A Dual-Protection Framework for Copyright Protection and Image Editing Using Multi-Label Conformal Prediction

Anonymous authors
Paper under double-blind review

Abstract

Recent advances in diffusion models have significantly enhanced image editing capabilities, raising serious concerns about copyright protection. Traditional watermarks often fail to withstand diffusion-based edits, making image protection challenging. To address this, we propose a method that embeds an imperceptible perturbation in images, serving as a watermark while simultaneously disrupting the output of latent diffusion models. Our approach employs a Score Estimator trained on select latent embeddings to embed the watermark by minimizing the score function. We then apply conformal inference to compute p-values for watermark detection. To distort the output of latent diffusion models, we shift watermarked image embeddings away from the distribution mean, distorting unauthorized generations. Experiments demonstrate our framework's superior performance in watermark detection, imperceptibility, and robustness against attacks, offering a comprehensive approach to protect images against latent diffusion models.

1 Introduction

Traditional methods for protecting image copyrights often rely on embedding imperceptible messages as digital watermarks into images (Zhu et al., 2018). These watermarks allow creators to verify ownership by detecting their presence in suspected unauthorized copies. Although effective against direct misuse, such approaches face limitations with the emergence of generative models, particularly diffusion models (Dhariwal & Nichol, 2021).

Generative diffusion models such as DALL · E (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2022) have emerged as powerful tools for generating high-quality synthetic images. By learning the underlying distribution of a given dataset, these models can produce novel images that closely resemble the training data (Song et al., 2020). However, their accessibility raises serious copyright concerns. First, feeding watermarked images into diffusion pipelines (e.g., for editing or style transfer) can degrade or distort embedded watermarks in the output, complicating detection (Mareen et al., 2024). Second, malicious users can exploit diffusion models to generate new images using watermarked images as training data, further complicating copyright protection efforts.

These vulnerabilities threaten creators' rights and creative integrity, for example, having their original works replicated or modified without permission. Traditional watermarking techniques that focus on embedding invisible information within images fail to address this challenge: they neither prevent the generation of new images nor the modification of the original using diffusion models; nor do they ensure watermark robustness against diffusion-based generation processes.

In this work, we propose a novel dual-protection framework designed to address both image watermarking and the prevention of misuse by latent diffusion models (LDM). Our approach introduces an adversarial perturbation that acts as a watermark by leveraging a Score Estimator trained on the latent embeddings generated by an LDM encoder (Esser et al., 2021) (Figure 1). The watermark is made invisible by constraining the perturbation strength to ensure minimal perceptual change. To statistically verify the presence of a watermark, we employ conformal inference (Angelopoulos & Bates, 2021), which calculates p-values to determine whether an image is watermarked and provides a rigorous guarantee for controlling type-1 errors.

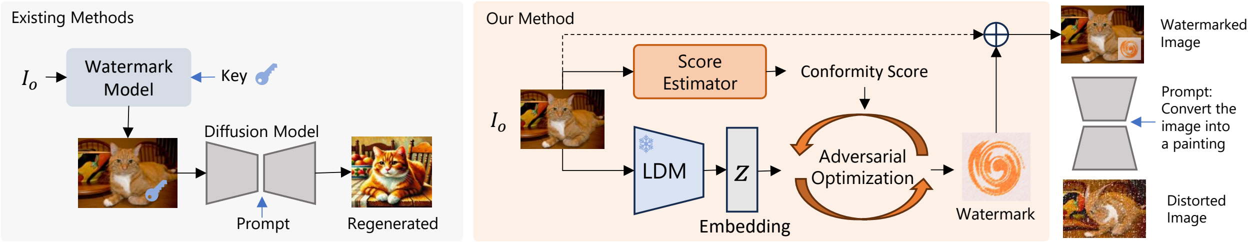


Figure 1: Existing methods that embed a predefined key as watermark cannot prevent malicious users from using these images to re-generate or train a latent diffusion model (LDM). Our method generates unique watermarks using adversarial optimization that minimizes a loss over the latent space and a conformity score. We shift the watermarked image's distribution to a low sampled region of the LDM's latent space which prevents the LDM to further train or sample from it, resulting in distorted outputs. Moreover, we can detect the presence of our watermark using a statistical test satisfying our dual protection goal.

Additionally, the adversarial perturbations are designed to shift the latent embedding of watermarked images away from the mean of the embedding distribution, placing them in low-density regions. As a result, when a LDM tries to use this image as input, it forces the model to generate visibly distorted outputs, ensuring detectable artifacts while maintaining watermarking objectives. The key contributions of our work are as follows:

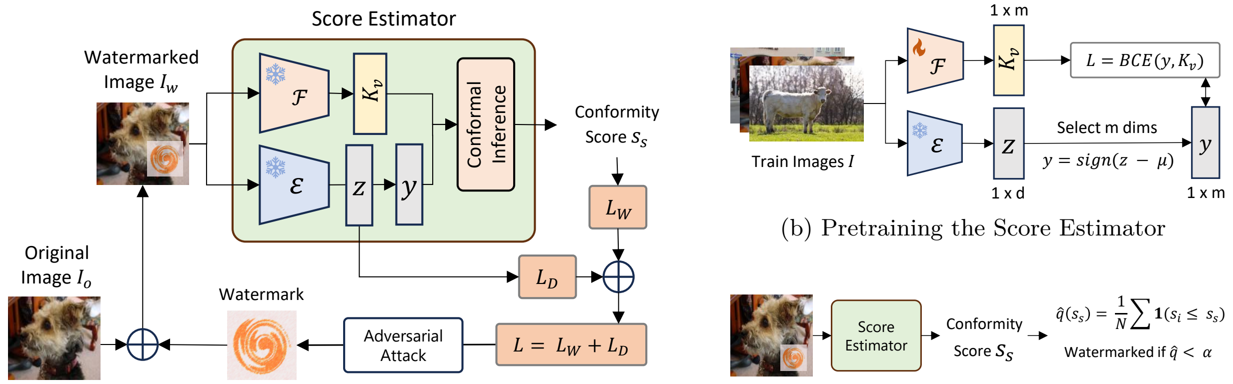
- We propose a dual-protection framework that introduces an invisible perturbation to images, serving as a watermark while simultaneously distorting the output of LDMs.
- Our method leverages conformal inference to calculate p-values for watermark detection, providing a statistically robust approach to identifying watermarked images.
- Unlike previous watermarking techniques, our framework is designed to prevent malicious users from directly claiming ownership of a watermarked image using its embedding (Case 2 of Section 5.7). By utilizing conformal inference, the selection of watermark dimensions remains hidden, making any malicious claim of ownership no better than a random guess. This significantly strengthens the defense against unauthorized ownership assertions.

In summary, our approach provides a comprehensive solution for copyright protection against known or anticipated LDMs. By embedding invisible perturbations that act as both watermarks and deterrents to misuse, we offer a novel mechanism for protecting creative works while ensuring that LDMs cannot be easily exploited to generate unauthorized content.

2 Related Work

2.1 Adversarial Attacks

In computer vision, adversarial examples are subtly altered images designed to manipulate neural network models while remaining nearly imperceptible (Szegedy et al., 2013). In image classification, these examples can cause models to misclassify images. In diffusion models, adversarial attacks introduce undetectable changes that result in distorted outputs, revealing tampering. Salman et al. (2023) identify two types of adversarial attacks on LDMs: encoder and diffusion attacks. Both involve small perturbations, with encoder attacks modifying the encoder's output to resemble a predefined embedding, and diffusion attacks aiming to match the model's output to a target image. The Glaze model protects artists' styles from unauthorized replication by incorporating invisible perturbations that distort styles learned by diffusion models (Shan et al., 2023).



