Finding needles in a haystack: A Black-Box Approach to Invisible Watermark Detection

Minzhou Pan[1,2], Zhenting Wang[2,3], Xin Dong[2]
Vikash Sehwag[2], Lingjuan Lyu[2], and Xue Lin[1]

[1]Northeastern University [2]Sony AI [3]Rutgers University

Abstract. In this paper, we propose WaterMark Detector (WMD), the first invisible watermark detection method under a black-box and annotation-free setting. WMD is capable of detecting arbitrary watermarks within a given detection dataset using a clean non-watermarked dataset as a reference, without relying on specific decoding methods or prior knowledge of the watermarking techniques. We develop WMD using foundations of offset learning, where a clean non-watermarked dataset enables us to isolate the influence of only watermarked samples in the reference dataset. Our comprehensive evaluations demonstrate the effectiveness of WMD, significantly outperforming naive detection methods, which only yield AUC scores around 0.5. In contrast, WMD consistently achieves impressive detection AUC scores, surpassing 0.9 in most single-watermark datasets and exceeding 0.7 in more challenging multi-watermark scenarios across diverse datasets and watermarking methods. As invisible watermarks become increasingly prevalent, while specific decoding techniques remain undisclosed, our approach provides a versatile solution and establishes a path toward increasing accountability, transparency, and trust in our digital visual content.

1 Introduction

For a long time, invisible digital image watermarks have served as a reliable solution for tracing plagiarism and unauthorized copying, safeguarding intellectual property rights without compromising image quality [14, 42, 53, 55]. Moreover, with the advent of generative models, such watermarks have been proposed as a means of identifying and sourcing AI-generated images [20, 49]. However, the blind detection of invisible watermarks in a given image dataset, without access to the corresponding decoding algorithms, poses significant challenges.

The inherent invisibility of watermarks makes manual screening of datasets an impractical task. Moreover, the wide variety of watermarking methods [11, 20, 32, 45, 49, 56], each employing different embedding techniques, complicates the development of a generalized Deep Neural Network (DNN) detector. Some watermarking methods are even black-box [33], lacking APIs for third-party users, further hindering the inclusion of all methods in the training process. Related techniques, such as Out-of-Distribution (OOD) detection and anomaly detection,

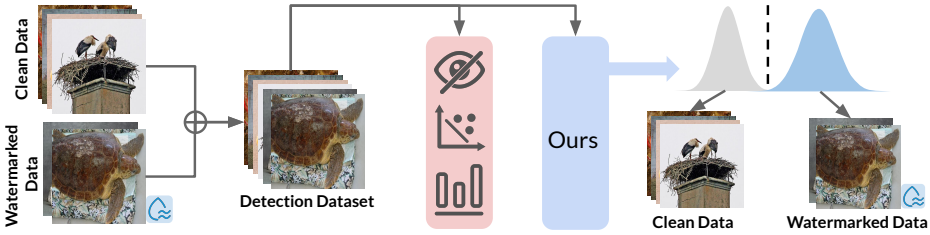

Fig. 1: Detecting invisible watermarked in a given dataset. Due to the invisibility of watermarks, human inspection and existing anomaly detection methods fail to distinguish watermarked images from clean ones within a dataset. To address this challenge, we propose WMD as the first invisible watermark detection capable of accurately identifying invisible watermarked samples in the black-box setting, where there is no need for prior knowledge of the watermarking techniques or decoding methods.

also struggle to effectively identify watermarks due to the subtle perturbations they introduce, as discussed in §2.3.

Failing to detect watermarks in a dataset can lead to severe consequences. Watermarked images may contain sensitive or copyrighted information, and generative models trained on such data may inadvertently memorize these images illegally [15]. Furthermore, recent studies have shown that the inclusion of watermarked images, particularly those generated by AI, in training datasets can degrade the performance of downstream models [9]. As policies increasingly mandate the use of watermarks in generative models [2, 3, 6], and with the rapid proliferation of these models [4, 7], watermarked images are expected to become more prevalent in the near future.

In response to these emerging threats and the growing importance of watermarks in the AI-generated content landscape, we introduce the first black-box invisible watermark detection method: **W**ater**M**ark **D**etection (WMD), a method for reliably detecting arbitrary watermarks in datasets, as depicted in Figure 1. Instead of relying on specific decoding methods for each watermark, WMD stands out as a versatile black-box approach that eliminates the need for prior knowledge of watermarking or decoding methods. By leveraging the similar distribution of clean datasets, WMD employs self-supervised learning to effectively identify watermarks. Extensive evaluation shows the effectiveness of WMD, consistently achieving detection AUC above 0.9 in most single-watermark datasets and above 0.7 in more challenging multi-watermark scenarios.

In this paper, we introduce WMD, the first black-box invisible watermark detection method capable of reliably identifying arbitrary watermarks in datasets without prior knowledge of watermarking or decoding techniques, leveraging self-supervised learning to exploit the similar distribution of clean datasets.

2 Background and Related Work

2.1 Importance of Detecting Watermarks

Invisible digital watermarks were initially designed to protect intellectual property and copyrights without compromising the visual quality of images. Such

watermarks have been widely implemented across various domains, becoming a popular solution for content sourcing [14, 42, 42, 55]. Recently, watermarks have become increasingly important with the development of generative models, particularly diffusion models, which can produce photo-realistic images that are challenging for humans to distinguish from real photos [22, 44]. The ability to detect these watermarks is crucial for several reasons:

Protecting Intellectual Property: Watermarks have long been used to trace intellectual property infringement. Detecting these watermarks is essential for ensuring that copyrighted content is not collected in model training datasets and for keeping downstream models from learning this information (models can remember these details, as revealed by certain attacks [15]).

Preventing Misuse: Generative models can be used to create fake news, propaganda, and other malicious content [17, 29]. As multiple legislation and executive orders [2, 3, 6] have been proposed to require watermarks on AI-generated images, detecting watermarks can help identify AI-generated images and prevent their misuse.

Maintaining Dataset Quality: According to recent surveys [4, 7], AI-generated images account for a significant and growing proportion of all images produced. Using these images in training datasets can introduce biases and inaccuracies that negatively impact the performance of downstream models [9]. Detecting and filtering out watermarked images is crucial for maintaining dataset quality and ensuring accurate model development.

The increasing prevalence of AI-generated content has led to a substantial rise in the number of watermarked images in circulation. Recent surveys [4, 7] indicate that AI-generated images account for a considerable portion of all images produced, with over 18 billion AI-generated images created within a year, and this number is growing rapidly. In comparison, human-generated images in the same period amount to around 355 billion [46]. Based on these statistics, we can estimate that approximately 5% of all images created from now on will be AI-generated and potentially watermarked. This proportion will be used as a basis for subsequent evaluation in §4. This trend underscores the critical importance of developing robust watermark detection methods to protect intellectual property, prevent misuse, ensure regulatory compliance, and maintain dataset quality.

2.2 Invisible Image Watermarks

Invisible image watermarking embeds non-visible markers into digital images to protect copyrights and identify sources. The primary objective of such watermarking is to ensure that these markers can be readily detected by a pre-designed method while remaining imperceptible to other detection attempts and during normal use. Furthermore, the watermark should be robust and resilient to image modifications and regenerations, enabling its creator to detect it even after the image has been altered or recreated.

Various watermarking methods have been proposed to embed watermarks into images. These methods can be divided into two categories: Post-processing Watermarks and Generative Watermarks.

Post-processing Watermarks. This watermarking category involves the integration of watermarks directly into the image content. Traditional methods include embedding secret information into the least significant bit (LSB) [11], or incorporating watermarks within the frequency domain via transforms such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) [12,34]. Other techniques exploit image decompositions, such as Singular Value Decomposition (SVD) [16], or employ composite transforms [32].

With the advent of deep learning, new approaches to learn watermarking have emerged. Many of these methods adopt an encoder-decoder architecture, where the encoder embeds the watermark, and the decoder is responsible for its extraction [45, 56]. The training of the encoder is enhanced by introducing simulated differentiable image distortions, which are placed between the encoder and decoder. This process endows the encoder with resilience against real-world image perturbations, significantly improving the robustness of the watermark. Moreover, the encoder's training regimen includes loss functions designed to minimize watermark visibility, thus advancing the limits of robustness and imperceptibility beyond what is achievable with conventional hand-designed transforms.

Generative Watermarks. Besides embedding watermarks into existing images, watermarking can be incorporated directly into the image generation process of generative models. A notable example is Stable Signature [20], which involves the training of an encoder-decoder framework. In this method, the latent decoder of the stable diffusion model is fine-tuned to act as a watermark encoder, embedding an imperceptible watermark into the generated images for subsequent detection and identification.

