Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DASMark: Dual-Anchor Synchronization of Diffusion Image Watermarks for Geometrically Robust Deepfake Detection

Jietao Zheng
Guizhou Normal University
Yanli Chen
yanli_027@163.com
Guizhou Normal University
Hanzhou Wu
Shanghai University

Article

Keywords: Deepfake detection, diffusion models, image watermarking, geometric synchronization, user identification

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11345391/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

DASMark: Dual-Anchor Synchronization of Diffusion Image Watermarks for Geometrically Robust Deepfake Detection

Jietao Zheng[1], Yanli Chen[1*], Hanzhou Wu[2]

[1]School of Big Data and Computer Science, Guizhou Normal University, Guiyang, 550025, China.
[2]School of Communication and Information Engineering, Shanghai University, Shanghai, 200444, China.

*Corresponding author(s). E-mail(s): yanli_027@163.com;
Contributing authors: jietao_gznu@163.com; hanzhou@shu.edu.cn;

Abstract

Diffusion models can generate highly realistic images, raising concerns about deepfakes. Watermarking supports active deepfake detection through model verification and user attribution. However, geometric attacks can disrupt watermark synchronization and impair detection reliability. Existing methods remain vulnerable to such attacks, particularly in large-scale user identification. We propose DASMark, a diffusion image watermarking framework based on dual-anchor geometric synchronization. Reproducible user-specific initial noise carries identity information, while two spatially separated anchors encoded by orthogonal channel codes provide geometric references. A localization network predicts their transformed positions, enabling latent geometric alignment for robust model verification and scalable user identification. Experiments show that DASMark maintains robust synchronization across arbitrary rotation angles and achieves an average verification AUC of 0.999. With 200,000 candidate identities, it achieves 99.7% identification average accuracy and outperforms WIND by 11.0 and 18.6 percentage points under rotation and crop-and-stretch, respectively. DASMark also preserves image quality, achieves lower end-to-end runtime and GPU memory usage than WIND, and exhibits cross-model compatibility.

Keywords: Deepfake detection, diffusion models, image watermarking, geometric synchronization, user identification.

1 Introduction

Text-to-image diffusion models [1] can synthesize realistic images, expanding creative applications while also being susceptible to misuse for deceptive visual content and deepfakes. These risks motivate the development of image forensic methods. Passive detectors primarily assess whether an image is synthetic or manipulated using visual or statistical evidence, but generalization across generators remains challenging [2, 3]. Detecting synthetic content alone does not establish its model or user provenance. For services that control image generation, watermarking supports active deepfake detection by embedding identifiable signals at the source, enabling verification of the generating model and attribution to registered users [4].

Post-processing approaches [5] embed watermarks into generated images, whereas fine-tuning-based methods [4] integrate them into generation by updating generator weights. Watermarking the initial diffusion noise instead embeds provenance signals in the initial state of diffusion sampling, thereby incorporating them into generation without image post-processing or updates to pretrained generator weights. Tree-Ring [6] embeds watermark patterns in the Fourier domain of the initial latent noise and detects them after diffusion inversion, enabling watermark verification. However, its verification robustness does not directly translate into reliable discrimination among many keys. RingID [7] addresses this limitation through heterogeneous watermark patterns and discretized ring encoding, improving multi-key identification. WIND [8] further expands the identity space by matching reproducible initial noises, using a Fourier-domain group identifier to narrow the search. These advances extend diffusion watermarking from model verification to multi-key identification and large-scale matching of user-specific initial noise.

Geometric distortions can disrupt spatial correspondence between the extracted and reference watermarks, causing identification failures even when watermark information survives. Tree-Ring [6] pursues rotation robustness through Fourier ring patterns, while RingID [7] strengthens this design by refining ring construction and embedding. WIND [8] adopts similar ring patterns as group identifiers. Such ring-based embedding can introduce circular artifacts and affect generation quality. WIND additionally compensates for geometric misalignment through rotation and sliding-window searches, incurring computational overhead without fully resolving geometric sensitivity. Identification under crop-and-stretch remains challenging.

To address the detection challenges arising from watermark desynchronization under geometric attacks, we propose DASMark, a diffusion image watermarking framework based on dual-anchor synchronization for robust model verification and scalable user identification. Each user's key determines reproducible initial noise for identity encoding. Two spatially separated templates are embedded in this noise using orthogonal channel codes to form distinguishable geometric anchors. After inversion, a localization network combines multi-resolution features and anchor-specific conditioning to estimate the transformed anchor centers. By comparing the predicted anchor pair with its original configuration, we estimate rotation and crop-and-stretch parameters to spatially align the latent noise. Cosine-similarity matching is then performed between the aligned latent noise and key-specific reference noise, with a validity mask restricting comparison to valid aligned regions. The localization network trained on

one diffusion model can be directly reused with other models without additional training, allowing the synchronization mechanism to serve as a plug-and-play module while keeping pretrained generator weights unchanged.

Our main contributions are as follows:

- We propose a diffusion image watermarking framework based on dual-anchor geometric synchronization for robust model verification and scalable user identification under geometric attacks.
- We develop a localization and synchronization pipeline that derives geometric transformation parameters from anchor positions predicted by a localization network to spatially align latent noise.
- Experiments demonstrate robust verification and identification, with 99.7% average identification accuracy at 200,000 candidate identities and gains of 11.0 and 18.6 percentage points over WIND under rotation and crop-and-stretch, respectively.
- Our method preserves image quality, achieves lower end-to-end runtime and GPU memory usage than WIND, and exhibits cross-model compatibility.

2 Related Work

2.1 Deepfake Detection

Deepfake detection approaches include passive forensic analysis and active detection through watermarking. Passive methods examine observed content for synthesis or manipulation cues without requiring intervention during its creation. FaceForensics++ [9] established a benchmark for facial manipulation detection. For broader synthetic-image detection, Ojha et al. [2] exploit pretrained vision-language features to improve generalization across generators, while DIRE [3] uses diffusion reconstruction error to detect diffusion-generated images. Despite these advances, generalization to unseen generators remains challenging, and detection decisions alone do not establish model or user provenance. For services that control content creation, active watermarking embeds identifiable signals at the source, enabling detection through verification against known watermark references and supporting source attribution through associations between watermark keys and source identities [4, 6]. This provides explicit provenance evidence beyond a real-versus-fake decision.

2.2 Watermarking for Diffusion-Generated Images

Post-processing methods, such as HiDDeN [5] and StegaStamp [10], learn image-domain embedding and extraction under simulated distortions, providing general-purpose mechanisms for embedding watermarks into images after generation. These approaches face a trade-off between visual quality and watermark robustness. Generator-integrated methods instead require fine-tuning of model components: Stable Signature [4] fine-tunes the latent decoder, while AquaLoRA [11] incorporates watermark information through low-rank adaptation of the denoising network.

By contrast, initial-noise watermarking embeds signals into the initial diffusion noise, incorporating watermarks into images during generation without modifying the

pretrained generator. Watermark evidence is then extracted through diffusion inversion [12]. Tree-Ring and RingID use structured Fourier watermark patterns [6, 7]; Gaussian Shading [13] maps messages into Gaussian-distributed latents through distribution-preserving sampling; and WIND [8] treats reproducible noise as the identity carrier. However, ring-based embedding can introduce circular artifacts, and expanding its effective key capacity entails trade-offs with robustness and image quality. WIND supports larger identity spaces, but its retrieval procedure includes rotation and sliding-window searches to compensate for geometric misalignment, introducing additional detection overhead. These limitations hinder the practical deployment of watermark-based deepfake detection at scale.

2.3 Geometric Synchronization for Watermark Recovery

Geometric robustness has been pursued through invariant representations and explicit synchronization. Log-polar mapping and phase correlation provide rotation, scaling, and translation invariance [14], while template matching estimates transformations before watermark recovery [15]. Learned synchronization methods such as SyncSeal [16] embed image-domain signals and predict transformation parameters to improve existing watermark decoders.

Explicit synchronization has also been explored to improve the geometric robustness of inversion-based diffusion watermarking. SynTag [17] injects synchronization features through VAE-decoder fine-tuning and uses a prediction network for correction, requiring generator adaptation. AnchorMark [18] improves rotation robustness through a central phase anchor and training-free registration, but does not explicitly address cropping and resizing, which are also common geometric operations.

3 Problem Setting

3.1 Threat Model

We consider model-specific detection and provenance tracing for AI-generated and deepfake images, involving four participants.

• **Alice:** Alice owns the protected diffusion model G_A and operates a watermark-enabled image generation API. Each registered user is assigned a unique watermark identity, imperceptibly embedded in every image generated for that user.

• **Bob:** Bob is a legitimate user who generates and uses images in accordance with Alice's service policy.

• **David:** David is a malicious registered user who uses Alice's service to generate deceptive, harmful, or policy-violating images and disseminates them online to mislead the public.

• **Eve:** Eve steals images generated by legitimate users such as Bob and redistributes them or falsely claims ownership. She may apply image processing operations or attacks to obscure their provenance and evade watermark detection.

Alice alone holds the secret seed and user-key database. David and Eve may know the scheme's design and access generated images, but cannot access these secrets or modify server-side watermark generation and detection.

Algorithm 1 User-Specific Initial Noise Generation

Input: user identifier u_i; model-specific secret master key κ_A; user-key database $\mathcal{K}_A$
Output: user-specific initial noise $\mathbf{z}_T^{(u_i)}$; updated user-key database $\mathcal{K}_A$
Derive the user-specific key: $k_{u_i} \leftarrow \mathrm{HMAC}_{\kappa_A}(u_i)$
/ HMAC: Hash-based Message Authentication Code */*
Sample $\mathbf{z}_T^{(u_i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using k_{u_i} as the random seed
Add the user-key association (u_i, k_{u_i}) to $\mathcal{K}_A$
return $\mathbf{z}_T^{(u_i)}, \mathcal{K}_A$

3.2 Task Formulation

Under the above threat model, we formulate the watermarking system's provenance requirements as two tasks: model verification and user identification.

