RADIUM: RadioActive Decay of Image-Underlaid Marks

Michel Meintz*, **Louis Kerner***, **Maitri Vignesh Shah, Simon Hector, Franziska Boenisch**, **Adam Dziedzic**
CISPA Helmholtz Center for Information Security
{michel.meintz,louis.kerner,maitri.shah,simon.hector,boenisch,adam.dziedzic}@cispa.de
*Equal contribution

Abstract

Modern image generative models are able to produce photorealistic images. As those images become increasingly indistinguishable from real data and are published online, they are often scraped for subsequent training runs of new generative models. This practice of training on generated data has been shown to degrade model performance and cause model collapse. A possible mitigation lies in embedding radioactive watermarks into generated content. Radioactive watermarks are robust marks that are detectable in outputs of new models trained on watermarked data, enabling provenance tracing of generated content. In this work, we analyze the persistence of image watermarks across multiple training-generation runs. To do so, we introduce a novel statistical testing method RADIUM (RadioActive Decay of Image-Underlaid Marks) for reliable radioactivity detection across various watermarking methods. Using our RADIUM method, we observe disparate radioactivity across watermarking methods for image generative models. Few watermarks are detectable when making up small fractions of training data, while most decay severely, especially when the architecture of models differs. Our analysis highlights the critical need for more radioactive watermarking methods in the vision domain. We release the code at this URL.

1 Introduction

The generation capabilities of large scale image generative models [7, 10, 26, 38, 43] have improved drastically, allowing them to generate high-quality images that are increasingly difficult to distinguish from natural, human-created data. These images are then published online, becoming part of the data that future models are trained on. As training these models requires large and diverse datasets [29], which are costly and time-consuming to collect and curate [2, 31], training data is often scraped from the internet [15, 31]. However, this scraped data contains not only natural images but also model-generated content and their quality makes it difficult to differentiate between them. This can create feedback loops where models are trained on their generated data leading to worse performance and ultimately to *model collapse* [1, 35, 42].

A promising solution to address these concerns is watermarking [3, 8, 17, 41]. Watermarking can help model owners to detect and track their generated data, enabling both the detection of misuse and prevention of training on their own generated data. Therefore, watermarks embed visually imperceptible signals into the generated images either during the generation process or after the image was generated. These imperceptible signals can then be identified by model owners, enabling them to detect their own generated images. While these watermarks are designed to be robust against removal attacks [23, 41], tracking their influence through the training of a new model requires an additional property: the watermark must remain detectable in the outputs of the model trained on watermarked data [17, 19]. This property is known as *radioactivity* [28, 30].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Radioactivity of watermarks for generative models has been analyzed in the language domain [30], demonstrating that fine-tuning a large language model (LLM) on watermarked text, results in the watermark being detectable in the output of the fine-tuned model. However, the extension of the radioactivity analysis to the image domain is non-trivial. Image generative models operate in high-dimensional continuous spaces, involving latent-space encoding or denoising processes that can destroy watermark signals. Existing work on radioactivity detection in the image domain [5] provided initial insights into the radioactivity of image watermarks, however, the existing evaluation lacked statistical guarantees.

In this work, we bridge this gap by extensively analyzing the radioactivity of image watermarking methods for different generation schemes under statistical guarantees. Therefore, we design a radioactivity test based on dataset inference [4, 6, 20, 24, 25, 27, 46], which aims to determine whether a collection of samples was part of a model's training data, by aggregating weak membership signals across the samples. We leverage the insights from dataset inference for our radioactivity test. Our method aggregates the watermark signals across many samples to detect their cumulative radioactivity. Beyond the previous dataset inference techniques, we further develop a new method for the radioactivity detection.

Our statistical testing method, *RADIUM* (*RadioActive Decay of Image-Underlaid Marks*), leverages recent advances in e-values to perform sequential, anytime-valid hypothesis testing [33]. We formulate radioactivity detection as a betting game played over a sequence of image pairs. At each step, the test places a *bet* against the null hypothesis that the suspect image contains no detectable watermark signal. As images are processed, evidence against the null is accumulated as statistical wealth, which is represented by an e-value: a nonnegative test statistic whose expectation is at most one under the null hypothesis. Consequently, large e-values provide evidence against the null, and rejecting the null whenever the e-value exceeds the value of $1/\alpha$, which yields a valid level-α test. The key advantage of this approach is that the evaluation is performed sequentially, and the detection procedure may be stopped at any point while providing valid statistical guarantees. In practice, this allows us to adaptively acquire additional samples when the evidence of radioactivity is inconclusive, or to terminate the test as soon as sufficient evidence of radioactivity has been accumulated.

In summary, we make the following contributions:

1. We introduce a new method *RADIUM* that combines dataset inference techniques with sequential testing to determine whether watermarks remain detectable after the watermarked images are used to train new generative models.
2. We develop an e-value-based statistical testing procedure for radioactivity detection, providing anytime-valid statistical guarantees across different image watermarking schemes.
3. We conduct extensive experiments across 7 watermarking methods and 3 model architectures, including diffusion and image autoregressive models: our results show how watermark signals decay over multiple rounds of model training and generation.
4. We analyze factors that influence watermark radioactivity, providing practical insights for the design of image watermarks that remain detectable after new models are trained on them.

2 Background and Related Work

Diffusion Models. Diffusion Models (DMs) [14, 36] are trained by progressively adding noise to the data and then learning to reverse this process. In the forward diffusion process, Gaussian noise $\varepsilon \sim \mathcal{N}(0, 1)$, is added to a clean image x, yielding the noisy sample $x_t \leftarrow \sqrt{\alpha_t}x + \sqrt{1-\alpha_t}\varepsilon$, where $t \in [0, T]$, is the diffusion timestep and $\alpha_t \in [0, 1]$ is a decaying parameter with $\alpha_0 = 1$ and $\alpha_T = 0$. The DM f_θ is then trained to predict the added noise ε, by minimizing $\mathcal{L}(x, t, \varepsilon; f_\theta) = \|\varepsilon - f_\theta(x_t, t)\|_2^2$. Latent diffusion models (LDMs) [26] improve over DMs, by performing the forward and reverse diffusion process in a latent space, rather than pixel space, significantly reducing computational complexity, making training scalable and more efficient. An encoder $\mathcal{E}$ encodes the input x to a lower dimensional latent representation $z = \mathcal{E}(x)$. The LDM diffusion loss is then formulated as $\mathcal{L}(z, t, \varepsilon; f_\theta) = \|\varepsilon - f_\theta(z_t, t)\|_2^2$.

Image Autoregressive Models. Image Autoregressive models (IARs) follow the next-token prediction paradigm from large language models (LLMs), treating images or image patches as sequences of

discrete tokens, allowing fast and high quality image generation. Their similarity to LLMs allows them to benefit from the clear power-law scaling [38]. Some works rely on randomized inputs permuted into different factorization orders with annealing probability [43], where others proposed the next-scale prediction [38], making autoregressive image generation even faster and producing higher resolution images. The current state-of-the-art IAR, Infinity, predicts multiple bits per token, rather than predicting discrete tokens directly, allowing for a near infinite vocabulary size [10].

Watermarking. Watermarking embeds visually imperceptible signals into data, enabling the subsequent detection, attribution, and tracing of generated content. For images produced by generative models, watermarking can be applied at different stages of the generation pipeline. *In-generation* methods embed the signal during the sampling or generation process [8, 16, 17, 19, 41], whereas *post-processing* methods modify the image after it has been generated [3, 37, 45]. Our analysis focuses on several representative state-of-the-art watermarking schemes. We consider the post-processing watermarks: TrustMark [3], RivaGAN [45], and StegaStamp [37]. We further evaluate SIREN [22], a watermarking method explicitly designed for dataset tracing in DMs, which introduces an additional training loss to mitigate the limited learnability of prior watermarking approaches. We also experiment with in-generation methods: StableSignature [8], a diffusion-model watermark designed to have a limited impact on image quality, TreeRing [41], which is known for its robustness to standard image perturbations, and finally, we include BitMark [17], the only in-generation watermarking method currently available for the Infinity model. We provide additional information in Appendix A.2.

Once an image has been watermarked, the embedded signal can be recovered or verified by an authorized detector, typically using a private key or message. To support reliable content tracing, watermarking schemes must remain robust to common image transformations and increasingly sophisticated removal attacks [23]. In general, image watermarking involves a three-way trade-off among *robustness*, *imperceptibility*, and *information capacity*. Robustness refers to the ability of the watermark to survive image augmentations and watermark removal attacks; imperceptibility captures the extent to which watermarked images remain indistinguishable from clean images; and information capacity denotes the amount of information encoded in the watermark, ranging from zero-bit detection to multi-bit messages. Improving one of these properties often comes at the expense of the others: for example, highly imperceptible watermarks may exhibit reduced robustness or capacity, whereas watermarks that have higher capacity might deteriorate more the visual quality and be less robust.

Watermark Radioactivity. Watermark radioactivity, *i.e.,* the persistence of watermarks through the training or fine-tuning process of a generative model is an additional robustness setting, that so far has had limited attention in the image domain. Recent studies in the language domain have demonstrated that training an LLM on watermarked data results in detectable traces in the outputs of the LLM [30]. Language watermarks change the prediction process of the model, by adapting its output probabilities to draw from a watermarked distribution. This distribution shift is then also detectable in the output of the second fine-tuned model. Sablayrolles et al. [28] show how to embed a radioactive signal into an image, such that a classifier trained on radioactive data is distinguishable from a classifier trained on clean data. Their method explicitly changes the models behavior, to trace the training data. A separate line of work with DIAGNOSIS [40], Recipe [47], and most recently SIREN [22], constructs watermarks for generative models, but with a fundamentally different goal: protecting natural training datasets by ensuring that any model trained on the watermarked data inherits a detectable signal. However they do not evaluate the radioactivity of existing in-generation watermarks to trace generated content. The closest prior work by Dubiński et al. [5], provides initial observations, but no statistical guarantees. We bridge this gap with our novel method that provides anytime-valid statistical detection of radioactivity across diverse watermarking schemes and models.