Another innovative approach, Tree-Ring watermarks [49], adapts traditional frequency-domain watermarking techniques. It leverages the Fast Fourier Transform (FFT) to transpose the diffusion latent space into a space amenable to watermarking. Therein, a unique watermark is embedded. For detection, the watermarked images undergo an inverse process through the same diffusion network and FFT, allowing for the watermark region to be cross-referenced with the original watermark for verification.

2.3 Watermark Detection

As discussed in §2.1, invisible watermarks play a crucial role in protecting intellectual property rights, preventing the misuse of AI-generated content, and maintaining the quality of image datasets. However, despite the long history and diversity of invisible watermarking methods, detection techniques have not yet emerged. This limitation can be attributed to several factors:

Infeasibility of Human Annotation. Due to the invisibility of watermarks, it is practically impossible for humans to identify and annotate watermarked images within large-scale datasets. Consequently, the approach of manually labeling a small subset of watermarked data and training a model to detect the remaining watermarks, as employed in visible watermark detection [18, 40], is not applicable to invisible watermarks.

Challenges in Collecting Comprehensive Watermarking Methods. As mentioned in §2.2, there exists a wide variety of watermarking methods, each with its specific embedding and decoding techniques. Even if it is possible to collect and generate watermarked datasets using a subset of these methods, it would be extremely difficult to encompass all known watermarking techniques. Moreover, many watermarking methods are proprietary and black-box [33], making it infeasible for ordinary users to obtain access and generate corresponding datasets.

Limitations of Self-Supervised Approaches. Since the label information is unavailable, several self-supervised approaches like Out-of-Distribution (OOD) detection, anomaly detection, and backdoor detection methods [35,36] have been proposed as substitutes to identify "abnormal" examples within a given dataset. However, these approaches fail to detect watermarked examples effectively because the perturbations introduced by watermarks are relatively small compared to typical anomalous features. Furthermore, watermarks do not cause obvious changes in model behavior, unlike backdoor examples.

To validate these findings, we conduct an experiment using the well-known DctDwtSvd [32] invisible watermarking method to embed watermarks in 5% of the samples (as stated in §2.1) within a 10,000-sample subset of ImageNet [19]. The results, summarized in Table 1, demonstrate the ineffectiveness of various existing detection methods in identifying even the most basic invisible watermarks. This highlights the need for novel and robust watermark detection techniques that can overcome the limitations of current approaches.

Table 1: Performance comparison of various detection methods on an ImageNet subset containing DctDwtSvd watermarks. The low AUC scores indicate the inability of these methods to effectively detect even the most basic invisible watermarks.

	Visible Watermark Detection		**Anomaly/OOD Detection**		**Backdoor Samples Detection**	
	TV-L1 [40]	LSW [18]	DROC [43]	RIAD [51]	CT [36]	ASSET [35]
AUC (↑)	0.508	0.512	0.513	0.522	0.514	0.518

Considering the growing importance of invisible watermarks and the challenges associated with their detection, an effective and reliable invisible watermark detection method is crucial. To address this need, we propose WMD (Watermark Detection), a novel self-supervised learning approach capable of successfully detecting watermarks in the given dataset with high probability without prior knowledge of the specific watermarking algorithm.

3 Method

3.1 Problem Setup

In this section, we present the threat model for our proposed invisible watermark detection method. Let $\boldsymbol{x}_i^d \in \mathcal{D}_d$ be the dataset of images awaiting watermark detection, where some images $\boldsymbol{x}_i^d$ may be watermarked by any existing invisible watermarking technique. The watermarked portion of the dataset is denoted as

$\boldsymbol{x}_i^w \in \mathcal{D}_d^w$, and the clean images are denoted as $\boldsymbol{x}_i^c \in \mathcal{D}_d^c$. The detection dataset can also be expressed as $\mathcal{D}_d = \mathcal{D}_d^w \cup \mathcal{D}_d^c$.

Objective. The objective of our watermark detector is to find a watermark detection method $f(.)$ that reliably identifies the watermarked images within the given dataset, such that $f(\mathcal{D}_d) \rightarrow \mathcal{D}_d^w$. In particular, the watermark detector aims to classify each image in $\mathcal{D}_d$ as watermarked or non-watermarked.

We assume that the watermark detector has no prior knowledge of the watermarking process including which images are watermarked and the type of watermark used, thus referring to the process as *black-box watermark detection*. However, the detector has access to the visual distribution of images in the detection dataset $\mathcal{D}_w$ and can obtain a clean dataset $\boldsymbol{x}_i^c \in \mathcal{D}_c$ that has similar visual distribution.

The watermark detector has full access to both the clean dataset $\mathcal{D}_c$ and the detection dataset $\mathcal{D}_d$ and is allowed to use these datasets to develop the detector. We further discuss the impact of our design choices in the development of the detection in §3.2 and §3.3.

3.2 Oracle Watermark Detection

We formulate the watermark detection problem as an offset optimization problem [35]. Offset optimization is a technique that identifies differences between two datasets by effectively canceling out the common elements. Consider the oracle detection case where the clean images in the detection dataset $\mathcal{D}_d$ and the clean dataset $\mathcal{D}_c$ have identical distributions, and the size of the two datasets is same and the number of watermarks is non-zero, $N = |\mathcal{D}_d| = |\mathcal{D}_c| > |\mathcal{D}_d^c| >> |\mathcal{D}_d^w|$. With this knowledge, we can initialize a deep neural network (DNN) model, $f(\cdot;\theta)$, and calculate the gradients of the loss function $\mathcal{L}$ with respect to the model parameters θ for each dataset:

$$\Delta\theta_c = \nabla_\theta \frac{1}{N} \sum_{\boldsymbol{x}_i^c \in \mathcal{D}_c} \mathcal{L}(f(x_i^c;\theta)) \tag{1}$$

$$\Delta\theta_d = \nabla_\theta \left(\frac{1}{N} \sum_{\boldsymbol{x}_i^c \in \mathcal{D}_d^c} \mathcal{L}(f(x_i^c;\theta)) + \frac{1}{N} \sum_{\boldsymbol{x}_i^w \in \mathcal{D}_d^w} \mathcal{L}(f(x_i^w;\theta)) \right) \tag{2}$$

We then get the total gradient by subtracting the gradients from the two datasets: $\Delta\theta = \Delta\theta_c - \Delta\theta_d$.

$$\Delta\theta = \nabla_\theta \left(\frac{1}{N} \sum_{\boldsymbol{x}_i^c \in \mathcal{D}_c} \mathcal{L}(f(x_i^c;\theta)) - \frac{1}{N} \sum_{\boldsymbol{x}_i^c \in \mathcal{D}_d^c} \mathcal{L}(f(x_i^c;\theta)) - \frac{1}{N} \sum_{\boldsymbol{x}_i^w \in \mathcal{D}_d^w} \mathcal{L}(f(x_i^w;\theta)) \right) \tag{3}$$

Under the Oracle assumption, the clean samples from both datasets have identical distributions and there are more numbers in the clean data set, so the gradient of $\mathcal{D}_d^c$ will be cancelled:

$$\Delta\theta = \nabla_\theta \frac{1}{N} \sum_{\boldsymbol{x}_i^c \in (\mathcal{D}_c - \mathcal{D}_d^c)} \mathcal{L}(f(x_i^c;\theta)) - \nabla_\theta \frac{1}{N} \sum_{\boldsymbol{x}_i^w \in \mathcal{D}_d^w} \mathcal{L}(f(x_i^w;\theta)) \tag{4}$$

Optimizing the model parameter by descent this gradient, the problem then becomes:

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{\boldsymbol{x}_i^c \in (\mathcal{D}_c - \mathcal{D}_d^c)} \mathcal{L}(f(\boldsymbol{x}_i^c; \theta)) + \arg\max_{\theta} \frac{1}{N} \sum_{\boldsymbol{x}_i^w \in \mathcal{D}_d^w} \mathcal{L}(f(\boldsymbol{x}_i^w; \theta)). \quad (5)$$

The optimized model $f(\cdot; \theta^*)$ will generate high output values for watermarked samples and low values for clean samples, enabling watermark detection based on the difference between model outputs.

This analysis reveals that the performance of the oracle detector heavily depends on the *similarity between the clean distributions of the two datasets and the number of watermarked samples in the detection dataset*. In practice, finding a clean dataset that perfectly offsets the gradient is challenging. To address this issue, we propose our method WMD, which relaxes the problem and enables practical watermark detection in real-world scenarios.