3.2.1 Verification

The verification function $\mathcal{V}_{G_A}$ determines whether a suspicious image $\mathbf{x}$ was generated by G_A, with output

$$\mathcal{V}_{G_A}(\mathbf{x}) \in \{\text{True, False}\}, \tag{1}$$

where True indicates generation by G_A and False indicates otherwise. This supports model attribution for deceptive or deepfake images disseminated by David and provides model-origin evidence for images falsely claimed by Eve.

3.2.2 Identification

For an image $\mathbf{x}$ generated by G_A, identification aims to determine its generating user. Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ denote the set of N registered users. The identification function $\mathcal{I}_{G_A}$ is defined as

$$\mathcal{I}_{G_A}(\mathbf{x}) = \hat{u}, \qquad \hat{u} \in \mathcal{U}, \tag{2}$$

where $\hat{u}$ denotes the identified user. This enables malicious content generated and disseminated by David to be traced to its generating user.

4 Proposed Method

As illustrated in Fig. 1, our proposed DASMark framework consists of watermarked generation and watermark detection. During generation, a user-specific key determines reproducible initial noise, into which dual-anchor localization signals are embedded before diffusion-based image generation. During detection, DDIM inversion recovers the estimated initial latent, and a dual-anchor localization network provides spatial references for geometric alignment. The synchronized latent is then matched against user-specific initial noises within valid spatial regions for model verification and user identification, supporting active deepfake detection and provenance tracing.

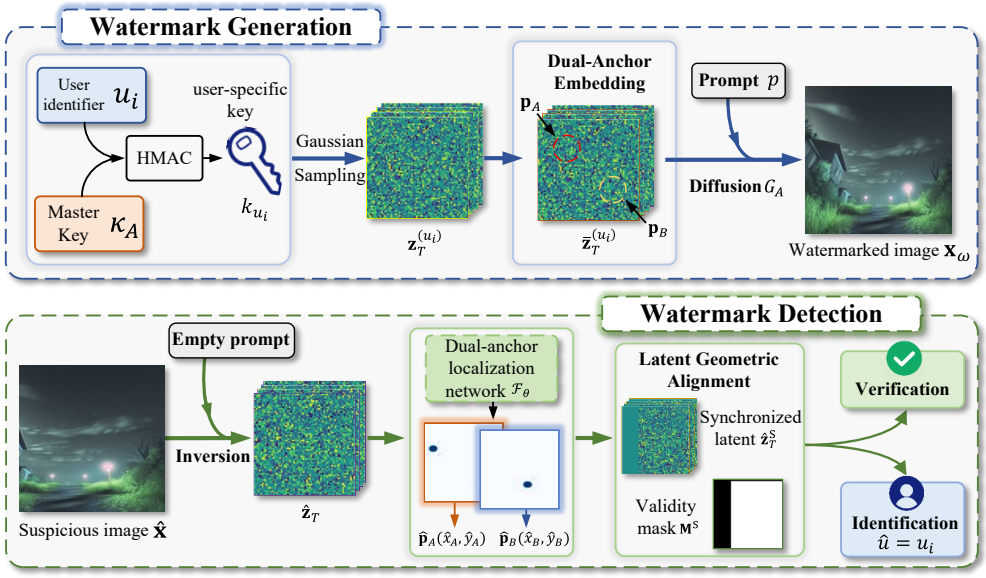


Fig. 1 Overview of the proposed DASMark framework. Watermark generation embeds user-specific identity information and dual-anchor localization signals into the initial noise for diffusion-based image generation. Watermark detection performs geometric synchronization and matching within valid latent regions for model verification and user identification.

4.1 Watermarked Generation

4.1.1 User-Specific Initial Noise Generation

Algorithm 1 generates reproducible initial noise for each registered user. A user-specific key k_{u_i}, deterministically derived from the model-specific master key κ_A and user identifier u_i, seeds Gaussian sampling to obtain $\mathbf{z}_T^{(u_i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with $\mathbf{z}_T^{(u_i)} \in \mathbb{R}^{C\times H\times W}$. Here, T denotes the diffusion sampling steps, and (C, H, W) are the latent dimensions required by G_A.

4.1.2 Dual-Anchor Localization Signal

Rotation, cropping, or scaling of generated images may misalign the inverted latent with the user-specific initial noise $\mathbf{z}_T^{(u_i)}$, impairing identity matching. To enable geometric synchronization, we embed two distinguishable localization anchors into the initial noise. During detection, their recovered positions are compared with predefined locations to estimate the geometric transformation. Each anchor uses a spatially extended pattern to provide redundancy against signal attenuation and blurring caused by diffusion inversion and geometric interpolation.

The predefined anchor centers $\mathbf{p}_A = (x_A, y_A)$ and $\mathbf{p}_B = (x_B, y_B)$ lie on the $H \times W$ latent grid. They are placed in the upper-left and lower-right regions, respectively, forming a diagonal arrangement with sufficient spatial separation while remaining away from the boundaries. For each anchor $j \in \{A, B\}$, the spatial template s_j is

constructed from a two-dimensional Gaussian with standard deviation σ over a circular support Ω_j of radius 5 latent-grid units centered at $\mathbf{p}_j$. The two supports are non-overlapping. The template values within each support Ω_j are mean-centered and normalized to unit maximum absolute amplitude.

We consider four-channel latent noise ($C = 4$) as an example. To distinguish the two anchors and distribute their localization signals across the latent channels, we assign channel codes $\mathbf{c}_A$ and $\mathbf{c}_B$ to anchors A and B, respectively:

$$\mathbf{c}_A = \frac{1}{2}\,[1, -1, 1, -1]^\mathsf{T}, \qquad \mathbf{c}_B = \frac{1}{2}\,[1, 1, -1, -1]^\mathsf{T}. \tag{3}$$

These codes form an orthonormal pair, satisfying $\|\mathbf{c}_A\|_2 = \|\mathbf{c}_B\|_2 = 1$ and $\mathbf{c}_A^\mathsf{T}\mathbf{c}_B = 0$. Their orthogonality enables separate extraction of the two anchor responses through channel-wise projection, while their equal-magnitude components distribute the localization signals across all four channels. The selected codes are not unique; other orthonormal pairs with balanced nonzero components can also be used.

For each anchor $j \in \{A, B\}$ and position $(x, y) \in \Omega_j$, $\mathbf{z}_T^{(u_i)}(:, x, y)$ denotes the channel vector of the user-specific initial noise. Its projection onto $\mathbf{c}_j$ is $\mathbf{c}_j^\mathsf{T}\mathbf{z}_T^{(u_i)}(:, x, y)$. To replace this projection with the target response $\alpha s_j(x, y)$, we add the corresponding correction along $\mathbf{c}_j$:

$$\bar{\mathbf{z}}_T^{(u_i)}(:, x, y) = \mathbf{z}_T^{(u_i)}(:, x, y) + \Big[\alpha s_j(x, y) - \mathbf{c}_j^\mathsf{T}\mathbf{z}_T^{(u_i)}(:, x, y)\Big]\,\mathbf{c}_j. \tag{4}$$

Here, α controls the anchor strength, and $\bar{\mathbf{z}}_T^{(u_i)}$ denotes the anchor-embedded initial noise. Since the channel codes have unit norm, the update sets the code-aligned projection to $\alpha s_j(x, y)$ while preserving components in its orthogonal subspace. Values outside the two anchor supports remain unchanged.

The extended templates and orthogonal channel codes provide distinguishable localization cues for geometric synchronization. The embedding is confined to two small spatial supports and, at each affected position, modifies only the projection along the corresponding channel code while preserving the orthogonal components. This restricted modification is intended to limit deviations from the original Gaussian noise statistics and reduce its impact on generated image quality.

4.1.3 Diffusion-Based Image Generation

Given a text prompt p, the protected diffusion model G_A generates a watermarked image $\mathbf{x}_w$ from the anchor-embedded initial noise $\bar{\mathbf{z}}_T^{(u_i)}$ through its standard reverse diffusion process:

$$\mathbf{x}_w = G_A\Big(\bar{\mathbf{z}}_T^{(u_i)}, p\Big). \tag{5}$$

The resulting image carries user-specific identity information and dual-anchor localization signals without requiring changes to the model parameters or the original diffusion sampling procedure.

4.2 Watermark Detection

4.2.1 Latent Inversion

Given a suspicious image $\hat{\mathbf{x}}$, we first encode it into the latent space using the VAE encoder of the protected diffusion model G_A. Deterministic DDIM inversion is then performed without text conditioning, mapping the encoded latent from timestep 0 to the terminal timestep T to obtain the estimated initial latent $\hat{\mathbf{z}}_T$. This process requires only the query image and G_A, without access to the original text prompt.

4.2.2 Dual-Anchor Localization

Geometric attacks such as rotation, cropping, and scaling disrupt the spatial correspondence between the inverted latent $\hat{\mathbf{z}}_T$ and the user-specific initial noise, impairing direct identity matching. The transformed positions of the predefined anchors provide references for geometric synchronization. However, inversion errors and geometric transformations may distort, blur, or truncate the anchor responses, hindering direct center estimation from the projected response maps.