Dataset Inference (DI). DI was introduced as a framework for model ownership resolution first in supervised [24] and subsequently in the self-supervised [6] learning. The framework allows model owners to make a strong statistical claim if a given suspect model is a stolen copy of their own target (also referred to as a victim) model. To this end the behavior of the suspect model is compared to the behavior of the target model on samples of the training set. Compared to membership inference (MI) methods [34], which predict the membership of a single sample in a models training set, DI enhances the membership signal by leveraging a collection of samples. However, this procedure faces a limitation in practice, namely that DI relies on statistical testing with p-values, which is limited to a

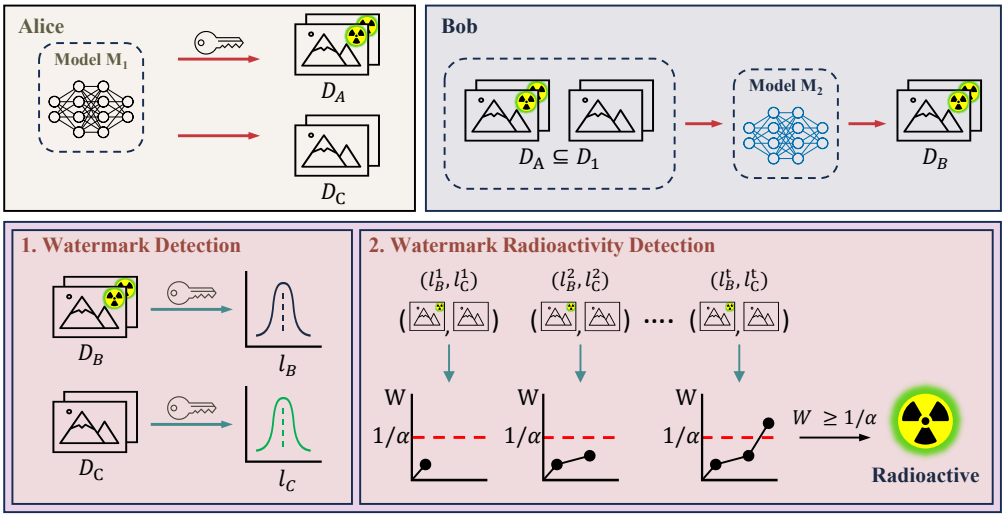


Figure 1: **Overview of our *RADIUM* method for radioactivity detection.** ***Alice*** creates model M_1. M_1 generates watermarked images D_A and clean images D_C. To train model M_2, ***Bob*** collects data D_1, which can contain a proportion of data D_A. M_2 then generates images D_B, which may carry traces of *Alice*'s watermark. *Alice* tests if *Bob* trained on her generated data: ***1. Watermark Detection.*** *Alice* uses her watermark detector on suspect images D_B generated from M_2 and on her clean images D_C generated from M_1 to obtain detection scores. ***2. Watermark Radioactivity Detection.*** *Alice* obtains the score distributions ℓ_B and ℓ_C. Our e-process tests whether D_B exhibits a significant watermark signal by accumulating wealth W. *Alice* detects radioactivity in D_B when W exceeds a given threshold $1/\alpha$.

fixed dataset size and predefined significance level. This causes issue as adding more data points, when the initial test fails, invalidates the statistical guarantees.

Sequential Testing by Betting. Sequential hypothesis tests [32, 39] frame the statistical tests as a betting game where data is drawn sequentially. Given a null hypothesis $H_0 : P \in \mathcal{P}_{null}$, which follows a distribution P and the alternative hypothesis $H_1 : P \in \mathcal{P}_{alt}$ as well as observations $Y_1, Y_2, \ldots,$ from space $\mathcal{Y}$, that are drawn i.i.d. according to P, a bettor can test H_0 by placing repeated bets on the observations Y_t, starting with initial wealth $W_0 = 1$. For testing, first a payoff function $S_t : \mathcal{Y} \to [0, \infty)$ is selected. Under the null, this function has an expected value of 1 and the future wealth after all observations is no larger than the current wealth. Then the next observation Y_t is revealed and the wealth changes by a factor of $S_t(Y_t)$. After t rounds, the wealth is described by $W_t = W_0 \prod_{i=1}^{t} S_i(Y_i)$. If H_1 is true, the wealth grows exponentially, otherwise the expected value of the wealth is at most W_0. Following [33], this process naturally leads to a sequential hypothesis test, where we reject the null if $W_t \geq 1/\alpha$, where α is the desired significance level. Ville's inequality ensures that this test controls the type-I error at level α. This sequential test naturally fits into our detection method, where we can sequentially generate samples until we find sufficient evidence against the null or until we exhaust the budget allocated to the test.

3 Our Radioactivity Detection Method

3.1 Threat Model

Setting. Consider a model provider *Alice* who deploys a generative image model M_1. To trace the provenance of the content generated with M_1, *Alice* watermarks the outputs with a watermarking scheme $\mathcal{W}$ which has the detector $\mathcal{T}$. A second party, *Bob*, fine-tunes a model M_2 on a dataset D_1 that contains images generated by M_1. *Alice* wants to determine whether *Bob* has used any of her data to train M_2. We make the following assumptions:

Model Access: *Alice* has black-box access to M_2 and can generate images using prompts, but has no access to model weights or gradients.

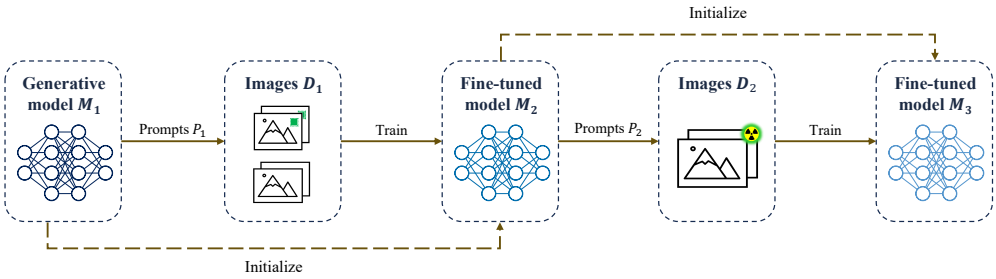


Figure 2: **Setup of the iterative model training mimicking the model collapse.** M_1 generates watermarked images D_1 with the prompts P_1. Then M_1 initializes the model M_2 and is trained on its own data D_1. The resulting model M_2 generates the radioactive dataset D_2 using the new prompts P_2. Iteratively the model M_i initializes the model M_{i+1} and is then trained on the data D_i.

Watermark Detection: *Alice* has access to her watermark detector $\mathcal{T}$, as well as to her own model M_1 and can generate watermarked and clean images with M_1. For the radioactivity detection, we use the detection scores of the detector, where higher scores mean stronger watermark detection. For example for BitMark we evaluate radioactivity using the z-score computed on the image. To provide outputs that are closer to being IID across different generative models and clean image distributions, we deploy whitening [8, 9], which computes biases in the detector on clean generated data and removes them via normalization. Therefore, we perform the PCA of the output of the watermark detectors on a set of clean generated images. Thus we obtain the mean μ and eigendecomposition of the covariance matrix $\Sigma = U\Lambda U^T$. The whitening is then applied on the logit output l of the detector, computing $y = \Lambda^{-\frac{1}{2}} U^T (l - \mu)$.

Prompt Knowledge: To detect traces of *Alice*'s watermark in *Bob*'s model, we consider two settings:

1. *supervised: Alice* knows the exact prompts of the watermarked images that *Bob* used to train M_2;
2. *unsupervised: Alice* has no knowledge about the prompts connected to the watermarked images that *Bob* used to train M_2.

3.2 Betting on Radioactivity

Overview. To help *Alice*, we sequentially generate clean images with *Alice*'s M_1 and (potentially) radioactive images with *Bob*'s M_2, using the same prompts. Then we apply the watermark detector $\mathcal{T}$ to obtain a detection score for both images. The goal is to determine whether the watermark scores of images generated by M_2 are significantly higher than those of the **clean** images generated by M_1. We therefore employ a statistical hypothesis testing by betting [33], where evidence is accumulated over round t, resulting in a wealth W_t corresponding to the significance of the test. We report significant detection when the e-value exceeds a predefined threshold. While similar results can be achieved with a t-test, this betting framework allows *Alice* to stop when sufficient evidence is gathered without committing to a fixed sample size and thus can make detection much faster.

Scoring procedure. Let $\mathcal{T}(x)$ denote a watermark score when evaluating an image x with the watermark detector $\mathcal{T}$. Higher values of $\mathcal{T}(x)$ indicate stronger evidence that the given image x contains the watermark. Depending on the watermark, $\mathcal{T}(x)$ may be a bit-wise accuracy, a z-score, or another detection statistic. Under the supervised setting, *Alice* has knowledge of the image-prompt pairs used by *Bob* during fine-tuning his model. *Alice* can generate test images using exactly these prompts, assuming that images generated from these prompts are more likely to carry the watermark signal. Under the unsupervised setting, *Alice* does not have knowledge of the prompts used by *Bob* during finetuning. Hence, she uses different prompts to generate image pairs.

Let $D_B = \{x_1, \ldots, x_{n_B}\}$ be the (potentially) radioactive images generated by the (Bob's) suspect model M_2 and $D_C = \{x'_1, \ldots, x'_{n_C}\}$ be the clean images generated by (Alice's) model M_1 without the watermarking scheme $\mathcal{W}$, where both x_i and x'_i were generated with the same prompt. We define the corresponding radioactive ℓ_B and clean ℓ_C scores as:

$$\ell_B := \{\mathcal{T}(x_i), \quad x_i \in D_B\}, \quad \ell_C := \{\mathcal{T}(x'_i), \quad x'_i \in D_C\}.$$

Let P_B and P_C denote the distributions of the radioactive ℓ_B and clean ℓ_C scores, respectively. To test whether the target model exhibits statistically significant watermark behavior, we use sequential testing by betting [33] and test the null hypothesis H_0 : *"The data was generated by a model that was trained without knowledge of the watermark."* (and also present the alternative hypothesis H_1):

$$H_0 : \ P_B \leq P_C, \qquad H_1 : \ P_B > P_C.$$

The test processes the score pairs (ℓ_B^t, ℓ_C^t), *i.e.,* observations, sequentially. At each step t, the payoff is defined as $S_t = 1 + \lambda_t u_t$, where $u_t \in \{-1, 0, 1\}$ is a bounded discrepancy score and λ_t is the wealth we bet on the observation.

Discrepancy metric. To compute u_t, we use the Kolmogorov-Smirnov (KS) witness function, estimated adaptively from all prior scores. This identifies the threshold, that maximally separates the two score distributions in the direction consistent with H_1. Under H_0 this discrepancy has zero conditional expectation given all prior observations, ensuring the validity of the test. The KS function then returns a discrepancy score $u_t \in \{-1, 0, 1\}$ based on whether the current observation agrees or disagrees with the null.

Betting strategy. For the betting fraction λ_t, we use the Online Newton Step (ONS) strategy [11]. ONS uses only the past observations to determine the betting fraction λ_t before seeing the current data point. It maintains a curvature accumulator that grows based on past observations. We provide more detailed information in Appendix A.1. A good betting strategy aims to grow wealth quickly, while controlling the risk of losing all wealth under the null.