3.3 WMD: Our Proposed Black-box Watermark Detector

To address watermark detection in real-world scenarios, where finding a perfect clean dataset corresponding to the detection dataset is challenging. To this end, we propose WMD. The key components of WMD are an asymmetric loss function and the iterative pruning strategy for the detection dataset. The following subsections provide a detailed design and analysis of each component of WMD:

Asymmetric Loss. Considering the high similarity between watermark detection and binary classification tasks, a straightforward design approach would be to use a symmetric loss. Symmetric loss employs the same loss function for both minimization and maximization objectives, that is minimizing the model output of samples in the clean dataset and maximization the model output of samples in the detection dataset. However, our ablation study in Appendix §4.3 reveals that symmetric loss functions fail to generate satisfactory results. The underlying reason for this is that symmetric loss functions have the same loss scale for both minimization and maximization goals, leading to an automatic balance between the two objectives. If the maximize loss is smaller, the optimization will shift its focus to minimization, and vice versa.

For the clean samples in the detection dataset, $\mathcal{D}_d^c$, its gradients are always offset by the clean samples in the clean dataset. Consequently, it is more challenging to maximize the output for $\mathcal{D}_d^c$ compared to the watermarked samples, $\mathcal{D}_d^w$, resulting in $\mathcal{D}_d^c$ consistently generating higher loss than $\mathcal{D}_d^w$. Furthermore, watermarked samples only make up a small portion of the detection dataset, meaning their total loss is already very small compared to the clean samples. The combination of these two factors causes the model to focus on optimizing the clean samples, $\mathcal{D}_d^c$, while neglecting the maximization of watermarked samples.

The evaluation results in Appendix §4.3, Table 4 provides further support for this explanation. The diagonal of the table presents the results of using symmetric loss functions, including symmetric exponential loss, symmetric softmax loss,

and symmetric BCE loss. All of these symmetric loss functions yield worse results compared to a symmetric linear loss. The reason for this is that these three loss functions will amplify the differences between losses, causing the model to focus more on the hard samples by emphasizing the high-loss samples (clean images) and downplaying the low-loss samples (watermarked images), thereby diverting attention away from the actual target. This observation confirms our previous conjecture about the limitations of symmetric loss functions in this context.

Following the analysis, we've identified the distinct characteristics of watermark detection compared to standard binary classification. As a result, we propose the asymmetric loss function to improve the detection performance.

For the clean dataset, considering the fact that all images are from trusted sources and therefore clean, we can encourage the model to pay more attention to hard examples, i.e., the samples that yield higher loss values. A common solution is to use an exponential loss function:

$$\mathcal{L}_{exp} = \exp(f(x_i^c; \theta)/\tau) \tag{6}$$

Here, τ is a temperature scaler that assigns higher loss values to samples with higher model outputs. This focuses the model on minimizing the loss for these hard examples, ensuring that all samples in the clean dataset are strictly minimized.

For the detection dataset, which contains both clean and watermarked samples, we want to ensure that the watermarked samples are always maximized. In other words, the model should give nearly equal focus to all examples, regardless of their output. To achieve this, we can use a linear loss:

$$\mathcal{L}_{lin} = -f(x_i^d; \theta) \tag{7}$$

Using a linear loss has two benefits. First, it ensures that the model gives almost the same weight to all examples, keeping the focus on maximizing the watermarked examples. Second, it generates lower gradients compared to the exponential loss used for clean examples, ensuring that the losses of the clean examples are strictly bound by minimization.

However, simply combining the exponential loss with the linear loss results in different scales, making it difficult to find a balancing factor between the two. To address this, we modify the exponential loss into a softmax loss:

$$\mathcal{L}_{sm} = \log(\exp(f(x_i^c; \theta)/\tau)) \cdot \tau \tag{8}$$

The overall loss then becomes:

$$\mathcal{L}_{total} = \log(\exp(f(x_i^c; \theta)/\tau)) \cdot \tau - f(x_i^d; \theta) \tag{9}$$

By using this asymmetric loss design, we can effectively optimize the model to detect watermarks while minimizing the impact of clean samples on the detection performance.

Iteration Pruning. In the detection dataset, the number of clean images often significantly exceeds the number of watermarked images, making the separation of model outputs challenging. As the detection model $f(\cdot; \theta)$ is an over-parameterized deep neural network (DNN) [10], it may eventually "memorize"

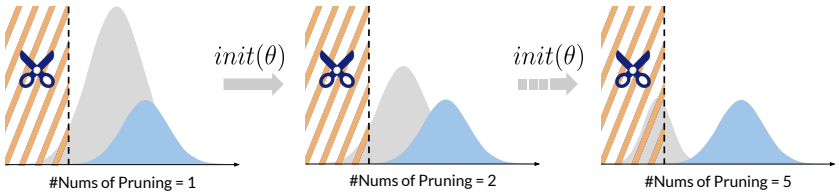

Fig. 2: Illustration of the Iterative Pruning process. As the number of pruning iterations increases, the detection dataset is gradually condensed by removing **clean samples** while retaining most of the **watermarked samples**.

all samples, leading to high loss values for both clean and watermarked images in the detection dataset and causing detection failure. Conversely, insufficient training results in poor separation between the clean and watermarked parts.

However, we observe that even when the model cannot perfectly separate these samples, the output values of the watermarked samples are consistently higher than those of some clean samples. This may be attributed to the fact that the watermark, even if slight, still has a different distribution compared to clean images. Leveraging this observation, we propose a strategy called "Iteration Pruning" to improve the detectability of watermarks.

The Iteration Pruning strategy is designed to efficiently refine the detection dataset and accelerate the learning process of the watermark detection model. This strategy involves two key hyperparameters: the pruning rate ρ and the pruning interval *interval*. The training process begins with the initial dataset, and at every *interval* training epochs, a percentage of data equal to ρ is removed from the detection dataset. This pruning targets the samples with the lowest loss values, effectively discarding the least informative or most easily learned samples. To prevent overfitting, the model is reinitialized after each pruning step. The pruning process continues iteratively throughout the training progress until only 5% of the total data remains. This iterative pruning approach effectively "condenses" the dataset by retaining the majority of the watermarked samples while progressively eliminating more clean samples. As the "condensing" process progresses, the model gradually focuses its attention on the watermarked samples, enabling it to quickly learn the optimal parameters for perfect separation between watermarked and clean data.

The effectiveness of Iteration Pruning and the impact of pruning hyperparameters are investigated through an ablation study in Appendix §4.3, providing insights into the optimal configuration for enhancing watermark detection performance.

3.4 Overall Method

The overall workflow of WMD is as follows: First, the model is initialized with the detection dataset $\mathcal{D}_d$ and a clean dataset $\mathcal{D}_c$. During each training epoch, mini-batches from both datasets are fed into the model, and the asymmetric loss function is calculated. The model parameters are then updated using the gradients computed from the total loss. After a fixed number of epochs (determined by the *interval* parameter), the iterative pruning process is triggered. The model's

Algorithm 1: Algorithm for training WMD

Input: θ (Model); E (Epochs); $\mathcal{D}_d$ (Detection Data); $\mathcal{D}_c$ (Clean Data)
Output: $\mathcal{D}_w$ (Watermarked Dataset)
Parameters: η (Learning Rate), ρ (Pruning Rate), *interval* (Pruning Interval)

$\mathcal{D}_{dt} \leftarrow \mathcal{D}_d$
for $e = 1$ **to** E **do**
  for *each* N*-sized batch* $\mathcal{B}_c, \mathcal{B}_{dt}$ *from* $\mathcal{D}_c, \mathcal{D}_{dt}$ **do**
    $p_c, p_{dt} \leftarrow f(\mathcal{B}_c; \theta), f(\mathcal{B}_{dt}; \theta)$ `// Get model outputs`
    $\mathcal{L}_{total} \leftarrow \mathcal{L}_{sm}(p_c) + \mathcal{L}_{lin}(p_{dt})$ `// Calculate the losses`
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{total}$ `// Update the model with gradients`
  if $e \mod interval = 0$ **then**
    $(p_d)\text{Ranked} \leftarrow rank(f(\mathcal{D}_d; \theta))$ `// Get model outputs and rank`
    `// Update` $\mathcal{D}_{dt}$ `by pruning samples with lowest outputs`
    $\mathcal{D}_{dt} \leftarrow (p_d)\text{Ranked}[: \lfloor |\mathcal{D}_d| \times (1 - (1 - \rho)^{\frac{e}{interval}}) \rfloor]$
    $\theta \leftarrow init(\theta)$ `// Reinitialize model to avoid overfitting`
return $\mathcal{D}_{dt}$

performance on the detection dataset is evaluated, and the samples are ranked based on their confidence scores. The bottom $(1-(1-\rho)^{\frac{e}{interval}})$ fraction of the ranked samples, which are most confidently predicted as non-watermarked, are removed from the detection dataset. The model is then reinitialized, and the training process continues with the updated detection dataset. This iterative pruning strategy is repeated until the specified number of epochs is reached. Finally, the detection dataset $\mathcal{D}_w$, containing the identified watermarked images, is returned.

