SEAL: Semantic Aware Image Watermarking

Kasra Arabi, R. Teal Witter, Chinmay Hegde, Niv Cohen
New York University

Abstract

Generative models have rapidly evolved to generate realistic outputs. However, their synthetic outputs increasingly challenge the clear distinction between natural and AI-generated content, necessitating robust watermarking techniques to mark synthetic images. Watermarks are typically expected to preserve the integrity of the target image, withstand removal attempts, and prevent unauthorized insertion of the watermark pattern onto unrelated images. To address this need, recent methods embed persistent watermarks into images produced by diffusion models using the initial noise of the diffusion process. Yet, to do so, they either distort the distribution of generated images or require searching a large dictionary of candidate noise patterns for detection.

In this paper, we propose a novel watermarking method that embeds semantic information about the generated image into the noise pattern, enabling a distortion-free watermark that can be verified without requiring a database of key patterns. Instead, the key pattern can be inferred from the semantic embedding of the image using locality-sensitive hashing. Furthermore, conditioning the watermark detection on the original image content improves its robustness against forgery attacks. To demonstrate that, we consider two largely overlooked attack strategies: (i) an attacker extracting the initial noise and generating a novel image with the same pattern; (ii) an attacker inserting an unrelated (potentially harmful) object into a watermarked image, while preserving the watermark. We empirically validate our method's increased robustness to these attacks. Taken together, our results suggest that content-aware watermarks can mitigate risks arising from image-generative models. Our code is available at https://github.com/Kasraarabi/SEAL.

1. Introduction

The growing capabilities of generative models pose risks to society, including misleading public opinion, violating privacy or intellectual property, and fabricating legal evidence [5, 16, 28]. Watermarking methods aim to mitigate such risks by allowing the detection of synthetically generated content.

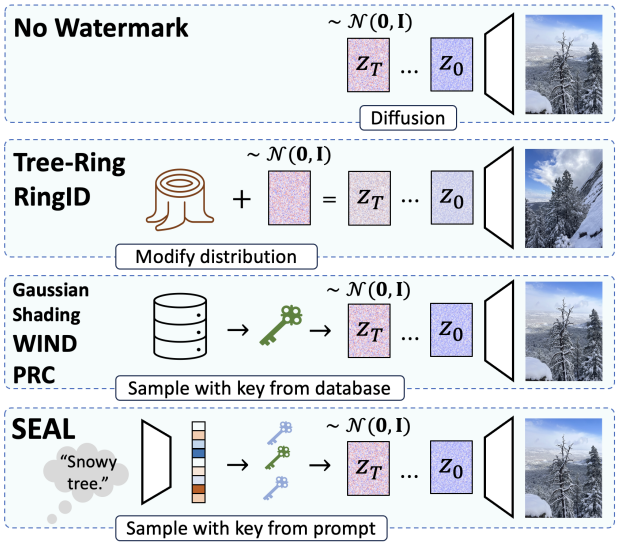


Figure 1. ***Illustration of different watermarking frameworks using the initial noise of diffusion models. No Watermark:*** *A diffusion model maps pure Gaussian noise to an image.* ***Tree-Ring:*** *A pattern is added to the initial noise, modifying the distribution of generated images in a detectable way.* ***Key-Based Watermarking:*** *A key is sampled to generate distortion-free images linked to the key.* ***Ours (SEAL):*** *The initial noise is conditioned on multiple keys derived from the image's semantic embedding, with each key influencing a different patch.*

Yet, many conventional watermarking techniques lack robustness against adversaries who attempt to remove them using regeneration attacks powered by recent generative models [2, 12, 36]. To address this, new watermarking techniques leveraging similar advances in generative models offer an increased robustness against such attacks [4, 31, 34]. Namely, these methods embed a watermarking pattern in the initial noise used by a diffusion model. These patterns have been shown to be more robust against existing removal attacks.

However, existing watermarks that utilize the diffusion model initial noise tend to be vulnerable to other attacks aim-

ing to "steal" the watermark and apply it to images unrelated to the watermark owners [17, 22, 32]. Some of these *watermark forgery* attacks can be evaded by using a distortion-free watermark - generating watermarked images from a similar distribution to the distribution of non-watermarked images; therefore exposing less information about the watermark identity. Even so, keeping track of a very large number of watermark identities requires maintaining a database of used noises, and might still be forgeable by other attacks [17, 22].

To address these challenges, we introduce SEAL - *Semantic Embedding for AI Lineage*, a method that embeds watermark patterns directly tied to image semantics. Our approach enables direct watermark detection from image samples and offers the following key properties: (i) *Distortion-free:* As in previous works, we utilize pseudo-random hash functions to generate an initial noise that is similar to the noise used by non-watermarked models, ensuring a similar distribution of generated images. (ii) *Robust to regeneration attacks:* Similar to prior watermarking methods based on DDIM inversion, our approach demonstrates resilience against regeneration-based removal attempts [36]. (iii) *Correlated with image semantics:* The applied watermark encodes semantic information from the image. (iv) *Independent of a historical database:* Our approach embeds watermarks without requiring access to a database of used noise patterns.

