arXiv:2603.01067v1 [cs.CR] 1 Mar 2026

# Hide&Seek：通过逐像素重建以极低成本去除图像水印

Huajie Chen[1], Tianqing Zhu[*], Hailin Yang[1], Yuchen Zhong[1], Yang Zhang[2], Hui Sun[1], Heng Xu[1],Zuobin Ying[1], Lihua Yin[3], Wanlei Zhou[1]

## 摘要

水印技术已成为防范机器生成图像（MGI）遭到滥用的一项关键防御手段。然而，这些防护措施的鲁棒性仍未得到充分研究。为揭示当前最先进的主动式图像水印防御的局限性，我们提出HIDE&SEEK （ HS ），这是一套灵活且经济高效的攻击方法，能够可靠地去除嵌入式水印，同时保持较高的视觉保真度。与以往方法不同，HS 只关注对水印最关键的像素。HS 分为两个阶段，即 HIDE 和 SEEK。在 HIDE 阶段，HS 会识别并掩蔽对水印结构最关键的像素；在 SEEK 阶段，HS 随后利用生成模型仅对这些像素进行逐像素重建，而图像其余部分保持不变。这种针对性策略使 HS 无需查询且无需知识，既不需要访问水印检测器，也不需要了解底层水印方案。针对多种最先进水印方法开展的大量实验表明，HS 能够持续击败现有防御，并且性能优于已有的去除攻击，同时实现更彻底的水印去除和更高的图像质量。我们的发现揭示了当前水印方法存在的根本性漏洞，并对其作为机器生成图像防护手段的鲁棒性提出质疑。代码将在论文录用后发布。

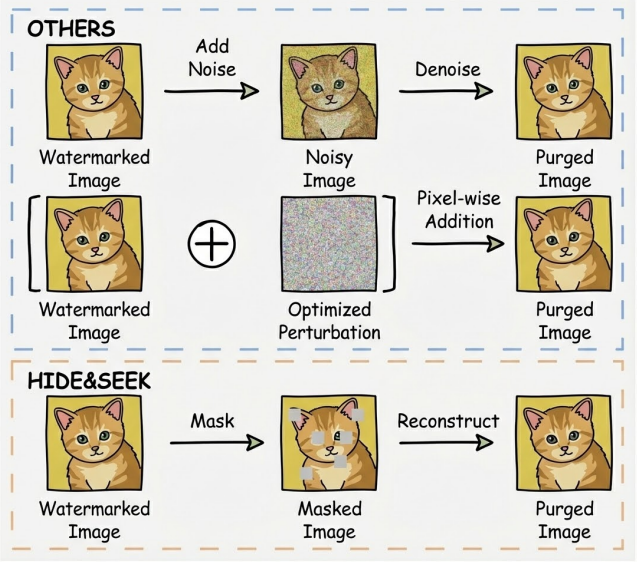



**图 1：HIDE&SEEK 概览。与现有攻击通过对带水印图像进行全局修改来移除嵌入式水印不同，HIDE&SEEK 会定位脆弱像素并对其进行修改，以清除水印，同时保持较高的视觉质量。**

## CCS 概念

• **安全与隐私 →数字版权管理**;

## 关键词

AIGC水印, 安全攻击, 深度学习

**ACM 参考格式：**

Huajie Chen[1], Tianqing Zhu[*, 1], Hailin Yang[1], Yuchen Zhong[1], Yang Zhang[2], Hui Sun[1], Heng Xu[1], Zuobin Ying[1],Lihua Yin[3],Wanlei Zhou[1]. 2018. Hide&Seek：通过逐像素重建以极低成本移除图像水印。载于请根据版权确认邮件中的信息，填写正确的会议标题（会议缩写“*XX*”）。ACM，纽约，NY，美国，共17页。https://doi.org/XXXXXXX.XXXXXXX

## 1 引言

机器生成图像（MGI）已在媒体 [40], 设计 [28], 和娱乐领域 [2] 等许多领域得到广泛应用。尽管广受欢迎且具有创造潜力，MGI也日益被滥用于恶意目的，包括政治宣传 [24], 、露骨内容 [19], 和仇恨言论 [25]。这促使人们开发多种防御机制，以确保负责任地使用和分发机器生成图像 [18]。

总体而言，这些防御可分为两类：被动机器生成图像检测和主动机器生成图像水印。被动机器生成图像检测通常通过训练深度神经网络来区分机器生成图像与真实图像 [6, 36]。尽管这类方法在分布内性能方面表现出色，但往往难以泛化至未见过的生成模型或扩散模型 [16]。另一方面，主动水印技术会在生成图像时嵌入不可感知、可验证信号（即水印），从而实现事后归因和问责 [5, 10, 18, 27, 31, 33, 34, 39]。凭借其模型无关保证，水印技术已成为研究人员和业界青睐的方法。

然而，水印研究人员与对手之间的猫鼠博弈仍在持续。水印技术仍容易受到移除策略的攻击，例如再生成 [1, 37], 对抗 [9, 23], 和扰动攻击 [12], ，这些攻击旨在消除嵌入的信号，同时保持图像保真度。然而，现有攻击在攻击有效性、图像保真度和计算效率之间面临根本性的权衡。迄今为止，还没有任何一种攻击能同时实现这三者，从而在现实场景中对水印方案造成实质性破坏。这促使我们着手弥补这一空白。

*通讯作者。 [1]澳门城市大学。 [2]CISPA亥姆霍兹信息安全中心。 [3]广州大学会议缩写“*XX*”，纽约州伍德斯托克© 2018 版权所有归所有者/作者所有。出版权授权给 ACM。ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

本文介绍了一类实用的水印移除机制，它们能够保持较高的攻击成功率、保留视觉保真度，同时实现较高的计算效率。我们的研究结果对当前水印方案在现实威胁模型下的鲁棒性提出了挑战，并为不断演变的 MGI 水印技术安全格局提供了新的见解。

**我们的工作**。为了揭示最先进的水印防御方法的局限性，我们提出了**Hide&Seek**，这是一种通过逐像素重建来移除图像水印的攻击。通常，针对 MGI 水印方案的攻击具有相同的目标和类似的方法：在带水印图像上仅添加极少的修改，以生成净化图像。此外，修改必须改变带水印图像的高频和低频分量，从而破坏非语义水印和语义水印 [12]。因此，我们的方法由两个阶段组成，即“隐藏”和“寻找”：前者分别在频域和语义域中最大化带水印图像与净化图像之间的差异，后者则最小化两者之间的视觉损失。

我们首先从一个名为 HIDE&SEEK Naive（HSN）的简单版本开始。HSN 的核心思想是修改带水印图像的特定区域，以在频域和语义域中产生差异，并生成净化图像。为实现这一目标，在*HIDE*阶段，带水印图像被划分为多个图像块，并按照选定的掩码策略和掩码比例进行掩码处理。接着，在*SEEK*阶段，使用自回归模型重建被掩码的图像块，以产生差异。通过调整掩码策略和掩码比例，对手可以只修改带水印图像的一部分，同时保持其余部分不变。因此，HSN 生成的净化图像具有最小的视觉损失。

尽管 HSN 在评估中展现出足够的攻击性能，其掩码和重建方案仍有改进空间。例如，物体边缘的像素对图像高频分量的影响更大。因此，如果对手能够衡量各像素的影响，并根据影响程度的排名对像素施加不同的扰动，净化图像就能在确保攻击有效性的同时获得更高的保真度。

随后，我们进一步将 HSN 开发为增强版本，命名为 HIDE&SEEK Plus（HS+）。在 HS+ 中，对手旨在定位对带水印图像的频域和语义域影响最大的一组像素（称为脆弱像素）。根据其脆弱程度对这些像素进行排序，并按照影响排名从高到低逐像素重建，以产生最大的差异，其中最脆弱的像素最后重建。在 *HIDE* 阶段，对手使用预训练掩码模型，输入任意带水印图像并生成预测掩码。预测掩码包含像素脆弱性的信息，用于指导掩码处理和重建过程。随后，对脆弱像素进行掩码处理，以便重建。在 *SEEK* 阶段，对手使用训练好的像素生成器，从掩码图像重建出净化图像。净化图像逐像素重建，其中最脆弱的像素最后重建。其思路是利用自回归模型的内在属性，即前一轮输出会成为下一轮迭代的输入。因此，在迭代重建过程中，预测偏差会不断累积。同时，生成器仅使用干净图像进行训练，从而保证净化图像的视觉质量和感知质量。

在 Hide&Seek 的设计与实现过程中，我们遇到了几个研究问题：**研究问题1**）如何以最少的掩码像素创建一种掩码，使其在频域和语义域中都产生足够的差异？**研究问题2**）如何确保净化图像具有较高的保真度，同时不保留嵌入式水印？**研究问题3**）如何在降低计算成本的同时保证攻击性能？

针对这些问题，我们的答案如下：**答案1**）在 HSN 中，通过选择合适的策略和合理的掩码比例，可以高效地创建有效掩码。对手可以通过调整策略和比例来控制修改的范围和强度。在 HS+ 中，我们使用专门设计的损失函数训练掩码模型，以预测覆盖脆弱像素的最优掩码。**答案2**）在 HSN 中，我们训练自回归模型以保持较高的图像保真度。在 HS+ 中，我们使用像素损失和感知损失训练逐像素生成器，以递归方式预测掩码像素。两个生成器都在干净图像数据集上训练，以确保净化图像无水印，且在视觉上与带水印图像相似。**答案3**）掩码模型和生成器都可以高效训练，并且具有良好的泛化能力。因此，无需针对特定图像进行攻击优化。

Hide&Seek通过全面实验进行了评估。与来自顶级会议的其他最先进的攻击方法相比，HSN和HS+均展现出更优的攻击性能和视觉质量，同时计算成本适中。此外，Hide&Seek揭示了当前水印防御的脆弱性。这要求在该领域开展进一步研究，以防止人工智能生成内容的滥用。我们的主要贡献如下。

- 我们提出Hide&Seek，一种通用的黑盒无查询攻击，能够有效去除机器生成图像中的水印，同时保持较高的视觉质量。
- 我们设计了两个版本的Hide&Seek，以揭示当前最先进的水印防御方法的脆弱性。HSN使对手能够通过多种掩码策略重建被掩码图像，从而对带水印图像发起高效且有效的攻击。此外，HS+ 生成更精细的掩码，并采用逐像素重建技术，结合理论分析提升攻击性能。
- 我们开展了全面实验，以证明HIDE&SEEK相较于最先进的基线方法具有更优的性能。

## 2 预备知识与相关工作

### 2.1 针对机器生成图像的防御

**被动检测**。神经网络检测器经过训练，能够根据高频伪影和语义伪造性区分机器生成图像与真实图像 [6, 14, 16, 36]。简而言之，这些方法

**表1：符号说明**

| 符号 | 定义 |
|---|---|
| $X, \tilde{X}$ | 原始/重建图像 |
| $L_{\text{pix}}$ | 像素损失 |
| $L_{\text{frq}}$ | 频率损失 |
| $L_{\text{pct}}$ | 感知损失 |
| $L_{\text{smt}}$ | 语义损失 |
| $C, c$ | 通道的数量/索引 |
| $w, h$ | 图像的宽度/高度 |
| $M$ | 掩码 |
| $I, \bar{I}, \hat{I}$ | 带水印的图像/掩码处理后的图像/净化图像 |
| $E, D$ | 编码器/解码器模型 |
| $H, G$ | 掩码模型/像素生成器 |
| $(x,\ y)$ | 图像像素的空间坐标 |
| $(u,\ v)$ | 图像像素的频率坐标 |
| $F$ | 二维离散傅里叶变换 |