We therefore develop a learning-based dual-anchor localization network $\mathcal{F}_\theta$ to estimate the transformed anchor centers:

$$(\hat{\mathbf{p}}_A, \hat{\mathbf{p}}_B) = \mathcal{F}_\theta(\hat{\mathbf{z}}_T)\,, \tag{6}$$

where $\hat{\mathbf{p}}_j = (\hat{x}_j, \hat{y}_j)$ denotes the predicted center of anchor $j \in \{A, B\}$ in the coordinate system of $\hat{\mathbf{z}}_T$, and θ denotes the trainable network parameters. The architecture is illustrated in Fig. 2.

Initial Anchor Response Map Extraction

The network first projects the four-channel latent vector at each spatial position of $\hat{\mathbf{z}}_T$ onto the predefined channel codes to obtain the initial anchor response maps:

$$\mathbf{R}_j(x, y) = \mathbf{c}_j^{\mathsf{T}} \hat{\mathbf{z}}_T(:, x, y), \qquad j \in \{A, B\}, \tag{7}$$

where $\mathbf{R}_A, \mathbf{R}_B \in \mathbb{R}^{H \times W}$. The projection aggregates each anchor's signal across the four latent channels into a two-dimensional spatial response, while the orthogonal codes reduce cross-anchor interference. However, the response maps still contain random latent background and inversion distortions. They therefore provide noisy anchor-specific spatial evidence that requires subsequent feature encoding and refinement for reliable localization.

Dual-Anchor Feature Encoding

The two anchors undergo the same global geometric transformation and exhibit correlated spatial variations. To exploit this correlation while preserving anchor-specific evidence, a three-branch stem extracts individual features from the two response maps and a joint feature from their concatenation. All branches follow the same encoder architecture:

$$\mathcal{E}(\mathbf{X}) = \mathrm{NAF}_5\left(\mathrm{GELU}\left(\mathrm{LN}\left(\mathrm{Conv}_{5\times5}(\mathbf{X})\right)\right)\right), \tag{8}$$

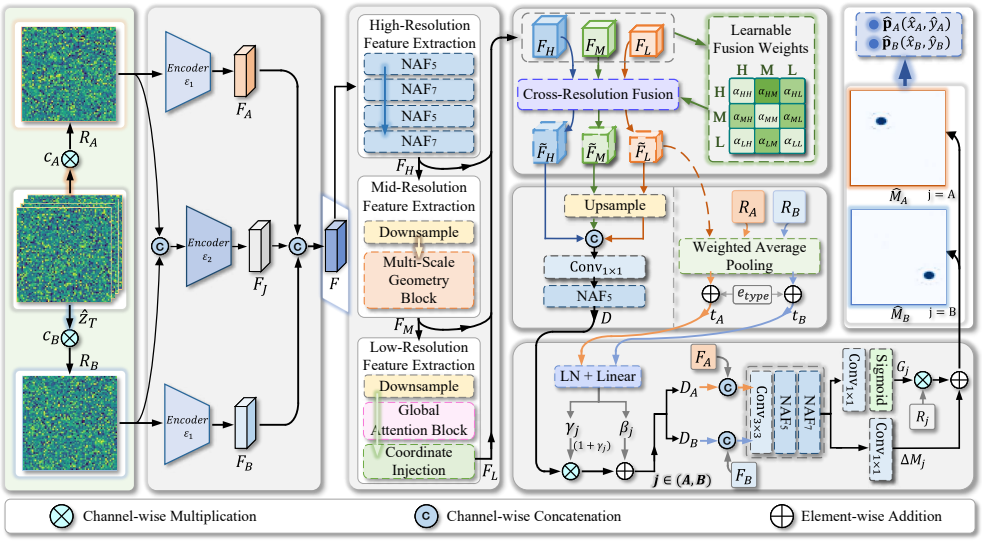


Fig. 2 Architecture of the proposed dual-anchor localization network. Multi-resolution feature extraction, adaptive cross-resolution fusion, and anchor-specific conditioning produce anchor localization maps, which are converted into center coordinates through differentiable spatial expectation.

where LN denotes layer normalization [19], GELU denotes the Gaussian error linear unit [20], and NAF_n denotes a NAFBlock [21] with an $n \times n$ depthwise convolution. The NAFBlock refines spatial and channel features through residual transformations using SimpleGate and simplified channel attention.

The two single-anchor branches share the parameters of $\mathcal{E}_1$, while the joint branch uses an independently parameterized encoder $\mathcal{E}_2$. Their outputs are

$$\mathbf{F}_A = \mathcal{E}_1(\mathbf{R}_A), \quad \mathbf{F}_B = \mathcal{E}_1(\mathbf{R}_B), \quad \mathbf{F}_J = \mathcal{E}_2\left(\mathrm{Concat}[\mathbf{R}_A, \mathbf{R}_B]\right), \tag{9}$$

where $\mathbf{F}_A, \mathbf{F}_B, \mathbf{F}_J \in \mathbb{R}^{16\times H\times W}$. The three features are concatenated along the channel dimension and fused into $\mathbf{F} \in \mathbb{R}^{32\times H\times W}$:

$$\mathbf{F} = \mathrm{NAF}_5\left(\mathrm{GELU}\left(\mathrm{LN}\left(\mathrm{Conv}_{1\times1}\left(\mathrm{Concat}[\mathbf{F}_A, \mathbf{F}_B, \mathbf{F}_J]\right)\right)\right)\right). \tag{10}$$

This fusion preserves anchor-specific evidence while integrating complementary geometric information from the two anchors, providing a shared representation for subsequent multi-resolution modeling.

Multi-Resolution Feature Extraction

The stem feature $\mathbf{F}$ is progressively encoded at three resolutions to capture fine spatial details, local geometric variations, and global spatial dependencies. The high-resolution stage applies four NAFBlocks with alternating 5×5 and 7×7 depthwise convolutions:

$$\mathbf{F}_H = \left(\mathrm{NAF}_7 \circ \mathrm{NAF}_5 \circ \mathrm{NAF}_7 \circ \mathrm{NAF}_5\right)(\mathbf{F}), \tag{11}$$

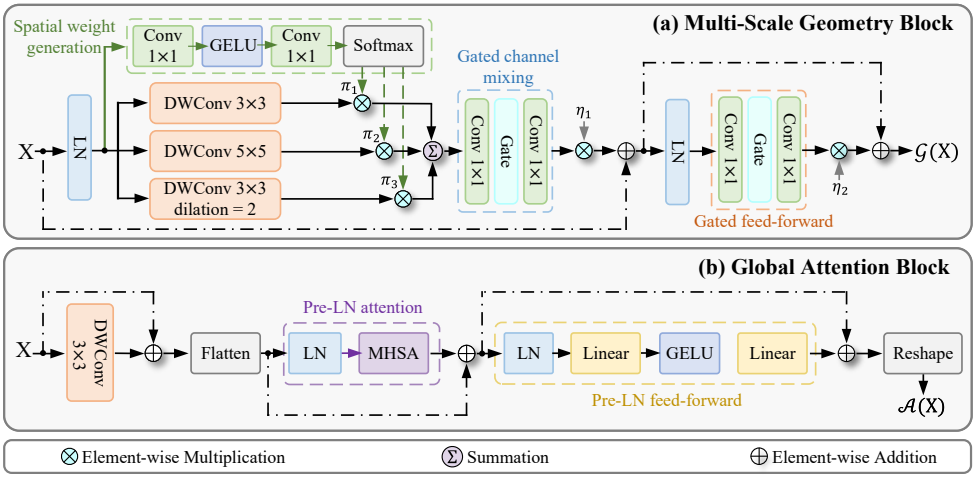


Fig. 3 Detailed architectures of (a) the Multi-Scale Geometry Block and (b) the Global Attention Block.

where $\mathbf{F}_H \in \mathbb{R}^{32\times H\times W}$ denotes the high-resolution feature that retains fine spatial structures.

To adaptively capture local geometric variations at different spatial scales, $\mathbf{F}_H$ is downsampled at the medium-resolution stage and then processed by a dynamic multi-scale geometry block:

$$\mathbf{F}_M = \mathcal{G}\left(\mathrm{GELU}\left(\mathrm{LN}\left(\mathrm{Conv}_{3\times3}^{s=2}(\mathbf{F}_H)\right)\right)\right), \tag{12}$$

where $\mathcal{G}$ denotes the Multi-Scale Geometry Block shown in Fig. 3(a), and $\mathbf{F}_M \in \mathbb{R}^{64\times\frac{H}{2}\times\frac{W}{2}}$. The block combines the outputs of parallel 3×3, 5×5, and dilated 3×3 depthwise convolutions using spatially adaptive Softmax weights. The fused feature is then refined through gated channel mixing and a gated feed-forward transformation, each with a residual connection.

To capture the global spatial relationship between the two anchors with explicit positional information, $\mathbf{F}_M$ is further downsampled at the low-resolution stage, augmented with normalized coordinates, and processed by a global attention block:

$$\mathbf{F}_L = \mathcal{A}\left(\mathrm{GELU}\left(\mathrm{LN}\left(\mathrm{Conv}_{3\times3}^{s=2}(\mathbf{F}_M)\right)\right) + 0.1\,\mathrm{Conv}_{1\times1}(\mathbf{C})\right), \tag{13}$$

where $\mathbf{F}_L \in \mathbb{R}^{128\times\frac{H}{4}\times\frac{W}{4}}$, and $\mathbf{C} \in \mathbb{R}^{2\times\frac{H}{4}\times\frac{W}{4}}$ is the normalized coordinate map:

$$\mathbf{C}(:,x,y) = \left[\frac{2x}{\frac{W}{4}-1} - 1,\ \frac{2y}{\frac{H}{4}-1} - 1\right]^{\mathsf{T}}, \tag{14}$$

for $x \in \{0,\ldots,\frac{W}{4}-1\}$ and $y \in \{0,\ldots,\frac{H}{4}-1\}$.