(a) Watermark Generation Pipeline

(b) Pretraining the Score Estimator

(c) Watermark Detection

Figure 2: Pipeline of our proposed method. (a) The input image I_o is passed through a feature embedding model $\mathcal{F}$ to obtain the key vector K_v and a latent diffusion model encoder $\mathcal{E}$ to get the latent space vector z. The label y is calculated using Equation 2. We use conformal inference to generate a score s_s of the key vector. The watermark for the input is a perturbation embedding generated by minimizing the loss over s_s and z. (b) Prior to watermark generation, we pretrain $\mathcal{F}$ using a large set of images in order to generate key vectors which are drawn from the mean distribution of the dataset. (c) To detect whether an image is watermarked or not, we calculate a p-value based on the score. If the value falls below a certain threshold α we can claim that the image is watermarked.

2.2 Image Watermarking

Image watermarking embeds invisible information within images to establish copyright claims (Cox et al., 2007; Urvoy et al., 2014). With the rise of diffusion models capable of producing high-quality images, traditional digital watermarking techniques, like HiDDeN (Zhu et al., 2018) and SSL (Fernandez et al., 2022) face new challenges. Zhu et al. (2018) introduced HiDDeN, an end-to-end CNN framework comprising an encoder, decoder, and adversarial network. Fernandez et al. (2022) employed self-supervised learning networks to embed watermarks in the latent spaces of pre-trained models. Recent work by Wen et al. (2023) proposes Tree-Ring Watermarking, which embeds imperceptible patterns into the initial noise vectors of diffusion models through Fourier-space structuring. Meanwhile Stable Signature Fernandez et al. (2023) fine-tunes the latent decoder of diffusion models to natively embed watermarks during generation. Secret Key Signature (SKS) (Chen et al., 2024) uses adversarial attacks to embed watermarks into images, accompanied by hypothesis tests for detecting watermarks with statistical guarantees.

In this work, we aim to achieve comparable watermarking performance while distorting the outputs of LDMs. Unlike previous watermark-only methods, such as SKS, our approach protects watermarked images from manipulation by LDMs. Furthermore, by incorporating conformal inference, our method resists direct ownership claims—an issue that SKS cannot address.

3 Methods

This section describes our dual-protection strategy – (i) integrate an invisible watermark in images, and (ii) distort the output of LDMs that attempt to utilize these images without authorization.

As shown in Figure 2, first we train a Score Estimator model to predict a conformity score $s_s \in \mathbb{R}$ for an input image I_o. Next, using the trained estimator, we run a joint optimization loop to add a small perturbation to I_o to produce a watermarked image I_w. The joint optimization needs to balance between the strength of the watermark to ensure imperceptibility as well as to change the distribution of the original input so that it becomes distorted when used by an LDM. Specifically, we achieve this by minimizing the following loss function:

$$L(I_o) = \lambda_{\mathrm{D}} L_{\mathrm{D}}(I_o) + \lambda_{\mathrm{W}} L_{\mathrm{W}}(I_o), \tag{1}$$

where L_{D} represents the loss associated with distorting the output of the LDM, and L_{W} represents the loss for embedding the watermark. The parameters λ_{D} and λ_{W} are adjustable weights that control the contribution of each term. We optimize this loss using a modified momentum-based iterative algorithm, MI-FGSM (Dong et al., 2018), adapted with a parameter β_{tg} to bound the perturbation's magnitude. A larger β_{tg} relaxes this constraint, permitting stronger perturbations. The detailed steps of MI-FGSM are provided in the appendix.

In the watermark detection stage, to determine whether a suspected image belongs to the image owner, the image is input into the score estimator. A p-value is calculated via conformal inference and a reference set of calibration images. If the value falls below the confidence level, we can assert that the image is watermarked.

3.1 Watermark Embedding

In this section, we describe the process of embedding a watermark into an image. Specifically, we elaborate on the definition of L_{W} in equation 1.

3.1.1 Score Estimator Model

We first introduce the Score Estimator, which contains a CNN-based feature embedding model $\mathcal{F}$ that learns to generate m dimensional binary key vectors that align with the indices of the latent space of a frozen LDM encoder $\mathcal{E}$. In practice, we find that moderate-capacity backbones provide more suitable uncertainty profiles for conformal inference, whereas overly confident predictors constrain adversarial optimization and degrade watermark imperceptibility. To ensure the watermark is secure, we generate unique labels instead of relying on known or easily accessible labels. To generate the training data, we sample an image I from a large dataset of images and pass it through $\mathcal{E}$ to obtain a latent embedding $z = \mathcal{E}(I) \in \mathbb{R}^d$. We then compare z with the embedding mean μ, which is calculated from the latent embeddings of the dataset. We compute the sign of this difference: $\text{sign}(\mathcal{E}(I) - \mu)$ and randomly select m dimensions in this sign vector as our m-dimensional binary label $y \in \{-1, 1\}^m$, i.e,

$$y = \text{sign}(\mathcal{E}(I) - \mu)_{i_1, i_2, \ldots, i_m}, \tag{2}$$

where $i_1, i_2, \ldots, i_m$ are randomly selected dimensions chosen by the image owner. The same set of m dimensions should be used consistently across all images for both training and watermarking. Since the total number of dimensions z is large, it is computationally infeasible for malicious users to identify the chosen subset selected by the user, thereby protecting the watermark's privacy.

Next, we train $\mathcal{F}$ to output the m-dimensional vector. A sigmoid activation function is applied to the model output to produce probabilities for each dimension. We train this predictor model using binary cross-entropy loss for multi-label classification on a large image dataset with our generated keys, shown in Figure 2b. We refer to this trained model as the Score Estimator. During watermark generation and detection, we calculate conformity scores using the the multi-label conformal prediction method (Cauchois et al., 2021) utilizing both the generated keys and the latent space labels.

3.1.2 Multilabel Classification Conformal Inference

We use the multilabel classification conformal inference method proposed by Cauchois et al. (2021), which forms the foundation of our watermark approach. Conformal inference is a statistical framework designed to provide valid confidence levels for predictions (Tibshirani, 2023). In the context of multilabel classification, it computes a conformity score for each label and constructs a prediction set that contains the true set of labels with a predefined confidence level. Given an image $I \in \mathcal{I}$ and its corresponding k-th label $y_k \in \{-1, 1\}$, our objective is to compute an overall conformity score for the image by considering dependencies between multiple labels. In Cauchois et al. (2021), the authors propose building a tree structure to capture these dependencies and then compute the conformity score based on it.

We begin by defining two factors for modeling label dependencies: interaction factors and marginal factors. The interaction factors $\psi : \{-1, 1\}^2 \to \mathbb{R}^4$ capture pairwise interactions between labels. Specifically, ψ is defined as:

$$\psi(-1, -1) = e_1,\ \psi(1, -1) = e_2,\ \psi(-1, 1) = e_3,\ \psi(1, 1) = e_4 \tag{3}$$

where e_1, e_2, e_3, e_4 are the standard basis vectors of $\mathbb{R}^4$. Meanwhile, marginal factors $\phi_k : \{-1, 1\} \times \mathcal{I} \to \mathbb{R}^2$ describe how the individual labels relate to I. The marginal factor for the k-th label is:

$$\phi_k(y_k, I) := \frac{1}{2} \begin{pmatrix} (y_k - 1) \cdot s_k(I) \\ (y_k + 1) \cdot s_k(I) \end{pmatrix} \tag{4}$$

where $s_k(\cdot)$ is the score function for the k-th label.