旨在训练一个检测器 D，使其

$$\mathbb{1}_{\mathbb{R}^+}(\mathcal{D}(X)) = \begin{cases} 0 & \text{if } X \text{ is real} \\ 1 & \text{if } X \text{ is Fake} \end{cases}, \tag{2.1}$$

其中，1 是指示函数。当面对传统生成器创建的机器生成图像时，这些方法能够提供有希望的防御效果，但当先进生成器出现时却无法做到这一点 [11]。也就是说，如果图像由未见过的模型生成，这些方法的泛化能力并不理想。

**主动水印**。根据 Kassis 等人 [12], 的分类，当前用于机器生成图像的水印技术主要分为两类：非语义水印和语义水印。这些水印方法将水印嵌入与不同频率层级相关联的全局频谱振幅中。通常，这些方法需要找到一对嵌入函数和提取函数 E(·) 与 R(·)，使得对于任意机器生成图像 $X$，

$$\min \mathbb{E}\Big[ l(\mathcal{R}(\mathcal{E}(X, m) + \delta), m) \Big], \tag{2.2}$$

其中，$l$ 表示某个距离度量函数；$\delta$ 是某种扰动噪声；$m$ 表示比特串水印。

非语义水印技术方法 [5, 18, 30, 32–34, 39] 将水印嵌入图像的高频分量中。这些方法并不旨在改变图像内容，因此会将水印嵌入人眼不太敏感的高频分量中。语义水印方法 [27, 31] 则将水印嵌入图像的低频分量中，这会大幅改变图像内容。原因在于，能量分布主要集中在低频分量中。因此，这种方法以降低视觉质量为代价增强了鲁棒性。

## 2.2 检测规避攻击

**再生成攻击**。一般来说，再生成攻击首先对带水印图像 $X$ 整体添加噪声 $\delta$，然后从含噪图像中重˜建净化图像 $X$，这与对抗样本的净化 [13, 20, 29]类似。其定义如下：

$$\tilde{X} = G(X + \delta), \tag{2.3}$$

其中 $G$ 表示生成器。这些攻击 [26, 37, 38] 属于黑盒且无需查询的攻击，这意味着对手无需访问检测器。

**扰动攻击**。通过优化扰动噪声 $\delta$，并将其随后引入带水印图像 $X$中，对手无需经过再生成过程即可移除嵌入式水印，这与对抗性扰动 [7]类似。定义如下：

$$\arg\min_{\delta} \mathbb{E}[\|\delta\|^2 + \mathcal{D}(X + \delta)]. \tag{2.4}$$

这类攻击 [9, 12, 17, 23] 会对带水印图像施加全局扰动，以去除嵌入式水印。

## 2.3 掩码自编码器

He 等人 [8]首次提出掩码自编码器（MAE），将其作为可扩展的视觉学习器。MAE 借鉴了自回归语言建模和掩码自编码的理念：对部分数据进行掩码处理，并训练 MAE 预测被掩码的内容。Li 等人 [15] 进一步设计了分形生成模型（FGM），以像素为单位递归生成图像。在递归过程中，下一个预测像素由当前掩码图像的全局状态生成。值得注意的是，与 DDPM 或生成对抗网络（GAN）相比，训练 FGM 相对容易，因为它不会像 GAN 那样出现模型坍塌，也不需要像 DDPM 那样在反向过程中执行多个步骤。训练 FGM 时采用交叉熵损失函数，使其能够像大语言模型中的词元一样预测像素的 RGB 通道。

# 3 问题表述

## 3.1 威胁模型

在这一攻击场景中，我们考虑两个参与方：**受害者**和**对手**。**受害者** 是建立生成式人工智能平台并提供 API 服务的组织或公司，并通过该平台实现盈利。**对手** 是试图规避防御机制并生成有害机器生成图像的恶意用户。

**受害者的目标**。受害者希望确保生成式人工智能平台不会被用于生成传播虚假信息的恶意人工智能生成内容。因此，生成式人工智能平台生成的所有图像都会加上水印，以便进行机器生成图像检测。对于任意可疑图像，受害者都会将其输入检测器网络，提取嵌入式水印，以确定该图像是否为机器生成图像。

**受害者的能力**。受害者可以访问所有现有的模型结构和训练算法。受害者拥有充足的资源，例如数据集和 GPU，足以训练生成式人工智能模型。此外，生成式人工智能 API 会接收查询并返回机器生成的图像。无需向用户提供中间结果。

**对手的目标**。对手旨在利用受害者的生成式人工智能 API 生成高质量的机器生成图像，同时移除其中嵌入的

水印，使任何现有算法都无法将该图像判定为机器生成图像。

**对手能力**。由于资源有限，对手无力购买高质量数据集或租用服务器进行长期大型模型训练。对手不允许直接访问受害者使用的训练数据集和模型结构。对手无法与受害者的检测器交互。不过，对手可以收集一个辅助数据集，其分布与受害者的训练数据集相似。对手还可以使用有限的计算资源训练相对简单的本地模型。

## 3.2 形式化目标

**攻击属性**。我们还可以进一步得出结论：一种良好的水印移除攻击必须具备以下属性：

- **有效性**：该攻击必须确保无法从净化图像中检测或提取出任何水印。
- **保真度**：净化图像不得与带水印图像偏离过多。
- **效率**：攻击的资源与时间消耗应较低。
- **通用性**：该攻击可应用于以任何水印方法添加水印的任何图像。

**目标**。上述攻击要么向带水印图像引入噪声以直接清除嵌入式水印，要么从含噪图像重建净化图像。抽象而言，这些方法可以归结为一句话：对手通过有选择地调整带水印图像的像素值来去除水印。因此，哪些像素需要改变以及像素应偏移多少，共同构成了关键问题。

令 $I \in \mathbb{R}^{C\times w\times h}$ 表示由受害者的生成式人工智能 API 生成的带水印图像，其中 $C$，$w$，$h$分别表示 $I$的通道数、宽度和高度。受害者拥有一个检测器 $D$：$\mathbb{R}^{C\times w\times h} \mapsto \{0, 1\}$，能够判断给定图像是否包含水印。$\hat{}$ 对手的最终目标是创建无水印图像 $I$，亦称 $\hat{}$ 净化图像，使得 $D(I) = 0$，表明 $\hat{}I$中不存在水印。此外，要求 $\hat{}I$尽可能与 $I$相似。随后，我们将该目标表述为一个优化问题：

$$\min_{\hat{I}} \mathbb{E}\left[\left\|\hat{I} - I\right\|^2 - \log\left(1 - D(\hat{I})\right)\right]. \tag{3.1}$$

然而，在我们的设置中，对手不得访问 $D$。还必须考虑图像质量测量的其他指标。因此，优化进一步表述如下。

研究表明，鲁棒水印存在于全局频谱幅度 [12] 中。由此可将水印分为两类：语义水印和非语义水印。一方面，非语义水印是指隐藏在图像全局频谱振幅中的水印，其能量集中在高频分量上。高频分量不易被人眼察觉，因此基于深度学习的水印方案会自然地选择它们。然而，只需对高频分量施加轻微扰动，就能轻易移除这类水印 [3]。另一方面，语义水印则嵌入在低频分量中，而低频分量会显著影响图像的感知效果。因此，这类水印不可避免地会被人眼察觉，但对扰动更加鲁棒。

基于这一假设，攻击目标可进一步表述为以下形式：

$$\max_{\hat{I}} \mathbb{E}\left[d_{\text{frq}}(\hat{I}, I) + d_{\text{smt}}(\hat{I}, I)\right], \tag{3.2}$$

其中，$d_{\text{frq}}(\cdot,\cdot)$ 和 $d_{\text{smt}}(\cdot,\cdot)$ 分别表示两个输入图像在频域和语义域中的距离度量。由此，对手最大化 $\hat{}I$ 与 $I$ 之间的频率差异和语义差异，以清除水印。

同时，对手不希望为了移除水印而过多牺牲图像质量。因此，对手旨在$\hat{}$保留 $I$ 与 $I$ 之间的像素相似度和感知相似度，并将其表述为

$$\min_{\hat{I}} \mathbb{E}\left[d_{\text{pix}}(\hat{I}, I) + d_{\text{pct}}(\hat{I}, I)\right], \tag{3.3}$$

其中，$d_{\text{pix}}$ 和 $d_{\text{pct}}$ 分别表示像素距离和感知相似度的度量。由此，对手最小化两个输入图像之间的像素距离和感知距离，从而维持 $\hat{}I$ 的视觉质量。

最终，我们可以将上述所有优化目标结合起来，推导出以下优化问题：

$$\begin{aligned}\min_{\hat{I}} \mathbb{E}\big[&\left(d_{\text{pix}}(\hat{I}, I) + d_{\text{pct}}(\hat{I}, I)\right)\\ &- \left(d_{\text{frq}}(\hat{I}, I) + d_{\text{smt}}(\hat{I}, I)\right)\big].\end{aligned} \tag{3.4}$$

一般而言，对手旨在以最小的视觉损失，在 $I$ 和 $I$ 的频域和语义域中造成最大的差异 $\hat{}$。

# 4 Hide&Seek

## 4.1 概述

尽管先前研究提出的方法具有出色的攻击性能，我们发现仍有若干问题亟待解决。再生成攻击 [9] 在应对非语义水印方法时效果显著，但面对语义水印方法时却无法取得同样的效果。扰动攻击 [12]对语义和非语义水印方法都展现出良好的效果。然而，扰动造成的净化图像视觉质量损失不可忽视，因为扰动必须影响净化图像的低频分量。此外，对于每张带水印图像，都需要专门优化其扰动噪声，导致计算成本高昂。归根结底，上述攻击都是通过全局修改带水印图像来生成净化图像，这本质上会降低净化图像的视觉质量。

因此，我们设计了水印移除攻击 HIDE&SEEK，它包含两个阶段：*HIDE* 和*SEEK*。总体而言，在*HIDE*阶段，对手试图为任意带水印图像创建掩码，以隐藏 像素；随后，在*SEEK*阶段，对手尝试寻找 隐藏的像素值。通过这两个阶段，对手只修改掩码像素的值，同时保持其余像素不变，从而完成水印移除攻击。与现有攻击相比，HIDE&SEEK 的独特之处在于，它通过部分图像修改来移除水印。