The Global Attention Block $\mathcal{A}$, shown in Fig. 3(b), first incorporates local spatial information by adding a 3×3 depthwise convolutional response to its input. The resulting feature map is flattened into 128-dimensional tokens, one per spatial location, and processed by multi-head self-attention [22] and a feed-forward network, both with pre-layer normalization and residual connections. The output sequence is then reshaped into a feature map encoding global spatial dependencies.

Adaptive Cross-Resolution Feature Fusion

To combine fine localization cues with broader geometric context, we align and adaptively fuse the complementary features from the three resolutions. For target resolution q and source resolution s, the normalized fusion weight is

$$\omega_{q,s} = \frac{\exp(\theta_{q,s})}{\sum_{r \in \{H,M,L\}} \exp(\theta_{q,r})}, \qquad q, s \in \{H, M, L\}, \tag{15}$$

where $\theta_{q,s}$ denotes a learnable fusion logit. The fused feature at each resolution is computed as

$$\widetilde{\mathbf{F}}_q = \sum_{s \in \{H,M,L\}} \omega_{q,s}\, \mathcal{T}_{s \to q}(\mathbf{F}_s), \tag{16}$$

where $\mathcal{T}_{s \to q}$ aligns the source feature with the target channel and spatial dimensions through channel projection and bilinear resizing, while $\mathcal{T}_{q \to q}$ is the identity mapping. We initialize $\theta_{q,q} = 4$ and $\theta_{q,s} = 0$ for $s \neq q$, initially favoring each resolution's own representation while allowing complementary contributions from other resolutions to be learned. Each fused feature $\widetilde{\mathbf{F}}_q$ has the same dimensions as $\mathbf{F}_q$.

Feature Decoding

Feature decoding uses a spatial decoding path for spatial reconstruction and an anchor-type distinction path for anchor differentiation. In the spatial path, the medium- and low-resolution features are projected to 32 channels and bilinearly resized to $H \times W$:

$$\begin{aligned} \widetilde{\mathbf{F}}_M^{\mathrm{up}} &= \mathrm{Resize}_{H \times W}^{\mathrm{bilinear}} \left(\mathrm{Conv}_{1\times1}^{64 \to 32} \left(\widetilde{\mathbf{F}}_M \right) \right), \\ \widetilde{\mathbf{F}}_L^{\mathrm{up}} &= \mathrm{Resize}_{H \times W}^{\mathrm{bilinear}} \left(\mathrm{Conv}_{1\times1}^{128 \to 32} \left(\widetilde{\mathbf{F}}_L \right) \right). \end{aligned} \tag{17}$$

The aligned features are concatenated with $\widetilde{\mathbf{F}}_H$ and fused into the decoded feature $\mathbf{D} \in \mathbb{R}^{48 \times H \times W}$:

$$\mathbf{D} = \mathrm{NAF}_5 \left(\mathrm{Conv}_{1\times1}^{96 \to 48} \left(\mathrm{Concat} \left[\widetilde{\mathbf{F}}_H, \widetilde{\mathbf{F}}_M^{\mathrm{up}}, \widetilde{\mathbf{F}}_L^{\mathrm{up}} \right] \right) \right). \tag{18}$$

Cross-resolution fusion integrates complementary information across different scales. However, the resulting shared features do not explicitly associate spatial evidence with anchor types A and B, which may weaken anchor differentiation. The anchor-type distinction path therefore combines each initial response map with the

fused low-resolution feature to construct an anchor-specific vector. Let $\mathbf{R}_j^L \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}$ be obtained by average-pooling $\mathbf{R}_j$ to match the spatial resolution of $\widetilde{\mathbf{F}}_L$. Normalized positive-response pooling yields the initial anchor-specific vector $\mathbf{t}_j^0 \in \mathbb{R}^{128}$:

$$\mathbf{t}_j^0 = \frac{\sum_{x,y} \mathrm{ReLU}\big(\mathbf{R}_j^L(x,y)\big)\, \widetilde{\mathbf{F}}_L(:,x,y)}{\sum_{x,y} \mathrm{ReLU}\big(\mathbf{R}_j^L(x,y)\big) + \epsilon}, \qquad j \in \{A, B\}. \tag{19}$$

The positive responses weight the aggregation of low-resolution features toward locations likely associated with anchor j, while $\epsilon > 0$ prevents division by zero.

To further distinguish anchors A and B, we add separate learnable anchor-type embeddings $\mathbf{e}_A, \mathbf{e}_B \in \mathbb{R}^{128}$:

$$\mathbf{t}_j = \mathbf{t}_j^0 + \mathbf{e}_j, \qquad j \in \{A, B\}. \tag{20}$$

The resulting anchor-type-aware vectors $\mathbf{t}_A, \mathbf{t}_B \in \mathbb{R}^{128}$ provide anchor-specific conditioning for anchor localization map prediction.

Anchor Localization Map Prediction

To derive anchor-specific representations from the shared decoded feature $\mathbf{D}$, the anchor-type-aware vector $\mathbf{t}_j$ generates two channel-wise modulation vectors for each anchor $j \in \{A, B\}$:

$$\begin{aligned} (\boldsymbol{\gamma}_j, \boldsymbol{\beta}_j) &= \mathrm{Linear}^{128 \to 96}\left(\mathrm{LN}(\mathbf{t}_j)\right), \\ \mathbf{D}_j &= \mathbf{D} \odot (1 + \boldsymbol{\gamma}_j) + \boldsymbol{\beta}_j. \end{aligned} \tag{21}$$

The linear output is split into two 48-dimensional vectors, $\boldsymbol{\gamma}_j$ and $\boldsymbol{\beta}_j$, which are broadcast across the spatial dimensions of $\mathbf{D}$ to control channel-wise scaling and shifting, respectively. Here, $\odot$ denotes element-wise multiplication. The resulting feature $\mathbf{D}_j \in \mathbb{R}^{48 \times H \times W}$ incorporates the type information of anchor j while preserving the spatial resolution.

Although $\mathbf{D}_j$ provides anchor-conditioned multi-resolution context, downsampling and fusion may attenuate fine spatial details needed for precise localization. We therefore fuse it with the corresponding shallow feature $\mathbf{F}_j$, which retains high-resolution local information from the initial response map:

$$\mathbf{U}_j = \Phi\left(\mathrm{Concat}\left[\mathbf{D}_j,\, \mathrm{Conv}_{1\times1}^{16\to32}\left(\mathbf{F}_j\right)\right]\right), \tag{22}$$

where Φ is a convolutional fusion block defined as

$$\Phi(\mathbf{X}) = \mathrm{NAF}_7\left(\mathrm{NAF}_5\left(\mathrm{Conv}_{3\times3}(\mathbf{X})\right)\right). \tag{23}$$

To retain anchor-specific evidence from the initial response while correcting incomplete or distorted responses, two independently parameterized 1×1 convolutional

branches predict a spatial retention map and a residual correction map from $\mathbf{U}_j$:

$$\mathbf{G}_j = \mathrm{Sigmoid}\left(\mathrm{Conv}_{1\times1}(\mathbf{U}_j)\right), \qquad \Delta\mathbf{M}_j = \mathrm{Conv}_{1\times1}(\mathbf{U}_j), \tag{24}$$

where $\mathbf{G}_j \in [0,1]^{H\times W}$ controls how much of $\mathbf{R}_j$ is retained at each position, and $\Delta\mathbf{M}_j \in \mathbb{R}^{H\times W}$ provides an additive correction. Their combination yields the anchor localization map:

$$\widehat{\mathbf{M}}_j = \mathbf{G}_j \odot \mathbf{R}_j + \Delta\mathbf{M}_j. \tag{25}$$

The resulting localization maps $\widehat{\mathbf{M}}_A, \widehat{\mathbf{M}}_B \in \mathbb{R}^{H\times W}$ are designed to retain the corresponding anchor patterns while suppressing spurious responses from the random-noise background.

Differentiable Anchor Coordinate Estimation

The predicted localization maps encode spatial anchor evidence, whereas geometric alignment requires explicit center coordinates. A learnable coordinate regressor introduces additional parameters to optimize, while non-differentiable peak selection with argmax blocks gradient propagation from coordinate-level supervision to the localization maps. These limitations can complicate end-to-end optimization. Following the spatial-expectation principle of DSNT [23], we adopt a parameter-free differentiable estimator that converts each map into a normalized positive spatial distribution and estimates the anchor center through spatial expectation.

For each anchor $j \in \{A, B\}$, positive localization evidence is extracted as

$$\mathbf{E}_j(x,y) = \mathrm{ReLU}\Big(\widehat{\mathbf{M}}_j(x,y)\Big). \tag{26}$$

The evidence is then normalized by its maximum response:

$$\overline{\mathbf{E}}_j(x,y) = \frac{\mathbf{E}_j(x,y)}{\max\limits_{u,v}\mathbf{E}_j(u,v) + \epsilon}, \tag{27}$$

where $\epsilon > 0$ ensures numerical stability. This normalization bounds the responses while preserving their relative spatial pattern. The normalized evidence is converted into a spatial weight distribution:

$$\rho_j(x,y) = \frac{\overline{\mathbf{E}}_j(x,y)^\tau + \epsilon}{\sum\limits_{u,v}\left(\overline{\mathbf{E}}_j(u,v)^\tau + \epsilon\right)}, \tag{28}$$

where the fixed exponent $\tau > 0$ adjusts the relative weighting of strong and weak responses, and $\rho_j(x,y)$ denotes the normalized spatial weight at (x,y). The anchor center is computed through spatial expectation:

$$\hat{\mathbf{p}}_j = \sum_{x=0}^{W-1}\sum_{y=0}^{H-1}\rho_j(x,y)\begin{bmatrix}x\\y\end{bmatrix}, \qquad j \in \{A,B\}. \tag{29}$$

The differentiable spatial expectation allows coordinate-level errors to propagate through the spatial weights and refine the predicted localization maps.