Starting with the initial wealth $W_0 = 1$, the cumulative wealth is updated each round as $W_t = W_{t-1} \cdot (1 + \lambda_t u_t)$. This wealth can be interpreted as an *e-value* and represents the accumulated evidence against H_0. A key advantage of this process is the anytime-validity. Under H_0, the wealth process is a nonnegative supermartingale, so by Ville's inequality, rejecting H_0 when $W_t \geq 1/\alpha$ defines a valid α-level test. This enables *Alice* to monitor evidence continuously and stop when sufficient evidence is collected.

4 Experimental Evaluation

Experimental Setup. We evaluate the radioactivity of the in-generation watermarks: BitMark [17], TreeRing [41], and StableSignature [8]. We also test the post-processing watermarks: TrustMark [3], StegaStamp [37], RivaGAN [45], and SIREN [22]. The post-processing watermarks are applied to clean images generated by the M_1 model architectures, *i.e.,* the Infinity-2B model [10] and Stable Diffusion 2.1 (SD2.1) [26].We fine-tune M_1, to obtain the second model M_2, on a total of 5,000 images to mimic model adaptation and report the hyperparameters in Appendix B.1. The watermarked images all share the same embedded key and we evaluate the impact of multiple keys in Table 21. For detection, we apply whitening on RivaGan, StegaStamp, TrustMark and StableSignature using 5,000 clean generated images as reference. For the evaluation we generate 1,000 image pairs of clean/watermarked images using MS-COCO Karpathy prompts and as threshold for the e-process we select the significance level $\alpha = 10^{-5}$.

4.1 Watermarks Exhibit Disparate Radioactivity

We report the main radioactivity detection results in Table 1. We fine-tune M_1 on a total of 5,000 self-generated images, of which a specific fraction is watermarked. We then generate images with M_2 using the training prompts (supervised) and disjoint prompts (unsupervised). For the evaluation, we sample these images generated by M_2 and their counterparts generated without the watermark by M_1 to infer them through the watermark detector $\mathcal{T}$, which yields the observation (ℓ_B^t, ℓ_C^t). We sample and bet on observations until either the wealth reaches the desired significance threshold $\alpha = 10^{-5}$ or we exceed the maximum number of 1000 steps. We report the number of samples required to reach $\alpha = 10^{-5}$ or the final e-value for the failed runs (that did not reject the H_0 after the 1000 steps). To prevent reporting false-positives, induced by fine-tuning on generated images, we also fine-tune an additional M_2 on non-watermarked images generated by M_1.

As expected, for all watermarks in the same model setting, we find that M_2 trained on non-watermarked images does not reach the significant detection. Our results show that all evaluated watermarks can be detected when they make up 100% of the fine-tuning data for M_2 being Infinity-2B.

Table 1: **We find disparate radioactivity across models and watermarks.** We train M_2 on different fractions of watermarked data % generated by M_1.† indicates whitening of the decoder with the M_1-clean null. We report the stopping time, measured as the number of paired observations, at which the e-process crosses the rejection threshold $1/\alpha = 10^5$, corresponding to a significance level of $\alpha = 10^{-5}$. Faded cells report the final e-value for runs that did not reject the H_0 within 1000 steps.

Watermark	Setting	Model	Fraction of Watermarked Data						
			0%	1%	5%	10%	25%	50%	100%
BitMark	Supervised	Inf-2B	0.64	108	74	50	35	50	32
	Unsupervised	Inf-2B	0.63	90	90	48	35	41	34
SIREN	Supervised	Inf-2B	20.18	3.43	0.57	910	277	49	35
	Unsupervised	Inf-2B	12.51	10.11	0.01	522.85	395	46	36
SIREN	Supervised	SD2.1	0.03	0.19	1.42	1.13	2.08	0.03	70
	Unsupervised	SD2.1	0.01	0.05	0.02	0.02	0.02	0.57	94
RivaGAN†	Supervised	Inf-2B	0.04	0.15	0.08	3.41	172	64	33
	Unsupervised	Inf-2B	0.75	0.03	0.14	581	226	86	45
RivaGAN†	Supervised	SD2.1	0.02	0.02	0.12	0.16	1.7	0.71	457
	Unsupervised	SD2.1	0.8	0.06	0.12	0.04	0.72	0.13	813
StegaStamp†	Supervised	Inf-2B	0.04	0.07	0.1	0.21	7.84	167	33
	Unsupervised	Inf-2B	0.02	0.03	0.02	0.44	678	155	34
StegaStamp†	Supervised	SD2.1	0.09	0.1	0.03	0.07	0.05	0.25	46
	Unsupervised	SD2.1	0.35	0.01	0.06	0.14	0.51	0.02	56
TrustMark†	Supervised	Inf-2B	0.04	0.05	0.04	0.61	0.1	64.14	274
	Unsupervised	Inf-2B	0.28	0.13	0.03	0.1	0.05	0.57	170
TrustMark†	Supervised	SD2.1	0.19	0.58	0.07	0.03	0.17	0.25	0.26
	Unsupervised	SD2.1	0.06	0.13	0.05	0.04	0.14	0.06	0.11
TreeRing	Supervised	SD2.1	17.09	17.09	2.33	11.39	4.49	0.4	12.17
	Unsupervised	SD2.1	5.06	11.39	48.48	25.63	57.67	33.45	767
Stable Signature†	Supervised	SD2.1	0.02	0.03	0.13	0.03	0.91	1.52e+04	716
	Unsupervised	SD2.1	31.23	0.3	0.07	0.44	0.67	0.2	963

However only BitMark remains detectable when constituting 1% of the fine-tuning data. We observe that the watermarks are radioactive for lower fractions of watermarked data on the Infinity-2B model compared to SD2.1.

There are two potential factors that influence these differences, namely (1) different variational auto-encoders (VAEs) and (2) different training and generation processes. In Figure 4 we detect watermarks after they were encoded and decoded with the VAEs and find that the watermarks are stronger impacted by encoding with the VAE in SD2.1. The VAE in Infinity leaves fine-grained signals intact, which makes the watermark more easily learnable in the latent space and acts as prerequisite for radioactivity. Furthermore, the exposure of the watermark differs between the two architectures. Infinity-2B computes its loss over all scales of an image, and thereby over the full frequency range, whereas SD2.1 only sees each image at a single sampled timestep, where higher noise levels mask the watermark signal. Additionally, the default SD2.1 training pipeline applies random horizontal flips, which distort watermarks that are not robust to this augmentation, such as RivaGAN and StegaStamp. We analyze two different knowledge assumptions concerning the prompts of the training images of M_2, the supervised setting, and the unsupervised setting. Our main results show that prompt access has limited impact on the detection of radioactivity, with the supervised and unsupervised setting agreeing in most cases. This prompt independence shows that radioactivity is not an artifact of memorization, where the mark is learned in combination to a specific prompt, but becomes a more general behavior of the model. To verify that our findings are not an artifact of the specific backbones we study, we additionally evaluate larger and more recent diffusion models and report the results in Appendix C.4.

4.2 Iterative Model Training

This section focuses on watermark radioactivity under the lens of model collapse [1], where models are iteratively trained on their own generated content. To formalize the setting, we define a set of prompts P_i that are used by model M_i to generate data D_i. We enforce that all prompt sets are pairwise distinct, to reduce the impact of the prompt on the detection of the radioactivity. Formally,

Table 2: **Radioactivity decay over multiple training and generation runs.** We use data generated by M_{i-1} to fine-tune M_i where M_i is initialized with M_{i-1}. Each row's bracketed percentage (%) gives the fraction of watermarked data used to train M_2. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	M_2	M_3	M_4	M_5	M_6
BitMark (5%)	Supervised	Inf-2B	74	164	165	386	0.4
	Unsupervised	Inf-2B	90	82	236	545	67.88
BitMark (100%)	Supervised	Inf-2B	32	34	32	34	34
	Unsupervised	Inf-2B	34	32	34	33	40
StegaStamp (50%)†	Supervised	Inf-2B	167	180	130	159	253
	Unsupervised	Inf-2B	155	134	159	237	392
StegaStamp (100%)†	Supervised	Inf-2B	33	34	34	34	32
	Unsupervised	Inf-2B	34	32	34	36	34
SIREN (50%)	Supervised	Inf-2B	49	64	64	80	107
	Unsupervised	Inf-2B	46	75	75	74	85
SIREN (100%)	Supervised	Inf-2B	35	39	43	43	42
	Unsupervised	Inf-2B	36	41	43	39	51
StegaStamp (100%)†	Supervised	SD2.1	46	65	122	414	378
	Unsupervised	SD2.1	56	105	121	584	20.62
SIREN (100%)	Supervised	SD2.1	70	7.35	–	–	–
	Unsupervised	SD2.1	94	14.74	–	–	–

we use M_i and P_i to generate dataset D_i and then fine-tune M_i on D_i to obtain M_{i+1}. We provide a visualization of the setup in Figure 2. Based on our findings in Table 1 we focus on BitMark, StegaStamp, and SIREN, since they showed higher levels of radioactivity. For $i > 2$ we train the model on 100% of the data generated by its predecessor. Only M_2 is trained on $p\%$ of watermarked and $1 - p\%$ of clean data. Our results for the iterative model training in Table 2 show that the watermarks on Infinity-2B experience no or hardly any degradation across multiple fine-tuning iterations. On the other hand StegaStamp on SD2.1 decays gracefully over multiple iterations, whereas SIREN is no longer detectable in the M_3 model. SIREN shows clear differences in the behavior on Infinity-2B and SD2.1, which is likely related to the resistance of the watermark against encoding and decoding, analyzed in Figure 4. We provide examples of the output of the SD2.1 in Figure 7 and Infinity-2B in Figure 8. To separate the radioactive traces between inheritance from the weights of M_i and its generated data, we re-initialize M_3 from the original M_1 rather than M_2 in Table 15. This re-initialization from clean weights leaves the stopping times essentially unchanged, so the signal is rather carried by the generated data than the prior models weights.

4.3 Cross Architecture Analysis

The results from the two previous sections, show that some watermarks can be radioactive and enable content tracing even through multiple model iterations on the **same model architecture**. However in most case the model trained on the watermark images generated by M_1 will have a different architecture. Evaluating the watermarks across architectures reveals a shift for BitMark, where the detector scores of a clean-trained M_2 of a different architecture is not distributed like those of the clean generations of M_1. While the per-sample shift is minor, the e-process aggregates it, and can produce false positives even without any watermarked training data. To guarantee the validity of our evaluation, we additionally assume access to clean generated images of M_2 to calibrate the decision threshold of BitMark. This failure case shows inherent limitations of existing watermarks. We further evaluate this in Appendix C.5. For the cross-architecture case we focus on settings where a larger model M_1 is used to generate data for a smaller model M_2. Our results in Table 3 indicate that the cross-architecture case is much harder than the same architecture case, where the watermarks are detected only when making up high fractions of the data and many samples are evaluated. Notably, this does not hold relative to SD2.1 as M_1, as SD2.1 trained on Infinity-2B generations learns StegaStamp and SIREN at 50% and TrustMark at 100% of the data, whereas from its own generations it learns StegaStamp and SIREN only at 100% and TrustMark not at all, indicating that the source images also influence how learnable a watermark is.

