Your Text Encoder Can Be An Object-Level Watermarking Controller

Naresh Kumar Devulapally[1†] Mingzhen Huang[1] Vishal Asnani[2]
Shruti Agarwal[2] Siwei Lyu[1] Vishnu Suresh Lokhande[1†]
[1]University at Buffalo, SUNY [2]Adobe Research
{devulapa, mhuang33, siweilyu, vishnulo}@buffalo.edu {vasnani, shragarw}@adobe.com

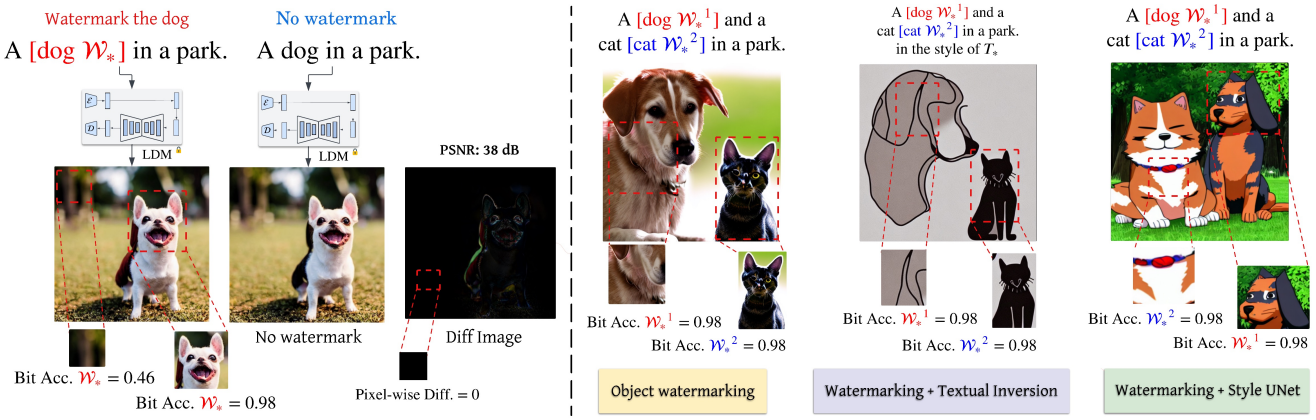


Figure 1. **Text prompt–controlled object-level watermarking:** *(Left Image): Our method embeds a watermarking token $\mathcal{W}_*$ into text-to-image generation, allowing users to watermark a full image or selected objects in an image. By leveraging cross-attention maps, any subset of prompt tokens $\{\mathcal{P}_i, \mathcal{P}_j, ...\}$ can be targeted while preserving non-watermarked regions. (Right Image): We also demonstrate personalized object watermarking using Styled UNet and Textual Inversion.*

Abstract

Invisible watermarking of AI-generated images can help with copyright protection, enabling detection and identification of AI-generated media. In this work, we present a novel approach to watermark images of T2I Latent Diffusion Models (LDMs). By only fine-tuning text token embeddings $\mathcal{W}_*$, we enable watermarking in selected objects or parts of the image, offering greater flexibility compared to traditional full-image watermarking. Our method leverages the text encoder's compatibility across various LDMs, allowing plug-and-play integration for different LDMs. Moreover, introducing the watermark early in the encoding stage improves robustness to adversarial perturbations in later stages of the pipeline. Our approach achieves 99% bit accuracy (48 bits) with a $10^5\times$ reduction in model parameters, enabling efficient watermarking. Code can be found at `github.com/naresh-ub/object_watermark`.

1. Introduction

As debates around copyright and intellectual property intensify, the need for effective watermarking solutions increases [33]. With generative AI models producing indistinguishable images from human-made art, maintaining authorship provenance in digital media is crucial. Efforts are underway, from legislative measures like the Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence [1], to industry initiatives that aim to watermark all AI-generated content [24] [30]. These developments highlight the growing importance of watermarking as a key area of study.

As a result, multiple Generative AI (GenAI) watermarking methods have been proposed [7, 9, 33] to invisibly watermark the entire image while generating the image. However, lots of current watermark methods [3, 6] can only encode a watermark to an open-box generative model as they need to access the latent space. Moreover, similar to other invisible watermarking techniques [2, 6, 7], these methods also suffer with the inherent trade-off between quality and robustness of the embedded signal. Owing to the high imperceptibility requirements for the high-quality image generation, these watermarks more often lacks robustness [9, 35]. To improve the watermark imperceptibility, partial watermarking has been proposed to limit the watermark related changes only to selected "non-salient" objects or regions in the image [22]. In our paper, we bring such

†Corresponding authors: N. K. Devulapally and V. S. Lokhande.

partial watermarking to text-to-image generation pipeline. Our partial watermarking is not only for improving the quality of the watermarked image, but also to give the user the flexibility to watermark only selected object of interest in the generated image without accessing the latent space. This is crucial for scenarios where the user would like to protect the unique object in the image, see Fig. 1.

Specifically, we propose a token-based watermarking approach that can embed watermarks into selected objects or partial regions of an image. Unlike previous works that have explored architectural modifications, such as the addition of secret encoders [3], distortion layers [6], and adapters [5], our watermarking method focuses solely on learning a watermark embedding. This allows for a *prêt-à-porter* style of training [28], leveraging pre-trained models with minimal adjustments. Additionally, compared with prior works [3, 5, 6], our model is blind, meaning that the entire generative model is neither modified nor requires additional training. A simple watermark token is learned by our model and can be plugged into any diffusion model for watermark generation. Previous works [7, 33] have investigated image watermarking without altering the architecture. Although effective, they lack the convenience of *prêt-à-porter* training, which offers the key advantage of personalizing large models with minimal retraining while using the model's core components as-is. In contrast, our approach enables users to apply watermarking directly at the prompt stage without modifying core model components such as the UNet or image decoder. We introduce new pseudo-tokens $\mathcal{W}_*$ into the model's vocabulary, where the $\mathcal{W}_*$ can be used as an input text prompt to be applied on any T2I diffusion models for generation watermarked images. This also enables users to selectively watermark specific image regions by leveraging the cross-attention between the special token and visual region in the image where the watermark needs to be embedded. By integrating watermarking functionality early in the text-to-image pipeline, our approach performs in-generation watermarking, offering greater robustness against image manipulation attacks compared to post-processing techniques while improving the overall quality of the watermarked images.