To model the dependencies among labels, we employ a tree-structured graphical model (Chow & Liu, 1968). For a tree $\mathcal{T} = ([K], E)$, the joint probability of the labels is modeled as:

$$p_{\mathcal{T},\alpha,\beta}(y \mid I) \propto \exp\left(\sum_{e=(k,l)\in E} \beta_e^T \psi(y_k, y_l) + \sum_{k=1}^{K} \alpha_k^T \phi_k(y_k, I) \right) \tag{5}$$

where α and β are parameters that describe the interaction and marginal contributions, respectively. Specifically, $\alpha_k \in \mathbb{R}^2$ for each label k, and $\beta_e \in \mathbb{R}^4$ for each edge $e \in E$.

The tree structure $\mathcal{T}$ that best represents the dependencies between labels is learned by maximizing the log-likelihood of the observed training data. Given a training dataset $\mathcal{D}_{\text{train}} = \{(I^{(i)}, y^{(i)})\}_{i=1}^{N_{\text{train}}}$, we optimize

$$\hat{\mathcal{T}}, \hat{\alpha}, \hat{\beta} = \arg\max_{\mathcal{T},\alpha,\beta} \sum_{i=1}^{N_{\text{train}}} \log p_{\mathcal{T},\alpha,\beta}(y^{(i)} \mid I^{(i)}) \tag{6}$$

To estimate the dependencies between labels, we compute the empirical mutual information between each pair of labels using single-edge trees. The optimal tree structure is then identified by solving for the maximum spanning tree based on the mutual information values (Chow & Liu, 1968). Once the tree structure and parameters are learned, we define a scoring function $s(I, y)$ for an image I and its label set y. The score is given by:

$$s_{\hat{\mathcal{T}},\hat{\alpha},\hat{\beta}}(I, y) := \sum_{e=(k,l)\in \hat{E}} \hat{\beta}_e^T \psi(y_k, y_l) + \sum_{k=1}^{K} \hat{\alpha}_k^T \phi_k(y_k, I) \tag{7}$$

3.1.3 Watermark Embedding Loss

Using the trained key generator $\mathcal{F}$ and the latent space labels y, we create a watermark via conformal inference. We start by defining the scoring function between the k-th output $\mathcal{F}(I)_k$ of the model and the k-th label y_k as:

$$s_k(I, y) = -|\mathcal{F}(I)_k - y_k|. \tag{8}$$

Using the multi-label conformal prediction method, we estimate the edge empirical mutual information for every pair of nodes in the m dimensions selected, and construct a tree-structured score function $\hat{s}(I, y)$ based on the Chow-Liu-type approximate maximum likelihood tree.

Next, we compute conformity scores for a calibration set of images by passing them through $\mathcal{F}$ to generate keys and evaluating $\hat{s}(I, y)$. Using these scores, we determine an empirical critical value s_{cv}. An image is classified as watermarked if its conformity score satisfies $\hat{s}(I, y) < s_{\text{cv}}$. To enforce watermarking, we optimize the original image I_o to minimize its score such that $\hat{s}(I_o, y) \leq s_{\text{cv}}$, yielding the watermarked image I_w.

In practice, when minimizing the scoring function $\hat{s}(I, y)$ with respect to I, we must account for the gradient passing through y: $\frac{\partial \hat{s}}{\partial y} \frac{\partial y}{\partial I}$. However, since y is discrete, $\frac{\partial y}{\partial I}$ is zero. To address this, we "soften" y using the sigmoid function σ, such that $\tilde{y} = \sigma(y)$. Therefore, the loss function for watermark embedding is:

$$L_{\text{W}}(I_o) = \hat{s}(I_o, \tilde{y}). \tag{9}$$

3.2 Distorting Latent Diffusion Models

The last part of our objective function is the distortion loss L_{D}, designed to distort the latent diffusion model's output. Given I_o we want to shift its latent embedding $\mathcal{E}(I_o)$ into low-probability regions of the latent space. These regions correspond to under-sampled patterns during the diffusion model's training. By doing so, we force the diffusion process to operate outside its learned manifold, inducing higher denoising errors and distortions in generated outputs.

As demonstrated later in the appendix, the distribution of image embeddings z within the latent space $\mathcal{Z}$ follows a Gaussian distribution $f(z)$, characterized by a mean vector $\mu \in \mathbb{R}^n$ and a covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$. Given I_o, our objective is to create a watermarked image I_w by minimizing the log-likelihood of the image embedding z_w. We minimize the following loss with respect to I_o,

$$L_{\mathrm{D}}(I_o) = -\frac{1}{2}(\mathcal{E}(I_o) - \mu)^\top \Sigma^{-1}(\mathcal{E}(I_o) - \mu). \tag{10}$$

4 Watermark Detection

We use two hypotheses to detect whether a suspected image I_s is watermarked or not: H_0 (the null hypothesis) states that I_s is not watermarked, and H_1 (the alternative hypothesis) states that I_s is watermarked. To test for the presence of a watermark, we assume access to a calibration dataset $\mathcal{D}_{\mathrm{cal}} = \{(I_i, y_i)\}_{i=1}^N$, where I_i is the i-th image, y_i the corresponding label calculated using equation 2, and N is the total number of images in the calibration dataset. This calibration dataset is distinct from the one used in Section 3.1.3 to maintain statistical validity of the hypothesis test. For each calibration image I_i, we compute its conformity score $s_i = \hat{s}(I_i, y_i)$ and sort these scores in ascending order. Using the same feature dimensions $\{i_1, i_2, \ldots, i_m\}$ selected during watermark embedding, we calculate the conformity score of the suspected image as $s_{\mathrm{s}} = \hat{s}(I_s, y_s)$.

The empirical critical value is then calculated as $s_{\mathrm{cv}} = s_{(\lceil (N+1)\alpha \rceil)}$, where $\lceil\cdot\rceil$ denotes the ceiling function and α is the desired significance level. We reject H_0 if $s_{\mathrm{s}} < s_{\mathrm{cv}}$, indicating the presence of a watermark. We compute the p-value of the suspected image under hypothesis testing as:

$$\hat{q}(s_{\mathrm{s}}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(s_i \leq s_{\mathrm{s}}) \tag{11}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

5 Experiment

We evaluate our method in six aspects: detection performance, imperceptibility, distortion analysis, generalization, robustness, and security.

5.1 Experiment Setup

Our experiments use the MSCOCO 2017 dataset (Lin et al., 2014) containing 118k training, 41k test, and 5k validation images. We adopted the VGG16 model (Simonyan, 2014) as the backbone for our Score Estimator to generate m-dimensional key vectors. While stronger backbones such as ResNet (He et al., 2016) achieve higher classification accuracy, we empirically found that moderate-capacity models yield more favorable uncertainty characteristics for conformal inference–based watermark embedding.

Specifically, overly confident predictors reduce uncertainty margins, which restricts adversarial optimization and makes it difficult to simultaneously achieve low conformity scores and imperceptibility. In contrast, VGG16 provides a better balance between predictive sensitivity and calibrated uncertainty, enabling effective watermark embedding without excessive visual distortion. We provide a qualitative comparison with a ResNet-101 backbone in the appendix.