Table 3: **Detection performance across model architectures.** We train M_2 on different fractions of watermarked data % generated by M_1, where M_1 and M_2 use different architectures. ‡ indicates scoring against the M_2 architecture's own clean baseline rather than M_1's. The reported values follow the same format as in Table 1.

Watermark	Setting	M_1	M_2	Fraction of Watermarked Data						
				0%	1%	5%	10%	25%	50%	100%
BitMark‡	Supervised	Inf-2B	SD2.1	0.05	76.92	0.28	0.03	980	182	80
	Unsupervised	Inf-2B	SD2.1	0.07	1.18	0.06	0.03	455	131	54
	Supervised	Inf-2B	SD1.4	1.85	0.69	0.33	0.42	1.59	282	58
	Unsupervised	Inf-2B	SD1.4	33.51	0.21	0.22	4.5	1.38e+03	466	107
RivaGAN†	Supervised	Inf-2B	SD2.1	0.02	0.01	0.02	0.07	0.25	0.5	292.26
	Unsupervised	Inf-2B	SD2.1	0.04	0.11	0.04	0.02	0.04	0.28	677
	Supervised	Inf-2B	SD1.4	49.36	0.05	0.21	0.28	1.82	15.76	1.77e+03
	Unsupervised	Inf-2B	SD1.4	7.1	0.41	0.25	0.13	15.23	200.02	3.94e+03
	Supervised	SD2.1	SD1.4	0.21	0.23	0.05	0.03	1.22	6.47	838
	Unsupervised	SD2.1	SD1.4	0.24	0.24	0.03	0.04	0.09	0.34	850.08
StegaStamp†	Supervised	Inf-2B	SD2.1	0.07	0.03	0.04	0.04	0.41	154	41
	Unsupervised	Inf-2B	SD2.1	0.04	0.41	0.08	0.02	0.09	155	51
	Supervised	Inf-2B	SD1.4	0.25	0.08	0.06	0.04	0.17	0.02	41
	Unsupervised	Inf-2B	SD1.4	0.16	0.04	0.08	0.07	0.01	0.02	46
	Supervised	SD2.1	SD1.4	0.07	0.11	0.05	0.01	0.01	0.26	529
	Unsupervised	SD2.1	SD1.4	0.03	0.07	0.24	0.14	0.01	0.02	733
TrustMark†	Supervised	Inf-2B	SD2.1	0.04	0.05	0.21	0.04	0.03	0.02	60
	Unsupervised	Inf-2B	SD2.1	0.05	0.04	0.7	0.08	0.03	0.17	101
	Supervised	Inf-2B	SD1.4	0.04	0.49	0.5	0.54	0.18	1.15	0.09
	Unsupervised	Inf-2B	SD1.4	0.01	0.54	0.36	0.03	0.02	0.55	0.04
	Supervised	SD2.1	SD1.4	0.1	0.05	0.06	0.15	0.05	0.34	0.36
	Unsupervised	SD2.1	SD1.4	0.06	0.02	0.04	0.78	0.03	0.02	0.21
SIREN	Supervised	Inf-2B	SD2.1	0.91	0.11	5.41	0.71	0.22	524	100
	Unsupervised	Inf-2B	SD2.1	0.73	0.71	1.23	0.21	0.08	449	100
	Supervised	Inf-2B	SD1.4	0.03	2.07	5.41	2.28	748	283	97
	Unsupervised	Inf-2B	SD1.4	1.13	1.76	0.12	0.11	0.15	271	84
	Supervised	SD2.1	SD1.4	1.03	0.11	0.63	0.1	0.18	0.12	79
	Unsupervised	SD2.1	SD1.4	0.03	0.01	0.01	0.05	0.04	0.06	121

4.4 Radioactivity Removal Robustness

Given the knowledge about watermark radioactivity, *Bob*, who owns M_2, now wants to make sure that his model is not radioactive, to avoid *Alice* detecting he illegitimately trained on her data. To remove any radioactive traces, *Bob* collects carefully curated data, ensuring that the curated dataset contains no watermarked images. He then performs a cleaning step, by carefully fine-tuning the model on the clean data. Therefore, we analyze the models M_2 trained on different fractions of watermarked data. Our results in Table 4, show that the radioactivity of current watermarks is very fragile and decreased drastically after additional fine-tuning on clean data. Only BitMark, when using 100% of the fine-tuning data, is radioactive to M_3. We provide additional results for the radioactivity robustness under an adversary that applies image augmentations in Appendix C.2. We find that augmentations before or during fine-tuning have the strongest negative effect on radioactivity, as they disrupt the watermark signal, before the model can learn it.

4.5 Hyperparameter Ablation

We analyze different hyperparameters that influence the radioactivity of watermarks in downstream models. Table 16 shows that a higher learning rate generally causes the e-process to stop earlier, consistent with the watermark being more strongly represented in the model's outputs. The number of fine-tuning epochs has a setup-dependent effect (Table 18): on the Infinity backbone, increasing the number of fine-tuning epochs markedly reduces detection, compared to a minor influence on the SD2.1 model. In contrast, the effective batch size has little impact on radioactivity (Table 19) - the watermark signal is not washed out by computing the gradient on a batch. In Table 17 we analyze the impact of changing the strength of the watermark on the radioactivity. We focus on BitMark and SIREN and change their strength in proportion to their default values ($\beta = 1.5$ for SIREN and $\delta = 2$ for BitMark). The results show that a higher watermark strength reduces the number of samples required for significant detection for SIREN. Notably, a reduced watermark strength results in the watermark no longer being detectable.

Two further ablations analyze the impact of the amount of data and the continued training on the observed radioactivity. Doubling the fine-tuning set to 10k images (Table 8) leaves detection at 25%

Table 4: **Clean fine-tuning removes radioactive traces in nearly all settings.** M_1 is the pretrained base. M_2 is fine-tuned on data generated by M_1, with the watermark fraction shown in brackets (%); M_3 further fine-tunes M_2 on clean data. For evaluation of this same model setting we leverage clean data from M_1. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	M_2	M_3
BitMark (5%)	Supervised	Inf-2B	74	3.17
	Unsupervised	Inf-2B	90	0.08
BitMark (100%)	Supervised	Inf-2B	32	293
	Unsupervised	Inf-2B	34	274
StegaStamp (100%)†	Supervised	Inf-2B	33	0.99
	Unsupervised	Inf-2B	34	4.21e+03
StegaStamp (100%)†	Supervised	SD2.1	46	0.02
	Unsupervised	SD2.1	56	0.01
SIREN (100%)	Supervised	Inf-2B	35	8.75
	Unsupervised	Inf-2B	36	202.17
SIREN (100%)	Supervised	SD2.1	70	1.69
	Unsupervised	SD2.1	94	0.08

and 100% essentially unchanged and roughly doubles the samples required at 5%, so it is the fraction of watermarked data that stronger governs radioactivity. Fully training a model from scratch *i.e.,* random M_1 weights trained on clean MS-COCO and a randomly initialized M_2 trained on 100% watermarked outputs of M_1 (Table 9), still yields a radioactive StegaStamp, whereas SIREN is not learned. Radioactivity is thus likely not an artifact of starting from pretrained weights, but rather a property of the data.

4.6 Core Factors for Radioactive Watermarks

This section formalizes the core factors making up a radioactive watermark and supports them with our experimental evidence. **(1) Robustness to the latent encoding is a prerequisite.** No watermark that is destroyed by the encoding-decoding cycle of the downstream model's VAE is radioactive in that model (Figure 4). This is the most predictive property we observe, and explains why StableSignature, RivaGAN, and SIREN are strongly weakened on SD2.1, whereas Infinity-2B leaves fine-grained signals intact. **(2) The mark must be reused across samples rather than tied to individual content.** A content-dependent signal appears to the downstream model as sample-specific noise, whereas a signal that reoccurs across the training set becomes a learnable regularity. This is especially apparent in our results with BitMark when attenuating the signal in Table 20. The same also shows when we increase the number of keys and the signals interfere with each other in StegaStamp (Table 21). **(3) The mark must survive the training process itself.** Augmentations applied before or during fine-tuning suppress radioactivity far more effectively than the same augmentations applied afterwards as we explore in Table 11. So a radioactive watermark must be robust to the standard augmentation pipeline, not only to post-hoc removal attacks.

5 Conclusions

We introduced a statistical method based on sequential betting that enables the analysis of the radioactivity of watermarks in downstream models. We investigated the radioactivity and radioactive decay of post-processing and in-generation watermarks spanning three different model architectures in the diffusion and autoregressive domain. Our results demonstrate that current watermarking techniques lack reliable detection in downstream models and can not guarantee long-term provenance tracing of their generated content. We show how robustness against encoding is a prerequisite for radioactivity. Our ablation studies identify several factors influencing the radioactivity of image watermarks, such as the model architecture, learning rate and watermark strength. By providing a unified statistical method for radioactivity detection across different image watermarks, we enable watermark designers to evaluate watermark radioactivity through model derivatives.

Acknowledgments

Adam Dziedzic received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101299753).

References

[1] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ShjMHfmPs0.

[2] Jerone Andrews, Dora Zhao, William Thong, Apostolos Modas, Orestis Papakyriakopoulos, and Alice Xiang. Ethical considerations for responsible data curation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=Qf8uzIT1OK.

[3] Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Robust watermarking and watermark removal for arbitrary resolution images. In *IEEE International Conference on Computer Vision (ICCV)*, October 2025.

[4] Jan Dubiński, Antoni Kowalczuk, Franziska Boenisch, and Adam Dziedzic. Cdi: Copyrighted data identification in diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 18674–18684, 2025.

[5] Jan Dubiński, Michel Meintz, Franziska Boenisch, and Adam Dziedzic. Are watermarks for diffusion models radioactive? In *The 1st Workshop on GenAI Watermarking*, 2025. URL https://openreview.net/forum?id=gtXbVRMwQh.

[6] Adam Dziedzic, Haonan Duan, Muhammad Ahmad Kaleem, Nikita Dhawan, Jonas Guan, Yannis Cattan, Franziska Boenisch, and Nicolas Papernot. Dataset inference for self-supervised models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=CCBJf9xJo2X.

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[8] 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 (ICCV)*, pages 22466–22477, October 2023.

[9] Enoal Gesny, Eva Giboulot, Teddy Furon, and Vivien Chappelier. Guidance watermarking for diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5ifzhjMCKq.