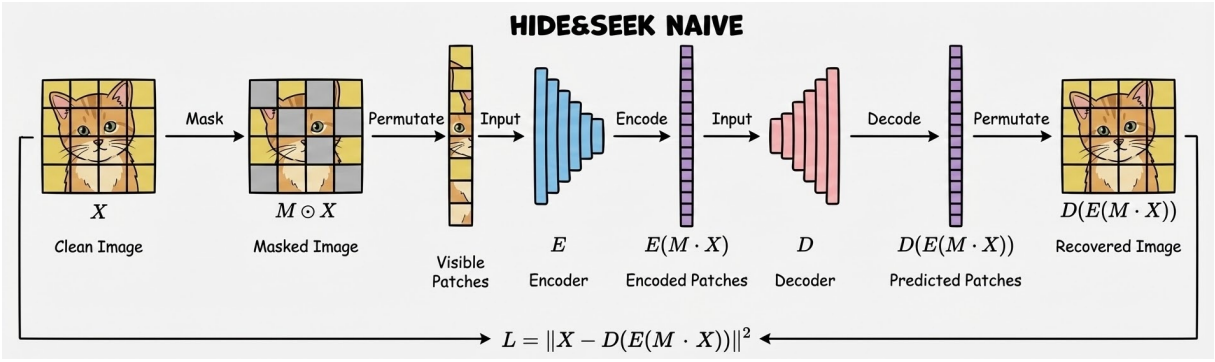



**图 2：HIDE&SEEK 朴素工作流。对手首先使用掩码，按照某种策略和比例对带水印图像进行掩码处理。随后，编码器对可见图块进行编码，并将其与掩码图块拼接，再由解码器解码以生成净化图像。**

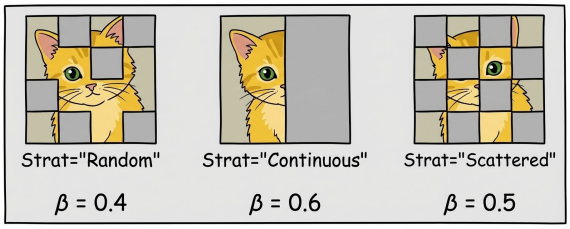



**图 3：掩码策略。HSN 提供三种掩码策略：随机、连续和分散，其中 $\beta$ 控制掩码比例。**

我们进一步开发了该攻击的两个版本，分别为 HIDE&SEEK Naive (HSN) 和 HIDE&SEEK Plus (HS+)。

## 4.2 HIDE&SEEK Naive

HSN 是 HS 的一个简单而有效的版本。掩码由固定算法生成，并采用选定的策略和超参数 $\gamma$ 来控制掩码比例。

**训练**。如算法1和图2所示，为启动HSN，对手首先收集一个干净数据集$\{X^1, \ldots\ldots X^n\}$，其中$X^i \in \mathbb{R}^{C\times w\times h}$表示一幅无水印图像。然后，对手训练生成器模型。在HSN中，掩码自编码器（MAE） [8] 被用作生成器的架构。一个MAE由编码器 $E: \mathbb{R}^{C\times w\times h} \mapsto \mathbb{R}^Q$和解码器 $D: \mathbb{R}^Q \mapsto \mathbb{R}^{C\times w\times h}$组成，其中 $Q$表示潜在空间中嵌入的维度。

在训练过程中，会随机生成一个掩码 $M \in \{0,1\}^{w\times h}$。然后使用$M$对干净图像 $X \in \{X^1, \ldots, X^n\}$进行掩码处理，记作 $M \odot X$，其中 $\odot$是逐元素乘法运算符。接下来， $E$接收 $M \odot X$，并将可见像素编码为词元。掩码像素被转换为词元，并与这些词元拼接在一起，按照 $X$中像素的顺序形成潜在空间中的向量 $E(M \odot X)$。最后， $D$接收 $E(M \odot X)$，利用可见词元预测掩码像素，从而生成净化图像 $D(E(M \odot X))$。通过最小化干净图像 $X$与净化图像 $D(E(M \odot X))$之间的再生损失，

对手完成对 $G$的训练。再生损失定义如下：

$$L = \|X - D(E(M \odot X))\|^2, \tag{4.1}$$

其中仅将掩码像素纳入损失计算，以降低计算成本。

**掩码策略**。不同类型的掩码最终会对净化图像产生不同的效果。如图3所示，我们大致将掩码策略分为三类：随机、连续和分散。图像首先被划分为多个图像块。随机掩码会自发创建；连续掩码通过随机选择一个初始图像块来创建。下一个待掩码的图像块必须与前一个相邻；分散掩码不得包含任何与其他已选图像块相邻的图像块。

这些策略彼此之间的区别在于各自带来的影响 [8]。连续掩码对 $I$低频分量的影响最显著，而分散掩码保留的频率信息最多。随机掩码具有更大的 $\beta$，这意味着在保证重建质量的前提下，可以掩蔽更多像素。但由于其随机性，它对 $I$频域的影响往往会波动。

**评估**。如算法1所示，在评估过程中，对手首先接收一张带水印图像 $I \in \mathbb{R}^{C\times w\times h}$。随后使用一种掩码策略生成掩码$M \in \{0,1\}^{w\times h}$，并通过超参数 $\beta$控制掩码比例。最终，对手ˆ使用训练好的 MAE 获得净化图像 $I = D(E(M \odot I))$。

**局限性**。尽管 HSN 在评估中提供了可靠的水印移除攻击性能，但对带水印图像所施加修改的面积和程度仍可优化。采用所选策略随机生成的掩码需要覆盖图像整个区域相当高的比例。部分掩码区域可能对水印移除并无贡献，因此对其进行修改是多余的。此外，MAE 无法提供最先进的图像再生成质量 [35]。为此，我们需要更精密的掩码模型来定位关键区域，以及更精细的生成器

**算法1：** HIDE&SEEK Random

/* 训练 */
**数据：** 训练数据集 $\{X^1, ..., X^n\}$
**结果：** 训练完成的编码器 $E$, 解码器 $D$
**当** $L$ 未收敛 **执行**
  $X \leftarrow$ 采样($\{X^1, ..., X^n\}$);
  $M \leftarrow$创建_随机_掩码();
  $L \leftarrow \|X - D(E(M \odot X))\|^2$; /* 损失 */
  优化器步骤;
**end**
**返回** $E$, $D$;

/* 评估 */
**数据：** 带水印图像 $I$，掩码比例 $\beta$
**结果：** 净化图像 $\hat{I}$
$M \leftarrow$创建_掩码_通过_策略($\beta$);
$\hat{I} \leftarrow D(E(M \odot I))$;
**return** $\hat{I}$;

来恢复这些区域。因此，我们提出 HIDE&SEEK 的进阶版本，即 HIDE&SEEK Plus （ HS+ ）。

### 4.3 HIDE&SEEK Plus

HS+ 的核心思想是定位在最轻微修改下对带水印图像 $I$影响最为显著的像素，因此这些像素被命名为“脆弱像素”。随后，这些像素按照排序后的顺序逐一被掩码并重建，其中最脆弱的像素最后重建，以便产生更大的偏差。

**训练**。为实现这一点，对手训练一个掩码模型$H$: $\mathbb{R}^{C\times w\times h} \mapsto R^{w\times h}$，该模型根据无水印图像 $X \in \{X^1, \ldots\ldots X^n\}$ 预测像素的掩码概率，如算法2和图4所示。给定逻辑值 $H(X)$, 对手随后需要衡量像素的脆弱程度，以便得到排序。由于语义水印和非语义水印分别存在于 $I$的高频分量和低频分量中，所选像素必须对频率以及 $I$上的语义域都产生最大影响。我们首先通过如下定义的可微操作创建一个软掩码:

$$M = \frac{1}{1 + \exp[-\gamma(H(X))]}, \tag{4.2}$$

其中 $\gamma$是温度参数；exp[ ] 表示逐元素指数运算。例如，在 exp[$X$]中，每个元素 $X_{ij} \in X$将被用作指数，以生成一个与 $X$ 维度相同的矩阵。exp[$X$] 中的每个元素将变为 $e^{X_{ij}}$。由此， $H$（$X$）被缩放，使正值映射到接近1，而负值映射到接近0。因此 $M$被称为软掩码，因为这种缩放并不能保证精确映射到0或1。

对手使用软掩码 $M$对 $X$进行最小限度的修改。像素的最小修改幅度应为 $\pm 1$，因为彩色像素是一个整数 $p \in [0, 255]^3$。因此，对手随机生成扰动 $\mathbb{1} \in \{-1, 1\}^{C\times w\times h}$。 $X$通过 $\mathbb{1}$ 和 $M$进行扰动，得到扰动图像 $\tilde{X}$，记作 $\tilde{X} = M \odot \mathbb{1} + X$。

$\tilde{X}$与 $X$之间的频率差异通过如下定义的频率损失进行度量:

$$\begin{aligned} L_{\text{frq}}(X, \tilde{X}) =& \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} \omega(u, v) \cdot \\ & |F_X(u, v) - F_{\tilde{X}}(u, v)|^2, \end{aligned} \tag{4.3}$$

其中 $\omega$ 是频谱权重矩阵，定义为

$$\omega(u, v) = |F_X(u, v) - F_{\tilde{X}}(u, v)|^{\alpha}. \tag{4.4}$$

此处， $\alpha$是缩放因子，默认设置为1。 $F_X$ （$u, v$）表示二维傅里叶变换在频谱上坐标（$u, v$）处的复频率值，定义为

$$\begin{aligned} F_X(u, v) =& \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} X_{x,y} \exp(-i2\pi(\frac{ux}{w} + \frac{vy}{h})) \\ =& a + bi, \end{aligned} \tag{4.5}$$

其中 $a$和 $bi$ 分别表示复数值 $F_X$ （$u, v$）的实部和虚部。

$X$与 $\tilde{X}$之间的语义差异的衡量定义为

$$L_{\text{smt}}(X, \tilde{X}) = \|\text{CLIP}(X) - \text{CLIP}(\tilde{X})\|^2, \tag{4.6}$$

其中，CLIP [21]表示对比语言-图像预训练模型，该模型将不同模态（文本和图像）的输入映射到同一潜在空间。因此，通过测量潜在空间中 $X$和 $X$的特征之间的欧氏距离，我们可以衡量语义差异。

对手的目标是修改尽可能少的像素，在带水印图像与净化图像之间的频域和语义域中制造最大差异。在训练过程中，我们需要将这个问题转化为损失最小化问题。因此，我们可以改写目标：对手修改尽可能多的像素，在带水印图像与净化图像之间的频域和语义域中制造最小差异。由此可推导出以下损失函数:

$$\begin{aligned} L_{\text{HIDE}} =& \lambda_1(WH - \|M\|^2) + \lambda_2 L_{\text{frq}}(X, \tilde{X}) \\ & + \lambda_3 L_{\text{smt}}(X, \tilde{X}), \end{aligned} \tag{4.7}$$

其中， $\lambda$s 表示权重参数；第一项衡量图像区域与掩码区域之间的距离。通过最小化 $L_{\text{HIDE}}$， $H$会为更脆弱的像素赋予更低的分数。

接下来，在 SEEK 中，对手逐像素重建掩码图像，以移除嵌入式水印。现在， $H(X)$ 提供的脆弱性排名用于指导这一重建过程，得分较低的像素会在较后阶段重建。原因在于，我们不能假设现实世界中任何现有的生成器都是能够提供完美预测的预言机模型。因此，在自回归过程中，由于生成模型的内在特性——前一次预测会成为下一次输入——它会不断累积误差。因此，预测偏差在最后一次预测时达到峰值。由此，在最后一步重建最脆弱的像素会导致更大的预测偏差，从而扩大对该像素的修改。