We make the following **key contributions**:

1. We introduce a novel pseudo-token with watermarking capabilities, learning a new embedding vector for it. We investigate the impact of applying this pseudo-token conditioning at different timesteps of the diffusion process, finding that timesteps closer to the VAE encoder in LDMs during forward process offer enhanced image generation quality.
2. We introduce object-level watermarking, allowing for selective watermarking with greater precision than traditional whole-image approaches.
3. We offer plug-and-play integration with various Stable Diffusion (SD) variants by embedding the watermark directly via the text encoder. Our approach achieves 99% bit accuracy with a $10^5\times$ reduction in model parameters, enabling efficient watermarking with a throughput of 48 bits.

2. Related Works

Text-Guided Image Generation Diffusion Models: Diffusion models have revolutionized text-to-image (T2I) generation, surpassing GANs in fidelity and diversity. By iteratively denoising random noise into images conditioned on text prompts, these models enable fine-grained control over generation. Frameworks like DALL-E 2, Imagen [13], and Stable Diffusion [31] democratize high-quality image synthesis, while diffusion-based editing methods [12] enable localized modifications. However, their widespread adoption raises challenges, including copyright infringement and harmful content generation, necessitating controllable synthesis mechanisms such as watermarking.

Textual Inversion for Personalized Image Generation: Textual Inversion [8] embeds visual or stylistic concepts into T2I models through learnable tokens, enabling personalization without altering model parameters. This lightweight approach has been extended to style transfer [34]. In this work, we find that training the special token not only allows for flexible concept learning but also could acts as a watermark controller.

Constraint-Based Control in Text-to-Image Generation: Text-to-image generation has gained significant popularity, especially to incorporate constraints that enhance control over generated content. Common constraints include spatial or layout constraints, which dictate object placement and dimensions within the generated image to meet predefined spatial requirements [17]. Another type, multi-view consistency constraints, ensures scene consistency across perspectives, preserving layout, lighting, and depth even in complex views like outdoor or stylized scenes [36]. Attribute preservation constraints maintain prompt-specified attributes (e.g., color, texture, size) in the generated image, ensuring semantic alignment. Finally, watermarking constraints embed an invisible watermark to verify ownership while preserving image quality. Watermarks can be embedded through polytope constraints defined by orthogonal Gaussian vectors or high-frequency components, allowing watermarking without image degradation. Constraint-based generation methods, which offer explicit control over outputs, draw inspiration from classifier training techniques that optimize using soft penalties [4]. Recent approaches such as [11] have achieved notable success with convergence guarantees and adaptability across data regimes .

In-generation Image watermarking in T2I generation: With the increasing popularity of diffusion models, recent research has explored embedding watermarks within the

diffusion process, either by manipulating the noise schedule to encode watermark information or by conditioning the model to output watermarked images. Tree-ring [33] proposed to encode a watermark into the noise space, it can be simply detected by applying a DDIM [31] inversion to obtain the noise. However, current in-generation watermark methods [6, 7, 26] cannot inject a watermark in a local region, *e.g.* an object. As many diffusion-based image editing method have been emerging for a long time, the object-level watermarking is long overdue. Recently, WAM [29] discusses localized watermarking and proposes a training mechanism for watermarked region detection from watermarked image. However, WAM still requires segmentation masks to *localize* watermark during watermark embedding in both training and inference.

Blind v/s Non-Blind Watermarking: Mentioned in [16], blind detection methods verify ownership using only the watermarked image, eliminating reliance on auxiliary metadata. Our work adopts this practical approach, ensuring compatibility with standard detection workflows.

Motivated by these insights, we explore similar methods in text-to-image generation, where constraints are often enforced as soft penalties via gradient-based optimization. Current diffusion-based watermarking methods lack spatial control, limiting object-centric applications. Recent advances in localized diffusion editing [12] remain unexplored for watermark integration. By unifying textual inversion with constraint-based control, we propose the first framework for object-level in-generation watermarking in T2I generation, addressing traceability and granularity.

3. Problem Statement

Our problem setting uses Latent Diffusion Models (LDMs) [27] for image generation. We consider an in-generation watermarking scenario, as defined in Sec. 2, where the aim is to utilize existing modules within the LDM pipeline to watermark images while generation.

Watermark Embedder: Given an input text prompt ($\mathcal{P} = \{p_0, p_1, ...p_n\}$) and/or an image $\mathcal{I}$, the aim of an in-generation watermark embedder (W_e) is to generate a watermarked image $\mathcal{I}_w = LDM(\mathcal{I} \mid W_e, \mathcal{P})$. Each embedder W_e is associated with a watermark key (m) (a bit string containing 0, 1 similar with prior works [6, 7]) of length k, i.e., $m \in \{0,1\}^k$. From here we shall use $W_{e,m}$ to denote a watermark embedder and $\mathcal{I}_{w,m}$ to denote a watermarked image.

Watermark Detector: Given a watermarked image $\mathcal{I}_{w,m}$, watermark detector $D_w(\cdot)$ is a neural network that predicts the key m from $\mathcal{I}_{w,m}$. Our method utilizes *Blind Watermarking* scenario mentioned in Sec. 2 where $D_w(\cdot)$ only requires the watermarked image $\mathcal{I}_{w,m}$ as input without the need for any additional metadata.

Attack Module: Attack module $\mathcal{A}(\cdot)$ represents unintentional (or) intentional transforms applied to the watermarked image $\mathcal{I}_{w,m}$ that could result in the loss of watermark key during watermark detection. A successful attack module results in imperfect watermark detection, i.e., $D_w(\mathcal{A}(\mathcal{I}_{w,m})) \neq m$. This module can usually be seen in the form of image compression by public platforms or an intentional malicious attacker that aims to remove watermark from $\mathcal{I}_{w,m}$. Resistance to an attack module is a fundamental requirement of a robust watermark embedder W_e.