[10] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[11] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, 69(2–3):169–192, December 2007. ISSN 0885-6125. doi: 10.1007/s10994-007-5016-8. URL https://doi.org/10.1007/s10994-007-5016-8.

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.

18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[16] Nikola Jovanović, Ismail Labiad, Tomas Soucek, Martin Vechev, and Pierre Fernandez. Watermarking autoregressive image generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hVdD72iom4.

[17] Louis Kerner, Michel Meintz, Bihe Zhao, Franziska Boenisch, and Adam Dziedzic. Bitmark: Watermarking bitwise autoregressive image generative models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VSir0FzFnP.

[18] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

[19] Jan Kociszewski, Hubert Jastrzębski, Tymoteusz Stępkowski, Filip Manijak, Krzysztof Rojek, Franziska Boenisch, and Adam Dziedzic. SERUM: Simple, efficient, robust, and unifying marking for diffusion-based image generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AiBUm6iKBf.

[20] Antoni Kowalczuk, Jan Dubiński, Franziska Boenisch, and Adam Dziedzic. Privacy attacks on image autoregressive models. In *Forty-Second International Conference on Machine Learning (ICML)*, 2025.

[21] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

[22] Boheng Li, Yanhao Wei, Yankai Fu, Zhenting Wang, Yiming Li, Jie Zhang, Run Wang, and Tianwei Zhang. Towards Reliable Verification of Unauthorized Data Usage in Personalized Text-to-Image Diffusion Models . In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 2564–2582, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. doi: 10.1109/SP61157.2025.00073. URL https://doi.ieeecomputersociety.org/10.1109/SP61157.2025.00073.

[23] Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mDKxlfraAn.

[24] Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=hvdKKV2yt7T.

[25] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. LLM dataset inference: Did you train on my dataset? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[27] Lorenzo Rossi, Bartłomiej Marek, Franziska Boenisch, and Adam Dziedzic. Natural identifiers for privacy and data audits in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=doaAUf9Pi7.

[28] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Radioactive data: tracing through training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8326–8335. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sablayrolles20a.html.

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[30] Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qGiZQb1Khm.

[31] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[32] Glenn Shafer. The language of betting as a strategy for statistical and scientific communication, 2019. URL https://arxiv.org/abs/1903.06991.

[33] Shubhanshu Shekhar and Aaditya Ramdas. Nonparametric two-sample testing by betting. *IEEE Transactions on Information Theory*, 70(2):1178–1203, 2024. doi: 10.1109/TIT.2023.3305867.

[34] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. doi: 10.1109/SP.2017.41.

[35] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022): 755–759, 2024.

[36] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

[37] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2126, 2020.

[38] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[39] Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021. doi: 10.1214/20-AOS2020. URL https://doi.org/10.1214/20-AOS2020.

[40] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N. Metaxas, and Shiqing Ma. DIAGNOSIS: Detecting unauthorized data usages in text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=f8S3aLm0Vp.

[41] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 58047–58063. Curran Associates, Inc., 2023.

[42] Sierra Wyllie, Ilia Shumailov, and Nicolas Papernot. Fairness feedback loops: training on synthetic data amplifies bias. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2113–2147, 2024.

[43] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation, 2024. URL https://arxiv.org/abs/2411.00776.

[44] Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=DztaBt4wP5.

[45] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention, 2019. URL https://arxiv.org/abs/1909.01285.

[46] Bihe Zhao, Pratyush Maini, Franziska Boenisch, and Adam Dziedzic. Unlocking post-hoc dataset inference with synthetic data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=a5Kgv47d2e.

[47] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

A Extended Background

A.1 Betting Strategy

The betting strategy determines the size of the bet to wager on the next sample, based on the already observed discrepancy between the two distributions. In our method we leverage the Online Newton Step (ONS) strategy.

Online Newton Step (ONS). First we use the discrepancy metric (in our case derived from Kolmogorov-Smirnov) to compute the bounded discrepancy score $u_t \in \{-1, 0, 1\}$. We then update a one-dimensional curvature accumulator A_t and calculate the subsequent betting amount λ_{t+1} for round $t + 1$:

$$z_t = \frac{u_t}{1 + \lambda_t u_t}, \quad A_t = A_{t-1} + z_t^2,$$

$$\lambda_{t+1} = \Pi_{[0,\,\lambda_{\max}]}\left(\lambda_t + \frac{2}{2 - \log 3} \cdot \frac{z_t}{A_t}\right),$$

with initializations $\lambda_1 = 0$ and $A_0 = 1$. Here $\Pi_{[0,\,\lambda_{\max}]}$ denotes projection onto the permissible, bounded stake interval. Restricting λ_t to $[0, \lambda_{\max}]$ rather than $[-\lambda_{\max}, \lambda_{\max}]$ enforces the one-sided nature of the test: the bettor only ever places stakes in the direction of H_1, consistent with the hypothesis that radioactive images produce stochastically higher watermark scores than clean images. While higher values of $\lambda_{\max}$ permit more aggressive betting, the upper bound is mathematically necessary to prevent the wealth process from reaching zero (ruin). We set $\lambda_{\max} = 0.5$ throughout. A good betting strategy aims to grow wealth quickly when the alternative hypothesis is true, while controlling the risk of losing all wealth under the null. ONS achieves this through its curvature accumulator A_t, which grows with the sum of squared past gradients and provides a data-dependent learning rate that achieves logarithmic regret without requiring prior knowledge of the problem difficulty.

A.2 Extended Related Work

Watermarking. The post-processing watermarks differ primarily in the space they are applied in and the robustness that they target. StegaStamp [37] is designed to embed signals that survive strong perturbation pipelines and can be detected in realistic physical detection scenarios. It incorporates a specific robustness training pipeline and trades visual fidelity for persistence. RivaGAN [45] was proposed for video and leverages an attention-based mechanism that hides messages in content-dependent regions. This design makes it robust against video-processing operations like compression. Additionally, it is resolution-agnostic by construction, as its encoder has no bottleneck. SIREN [22] is a watermarking method designed to be learned by models fine-tuned on watermarked data. Therefore the watermark leverages a proxy student model and optimizes the watermark embedder, such that the watermark *decreases* the loss of the proxy model on the watermarked images, resulting in the watermark providing helpful features for the model and being learned more easily. Finally, TrustMark [3] targets broad robustness to common image edits, while preserving high image quality. To this end it incorporates a focal frequency loss (FFL) preserving the high frequency content of the images. Compared to RivaGAN and StegaStamp, which disrupt high-frequency content of the images, the FFL results in TrustMark being primarily applied in lower frequencies.

The evaluated in-generation watermarks differ substantially in the watermark embedding techniques. StableSignature [8] fine-tunes a latent-models decoder, so that every decoded image carries the watermark. This approach hardly affects the generation process, resulting in only minor changes to the generated images while staying robust under strong edits. TreeRing [41] is a watermark specifically designed for diffusion models. It applies a ring in the low frequencies of the initial noise in the diffusion process, which makes it robust against geometric perturbations, such as cropping, rotating, and flipping. However, its detection is highly expensive, as it requires a full DDIM-inversion from the image space back to the noise space. Finally, BitMark [17] is a watermark specifically designed for bit-wise generative models. These models produce images by generating bit-strings of varying length, sampling from a vocabulary consisting of bitstrings. BitMark influences the sampling process by separating the vocabulary into a green list and a red list of bitstrings with length n and biases the generation process such that bitstrings from the green list are sampled more frequently.

A.3 Image and Text Radioactivity

Text Watermarking. Existing work on LLM radioactivity [30] detects watermarks through a principled analysis of token-level generation in LLMs. This line of work focuses on watermarking methods that modify the decoding procedure [18]. Specifically, LLM decoding can be viewed as taking a context sequence $(x^{(-C)}, \ldots, x^{(-1)}) \in \mathcal{V}^C$, where $\mathcal{V}$ denotes the model vocabulary and C the context length. Given this context, the LLM outputs a logit vector $l \in \mathbb{R}^{|\mathcal{V}|}$, which is converted into next-token probabilities via $p = \mathrm{softmax}(l) \in [0,1]^{|\mathcal{V}|}$. The next token $x^{(0)}$ is then sampled from this distribution. Watermarking schemes influence this generation process using a secret key: the key and the current context are provided to a hash function, whose output serves as a seed s for a random generator. This seed determines how the watermark biases the next-token distribution, for example by increasing the sampling probabilities of tokens in a designated green list [18]. For detection, the generated text is re-tokenized, the corresponding contexts are reconstructed, and a watermark score is assigned to each token.

Image vs. Text Radioactivity. Sander et al. [30] make two key design choices in their radioactivity detection scheme. First, to improve detection, they introduce a filter ϕ that restricts the analyzed token stream to k-grams likely to have appeared during training. Tokens generated by M_2 are scored only when their context is not filtered out by ϕ. Second, they apply token de-duplication: a token is scored only if its context has not been observed previously during detection. Together, these two design choices yield faster and unbiased detection. However, these insights do not directly extend to the image domain. Unlike text watermarking methods that modify the sampling or decoding procedure [18], most image watermarking methods do not influence the generative model's sampling process directly, with a few exceptions [16, 17]. Instead, image watermarks are typically applied either before generation, e.g., to the initial noise in DMs [19, 41], or after generation, e.g., to generated outputs [3, 37]. Furthermore, image watermarks lack an inherent sequential context analogous to text k-grams, making it unclear how to de-duplicate watermark signals or remove context-induced biases in a comparable way. As a result, radioactivity in the image domain differs substantially from radioactivity in the text domain. In addition, image-specific transformations, such as changes in resolution, compression, cropping, or other augmentations, can further degrade watermark signals and negatively affect radioactivity detection.

B Experimental Setup

B.1 Hyperparameters

Table 5: **Hyperparameters used for model training.**

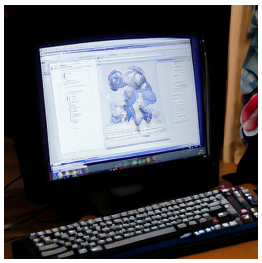
	Optimizer	Learning Rate	Batch Size	Epochs
Stable Diffusion 2.1	Adam	1×10^{-5}	4	5
Stable Diffusion 1.4	AdamW	1×10^{-5}	4	5
Stable Diffusion 3.5	AdamW	5×10^{-6}	4	5
Infinity-2B	AdamW	6×10^{-4}	4	2
FLUX.1-dev	AdamW	5×10^{-6}	4	5

For Stable Diffusion 2.1, we used Adam optimizer with a learning rate of 1×10^{-5} and fine-tune for 5 epochs with batch size of 4, following CLiD [44] with horizontal-flipping turned on by default. We fine-tune Infinity-2B [10] following their official implementation, using the AdamW optimizer with a learning rate of 6×10^{-4} for 2 epochs. This learning rate is lower than the default configuration (6×10^{-3}), as we adjusted for our specific fine-tuning task. The input resolution is set to 1024×1024.