Our key insight is that we can encode semantic information about the image content in a distortion-free watermark by embedding a semantic encoding of the generation directly into the initial noise. Namely, we may use an encoding of the requested image semantics to seed different pseudo-random patches that compose the initial noise. We ensure the encoded embedding correlates strongly with the resulting image content, and not just with the prompt, which is important since the prompt is not available during detection. At detection time, our approach identifies an image as watermarked only when the watermark pattern is both present and properly correlated with the semantics of the given image. We describe in detail our watermarking technique in Section 3.

Correlating our watermarking algorithm to the image semantics also allows us to resist forgery attempts that are challenging for many existing approaches. An attacker attempting to forge our watermark onto unauthorized content would alter the image's semantic embedding, breaking its correlation with the embedded pattern and rendering the watermark invalid.

One mostly overlooked attack involves an adversary altering only small portions of a watermarked image while preserving the rest of its content. In such cases, the attacker can manipulate the image to be offensive, illegal, or damaging to the watermark owner's reputation, all while the original watermark remains detectable. We term this attack the CAT ATTACK, as the attacker may add an object to the image (e.g., a cat) and expect the watermark to persist. We evaluate the potential damage of such tamperings and demonstrate that our method provides robustness against both the CAT ATTACK and other forgery attempts, even by adversaries who obtain accurate copies of our initial noise. Our experiments confirm our method's effectiveness against these novel threats as well as previously studied attack vectors.

Our contributions are as follows:

- We propose SEAL, a semantic-aware database-free watermarking method that integrates image semantics into the watermark, ensuring it becomes invalid under severe semantic changes.
- We investigate the CAT ATTACK, highlighting the risks of local edits to watermark owners and assessing their potential impact.
- We empirically demonstrate the effectiveness of our watermark against various attacks, including its resistance to adversarial edits.

2. Related Works

Recent research on image watermarking can be broadly categorized into post-processing and in-processing approaches, each offering distinct trade-offs between quality, robustness, and deployment practicality [2]. We cover here *In-Processing Methods*, and for *Post-Processing Methods* refer to Section 7.

In-Processing Watermarking Methods. In-processing approaches integrate the watermark directly within the image generation process. Some methods modify the generative model by fine-tuning specific components, as demonstrated in Stable Signature [12, 26, 35]. An alternative class of techniques manipulates the initial noise of the generation process, thereby embedding the watermark without extensive model retraining. For example, Tree-Ring [31] embeds a Fourier-domain pattern into the initial noise, which can be detected through DDIM inversion [29], while RingID [8] extends this idea to support multiple keys. Other notable methods include Gaussian Shading, which produces a unique key for each watermark owner [34], PRC that leverages pseudo-random error-correcting codes for computational undetectability [14], and WIND, which employs a two-stage detection process to enables a very large number of keys [4].

Locally Sensitive Hashing in High-Dimensional Spaces. Recent advances in approximate nearest neighbor (ANN) search have increasingly relied on the power of Locally Sensitive Hashing (LSH) to address the challenges of dealing with high-dimensional data. Originally introduced by Indyk and Motwani [15] and further refined by Gionis et al. [13], LSH employs randomized hash functions that ensure similar