Full-Image and Object-Level Watermarking: Given a text prompt for generation, *full-image watermarking* refers watermarking *entire image* and *object-level watermarking* refers to the scenario that watermarks *specific objects* $O_i, ...O_j$ in the image selected by a user. Information about O_i can be provided in various ways. For example [7] uses post-processing techniques to localize watermark on various regions using masks. However, using such post-processing methods introduces additional computational overhead and may be susceptible to removal or modification. Instead, integrating object-level watermarking into the generative process enables seamless and robust embedding within the chosen object $O_i, ...O_j$ regions.

Our problem setting considers both image and object-level watermarking scenarios with a constraint that the information for watermarking a specific object O_i shall not be provided in the form of any additional information such as segmentation masks. This information can only be obtained right from the text prompt. Specifically, for our setting, a text prompt "[A photo of a cat $\mathcal{W}_*$]" denotes full-image watermarking and a text prompt "A photo of a [cat $\mathcal{W}_*$]" denotes that the object cat is to be watermarked.

4. Method

Given an image, $\mathcal{I}$, our training pipeline aims to generate $\mathcal{I}_{w,m}$ where m is the watermark key, $m \in \{0,1\}^k$. We utilize a differentiable watermark detector from [6] $D_w(I_{w,m})$ that permits gradient flow.

It is known that training larger modules of an LDM pipeline, such as VAE Encoder/Decoder or the UNet, is computationally expensive and do not provide a lightweight, seamless way to integrate watermarking into various other LDM pipelines. There is a need for a unified watermark embedder that can be integrated with various LDM variants.

4.1. Token Embeddings for Watermarking

Our method utilizes the relatively under-explored Text Encoder of the LDM pipeline [19]. We introduce a new token in the Text Encoder, denoted by $\boldsymbol{\mathcal{W}}_*$, and fine-tune the text embeddings of $\boldsymbol{\mathcal{W}}_*$ to watermark $\mathcal{I}$. In addition to significant lower parameter requirement compared with prior works [6, 7], this token $\boldsymbol{\mathcal{W}}_*$ act as the watermark trigger

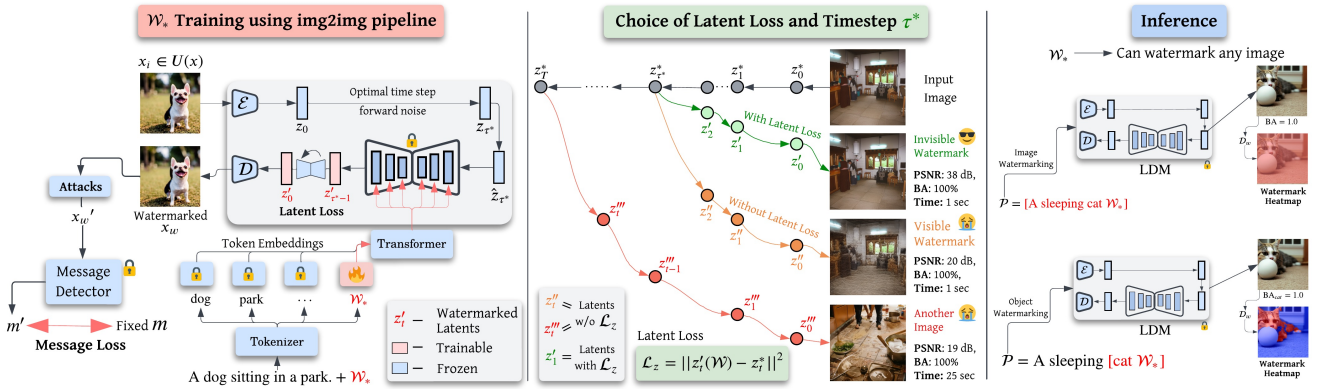


Figure 2. **$\mathcal{W}_*$ training pipeline.** *(Left) To find $\mathcal{W}_*$ token embeddings, we use an Img2Img generation pipeline. $\mathcal{D}$ and $\mathcal{D}_w$ represent VAE decoder in the LDM, and Watermark Detector respectively. While training, we send the input image through LDM encoder to retrieve the latent z_0, we then add a forward diffusion noise of τ^* timesteps, followed by iteratively denoising z_{τ^*} using Classifier-Free Guidance [11] from $[\tau^* \rightarrow 0]$ to retrieve $z'_{0,w}$. During the denoising process, we train for $\mathcal{W}_*$ token embeddings. (Middle) We use latent matching loss to control the trajectory of watermarked latents and bit loss to find $\mathcal{W}_*$ token embeddings. (Right) We then use trained $\mathcal{W}_*$ embeddings to generate watermarked images.*

that can be seamlessly integrated into the text encoder of *any* LDM pipeline.

Our method initially aims to train $\boldsymbol{\mathcal{W}_*}$ token embeddings, similar with Textual Inversion [8], to generate watermarked latents $z_{t,w} = \epsilon_\theta(z_t, t, \{\mathcal{P}, \mathcal{W}_*\})$. However, we identify the need to *carefully select the optimal noise timesteps* during $\boldsymbol{\mathcal{W}*}$ training as different choices impact image quality, watermark robustness and image generation time.

4.2. Optimal Timestep and Latent Loss

We perform empirical studies to observe the effect of different noise timesteps during forward diffusion process while $\boldsymbol{\mathcal{W}_*}$ training. As depicted in Fig. 2 (middle) (and in the section H. of the supplement), we observe a trade-off between image quality and watermarking performance when choosing different noise timesteps. A relatively large timestep $(t \sim T)$ would degrade the image quality while a relatively small timestep $(t \sim 0)$ would lead a lower bit accuracy but enhanced image generation quality. This observation is consistent with prior findings, as mentioned in [15, 28], which highlight the impact of noise strength on conditioning fidelity. We identify an optimal timestep $\tau^* = 8$ that balances this trade-off between image quality and watermark bit accuracy. During $\boldsymbol{\mathcal{W}_*}$ training, we add a forward noise of τ^* to the image followed by iterative de-noising to generate watermarked latent $z'_{0,w}$. $z'_{0,w}$ is then passed into the VAE decoder $Dec(\cdot)$ to generate a watermarked image $I_{w,m}$. D_w takes $I_{w,m}$ as input, $D_w(I_{w,m}) = m'$, to train $\boldsymbol{\mathcal{W}_*}$ token embeddings on watermark loss $\mathcal{L}_w$. We utilize BCE loss to embed a specific bit key m using $\boldsymbol{\mathcal{W}_*}$.

