# AWPD: Frequency Shield Network for Agnostic Watermark Presence Detection

Appendix

March 31, 2026

## A UniFreq-100K Dataset Construction Details

To comprehensively evaluate the model's performance on the Agnostic Watermark Presence Detection (AWPD) task, we constructed the large-scale UniFreq-100K dataset. This appendix elaborates on the sources of various watermark carrier images in the dataset, the embedding settings for different watermarking algorithms, specific engineering implementation details, and related copyright statements.

### A.1 Selection and Sampling of Watermarked Images

To ensure the model can generalize to real-world open scenarios, we sampled from the following distributions when constructing the unwatermarked "Clean Images" and the "Watermarked Images" used for embedding:

- **Real Photos**: Uniformly and randomly sampled from the widely used COCO 2017 training set (train2017).
- **AIGC Images**: Covering various artistic style images generated by mainstream generative foundation models.
- **2D/Digital Art, Electronic Posters, and Scanned Drawings**: Proportionally collected from various image galleries.

### A.2 Watermarking Algorithm and Parameter Settings

For the 9 representative invisible watermarking algorithms selected, we implemented refined experimental controls during the embedding phase to ensure the fairness and rigor of the evaluation system. Unless otherwise specified, all watermarked images are ultimately uniformly resized to a $256 \times 256$ resolution. To simulate fixed identity tracing in real-world scenarios, we uniformly adopted a random 32-bit binary sequence (32-bit Payload) as the embedding target for

traditional algorithms and deep learning baseline algorithms. For algorithms with a payload length exceeding 32 bits, we applied 32-bit cyclic padding or random padding to reach the specified length.

### A.2.1 Traditional Spatial and Frequency Domain Baseline Algorithms

The specific implementation details for the four classical algorithms (LSB, Patchwork, DCT, DWT) are as follows:

- **Least Significant Bit (LSB) Steganography**: As the basic method for spatial domain steganography, this algorithm directly manipulates the B (Blue) channel in the image's BGR color space. To enhance the watermark's robustness (especially against local cropping attacks), we introduced a full-image redundancy mechanism: the 32-bit target payload is cyclically tiled to cover and replace the least significant bits of all pixels in the B channel of the entire image.
- **Patchwork Algorithm**: This is a spatial domain modification technique based on statistical features. For each bit of the 32-bit information, the algorithm uses a fixed random seed to generate 100 pseudo-random pixel pairs. Depending on the embedded bit value (1 or 0), a luminance shift is applied to the corresponding pixel pairs on the B channel, with a fixed modification step size of $d = 5$.
- **Discrete Cosine Transform (DCT) Watermarking**: The algorithm first converts the image to the YUV color space. A global 2D discrete cosine transform is applied to the Y (luminance) component, concentrating the image energy in the low-frequency region. We use a random seed to generate a standard normal distribution noise pattern and modulate the 32-bit information into this pattern using an intensity coefficient of $\alpha = 15.0$. To ensure imperceptibility, the embedding process avoids the direct current (DC) component and only superimposes on the alternating current (AC) coefficients.
- **Discrete Wavelet Transform (DWT) Watermarking**: On the Y channel of the YUV space, a two-level decomposition is performed using the Haar wavelet basis, obtaining four frequency sub-bands: LL, HL, LH, and HH. We embed the spread-spectrum random signal carrying the information (intensity coefficient $\alpha = 15.0$) into the HL sub-band, which contains horizontal high-frequency information. This method offers strong resistance to cropping and scaling attacks.

For the convenience of future reproducibility, the codes for the aforementioned baseline algorithms are provided in Appendix D.

### A.2.2 Deep Learning Watermarking Algorithms

For invisible watermarks driven by end-to-end neural networks, this study utilized officially released open-source code implementations.

- **HiDDeN**: Utilizes an adversarial training architecture containing an Encoder, a Decoder, and a Noise Layer. We used the training scripts from the original code to train the model. Subsequently, we embedded a random 32-bit code into the images using the trained model.

- **StegaStamp**: A robust algorithm designed against physical-world imaging perturbations. Its encoder embeds information into global image features, demonstrating exceptionally strong resistance to rotation, illumination changes, and physical recapture. We implemented it using its open-source code and weights. Since this algorithm can embed actual text content, our embedded payload was "Hello_world".