4 Evaluation

In this section, we provide a comprehensive evaluation of our proposed method, WMD, for detecting invisible watermarks in a given dataset.

4.1 Setting

Evaluation metrics. As our method focuses on detecting the presence of watermarks and outputs a binary result, either "watermarked" or "non-watermarked", we employ three widely used binary detection metrics: *Area Under the Curve* (AUC): Assesses the watermark system's discernment between watermarked and non-watermarked images, with higher AUC reflecting greater detected performance. *True Positive Rate at 10% False Positive Rate* (TPR @ 10% FPR): Demonstrates watermark detection capabilities at the expense of a small number of clean samples. *False Positive Rate at 10% True Positive Rate* (FPR @ 90% TPR): Demonstrates the false positive rate for clean images when detecting most watermarked images.

Baseline Watermark. We selected representative works among different watermarking methods as baseline watermarking methods to evaluate our watermark detection performance. For post-processing watermarking, we choose the least significant bit (LSB) [11], which embeds the watermark into the image's lowest bit, and DctDwtSVD (DDS) [32], integrating the watermark into the DCT space and SVD vectors. Within the deep learning paradigm, HiDDeN [56] utilizes an encoder-decoder architecture with a noise simulation layer for embedding, whereas StegaStamp (SS) [45] enhances robustness by incorporating higher noise levels and improving model structure. For generative watermarking, Stable-Signature (SSig) [20] embeds watermarks into latent diffusion models by fine-tuning the last layer, and Tree-Ring watermark (TR) [49] embeds the watermark into the diffusion model latent frequency space. All the watermarks embedded 64 bits of information into the image, for the Tree-Ring watermark, we use the Tree-Ring$_{\mathbf{Rings}}$ variant.

Datasets & Models. In the main evaluation, we use three datasets as our evaluation datasets for post-processing watermarks: ImageNet [19], COCO [28], and Caltech-256 [21]. We randomly select 20,000 images from each dataset and split them into two subsets: 10,000 samples detection dataset and 10,000 samples clean dataset. All images are resized to a consistent 256x256 resolution. However, for diffusion model-specific watermarks, we utilize the image prompt datasets DiffusionDB [48], MidJourney Prompt dataset [5] and image captions from COCO dataset [28]. Similarly, we randomly choose 20,000 text prompts and divide them into two splits. We employ Stable Diffusion V1.4 [37] as the image generation model, maintaining the generated image size at 256x256.

Regarding the detection model, WMD is designed as a watermark detection method, allowing any DNN model to be plugged in as the detection model. For efficiency, we construct a simple network consisting of only 5 ConvNext-V2 blocks [50] with only 1.93M parameters.

4.2 Watermark Detection

In this section, we will test WMD detection performance across multiple dataset and watermark method, our evaluation will have two parts, single watermark and multiple watermarks.

Single Watermark. In this set of experiments, we consider the scenario where only one type of watermark method is applied to the detection dataset. As discussed in Section 2.1, we randomly watermarked 5% of the images in the detection dataset using a single watermark method.

The upper part of Table 2 presents the results for post-processing watermarks. WMD achieves remarkable detection performance across all methods and datasets, with AUC scores consistently above 0.8. However, the performance on the LSB watermark is relatively lower compared to other methods. This can be attributed to the fact that LSB watermarks introduce minimal modifications to the image, as evidenced by their lower PSNR values (see Appendix E). The lower part of Table 2 shows the results for generative watermarks. WMD maintains AUC scores above 0.9 for the Stable Signature (SSig) watermark across

all datasets. However, the detection rates for the Tree-Ring (TR) watermark are comparatively lower. This may be due to the fact that the changes brought by TR are smaller than other watermarking methods since it embeds the watermark into the diffusion latent space.

Multiple Watermarks. In this set of experiments, we evaluate the performance of WMD when multiple watermark methods are simultaneously present in the dataset. This scenario more closely resembles real-world conditions where different watermarking techniques may appear in the same dataset. For post-processing watermarks (LSB, DDS, HiDDeN, and SS), we randomly apply each method to 1.25% of the images in the detection dataset, resulting in a total of 5% watermarked images. For generative model watermarks (SSig and TR), each method is applied to 2.5% of the images, again resulting in a total of 5% watermarked images.

The upper part of Table 3 presents the results for post-processing watermarks in the multi-watermark setting. Compared to the single watermark scenario, the detection performance of WMD decreases slightly across all metrics and datasets. This is expected, as the presence of multiple watermark types introduces additional variability and complexity. The lower part of Table 3 shows the results for generative watermarks, where each watermark method (SSig and TR) is applied to 2.5% of the images. The AUC scores for both SSig and TR watermarks are lower compared to the single watermark setting but remain above 0.7. However, the greater decrease in performance for the TR watermark reflects the fact that stealthier watermarks will be further weakened in the presence of multiple watermarks.

4.3 Ablation Study

Loss Design. The ablation study in Table 4 demonstrates the importance of selecting appropriate loss functions for the "Maximize" and "Minimize" objectives in WMD. Using symmetric loss leads to a performance drop due to the presence of clean samples in the maximizing divert the offset goal. As our analysis in §3.3, asymmetric loss design can greatly improve model detection capabilities. The

Table 2: Watermark detection performance of WMD across different datasets and watermark methods. Methods marked with * represent generative watermarks, which are directly embedded during the image generation process. Higher AUC and TPR @ 0.1 FPR indicate better performance, while lower FPR @ 0.9 TPR is desirable.

	ImageNet			COCO			Caltech		
	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR
LSB	0.852	0.742	0.145	0.837	0.710	0.156	0.845	0.693	0.102
DDS	0.968	0.938	0.020	0.961	0.927	0.070	0.955	0.941	0.080
HiDDeN	0.952	0.892	0.115	0.954	0.918	0.042	0.944	0.870	0.145
SS	0.912	0.819	0.082	0.936	0.921	0.102	0.889	0.893	0.094
	COCO			DiffusionDB			MidJourney		
SSig*	0.939	0.879	0.118	0.941	0.903	0.045	0.932	0.858	0.147
TR*	0.821	0.749	0.142	0.823	0.782	0.189	0.811	0.704	0.108

Table 3: Watermark detection performance of WMD across different datasets and watermark methods in a multi-watermark setting. For post-processing watermarks, each method is applied to 1.25% of the dataset, resulting in a total of 5% watermarked images. Methods marked with * represent generative watermarks, where each method is applied to 2.5% of the dataset during the image generation process, also resulting in a total of 5% watermarked images.

	ImageNet			COCO			Caltech		
	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR	AUC (↑)	TPR (↑) @ 0.1 FPR	FPR (↓) @ 0.9 TPR
LSB	0.732	0.642	0.245	0.715	0.595	0.256	0.725	0.573	0.202
DDS	0.898	0.828	0.120	0.911	0.847	0.170	0.905	0.831	0.180
HiDDeN	0.842	0.782	0.155	0.844	0.778	0.142	0.834	0.737	0.145
SS	0.802	0.709	0.182	0.826	0.811	0.202	0.779	0.783	0.194
	COCO			DiffusionDB			MidJourney		
SSig*	0.829	0.769	0.218	0.831	0.793	0.145	0.822	0.748	0.247
TR*	0.711	0.639	0.242	0.713	0.672	0.289	0.701	0.594	0.208

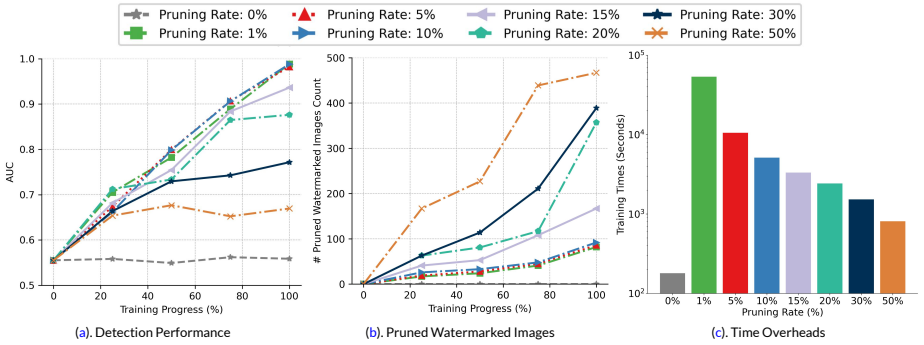

Fig. 3: Impact of pruning rate on watermark detection and training overheads. **(a)** Detection performance measured by AUC decreases as the pruning rate increases, with higher pruning removing more watermarked images during training. **(b)**Number of pruned watermarked images increases with higher pruning rates throughout the training process. **(c)** Time overheads for training increase substantially with higher pruning rates.

combination of linear loss and softmax loss achieves the best performance (AUC 0.968) by providing a balanced and complementary optimization approach.