$$\mathcal{L}_w = BCE(D_w(Dec(z'_{0,w})), m). \tag{1}$$

While the optimal timestep τ^* preserves overall information in $\mathcal{I}$ and significantly reduces the time taken for $I_{w,m}$ generation, we still observe visible corruption in $I_{w,m}$ that hurts imperceptibility watermarking constraint. As a remedy, we employ latent matching loss $\mathcal{L}_z$ that aids our method to perform invisible watermarking.

$$\mathcal{L}_z = \min_{\mathcal{W}_*} \mathbb{E}_t \left[\|z^*_t - z'_t(\mathcal{W}_*)\|_2^2 \right]. \tag{2}$$

Our method uses $\mathcal{L} = \alpha\mathcal{L}_w + \beta\mathcal{L}_z$ to train $\boldsymbol{\mathcal{W}_*}$ token embeddings for watermarking where α and β are tunable hyperparameters.

4.3. Object-level watermarking

In the original LDM [27], the query feature Q and key feature K from cross attention operation defines a token-wise attention mask M where $M = \text{softmax}(Q \cdot K^T)/\sqrt{d})$. As the architecture of the LDM remains unchanged in our model, we inherit this capability to generate corresponding attention mask from text tokens. Such feature is the key in our model unlocking new possibilities for object-level watermarking utilizing these cross-attention maps.

Once the watermarking token embeddings are fine-tuned, we proceed to generate images using a standard text-to-image LDM model, but with an ability to perform object-level watermarking. We utilize cross-attention maps at each timestep t to localize watermark on any chosen object O_i right from the text prompt $\mathcal{P}$ as a token $\mathcal{P}_i (\in \mathcal{P})$. Specifically, we utilize [10] to obtain cross-attention map M_t of O_i given by:

$$z'_{t-1}, M_{t,O_i} \leftarrow \text{LDM}(z'_t, \mathcal{P}_i, t) \tag{3}$$

Algorithm 1 Object-level Watermarking during Text-to-Image Generation in Latent Diffusion Models

Input: LDM Decoder $\mathcal{D}$; Watermark Token $\mathcal{W}_*^i$ for object i; Text Prompt $\mathcal{P}$; overlay strength $\pi(t)$.

Select objects to watermark, defining the subset $\{\mathcal{P}_0^*, \mathcal{P}_1^*, \ldots, \mathcal{P}_k^*\}$ with corresponding watermarking tokens $\{\mathcal{W}_0^*, \mathcal{W}_1^*, \ldots, \mathcal{W}_k^*\}$ from the text prompt $\mathcal{P} = \{\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_n\}$.

$z_T \in \mathcal{N}(0, I)$

for $t = T, \ldots, 0$ **do**

  for each token $\mathcal{P}_i^* \in \{\mathcal{P}_0^*, \mathcal{P}_1^*, \ldots, \mathcal{P}_k^*\}$ **do**

    $\mathcal{M}_{\mathcal{P}_i^*}^{(t)} / \mathcal{M}_{\mathcal{W}_i^*}^{(t)} \leftarrow$ Attention map of $\mathcal{P}_i^* / \mathcal{W}_i^*$

    $\mathcal{M}_{\mathcal{W}_*}^{(t)} \leftarrow \pi(t) \cdot \mathcal{M}_{\mathcal{W}_*}^{(t)}$ (Adjust watermark-strength per timestep)

    $\mathcal{M}_{\mathcal{P}_i}^{(t)} \leftarrow (1-\alpha) \cdot \mathcal{M}_{\mathcal{P}_i}^{(t)} + \alpha \cdot \mathcal{M}_{\mathcal{W}_*}^{(t)}$ (Overlay attention map of $\mathcal{W}_*^i$ on that of $\mathcal{P}_*^i$)

  end for

  $\hat{\epsilon} = \epsilon_\theta(z_t, t, \psi(\mathcal{P}), \mathcal{M}_{\mathcal{P}_{0:k}^*}^{(t)}, \mathcal{M}_{\mathcal{W}_{0:k}^*}^{(t)})$

  $z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \hat{\epsilon}$

end for

Output: Decoder output of z_0, that is $\mathcal{D}(z_0)$

Given a text prompt with multiple tokens $\mathcal{P} = \{\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_n\}$, the user has the flexibility to choose specific object tokens $\{\mathcal{P}_i, \mathcal{P}_j, \ldots\}$ to watermark. During the image generation process, we extract the cross-attention maps for the objects intended for watermarking from the UNet. These are then combined with the attention maps of the corresponding watermarking tokens $\mathcal{W}_*^i, \mathcal{W}_*^j, \ldots$ to precisely localize the watermark on each object allowing us to seamlessly perform multi-object watermarking.

To bring in the effect of optimal timestep τ^* (seen during training Sec. 4.2), we introduce a watermark overlay strength controller $\pi(t)$. $\pi(t)$ could be set to a step function with values 0 (if $t > \tau^*$) and 1 (if $t \leq \tau^*$) which exactly mimics the generation scenario seen during training. In addition to the step function, we also test with a smoothing function that brings the effect of τ^* by giving more weight to timesteps closer to the VAE decoder. At these timesteps we observe increased reliability of the attention maps enhancing the precision of the watermark placement. The complete procedure is outlined in Algorithm 1.

5. Experiments

Datasets: We evaluate our method on two datasets, namely, MS-COCO [18] and WikiArt [32]. We use a subset of 2,000 images from MS-COCO dataset, similar to [7], for training $\mathcal{W}_*$ token embeddings. Our method is evaluated on 1000 validation captions from MS-COCO and WikiArt each via image-to-image generation and on 100 prompts from [6] for text-to-image generation.

Evaluation Metrics: In line with prior work [6, 7], we assess our watermarking method using the following metrics:

(a) *Robustness*, measured by Bit Accuracy under various attacks (mentioned in Tab. 1). Bit accuracy represents the percentage of correctly detected bits in the watermark key m output from the Detector $D_w(\cdot)$. In addition to Bit Accuracy, we also compare our method with techniques [33] that use True Positive Rate as the metrics in supplement (section D). These methods do not embed a bit string m.