为实现像素生成器，我们采用最先进的分形生成网络（FGN）[15] 作为模型架构。

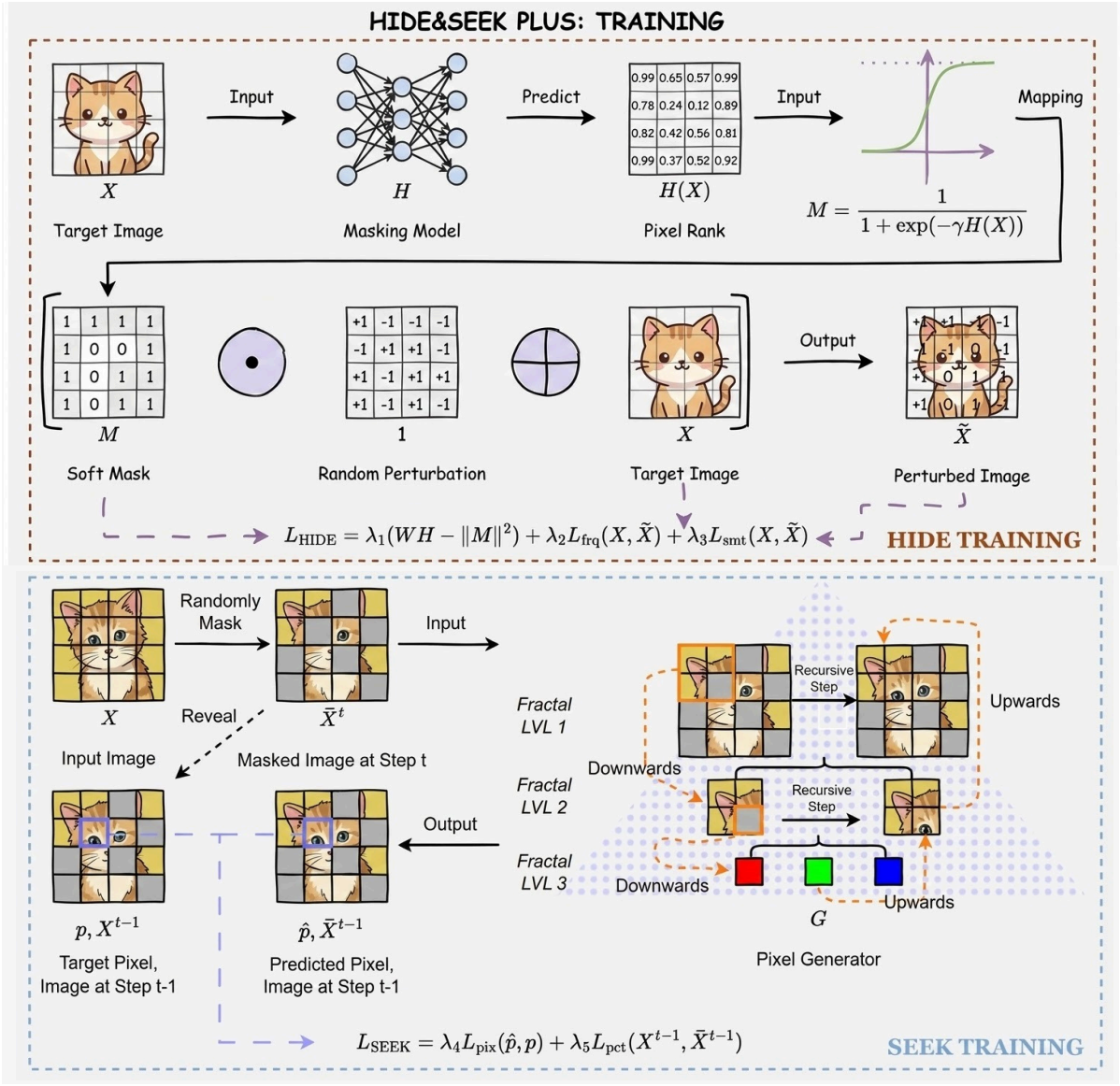



**图4：HIDE&SEEK Plus训练流程。在阶段*HIDE*中，掩码模型接收目标图像，并预测图像中所有像素的排序。随后将该排序映射为用于随机扰动的软掩码。利用软掩码、扰动图像和目标图像计算HIDE损失。在阶段*SEEK*中，将随机掩码图像输入像素生成器，以预测下一像素。随后利用预测像素和目标像素计算SEEK损失。**

FGN 基于“分而治之”策略设计。这意味着 $G$： $\mathbb{R}^{C\times w\times h} \mapsto \mathbb{R}^{C\times S}$ 是一种由多个分形层级组成的自回归模型， $G$会递归处理掩码图像。此处， $S$表示像素取值范围，默认为 256。在每个分形层级，掩码图像都会被划分为多个图像块。随后，每个图像块都会传递到下一级进行进一步处理。我们将这一递归过程表示为 $G(X) = G^n \ldots G^2\ G^1\ X$ $G^j$ ( ( ) )，其中 是 $G$中第 $j$个分形层的块。在逐像素重建的每次迭代中， $G$给出仅针对一个像素的逻辑值。也就是说，对于大小为 $|M|$的掩码 $M$，总共需要经过$|M|$次迭代才能完全重建掩码图像。令$\bar{X}^t$表示步骤 $t$中的重建图像，则有 $\bar{X}^{t-1} = \text{enc}(\bar{X}^t,\ G(\bar{X}^t))$. enc(·，·)是一个编码函数，它接收 $\bar{X}^t$以及预测像素 $G(\bar{X}^t)$的逻辑值，并输出 $\bar{X}^{t-1}$，其中下一像素得到重建。

为了训练 $G$，对手会随机创建一个掩码 $M \in \{0,\ 1\}^{w\times h}$，以获得掩码图像 $\bar{X} = M \odot X$。然后，推导出下一像素 $\hat{}p = G(\bar{X})$ 的 logit。对手使用定义如下的像素损失

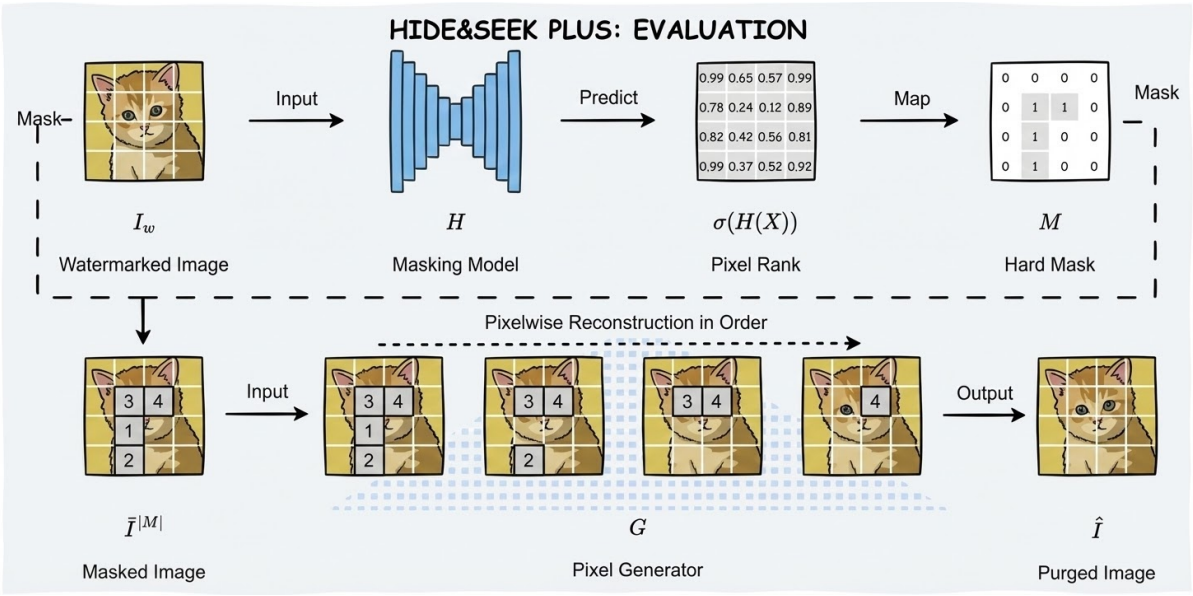



**图5：HIDE&SEEK Plus 评估流程。带水印图像被输入掩码模型，以生成像素排序。概率小于0.5 的像素会直接被掩码以进行重建。按照像素排序给出的顺序，逐步重建被掩码的像素，其中最脆弱的像素最后重建。**

来衡量预测像素与目标像素之间的距离：

$$L_{\text{pix}}(\hat{p}, p) = -\sum_{c=0}^{C}\sum_{b=0}^{B} \log \frac{e^{\hat{p}_{c,b}} \cdot \text{OneHot}(p)_{c,b}}{\sum_{b=0}^{B} e^{\hat{p}_{c,b}}}, \tag{4.8}$$

其中，$\hat{}p \in \mathbb{R}^{c\times b}$ 表示由 $G$预测的逻辑值；$\hat{}p_{c,b}$ 表示该像素在通道 $c$中取值为 $b$ 的概率； $p$ 表示与 $\tilde{X}$中$\hat{}p$ 位置相同的原始像素；OneHot(·)表示独热编码函数。这使 FGM 有别于其他生成器，因为图像不再被视为一个整体，而是像自然语言处理中的文本一样被视为词元序列。通过最小化这一损失，对手训练 $G$，使其能够根据可见像素正确预测下一像素。

对手还希望保持最佳感知质量。因此，重建图像 $\tilde{X}^{t-1}$ 与对应目标图像 $X^{t-1}$ 在第 $t-1$ 步的感知距离由如下定义的感知损失来衡量：

$$L_{\text{pct}}(X^{t-1}, \tilde{X}^{t-1}) = \|\text{Alex}(X^{t-1}) - \text{Alex}(\tilde{X}^{t-1})\|^2, \tag{4.9}$$

其中，Alex(·) 表示预训练的 AlexNet，用于提取输入图像的特征； $X^{t-1}$ 表示显示了对应像素的原始图像。

因此，优化 $G$的完整损失函数可定义为：

$$L_{\text{SEEK}} = \lambda_4 L_{\text{pix}}(\hat{p}, p) + \lambda_5 L_{\text{pct}}(X^{t-1}, \tilde{X}^{t-1}) \tag{4.10}$$

对于每张输入图像 $X$， $G$ 在训练阶段只需执行一步来计算损失。由于掩码是均匀随机生成的，训练能够覆盖掩码像素的所有预测状态。

**评估**。训练好 $H$和 $G$后，对手便可针对任意带水印图像 $I$发起水印移除攻击。如算法2的评估 部分所示，给定$I$，对手使用 $I$和 $H$创建一个硬掩码，记为 $M = \text{get_hard_mask}(\sigma(H(I)))$。其中， $\sigma(\cdot)$ 是 Sigmoid函数；get_hard_mask(·) 是一个函数，它将输入中大于 0.5 的值映射为 1，将小于 0.5 的值映射为 0。接下来，对手使用 $M$对 $I$进行掩码处理，得到 $\bar{I}^{|M|} = M \odot I$。这里，我们用 $|M|$表示递归重建过程的总步数。令 $t$为当前步的索引。在每次迭代中， $G$接收第 $t$ 步的掩码图像，以预测下一个词元 $\hat{}p = G(\bar{I}^t)$。然后，利用编码函数可推导出下一步的掩码图像 $\bar{I}^{t-1}$，记为 $\bar{I}^{t-1} = \text{enc}(\bar{I}^t, \hat{}p)$。最后，重建最后一个掩码像素后，步索引 $t \leftarrow 0$。最终，对手可以得到净化图像 $\hat{I} = \bar{I}^0$。