Pruning Rate. We investigate the impact of the pruning rate on the performance and efficiency of WMD and present the results in Figure 3. As shown in **(a)**, excessively high pruning rates result in reduced detection performance because the model has not successfully separated watermarked images from clean images at this stage. Pruning too many images in this scenario will cause a large

Table 4: Ablation study on the impact of loss function choices for the "Minimize" (clean dataset) and "Maximize" (detection dataset) objectives on watermark detection performance (AUC).

		Minimize			
		BCE	Linear	Exp	Softmax
Maximize	BCE	0.757	0.612	0.831	0.877
	Linear	0.891	0.786	0.842	**0.968**
	Exp	0.512	0.516	0.716	0.513
	Softmin	0.583	0.508	0.533	0.752

number of watermarked images to be removed, as shown in **(b)**, thus affecting the model's ability to learn from the remaining watermarked images in the next round of learning. However, while lower pruning rates maintain high detection performance, they require more iterations and incur significant time overhead. As illustrated in Figure 3(c), a 1% pruning rate incurs approximately 66.2 times more overhead compared to a 50% pruning rate.

5 Discussion

Limitations and Future Work. While WMD demonstrates strong detectability, our evaluations do reveal some limitations. One key challenge is that the detection performance relies on the clean dataset and the detection dataset having similar distributions, which may be difficult to achieve in practice. Additionally, the current method relies on hyperparameter tuning to work effectively. Future research should focus on addressing these limitations by exploring adaptive techniques such as domain adaptation to handle distribution mismatches or enhance detectability through optimizable hyperparameter selection.
Wider Applications. In addition to detecting invisible watermarks, WMD demonstrates versatility in supporting a range of other applications. As detailed in Appendix §B, WMD can be utilized to facilitate watermark removal attacks (Appendix §B.1) and to filter out harmful examples from datasets (Appendix §B.2). These use cases showcase the potential of WMD to make a significant impact across various domains, extending beyond its core functionality.
Broaden Impacts. By enabling reliable invisible watermark detection, WMD allows users to make informed decisions about image usage, deters unauthorized watermarking, and promotes responsible practices. Its implications extend to industries like digital forensics, assisting in identifying image tampering and unauthorized distribution. As invisible watermark usage evolves, WMD's impact is poised to grow, fostering transparency, accountability, and trust in digital visual content handling, ultimately contributing to a more secure digital image ecosystem.

6 Conclusion

In this paper, we proposed WMD, a Black-box invisible watermark detection method that achieves robust performance across diverse watermarking techniques without prior knowledge of the specific method used to apply the watermarks. Extensive evaluations demonstrated WMD's effectiveness, with AUC scores consistently above 0.9 in most single-watermark settings and above 0.7 in challenging multi-watermark scenarios. WMD's potential extends beyond detection to supporting watermark removal attacks and filtering harmful examples. As invisible watermarks become increasingly prevalent, especially in AI-generated imagery, WMD capability to identify watermarked samples lays the foundation for promoting transparency, accountability, and trust in digital visual content.

References

1. Stable diffusion image variations, https://huggingface.co/lambdalabs/sd-image-variations-diffusers 22
2. (Dec 2023), https://www.europarl.europa.eu/thinktank/de/document/EPRS_BRI(2023)757583 2, 3
3. (Oct 2023), https://www.whitehouse.gov/briefing-room/statements-releases/2023/10/30/fact-sheet-president-biden-issues-executive-order-on-safe-secure-and-trustworthy-artificial-intelligence/ 2, 3
4. (Aug 2023), https://journal.everypixel.com/ai-image-statistics 2, 3
5. Midjourney v5 prompt dataset (2023), https://huggingface.co/datasets/tarungupta83/MidJourney_v5_Prompt_dataset 11
6. (Jan 2024), https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240AB1824 2, 3
7. (Feb 2024), https://photutorial.com/midjourney-statistics/ 2, 3
8. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 22
9. Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A.I., Babaei, H., LeJeune, D., Siahkoohi, A., Baraniuk, R.G.: Self-consuming generative models go mad (2023) 2, 3
10. Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: International conference on machine learning. pp. 233–242. PMLR (2017) 8
11. Bamatraf, A., Ibrahim, R., Salleh, M.N.B.M.: Digital watermarking algorithm using lsb. In: 2010 International Conference on Computer Applications and Industrial Electronics. pp. 155–159 (2010). https://doi.org/10.1109/ICCAIE.2010.5735066 1, 4, 11, 19
12. Boland, F., O'Ruanaidh, J., Dautzenberg, C.: Watermarking digital images for copyright protection. In: Fifth International Conference on Image Processing and its Applications, 1995. pp. 326–330 (1995). https://doi.org/10.1049/cp:19950674 4
13. Boroumand, M., Chen, M., Fridrich, J.: Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security **14**(5), 1181–1193 (2018) 25
14. Byrnes, O., La, W., Wang, H., Ma, C., Xue, M., Wu, Q.: Data hiding with deep learning: A survey unifying digital watermarking and steganography. arXiv preprint arXiv:2107.09287 (2021) 1, 3
15. Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5253–5270 (2023) 2, 3
16. Chang, C.C., Tsai, P., Lin, C.C.: Svd-based digital image watermarking scheme. Pattern Recognition Letters **26**(10), 1577–1586 (2005) 4
17. Cheetham, K.D., Joshua: Fake trump arrest photos: How to spot an ai-generated image (Mar 2023), https://www.bbc.com/news/world-us-canada-65069316 3
18. Cheng, D., Li, X., Li, W.H., Lu, C., Li, F., Zhao, H., Zheng, W.S.: Large-scale visible watermark detection and removal with deep convolutional networks. In: Pattern Recognition and Computer Vision: First Chinese Conference, PRCV 2018, Guangzhou, China, November 23-26, 2018, Proceedings, Part III 1. pp. 27–40. Springer (2018) 4, 5

19. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848 5, 11
20. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. arXiv preprint arXiv:2303.15435 (2023) 1, 4, 11, 19, 21
21. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset (Mar 2007) 11
22. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 3
23. Jiang, Z., Zhang, J., Gong, N.Z.: Evading watermark based detection of ai-generated content. arXiv preprint arXiv:2305.03807 (2023) 20
24. Kheddar, H., Hemis, M., Himeur, Y., Megías, D., Amira, A.: Deep learning for steganalysis of diverse data types: A review of methods, taxonomy, challenges and future directions. Neurocomputing p. 127528 (2024) 25
25. Li, G., Chen, Y., Zhang, J., Li, J., Guo, S., Zhang, T.: Towards the vulnerability of watermarking artificial intelligence generated content. arXiv preprint arXiv:2310.07726 (2023) 20
26. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023) 22
27. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: IEEE International Conference on Computer Vision (ICCV) (2021) 21
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 11
29. Lu, Z., Huang, D., Bai, L., Liu, X., Qu, J., Ouyang, W.: Seeing is not always believing: A quantitative study on human perception of ai-generated images. arXiv preprint arXiv:2304.13023 (2023) 3
30. Lukas, N., Diaa, A., Fenaux, L., Kerschbaum, F.: Leveraging optimization for adaptive attacks on image watermarks. arXiv preprint arXiv:2309.16952 (2023) 20
31. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017) 20
32. Navas, K.A., Ajay, M.C., Lekshmi, M., Archana, T.S., Sasikumar, M.: Dwt-dct-svd based watermarking. In: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08). pp. 271–274 (2008). https://doi.org/10.1109/COMSWA.2008.4554423 1, 4, 5, 11, 19, 22
33. OpenAI: Watermark in dall·e 3 (2023), https://help.openai.com/en/articles/8912793-c2pa-in-dall-e-3 1, 5
34. O'Ruanaidh, J.J., Pun, T.: Rotation, scale and translation invariant digital image watermarking. In: Proceedings of International Conference on Image Processing. vol. 1, pp. 536–539. IEEE (1997) 4
35. Pan, M., Zeng, Y., Lyu, L., Lin, X., Jia, R.: ASSET: Robust backdoor data detection across a multiplicity of deep learning paradigms. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 2725–2742. USENIX Association, Anaheim, CA (Aug 2023), https://www.usenix.org/conference/usenixsecurity23/presentation/pan 5, 6