(b) *Imperceptibility*, measured by Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [19, 23], and Fréchet inception distance (FID) [14], comparing watermarked and non-watermarked images.

(c) *Parameter Efficiency*, where our method trains only the textual embeddings, with parameter usage significantly reduced ($10^5\times$) compared to baselines [6, 7], as the T2I pipeline and watermark decoder remain frozen.

Watermark Detector: For our experiments we utilize two watermark detectors from [6] and [7].

5.1. Full-Image Watermarking results

We evaluate full-image watermarking by integrating $\mathcal{W}_*$ into the Stable-Diffusion v1.5 [31], using the AquaLora watermark detector [6]. Our method's performance is compared to existing watermarking baselines in LDMs, focusing on robustness and imperceptibility. Tab. 1 presents the results, showing that our approach requires significantly fewer parameters while maintaining high bit accuracy under attacks. We categorize attacks encountered after watermarking into two categories, namely, basic image processing attacks such as Rotation, Resize, Crop, JPEG compression etc as listed in Tab. 1, and adversarial attacks such as DiffPure [21], WMAttacker [35] etc. We provide implementation details of attack module in supplement (section I).

We consider that robustness to adversarial attacks including Diff-Pure [21] and SDEdit [20], while beneficial, is not essential for watermarking techniques. These attack methods are complex and go beyond common attacks like crop, resize, and rotations. However, our watermarking method integrated into the denoising process demonstrates enhanced robustness to both basic and adversarial attacks outperforming several in-generation watermarking techniques thereby pushing the benchmark for robust watermarking. When compared to AquaLORA [6], while using the same watermark detector, we see an increase in bit accuracy on both common attacks by over 20%, and over 7dB in PSNR. Additionally, our method retains high robustness and personalization when used alongside a personalized textual inversion token and/or a personalized UNet.

Summary. Our Full-image watermarking achieves improved robustness, lower parameter requirements, and enhanced imperceptibility compared to baselines.

Method	I.W.	O.W.	L.P.	# Params ↓	Imperceptibility			Robustness to basic attacks (BA):							Adversarial attacks (BA):		
					PSNR ↑	SSIM ↑	FID ↓	None ↑	Brightness ↑	Contrast ↑	Blur ↑	Crop ↑	Rot. ↑	JPEG ↑	SDEdit ↑	WMAttacker ↑	DiffPure ↑
WikiArt (48 bits) - In-Generation Watermarking																	
Stable Sig. [7]	✓	✗	✗	10^5+	31.57	0.88	24.71	0.99	0.93	0.87	0.78	0.79	0.70	0.55	0.58	0.53	0.52
LaWa [26]	✓	✗	✗	10^5+	32.52	0.93	18.23	0.99	**0.99**	0.98	0.94	0.93	0.83	0.89	0.68	0.76	0.78
TrustMark [2]	✗	✗	✗	10^5+	39.90	0.97	15.83	0.99	0.98	0.99	0.93	0.89	0.87	0.86	0.68	0.77	0.73
RoSteALS [3]	✓	✗	✗	10^5+	32.68	0.88	16.63	0.98	0.96	0.94	0.88	0.88	0.75	0.80	0.75	0.72	0.72
AquaLoRA [6]	✓	✗	✗	10^5+	31.46	0.92	17.27	0.94	0.91	0.91	0.81	0.90	0.58	0.76	0.68	0.67	0.66
WAM [29]	✓	✓	✗	10^5+	36.46	0.97	16.27	0.97	0.93	0.92	0.84	0.92	0.76	0.84	0.72	0.71	0.72
Ours + SD_{style}	✓	✓	✓	**768**	35.88	0.93	16.72	0.99	0.97	0.99	0.94	0.97	0.94	0.93	0.81	0.83	0.84
Ours + TI	✓	✓	✓	**768**	36.89	0.94	15.98	0.99	0.96	0.99	0.95	0.98	0.95	0.92	0.82	0.84	0.86
Ours + SD	✓	✓	✓	**768**	**39.92**	**0.97**	**14.89**	**0.99**	0.98	**0.99**	**0.97**	**0.98**	**0.95**	**0.95**	**0.85**	**0.87**	**0.88**
MS-COCO (48 bits) - In-Generation Watermarking																	
Stable Sig. [7]	✓	✗	✗	10^5+	31.93	0.88	24.74	0.99	0.94	0.89	0.81	0.82	0.72	0.59	0.62	0.63	0.57
LaWa [26]	✓	✗	✗	10^5+	33.53	0.95	17.45	0.99	**0.99**	0.98	0.94	0.93	0.86	0.90	0.71	0.79	0.79
TrustMark [2]	✗	✗	✗	10^5+	**40.98**	**0.98**	14.89	0.99	0.98	0.99	0.95	0.91	0.89	0.88	0.69	0.78	0.74
RoSteALS [3]	✓	✗	✗	10^5+	32.68	0.88	16.63	0.99	0.98	0.93	0.89	0.87	0.78	0.81	0.76	0.72	0.75
AquaLoRA [6]	✓	✗	✗	10^5+	31.46	0.92	17.27	0.95	0.91	0.90	0.80	0.92	0.68	0.78	0.67	0.70	0.68
WAM [29]	✓	✓	✗	10^5+	36.98	0.97	16.85	0.98	0.94	0.92	0.86	0.94	0.77	0.86	0.73	0.73	0.72
Ours + SD_{style}	✓	✓	✓	**768**	36.99	0.93	16.72	0.99	0.98	0.99	0.94	0.97	0.94	0.93	0.81	0.83	0.84
Ours + TI	✓	✓	✓	**768**	36.89	0.94	15.23	0.99	0.96	0.99	0.95	0.98	0.96	0.94	0.83	0.84	0.86
Ours + SD	✓	✓	✓	**768**	40.92	0.97	**14.83**	**0.99**	0.98	**0.99**	**0.97**	**0.98**	**0.97**	**0.96**	**0.86**	**0.88**	**0.89**