#### A.2.3 Generative and AIGC Watermarking Algorithms

To address the copyright and authenticity identification challenges brought by generative AI such as diffusion models, the dataset integrates watermarking technologies directly injected into the generation pipeline:

- **Stable Signature**: We adopted the official open-source code, using Stable Diffusion 1.5 as the base model. This method lightly fine-tunes the decoder of the LDM, enabling the generated images to automatically carry a specific binary signature when transitioning from the latent space to the pixel space.

- **Tree-Ring**: During the initial latent noise phase of the diffusion model, this algorithm embeds information by applying ring-symmetric constraints on the frequency domain energy distribution. We implemented this algorithm manually. During the experiment, the initial image generation size was set to $400 \times 400$, which was ultimately resized to $256 \times 256$. The experiment used a 32-bit message length and a batch size of 32. The encoder and decoder employed a 4-layer convolutional structure with 64 hidden channels and ReLU activation functions. The model was trained for 300 epochs using the Adam optimizer and a Cosine Annealing learning rate scheduling strategy (initial learning rate of $1 \times 10^{-4}$), with a weight decay of $1 \times 10^{-5}$ set to improve generalization. To strengthen detector performance, a complex augmentation combination was introduced during the training phase, including random cropping ($0.2 \sim 0.25$), Dropout ($0.55 \sim 0.6$), scaling ($0.7 \sim 0.8$), and default-enabled JPEG compression perturbations.

- **SynthID**: For SynthID watermark samples, we generated them by calling Google's provided `imagen-4.0-fast-generate-001` model API. This model natively supports injecting advanced, deep learning-based covert watermarks directly into the generated image pixels, deeply coupling the watermark signal with the image semantics while ensuring extremely high generation quality.

## A.3 Detailed Cross-Distribution Table

Table 1 presents the complete cross-distribution of the UniFreq-100K dataset, showing the exact number of images for each combination of image category and watermarking algorithm.

Table 1: Detailed cross-distribution of image categories and watermarking algorithms in the UniFreq-100K dataset (all values in thousands)

| **Algorithm** | **Real** | **2D/Art** | **AIGC** | **E-Post.** | **Scan.** | **Total** |
|---|---|---|---|---|---|---|
| Unwatermarked | 30 | 12 | 30 | 12 | 12 | 96 |
| LSB | 4 | 2 | 2 | 2 | 2 | 12 |
| Patchwork | 4 | 2 | 2 | 2 | 2 | 12 |
| DCT | 4 | 2 | 2 | 2 | 2 | 12 |
| DWT | 4 | 2 | 2 | 2 | 2 | 12 |
| HiDDeN | 4 | 2 | 2 | 2 | 2 | 12 |
| StegaStamp | 4 | 2 | 2 | 2 | 2 | 12 |
| Stable Sig. | 0 | 0 | 10 | 0 | 0 | 10 |
| Tree-Ring | 0 | 0 | 10 | 0 | 0 | 10 |
| SynthID | 0 | 0 | 2 | 0 | 0 | 2 |
| **Watermarked Sub.** | **24** | **12** | **34** | **12** | **12** | **94** |
| **Grand Total** | **54** | **24** | **64** | **24** | **24** | **190** |

As shown in Table 1, the dataset maintains a balanced distribution across different image categories while accommodating the specific characteristics of different watermarking algorithms. Notably, generative watermarking algorithms (Stable Signature, Tree-Ring, and SynthID) are exclusively applied to AIGC images, as these algorithms are specifically designed for integration with generative models.

## A.4 Dataset Copyright and Open-Source Statement

The UniFreq-100K dataset constructed in this study strictly complies with the open-source licenses and copyright constraints of relevant data and codes to ensure no intellectual property disputes. The specific copyright statements are as follows:

- **Open-Source Algorithm Codes and Models**: The implementations used to generate some deep learning and generative watermark samples (such as HiDDeN, StegaStamp, Stable Signature, etc.) in the dataset are directly used or based on their official open-source repositories. We strictly adhered to the respective open-source licenses of each algorithm during dataset generation, utilizing them solely for non-commercial academic research purposes.
- **Real Photos**: The real natural image carriers proportionally sampled in the dataset are sourced from the public COCO 2017 dataset. The use and

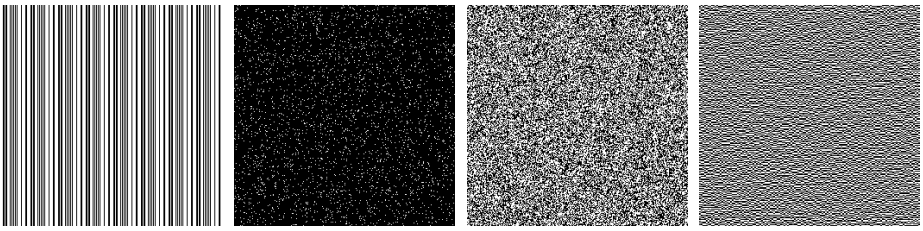
redistribution of this data fully comply with the COCO dataset's original terms of use (Creative Commons Attribution 4.0 License).