Training and Optimization

During training, the diffusion model generates a watermarked image $\mathbf{x}$ from initial noise containing both anchor patterns. The same geometric transformation, such as rotation or crop-and-stretch, is applied to $\mathbf{x}$, anchor templates $\mathbf{M}_j$, and centers $\mathbf{p}_j$, yielding the transformed image $\mathbf{x}^*$, target localization maps $\mathbf{M}_j^*$, and target coordinates $\mathbf{p}_j^* = (x_j^*, y_j^*)$, respectively, for $j \in \{A, B\}$. DDIM inversion of $\mathbf{x}^*$ produces the estimated initial latent $\hat{\mathbf{z}}_T$, which is fed into $\mathcal{F}_\theta$ to predict localization maps $\widehat{\mathbf{M}}_j$ and estimate anchor centers $\hat{\mathbf{p}}_j$.

Rather than directly regressing coordinates from $\hat{\mathbf{z}}_T$, we supervise the localization maps to facilitate coordinate estimation. A coordinate loss further constrains the maps through differentiable spatial expectation, encouraging cleaner anchor responses and more accurate centers. Since anchor patterns occupy limited spatial regions, we supervise the maps using a spatially weighted version of the Smooth L1 loss [24]. The spatial weights $\bar{w}_j(x, y)$ are derived from the absolute responses of the target localization map $\mathbf{M}_j^*$ and normalized over the spatial grid such that $\sum_{x,y} \bar{w}_j(x, y) = 1$, assigning larger weights to positions with stronger target response magnitudes. The localization map loss is

$$\mathcal{L}_{\text{map}} = \frac{1}{2} \sum_{j \in \{A,B\}} \sum_{x,y} \bar{w}_j(x, y)\, \ell_{\text{SL1}} \left(\widehat{\mathbf{M}}_j(x, y) - \mathbf{M}_j^*(x, y) \right), \tag{30}$$

where ℓ_{SL1} denotes the element-wise Smooth L1 loss. The coordinate loss averages the absolute errors over the four coordinate components:

$$\mathcal{L}_{\text{coord}} = \frac{1}{4} \sum_{j \in \{A,B\}} \left\| \hat{\mathbf{p}}_j - \mathbf{p}_j^* \right\|_1 . \tag{31}$$

The overall training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{map}} + \lambda \mathcal{L}_{\text{coord}}, \tag{32}$$

where $\lambda > 0$ weights coordinate supervision relative to map supervision. We optimize the parameters θ of $\mathcal{F}_\theta$ by minimizing $\mathcal{L}_{\text{total}}$.

4.2.3 Latent Geometric Alignment

The predicted anchor centers are used to estimate geometric transformation parameters and spatially align $\hat{\mathbf{z}}_T$ with the initial noise. A corresponding validity mask is generated to exclude missing or invalid regions from subsequent watermark detection.

Geometric Attack-Type Determination

We compute the maximum displacement between the predefined anchor centers $\mathbf{p}_j$ and predicted centers $\hat{\mathbf{p}}_j$ as

$$\Delta_{\text{pos}} = \max_{j\in\{A,B\}} \|\hat{\mathbf{p}}_j - \mathbf{p}_j\|_2 . \tag{33}$$

The reference and predicted inter-anchor distances are

$$d_{AB} = \|\mathbf{p}_B - \mathbf{p}_A\|_2 , \qquad \hat{d}_{AB} = \|\hat{\mathbf{p}}_B - \hat{\mathbf{p}}_A\|_2 . \tag{34}$$

Their relative change is

$$\Delta_{\text{dist}} = \frac{\left|\hat{d}_{AB} - d_{AB}\right|}{d_{AB}}. \tag{35}$$

With $\varepsilon_{\text{dist}}$ as the tolerance for relative distance changes, the geometric attack type is estimated as

$$\xi_{\text{geo}} = \begin{cases} 0, & \Delta_{\text{pos}} \le 1, \\ 1, & \Delta_{\text{pos}} > 1, \quad \Delta_{\text{dist}} \le \varepsilon_{\text{dist}}, \\ 2, & \Delta_{\text{pos}} > 1, \quad \Delta_{\text{dist}} > \varepsilon_{\text{dist}}, \end{cases} \tag{36}$$

where the labels 0, 1, and 2 denote no geometric attack, rotation, and crop-and-stretch, respectively. Without a geometric attack, the predicted centers are expected to remain within one latent-grid unit of their predefined positions. Rotation changes their positions while approximately preserving their distance, whereas crop-and-stretch typically changes both.

Rotation Alignment

For a rotation attack, we estimate the rotation angle from the directional change of the anchor pair and apply inverse rotation to restore the reference orientation of the inverted latent. The reference and predicted anchor-pair vectors are defined as

$$\mathbf{v}_{AB} = (v_x, v_y)^{\mathsf{T}} = \mathbf{p}_B - \mathbf{p}_A, \qquad \hat{\mathbf{v}}_{AB} = (\hat{v}_x, \hat{v}_y)^{\mathsf{T}} = \hat{\mathbf{p}}_B - \hat{\mathbf{p}}_A. \tag{37}$$

With the x-axis pointing right and the y-axis downward, the clockwise-positive angle $\theta_{\text{rot}} \in (-\pi, \pi]$ from $\mathbf{v}_{AB}$ to $\hat{\mathbf{v}}_{AB}$ is computed as

$$\theta_{\text{rot}} = \operatorname{atan2}\left(v_x\hat{v}_y - v_y\hat{v}_x,\, \mathbf{v}_{AB}^{\mathsf{T}}\hat{\mathbf{v}}_{AB}\right). \tag{38}$$

Here, atan2 denotes the two-argument arctangent, which uses the scalar cross product and inner product to determine the rotation magnitude and direction. Let $\text{Rotate}(\mathbf{X}, \alpha)$ denote rotation of $\mathbf{X}$ by angle α about the latent-grid center under the same sign convention. The rotation-aligned latent is obtained as

$$\hat{\mathbf{z}}_T^{\text{rot}} = \text{Rotate}\left(\hat{\mathbf{z}}_T, -\theta_{\text{rot}}\right). \tag{39}$$

To exclude invalid boundary regions introduced by rotation, we apply the same inverse rotation to an all-one binary mask $\mathbf{M}_0 \in \{0,1\}^{H\times W}$:

$$\mathbf{M}_{\text{rot}} = \text{Rotate}\left(\mathbf{M}_0, -\theta_{\text{rot}}\right). \tag{40}$$

The entries of $\mathbf{M}_{\text{rot}}$ are 1 in valid regions of $\hat{\mathbf{z}}_T^{\text{rot}}$ and 0 in invalid boundary regions.

Crop-and-Stretch Alignment

For a crop-and-stretch attack, we estimate the surviving crop window from changes in the anchor-pair geometry and map the inverted latent back to the reference coordinate system. Using the reference components v_x, v_y and their predicted counterparts $\hat{v}_x, \hat{v}_y$ defined in Eq. (37), we estimate the horizontal and vertical scale factors as

$$s_x = \frac{\hat{v}_x}{v_x}, \qquad s_y = \frac{\hat{v}_y}{v_y}. \tag{41}$$

The crop-window width and height in the reference coordinate system are then estimated as

$$w_c = \frac{W}{s_x}, \qquad h_c = \frac{H}{s_y}. \tag{42}$$

Let $\mathbf{m} = (m_x, m_y)$ and $\hat{\mathbf{m}} = (\hat{m}_x, \hat{m}_y)$ denote the reference and predicted anchor-pair midpoints, respectively. The upper-left coordinate of the crop window is estimated as

$$x_c = m_x - \frac{\hat{m}_x}{s_x}, \qquad y_c = m_y - \frac{\hat{m}_y}{s_y}. \tag{43}$$

We resize $\hat{\mathbf{z}}_T$ to width w_c and height h_c and place it at (x_c, y_c) on the reference grid, yielding the crop-corrected latent $\hat{\mathbf{z}}_T^{\text{crop}}$. The corresponding binary validity mask $\mathbf{M}_{\text{crop}} \in \{0,1\}^{H\times W}$ is set to 1 within the estimated crop window and 0 elsewhere, excluding regions removed by cropping, whose original spatial information cannot be recovered by geometric alignment.

Type-Conditioned Geometric Synchronization

Based on the estimated attack type ξ_{geo}, we select the synchronized latent $\hat{\mathbf{z}}_T^{\text{S}}$ and its validity mask $\mathbf{M}^{\text{S}}$ as

$$\left(\hat{\mathbf{z}}_T^{\text{S}}, \mathbf{M}^{\text{S}}\right) = \begin{cases} \left(\hat{\mathbf{z}}_T, \mathbf{1}_{H\times W}\right), & \xi_{\text{geo}} = 0, \\ \left(\hat{\mathbf{z}}_T^{\text{rot}}, \mathbf{M}_{\text{rot}}\right), & \xi_{\text{geo}} = 1, \\ \left(\hat{\mathbf{z}}_T^{\text{crop}}, \mathbf{M}_{\text{crop}}\right), & \xi_{\text{geo}} = 2. \end{cases} \tag{44}$$

Here, $\mathbf{1}_{H\times W}$ denotes an all-one validity mask.