Table 1. **Comparison to watermark baselines.** *(I.W.: In-generation Watermarking, O.W.: Object-level Watermarking, L.P.: Less than 10^5 parameters, BA: Bit Accuracy) We compare our method several baselines. In addition to watermark invisibility and robustness of watermarking, we present the number of parameters used for training. We see that our method uses $10^5\times$ lower parameters. We see that early integration for watermarking improves robustness to attacks, we see a consistent trend of this improvements in basic image processing attacks and adversarial attacks. We also present the performance of our method in the presence of personalization using fine-tuned UNet and Textual Inversion. From the above table we see that our method can be plugged into various LDM pipelines. More details on specific implementation of attacks can be found in the supplement.*

5.2. Key results on Object Watermarking

Object Watermarking and Identification: We present the effectiveness of object-level watermarking, as illustrated in Fig. 3, where we apply watermarks to up to three distinct objects within a single image, mentioned in Algorithm 1. Using the watermark detector D_w, and without the need for any additional information such as segmentation masks, heatmaps are generated by evaluating the bit accuracy of small patches, each covering approximately 10% of the image area. Each patch's bit accuracy score is determined by D_w , and these scores are aggregated across all patches to produce a comprehensive heatmap. In this map, a score of 1 represents a bit accuracy of 100%, indicating high fidelity in watermark retrieval, while a score of 0.5 indicates retrieved watermark key does not match m indicating no watermark. When watermarking a single object, the resulting heatmap shows precise retrieval of the watermark. For images with two or three watermarked objects, the detector reliably identifies each watermark, and accurate retrieval is observed. In cases involving four or more objects (bottom-right of Fig. 3), the heatmap confirms that the watermark is detected across most of the image, likely due to the accuracy of attention maps during inference. For images with multiple objects, separate heatmaps can be generated for each object to enhance clarity in visualization. The overall heatmap presented in Fig. 3 is an aggregate of individual heatmaps detailed in Fig. 6. Further technical details on the heatmap generation process are available in the supp. (§B).

	Bit Accuracy					
	None	**Bright.**	**Cont.**	**Blur**	**Rot.**	**JPEG**
Single object						
Segment + white bg	0.99	0.97	0.96	0.97	0.96	0.97
Segment + style bg	0.95	0.92	0.90	0.96	0.94	0.93
Crop object (0.8 × size)	0.96	0.94	0.92	0.95	0.92	0.93
Crop object (0.5 × size)	0.92	0.91	0.90	0.91	0.92	0.90
Crop object (0.4 × size)	0.90	0.89	0.88	0.89	0.89	0.90
Multiple objects						
1 object	0.99	0.97	0.96	0.97	0.95	0.98
2 objects (no overlap)	0.94	0.93	0.95	0.95	0.94	0.96
3 objects (no overlap)	0.90	0.89	0.90	0.90	0.90	0.99
2 objects (overlap $\geq 40\%$)	0.79	0.76	0.70	0.80	0.74	0.74

Table 2. **Object-level watermarking robustness performance with attacks.** *We evaluate the performance of our method to perform object-level watermarking in the presence of attacks. We clearly see robust performance under rotation and various crops within watermarked objects, for single and up to three objects.*

Stress Tests: Object Size and Multiple Object Watermarking: We evaluate the sensitivity of the watermark detection to small object sizes and multiple objects. As shown in Tab. 2, even with a 40% crop of the object, the detector achieves 89% bit accuracy. This accuracy remains robust to transformations. We also test watermark detection across multiple objects, with and without overlap. For non-overlapping objects, detection remains above 90% without attacks and above 89% under attacks. However, performance decreases when the overlap exceeds 40%, particularly in the case of multiple overlapping objects. Our qualitative results indicate that single-object watermarking is not