We use the encoder of Stable Diffusion as $\mathcal{E}$ to generate latent embeddings and train the VGG model using binary cross-entropy loss (see equation 1). Detailed training configurations are provided in the appendix.

We optimize equation 1 using the Momentum Iterative Fast Gradient Sign Method (MI-FGSM) (Dong et al., 2018), evaluating performance in three aspects: (1) detection rate: measured as the percentage of

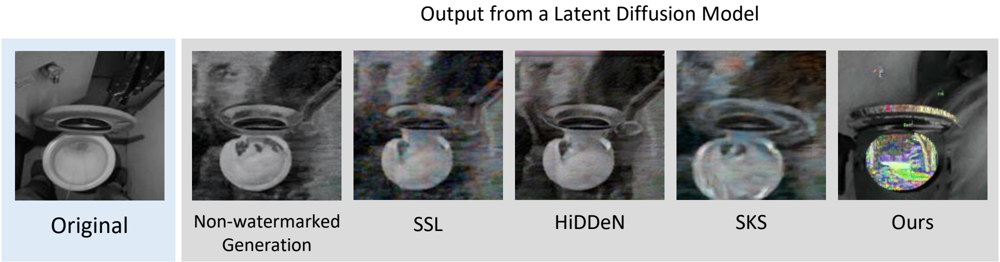

Output from a Latent Diffusion Model
Original
Non-watermarked Generation
SSL
HiDDeN
SKS
Ours

Figure 3: Comparison of the original image with generated outputs using different watermarked images as input to a LDM with the prompt - *Generate an image in the impressionism style of the original image.* Our method produces the highest distortion in the generated result.

watermarked images with p-values smaller than significance level; (2) image quality, quantified using PSNR, SSIM (Wang et al., 2004), MAE, and RMSE; and (3) distortion, evaluated via FID scores between latent diffusion outputs and reference WikiArt images (Tan et al., 2019).

We compare against three adapted baselines: SSL (Fernandez et al., 2022) in zero-bit mode ($\alpha = 0.05$), HiDDeN (Zhu et al., 2018) with a bit error rate threshold of 0.05, and SKS (Chen et al., 2024). All methods are modified by adding L_{D} loss to the original watermark loss, and use identical MSCOCO training/validation splits for fair comparison. For methods such as Stable Signature (Fernandez et al., 2023), which employ in-diffusion watermarking, watermarks are embedded into outputs of diffusion models. In contrast, our method embeds watermarks into images themselves, making a direct comparison of watermark properties unsuitable. Additionally, metrics like PSNR are computed based on the diffusion model's output. Since one of our primary objectives is to distort the output of LDMs, a direct comparison with these methods is not feasible.

Our experiments demonstrate that at equivalent image quality levels, our method achieves comparable detection rates (99.94%) while inducing significantly stronger diffusion model distortion (FID 108.65 vs 72.68˘92.35). Watermark embedding requires 200 optimization iterations and approximately 16 seconds per image on a single NVIDIA H100 GPU, while detection incurs negligible overhead.

5.2 Detection Performance Analysis

We evaluated the performance of our watermark detection method by analyzing the p-values generated for all watermarked images. We compute the mean and standard deviation of these p-values and record the percentage of p-values below significance levels of 0.05 and 0.01. To assess the false positive rate, we compared these results with clean images. As shown in Table 1, our method achieves near-perfect detection on watermarked images (> 99%), with low mean p-values. The results on clean images confirms a false positive rate consistent with the chosen significance levels.

Table 1: Watermark detection performance.The table shows the mean and standard deviation (Std) of the detector's p-values, and the percentage of p-values below the significance levels $\alpha = 0.05$ and 0.01 for both watermarked and clean images.

	Mean	Std	<0.05	<0.01
Clean	0.4950	0.2871	4.6%	0.92%
Watermarked	0.0013	0.0039	99.94%	99.52%

5.3 Distortion Analysis

To measure the level of distortion introduced by different methods we used 5,000 images from the MSCOCO validation set and generate watermarked images using different models. Next we input these watermarked images to the LDM with the prompt *Generate an image in the impressionism style of the original image.*

We then compared the generated images with authentic Impressionist-style images from the WikiArt dataset (Tan et al., 2019), and compute FID as a measure of distortion.

This addresses a critical challenge in image protection: preventing diffusion models from learning to replicate the styles of artists. A higher FID score indicates greater distortion in the generated images, suggesting stronger protection against unauthorized generation. As shown in Table 2, our method introduces more distortion compared to other methods, providing better protection. Figure 3 visually compares the original image with various generated images. While unwatermarked inputs yield high-quality impressionist outputs that closely follow the text prompts, images watermarked with our method produce outputs with noticeable artifacts and blurry segments in the LDM generations. Additional visual results and analysis across other art styles are available in the appendix.

Table 2: FID comparison of the distortion introduced in LDM outputs across different methods. Higher FID indicates more distortion, offering better protection.

Method	Orig.	SSL	HiDDeN	SKS	Ours
FID↑	66.97	72.68	92.35	80.11	**108.65**

Table 3: FID comparison between original and SD2-watermarked images after processing by different models.

Model	Original	Watermarked
SDXL	117.54	122.69
DiffEdit	99.41	94.43

To complement FID, we additionally evaluate *instruction adherence* using CLIP similarity between the text prompt and generated images. The lower CLIP score achieved by our method indicates weaker semantic alignment between the generated images and the intended prompt, supporting the claim that diffusion-based editing and instruction following are meaningfully disrupted.

Method	HiDDeN	SSL	SKS	Ours
CLIP score ↓	0.2624	0.2641	0.2724	**0.2581**

5.4 Imperceptibility Analysis

We assess the imperceptibility of the watermarks by comparing image quality of watermarked images relative to the original. The HiDDeN model achieved a PSNR of 33.56, and for a fair comparison, we adjusted SSL and SKS to produce a similar PSNR of 33. Despite superior distortion effects, our method maintains better image quality across all metrics, as shown in Table 4. Additionally, by increasing β_{tg} to allow greater perturbation in watermarked images, we matched the baseline PSNR (32) while achieving a FID of 150.65 and a 99.74% detection rate at $\alpha = 0.05$. This demonstrates that increasing perturbation strength enhances the ability to distort LDMs effectively. Visual examples of watermarked images and the perturbation is shown in Figure 4.

Table 4: Comparison of watermark imperceptibility across different methods.

Method	PSNR↑	SSIM↑	MAE↓	RMSE↓
HiDDeN	32.72	0.9214	0.0242	0.0428
SSL	33.58	0.9408	0.0168	0.0222
SKS	32.01	0.9405	0.0165	0.0251
Ours	**37.30**	**0.9412**	**0.0109**	**0.0195**
Ours (32)	32.92	0.9147	0.0200	0.0278

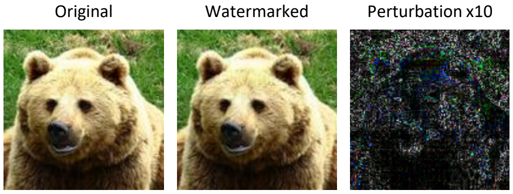
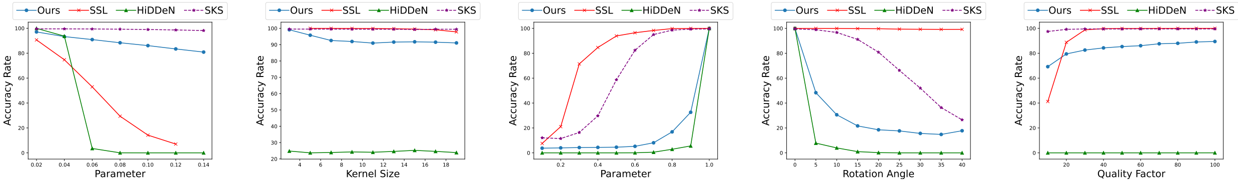


Figure 4: Visual comparison of original images, watermarked images, and 10× amplified structural differences.