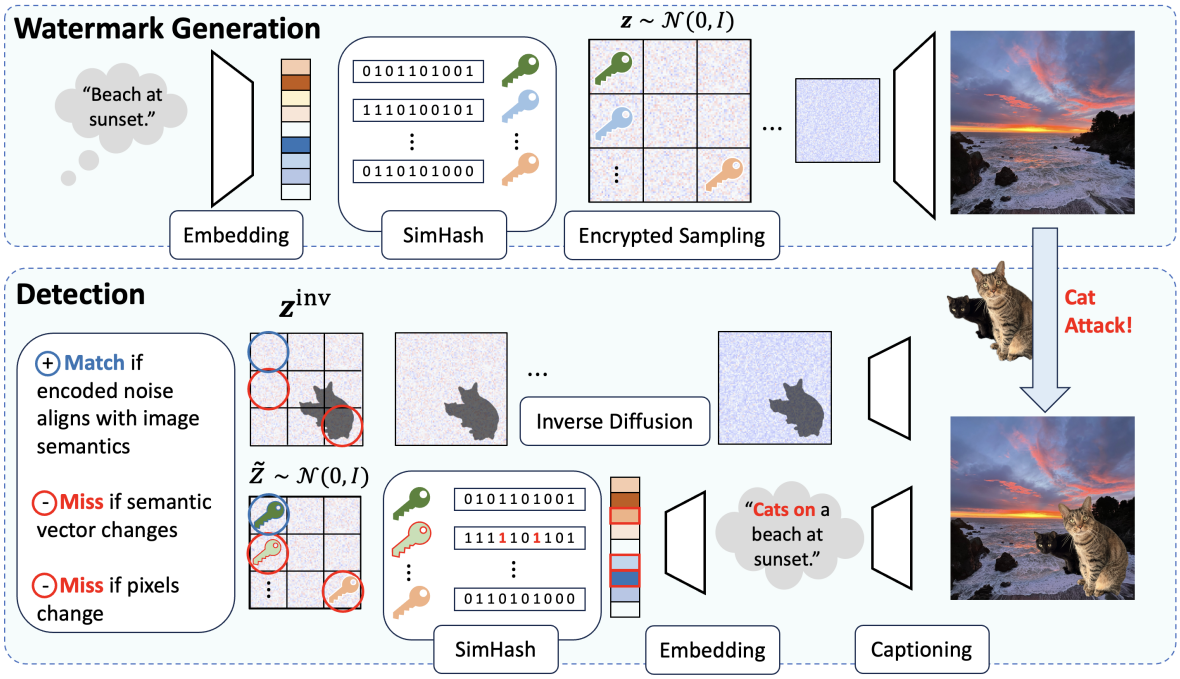


Figure 2. ***Illustration of the* SEAL *watermarking framework for diffusion models using semantic-aware noise patterns. Watermark Generation:*** *A textual prompt (e.g., "Beach at sunset.") is first embedded into a semantic space. The embedding is then processed using* ***SimHash*** *to generate discrete keys, which are used in* ***Encrypted Sampling*** *to choose the initial noise* $\mathbf{z} \sim \mathcal{N}(0, I)$*. The watermarked noise then undergoes standard diffusion to generate the final image.* ***Watermark Detection:*** *The image is captioned to obtain an embedding, which is then processed by* ***SimHash*** *to generate a reference noise, similarly to watermark generation. This noise remains correlated with the initial noise used during generation as long as the image semantics remain unchanged. The initial noise is also estimated directly through* ***Inverse Diffusion*** *to approximate the actual initial noise used during its generation. If there are insufficient matches between the reference noise and the noise obtained from inversion, the watermarking framework flags the image as non-watermarked. If a key match is found but the image is still deemed suspicious, a detailed inspection of the patches can be performed to identify local edits.*

data points are mapped to the same bucket with high probability. For a hash family $\mathcal{H}$, the collision probability is given by

$$P(h(x) = h(y)) \approx \text{similarity}(x, y), \quad h \in \mathcal{H}.$$

Subsequent improvements by Datar et al. [9] and Andoni and Indyk [3] have enhanced both the efficiency and robustness of LSH methods, making them key for large-scale, high-dimensional search tasks.

3. SEAL: Semantic Aware Watermarking

3.1. Motivation

Watermarking methods suffer from an inherent trade-off: a watermark that is harder to remove is also easier to attach to unrelated generations, compromising the reputation of the watermark owner [5]. One suggested solution to overcome this trade-off, might be maintaining a database of past generations, such that the owner could compare a seemingly watermarked image to the actual past generations. Yet, this solution is not without its problems. First, maintaining and searching a rapidly growing database, which expands with each new generation, can be challenging. Second, safeguarding the database itself may pose security risks. Finally, in various use cases, the watermark owner may not only wish to detect if an image is watermarked but also provide to a third party evidence that it was. We therefore turn to suggest a watermarking scheme that is hard to remove, hard to forge, and does not rely on maintaining a database of past generations.

Our core idea is to use a distortion-free initial noise pattern not only to indicate the origin of the image but also to encode which semantic information the image may contain. We do so in three stages (see also Figure 2): (i) *Semantic Embedding* – we obtain a vector representing the expected semantic content in each generated image (ii) *SimHash Encoding* – we encode the semantic vector using a set of multi-bit hash functions (iii) *Encrypted Sampling* – The pseudo-random outputs of the hash functions are combined to produce the initial noise for the diffusion denoising process. Taken together, these steps set an initial noise that is both distortion-free with respect to standard random initialization and correlated

(a) (b)

Figure 3. ***Effect of the Cat Attack on* SEAL.** *(Left) A cat image is pasted onto a watermarked image at a random position and scale. (Right) Our method detects this modification by identifying elevated ℓ_2 norms in affected patches. Note that the displayed norms are rounded to the nearest integer.*

with the semantics of the input prompt (see Section 3.3). We describe our watermarking method in detail below.

3.2. Method

Formally, our method first creates a semantic vector $\mathbf{v}$ and uses it to sample the initial noise $\mathbf{z}$ for the watermarked image. During detection, we aim to verify the connection between the used initial noise $\mathbf{z}$ and the semantic embedding of the image. When approximating $\mathbf{z}$ from the generated image during detection and verifying it, we consider the following error sources:

- We do not have access to $\mathbf{v}$ at detection time; instead, we must use an approximate version $\tilde{\mathbf{v}}$ derived from the image we are analyzing. Using $\tilde{\mathbf{v}}$, we produce an approximate version of the used initial noise $\tilde{\mathbf{z}}$.
- Because of the randomness in the diffusion process and its inversion, we cannot estimate $\mathbf{z}$ accurately; instead, we get an approximation of the inverted noise $\mathbf{z}^{\text{inv}}$.

Ideally, a watemarked image would yield a perfect match between the noisese derived from the image semantics $\mathbf{z}^{\text{inv}}$ and the inverted $\tilde{\mathbf{z}}$ but this is not guaranteed because both differ from $\mathbf{z}$ due to the error sources mentioned above. Yet, we can mitigate this by independently embedding the image semantics across multiple patches. Therefore, our method provides a high likelihood that even if some patches do not match because of the challenges discussed above, many of the patches will match as long as the suspect image is watermarked.

Watermark Generation

The first step of the generation process is to find a semantic vector $\mathbf{v}$ describing the image that will be generated. Ideally, the semantic vector depends only on the prompt and correlates exclusively with images generated from it. Yet, in practice, predicting the final image semantics based on the user prompt is difficult.