- **Paid Licensed Images**: Other types of images included in the dataset (such as 2D/digital art, electronic posters, scanned drawings, etc.) were purchased through formal channels by the research team. Additionally, the team purchased access to Google's paid image generation API. We possess full legal usage rights and distribution authorization for academic research purposes for these images, ensuring the compliance of the benchmark test data.

- **Open-Source Commitment**: Due to anonymous submission requirements and file size limitations, we cannot upload the complete dataset and open-source links at this time. The UniFreq-100K dataset will be open-sourced to the academic community following the formal acceptance of this paper. The open-sourcing of the dataset will be accompanied by a Non-Commercial Research License. Users must simultaneously comply with the copyright agreements of the aforementioned original image sources and baseline algorithms when downloading and using the data.

# B Degradation Analysis of Agnostic Invisible Watermark Detection Performance Based on Weak Spatial Characteristics

In the evaluation experiments presented in the main text, we observed a significant common phenomenon: all mainstream visual baseline models, encompassing both Convolutional Neural Networks (e.g., ResNet) and Vision Transformers (e.g., ViT), including the FSNet proposed in this paper, exhibited severe performance degradation when detecting the two traditional spatial domain watermarking algorithms: Least Significant Bit (LSB) and Patchwork. To isolate the interference of the complex semantic background noise of natural images and intuitively explore this underlying mechanism, we designed a set of rigorous, controlled visualization experiments. Specifically, we constructed a pure white background image with a resolution of $256 \times 256$ as the carrier and embedded the same 32-bit random payload using the LSB, Patchwork, DCT, and DWT algorithms, respectively. Subsequently, we calculated the absolute residual matrix between the original image and the watermarked image, extracted the maximum variation across channels, and applied extremum visual amplification (i.e., any minute perturbation is binarized to a pure white pixel). The residual visualization results of this experiment are shown in Figure 1. Based on these results, we conducted an in-depth mathematical and physical analysis from the dimensions of spatial sparsity and amplitude signal-to-noise ratio.



(a) LSB (b) Patchwork (c) DCT (d) DWT

Figure 1: Visualization of absolute residual extremum binarization triggered by four watermarking algorithms under a pure white background. From left to right: Dense periodic vertical stripes of LSB; extremely sparse impulse dot matrix of Patchwork; continuous and dense high-frequency grid perturbations of DCT and DWT.

## B.1 Patchwork: Extreme Spatial Sparsity and Signal Annihilation During Downsampling

The core embedding logic of the Patchwork algorithm relies on using a pseudo-random sequence to select a very small number of pixel pairs in the image and applying minute positive and negative shifts to their luminance values. As shown in the Patchwork residual results in Figure 1, the pixel-level modifications it introduces manifest as an extremely sparse array of isolated impulses in 2D space. Assuming the original input image is $x_c \in \mathbb{R}^{H \times W}$, the image after embedding the Patchwork watermark can be represented as $x_w = x_c + \delta_{pw}$, where the number of non-zero elements in the perturbation matrix $\delta_{pw}$ is far less than the total number of pixels in the image, i.e., $\|\delta_{pw}\|_0 \ll H \times W$.

Regardless of the architecture employed by modern visual foundation models, the core of their hierarchical representation construction relies heavily on the spatial downsampling mechanism. Let a local pooling or downsampling operation of a certain layer in the network be $\mathcal{P}(\cdot)$. Within the receptive field $\Omega$, due to the extreme spatial sparsity of $\delta_{pw}$, in the vast majority of local neighborhoods, $\delta_{pw}(i, j) = 0$. Even in neighborhoods containing perturbations, the local pooling operation will cause the sparse high-frequency impulses to be diluted by the surrounding dominant constant pixel values:

$$\mathcal{P}_\Omega(x_w) = \mathcal{P}_\Omega(x_c + \delta_{pw}) \approx \mathcal{P}_\Omega(x_c) \tag{1}$$

This extreme sparsity causes the microscopic geometric anomalies triggered by the Patchwork watermark to be irreversibly smoothed or discarded early in the network's forward propagation, rendering general binary classifiers unable to find effective decision boundaries in the deep feature space.