5.5 Generalization Analysis

We evaluate the generalization of our method across different LDMs by addressing two questions:

Case 1: Can watermarked images trained on one diffusion model withstand attacks from another? We trained our Score Estimator module on Stable Diffusion 2 (SD2) (Ramesh et al., 2022) and



(a) Gaussian Noise (b) Gaussian Blur (c) Cropping (d) Rotation (e) JPEG Comp.

Figure 5: Comparison of the robustness of different watermarking methods under various image perturbations: Gaussian Noise, Gaussian Blur, Cropping, Rotation, and JPEG Compression. Our method shows strong robustness under Gaussian Noise and JPEG Compression while being outperformed by SSL under some spatial transformations. Notably, our method achieves the highest resilience against diffusion model editing.

input the watermarked images into Stable Diffusion XL (SDXL) (Podell et al., 2023) and DiffEdit (Couairon et al., 2022). We then compared the FID scores of their outputs against those of the original images, as shown in Table 3. The results indicate that watermarked images trained for SD2 do not effectively distort other models.This is because the watermark relies on the latent encoder, and different LDMs use different encoders, limiting cross-model generalization.

Case 2: Can the entire watermark pipeline transfer to another diffusion model? To test this, we retrained the Score Estimator for InstructPix2Pix (Brooks et al., 2023) and used it to generate watermarked images. We then evaluated the watermark detection rate, image quality, and FID (Table 5). Our method achieved a 98.84% detection rate at $\alpha = 0.05$ while maintaining superior image quality compared to baselines (Table 4). Furthermore, it significantly distorted outputs of InstructPix2Pix model, as shown in the increased FID score.

Table 5: Image quality and FID comparison for watermark generated using InstructPix2Pix encoder.

Method	PSNR	SSIM	MAE	RMSE
InstructPix2Pix	36.38	0.9602	0.0119	0.0203

Method	Original	Watermarked
FID	96.73	132.79

In conclusion, watermarked images trained on one diffusion model do not effectively resist attacks from another model. However, our method generalizes well to different diffusion models when trained from scratch, demonstrating adaptability and robustness across architectures.

5.6 Robustness Analysis

To evaluate the robustness of our watermarking method, we tested its performance under a variety of image perturbations commonly used in real-world scenarios. Specifically, we applied Gaussian noise, Gaussian blur, cropping, rotation, and JPEG compression to the watermarked images, and then measured the watermark detection rate. As shown in Figure 5, our method demonstrates strong robustness under Gaussian noise, outperforming both HiDDeN and SSL. Besides, it performs better than HiDDeN across all other perturbations. However, SSL and SKS surpass our method in Gaussian blur, cropping, rotation, and JPEG compression. We attribute this difference to their higher watermark embedding dimensionality (2048 dimensions for SSL, 32 for SKS, and 6 for our method), which allow them to better withstand spatial transformations.

Note that we retrain the baselines with the distortion loss L_{D}. This loss encourages the watermark to also resist LDM-based editing and, to some extent, reduces robustness to image perturbations. This suggests that simply combining standard watermark loss with distortion loss is not a good solution to achieve both aims.

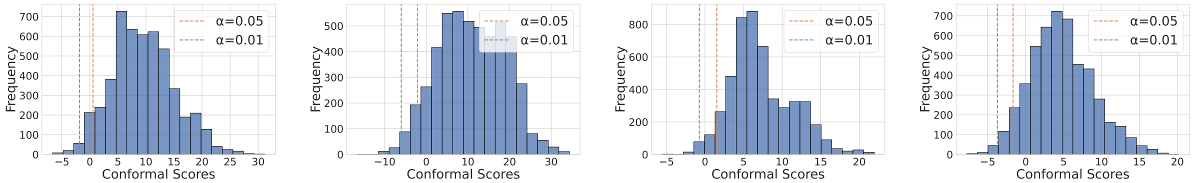


Figure 6: Distribution of conformity scores from four different sets of randomly selected dimensions. Since the conformity scores of randomly selected dimensions typically follow a nearly Gaussian distribution, it is challenging for Bob to directly train a Score Estimator model to claim ownership of Alice's image.

5.7 Security Analysis

We evaluated the security of our watermarking framework by simulating attacks in three scenarios involving two users: Alice, the owner of the image, and Bob, a malicious user attempting to claim ownership. We compared the performance with other 0-bit watermarking methods: SSL and SKS (Chen et al., 2024). For Cases 1 and 3, we took the results from Chen et al. (2024) without adding the distortion loss L_{D}.

Case 1: Fake Watermark Generation Bob attempts to generate fake watermarks on clean images, hoping to bypass Alice's watermark detection. To test this, we randomly selected m dimensions from the latent embedding z, trained a corresponding Score Estimator, and calculated the parameters for the conformity score in equation 7. We then watermarked the image using the loss function described in equation 1. Afterward, we evaluated the watermark detection rate using Alice's model to determine the effectiveness of Bob's attack. Our method suffers from a higher detection rate than SKS, though it is still lower compared to SSL. Notably, our method achieves a baseline detection rate comparable to SKS at a p-value threshold of 0.01. When applying the same threshold to evaluate Bob's attack, our detection rate improves to 1.88%—surpassing SKS's performance.

Table 6: Detection rate for fake watermark generation attacks across different methods. The values in parentheses for "Ours" represent the detection thresholds used (0.05 and 0.01).

Method	SSL	SKS	Ours(0.05)	Ours(0.01)
Detection rate	12.84%	1.94%	7.64%	1.88%

Case 2: Direct Ownership Claim Bob tries to claim ownership of Alice's watermarked image without modifying it. To simulate this, we trained four Score Estimators with different randomly selected dimensions. Using the MSCOCO 2017 test set as a calibration set, we examined the conformity score distribution to determine whether Bob could achieve a p-value below 0.05 by selecting dimensions at random. Since the conformity score distribution concentrates around the mean, without using adversarial attacks, Bob is unlikely to generate a valid watermark with a p-value below 0.05. Figure 6 presents the distribution of the conformity scores for the four Score Estimators. In this scenario, SKS is vulnerable: if Bob simply takes Alice's watermarked image and uses his SKN to generate a signature, he could falsely claim ownership. However, both SSL and our proposed method are resistant to this type of attack.

Case 3: Watermark Removal and Replacement Bob attempts to remove Alice's watermark by embedding his own watermark into the image. To test this, we embedded a watermark into an image (acting as Alice's watermark), then added another four watermarks as Bob's attack. We then checked whether Alice's watermark could still be detected after Bob's attack. The results in Table 7 show the detection rate of Alice's original watermark as more Bob's watermarks are added. SKS maintained a high detection rate (above 89%). Our method also showed resilience, with a detection rate of 84.62% after three additional watermarks, outperforming SSL. SSL's detection rate dropped significantly to 19.5% after three watermarks. Although our method does not perform as well as SKS, this difference may be attributed to the additional loss function L_{D}, which affects all pixels, whereas the watermark loss L_{W} only affects a portion of the pixels (approximately 20%). Balancing watermark detection and distortion introduces a trade-off, which may explain the performance.