## 5 理论分析

本节将从理论上讨论掩码像素的重建顺序如何影响攻击性能。首先，我们将介绍一些符号说明和假设。

**掩码集**。我们将掩码像素的颜色向量集合记为 $M$，其基数记为 $M$，为 $|M|$。然后，我们需要$|M|$ 步，逐步使用像素生成器恢复图像。

**颜色向量**。我们将第 $i$ 步中所选预测像素的原始颜色向量记为 $\boldsymbol{x}^i = (r_i, g_i, b_i)$，并将第 $i$ 步中的预测值记为 $\boldsymbol{x}_i = (\hat{}r_i, \hat{}g_i, \hat{}b_i)$。

**元素与顺序**。顺序（$\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots \boldsymbol{x}^{|M|}$）由掩码模型决定，这意味着 $\boldsymbol{x}^i$ 的扰动可能

**算法2：** HIDE&SEEK Plus

/* 训练 */
**数据：** 训练数据集 $\{X^1, ..., X^n\}$
**结果：** 已训练的掩码器 $H$ 和生成器 $G$
**当** $L_{HIDE}$ 未收敛 **时**
  /* 训练 $H$ */
  $X \leftarrow$ 样本($\{X^1, \ldots\ldots, X^n\}$);
  $M \leftarrow \frac{1}{1+\exp[-\gamma H(X)]}$; /* 预测软掩码 */
  $\tilde{X} \leftarrow M \odot \mathbb{1} + X$; /* 获取扰动图像 */
  $L_{\text{HIDE}} \leftarrow \lambda_1 \|M\|^2 + \lambda_2 L_{\text{frq}}(X, \tilde{X}) + \lambda_3 L_{\text{smt}}(X, \tilde{X})$; /* 计算损失 */
  优化器步进;
**end**
**当** $L_{SEEK}$ 未收敛 **执行**
  /* 训练 $G$ */
  $X \leftarrow$ 样本($\{X^1, \ldots\ldots, X^n\}$);
  $M \leftarrow$创建_随机_掩码();
  $\bar{X}^{|M|} \leftarrow M \odot X$; /* 掩码图像 */
  $\hat{p} \leftarrow G(\bar{X}^{|M|})$; /* 预测下一个词元 */
  $\bar{X}^{|M|-1} \leftarrow \text{enc}(\bar{X}^{|M|}, \hat{p})$;
  $L_{\text{SEEK}} \leftarrow \lambda_4 L_{\text{pix}}(\hat{p}, p) + \lambda_5 L_{\text{pct}}(X^{|M|-1}, \bar{X}^{|M|-1})$; /* 计算损失 */
  优化器步骤;
**end**
**返回** $H$, $G$;
/* 评估 */
**数据：** 带水印图像 $I$
**结果：** 净化图像 $\hat{I}$
$M \leftarrow$ get_hard_掩码($\sigma(H(I))$); /* 硬掩码 */
$I^{|M|} \leftarrow M \odot I$; /* 掩码图像 */
**对** $t \in [|M|, 0]$ **执行**
  $\hat{p} \leftarrow G(I^t)$; /* 预测下一个词元 */
  $I^{t-1} \leftarrow \text{enc}(I^t, \hat{p})$;
**end**
$\hat{I} \leftarrow I^0$;
**返回** $\hat{I}$;

造成的差异少于对 $\boldsymbol{x}^{i+1}$（ $i =$ 1 … „ $|M|-1$）施加相同扰动所造成的差异。事实上，我们使用掩码模型为每个像素给出分数 $\alpha_i > 0$，并在将颜色向量从 $\boldsymbol{x}^i$ 更改为 $\boldsymbol{x}_i$ 时，通过 $\alpha_i \|\boldsymbol{x}_i - \boldsymbol{x}^i\|$ 估计这些差异。

**累积预测误差与差异**。累积预测误差定义为

$$APE = \left( \sum_{i=1}^{|M|} \|\boldsymbol{x}_i - \boldsymbol{x}^i\|^2 \right)^{\frac{1}{2}}, \tag{5.1}$$

并且累积预测差异由下式给出

$$APD = \sum_{i=1}^{|M|} \alpha_i \|\boldsymbol{x}_i - \boldsymbol{x}^i\|. \tag{5.2}$$

**假设**。我们假设该预测器是预言机预测器，使得第 $i$步的误差与预测顺序无关，并将第 $i$步的预测误差记为$y_i > 0$。例如，若我们通过$\{1, 2, \ldots, |M|\}$的置换 $\pi$恢复图像，则有$\|x_{\pi(i)} - x^{\pi}{}_{(i)} \| = y_i (i = 1, 2, \ldots, |M|)$。于是， $APE = \left( \sum_{i=1}^{|M|} y_i^2 \right)^{\frac{1}{2}}$与预测顺序无关。此外，我们假设 $\bar{y_i} \leq y_{i+1} (i = 1, \ldots, |M| - 1)$。

引理 5.1（重排不等式， [4]）。设 $a_1 \leq a_2 \leq \cdots \leq a_n$和 $b_1 \leq b_2 \leq \cdots \leq b_n$为两个实数序列。那么，对于任意置换 $\pi${1、2、. . .、 $n$}，有

$$\sum_{i=1}^{n} a_i b_{n-i+1} \leq \sum_{i=1}^{n} a_i b_{\pi(i)} \leq \sum_{i=1}^{n} a_i b_i. \tag{5.3}$$

定理 5.1。若我们按升序 $O_1 = (\boldsymbol{x}^1、 \boldsymbol{x}^2、 \cdots \boldsymbol{x}^{|M|})$和随机顺序 $O_2 = (\boldsymbol{x}^{\pi(1)} \boldsymbol{x}^{\pi(2)}、 \cdots \boldsymbol{x}^{\pi(|M|)})$恢复图像，则

$$APD(O_2) = \sum_{i=1}^{|M|} \alpha_i y_{\pi(i)} \leq APD(O_1) = \sum_{i=1}^{|M|} \alpha_i y_i. \tag{5.4}$$

证明。根据我们的假设，有 $\alpha_i \leq \alpha_{i+1}$ $y_i \leq y_{i+1}$ $i = 1 \ldots |M| - 1$ 5 4 ，(„)。因此，根据引理 5.1，等式 . 成立。□

定理 5.1 论证了我们在 HS+ 中的攻击策略。从理论上讲，给定一个训练良好且能提供正确重建顺序的掩码模型，像素生成器按照该顺序进行重建，即可获得最大的累积预测差异。

# 6 评估

## 6.1 实验设置

图像处理、数据集、实现，以及攻击和防御的基线方法的详细信息列于附录 A。

**评估指标**。以下指标用于评估攻击性能。

- **峰值信噪比** (PSNR)：峰值信噪比用于通过比较图像与其处理后的版本来衡量图像质量。
- **结构相似性指数**（SSIM）：SSIM 用于通过比较图像与其处理后的版本来衡量图像的感知质量。
- **学习感知图像块相似度**（LPIPS）：LPIPS 利用深度学习，使衡量两幅图像之间的相似度与人类视觉感知相契合。
- **比特准确率**（BA）：给定原始水印和以二进制形式提取的水印，BA 是正确恢复的比特数与水印长度之比。
- **逆距离** (ID)：ID 仅用于评估 TRW 提取出的水印。只有当提取序列与水印之间的 $l_1$ 距离低于预定义阈值时，才认定检测到水印。我们遵循先前的研究 [12]，将阈值设为$_1$ $_{71}$ ，以提高攻击任务的难度。 —

- **检测率** (DR)：检测到的水印数量与尝试总次数之比。

## 6.2 有效性和保真度

**防御水印的评估**。在评估攻击之前，我们先评估防御水印的鲁棒性。防御水印的鲁棒性通过带水印图像遭受攻击时的抵抗程度来衡量。我们使用图像处理来测试水印方法，因为这类处理易于实施且高效。此次评估中，我们重点关注水印的有效性和鲁棒性，因为先前的研究已证明其隐蔽性。

如图8所示，我们使用五种不同参数的图像处理来测试防御水印。此处不评估视觉质量指标，因为防御水印的鲁棒性才是主要关注点。此外，这些处理不会显著影响图像的视觉质量 [12, 18]。我们未纳入图像旋转，因为在验证过程之前，可以轻松检测并纠正图像旋转 [37]。

实验结果表明，在未对带水印图像进行任何图像处理时，各防御水印的检测率均达到 1.00。图像裁剪可以有效去除 GANF 水印，并降低 HiDDeN 和 PTW 的检测率。然而，其余水印对裁剪表现出很强的鲁棒性，这表明它们在一定程度上不受空间处理的影响。模糊、量化和 JPEG 压缩等频谱处理会显著影响带水印图像的高频分量。因此，非语义水印（从 AF 到 StableSig.）的检测率会不同程度地下降。相比之下，语义水印（StegaStamp 和 TRW）表现出卓越性能，在大多数情况下检测率仍保持在 1.00，证明了它们对图像处理的鲁棒性。

**水印移除攻击的评估**。现在，我们通过对防御水印生成的带水印图像实施攻击，将 HIDE&SEEK 与最先进攻击进行比较。如图6所示，扩散攻击和 VAE攻击在去除非语义水印方面效果显著。然而，由于位于图像低频分量中的水印具有很强的鲁棒性，这些攻击无法去除语义水印。这类攻击的关键思路是引入足够的噪声，以干扰水印信号。随后，利用强大的生成器恢复净化图像。这解释了为何扩散攻击在大多数情况下比 VAE攻击更能保持视觉质量，因为扩散模型被认为比基础自编码器更为先进。我们注意到，Un-Marker、HSN 和 HS+ 能够有效去除语义和非语义水印。这是因为这些攻击旨在使带水印图像的低频和高频域均产生频率差异。图9展示了可视化的攻击结果。

在大多数情况下，HSN 的攻击性能更好，其检测率在所有攻击中最低。面对 GANF 时，与 HSN 相比，VAE攻击的检测率更低。这是因为 GANF 对空间操纵很敏感

**表2：对一百张机器生成图像进行水印移除攻击的平均消耗**

| 攻击 | HSN | HS+ | UnMarker | DiffAtk | VAE攻击 |
|---|---|---|---|---|---|
| 时间（秒） | 871 | 1302 | 2553 | 1454 | 839 |
| 费用（美元） | 0.99 | 1.48 | 2.91 | 1.66 | 0.96 |

例如裁剪，如图8所示。两个版本的 HIDE&SEEK 都不包含裁剪，也不会进行全局图像扰动，以保持较高的图像质量。因此，与其他方法相比，它们并非总是表现最佳。HS+ 成功移除了水印，同时保持了较高的视觉质量。在所有测试中，HS+ 的峰值信噪比均高于 30，结构相似性均高于 0.85。尽管 HS+ 的攻击性能并非总是最佳，但面对 HiDDeN 和 StableSignature 时效果显著，检测率分别降至 0.0 和 0.03。研究发现，面对先进攻击时，语义水印表现出令人瞩目的鲁棒性。与 UnMarker 相比，HS+ 的攻击过程不包含裁剪，因此预期能获得更高的视觉质量。此外，HS+ 在大多数情况下优于其他攻击。这表明，自回归模型在预测过程中引入的扰动足以清除水印。