## B.2 LSB: Extremely Low-Amplitude Perturbations and Feature Masking at the Normalization Level

In stark contrast to the sparse distribution of Patchwork, the LSB algorithm embeds by directly replacing the least significant bit of the pixel channels. As shown on the far left of Figure 1, the modifications brought by cyclically tiling the 32-bit payload information achieve complete spatial alignment, forming extremely dense, periodic vertical stripes. However, the fundamental reason deep models falter on LSB lies not in its spatial distribution, but in its extreme low-amplitude characteristics. The maximum modification amplitude of LSB is strictly confined to an extremely tiny range (i.e., $\pm 1$), and the energy level of this perturbation signal $\delta_{lsb}$ is typically around $1/255$.

In deep neural networks, to ensure training stability, normalization operations (such as Batch Normalization) are widely deployed at every layer. Assuming the feature map input of a certain hidden layer dimension is $X = X_c + \Delta X_{lsb}$, the computation at the normalization layer can be formalized as:

$$\hat{X} = \frac{(X_c + \Delta X_{lsb}) - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{X_c - \mu}{\sqrt{\sigma^2 + \epsilon}} + \frac{\Delta X_{lsb}}{\sqrt{\sigma^2 + \epsilon}} \tag{2}$$

In natural images, the variance $\sigma^2$ brought by macroscopic semantic content is absolutely dominant, and $\sigma^2 \gg \mathrm{Var}(\Delta X_{lsb})$. This means that after normalization, the relative response value of the extremely low-amplitude perturbations introduced by LSB is severely compressed. Combined with floating-point precision truncation and non-linear activation functions, this ultra-weak signal, which falls below the noise floor of the network, rapidly undergoes "Feature Masking," naturally being treated as meaningless quantization error by the model and consequently filtered out.

## B.3 Accurate Capture of Dense Frequency Anomalies and Agnostic Detection Boundaries

To further verify the aforementioned mechanisms, we compared the controlled experimental results of the transform domain-based DCT and DWT watermarks. As shown in the two rightmost images of Figure 1, after the inverse transform back to the spatial domain, frequency domain embedding forms interwoven, dense, and continuous "high-frequency grid perturbations." This perturbation is not only absolutely dense spatially (successfully overcoming Patchwork's sparse pooling dilution issue), but due to the inverse mapping diffusion of frequency domain energy, its local microscopic structure possesses sufficient variance to penetrate the network's normalization layers (successfully overcoming LSB's low-amplitude masking issue).

The experiments and derivations above profoundly reveal the capability boundaries of current models in the AWPD task. Existing visual models (including FSNet, which performs specific high-frequency enhancement) inherently rely heavily on capturing continuous frequency anomalies with a certain energy density. LSB and Patchwork successfully bypassed the feature extraction

preferences of modern networks by exploiting extreme spatial sparsity and low amplitude. However, because these early algorithms completely lack robustness against routine image post-processing like JPEG compression, they have been eliminated in real-world copyright protection scenarios. Therefore, the models' performance on such algorithms paradoxically proves their accurate locking capability on the core common feature of modern invisible watermarks: "dense high-frequency spectral anomalies."

# C Visual Analysis of the Feature Capture Mechanism of FSNet

In the analyses of the main text and Appendix B, we theoretically demonstrated the dilemmas faced by traditional visual baseline models in the AWPD task. To intuitively dissect how FSNet breaks through this limitation and achieves outstanding zero-shot generalization capabilities under extremely weak signals, we designed two sets of in-depth controlled visualization experiments targeting the network's core innovative modules (ASPM and DMSA).

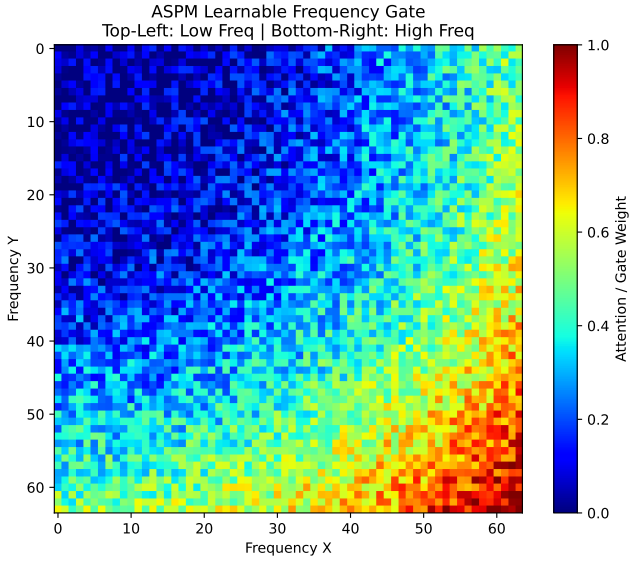



Figure 2: Learnable frequency domain gating heatmap of ASPM after training convergence. The weights in the top-left low-frequency region are significantly suppressed, while the weights in the mid-to-high-frequency regions are adaptively amplified.