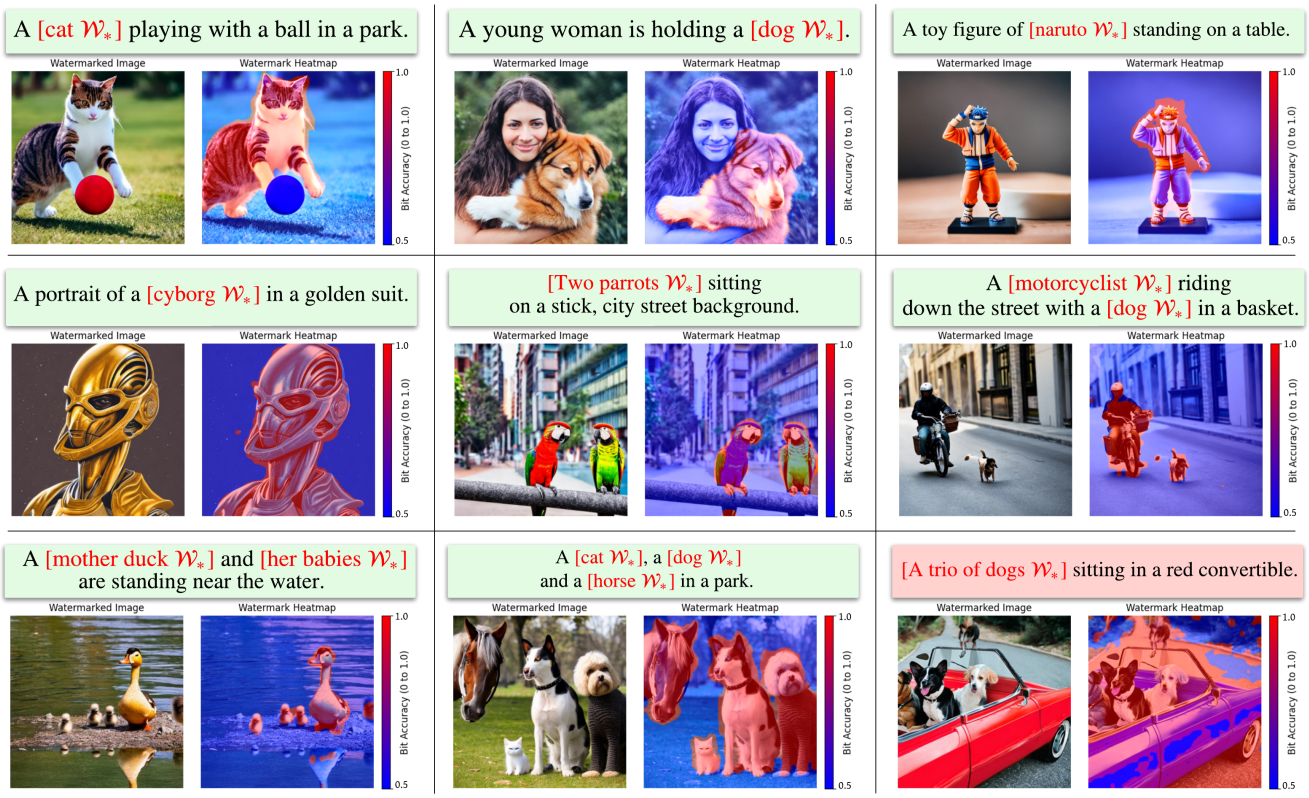


Figure 3. **Qualitative results and watermark heatmaps.** *We show qualitative results of our watermarking approach for up to three objects in a single image, all embedded within the T2I generation pipeline. As described in Sec. 4.3, object-level watermarking is controlled directly via the text prompt $\mathcal{P}$. (Top row) shows single-object watermarking with corresponding heatmaps, achieving high bit accuracy within the selected object and 0 outside. (Second row, third row) show results for two and more than three objects, respectively, with accurate retrieval and high bit accuracy. (Bottom row, last column) illustrates a case where imperfect attention maps cause watermarking to leak outside objects, yet high bit accuracy is still achieved.*

affected by object overlap, but bit accuracy decreases with significant overlap due to multiple layers of interference.

5.3. Applications: Integration with LDM Pipelines and Textual Inversion Compatibility

We demonstrate the versatility of our method by integrating the plug-in watermark token $\mathcal{W}_*$ across diverse text-to-image (T2I) generation pipelines, assessing both watermark robustness and imperceptibility. The schematic in Fig. 2 illustrates this integration process, where $\mathcal{W}_*$ is loaded into the text encoder of different LDMs, including pipelines that employ personalized or fine-tuned diffusion models. Additionally, $\mathcal{W}_*$ can be combined with Textual Inversion tokens, as shown in qualitative results in Fig. 3. Our method achieves a high PSNR of 35 dB and bit accuracy above 92%, outperforming the Stable Signature [7] in robustness while maintaining watermark invisibility and resistance to attacks.

5.4. Ablation Studies

Training with SAM Segmentation Masks: Our watermarking pipeline relies on attention maps generated by prompts, though these maps can sometimes lack precision (e.g., spillover beyond the intended object, seen in bottom-right scenario of Fig. 3). This can pose challenges for applications like medical imaging, where precise watermark localization is crucial. In case of a need for such high precision localization of watermark target, our method is flexible to utilize segmentations masks within watermark generation and need not rely on post-generation localization. In this ablation study, we incorporate SAM (Segment-Anything) segmentation masks directly into our generation process as an alternative to attention maps, particularly in Step 4 of Algorithm 1. Our results (Fig. 5) show that SAM masks significantly improve watermark localization accuracy.

Additional analysis on more complex scenes with multiple overlapping objects: We clarify that our setting is indeed very challenging: We retrieve multiple watermarks from overlapping regions using the same detector that is natively pretrained to retrieve single watermark. This setting is known to cause a major drop in robustness (Fig. 4 of [25]) and remains an open research challenge on how to extract more than one watermark from an image region. To further support this, we provide a quantitative analysis.

No drop is expected if we are permitted to use more than

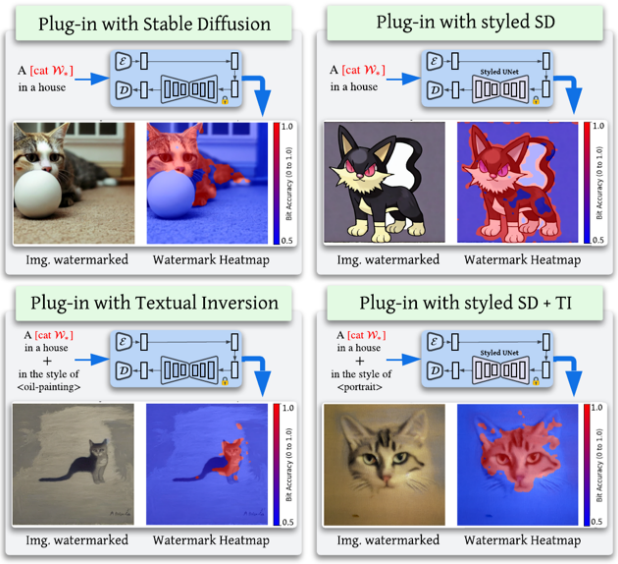
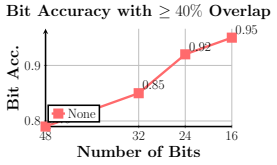


Figure 4. **Plug-and-Play ability.** *We present our method's ability to be plugged into any combination of personalized T2I model. Above image shows four such combinations where we use Textual Inversion style tokens and styled T2I pipelines can be seen. It can be observed that the object-level control and watermarking ability of our method is preserved across all these pipelines.*

one watermark detector. Even with the same detector 36 bits can be retrieved with 0.85 Acc. with $\geq 40\%$ overlap.

S.K.: Same Key, D.K.: Different Keys, S.D.: Same Detector, D.D.: Different Detector, O: $\geq 40\%$ Overlap.

Retrieval Accuracy	
S.K. + O + S.D.	**0.99**
D.K. + O + D.D.	**0.99**
D.K. + O + S.D.	**0.79**



Left: Retrieval accuracy under different configurations. Right: Performance on the (D.K. + O + S.D.) setting under varying bit counts.

Ablation on Number of bits for Watermarking. We perform an ablation on number of bits that can be embedded into image by our method and present our results as a plot in the supplement (section C.). We observe that our method can watermark with $> 91\%$ bit accuracy for 128 bits under various attacks.