Finally, we adjusted the noise strength parameter. While the default configuration uses noise strength of $\sigma = 0.3$, we noticed this degraded our image generation quality. To address this, we ran tests fine-tuning Infinity-2B for 1 epoch for the noise strengths 0.1, 0.2 and 0.3 (see Figure 3). After visually comparing the outputs, we selected 0.1 as it produced best images among the three settings.



(a) $\sigma = 0.1$ image.



(b) $\sigma = 0.2$ image.



(c) $\sigma = 0.3$ image.

Figure 3: **Effect of Noise Strength on Infinity-2B.** Prompt: *"A young boy standing in front of a computer keyboard."*

B.2 Compute Environment

Our experiments are performed on Ubuntu 22.04 with NVIDIA A100 Graphics Cards with 40GB and 80GB of memory. To run our experiments we used CUDA Version 12.5 and Python 3.12.4.

B.3 Computational Cost

The radioactivity analysis relies on multiple different components, most importantly: data generation, watermark detection and the e-process. Running the e-process when all 1,000 samples are generated and detected requires only 0.2 seconds. However, the image generation and watermark detection cost do not depend on *RADIUM*, but on the generative model and watermark scheme. Therefore we provide the generation and detection time in seconds per image in Table 6 and Table 7.

Table 6: **Image Generation Cost.** We compute the speed for image generation for single image batch sizes for the two generative backbones. We report the generation in seconds / image averaged across 1,000 images.

Model	SD2.1	Inf-2B
Generation (seconds / image)	1.38	1.97

Table 7: **Watermark Detection Cost.** We compute the speed for watermark detection for single image batch sizes for the different watermarking schemes. We report the detection speed in seconds / image averaged across 1,000 images.

Model	StegaStamp	TrustMark	RivaGAN	SIREN	StableSignature	TreeRing	BitMark
SD2.1	0.006	0.008	0.009	0.009	0.009	1.198	0.134
Inf-2B	0.01	0.013	0.028	0.02	-	-	0.134

B.4 Discussion and Limitations

Training large-scale generative models, such as Infinity-2B [10] or Stable Diffusion 2.1 [26], from scratch is prohibitively expensive, especially for many training-generation runs or dataset compositions. We therefore do not perform full pretraining experiments at this scale. Instead, we study a fine-tuning setting as a proof of concept, following prior analyses of model collapse [35] and LLM radioactivity [30]. This setting allows us to systematically evaluate watermark radioactivity across multiple training-generation runs while keeping the experimental setup computationally feasible.

This choice, however, limits the extent to which our results transfer to large-scale pretraining regimes. Fine-tuning may preserve, amplify, or attenuate watermark signals differently from training from scratch, depending on the amount of synthetic data, the strength of the watermark, the model architecture, and the optimization procedure. Similarly, our conclusions are based on the set of watermarking methods, model architectures, and hyperparameters considered in our experiments. Other watermarking schemes or parameters may exhibit different levels of persistence.

To approximate radioactivity at a larger scale and closer to pre-training, we perform two additional ablations. (1) We double the number of training samples, evaluating the radioactivity of the models when trained on 10,000 samples and (2) we initialize SD2.1 from scratch and first train on clean natural MS-COCO data for 5 epochs to obtain M_1 and then train a new randomly initialized model on the watermarked outputs from M_1. We report our results in Table 8 and Table 9 respectively. We find that a larger fine-tuning set has limited impact on the observed radioactivity, with only the 5% setting requiring roughly twice as many samples. Training from a random initialization results in StegaStamp remaining radioactive, whereas SIREN can no longer be learned. Beyond these ablations, we expect the core factors identified in Section 4.6 to also govern radioactivity under downstream retraining or web-scale data reuse. A watermark that does not survive the encoding and training pipeline of a model is unlikely to be learned by it at a larger scale either. Furthermore, as our results are evaluated in a controlled setting, watermarks that are not radioactive here, will likely also not be radioactive under more difficult realistic use cases.

Table 8: **Detection performance with 10,000 samples.** Cells report the stopping time (in number of paired observations) at which the e-process crossed the rejection threshold $1/\alpha = 10^5$ (significance level $\alpha = 10^{-5}$). The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Fraction of Watermarked Data			
			0%	5%	25%	100%
BitMark	Supervised	Inf-2B	0.0358	143	53	34
	Unsupervised	Inf-2B	0.0438	208	51	33
StegaStamp†	Supervised	SD2.1	0.099	0.0225	0.110	46
	Unsupervised	SD2.1	0.79	0.135	0.105	38

Table 9: **Detection performance for SD2.1 trained from scratch.** We train a randomly initialized M_2 on watermarked data generated by the randomly initialized and then trained M_1. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Fraction of Watermarked Data
			100%
StegaStamp†	Supervised	SD2.1	37
	Unsupervised	SD2.1	40
SIREN	Supervised	SD2.1	1.71
	Unsupervised	SD2.1	0.156

C Additional Results

C.1 VAE Robustness

For the watermarks to be learnable by the generative model, the watermark has to persist both in the latent space of that model and be detectable after decoding. The generative models that we analyze operate in a latent space created by a VAE consisting of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$. To analyze the robustness of the watermark, we take 5000 watermarked samples, obtain their reconstruction via $\mathcal{D}(\mathcal{E}(x)) = x'$ and finally detect the watermark on x'. We plot the TPR@1%FPR of the watermark detection of x' in Figure 4 and find that StableSignature, RivaGAN and SIREN when embedded on SD2.1 images and reconstructed by the SD2.1-VAE are strongly weakened.

C.2 Radioactivity Robustness

Adversarial Robustness. We additionally consider an adversary, that suspects *Alice* to embed a radioactive watermark signal, and that tries to remove it through augmentations. As the adversary must preserve utility as well as fast inference, we consider common image augmentations such as cropping to 90%, JPG compression to 75% quality, and horizontal flipping. We consider three

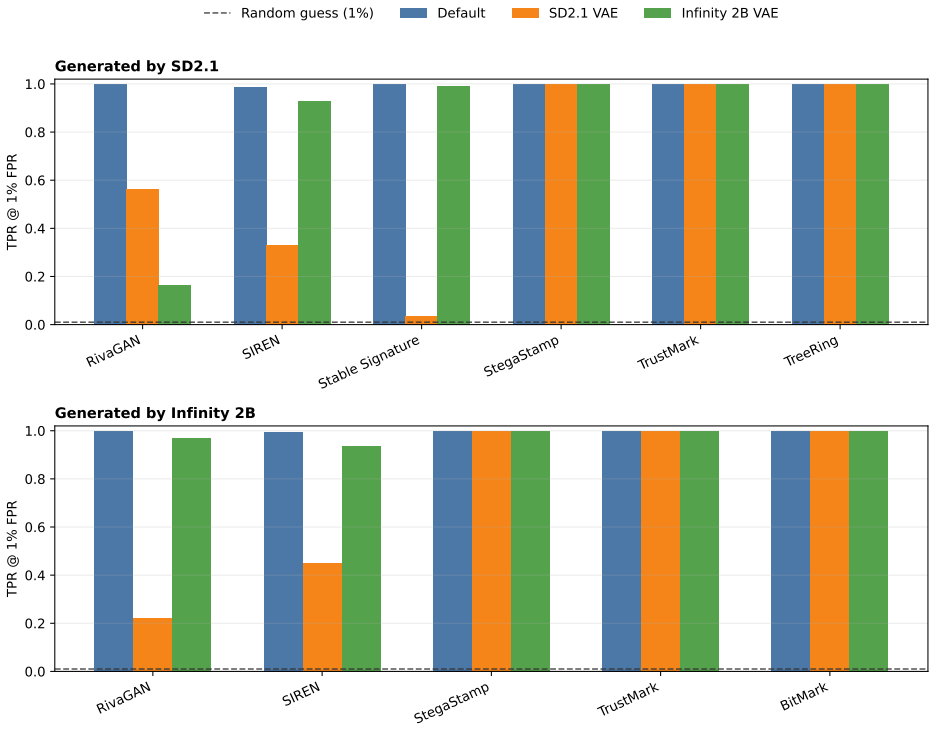


Figure 4: **Watermark robustness against VAE encoding-decoding.**

different attack stages, namely (1) on the M_1 generated images before fine-tuning, (2) during fine-tuning and (3) on the M_2 generated images. We provide the results in Table 10, Table 11 and Table 12 respectively. We find that augmentations before or during fine-tuning are more effective than augmentations on the generated data after fine-tuning. Fine-tuning on the watermarked data enables the model to reproduce the watermark signal, whereas fine-tuning on the attacked images reduces the potential signal that can be picked up. This makes attacks before or during fine-tuning much more effective.

Table 10: **Pre-fine-tuning Robustness.** We apply crop, horizontal flip and JPEG compression on the data generated by M_1 before fine-tuning M_2. The reported values follow the same format as in Table 1.

Watermark	Model	Attack	Setting	Fraction of Watermarked Data			
				5%	10%	25%	100%
BitMark	Inf-2B	Clean	Supervised	74	50	35	32
			Unsupervised	90	48	35	34
		Combined	Supervised	0.164	0.0688	635	75
			Unsupervised	0.321	0.907	0.0153	83

Lower Watermarking Fractions. A central question for watermarking-based radioactivity detection is how small a fraction of watermarked data is sufficient to leave a detectable trace in the downstream model. To probe this regime, we extend the BitMark sweep on Inf-2B to fractions below the 1% setting reported in Table 1. As shown in Table 13, the number of samples required to stop the e-process grows sharply as the fraction decreases, and below 1% watermark fraction in the training data the signal is no longer significant under either supervised or unsupervised detection, neither at 0.1% (i.e., 5 images) nor at 0.02% (i.e., 1 image). This indicates a practical lower bound, for BitMark

Table 11: **During-training Robustness under Horizontal Flipping.** Cells report the stopping time (in number of paired observations) at which the e-process crossed the rejection threshold $1/\alpha = 10^5$ in the unsupervised setting; faded cells report the final e-value of runs that did not reject. We indicate active horizontal flipping (Flip) by •.

Watermark	M1	M2	Flip	Fraction of Watermarked Data		
				25%	50%	100%
RivaGAN†	Inf-2B	Inf-2B	○	226	86	45
			•	0.32	191	69
BitMark	Inf-2B	Inf-2B	○	35	41	34
			•	35	45	34

Table 12: **Post-training Robustness.** We apply the attacks on the images generated by M_2. The reported values follow the same format as in Table 1.