## C.1 Shallow Frequency Domain Gating Analysis

The Adaptive Spectrum Perception Module (ASPM) is deployed at the shallowest layer of the network. Its core is to dynamically intercept and amplify high-frequency watermark signals through a Learnable Frequency Gate. To verify whether this module operates as expected, we extracted the model's weight parameters after training convergence on the UniFreq-100K dataset and mapped them into a pseudo-color heatmap in 2D polar coordinates.

As shown in Figure 2, this frequency domain mask exhibits a highly regular activation distribution in the 2D spectral space. In the 2D DCT spectrum, the top-left corner typically corresponds to the low-frequency direct current (DC) component of the image, representing macroscopic semantics and smooth backgrounds; whereas the bottom-right and edge regions correspond to high-frequency alternating current (AC) components, containing rich edge textures and minute perturbations.

It can be clearly observed that, driven by backpropagation optimization, the model adaptively assigned extremely low pass-through weights (appearing in cool tones) to the low-frequency region in the top-left, while allocating significantly high weights (appearing in warm tones) to the mid-to-high-frequency regions. This visualization compellingly proves that ASPM successfully acts as a data-driven "dynamic high-pass filter." Before irreversible spatial downsampling occurs in deep networks, it proactively strips away the dominant background noise of natural images, forcing the model to complete a fundamental perspective shift from "capturing macroscopically visible semantics" to "capturing microscopically invisible artifacts."

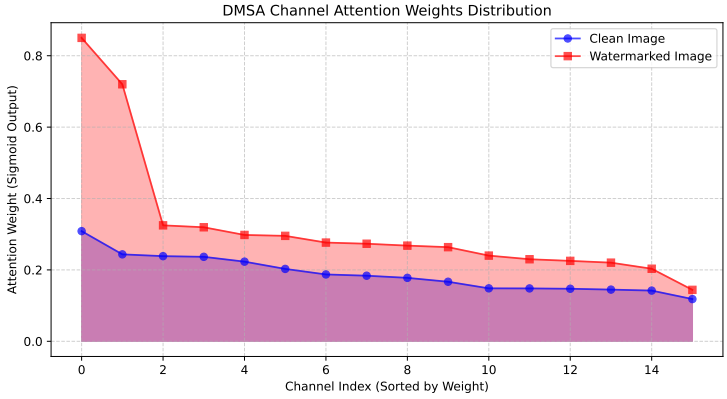



Figure 3: Comparison of channel attention weight distributions of the DMSA module under clean images and watermarked images. Watermarked images trigger significant activation peaks in specific frequency bands.

## C.2 Deep Multi-band Attention Analysis

Retaining high-frequency signals at shallow layers is not sufficient to handle the complex heterogeneity of unknown watermarking algorithms in the AWPD task. Modern invisible watermarks often concentrate energy in specific narrow-band frequencies. The primary design intention of the Dynamic Multi-band Attention mechanism (DMSA) is to recalibrate channel weights in the deep feature space using multi-branch discrete cosine transforms and extremum pooling.

To explore the channel activation preferences of DMSA, we fed Clean Images and Watermarked Images into the network separately and extracted the attention weight distributions of the 16 specific frequency branches output by the DMSA module.

As indicated by the distribution curves in Figure 3, when the input is a clean image, the attention weights across all frequency band channels remain in an overall suppressed state with a relatively flat distribution, representing the normal baseline noise floor of the natural image's deep features. However, when the input is an image containing an invisible watermark, the weight curve exhibits dramatic "impulse-like" peaks at specific frequency indices.

This significant distributional difference indicates that the DMSA module does not blindly respond to all high-frequency noise. Instead, it possesses an accurate localization capability similar to a "frequency radar." Faced with spectral energy anomalies injected by different concealment algorithms, DMSA can acutely capture local energy fluctuations (whether peaks or valleys) through three-stream extremum pooling. It dynamically assigns extremely high activation weights to the feature channels encompassing these sensitive frequency bands, thereby refining the most critical true-or-false decision boundaries for the classifier within the information-aliased deep space.