**效率评估**。我们进一步从效率方面评估 HS 与 SOTA攻击的表现。结果如表2所示。在相同设置下，我们随机选取 1,00 张带水印图像进行攻击。我们重复此操作 10 次，计算平均时间成本。VAE攻击的时间成本最低，因为基础自编码器的结构相对简单。值得注意的是，HSN 达到与 VAE攻击相近的结果，但如图6所示，其攻击性能更好。相比之下，Unmarker 的时间成本最高，因为每张带水印图像的扰动都需要单独优化。为方便读者，我们根据亚马逊云服务上租用一块 A100 GPU 的价格，将时间成本换算为费用，该价格为每小时 4.1 美元。总之，HSN 和 HS+ 都能实现高效的水印移除攻击。

## 6.3 消融研究

**掩码策略和比例的影响**。我们首先研究掩码策略和比例如何影响 HSN 的攻击性能。我们将三种掩码策略与从 {0.60,0.65, 0.70, 0.75,0.80} 中选取的掩码比例组合起来，评估它们对抗 StegaStamp 时对 HSN 的影响。选择 StegaStamp 是因为其鲁棒性使 HSN 能取得从低到高的各种结果，而非仅有较低的检测率。如图 7 所示，分散掩蔽能最好地保持图像质量，但会导致较高的检测率。在 $\beta = 0.60$ 下，分散式 HSN 可达到 31.18 的峰值信噪比和 0.88 的结构相似性，而水印检测率为 1.0。即使提高 $\beta$，在相同设置下，分散式 HSN 去除水印的效果仍不如另外两种策略。这表明分散掩蔽倾向于保留更多原始频率信息。

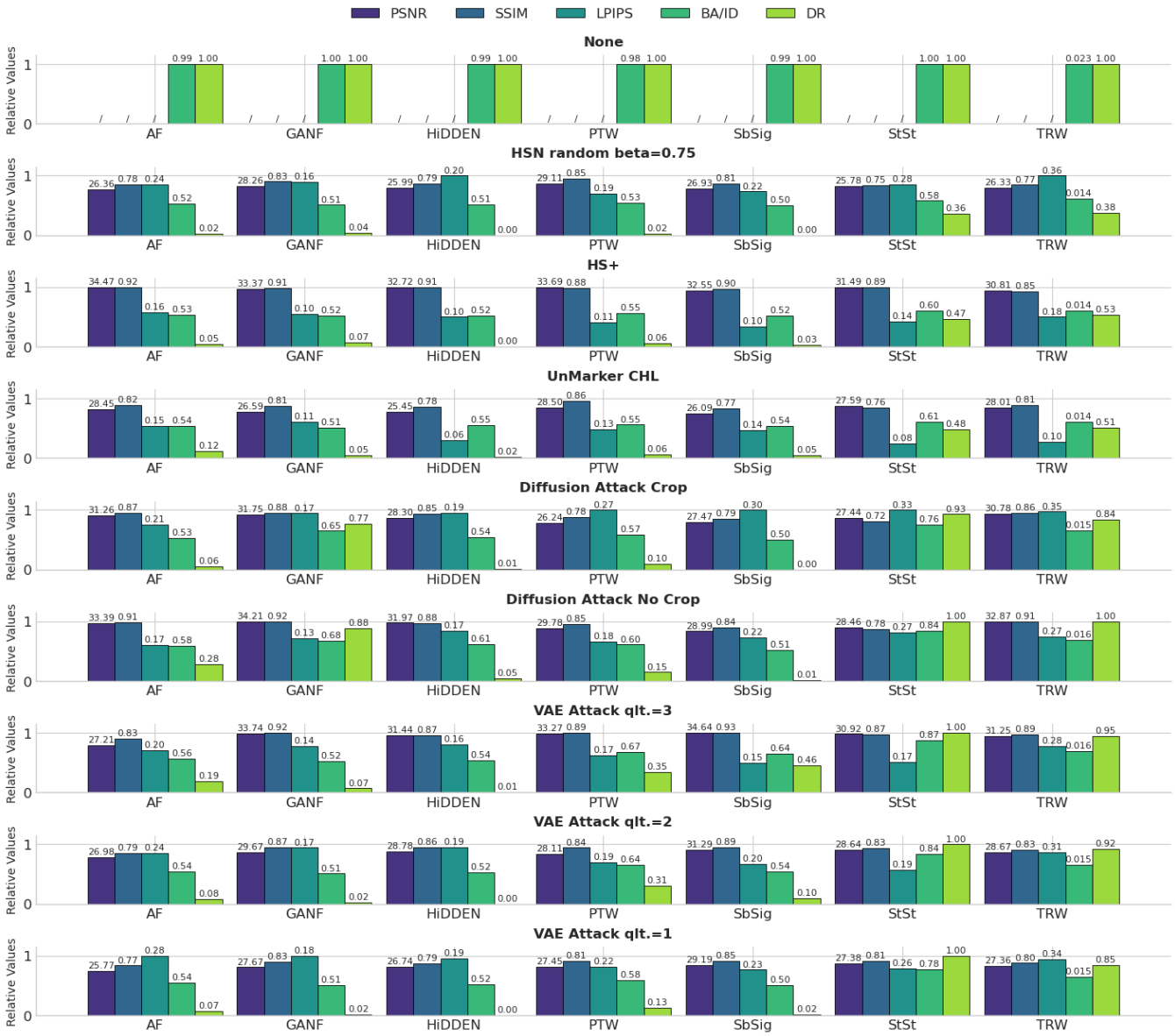



**图6：对防御水印的攻击评估。每个子图展示了一种防御水印应对图像处理的结果。StableSignature 和 StegaStamp 分别缩写为 SbSig 和 StSt。TRW 的第四根柱表示逆序距离，其余第四根柱表示比特准确率。柱子的高度表示某项指标下的归一化值，实际数值标注在柱子上方。**

此外，与随机掩蔽相比，当 $\beta$ 较低时，连续掩蔽会导致较差的图像质量；但随着 $\beta$ 增大，两者的图像质量相近。这是因为较大的 $\beta$ 会覆盖带水印图像的更大区域，使两个掩码的重叠部分增多。我们还注意到，当 $\beta = 0.60$ 时，随机掩蔽和连续掩蔽都已取得相当可观的攻击性能，检测率分别降至 0.58 和 0.55。给定较小的 $\beta$ 时，连续掩蔽的检测率往往低于随机掩蔽，因为每个被选中的

图像块彼此相邻。这会在频域中引发更强烈的扰动。

**损失项的影响**。接下来，我们评估 HS+ 训练过程中的损失项如何影响攻击性能。结果列于表

3, $WH - $中，其 $\|M\|^2$ $L_{\text{pix}}$ $WH - \|M\|^2$ 中排除了 ( ) 和 的结果。移除 ( ) 后，掩码的图像块数量自然会降至 0，因此我们省略了这项测试。 $L_{\text{pix}}$是训练像素生成器的基本损失项，因此无法省略。这里再次使用 StegaStamp 作为

**表3：损失项的影响**

| $L_{\text{frq}}$ | $L_{\text{smt}}$ | $L_{\text{pct}}$ | 峰值信噪比↑ | 结构相似性↑ | 学习感知图像块相似度↓ | 比特准确率↓ | 检测↓ |
|---|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | 31.49 | 0.89 | 0.14 | 0.63 | 0.52 |
| × | ✓ | ✓ | 30.65 | 0.86 | 0.18 | 0.61 | 0.45 |
| ✓ | × | ✓ | 30.89 | 0.87 | 0.13 | 0.67 | 0.66 |
| ✓ | ✓ | × | 31.15 | 0.88 | 0.16 | 0.63 | 0.53 |

**表4：重建顺序的影响**

| 顺序 | 峰值信噪比↑ | 结构相似性↑ | 学习感知图像块相似度↓ | 比特准确率↓ | 检测↓ |
|---|---|---|---|---|---|
| 原始 | 31.49 | 0.89 | 0.14 | 0.63 | 0.52 |
| 逆序 | 32.18 | 0.90 | 0.12 | 0.65 | 0.58 |
| 随机 | 31.91 | 0.89 | 0.14 | 0.63 | 0.54 |

防御水印，因为它具有出色的鲁棒性。我们通过在训练过程中将某个损失项的 $\lambda$ 设为 0，来消除该损失项的影响。

$L_{\text{frq}}$ 有助于定位在图像频域中影响更大的像素。当绕过 $L_{\text{frq}}$ 时，$L_{\text{smt}}$ 的权重增加，导致掩码模型专注于掩蔽对图像语义域影响更大的像素。因此，净化图像的视觉质量下降，而攻击性能提高。然而，当绕过 $L_{\text{smt}}$ 时，尽管净化图像的视觉质量下降，我们并未观察到攻击性能有所提升。由于 StegaStamp 是一种语义水印，其水印信号存在于低频分量中。移除语义损失可能导致掩码模型无法掩蔽显著影响带水印图像语义域的像素。这是因为与能量集中的低频分量相比，图像的高频分量更容易修改。最后，当绕过 $L_{\text{pct}}$ 时，我们观察到净化图像的视觉质量略有下降，但攻击性能保持不变。这表明 $L_{\text{pct}}$ 有助于提升净化图像的视觉质量。由于该损失项未参与掩码模型的训练，因此几乎不会影响攻击性能。因此，我们认为掩码模型在攻击过程中发挥着更重要的作用。

**重建顺序的影响**。我们进一步研究了重建顺序对攻击性能的影响。我们的初始设想是按照一定顺序恢复像素：先重建最不易受攻击的像素，最后重建最易受攻击的像素。我们以 StegaStamp 作为防御水印，进行了两项并行实验，分别按照逆序和随机顺序重建像素。结果如表4所示，原始顺序取得了最佳攻击性能，比特准确率为 0.63，检测率为 0.52，但视觉质量最差。逆序重建取得了最佳视觉质量，但攻击性能最差；随机顺序的结果则与原始顺序略有差异。这表明，像素生成器的重建顺序会影响净化图像的视觉质量。此外，这也体现了水印移除攻击有效性与保真度之间的权衡。

## 7 讨论

**可行性**。HSN 和 HS+ 均不需要访问水印检测器或获取其反馈。生成器和掩码模型可以在配备 32GB 显存的 NVIDIA RTX5090 上训练，个人也能负担得起。ImageNet 等训练数据集可公开获取。因此，个体攻击者能够实施此类攻击。

与重建攻击和扰动攻击相比，HIDE&SEEK 不会对带水印图像进行全局扰动。全面评估表明，部分掩蔽和像素重建造成的差异足以去除水印。此外，与 UnMarker 等扰动攻击不同，HIDE&SEEK 不会执行裁剪等会显著降低净化图像视觉质量的图像处理。另外，掩码模型和生成器一经训练，便可用于一般攻击目的。相反，针对每一张带水印图像，UnMarker 都会专门优化两种扰动，以扭曲带水印图像的高频和低频域。随着带水印图像数量增加，这会带来额外的时间成本。

**未来工作**。所有用于机器生成图像的防御水印都面临一个普遍问题：对手始终可以在水印检测率与带水印图像的视觉质量之间权衡。换言之，降低带水印图像的质量，最终会得到一张无法再检测出嵌入式水印的净化图像。先进的水印去除攻击应在提高净化图像质量的同时，确保水印检测率较低。与此同时，我们的研究表明，当前的防御水印十分脆弱，可以在视觉质量损失可接受的情况下被去除。我们需要设计更稳健的水印方案，并将其他类型的机器生成图像检测方法纳入考虑。