4.2.4 Verification and Identification

The geometrically synchronized latent $\hat{\mathbf{z}}_T^{\mathrm{S}}$ is matched against the user-specific initial noises associated with the protected diffusion model. Using each user-specific key k_{u_i} stored in $\mathcal{K}_A$ as the random seed, Alice deterministically reproduces the corresponding initial noise $\mathbf{z}_T^{(u_i)}$. The validity mask $\mathbf{M}^{\mathrm{S}}$ restricts the matching to spatial regions containing valid information.

Verification

Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ denote the set of registered users associated with G_A, where N is the total number of registered users. For each $u_i \in \mathcal{U}$, the masked cosine similarity ρ_i^{sim} between $\hat{\mathbf{z}}_T^{\mathrm{S}}$ and $\mathbf{z}_T^{(u_i)}$ is computed as

$$\rho_i^{\mathrm{sim}} = \frac{\left\langle \mathbf{M}^{\mathrm{S}} \odot \hat{\mathbf{z}}_T^{\mathrm{S}}, \mathbf{M}^{\mathrm{S}} \odot \mathbf{z}_T^{(u_i)} \right\rangle_F}{\left\| \mathbf{M}^{\mathrm{S}} \odot \hat{\mathbf{z}}_T^{\mathrm{S}} \right\|_F \left\| \mathbf{M}^{\mathrm{S}} \odot \mathbf{z}_T^{(u_i)} \right\|_F}, \qquad i = 1, \ldots, N, \tag{45}$$

where $\odot$ denotes element-wise multiplication with $\mathbf{M}^{\mathrm{S}}$ broadcast across the latent channels, and $\langle \cdot, \cdot \rangle_F$ and $\| \cdot \|_F$ denote the Frobenius inner product and norm over all channel and spatial dimensions, respectively. The maximum similarity $\rho_{\max}^{\mathrm{sim}}$ over all user-specific initial noises is computed as

$$\rho_{\max}^{\mathrm{sim}} = \max_{1 \le i \le N} \rho_i^{\mathrm{sim}}. \tag{46}$$

Let τ_{ver} denote the verification threshold. The query image is verified as originating from G_A if $\rho_{\max}^{\mathrm{sim}} \ge \tau_{\mathrm{ver}}$ and rejected otherwise.

Identification

For a query image verified as originating from G_A, the best-matching user index $i^\star$ and the corresponding generating user $\hat{u}$ are obtained as

$$i^\star = \arg\max_{1 \le i \le N} \rho_i^{\mathrm{sim}}, \qquad \hat{u} = u_{i^\star}. \tag{47}$$

Since $\mathbf{z}_T^{(u_i)}$ is deterministically reproduced from k_{u_i} and the association (u_i, k_{u_i}) is maintained in $\mathcal{K}_A$, the index $i^\star$ directly maps the query image to its registered generating user, thereby enabling reliable tracing of the generating user.

Code availability

The source code for DASMark, including scripts for generating the training dataset, will be made publicly available in a public repository upon acceptance of the manuscript. During peer review, the code is available to editors and reviewers upon request.

5 Experimental Results

5.1 Experimental Setup

Following Tree-Ring [6], RingID [7], and WIND [8], we use Stable Diffusion 2.1 [1] as the default diffusion model to generate 512×512 images. The corresponding initial latent is $\mathbf{z}_T \in \mathbb{R}^{4\times64\times64}$. Generation uses DPMSolver [26] with $T = 50$ sampling steps and a guidance scale of 7.5. Detection uses 50-step deterministic DDIM inversion [12] with an empty text condition. All experimental images are generated using text prompts from the `test` split of the Stable-Diffusion-Prompts dataset [25], available on Hugging Face. Prompts at indices 0–999 are used for evaluation, while those at indices 3000–7999 are used to generate training images for the anchor localization network.

Using the upper-left corner of the latent grid as the coordinate origin, we place the two anchors at $\mathbf{p}_A = (20, 20)$ and $\mathbf{p}_B = (44, 44)$, with a Gaussian standard deviation of $\sigma = 2.5$ and an embedding strength of $\alpha = 2.0$. We set the response-weighting exponent to $\tau = 5$ and the coordinate loss weight to $\lambda = 1$. For geometric alignment, the relative inter-anchor distance tolerance is set to $\varepsilon_{\text{dist}} = 0.08$. The dual-anchor localization network is trained on 20,800 samples, comprising 10,000 random crop-and-stretch samples with a crop ratio of 0.75 and 10,800 rotation samples, with 30 samples generated per rotation angle.

For robustness evaluation, we consider six representative image distortions: rotation by 75°, JPEG compression with a quality factor of 25, crop-and-stretch (C&S) with a crop scale of 0.75, Gaussian blur with a radius of 4, Gaussian noise with a standard deviation of 0.1, and brightness adjustment with the brightness factor uniformly sampled from $[0, 6]$. The same attack settings are applied to all compared methods.

Model verification is evaluated on 1,000 pairs of watermarked and non-watermarked images, with performance measured by the area under the ROC curve (AUC) and the true positive rate at a false positive rate of 1% (TPR@1%FPR). For user identification, we randomly select a key from the evaluated key space for each of 1,000 samples and embed the corresponding user identity. Identification accuracy (Acc.) is reported across different key-space sizes, defined as the proportion of samples correctly attributed to their generating users.

Image generation quality is assessed using Fréchet Inception Distance (FID) [27] and CLIP score [28]. We compute FID between 5,000 generated images and reference images from the COCO 2017 [29] to measure distributional similarity. Alignment with the input prompts is measured using CLIP scores on 1,000 generated image–prompt pairs with the OpenCLIP ViT-g/14 model [30]. Both metrics are averaged over five independent runs with different random seeds.

5.2 Robustness Evaluation

5.2.1 Verification Performance

The model verification results for our proposed DASMark, Tree-Ring [6], and RingID [7] are reported in Table 1. In this verification setting, both baselines use a single fixed watermark key ($N = 1$), making decisions based on the distance between

Table 1 Model verification performance under clean and attacked conditions.

Methods	Metric	Clean	Rotate	JPEG	C&S	Blur	Noise	Bright.	Avg
Tree-Ring [6]	AUC	1.000	0.935	0.999	0.961	0.999	0.944	0.983	0.974
	T@1%F	1.000	0.516	0.998	0.648	0.999	0.724	0.904	0.827
RingID [7]	AUC	1.000	1.000	1.000	0.987	0.989	0.998	0.994	0.995
	T@1%F	1.000	0.998	1.000	0.596	0.995	0.947	0.980	0.931
Ours	AUC	1.000	0.996	1.000	1.000	1.000	1.000	1.000	**0.999**
	T@1%F	1.000	0.993	1.000	0.999	1.000	1.000	0.997	**0.998**

the extracted watermark and the reference watermark. Repeated use of a fixed pattern may facilitate steganalytic detection [31]. We therefore randomly select a key from a fixed key space of size $N = 1{,}000$ for each image. This introduces watermark diversity while keeping the verification search space modest. Despite the larger verification search space, our method achieves the highest average AUC (0.999) and TPR@1%FPR (0.998), with particularly strong robustness to C&S. For our method, the mean similarity scores for watermarked images range from 0.244 under rotation to 0.886 without attacks, all substantially higher than the mean of 0.036 for unwatermarked images. In practical application tests, a fixed verification threshold of $\tau_{\text{ver}} = 0.04665$ yields an average TPR of 0.9881 across the evaluated conditions.

5.2.2 Identification Performance

The user identification results across different key-space sizes are reported in Table 2. The structured ring-based encoding of Tree-Ring and RingID limits their effective key capacity, while geometric distortions, particularly C&S, disrupt the correspondence between recovered and reference Fourier patterns [7]. WIND [8] and our method accommodate larger key spaces using key-dependent initial noise. However, WIND remains sensitive to geometric attacks: its C&S accuracy drops from 0.930 to 0.810 as N increases from 100,000 to 200,000. Our method maintains an average accuracy of 0.997 at both scales and achieves accuracies of 0.990 under rotation and 0.996 under C&S at $N = 200{,}000$. These results demonstrate robust user identification with minimal performance degradation as the key space expands.

5.2.3 Robustness under Different Attack Strengths

The user identification results under varying attack strengths are presented in Fig. 4. Our method maintains consistently high accuracy across the tested rotation angles and exhibits smaller performance losses under strong blur, noise, and brightness adjustments than WIND, which fluctuates substantially under rotation and degrades more severely under these distortions. Both methods achieve nearly perfect accuracy across the tested JPEG quality range. Although aggressive C&S degrades both methods, ours retains a clear accuracy advantage even at crop scales where WIND approaches zero accuracy.

Table 2 User identification accuracy under different key-space sizes and image distortions, where Keys = N denotes the number of candidate identities.