Table 7: Robustness test against multiple overlapping watermark embeddings. Detection rate of Alice's original watermark after embedding Bob's additional watermarks.

	1	2	3	4	5
SSL	92.62%	60.50%	27.00%	19.50%	12.00%
SKS	99.50%	98.60%	96.10%	89.70%	89.70%
Ours	99.94%	96.52%	92.80%	84.62%	75.46%

6 Ablation Study

Effect of Dimensionality in Feature Selection: We analyze how the number of selected feature dimensions (m) impacts system performance. Compared to the baseline configuration with $m = 6$ dimensions (FID = 108.65), reducing the dimensions to $m = 4$ yields an FID score of 69.20, indicating weaker protection against unauthorized generation. The mean p-value for watermark detection is 0.0085, with 96.90% of p-values below 0.05 and 93.66% below 0.01, showing reduced detection robustness.

Effect of Removing the Loss Component L_{D}: To evaluate the effect of removing the distortion loss L_{D}, we set $\lambda_{\mathrm{SD}} = 0$ in our optimization objective. Without L_{D}, our method achieved an FID score of 68.45 compared to 66.97 for the original images. The output of the LDM was only slightly more distorted, as expected. Surprisingly, the mean p-value for watermark detection was 0.0030, with 99.24% of p-values below 0.05 and 97.66% below 0.01. The detection rate performance was worse than when we included L_{D}. We hypothesize this stems from the non-convex optimization landscape in equation 1. Adding L_{D}, which acts as a small perturbation to the optimization, helps escape from local optima, as observed in the experiment.

7 Conclusion

In this work, we propose a dual-protection framework to safeguard image copyrights against latent diffusion models. Our method embeds a perturbation into the image that serves both as a watermark and a mechanism for disrupting the outputs of latent diffusion models. Additionally, we leverage conformal inference to develop a statistically robust approach for detecting watermarked images. Experimental results demonstrate that our method ensures strong watermark detection, enhanced imperceptibility, and resilience against various image perturbations. Furthermore, we show that the entire pipeline generalizes across attacks involving different diffusion models. Notably, our approach resists direct ownership claims and multiple watermark embeddings, showing its potential as a reliable solution for protecting image copyrights in the era of generative AI.

References

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Maxime Cauchois, Suyash Gupta, and John C Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *Journal of machine learning research*, 22(81):1–42, 2021.

F Chen, W Lin, Z Liu, A Chan, et al. A secure image watermarking framework with statistical guarantees via adversarial attacks on secret key networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

CKCN Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

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)*, pp. 3054–3058. IEEE, 2022.

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*, pp. 22466–22477, 2023.

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

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*, pp. 740–755. Springer, 2014.

Hannes Mareen, Kobe De Meulenaere, Peter Lambert, and Glenn Van Wallendael. Diffusion denoising watermark removal models to attack invisible image watermarks. In *2024 17th International Conference on Signal Processing and Communication System (ICSPCS)*, pp. 1–6. IEEE, 2024.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

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*, pp. 10684–10695, 2022.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023.

Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, and Kiyoshi Tanaka. Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1):394–409, 2019. doi: 10.1109/TIP.2018.2866698. URL `https://doi.org/10.1109/TIP.2018.2866698`.

R Tibshirani. Conformal prediction: Advanced topics in statistical learning(lecture note). 2023.

Matthieu Urvoy, Dalila Goudia, and Florent Autrusseau. Perceptual dft watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security*, 9(7):1108–1119, 2014.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *Advances in neural information processing systems*, 2023.

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)*, pp. 657–672, 2018.

A Appendix

B Smoothing Function for Conformal Inference

In the multi-label conformal inference method introduced by Cauchois et al. (2021), the function ψ is defined in a discrete manner. Here, we propose a smoothing technique for this function. Specifically, we smooth the function $\psi(y_k, y_l)$ used in the multi-label conformal prediction set by defining:

$$\psi_c(y_k, y_l) = (1-\pi_k)(1-\pi_l)\cdot e_1 + (1-\pi_k)\pi_l \cdot e_3 + \pi_k(1-\pi_l)\cdot e_2 + \pi_k \pi_l \cdot e_4 \tag{12}$$

where π_k and π_l are smoothing parameters, and e_1, e_2, e_3, and e_4 are the standard basis vectors.

Note that ψ_c is equivalent to ψ when $y_k, y_l \in \{-1, +1\}$. The corresponding loss function for the smoothed conformal prediction set is defined as:

$$L_c = \sum_{(k,l)\in\hat{E}} \beta_e^T \psi_c(y_k, y_l) + \sum_{k=1}^{K} \alpha_k^T \phi_k(y_k, x) \tag{13}$$

where $\hat{E}$ denotes the set of edges in the maximum spanning tree, β_e represents edge-specific parameters, and $\phi_k(y_k, x)$ is the marginal factor as defined in the aforementioned paper.

C MI-FGSM

To minimize the loss defined in equation 1, we employed an approach inspired by the MI-FGSM (Dong et al., 2018). For completeness, we outline the key aspects of this technique here. MI-FGSM is a widely used method for generating adversarial examples by iteratively adjusting the perturbation η added to the input data (Dong et al., 2018). This adjustment aims to minimize the adversarial loss while ensuring that the perturbations remain minimal and imperceptible to the human eye. The update rule for the perturbation at each iteration t is expressed as:

$$g_{t+1} = \mu g_t + \frac{\nabla_I L(f(I+\eta_t))}{\|\nabla_I L(f(I+\eta_t))\|_1} \tag{14}$$

$$\eta_{t+1} = \eta_t - \alpha \cdot \mathrm{sign}(g_{t+1}) \tag{15}$$

In this equation, $L(f(I))$ represents the adversarial loss function, where $f(\cdot)$ is a deep neural network model, and I is the input image. g_t is the accumulated velocity vector in the gradient direction, initialized as $g_0 = 0$, and μ is the decay factor. The gradient $\nabla_I L$ is computed with respect to the input image I. To ensure the imperceptibility of the perturbations, a constraint on the magnitude of η is typically enforced, such that $\|\eta_{t+1}\|_\infty < \epsilon$, where ϵ is a small threshold.

We made several modifications to MI-FGSM inspired by Chen et al. (2024) to better suit our specific problem:

1. **Direct Gradient Application:** Instead of using the sign of the gradient, we directly apply the gradient values to update the perturbation η, enhancing control over the perturbation process.

2. **Scaling Factor Introduction:** We introduced a scaling factor β to expand the constraint $\|\eta_{t+1}\|_\infty < \epsilon$. The perturbation update is modified as follows:

$$\eta'_{t+1} = \beta \cdot \eta_{t+1} \tag{16}$$

where β is determined by the following formula:

$$\beta = \text{clip}\left(\sqrt{\frac{\beta_{\text{tg}}}{\text{mean}(\eta_{t+1}^2)}}, 0, 1\right) \tag{17}$$

Here, β_{tg} is a target value similar to α, serving to set an upper bound on the perturbation magnitude.

3. **Data Augmentation:** Before being input into the model $f(\cdot)$, the image I undergoes data augmentation techniques, such as rotation and cropping, to improve the robustness of watermark detection. The updated rule becomes:

$$\eta_{t+1} = \eta_t - \alpha \cdot \nabla_I L(f(\text{da}(I)), y_{\text{target}}) \tag{18}$$

where $\text{da}(\cdot)$ denotes the data augmentation module.