就泛化能力而言，仅使用 ImageNet 等真实世界图像数据集进行训练时，HS 在真实感图像上表现良好。但 HS 对 AnimeDL-2M 等由非写实动漫图像构成的跨领域数据集，泛化能力较差。不过，将这类非写实数据集纳入训练数据集，有望进一步提升 HS 的泛化能力。

## 8 结论

在本研究中，我们设计了一种新颖的图像水印移除攻击，称为 HIDE&SEEK，它有两个版本，分别命名为 HIDE&SEEK Naive 和 HIDE&SEEK Plus。HSN 使用掩码自编码器对带水印图像进行掩码处理，然后恢复被掩码的图像块，以生成净化图像。HS+ 采用掩码模型，巧妙地选择脆弱像素进行掩码，并使用像素生成器递归重建被掩码的像素。最后一步重建最脆弱的像素，以产生最大的差异。通过全面评估，我们证明 HSN 和 HS+ 均能有效对抗当前的防御水印。就攻击性能而言，HSN 优于最先进攻击；而面对每种防御水印时，HS+ 都能保持较高的视觉质量，同时将水印检测率降至相当低的水平。本研究也进一步表明，有必要在这一领域开展研究，以防止生成式人工智能遭到滥用。

## 伦理考量

开展本研究时，我们严格遵循“无伤害”原则。我们设计的方法旨在纯理论且隔离的环境中评估水印算法的鲁棒性，确保不会对任何个人、组织或运行系统造成负面影响。

**与运营系统隔离**。本文所述的所有实验均在完全离线的本地环境中进行。我们使用了水印协议的开源实现，并将其部署在自有硬件上。我们的攻击方法从未与在线商业 API 或外部网络服务交互、向其发送查询或对其进行压力测试。因此，这项研究没有导致任何模型提供商或平台的服务中断、延迟或财务成本。

**使用公开且非敏感的数据**。评估完全采用标准的公开学术数据集进行。我们没有收集私人用户的数据，也没有针对特定内容创作者。我们仅依赖成熟的基准数据集，确保研究过程中未处理任何个人数据，也未侵犯任何个人的隐私或知识产权。

**安全的安全性评估**。我们的工作将水印信号视为待分析的数学产物，而非攻击承载它的基础设施。通过将我们的“红队测试”工作限制在沙箱环境中，我们证明可以在不危及更广泛的数字生态系统或其利益相关者的情况下识别并记录安全漏洞。

## 开放科学

为促进透明度与可复现性，我们将发布工件，其中包括本文所采用的所有实证研究的全部代码。我们的代码将在论文被接收后发布。由于我们所使用的所有数据集均可公开获取，因此它们均包含在该工件中。

## 生成式人工智能的使用

在本稿件的准备过程中，我们严格仅将生成式人工智能工具用于语言润色。具体而言，这些工具被用于检查语法错误，并改善文本的流畅性与可读性。

我们确认：

- **未使用内容生成：**核心概念、实验设计、数据分析和科学结论均由作者独立完成。
- **未生成合成文本：**文本中没有任何实质性内容由人工智能生成。相关工具仅用作文字编辑辅助工具。
- **人工监督：**人工智能提出的所有建议均由作者人工审阅并核实；作者对最终出版物的准确性和完整性承担全部责任。

## 参考文献

[1] Bang An、Mucong Ding、Tahseen Rabbani、Aakriti Agrawal、Yuancheng Xu、Chenghao Deng、SichengZhu、AbdirisakMohamed、Yuxin Wen、Tom Goldstein，等人。2024。WAVES：图像水印鲁棒性基准测试。载于《第41届国际机器学习会议论文集》。1456–1492。[2] Yihan Cao、Siyu Li、Yixin Liu、Zhiling Yan、Yutong Dai、Philip Yu和Lichao Sun。2025。人工智能生成内容（AIGC）综述。《计算机调查》57卷5期（2025），1–38。[3] Huajie Chen、Tianqing Zhu、Chi Liu、Shui Yu和Wanlei Zhou。2024。高频至关重要：图像处理模型水印的攻击与防御。《IEEE服务计算汇刊》17卷4期（2024），1565–1579。[4] Peter W. Day。1972。《重排不等式》。《加拿大数学杂志》24卷5期（1972），930–943。doi:10.4153/CJM-1972-093-x[5] Pierre Fernandez、Guillaume Couairon、Hervé Jégou、Matthijs Douze和Teddy Furon。2023。稳定签名：将水印植入潜在扩散模型。载于《IEEE/CVF国际计算机视觉会议论文集》。22466–22477。[6] Joel Frank、Thorsten Eisenhofer、Lea Schönherr、Asja Fischer、Dorothea Kolossa和Thorsten Holz。2020。利用频率分析进行深度伪造图像识别。载于《国际机器学习会议》。PMLR，3247–3258。[7] Jamie Hayes和George Danezis。2018。利用生成模型学习通用对抗扰动。载于《2018年IEEE安全与隐私研讨会（SPW）》。IEEE，43–49。[8] Kaiming He、Xinlei Chen、Saining Xie、Yanghao Li、Piotr Dollár和Ross Girshick。2022。掩码自编码器是可扩展的视觉学习器。载于《IEEE/CVF计算机视觉与模式识别会议论文集》。16000–16009。[9] Zhengyuan Jiang、Jinghuai Zhang和Neil Zhenqiang Gong。2023。规避基于水印的人工智能生成内容检测。载于《2023年ACM SIGSAC计算机与通信安全会议论文集》。1168–1181。[10] Tushar M Jois、Gabrielle Beck和Gabriel Kaptchuk。2024。Pulsar：面向扩散模型的安全隐写术。载于《2024年ACM SIGSAC计算机与通信安全会议论文集》。4703–4717。[11] Andre Kassis和Urs Hengartner。2023。攻破安全关键型语音认证。载于《2023年IEEE安全与隐私研讨会（SP）》。IEEE，951–968。[12] Andre Kassis和Urs Hengartner。2025。UnMarker：针对防御性图像水印的通用攻击。载于《2025年IEEE安全与隐私研讨会（SP）》，第2卷。8。[13] Minjong Lee和Dongwoo Kim。2023。对基于扩散的对抗净化进行鲁棒评估。载于《IEEE/CVF国际计算机视觉会议论文集》。134–144。[14] Lingzhi Li、Jianmin Bao、Ting Zhang、Hao Yang、Dong Chen、Fang Wen和Baining Guo。2020。用于更通用人脸伪造检测的人脸X射线。载于《IEEE/CVF计算机视觉与模式识别会议论文集》。5001–5010。[15] Tianhong Li、Qinyi Sun、Lijie Fan和Kaiming He。2025。分形生成模型。arXiv预印本 arXiv:2502.17437（2025）。[16] Chi Liu、Huajie Chen、Tianqing Zhu、Jun Zhang和Wanlei Zhou。2023。让深度伪造更具欺骗性：通过痕迹移除攻击规避人脸伪造检测。《IEEE可靠与安全计算汇刊》20卷6期（2023），5182–5196。[17] Nils Lukas、Abdulrahman Diaa、Lucas Fenaux和Florian Kerschbaum。2024。利用优化对图像水印实施自适应攻击。载于《第十二届国际学习表征会议》。https://openreview. net/forum?id=O9PArxKLe1[18] Nils Lukas和Florian Kerschbaum。2023。{PTW}：面向{预训练}图像生成器的关键调优水印。载于《第32届USENIX安全研讨会（USENIX Security 23）》。2241–2258。[19] Yihan Ma、Xinyue Shen、Yiting Qu、Ning Yu、Michael Backes、Savvas Zannettou和Yang Zhang。2025。从迷因到威胁：开源视觉语言模型中的仇恨迷因理解及其诱发的仇恨内容生成。载于《USENIX安全研讨会（USENIX Security）》。USENIX。[20] Weili Nie、Brandon Guo、Yujia Huang、Chaowei Xiao、Arash Vahdat和Animashree Anandkumar。2022。用于对抗净化的扩散模型。载于《国际机器学习会议》。PMLR，16805–16827。[21] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark等。2021。通过自然语言监督学习可迁移视觉模型。载于《国际机器学习会议》。PmLR，8748–8763。[22] Olaf Ronneberger、Philipp Fischer和Thomas Brox。2015。U-Net：用于生物医学图像分割的卷积网络。载于《医学图像计算与计算机辅助干预——MICCAI 2015：第18届国际会议，德国慕尼黑，2015年10月5日至9日，论文集，第III部分第18卷》。Springer，234–241。[23] Mehrdad Saberi、Vinu Sankar Sadasivan、Keivan Rezaei、Aounon Kumar、Atoosa Chegini、Wenxiao Wang和Soheil Feizi。2024。人工智能图像检测器的鲁棒性：基本界限与实际攻击。载于《第十二届国际学习表征会议》。https://openreview.net/forum?id=dLoAdIKENc [24] Xinyue Shen、Yun Shen、Michael Backes和Yang Zhang。2025。GPTracker：对滥用的GPT进行大规模测量。载于《2025年IEEE安全与隐私研讨会（SP）》。IEEE，336–354。[25] Xinyue Shen、Yixin Wu、Yiting Qu、Michael Backes、Savvas Zannettou和Yang Zhang。2025。HateBench：评估仇恨言论检测器对大型语言模型生成内容和仇恨活动的检测能力。（2025）。

[26] Yucheng Shi、Mengnan Du、Xuansheng Wu、Zihan Guan、Jin Sun和Ninghao Liu。2023。通过零样本图像净化进行黑盒后门防御。神经信息处理系统进展36（2023），57336–57366。[27] Matthew Tancik、Ben Mildenhall和Ren Ng。2020。StegaStamp：实体照片中的隐形超链接。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。2117–2126。[28] Shuai Tang、Peisong He、Haoliang Li、Wei Wang、Xinghao Jiang和Yao Zhao。2025。通过内容无关、基于适配器且具备类别感知的增量学习，实现可扩展的人工智能生成图像检测。*IEEE*信息取证与安全汇刊（2025）。[29] Florian Tramer和Dan Boneh。2019。针对多种扰动的对抗训练与鲁棒性。神经信息处理系统进展32（2019）。[30] Guanjie Wang、Zehua Ma、Chang Liu、Xi Yang、Han Fang、Weiming Zhang和Nenghai Yu。2024。Must：面向多源追踪的鲁棒图像水印。载于*AAAI*人工智能会议论文集，第38卷。5364–5371。[31] Yuxin Wen、John Kirchenbauer、Jonas Geiping和Tom Goldstein。2023。Tree-Ring水印：扩散图像中的隐形指纹。神经信息处理系统进展36（2023），58047–58063。[32] Zijin Yang、Kai Zeng、Kejiang Chen、Han Fang、Weiming Zhang和Nenghai Yu。2024。高斯着色：可证明性能无损的扩散模型图像水印。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。12162–12171。[33] Ning Yu、Vladislav Skripniuk、Sahar Abdelnabi和Mario Fritz。2021。生成模型的人工指纹标记：将深度伪造溯源植根于训练数据。载于*IEEE/CVF*国际计算机视觉会议论文集。14448–14457。[34] Ning Yu、Vladislav Skripniuk、Dingfan Chen、Larry S Davis和Mario Fritz。2022。利用可扩展指纹标记负责任地披露生成模型。载于国际学习表征会议。[35] Qi Zhang、Yifei Wang和Yisen Wang。2022。掩码为何重要：迈向对掩码自编码器的理论理解。神经信息处理系统进展35（2022），27127–27139。[36] Hanqing Zhao、Wenbo Zhou、Dongdong Chen、Tianyi Wei、Weiming Zhang和Nenghai Yu。2021。多注意力深度伪造检测。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。2185–2194。[37] Xuandong Zhao、Kexun Zhang、Zihao Su、Saastha Vasan、Ilya Grishchenko、Christopher Kruegel、Giovanni Vigna、Yu-Xiang Wang和Lei Li。2024。生成式人工智能可证明能够移除不可见图像水印。神经信息处理系统进展37（2024），8643–8672。[38] Jiachen Zhou、Peizhuo Lv、Yibing Lan、Guozhu Meng、Kai Chen和Hualong Ma。2024。Dataelixir：通过扩散模型净化中毒数据集，以缓解后门攻击。载于*AAAI*人工智能会议论文集，第38卷。21850–21858。[39] Jiren Zhu、Russell Kaplan、Justin Johnson和Li Fei-Fei。2018。HiDDeN：利用深度网络隐藏数据。载于欧洲计算机视觉会议论文集（*ECCV*）。657–672。[40] Mingjian Zhu、Hanting Chen、Qiangyu Yan、Xudong Huang、Guanyu Lin、Wei Li、Zhijun Tu、Hailin Hu、Jie Hu和Yunhe Wang。2023。Genimage：用于检测人工智能生成图像的百万规模基准测试。神经信息处理系统进展36（2023），77771–77782。