6. Limitations

As our method performs in-generation watermarking. For localizing the watermark on an object, the method relies on Cross Attention maps for each token in the input prompt $\mathcal{P}$. Hence, the method relies strongly on the accuracy of these cross attention maps. Our ablation studies conducted on using the segmentation masks from SAM enhance the localization of watermark on top of the object. However, relying on an external segmentation model such as SAM could be undesirable and overly relying on Cross Attention maps could be a limitation when the attention maps are not properly defined.

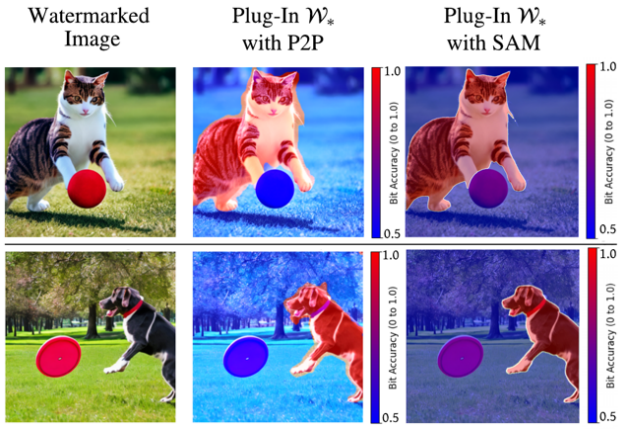


Figure 5. **Watermark heatmap enhancement using SAM.** *Our watermarking is embedded into the generation pipeline using Attention maps. We test the performance of our watermark detection by plugging our method into P2P and SAM.*

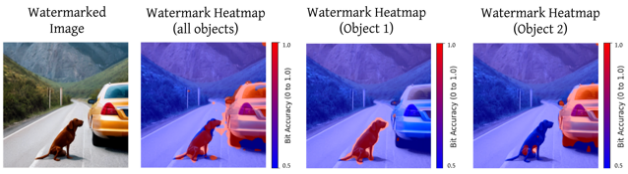


Figure 6. **Multiple Object watermarking heatmap breakdown.** *Our goal is to embed watermark in specific regions of the image while not corrupting the entire image. In the above example, a user can choose to watermark either the dog or the car or both. Our method provides the versatility to choose to watermark any object(s) in an image while preserving other regions.*

7. Conclusions

In this paper, we propose a novel in-generation watermarking technique to integrate watermarking into the latent within the denoising process of T2I generation. Our watermarking technique provides watermarking control directly from text and fine-tunes token embeddings of a single token. Our method contributes to a novel application of object-level watermarking within T2I generation. We show that our early watermarking technique shows improvements in watermarking robustness across several post generation attacks. Our method aims to motivate future research towards training-free watermarking, controllable watermarking with any T2I generation pipeline.

Acknowledgments Prof. Lokhande thanks support provided by University at Buffalo Startup funds, Adobe Research Gift and internal funding from the University at Buffalo's Research and Economic Development office.

References

[1] Joseph R Biden. Executive order on the safe, secure, and trustworthy development and use of artificial intelligence. 2023. 1

[2] Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images, 2023. 1, 6

[3] Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space, 2023. 1, 2, 6

[4] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G. Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions, 2022. 2

[5] Weitao Feng, Jiyan He, Jie Zhang, Tianwei Zhang, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. Catch you everything everywhere: Guarding textual inversion via concept watermarking, 2023. 2

[6] Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. AquaLoRA: Toward white-box protection for customized stable diffusion models via watermark LoRA. In *Proceedings of the 41st International Conference on Machine Learning*, pages 13423–13444. PMLR, 2024. 1, 2, 3, 5, 6

[7] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22409–22420, 2023. 1, 2, 3, 5, 6, 7

[8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2, 4

[9] David Haden. Now you see it... now you don't: Detecting ai use via image watermarking. *Information Today*, 40(9): 31–33, 2023. 1

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. 4

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2, 4

[12] Mingzhen Huang, Jialing Cai, Shan Jia, Vishnu Suresh Lokhande, and Siwei Lyu. Paralleledits: Efficient multi-aspect text-driven image editing with attention grouping, 2024. 2, 3

[13] Imagen-Team-Google. Imagen 3, 2024. 2

[14] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation, 2024. 5

[15] Maomao Li, Yu Li, Yunfei Liu, and Dong Xu. Exploring iterative manifold constraint for zero-shot image editing, 2025. 4

[16] Yue Li, Hongxia Wang, and Mauro Barni. A survey of deep neural network watermarking techniques, 2021. 3

[17] Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N. Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models, 2024. 2

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 5

[19] Yuan Ma, Kewen Liu, Hongxia Xiong, Panpan Fang, Xiaojun Li, Yalei Chen, and Chaoyang Liu. Perception-oriented single image super-resolution via dual relativistic average generative adversarial networks, 2020. 3, 5

[20] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022. 5

[21] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification, 2022. 5

[22] Athanasios Nikolaidis and Ioannis Pitas. Region-based image watermarking. *IEEE Transactions on image processing*, 10(11):1726–1740, 2001. 1

[23] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020. 5

[24] Jacob Noti-Victor. Regulating hidden ai authorship. 2024. 1

[25] Aleksandar Petrov, Shruti Agarwal, Philip H. S. Torr, Adel Bibi, and John Collomosse. On the coexistence and ensembling of watermarks, 2025. 7

[26] Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. Lawa: Using latent space for in-generation image watermarking, 2024. 3, 6

[27] 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 (CVPR)*, pages 10684–10695, 2022. 3, 4

[28] Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Rb-modulation: Training-free personalization of diffusion models using stochastic optimal control, 2024. 2, 4

[29] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages, 2024. 3, 6

[30] Nathalie A Smuha, Emma Ahmed-Rengers, Adam Harkens, Wenlong Li, James MacLaren, Riccardo Piselli, and Karen Yeung. How the eu can achieve legally trustworthy ai: a response to the european commission's proposal for an artificial intelligence act. *Available at SSRN 3899991*, 2021. 1

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 2, 3, 5

[32] 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. 5

[33] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 5

[34] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 2

[35] 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, 2024. 1, 5

[36] Jinghao Zhou, Tomas Jakab, Philip Torr, and Christian Rupprecht. Scene-conditional 3d object stylization and composition, 2023. 2