Algorithm 1 Watermark Generation

Input: `prompt`: text prompt, n: number of patches, b: number of bits per patch, `salt`: secret salt
Output: Watermarked image of `prompt`
$\mathbf{z}^{\text{pre}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{x}^{\text{pre}} \leftarrow \text{Diffusion}(\mathbf{z}^{\text{pre}}, \texttt{prompt})$
$\mathbf{v} \leftarrow \text{Embed}(\text{Caption}(\mathbf{x}^{\text{pre}}))$
for $i = 1, \ldots, n$ **do**
  $\mathbf{z}_i \leftarrow \texttt{SimHash}(\mathbf{v}, i, \texttt{salt})$
end for
return $\text{Diffusion}(\mathbf{z}, \texttt{prompt})$

Algorithm 2 `SimHash`

Input: $\mathbf{v}$: semantic vector, i: patch index, `salt`: secret salt, b: number of bits, `hash`: cryptographic hash function
Output: Semantic, secure, normally distributed noise
$\texttt{bits} \leftarrow \mathbf{0}$ // Initialize hash input
for $j = 1, \ldots, b$ **do**
  // Reproducibly sample random vector
  $s \leftarrow \texttt{hash}(i, j, \texttt{salt})$
  Sample $\mathbf{r}_j^{(i)} \overset{s}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\texttt{bits}[j] \leftarrow \text{sign}(\langle \mathbf{v}, \mathbf{r}_j^{(i)} \rangle)$ // Random projection
end for
$s_i \leftarrow \texttt{hash}(\texttt{bits}, i, \texttt{salt})$
return $\mathbf{z}_i \overset{s_i}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$

To approximate the generated image semantics, our solution begins by generating a proxy image $\mathbf{x}^{\text{pre}}$. We first caption the image using BLIP-2 model [19]. Then, the caption is embedded into a latent semantic space using the Paraphrase Mpnet Base V2 model [24], resulting in a semantic vector $\mathbf{v}$ which captures the high-level semantics of the generated image by the prompt (a similar concept also explored in the concurrent work of SWIFT [10]).

Semantic Embedding Optimization. During detection, the generated image will be captioned to obtain a semantic vector $\tilde{\mathbf{v}}$, approximating the semantic vector $\mathbf{v}$ that was used to seed the random noise. To ensure a similarity between $\mathbf{v}$ and $\tilde{\mathbf{v}}$, it is not enough only to generate the proxy image $\mathbf{x}^{\text{pre}}$ with the same prompt (a qualitative comparison of the captioned proxy image and the final generated image can be found at Figure 9). Therefore, to encourage the embedding $\mathbf{v}$ to correlate to $\tilde{\mathbf{v}}$ and not to unrelated vectors, we fine-tuned the embedding model using 10k pairs of related captions, leading to additional improvements (Figure 5). The full implementation details of the fine-tuning process can be found in Section 9.4.

After obtaining the desired semantic embedding, we generate the watermarked noise $\mathbf{z}$ using the semantic vector $\mathbf{v}$

and the SimHash algorithm described below. Finally, we will use the diffusion mode to generate the image with the watermarked initial noise. The generation algorithm is summarized in Algorithm 1.

Semantic Patterns with SimHash

The core subroutine of our watermarking method is SimHash [7], used to generate initial noise patches correlated to a given vector (Algorithm 2). SimHash takes a vector $\mathbf{v}$ and generates an initial noise $\mathbf{z}_i$ for patch i, allowing a verifier to later determine whether $\mathbf{z}_i$ is related to $\mathbf{v}$. Namely, the semantic vector $\mathbf{v}$ is passed through a locality-sensitive hashing method that generates representations of $\mathbf{v}$ in terms of its projections onto random directions.

Specifically, SimHash projects $\mathbf{v}$ onto a set of random vectors for each patch of the initial noise map. It uses b projection vectors for each of the k noise patches. Each noise patch is generated using a seed determined by the sign of the projection of the semantic vector onto each of the b projection directions. For $i \in \{1, \ldots, k\}$, the seed and the noise for patch i are:

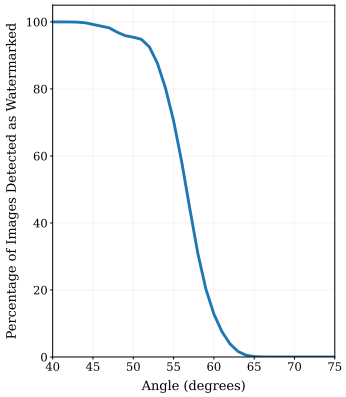
$$s_i = \texttt{hash}(\text{sign}(\langle \mathbf{v}, \mathbf{r}_1^i \rangle), \ldots, \text{sign}(\langle \mathbf{v}, \mathbf{r}_b^i \rangle), i, \texttt{salt}).$$
$$\mathbf{z}_i \overset{s_i}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$$

This ensures that similar semantic vectors would yield similar hash values.

Yet, having repetitive bit inputs (s_i) may result in repetitive patches in the initial noise, and therefre may distort image generation. Therefore, we include the patch index in the hash function input to ensure that $s_i \neq s_j$ even when the input bits are identical (see Figure 8 for generation samples in the case of repetitive noise patches). For cryptographic security, we also use a user-specific secret salt.