36. Qi, X., Xie, T., Wang, J.T., Wu, T., Mahloujifar, S., Mittal, P.: Towards a proactive {ML} approach for detecting backdoor poison samples. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 1685–1702 (2023) 5
37. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 11, 22
38. Saberi, M., Sadasivan, V.S., Rezaei, K., Kumar, A., Chegini, A., Wang, W., Feizi, S.: Robustness of ai-image detectors: Fundamental limits and practical attacks. arXiv preprint arXiv:2310.00076 (2023) 20
39. Sandoval-Segura, P., Singla, V., Geiping, J., Goldblum, M., Goldstein, T., Jacobs, D.: Autoregressive perturbations for data poisoning. Advances in Neural Information Processing Systems **35**, 27374–27386 (2022) 21
40. Santoyo-Garcia, H., Fragoso-Navarro, E., Reyes-Reyes, R., Sanchez-Perez, G., Nakano-Miyatake, M., Perez-Meana, H.: An automatic visible watermark detection method using total variation. In: 2017 5th International Workshop on Biometrics and Forensics (IWBF). pp. 1–5. IEEE (2017) 4, 5
41. Shan, S., Cryan, J., Wenger, E., Zheng, H., Hanocka, R., Zhao, B.Y.: Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 2187–2204 (2023) 21
42. Singh, H.K., Singh, A.K.: Comprehensive review of watermarking techniques in deep-learning environments. Journal of Electronic Imaging **32**(03) (Nov 2022). https://doi.org/10.1117/1.jei.32.3.031804 1, 3
43. Sohn, K., Li, C.L., Yoon, J., Jin, M., Pfister, T.: Learning and evaluating representations for deep one-class classification. arXiv preprint arXiv:2011.02578 (2020) 5
44. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 3
45. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2117–2126 (2020) 1, 4, 11, 19, 20, 21
46. Villalobos, P., Sevilla, J., Heim, L., Besiroglu, T., Hobbhahn, M., Ho, A.: Will we run out of data? an analysis of the limits of scaling datasets in machine learning. arXiv preprint arXiv:2211.04325 (2022) 3
47. Wang, Z., Chen, C., Lyu, L., Metaxas, D.N., Ma, S.: Diagnosis: Detecting unauthorized data usages in text-to-image diffusion models. In: The Twelfth International Conference on Learning Representations (2024) 21
48. Wang, Z.J., Montoya, E., Munechika, D., Yang, H., Hoover, B., Chau, D.H.: Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. arXiv preprint arXiv:2210.14896 (2022) 11
49. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. arXiv preprint arXiv:2305.20030 (2023) 1, 4, 11, 19, 20, 21
50. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16133–16142 (June 2023) 11
51. Zavrtanik, V., Kristan, M., Skočaj, D.: Reconstruction by inpainting for visual anomaly detection. Pattern Recognition **112**, 107706 (2021) 5

52. Zeng, Y., Park, W., Mao, Z.M., Jia, R.: Rethinking the backdoor attacks' triggers: A frequency perspective. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16473–16481 (2021) 21
53. Zhang, C., Lin, C., Benz, P., Chen, K., Zhang, W., Kweon, I.S.: A brief survey on deep learning based data hiding. arXiv preprint arXiv:2103.01607 (2021) 1
54. Zhao, X., Zhang, K., Su, Z., Vasan, S., Grishchenko, I., Kruegel, C., Vigna, G., Wang, Y.X., Li, L.: Invisible image watermarks are provably removable using generative ai (2023) 20
55. Zhong, X., Das, A., Alrasheedi, F., Tanvir, A.: Deep learning based image watermarking: A brief survey. arXiv preprint arXiv:2308.04603 (2023) 1, 3
56. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 657–672 (2018) 1, 4, 11, 19, 21

Finding needles in a haystack: A Black-Box Approach to Invisible Watermark Detection

Supplementary Material

A Detailed Experiments Setting & Hyperparameters

In this Section, we will detail the report the experiments setting & hyperparameters that we used in the §4:

Hardware & Software. All experiments were conducted on a server equipped with 4 NVIDIA Tesla V100 GPUs, an AMD EPYC 7763 CPU, and 256 GB of RAM. The software environment includes CUDA 12.0 and PyTorch 2.2.1.

Watermarks. In §4, we use a series of the watermark method to test the performance of WMD, we report their setting and hyperparameters in Table 5.

Table 5: Hyperparameter settings for different watermarking methods used in the experiments.

Watermark	Embedded Bits	Parameters	Values
LSB [11]	64 bits	NA	NA
DctDwtSVD [32]	64 bits	Scales (Y, U, V)	0, 36, 0
		Block	64
HiDDeN [56]	64 bits	Crop	0.2-0.25
		Cropout	0.55-0.6
		Dropout	0.55-0.6
		JPEG	0.8
StegaStamp [45]	64 bits	Brightness	0.3
		Random Noise	0.02
		Saturation	1.0
		Hue	0.1
		Contrast	0.5-1.5
		JPEG	0.5
StableSignature [20]	64 bits	Diffusion Model	Stable Diffusion V1.4
Tree-Ring [49]	NA	Diffusion Model	Stable Diffusion V1.4
		Type	$\text{Tree-Ring}_{\textbf{Rings}}$

Detection. We will report the hyperparameters and settings used for WMD in §4. The results are presented in Table 6.

Table 6: Hyperparameter settings for the WMD watermark detection method.

Parameters	Values
Optimizer	AdamW
Base Learning Rate	1e-4
Weight Decay	0.01
Momentum	$\beta 1$,$\beta 2$ = 0.9,0.999
Batch Size	32
Training Epochs	50
Pruning Rate	0.10
Pruning Interval	10

B Wider Applications

B.1 Watermark removal attacks

Several watermark removal attacks [23, 30, 38, 54] have been proposed to remove or break the watermark in an image, causing the decoding algorithm to fail. These methods can be classified into two categories: black-box and white-box. Black-box methods, such as Regenerate [54], employ a diffusion model to regenerate the watermarked image, hoping that the prior knowledge from the diffusion model will remove the watermark. However, such methods have limited removal performance and fail when attempting to remove advanced watermarks like StegaStamp [45] or Tree-Ring [49]. White-box algorithms, such as Warfare [25], WEvade-B-S [23], and SurPGD [38], require a pure watermarked dataset and a clean dataset to train a surrogate model and perform PGD [31] adversarial attacks on the surrogate model to achieve watermark removal. However, obtaining a pure watermarked dataset in real life is challenging, as discussed in §2.3.

WMD bridges the gap between white-box and black-box removal attacks. As WMD can detect watermarks in real-world, watermark-blended detection datasets, we can use the detection results to form a relatively pure watermarked dataset, enabling downstream black-box watermark removal attacks. We present the results in Table 7. In this evaluation, we increase the dataset size to 50,000 while maintaining the watermark ratio at 5%. The PGD attack settings are kept the same as in SurPGD [38].

Table 7: PGD watermark removal attack results using WMD for watermark detection. Lower AUC scores indicate better removal performance.

	HiDDeN	SS	SSig	TR
AUC↓	0.572	0.613	0.552	0.504

The results demonstrate that by leveraging WMD's watermark detection capabilities to create a relatively pure watermarked dataset, the effectiveness of

black-box watermark removal attacks can be significantly improved. The lower AUC scores indicate that the PGD attack, guided by WMD's detection, achieves better removal performance across various watermarking methods, including HiDDeN [56], StegaStamp (SS) [45], StableSignature (SSig) [20], and Tree-Ring (TR) [49]. This highlights the potential of WMD to support and enhance watermark removal attacks in real-world scenarios where pure watermarked datasets are not readily available.