Methods	#Keys	Clean	Rotate	JPEG	C&S	Blur	Noise	Bright.	Avg
Tree-Ring [6]	32	0.790	0.020	0.420	0.040	0.610	0.530	0.420	0.404
	128	0.450	0.010	0.120	0.020	0.280	0.230	0.170	0.183
	2048	0.200	0.000	0.040	0.000	0.090	0.070	0.060	0.066
RingID [7]	32	1.000	1.000	1.000	0.530	0.990	1.000	0.960	0.926
	128	1.000	0.980	1.000	0.280	0.980	1.000	0.940	0.883
	2048	1.000	0.860	1.000	0.080	0.970	0.950	0.870	0.819
WIND [8]	100,000	1.000	0.880	1.000	0.930	1.000	0.990	0.980	0.969
	200,000	1.000	0.880	1.000	0.810	1.000	0.990	0.910	0.941
Ours	32	1.000	0.993	1.000	0.999	1.000	1.000	0.996	**0.998**
	128	1.000	0.991	1.000	0.998	1.000	1.000	0.996	**0.998**
	2048	1.000	0.991	1.000	0.998	1.000	1.000	0.996	**0.998**
	100,000	1.000	0.991	1.000	0.996	1.000	1.000	0.995	**0.997**
	200,000	1.000	0.990	1.000	0.996	1.000	1.000	0.995	**0.997**

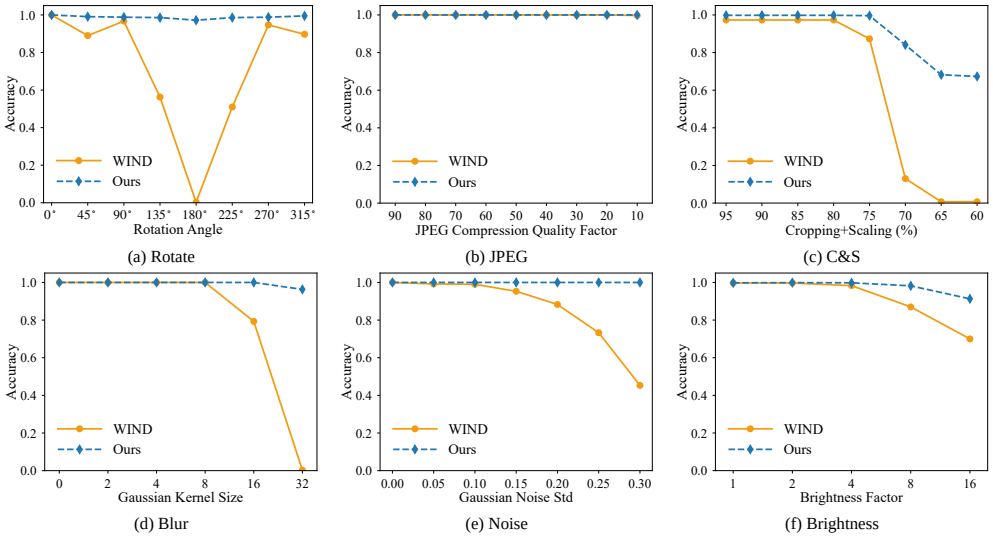


Fig. 4 User identification accuracy under varying attack strengths with 100,000 keys.

5.2.4 Robustness under Hybrid Attacks

We evaluate our method's robustness in user identification under hybrid attacks, with the results reported in Table 3. WIND suffers substantial accuracy losses under combinations involving geometric distortions, particularly when coupled with JPEG compression or noise. Our method achieves considerably higher accuracy under these combinations and maintains high accuracy under non-geometric hybrid

Table 3 User identification accuracy under hybrid attacks with 100,000 keys.

Methods	Rotate				C&S				Bright.		Noise		JPEG	
	JPEG	Blur	Noise	Bright.	JPEG	Blur	Noise	Bright.	Blur	JPEG	Noise	Blur	JPEG	Blur
WIND [8]	0.263	0.891	0.007	0.390	0.430	0.493	0.043	0.430	0.710	0.933	0.897	0.917	0.977	0.996
Ours	**0.910**	**0.962**	**0.772**	**0.400**	**0.836**	**0.870**	**0.765**	**0.459**	**0.953**	**0.990**	**0.973**	**1.000**	**1.000**	**1.000**

Table 4 Latent cosine similarity to the correct key's initial noise and user identification accuracy before and after geometric alignment with 100,000 candidate keys.

Attack	Latent Similarity		Identification Accuracy (%)	
	Before	After	Before	After
Rotate	0.000350	**0.274898**	0.00	**99.10**
C&S	0.044289	**0.650596**	79.10	**99.60**

attacks, demonstrating greater robustness to combined distortions. However, geometric distortions combined with brightness adjustments remain challenging for both methods.

5.3 Anchor Localization and Geometric Synchronization

We evaluate dual-anchor localization on a test set comprising 900 C&S samples with a crop ratio of 0.75 and 720 rotation samples spanning a full 360° range. On the 64×64 latent grid, the mean Euclidean localization error between predicted and ground-truth anchor coordinates is 0.315 pixels, with an RMSE of 1.734 pixels, computed over both anchors and all test samples. The percentage of correct keypoints (PCK) reaches 98.67%, 99.10%, and 99.35% at Euclidean distance thresholds of 1, 2, and 4 pixels, respectively. These results support accurate dual-anchor localization, providing spatial references for subsequent geometric synchronization.

We further assess the effect of geometric alignment on latent matching and user identification, with the results reported in Table 4. Alignment substantially increases the similarity between the recovered latent and the initial noise generated from the correct key, raising identification accuracy from 0.00% to 99.10% under rotation and from 79.10% to 99.60% under C&S. The examples in Fig. 5 illustrate accurate anchor localization and alignment of the recovered latent to the reference geometry, with invalid regions excluded from matching. These results demonstrate the effectiveness of dual-anchor synchronization in restoring spatial correspondence and improving identification robustness under geometric distortions.

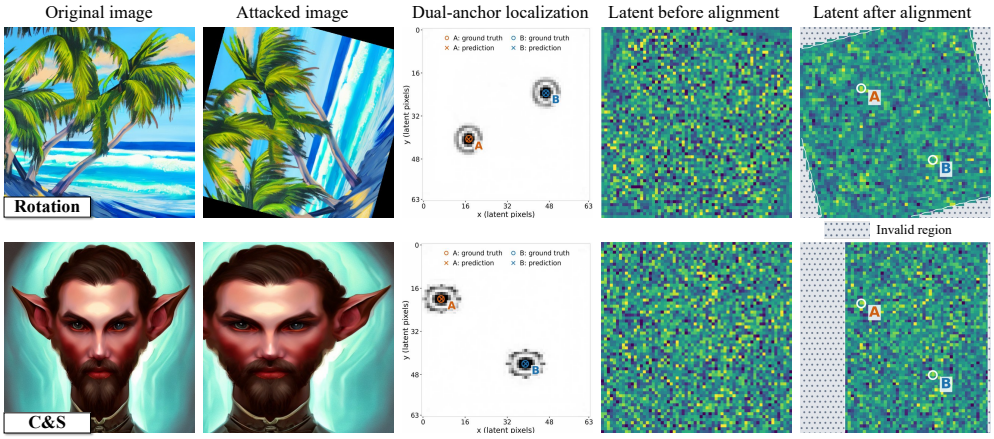


Fig. 5 Visualization of dual-anchor localization and latent geometric alignment under rotation and C&S. Regions filled with a dot pattern indicate invalid areas excluded from similarity computation.

Table 5 Ablation of dual-anchor localization network components. Identification robustness under geometric attacks is evaluated with 100,000 candidate keys.

Settings	Mean Error (px) ↓	PCK@2px (%) ↑	Rotation ACC (%) ↑	C&S ACC (%) ↑
Full model	**0.315**	**99.105**	**99.10**	**99.60**
w/o Multi-Scale Geometry	0.942	95.802	81.40	98.80
w/o Global Attention	0.836	96.111	88.30	98.00
w/o Cross-Resolution Fusion	0.852	95.926	84.20	98.90
w/o Anchor-Specific Conditioning	0.947	95.710	82.40	98.70
w/o Coordinate Supervision	0.377	98.765	97.00	99.00

5.4 Ablation Study

We assess the contributions of the dual-anchor localization network components and coordinate supervision, with the results reported in Table 5. Multi-scale geometry integrates cues across receptive fields, while anchor-specific conditioning preserves distinct anchor representations as rotation alters their spatial configuration. Removing these mechanisms reduces rotation accuracy by 17.70 and 16.70 percentage points, respectively, supporting their roles in localizing transformed anchors within noisy response maps. Global attention and cross-resolution fusion further improve localization by integrating global spatial context with fine spatial detail. Coordinate supervision complements map supervision, reducing the mean coordinate error from 0.377 to 0.315 pixels and raising rotation accuracy from 97.00% to 99.10%. These results support the effectiveness of the dual-anchor localization network design in combining geometric feature modeling and coordinate optimization for accurate localization and robust user identification.

Table 6 User identification accuracy across diffusion models with 100,000 keys.

Models	Clean	Rotate	JPEG	C&S	Blur	Noise	Bright.	Avg
SD 2.1	1.000	0.991	1.000	0.996	1.000	1.000	0.995	0.997
SD 2.0	1.000	0.987	1.000	0.996	1.000	1.000	1.000	0.998
SD 1.5	1.000	0.991	1.000	0.996	1.000	1.000	0.998	0.998
SD 1.4	1.000	0.991	1.000	0.990	1.000	1.000	0.997	0.997

Table 7 Image quality comparison across different watermarking methods.

Metrics	No watermark	Tree-Ring [6]	RingID [7]	WIND [8]	Ours
CLIP-Score ↑	0.364	0.364	**0.365**	0.360	0.364
FID ↓	25.56	25.93	26.13	26.17	**25.78**

5.5 Cross-Model Transferability

We evaluate user identification across different versions of Stable Diffusion, with the results reported in Table 6. The dual-anchor localization network used for SD2.1 is applied unchanged to the other models without additional training. Average identification accuracy remains between 0.997 and 0.998, demonstrating that robustness is preserved across the tested models and supporting cross-model compatibility.

5.6 Image Quality Evaluation

We visually compare images generated without watermarking and with different watermarking methods in Fig. 6. RingID [7] embeds structured ring patterns, which WIND [8] also uses for group identification; central circular artifacts appear in some samples from both methods, whereas our examples show no such conspicuous artifacts. As reported in Table 7, the CLIP-Scores [28] are close across methods and to the unwatermarked baseline, suggesting that watermark embedding has little effect on text-image alignment. Our method also achieves the FID [27] closest to the unwatermarked baseline among watermarking methods, indicating well-preserved generation quality.