Algorithm 3 Watermark Detection

Input: $\tilde{\mathbf{x}}$: suspect image, τ : patch distance threshold, n: number of patches, m^{match}: match threshold, b: number of bits per patch, `salt`: secret salt
Output: Watermark detection (**True/False**)
$\tilde{\mathbf{v}} \leftarrow \text{Embed}(\text{Caption}(\tilde{\mathbf{x}}))$
$\mathbf{z}^{\text{inv}} \leftarrow \text{InverseDiffusion}(\tilde{\mathbf{x}})$
$m \leftarrow 0$
for $i = 1, \ldots, n$ **do**
  $\tilde{\mathbf{z}}_i \leftarrow \texttt{SimHash}(\tilde{\mathbf{v}}, i, \texttt{salt})$
  if $\|\tilde{\mathbf{z}}_i - \mathbf{z}_i^{\text{inv}}\|_2 < \tau$ **then**
    $m++$
  end if
end for
return $m \geq m^{\text{match}}$



Angle $\theta(\mathbf{v}, \tilde{\mathbf{v}})$	Detection Probability
65°	8.55×10^{-4}
60°	0.053
55°	0.551
50°	0.998
45°	1.000

Figure 4. ***Watermark Detection vs. Semantic Similarity.*** We plot the empirical probability of detecting a watermark as a function of the angle between the semantic embedding used for watermark generation and that of the inspected image ($n = 1024$, $b = 7$). The table shows the analytical detection probabilities at key angles calculated by Lemma 3.2, illustrating how sharply SimHash distinguishes semantically related images from unrelated ones.

Watermark Detection

For detection, we generate noise based on the semantic content of the image and check how well it corresponds to the reconstructed noise obtained through DDIM inversion (Algorithm 3). We begin by embedding the image to get a semantic vector $\tilde{\mathbf{v}}$ that captures the content of the image. SimHash is then applied to this vector as in the watermark generation process, generating an estimated initial noise $\tilde{\mathbf{z}}$. Finally, we use inverse diffusion (e.g., DDIM [29]) to approximately reconstruct the initial noise $\mathbf{z}^{\text{inv}}$ from the image.

Since $\mathbf{v}$ and $\tilde{\mathbf{v}}$ may differ, the originally used noise $\mathbf{z}$ and $\tilde{\mathbf{z}}$ are not necessarily the same. However, by the similarity property of SimHash, $\mathbf{z}$ and $\tilde{\mathbf{z}}$ will be identical on some patches with very high probability as long as $\mathbf{v}$ and $\tilde{\mathbf{v}}$ are close. On any patch number i where the SimHash patch match ($\tilde{\mathbf{z}}_i = \mathbf{z}_i$), we get:

$$\|\tilde{\mathbf{z}}_i - \mathbf{z}_i^{\text{inv}}\|_2 = \|\mathbf{z}_i - \mathbf{z}_i^{\text{inv}}\|_2. \tag{1}$$

For such patches, the only challenge in identifying the watermark stems from the difference between the originally used noise and the reconstructed noise through DDIM. Empirically, we find that the ℓ_2-norm of the difference between the inverted and expected noise patches ($\|\tilde{\mathbf{z}}_i - \mathbf{z}_i^{\text{inv}}\|_2$) allows us to detect whether the inverted noise patches originated from the suspected noise patch with a $> 99.9\%$ ROC-AUC.

Semantic Similarity Detection. Finally, in order to detect whether an image was initially generated with our watermark, we count the number of patches that *match* (i.e., their ℓ_2-norm distance is below a threshold τ). If the number of matches is above a set threshold n^{match} then we declare the image is watermarked. In Section 3.3, we analyze the probability of correctly identifying a watermarked image.

Tampering Detection with a Spatial Test. In addition to the association between the watermark and the semantic embedding, edits such as object addition, removal, or modification are likely to alter the estimated initial noise in the affected image regions. This enables our watermark to provide localized information about edits that might have been made to the image. Consequently, even when the semantic embedding of the image $\tilde{\mathbf{v}}$ aligns well with the initial embedding used to seed the noise $\mathbf{v}$, such tampering edits can still be detected by identifying localized patches in the reconstructed initial noise that neither match the expected noise nor any other valid input to the hash function.

To detect such cases, we may inspect the noise patches one by one. Given the model owner's private information, we may recover the b input bits used to seed each patch with an exhaustive search over the 2^b options per patch, and recover a matching initial noise. Comparing this reconstructed noise to the inverted noise $\mathbf{z}^{\text{inv}}$ allows us to detect which patches may have been modified. The total time for this search scales as $n \cdot 2^b$ (which is much faster than naively searching over all $2^{(b \cdot n)}$ possible initial noise). After obtaining a per patch noise-matching map (see Figure 3b), we may apply a *spatial test* as the one described in Section 9.2 to detect tampering attempts. In any case, the local patch inspection is only required when an image is deemed watermarked by semantic similarity detection; but the watermark owner would like to have a finer understanding of the edits that might have been applied to it. This inspection is especially useful against the CAT ATTACK, described in Section 4.