B.2 Filtering invisible anomalies

Besides watermarks, numerous techniques have been proposed to insert invisible information into image datasets for various purposes, such as data poisoning [27,52], tracing dataset usage [**?**], or preventing models from learning specific information [39,41]. As our framework formulation in §3.3 shows, WMD is capable of detecting any data that does not follow the same distribution as clean data. This naturally raises the question: can WMD also detect these special samples in the dataset? To investigate this, we select representative works from each category. For data poisoning, we choose SSBA [27], which uses an autoencoder to generate backdoor samples and then poisons the downstream model to predict the samples to a target class with a pre-defined backdoor trigger. Another data poisoning method is Frequency [52], which embeds frequency-adaptive noise into the dataset to achieve a backdoor attack while evading frequency-based detection. For dataset tracing, we use DIAGNOSIS [47], which inserts imperceptible distortions into images. The downstream trained diffusion model learns these distortions and generates examples with similar distortions, enabling the tracing of the model's training dataset source. Other methods include unlearnable examples, such as AR [39], which inserts autoregressive noise into images, causing the downstream classifier to focus on learning the noise rather than the image features. This protects the visual information in the image and leads to poor model performance on clean images. GLAZE [41] inserts invisible optimized noise into images, causing diffusion models to fail to learn the visual information and protecting artists' work from being stolen by diffusion models. We maintain the same settings as the evaluation in §4 and adjust the insertion ratio to match their original work. The results are presented in Table 8.

The evaluation results demonstrate that WMD is highly effective in detecting various types of invisible information inserted into image datasets. It achieves high AUC scores for data poisoning methods like SSBA and Frequency, as well as for unlearnable examples such as AR and GLAZE. However, WMD's performance on dataset tracing using DIAGNOSIS is relatively lower, suggesting that the imperceptible distortions inserted by this method may be more challenging to detect. Overall, the results highlight the versatility and effectiveness of WMD in identifying a wide range of invisible information in image datasets, demonstrating its potential as a powerful tool for ensuring dataset integrity and protecting against malicious manipulations.

Table 8: Detection results for various types of invisible information inserted into image datasets, along with their respective insertion ratios.

	Data Poisoning		Dataset Tracing	Unlernable Example	
	SSBA 10%	Frequency 10%	DIAGNOSIS 20%	AR 10%	GLAZE 25%
AUC (↑)	0.987	0.975	0.653	0.903	0.934
TPR (↑) @ 0.1 FPR	0.944	0.931	0.554	0.823	0.856
FPR (↓) @ 0.9 TPR	0.010	0.016	0.273	0.048	0.014

B.3 Synthesis clean data

In some scenarios, obtaining a clean dataset to serve as a reference for WMD can be challenging, which may hinder the effectiveness of the watermark detection process. To mitigate the challenge of obtaining clean data in some scenarios, we explore the use of synthesized data to extend the clean dataset. We choose the DctDwtSvd [32] watermarking method, set the watermark ratio to 5%, and use a detection dataset size of 10,000. We then evaluate several synthesis methods:

- **Case 1, Detection Dataset to Text to Image:** In this case, we use BLIP-2 [26] as the caption model to generate captions for the detection dataset. Since the watermarks are invisible, we expect the generated captions to not contain watermark information. We then use these generated prompts to feed the generation model and synthesize high-quality clean images for the clean dataset.
- **Case 2, Clean Dataset to Text to Image:** This case follows the same setting as Case 1, but the captions are generated from the clean dataset. As the amount of real clean data may be less than 50% of the total clean dataset, we use the same prompt multiple times with different random seeds to generate different images.
- **Case 3, Clean Dataset Image Variation:** This case has the same settings as **Case 2**, but instead of using prompts to generate the synthesized images, we directly use the clean images as the condition to guide the generation [1] and obtain variations of the clean data.
- **Case 4, Synthesized Text Prompt to Image:** In this case, we use the captions from **Case 2** and input them into ChatGPT [8] to generate variations of these prompts. We then use these prompts to synthesize clean images.

For all generation models, we use Stable Diffusion V2 [37] with a step size of 100, keeping all other hyperparameters at their default values. The results are shown in Table 9.

The evaluation results show that using synthesized data to extend the clean dataset can be effective, but the performance degrades as the ratio of synthesized data increases. Case 1, which generates captions from the detection dataset and

Table 9: AUC scores for different ratios of synthesized data in the clean dataset. The results demonstrate the impact of using synthesized data on watermark detection performance.

	0%	20%	40%	60%	80%
Case 1	0.968	0.966	0.958	0.784	0.684
Case 2		0.942	0.911	0.745	0.689
Case 3		0.910	0.886	0.613	0.500
Case 4		0.940	0.924	0.734	0.702

uses them to synthesize clean images, achieves the best performance among the synthesis methods. This suggests that the captions generated from the detection dataset capture relevant content while excluding watermark information. Cases 2 and 4, which use captions from the clean dataset or variations of those captions, also demonstrate good performance, although slightly lower than Case 1. Case 3, which directly uses image variations of the clean data, shows the lowest performance among the synthesis methods, indicating that image-based variations may introduce more noise or artifacts that affect watermark detection. Overall, these results highlight the potential of using synthesized data to mitigate the lack of clean data, but the ratio of synthesized data should be carefully considered to maintain high watermark detection performance.

C Further Ablation Study

Table 10: Watermark detection performance (AUC) across different combinations of clean (reference) datasets and detection (watermarked) datasets. Diagonal elements represent same-domain scenarios, while off-diagonal elements represent cross-domain scenarios. Watermark detection is most effective when the clean and detection datasets are from the same domain, with some domains generalizing better than others in cross-domain cases.

		Clean					
		ImageNet	COCO	Caltech-256	CelebA	iNaturalist	Stanford Cars
Detection	ImageNet	0.968	0.957	0.942	0.612	0.909	0.632
	COCO	0.948	0.961	0.916	0.607	0.914	0.658
	Caltech-256	0.916	0.937	0.955	0.599	0.893	0.686

Mismatch Dataset. The ablation study in Table 10 investigates the impact of domain mismatch between the clean (reference) and detection (watermarked) datasets on the performance of WMD. The highest detection performance, with

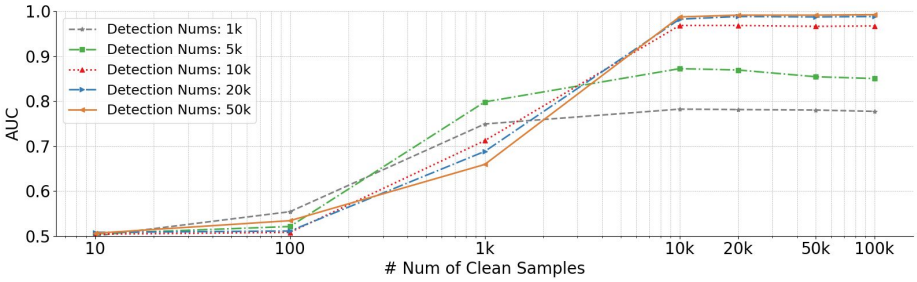

Fig. 4: Watermark detection performance (measured by AUC) with varying sizes of the clean reference dataset and different numbers of clean samples used during training. Larger reference datasets and more clean training samples generally lead to better detection performance, with diminishing returns after a certain point.

AUC scores above 0.9, is achieved when both datasets are from the same domain (diagonal elements). Cross-domain scenarios (off-diagonal elements) exhibit varying performance, with some combinations, like ImageNet and COCO, showing strong generalization (AUC 0.957), while others, like Celeb and other domains, have limited effectiveness (AUC 0.599 to 0.612).

The study highlights the importance of similarity between the clean and detection datasets for optimal performance. Domain mismatch can degrade the model's ability to generalize and learn distinguishing features. To mitigate this, it is recommended to use a clean dataset that closely matches the detection dataset or exhibits strong generalization, such as ImageNet. Further research could explore domain adaptation techniques and incorporating diverse clean images to improve robustness to dataset mismatch.

Clean-set Size. The ablation study in Figure 3 investigates the impact of the clean reference dataset size and the number of clean samples used during training on the watermark detection performance of WMD. The results show that increasing the dataset size and the number of clean training samples leads to better detection performance, with AUC scores consistently exceeding 0.9 for the largest dataset size (50,000 samples). However, the performance gains exhibit diminishing returns beyond a certain point, suggesting a trade-off between computational cost and marginal improvements. The study also reveals that using a small detection dataset can limit the detection performance due to overfitting and the inability to generalize to the full watermark distribution. To mitigate this issue, it is crucial to ensure that both the clean reference dataset and the detection dataset are sufficiently large and diverse to capture the variability in watermark patterns and image characteristics, enabling the model to learn robust and generalizable features for effective watermark detection.