4. **Adaptive Parameters:** Instead of using fixed parameters, we employ adaptive values for β_{tg} and λ_{W}, allowing for a more flexible approach that enhances the success rate of watermark embedding.

These modifications to MI-FGSM make it more effective and suitable for our watermarking task, ensuring imperceptibility while achieving high watermark detection success.

D Assessing the Normality of Latent Space Embeddings

A critical assumption underlying our adversarial attack strategy is that the embeddings in the latent space follow a Gaussian distribution. To validate this assumption, we processed 118,000 images from the MSCOCO training set through the latent diffusion model encoder and analyzed the resulting embeddings using the Henze-Zirkler test for multi-dimensional normality. The test yielded a statistic of 0.0503, with a p-value of 1.0, allowing us to accept the null hypothesis that the latent space embeddings follow a Gaussian distribution. Figure 7 illustrates the distribution of one dimension of these embeddings.

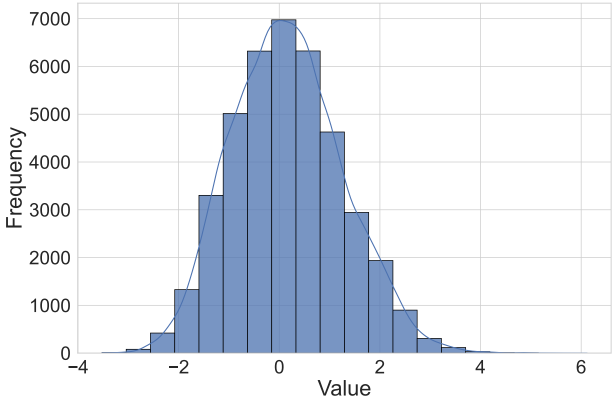


Figure 7: Distribution of one dimension of the embeddings from the latent diffusion model encoder.

E Experiment Details

In our setting, we select the number of features for the watermark as $m = 6$. The step size is set to $\alpha = 0.1$, and the decay factor $\mu = 0.9$. For training, we use $\lambda_{\text{SD}} = 1$ and $\lambda_{\text{W}} = 1.6$. The scaling factor is initialized with $\beta_{\text{tg}} = 8 \times 10^{-4}$. For the adaptive parameters in MI-FGSM, after 200 iterations, we multiply λ_{W} by 30 and β_{tg} by 10. Additionally, after every 50 subsequent iterations, both parameters are multiplied by 3.

For training the VGG model, we use a learning rate of 1×10^{-4}, weight decay of 1×10^{-4}, batch size of 64, and apply a learning rate decay of 0.8 every 10 steps.

When training the conformality score Chow-Liu tree parameter, we use 5,000 images from the COCO test dataset. We then use an additional 5,000 images from the COCO validation dataset as the calibration set for computing the p-values.

F Watermarking Algorithm Details

Our proposed watermarking framework consists of three key components: (1) score estimator pretraining, (2) watermark embedding through adversarial perturbation, and (3) watermark detection. The complete pseudocode is provided in Algorithms 1-3.

Algorithm 1 Score Estimator Pretraining

```
Require: Training images $I$, encoder $\mathcal{E}$, mean vector $\mu$, selected indices $\{i_1, \ldots, i_m\}$
Ensure: Trained CNN network $\mathcal{F}$
for each image $I$ in dataset do
    Compute binary label: $y \leftarrow \{\text{sign}(\mathcal{E}(I) - \mu)\}_{\{i_1,\ldots,i_m\}}$
end for
Initialize CNN network $\mathcal{F}$ with random weights
while not converged do
    Predict $K_v \leftarrow \mathcal{F}(I)$
    Compute loss $L \leftarrow \text{BCE}(y, K_v)$
    Update $\mathcal{F}$ using gradient descent
end while
```

Algorithm 2 Watermark Embedding

```
Require: Original image $I_o$, encoder $\mathcal{E}$, mean $\mu$, covariance $\Sigma$, iterations $T$, loss weights $\lambda_D, \lambda_W$
Ensure: Watermarked image $I_w$
Initialize $I_w \leftarrow I_o$
for $t = 1$ to $T$ do
    Compute distortion loss: $L_D \leftarrow -\frac{1}{2}(\mathcal{E}(I_w) - \mu)^\top \Sigma^{-1}(\mathcal{E}(I_w) - \mu)$
    Generate binary label: $y \leftarrow \{\text{sign}(\mathcal{E}(I_w) - \mu)\}_{\{i_1,\ldots,i_m\}}$
    Compute softened label: $\tilde{y} \leftarrow \sigma(y)$                ▷ Sigmoid activation
    Calculate watermark loss: $L_W \leftarrow S_\theta(I_w, \tilde{y})$
    Total loss: $L \leftarrow \lambda_D L_D + \lambda_W L_W$
    Compute perturbation $\eta_t$ using MI-FGSM: $\eta_t \leftarrow \text{MI-FGSM}(\nabla_{I_w} L)$
    Update image: $I_w \leftarrow \text{Clip}(I_w + \eta_t)$
end for
```

G Watermark Visualization

We provide more visualizations of the watermark to illustrate its characteristics. As shown in Figure 8, the watermark embedded in the images exhibits a strong correlation with the image content itself, ensuring seamless integration while maintaining imperceptibility.

We also include visual comparisons of the 10× amplified watermark deltas between original and watermarked images for both baseline methods and our proposed approach in Figure 9.

Algorithm 3 Watermark Detection

```
Require: Suspected image $I_s$, significance level $\alpha$, calibration distribution scores $\{s_i\}_{i=1}^{N}$
Ensure: Detection decision
Retrieve $y$ from training phase
Compute test statistic: $s_s \leftarrow S_\theta(I_s, y)$
Calculate empirical p-value: $\hat{q}(s_s) \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(s_i \leq s_s)$
if $\hat{q}(s_s) < \alpha$ then
    return “Reject $H_0$ (Watermarked)”
else
    return “Retain $H_0$ (Not watermarked)”
end if
```

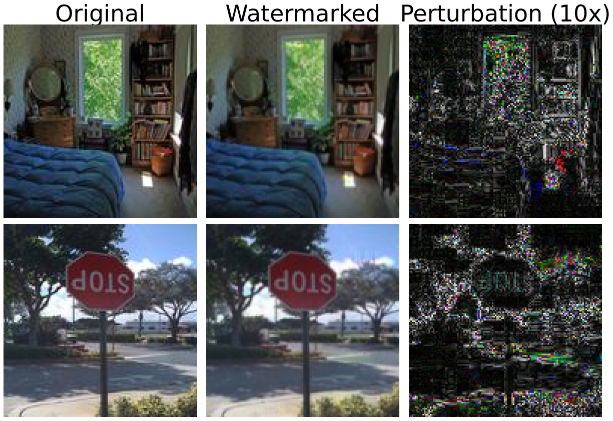


Figure 8: Visual comparison of original images, watermarked images, and 10× amplified structural differences.