3.3. Analysis

Before formally analyzing our watermarking scheme, we state a simplifying assumption on the distance between the initial and reconstructed noise patches. We assume the noise patches are close if and only if the suspect image was produced from the same noise as the one given by our watermarking scheme. The impact of low-likelihood events, where unrelated patches end up close after noise reconstruction, remains part of our empirical analysis in Section 4.

Assumption 3.1 (Patch Distance Separation)**.** *There is a threshold τ^{dist} so that, for all generation noises $\mathbf{z}$, inverted noises $\mathbf{z}^{\text{inv}}$, and patches indeces $i \in [k]$,*

$$\|\mathbf{z}_i - \mathbf{z}_i^{\text{inv}}\|_2 \leq \tau^{\text{dist}}$$

if and only if $\mathbf{z}^{\text{inv}} = \text{InverseDiffusion}(\text{Diffusion}(\mathbf{z}))$.

An immediate consequence of the patch distance separation assumption is that we never declare an image as watermarked if its initial noise was not generated using our watermarking scheme.

Unrelated Prompts. A key property of our watermarking approach is its resistance to forgeries generated from unrelated prompts. Prior watermarking methods declare an image as watermarked as long as the watermarking pattern is embedded in the initial noise and the diffusion and inverse diffusion processes remain reasonably accurate. However, this creates vulnerabilities - an adversary could take an existing watermark and apply it to an unrelated, potentially offensive, or misleading prompt. In contrast, our approach strengthens watermark integrity by requiring that the new prompt remains semantically close to the original. This ensures that watermarks are not erroneously detected in entirely unrelated images. We formalize this claim the the lemma below.

Lemma 3.2 (Detection Probability)**.** *Consider a suspect image $\tilde{\mathbf{x}}$ produced from our watermarking scheme with initial semantic vector $\mathbf{v}$. Let $\tilde{\mathbf{v}}$ be the (possibly quite different) semantic embedding of $\tilde{\mathbf{x}}$, and $\theta \in [-90°, 90°]$ be the angle between $\mathbf{v}$ and $\tilde{\mathbf{v}}$. Set θ^{mid} as the threshold between semantic vectors we deem* related *vs.* unrelated*. The probability that we identify the image as watermarked is*

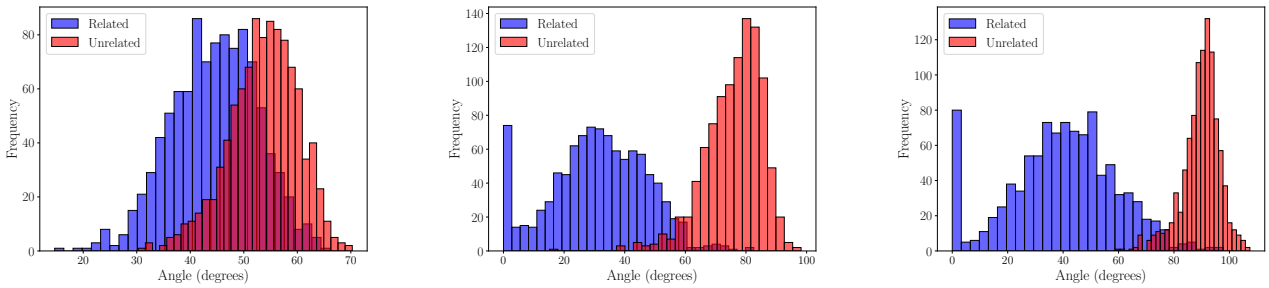
$$\sum_{k=\lfloor n\rho(\theta^{mid})\rfloor}^{n} \binom{n}{k} \rho(\theta)^k (1-\rho(\theta))^{n-k}. \tag{2}$$

where $\rho(\theta) = \left(1 - \frac{\theta}{180°}\right)^b$.

We illustrate in the example below the sharp detection thresholds Lemma 3.2 implies. Namely, we show how the watermark detection probability varies with semantic similarity between the original and a potentially modified image. We delay the proof of Lemma 3.2 to the appendix.

Example 3.3 (Sharp Detection Thresholds)**.** *Our watermarking scheme embeds a semantic vector $\mathbf{v}$ into an image at generation time. When evaluating a suspect image that was generated via our watermark, we extract its current semantic vector $\tilde{\mathbf{v}}$. The probability of a watermark detection depends on the semantic angle $\theta(\mathbf{v}, \tilde{\mathbf{v}})$ between $\mathbf{v}$ and $\tilde{\mathbf{v}}$.*

For instance, Figure 5c illustrates a separation between vectors associated with the original image and those that are unrelated, occurring at a threshold of approximately $\theta^{mid} \approx 55°$. When our watermarking scheme is run with $\theta^{mid} = 55°$, $n = 1024$, and $b = 7$ (see Figure 7 for ablation of these parameters). Figure 4 quantifies the probability of a watermark detection: We observe near-perfect separation between related and unrelated watermarked images



(a) ***Image Feature Vector.*** Direct use of image feature embedding fails to separate related from unrelated images.

(b) ***Caption Embeddings.*** Employing caption embeddings from *blip2-flan-t5-xl* and *paraphrase-mpnet-base-v2* yields improved separation.

(c) ***Fine-tuned Caption Embeddings.*** Fine-tuning the embedding model on 10k caption pairs further enhances separation.

