024. 1

[16] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 41–49, 2021. 6

[17] Hoon Kang and Joonsoo Ha. Projection spectral analysis. *International Journal of Control, Automation and Systems*, 13(6):1530–1537, 2015. 1, 2, 4

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 6

[19] SS Kelkar, LL Grigsby, and J Langsner. An extension of parseval's theorem and its use in calculating transient energy in the frequency domain. *IEEE Transactions on Industrial Electronics*, (1):42–45, 1983. 2, 5, 1

[20] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*, 2021. 2

[21] Jae-Eun Lee, Young-Ho Seo, and Dong-Wook Kim. Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. *Applied Sciences*, 10(19):6854, 2020. 2

[22] Yicheng Leng, Chaowei Fang, Gen Li, Yixiang Fang, and Guanbin Li. Removing interference and recovering content imaginatively for visible watermark removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2983–2990, 2024. 2

[23] Xiao Li, Liquan Chen, Ju Jia, Zhongyuan Qin, and Zhangjie Fu. A lightweight image forgery prevention scheme for iot using gan-based steganography. *IEEE Transactions on Industrial Informatics*, 2024. 2

[24] Dongdong Lin, Benedetta Tondi, Bin Li, and Mauro Barni. A cyclegan watermarking method for ownership verification. *IEEE Transactions on Dependable and Secure Computing*, 2024. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5, 6

[26] Yang Liu, Zhen Zhu, and Xiang Bai. Wdnet: Watermark-decomposition network for visible watermark removal. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3685–3693, 2021. 2

[27] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13548–13557, 2020. 2

[28] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. 1

[29] Guangyu Nie, Changhoon Kim, Yezhou Yang, and Yi Ren. Attributing image generative models using latent fingerprints. In *International Conference on Machine Learning*, pages 26150–26165. PMLR, 2023. 2

[30] Li Niu, Xing Zhao, Bo Zhang, and Liqing Zhang. Fine-grained visible watermark removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12770–12779, 2023. 2

[31] Konstantin A Pantserev. The malicious use of ai-based deepfake technology as the new threat to psychological security and political stability. *Cyber defence in the age of AI, smart societies and augmented humanity*, pages 37–55, 2020. 1

[32] Ram Shankar Pathak. *The wavelet transform*. Springer Science & Business Media, 2009. 1, 2, 4

[33] Fernandez Pierre, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3054–3058. IEEE, 2022. 2

[34] Fernandez Pierre, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 1, 6, 7

[35] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7

[36] Tong Qiao, Yuyan Ma, Ning Zheng, Hanzhou Wu, Yanli Chen, Ming Xu, and Xiangyang Luo. A novel model watermarking for protecting generative adversarial network. *Computers & Security*, 127:103102, 2023. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 7

[38] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6, 1986. 3

[39] Mehrdad Saberi, Vinu Sankar Sadasivan, Keivan Rezaei, Aounon Kumar, Atoosa Chegini, Wenxiao Wang, and Soheil Feizi. Robustness of ai-image detectors: Fundamental limits and practical attacks. *arXiv preprint arXiv:2310.00076*, 2023. 7

[40] Sunpreet Sharma, Ju Jia Zou, Gu Fang, Pancham Shukla, and Weidong Cai. A review of image watermarking for identity protection and verification. *Multimedia Tools and Applications*, 83(11):31829–31891, 2024. 2

[41] Qingtang Su, Huanying Wang, DeCheng Liu, Zihan Yuan, and Xueting Zhang. A combined domain watermarking algorithm of color image. *Multimedia Tools and Applications*, 79(39):30023–30043, 2020. 2

[42] Qingtang Su, Xueting Zhang, and Huanying Wang. A blind color image watermarking algorithm combined spatial domain and svd. *International Journal of Intelligent Systems*, 37(8):4747–4771, 2022. 2

[43] Ruizhou Sun, Yukun Su, and Qingyao Wu. Denet: disentangled embedding network for visible watermark removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2411–2419, 2023. 2

[44] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2126, 2020. 1, 6, 7

[45] D Vaishnavi and TS Subashini. Robust and invisible image watermarking in rgb color space using svd. *Procedia Computer Science*, 46:1770–1777, 2015. 2

[46] Guanjie Wang, Zehua Ma, Chang Liu, Xi Yang, Han Fang, Weiming Zhang, and Nenghai Yu. Must: Robust image watermarking for multi-source tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5364–5371, 2024. 6

[47] Huanying Wang and Qingtang Su. A color image watermarking method combined qr decomposition and spatial domain. *Multimedia Tools and Applications*, 81(26):37895–37916, 2022. 2

[48] Zilan Wang, Junfeng Guo, Jiacheng Zhu, Yiming Li, Heng Huang, Muhao Chen, and Zhengzhong Tu. Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2412.04852*, 2024. 6

[49] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022. 5

[50] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, pages 58047–58063. Curran Associates, Inc., 2023. 1, 6, 7

[51] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[52] Bing Xu. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 3

[53] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024. 1, 6

[54] Zihan Yuan, Qingtang Su, Decheng Liu, and Xueting Zhang. A blind image watermarking scheme combining spatial domain and frequency domain. *The visual computer*, 37:1867–1881, 2021. 2

[55] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33:10223–10234, 2020. 2

[56] Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11964–11974, 2024. 6

[57] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. *Advances in Neural Information Processing Systems*, 37:8643–8672, 2025. 7

[58] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 6