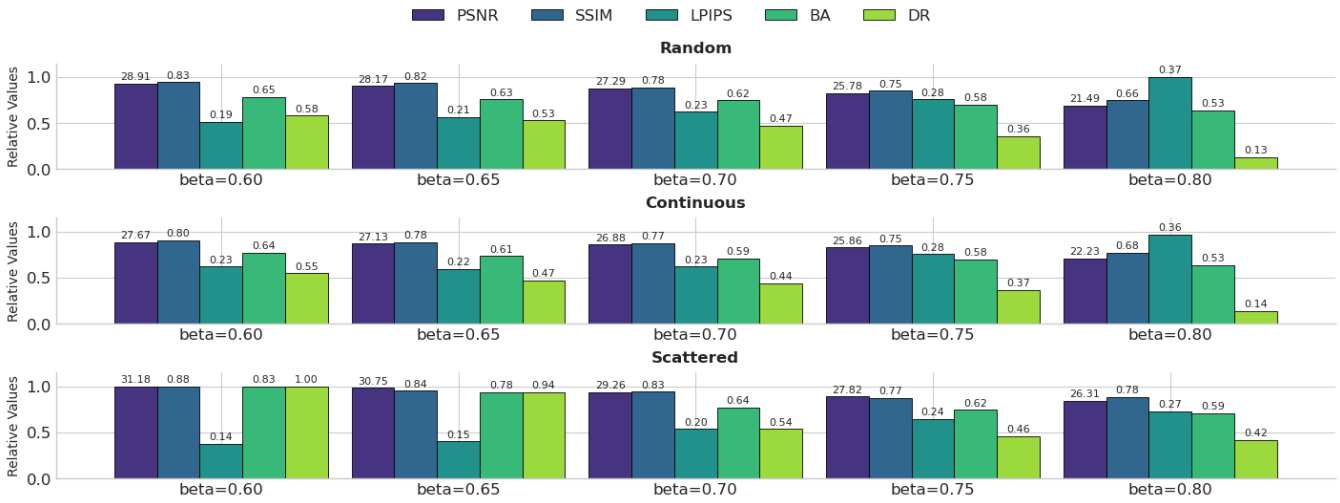



**图7：掩码策略和比例的影响。每个子图展示了一种掩码策略在不同掩码比例下的实验结果，比例范围从**0.6 **到**0.8 **，步长为**0.05 **。柱形的高度表示某项指标下的归一化值，实际数值标注在柱形上方。**

## A 实验设置

**图像处理**裁剪、JPEG压缩、量化、高斯模糊和引导滤波模糊被选作图像处理方法，用于扭曲带水印图像以去除嵌入式水印。参数的选择遵循先前研究 [12, 18]。

**数据集**。已实现并发布的防御基线是在各种数据集上训练的。因此，我们在与各防御方案对应的数据集上评估攻击：TRW 和 StableSignature 使用 LAION-5B，GANF 和 AF 使用 CelebA，PTW 使用 FFHQ，HiDDeN 使用 COCO，StegaStamp 使用缩小至 $400 \times 400$ 的 CelebA-HQ-1024。根据先前研究 [9, 12]，评估结果为从 100 张随机图像中得出的平均值。

**实现细节**。对于 HSN 中的生成器，我们采用掩码自编码器视觉Transformer [8] 作为其模型架构。在 HS + 中，掩码模型使用 UNet [22] 架构，像素生成器则遵循分形生成模型 [15]。评估时，我们直接沿用其他方法已发布实现的默认设置。

**防御基线**。在现有的所有图像防御水印方案中，我们选取以下最先进的水印方法作为基线方法。这是因为这些工作非常适用于实际场景：在该场景中，假设对手只能黑盒访问生成式人工智能 API，且无法访问检测器。

- **人工指纹 (AF)** [33] 将指纹嵌入训练生成模型所用的训练数据集中。训练充分的生成模型会因可迁移性而自动生成带有这些指纹的合成图像。

- **生成式人工网络指纹 (GANF)**[34] 创建比特串指纹嵌入，并将这些嵌入融入生成模型的多个潜在输出中。因此，该生成模型生成的合成图像会携带这些指纹，并可通过逐比特重建来进行验证。

- **关键调优水印 (PTW)** [18] 冻结一个预训练生成模型，并复制其参数以创建一个对应模型。随后，通过专门设计的约束训练该对应模型，使其性能与预训练模型相近，同时输出带水印图像。

- **StegaStamp** [27] 将不可见的比特串水印嵌入合成图像中。这些水印具有很高的鲁棒性，即使图像在真实场景中遭受严重失真，也能成功重建。

- **HiDDeN** [39]使用编码器网络将水印嵌入合成图像。在训练过程中，利用噪声层对合成图像进行失真处理，然后由解码器网络重建水印，从而提升水印的鲁棒性。

- **StableSignature** [5] 首先训练水印编码器和解码器，以嵌入并重建比特串水印。接着，对潜在扩散模型的解码器进行微调，使其输出带有固定比特串水印的图像，并确保预训练的解码器能够正确解码该水印。

- **树环水印 (TRW)** [31] 在去噪扩散隐式模型的反向过程中，将水印嵌入初始噪声的傅里叶空间。受到攻击后，可将合成图像反演为初始噪声，并据此恢复水印。这些水印对失真和攻击具有很强的抵抗力。

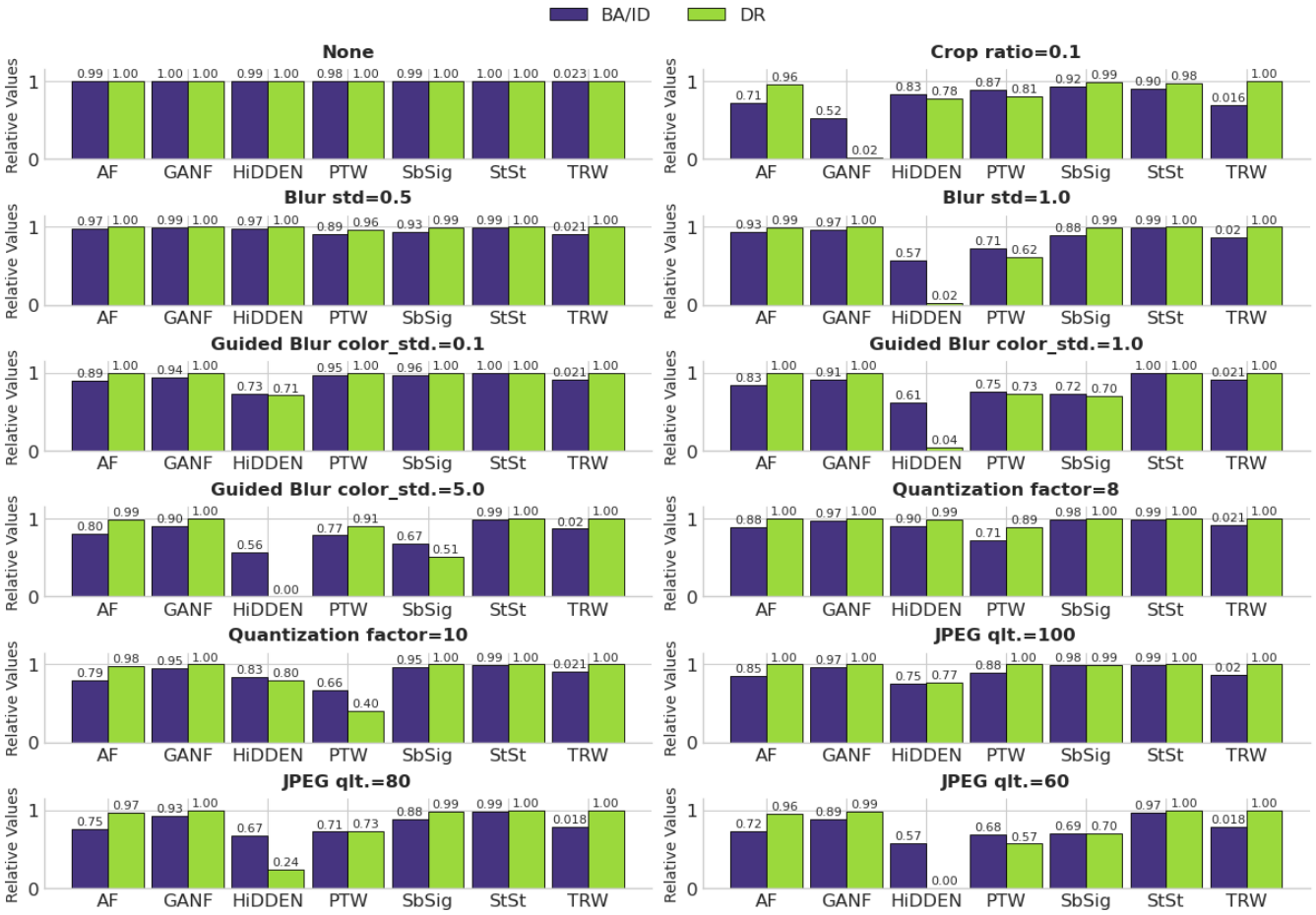



**图8：评估防御水印对抗图像操纵的效果。每个子图展示了一种图像操纵攻击不同防御水印的结果。StableSignature和StegaStamp分别缩写为SbSig和StSt。TRW的蓝色柱表示逆距离，其余蓝色柱表示比特准确率。柱高表示某项指标下的归一化值，实际数值标注在柱顶。**

**攻击基线**。选择以下最先进攻击，是因为它们具有出色的攻击性能，且非常适用于我们的攻击场景：对手无法访问检测器，只能黑盒访问生成式人工智能 API。

- **DiffAttack** [1, 23, 37] 采用扩散模型对受扰动的带水印图像进行复原，从而生成净化图像。
- **VAEAttack** [1, 37] 与 DiffAttack 类似，但将其生成模型替换为变分自编码器模型。
- **UnMarker** [12]分别优化两种扰动，分别用于扰动带水印图像的高频和低频分量。




**图9：带水印图像遭受不同方法攻击后的视觉效果。**