Figure 5. ***Ablation of Embedding Strategies for Watermark Detection.*** Comparison of angle separation between related and unrelated images using different embedding approaches. The raw image feature vector (left) fails to distinguish semantic relationships, while caption embeddings (center) substantially improve separation. Fine-tuning the embedding model (right) yields additional gains in detection accuracy.

for angles exceeding $5°$ *beyond the threshold. For comparison, Section 13.2 presents the semantic angle shift resulting from the simple insertion of different objects.*

4. Empirical Analysis

In this section, we empirically evaluate the robustness of SEALto different attacks.

Setting. To ensure a fair comparison with prior work [4, 8, 31], we use Stable Diffusion-v2 [25] with 50 inference steps for both generation and inversion for all methods. Evaluations were conducted on a set of prompts sourced from [27]. We set $n = 1024$ and $b = 7$ for all experiments. An ablation study on the effects of n and b is available in Section 10.

Regeneration with the Private Model. Prior works assume that the attacker lacks access to the model weights (which are needed for accurate DDIM noise inversion) and that the noise used during generation cannot be forged or approximated with sufficient accuracy [4]. Going beyond previous studies, we consider here a more challenging scenario in which the attacker has full access to the model weights and can invert the generated image using the same model that produced it. The attacker's access to the private model is taken as an upper bound for the attacker's capability in practical forgery attacks [4, 17, 22].

In our experiment, we first generate an image using watermarked noise. We then perform an inversion with the same model to recover an approximate initial noise, which is subsequently used to generate a second, forged, image. Because the attack prompt differs from the original prompt, the semantic embedding of the image $\tilde{\mathbf{v}}$ changes to $\mathbf{v}_{attack}$. The detection algorithm, therefore compares the estimated noise to a reference derived from $\mathbf{v}_{attack}$ (and not from $\tilde{\mathbf{v}}$). The noise pattern derived from $\mathbf{v}_{attack}$ during detection is less likely to correlate to the pattern embedded in the image, enabling the detection algorithm to declare the image as not watermarked and evade the attack. As can be seen in Table 2, our method uniquely provides non-trivial robustness in this setting.

We also evaluate the Latent Forgery Attack directly [17] in Section 11.

Cat Attack. A significant practical threat to the reputation of a watermark owner arises from localized modifications that shift the semantic interpretation of a watermarked image, as opposed to producing a wholly new image. To evaluate our method's resilience against such tampering, we introduce an evaluation we term the CAT ATTACK.

In this experiment, a cropped object (e.g., a cat) is pasted onto a watermarked image. The cat image is randomly resized to between 30% and 60% of the watermarked image's dimensions and placed at a random location, as exemplified in Figure 3a. Unlike previous watermarking techniques that may overlook semantic content, our approach is designed to detect such alterations.

As shown in Figure 3b, the pasting of the object leads to elevated ℓ_2 norms in the affected patches. Quantitative results are presented in Table 1, and comparison of different object sizes can be found in the Table in Table 4. These results reveal that while our basic detection offers some robustness, integrating the local spatial test (described in Section 3) significantly improves the detection of these edits. This demonstrates a key advantage of our method. We note that in a different setting, methods such as [26, 35] offer a strong solution for tampering detection via post-hoc watermarking. Since robustness to tampering can be in tension with resistance to removal attacks, we next analyze our method's performance against standard removal attacks.

Table 1. ***Detection of the Cat Attack.*** *ROC-AUC of detecting edits in generated images, as described in Section 4.*

Method	AUC
WIND	0.000
Tree-Ring	0.000
Gaussian Shading	0.000
SEAL	0.551
SEAL+ Spatial Test	**0.982**

Table 2. ***Robustness to Private Model-Based Forgery Attack.*** *An attacker with access to the private model weights can approximate the watermarked initial noise by inverting a watermarked image using the private model. We evaluate how accurately different methods evade the false identification of unrelated images, generated with this initial noise, as watermarked.*

Method	AUC
WIND	0.000
Tree-Ring	0.000
Gaussian Shading	0.000
SEAL	**0.708**

Regeneration Based Removal Attack. Our method is robust to regeneration-based removal attacks [36], similarly to other initial-noise-based approaches [4, 14, 34], and it significantly outperforms classical watermarking methods (see Section 13.3).

Steganalysis Removal Attack. We evaluate the robustness of our method against a steganalysis attack [33] that attempts to approximate the watermark by subtracting non-watermarked images from watermarked ones. As shown in Table 3, SEAL maintains high performance under this attack.

Robustness to Image Transformations. We evaluated the robustness of SEAL under a standard suite of image transformations (see Section 9.3). As shown in Figure 6, SEAL achieves an average detection rate of 0.896 under these conditions. This is comparable to some watermarking techniques and somewhat lower than others [31]. Yet, our method provides a unique resistance to forgery. Further enhancements, such as incorporating rotation search or sliding-window search during detection (see [4]), could improve its robustness against removal attempts.