Watermark	Model	Attack	Setting	Fraction of Watermarked Data					
				1%	5%	10%	25%	50%	100%
BitMark	Inf-2B	Clean	Supervised	108	74	50	35	50	32
			Unsupervised	90	90	48	35	41	34
		Crop	Supervised	2.51e+04	364.804	353	52	112	34
			Unsupervised	3.934	78.839	687	55	93	34
		Horizontal Flip	Supervised	1.642	3.689	0.473	74	168	36
			Unsupervised	2.327	1.316	0.667	83	167	38
		JPEG	Supervised	148	96	58	37	50	34
			Unsupervised	188	126	66	37	52	34
		Combined	Supervised	1.965	0.232	0.096	261	401	68
			Unsupervised	1.359	0.115	0.431	383	570	70

in this setting, of roughly 1% watermarked training data (i.e., 50 images) for reliable downstream detection.

Table 13: **Extended BitMark watermark fraction ablation.** Continuation of Table 1, reporting BitMark detection at lower watermarked-data fractions.

Watermark	Setting	Model	Fraction of Watermarked Data								
			0%	0.02%	0.1%	1%	5%	10%	25%	50%	100%
BitMark	Supervised	Inf-2B	0.64	0.27	0.13	108	74	50	35	50	32
	Unsupervised	Inf-2B	0.63	0.05	0.11	90	90	48	35	41	34

Timestep analysis. In the same architecture setting, the bias of BitMark, which is added during generation, is picked up by the Infinity model, similar to the bias of the KGW watermark in language models. Further differences arise due to the training paradigm. In Infinity, all frequencies (scales) of an image are processed per loss calculation, whereas in SD2.1, only one out of 1000 timesteps is seen per sample per epoch. We further analyze SD2.1 by training the model with noise sampled only from specific timesteps. Specifically, we train SD2.1 on 100% watermarked StegaStamp images and report the mean bit accuracy in M_2. The results show that the watermark becomes much more radioactive and easier to detect in M_2 when the model is trained only on the lower timestep range $[0, 500)$, compared with training across the full timestep range.

Inheritance vs. Watermark. Our fine-tuning setting in both the single-hop as well as multi-hop cases initialize M_{i+1} by its predecessor M_i. This opens the question, whether radioactivity in the case of $i \geq 2$ is a property of the model weights, or a property of the data generated by M_i. To evaluate this, we train M_3, but instead of initializing the model with its prior M_2 (fine-tuned on watermarked data from M_1), we initialize it with the original model architecture M_1. Our results in Table 15 show that for both model and watermark combinations, the initialization with M_1 has an impact on the

Table 14: **Effect of training timesteps on radioactivity in SD2.1.** We train SD2.1 on 100% watermarked StegaStamp images using noise sampled from specific timestep ranges and report the stopping time and mean bit accuracy for the unsupervised setting in M_2.

Model	Training Timesteps	Stopping Time	Mean Bit Accuracy (M_2)
SD2.1	$[0, 1000)$	56	0.870
SD2.1	$[0, 500)$	33	0.924
SD2.1	$[500, 1000)$	0.011	0.492

radioactivity of the watermarks (in 100% watermarked data used for fine-tuning: slightly for Inf-2B and more for SD2.1). This is to be expected, as the model now no longer carries the initial signal it obtained from the originally watermarked data, but still learns the weaker signal of the generations from M_2. SIREN becomes detectable, one potential reason is the limited perturbations of the images due to re-initialization with non-fine-tuned weights. The results show that while inheritance through initialization is one factor of radioactivity, the core impact stems from the watermark itself.

Table 15: **Radioactivity is a property of the data and the model.** We initialize M_3 with the original clean weights M_1 and fine-tune on data generated by M_2. We report the baseline results from Table 2 in "()".

Watermark	Model	Setting	M_3 **100%**
StegaStamp†	SD2.1	Supervised	42 (65)
StegaStamp†	SD2.1	Unsupervised	61 (105)
StegaStamp†	Inf-2B	Supervised	35 (34)
StegaStamp†	Inf-2B	Unsupervised	33 (32)
SIREN	SD2.1	Supervised	203 (7.35)
SIREN	SD2.1	Unsupervised	456 (14.74)
BitMark	Inf-2B	Supervised	34 (34)
BitMark	Inf-2B	Unsupervised	33 (32)

C.3 Hyperparameter Ablations

Learning Rate. The learning rate influences how strongly the model adapts to the training data and consequently impacts the strength at which the watermark is learned. In our experiments in Table 16, we fix all hyperparameters and settings and fine-tune M_2 varying only the learning rate. We find that a higher learning rate results in radioactivity being detected with fewer samples up to a setting-specific maximum: beyond this point overall model utility degrades (see Figure 5).

Watermark Strength. The watermark strength controls the magnitude of the embedded signal and in turn the imperceptibility and robustness. In SIREN [22], the watermark encoder $\mathcal{E}_w$ first determines the watermark w based on an image x, with $w = \mathcal{E}_w(x)$. This watermark is then finally embedded onto the image via $x_w = x + \beta \cdot w = x + \beta \cdot \mathcal{E}_w$, where β is the parameter that controls the watermark strength. BitMark [17] embeds the watermark at generation time when the next bit is predicted. It embeds a green bit list, e.g. $\{01, 10\}$, into the generated bit stream by slightly nudging the next

Table 16: **Larger learning rates increase radioactivity, up to a regime-specific threshold.** M_2 is fine-tuned on data generated by M_1, with the watermark fraction shown in brackets (%) and vary only the learning rate. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Learning Rate						
			1×10^{-6}	6×10^{-6}	1×10^{-5}	6×10^{-5}	1×10^{-4}	6×10^{-4}	6×10^{-3}
BitMark (5%)	Supervised	Inf-2B	218	–	98	–	118	74	0.45
	Unsupervised	Inf-2B	527	–	108	–	131	90	1
SIREN (50%)	Supervised	Inf-2B	125	–	93	–	88	49	67
	Unsupervised	Inf-2B	286	–	105	–	85	46	59
SIREN (100%)	Supervised	SD2.1	–	122	70	72	77	34	–
	Unsupervised	SD2.1	–	90	94	59	53	32	–



Figure 5: **An excessively high learning rate degrades model utility and weakens the watermark signal,** making radioactivity harder to detect.

Table 17: **Watermark strength impacts radioactivity.** We fine-tune M_2 at each watermark's default training configuration and vary the watermark's strength parameters. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Watermark Strength (% of default)			
			25%	50%	100% (default)	150%
BitMark (5%)	Supervised	Inf-2B	392.6	782	74	91
	Unsupervised	Inf-2B	0.56	0.36	90	73
SIREN (50%)	Supervised	Inf-2B	286	75	49	205
	Unsupervised	Inf-2B	300	116	46	181
SIREN (100%)	Supervised	SD2.1	47.22	193	70	68
	Unsupervised	SD2.1	15.16	558	94	54

predicted bit to take a specific value based on the previous bit. This is done by adding a constant δ to the logit of the next green bit, e.g. if the previous bit was 1, then the bias will be added to the logit predicting 0. The constant δ influences the strength of the watermark. Per default we use $\beta = 1.5$ and $\delta = 2$. In Table 17 we vary the watermark embedding strength from our default value and find that higher watermark strength can result in stronger radioactivity, as long as the image quality is not impacted negatively. Reducing the strength, on the other hand, can result in the watermark being no longer detectable in the downstream model at all.

Fine-tuning Epochs. We study the effect of the number of fine-tuning epochs on the models radioactivity in Table 18. The results show that the impact of the number of epochs depends on the model architecture.

Table 18: **The optimal number of fine-tuning epochs is setup-specific.** We fine-tune M_2 with each watermark's default training configuration and vary the number of epochs. The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Number Of Epochs					
			1	2	4	5	10	20
BitMark (5%)	Supervised	Inf-2B	89	74	314	–	–	–
	Unsupervised	Inf-2B	103	90	134.88	–	–	–
SIREN (50%)	Supervised	Inf-2B	98	49	187	–	–	–
	Unsupervised	Inf-2B	86	46	106	–	–	–
SIREN (100%)	Supervised	SD2.1	79	75	–	70	79	110
	Unsupervised	SD2.1	79	101	–	94	75	112

Batch Size. We analyze the impact of the training batch size on the watermark radioactivity. Larger batch sizes average the model gradients across more samples potentially diluting the watermark signal during training. However our results in Table 19 suggest that the watermark radioactivity is mostly robust to differing batch sizes. This indicates that the watermark signal is sufficiently detectable across the batch of samples.

Table 19: **Ablation over the effective batch size.** We fine-tune M_2 with each watermark's default training configuration and vary the effective batch size. The reported values follow the same format as Table 1.

Watermark	Setting	Model	Effective Batch Size			
			1	4	16	64
BitMark (5%)	Supervised	Inf-2B	115	74	218	64
	Unsupervised	Inf-2B	110	90	164	69
SIREN (50%)	Supervised	Inf-2B	60	49	96	91
	Unsupervised	Inf-2B	66	46	72	76
SIREN (100%)	Supervised	SD2.1	110	70	96	77
	Unsupervised	SD2.1	87	94	74	93

Bit Capacity. To evaluate the impact of the bitcapacity on the radioactivity, we perform additional experiments using BitMark. By default, BitMark leverages a 2 bit pattern, embedding dense information across the image. We extend this pattern to 3 and 4 bits respectively, resulting in more embedded information, however it becomes less dense due to the increase in the context window size. We report the results in Table 20, following Table 2, fine-tuning M_{i+1} with 5% watermarked data. As also observed by Sander et al. [30], a larger context window drastically reduces the radioactivity of the watermark.

Table 20: **Detection performance for different BitMark payload sizes.** The reported values follow the same format as in Table 1.

Watermark	Setting	Model	Target Model		
			M_2	M_3	M_4
BitMark (2-bit)	Supervised	Inf-2B	74	164	165
	Unsupervised	Inf-2B	90	82	236
BitMark (3-bit)	Supervised	Inf-2B	86	375	454
	Unsupervised	Inf-2B	141	193	786
BitMark (4-bit)	Supervised	Inf-2B	215	10200.0	2670.0
	Unsupervised	Inf-2B	396	630	3600.0

Multiple Keys. In all of our experiments, we actively reuse the same key in all images to amplify the signal of the watermark.To evaluate the impact of using multiple keys, we embed the StegaStamp watermark with Key A on 50% of the training images and with Key B on the other 50% training images, resulting in 100% watermarked data, but less per key information. Our results in Table 21 show that while the watermark remains radioactive and detectable, more samples are required for detection. The per key signal is reduced.

Table 21: **Reusing the same key increases radioactivity.** We embed two distinct keys A and B onto half of the training data each and compute the radioactivity. We report the stopping time following Table 1.