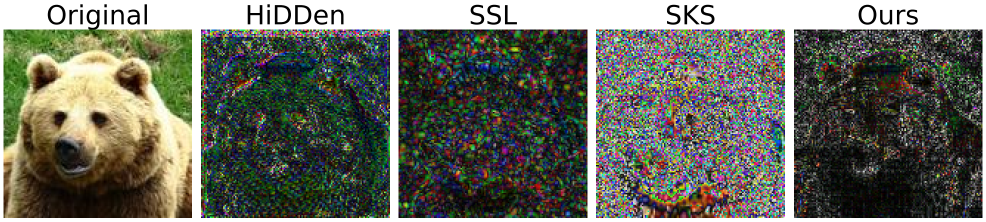


Figure 9: Visual comparison of watermark patterns across different methods. (Left) Original image; (Right) Corresponding 10× amplified watermark deltas.

H Additional Distortion Analysis Results

We provide additional results comparing the distortion introduced by different watermarking methods in the output of the latent diffusion model with the prompt *Generate an image in the impressionism style of the original image*. Figure 10 to Figure 14 illustrate the generated images for various watermarking techniques.

We also performed an evaluation on generating Expressionism-style outputs and obtained an FID score of 162.16 using our method, compared to 95.09–105.60 for the baselines, indicating the effectiveness of our method across different artistic styles. A corresponding visualization is shown in Figure 15.

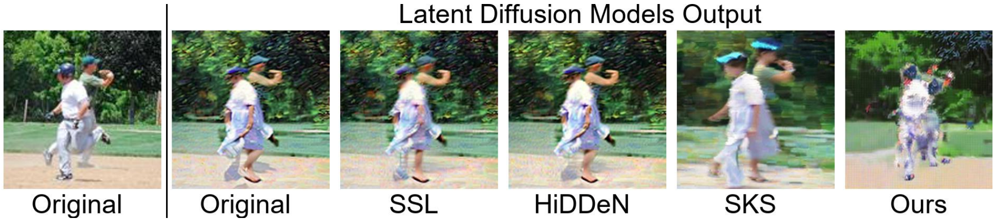


Figure 10

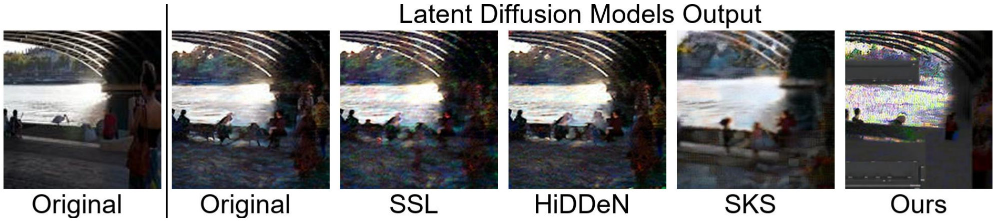


Figure 11

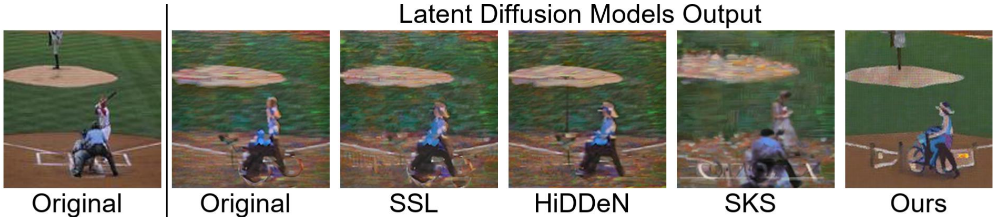


Figure 12

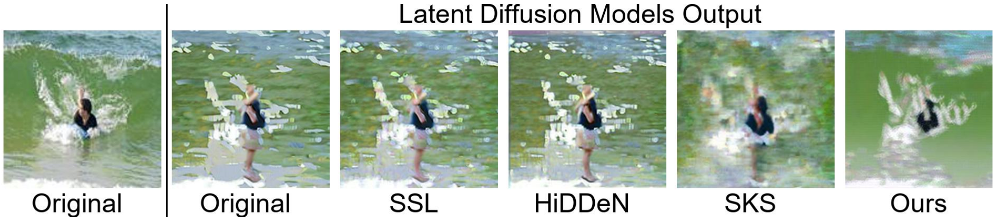


Figure 13

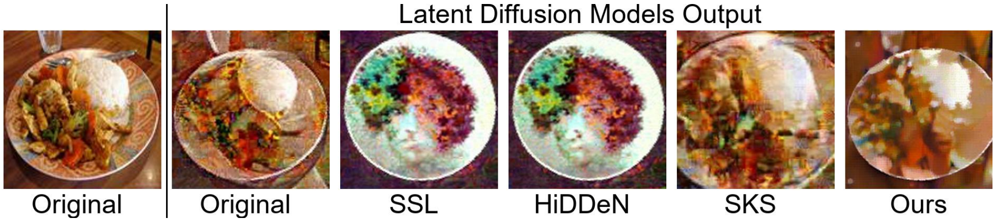


Figure 14

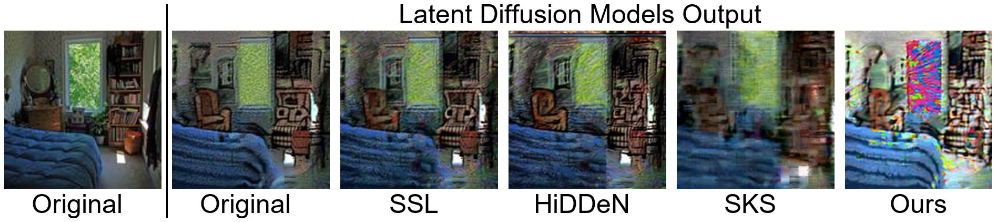


Figure 15

I Effect of Score Estimator Backbone Choice

To study the impact of Score Estimator capacity on watermark imperceptibility, we replaced the VGG16 backbone with a ResNet-101 model while keeping all hyperparameters identical.

Although the adversarial optimization was still able to reduce conformity scores below the detection threshold, the resulting watermarked images exhibited substantially increased visual noise and perceptual degradation. This indicates that stronger predictors, while more accurate, reduce predictive uncertainty and leave less room for adversarial optimization under imperceptibility constraints.

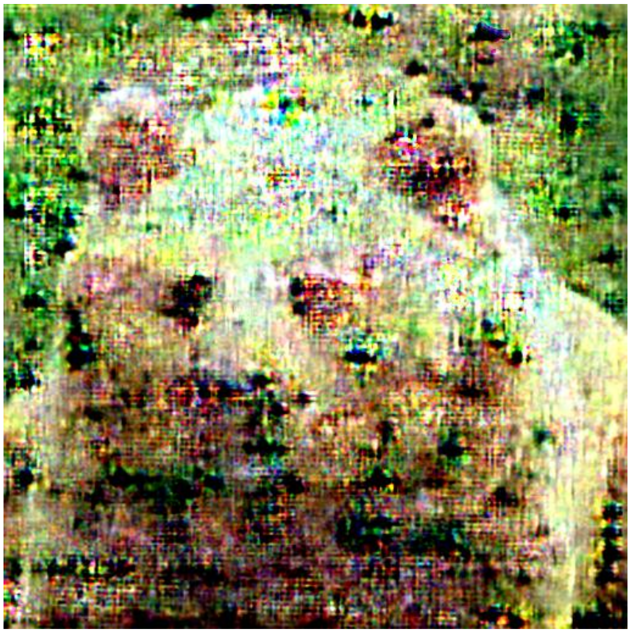


Figure 16: Watermarked image generated using a ResNet-101–based Score Estimator. Compared to VGG-based watermarking (Figure 4 in the main paper), the ResNet backbone produces substantially higher visual noise and perceptual distortion, despite achieving low conformity scores.