Ablation of Captioning and Embedding Models. A straightforward approach for our method to approximate the final image semantics to embed it in the noise would be to use the visual feature vector from the proxy-generated $\mathbf{x}^{\text{pre}}$ rather than the embedding of its caption. However, as illustrated in Figure 5a, this approach fails to yield a clear separation between related and unrelated images. Consequently, we employ the captioning and caption-embedding for deriving caption embeddings, which results in a more distinct separation as shown in Figure 5b. To further enhance our method's accuracy, we fine-tuned the embedding model using 10k pairs of related captions, leading to additional improvements (Figure 5c).

Generation Quality. Our watermarking method is distortion-free at the single-image level, since the added noise is sampled from a pseudo-random Gaussian distribution, similarly to non-watermarked image generation. As a result, all single-image quality metrics remain identical to those of non-watermarked images (see Table 7, Figure 10).

5. Limitation and Discussion

Stronger Forgery Attacks. Although we evaluated a stronger set of forgery attacks compared to previous works, other types of forgery attacks might still potentially compromise our watermark. For example, a highly persistent attacker might attempt to gather information about the correlation between individual initial noise patches and the image semantics. While not theoretically impossible, an attacker would face several practical limitations in carrying out such an attack. Among them are the lack of access to the private model weights, the inherent stochasticity of the watermark, and the watermark owner's ability to deploy multiple instances of the hash function by using multiple secret salts.

Attacker Advantage and Removal Attacks. Our method is more vulnerable to removal attacks than some existing methods. However, we believe that a sufficiently persistent attacker can remove most current watermarks. Nonetheless, improving watermark robustness against forgery attacks holds significant societal value - it is essential for protecting the model owner's reputation and, consequently, for enabling practical deployment.

Additional limitations and discussion points can be found in Section 12.

6. Conclusion

We introduce the first initial noise-based watermarking method for diffusion models that is both database-free and semantic-aware. Our suggested watermark is uniquely robust against a new class of stronger forgery attacks. We hope our work highlights the potential of semantic-aware watermarking and helps pave the way forward for further research in this area.

References

[1] Ali Al-Haj. Combined dwt-dct digital image watermarking. *Journal of computer science*, 3(9):740–746, 2007. 1

[2] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, et al. Waves: Benchmarking the robustness of image watermarks. *arXiv preprint arXiv:2401.08573*, 2024. 1, 2

[3] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 3

[4] Kasra Arabi, Benjamin Feuer, R Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. *arXiv preprint arXiv:2412.04653*, 2024. 1, 2, 7, 8, 3

[5] Charlotte Bird, Eddie Ungless, and Atoosa Kasirzadeh. Typology of risks of generative text-to-image models. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 396–410, 2023. 1, 3

[6] Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images, 2023. 4

[7] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002. 5, 1

[8] Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European Conference on Computer Vision*, pages 338–354. Springer, 2024. 2, 7

[9] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004. 3

[10] Gautier Evennou, Vivien Chappelier, Ewa Kijak, and Teddy Furon. Swift: Semantic watermarking for image forgery thwarting. In *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2024. 4

[11] Pierre Fernandez, 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. 1

[12] Pierre Fernandez, 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, 2, 4

[13] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, pages 518–529, 1999. 2

[14] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. *arXiv preprint arXiv:2410.07369*, 2024. 2, 8, 3

[15] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. 2

[16] Kokil Jaidka, Tsuhan Chen, Simon Chesterman, Wynne Hsu, Min-Yen Kan, Mohan Kankanhalli, Mong Li Lee, Gyula Seres, Terence Sim, Araz Taeihagh, et al. Misinformation, disinformation, and generative ai: Implications for perception and policy. *Digital Government: Research and Practice*, 6 (1):1–15, 2025. 1

[17] Anubhav Jain, Yuya Kobayashi, Naoki Murata, Yuhta Takida, Takashi Shibuya, Yuki Mitsufuji, Niv Cohen, Nasir Memon, and Julian Togelius. Forging and removing latent-noise diffusion watermarks using a single image. *arXiv preprint arXiv:2504.20111*, 2025. 2, 7

[18] Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models, 2024. 4

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 4

[20] 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

[21] Shweta Mahajan, Tanzila Rahman, Kwang Moo Yi, and Leonid Sigal. Prompting hard or hardly prompting: Prompt inversion for text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6808–6817, 2024. 3

[22] Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20937–20946, 2025. 2, 7

[23] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*, pages 271–274. IEEE, 2008. 1

[24] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 4

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 7

[26] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. *arXiv preprint arXiv:2411.07231*, 2024. 2, 7

[27] Gustavo Santana. Stable-diffusion-prompts. https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts, 2024. Accessed: 2024-11-20. 7, 5

[28] Irene Solaiman, Zeerak Talat, William Agnew, Lama Ahmad, Dylan Baker, Su Lin Blodgett, Canyu Chen, Hal Daumé III, Jesse Dodge, Isabella Duan, et al. Evaluating the social impact of generative ai systems in systems and society. *arXiv preprint arXiv:2306.05949*, 2023. 1

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5

[30] 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

[31] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023. 1, 2, 7, 8

[32] Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? *Advances in Neural Information Processing Systems*, 37:56644–56673, 2024. 2

[33] Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Steganalysis on digital watermarking: Is your defense truly impervious? *arXiv preprint arXiv:2406.09026*, 2024. 8

[34] 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, 2, 8, 3

[35] 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. 2, 7

[36] 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. 1, 2, 8, 4

[37] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018. 1, 4