Watermark	Model	Setting	100% Sup	100% Unsup
StegaStamp†	SD2.1	Single Key	46	56
		Dual Key (Key A / Key B)	90 / 97	101 / 95
StegaStamp†	Inf-2B	Single Key	33	34
		Dual Key (Key A / Key B)	40 / 39	44 / 43

Detection Without Key. While *RADIUM* is designed for watermark owners with access to the watermark detector, we additionally evaluate the detection performance without access to the correct watermarking key or detector. To overcome this limitation, we assume an upper bound setting, where we have access to clean and watermarked data from the same distribution. We then train a ResNet-50 to differentiate between the clean and watermarked data, returning a score of 1 for watermarked images and 0 for clean images. Our results in Table 22 show that under these strong assumptions a blind detector can replace the original watermark detection scheme and detect the radioactive traces in the M_2 model. We find that the blind detector partially improves over the watermark owner detection, as the blind detector is trained on a single key, whereas the watermark owner can detect arbitrary keys.

Table 22: Radioactivity detection without access to the secret detector. We train a ResNet-50 on paired clean and watermarked data. We report the results under the *RADIUM* unsupervised setting in "()". The reporting scheme follows Table 1.

Watermark	Model	25%	50%	100%
SIREN	Inf-2B	271 (395)	53 (46)	37 (36)
SIREN	SD2.1	616 (0.02)	544 (0.57)	51 (94)
StegaStamp†	Inf-2B	244 (678)	107 (155)	32 (34)
StegaStamp†	SD2.1	2.55 (0.51)	8788.36 (0.02)	35 (56)
BitMark	Inf-2B	99 (35)	91 (41)	31 (34)

C.4 Larger Models

We additionally explore the radioactivity of image watermark on larger diffusion model architectures, namely Stable Diffusion 3.5-medium with 2B parameters and FLUX.1-dev [21] with 12B parameters. We report our results in Table 23 and Table 24, following the convention of Table 3 and provide the hyperparameters in Table 5. Our results show similar behavior between the two large model architectures, with SIREN and RivaGAN being detectable when making up 25% or more of the training data and StegaStamp being radioactive, but only under higher data fractions. TrustMark is non-radioactive. Comparing the results for the larger DMs and SD2.1 from Table 1, we find that the behavior of the larger models more closely aligns to the behavior of Infinity-2B, suggesting that model size could be a factor influencing the radioactivity of watermarks.

C.5 Watermark Control

In this section we analyze the assumption that the watermarks behave the same across clean models. Therefore we perform a hypothesis test using e-processes. We find that for BitMark [17] this assumption is broken, when the architecture between M_1 and M_2 differ. Therefore we use the model

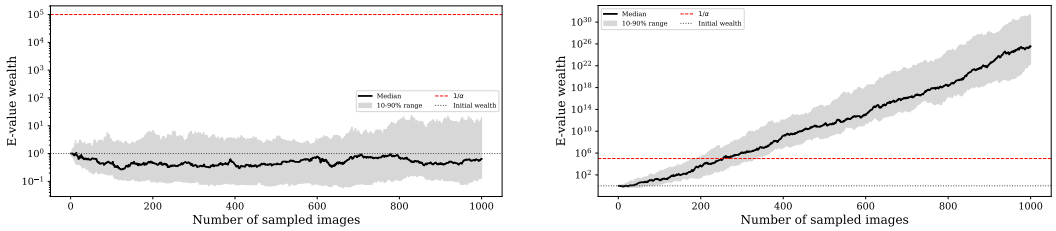
Table 23: **Detection performance for SD3.5.** We train M_2 on different fractions of watermarked data % generated by M_1. The reported values follow the same format as in Table 1.

Watermark	Setting	M_1	M_2	Fraction of Watermarked Data			
				0%	25%	50%	100%
StegaStamp†	Supervised	SD3.5	SD3.5	0.11	0.01	0.12	99
	Unsupervised	SD3.5	SD3.5	0.10	0.03	0.01	169
RivaGAN†	Supervised	SD3.5	SD3.5	0.10	880	492	135
	Unsupervised	SD3.5	SD3.5	0.02	1117.07	799	179
SIREN	Supervised	SD3.5	SD3.5	0.10	249	139	46
	Unsupervised	SD3.5	SD3.5	0.01	222	116	38
TrustMark†	Supervised	SD3.5	SD3.5	0.06	0.04	0.26	0.05
	Unsupervised	SD3.5	SD3.5	0.01	0.02	0.02	0.02
BitMark	Supervised	Inf-2B	SD3.5	0.10	137	48	38
	Unsupervised	Inf-2B	SD3.5	0.30	511	62	35

Table 24: **Detection performance for FLUX.1-dev.** We train M_2 on different fractions of watermarked data % generated by M_1. The reported values follow the same format as in Table 1.

Watermark	Setting	M_1	M_2	Fraction of Watermarked Data			
				0%	25%	50%	100%
StegaStamp†	Supervised	FLUX.1	FLUX.1	0.05	0.39	79	34
	Unsupervised	FLUX.1	FLUX.1	0.03	0.34	103	55
RivaGAN†	Supervised	FLUX.1	FLUX.1	0.01	861	701	56
	Unsupervised	FLUX.1	FLUX.1	2.02	565	368	58
SIREN	Supervised	FLUX.1	FLUX.1	0.06	464	77	39
	Unsupervised	FLUX.1	FLUX.1	0.07	786	96	39
TrustMark†	Supervised	FLUX.1	FLUX.1	0.04	0.03	0.02	0.72
	Unsupervised	FLUX.1	FLUX.1	0.02	0.12	0.14	0.04

M_2 that was trained on only **clean generated data** and compute the watermark scores and compare these against the clean data generated by M_1.



(a) M_1: Infinity-2B; M_2: Infinity-2B, Watermark: BitMark

(b) M_1: Infinity-2B; M_2: SD2.1, Watermark: BitMark

Figure 6: **Bias in BitMark.** We find that BitMark is slightly biased on the tested cross-architecture distribution. While the per-sample test is still insignificant for SD2.1 generated samples, aggregating this bias can lead to false positive detection with our test.

C.6 Visual Quality Evaluation

C.6.1 Model Quality

We evaluate the impact of training on watermarked data on model quality under FID [13] and CLIP score [12]. Therefore we generate 5,000 MS-COCO images, using validation prompts not seen during training and compute the CLIP score, as well as the FID against the matched 5,000 natural MS-COCO images. Our results in Table 25 show, that in general, training on generated data reduces the quality of M_2 by reinforcing existing model biases. However, detectability in M_2 is not solely driven by image quality. Rather image quality and detectability are driven by a confounder, training on generated content. We further evaluate the image quality across multiple model iterations in Table 26. Our results show that especially for SD2.1 image quality degrades drastically with higher model iterations, whereas in Infinity quality degrades more gracefully.

Table 25: FID and CLIP scores for BitMark and StegaStamp for different watermark fractions in M_2. Lower FID and higher CLIP scores are better.

Watermark	Metric	M1 Clean	M1 Watermarked	0%	1%	5%	10%	25%	100%
BitMark (Inf-2B)	FID ↓	40.68	38.10	44.91	45.98	44.53	42.92	43.41	42.17
	CLIP ↑	31.29	31.30	31.37	31.43	31.41	31.38	31.42	31.42
StegaStamp (SD2.1)	FID ↓	25.60	27.04	34.56	35.30	34.74	34.78	35.16	35.01
	CLIP ↑	31.37	31.45	31.33	31.41	31.47	31.48	31.42	31.46

Table 26: FID and CLIP scores for different BitMark and StegaStamp in the iterative training. Lower FID and higher CLIP scores are better.

Watermark	Metric	M1 Clean	M1 Watermarked	M2	M3	M4	M5	M6
StegaStamp (Inf-2B)	FID ↓	40.68	39.98	44.20	45.22	46.06	47.69	52.14
	CLIP ↑	31.29	31.17	31.32	31.46	31.47	31.48	31.39
BitMark (Inf-2B)	FID ↓	40.68	38.10	42.17	43.68	43.79	44.47	46.64
	CLIP ↑	31.29	31.30	31.42	31.57	31.75	31.69	31.58
StegaStamp (SD2.1)	FID ↓	25.60	27.04	35.01	61.40	110.86	172.65	214.96
	CLIP ↑	31.37	31.45	31.46	30.81	29.02	26.34	23.61

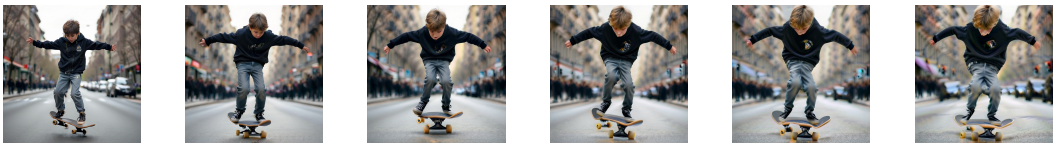
C.6.2 Example Images

We provide examples of the generated images of the models in the iterative training experiment. Therefore we generate an image with all trained models M_i for Infinity-2B trained on BitMark in Figure 8 and for SD2.1 trained on the SIREN watermark in Figure 7. Our results show that with progressing iterations the image quality drops, as the errors and perturbations of the generated images propagate through the model. Since M_{i+1} is the model M_i trained on its own data, errors propagate and can not be fixed, resulting in drastic decay of model quality.



(a) M_1 (b) M_2 (c) M_3

Figure 7: **Impact of the radioactive decay for the SIREN watermark on image quality.** Each successive model is trained on 100% of the data of the preceding model. Training SD2.1 on generated SIREN images results in a destruction of the watermark and degradation of the image quality.



(a) M_1 (b) M_2 (c) M_3 (d) M_4 (e) M_5 (f) M_6

Figure 8: **Impact of multiple training-generation runs on image quality.** We train consecutive models $M_2 \dots M_6$ on 100% of generated data that were watermarked with the radioactive BitMark.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a new method to measure the radioactivity of existing image watermarks.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a discussion of the limitations in Appendix B.4.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: -

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the code to reproduce our results and the paper contains detailed information about the experimental setup.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide full information about the experimental setup and the code in the supplementary material. Additionally, we release the code at this URL.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the information of hyperparameters, how they were chosen and an ablation thereof in the paper.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our results are based on statistical tests that provide information about the detection significance.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We provide information about the compute environment, however given the amount and variety of experiments estimating the full compute and time for each experiments exactly is not possible.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We fully follow the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the impact of our work in the conclusion.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release new models or new datasets.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the owners of all existing assets that we use and we respect their licenses.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide code for reproduction of our experiments and the usage of our framework.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: -

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: -

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification:-