D Discussion

Relation to Steganalysis. Steganalysis [13, 24] refers to methods that detect secret messages embedded in digital media using steganography. Given the similarity between steganography and watermarking, WMD may remind people of steganalysis. Although both watermarking and steganography involve embedding information into digital media, they serve different purposes. Watermarking aims to protect intellectual property rights and ensure the authenticity of digital content by embedding a unique identifier or signature. The embedded watermark is typically designed to be robust against various image processing operations and attacks, and its presence should be detectable even if the image undergoes modifications. On the other hand, steganography focuses on concealing the existence of a hidden message within digital media, prioritizing undetectability over robustness. The goal is to communicate secretly without raising suspicion, and the success of steganography relies on the inability of an adversary to distinguish between normal and steganographic media.

Besides the difference between steganography and watermarking, the detection assumptions and goals of WMD and steganalysis are different. Steganalysis requires a paired dataset, where the original image (cover) and its steganographic version (stego) are provided for analysis and detection. This means that steganalysis relies on knowledge of the specific steganography algorithm and access to the original version of the image. In contrast, WMD does not require such knowledge or the original version of the watermarked image. Furthermore, the detection goals of steganalysis and WMD differ. Steganalysis focuses on training on a labeled dataset and aims to generalize to a testing dataset, ensuring that both datasets use the same steganography method. The success of steganalysis relies on the consistency of the steganography algorithm across the training and testing data. On the other hand, WMD operates in a more realistic and challenging scenario where there is no separate training and testing dataset. Instead, WMD performs detection solely on the given detection dataset, aiming to split it into watermarked and clean parts without prior knowledge of the watermarking algorithm or access to labeled data. This makes WMD more flexible and applicable to real-world situations where the watermarking method may be unknown, and labeled data is unavailable. While steganalysis relies on the consistency of the steganography algorithm between the training and testing data, WMD can handle the presence of multiple, unknown watermarking techniques within a single dataset. This makes WMD a more versatile tool for detecting watermarks in real-world scenarios where the watermarking methods may be diverse and unknown.

In summary, although WMD and steganalysis share the goal of detecting hidden information in digital media, they differ in their assumptions, requirements, and detection goals.

Safety & Security Concern. While WMD demonstrates significant potential in detecting invisible watermarks, it is essential to address the safety and security concerns that may arise from its use. One major concern is the possibility of WMD leaking secret information embedded in the watermarks. As WMD is

designed to detect the presence of watermarks, it may inadvertently expose sensitive data, such as copyright information, ownership details, or hidden messages, to unauthorized parties. This could compromise the privacy and security of the watermark owners and the intended recipients of the embedded information.

Another concern is the potential misuse of WMD to support watermark removal attacks. As discussed earlier, WMD can be used to create a relatively pure watermarked dataset, which can then be exploited to train models for watermark removal. This may encourage malicious actors to use WMD to circumvent copyright protection and remove watermarks from digital content without permission. Such actions could undermine the effectiveness of watermarking as a security measure and lead to the infringement of intellectual property rights.

To mitigate these concerns, it is crucial to develop safeguards and responsible usage guidelines for WMD. One approach could be to incorporate a mechanism that prevents the extraction or decoding of the actual watermark information, ensuring that only the presence of watermarks is detected without revealing the embedded data. Additionally, implementing access controls and authentication measures could help restrict the use of WMD to authorized parties and prevent its misuse for malicious purposes.

Law & Policy Implications. The development and use of WMD for invisible watermark detection raise important legal and policy considerations. As watermarking plays a crucial role in protecting intellectual property rights and ensuring the authenticity of digital content, the ability to detect and identify watermarks has significant implications for copyright law and digital rights management.

From a legal perspective, WMD could serve as a valuable tool for copyright holders to enforce their rights and detect unauthorized use of their watermarked content. By enabling the detection of invisible watermarks, WMD can help identify instances of copyright infringement and provide evidence for legal action. This could strengthen the position of copyright holders and deter potential infringers from misusing watermarked content.

However, the use of WMD also raises concerns about privacy and the potential for abuse. If WMD falls into the wrong hands, it could be used to illegally remove watermarks from copyrighted content, facilitating unauthorized distribution and use. This could undermine the effectiveness of watermarking as a copyright protection measure and lead to financial losses for content creators and owners.

To address these concerns, policymakers may need to consider updating existing copyright laws and regulations to account for the emergence of advanced watermark detection techniques like WMD. This could involve clarifying the legal status of watermark detection tools, defining the permissible uses of such tools, and establishing penalties for their misuse.

E Visualization

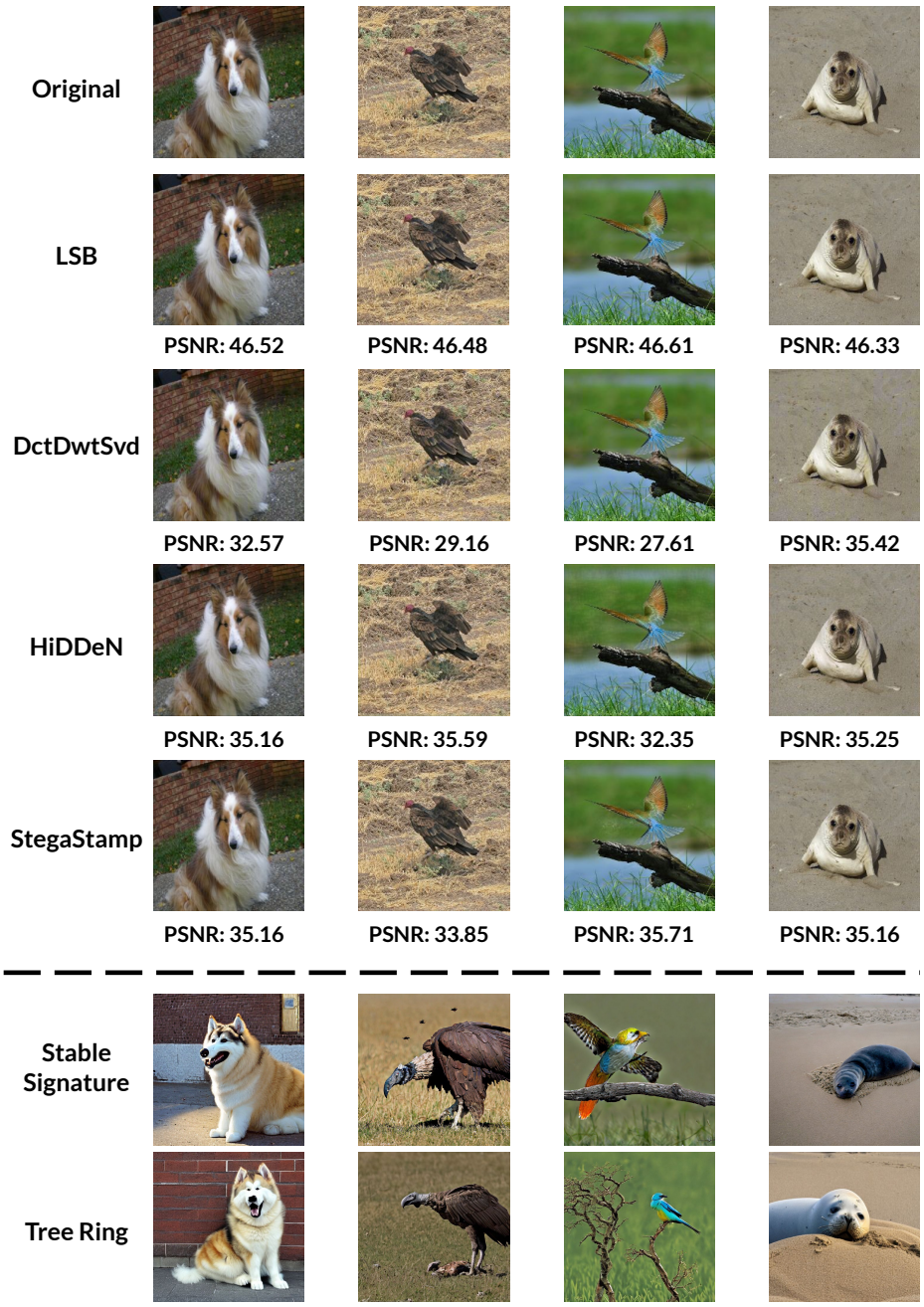

Fig. 5: Visual examples of original images and their watermarked counterparts using different watermarking methods. The top row shows the original images. The PSNR values are provided for each post-processing watermarked image, lower PSNR indicating the higher level of distortion introduced by the watermarking process.