5.7 Computational Efficiency and Scalability

We evaluate computational efficiency and scalability in terms of end-to-end per-image runtime and peak GPU memory usage on an NVIDIA RTX 4090 GPU. The reported runtime includes diffusion-based image generation, inversion, and watermark detection. With 100,000 and 200,000 candidate keys, our method requires 0.817 and 0.975 seconds per image, respectively, compared with 4.293 and 6.721 seconds for WIND [8]. Peak GPU memory usage is also lower, at 12.90 and 15.96 GB, versus 13.70 and 22.84 GB for WIND. Doubling the key space increases our runtime by 19.3%, compared with 56.6% for WIND, demonstrating lower end-to-end runtime, lower memory requirements, and more favorable scaling within the evaluated range.

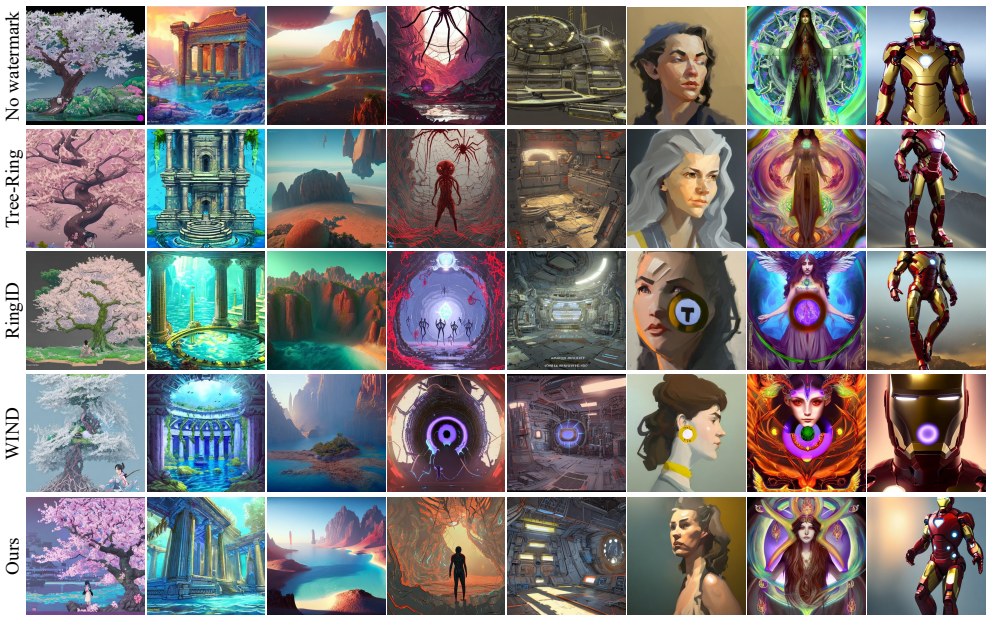


Fig. 6 Visual comparison of images generated without watermarking and with different watermarking methods.

6 Conclusion

We presented DASMark, a diffusion image watermarking framework for robust model verification and scalable user identification under geometric attacks. Reproducible user-specific initial noise and orthogonally coded dual anchors restore latent spatial correspondence for reliable identity matching. Experiments demonstrate robust verification and 99.7% average identification accuracy with 200,000 candidate identities, outperforming WIND by 11.0 and 18.6 percentage points under rotation and cropping followed by resizing, respectively. DASMark preserves image quality, achieves lower end-to-end runtime and GPU memory usage than WIND, and exhibits cross-model compatibility, supporting provenance tracing for active deepfake detection. Future work will address sensitivity to aggressive cropping and geometric distortions combined with brightness adjustments, and extend synchronization to more complex transformations.

Data availability

The experimental data supporting this study are available from the corresponding author upon reasonable request.

References

[1] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-resolution image synthesis with latent diffusion models. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 10684–10695 (2022).

[2] Ojha, U., Li, Y. & Lee, Y. J. *Towards universal fake image detectors that generalize across generative models. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 24480–24489 (2023).

[3] Wang, Z. *et al. DIRE for diffusion-generated image detection. Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 22445–22455 (2023).

[4] Fernandez, P., Couairon, G., Jégou, H., Douze, M. & Furon, T. *The Stable Signature: Rooting watermarks in latent diffusion models. Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 22466–22477 (2023).

[5] Zhu, J., Kaplan, R., Johnson, J. & Fei-Fei, L. *HiDDeN: Hiding data with deep networks. Proc. Eur. Conf. Comput. Vis. (ECCV)*, Vol. 11219 of *Lecture Notes in Computer Science*, 682–697 (Springer, Cham, 2018).

[6] Wen, Y., Kirchenbauer, J., Geiping, J. & Goldstein, T. *Tree-Rings watermarks: Invisible fingerprints for diffusion images. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 36, 58047–58063 (2023).

[7] Ci, H., Yang, P., Song, Y. & Shou, M. Z. *RingID: Rethinking Tree-Ring watermarking for enhanced multi-key identification. Proc. Eur. Conf. Comput. Vis. (ECCV)*, Vol. 15086 of *Lecture Notes in Computer Science*, 338–354 (Springer, Cham, 2024).

[8] Arabi, K., Feuer, B., Witter, R. T., Hegde, C. & Cohen, N. *Hidden in the noise: Two-stage robust watermarking for images. Proc. Int. Conf. Learn. Represent. (ICLR)* (2025). Available at https://openreview.net/forum?id=ll2nz6qwRG.

[9] Rössler, A. *et al. FaceForensics++: Learning to detect manipulated facial images. Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 1–11 (2019).

[10] Tancik, M., Mildenhall, B. & Ng, R. *StegaStamp: Invisible hyperlinks in physical photographs. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2117–2126 (2020).

[11] Feng, W. *et al. AquaLoRA: Toward white-box protection for customized Stable Diffusion models via watermark LoRA. Proc. 41st Int. Conf. Mach. Learn. (ICML)*, Vol. 235 of *Proceedings of Machine Learning Research*, 13423–13444 (PMLR, 2024).

[12] Song, J., Meng, C. & Ermon, S. *Denoising diffusion implicit models. Proc. Int. Conf. Learn. Represent. (ICLR)* (2021). Available at https://openreview.net/forum?id=St1giarCHLP.

[13] Yang, Z. *et al. Gaussian Shading: Provable performance-lossless image watermarking for diffusion models. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 12162–12171 (2024).

[14] Zheng, D., Zhao, J. & El Saddik, A. RST-invariant digital image watermarking based on log-polar mapping and phase correlation. *IEEE Trans. Circuits Syst. Video Technol.* **13**, 753–765 (2003).

[15] Pereira, S. & Pun, T. Robust template matching for affine resistant image watermarks. *IEEE Trans. Image Process.* **9**, 1123–1129 (2000).

[16] Fernandez, P. *et al. Geometric image synchronization with deep watermarking. Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2026).

[17] Fang, H. *et al. SynTag: Enhancing the geometric robustness of inversion-based generative image watermarking. Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 15416–15425 (2025).

[18] Qian, Y. *et al.* AnchorMark: Robust diffusion watermarking via latent-space rotation synchrony (2026). Preprint at https://arxiv.org/abs/2607.27551.

[19] Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization (2016). Preprint at https://arxiv.org/abs/1607.06450.

[20] Hendrycks, D. & Gimpel, K. Gaussian error linear units (GELUs) (2016). Preprint at https://arxiv.org/abs/1606.08415.

[21] Chen, L., Chu, X., Zhang, X. & Sun, J. *Simple baselines for image restoration. Proc. Eur. Conf. Comput. Vis. (ECCV)*, Vol. 13667 of *Lecture Notes in Computer Science*, 17–33 (Springer, Cham, 2022).

[22] Vaswani, A. *et al. Attention is all you need. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 30 (2017).

[23] Nibali, A., He, Z., Morgan, S. & Prendergast, L. Numerical coordinate regression with convolutional neural networks (2018). Preprint at https://arxiv.org/abs/1801.07372.

[24] Girshick, R. *Fast R-CNN. Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 1440–1448 (2015).

[25] Gustavosta. Stable-Diffusion-Prompts. Hugging Face Datasets (n.d.). https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts. Accessed 6 October 2026.

[26] Lu, C. *et al. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 35, 5775–5787 (2022).

[27] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. *GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 30 (2017).

[28] Radford, A. *et al. Learning transferable visual models from natural language supervision. Proc. Int. Conf. Mach. Learn. (ICML)*, Vol. 139 of *Proceedings of Machine Learning Research*, 8748–8763 (PMLR, 2021).

[29] Lin, T.-Y. *et al. Microsoft COCO: Common objects in context. Proc. Eur. Conf. Comput. Vis. (ECCV)*, Vol. 8693 of *Lecture Notes in Computer Science*, 740–755 (Springer, Cham, 2014).

[30] Cherti, M. *et al. Reproducible scaling laws for contrastive language-image learning. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2818–2829 (2023).

[31] Yang, P., Ci, H., Song, Y. & Shou, M. Z. *Can simple averaging defeat modern watermarks? Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 37, 56644–56673 (2024).

Funding. This work was supported in part by the Basic Research Program for Natural Science of Guizhou Province under Grant No. QIANKEHEJICHU-ZD[2025]-043 and the Doctoral Research Project of Guizhou Normal University under Grant No. GZNYD[2025]01.

Author contributions. Jietao Zheng and Yanli Chen conceived the study, developed the methodology, conducted the analysis and validation, and wrote the main manuscript text. Hanzhou Wu assessed the feasibility of the proposed method, provided guidance on manuscript preparation, and reviewed the manuscript. All authors reviewed and approved the manuscript for submission.

Additional information

Competing interests. The authors declare no competing interests.