基于零空间螺旋相位的扩散模型高容量鲁棒水印

生成模型第 10 小组

2026 年 1 月 17 日

1 项目简介与代码实现

本项目旨在解决现有 Stable Diffusion 模型水印方案中容量不足的问题。基于"Shallow Diffuse" 框架，我们引入了创新的 **螺旋相位 (Circular Harmonic Decomposition)** 编码策略，在保证图像质量的前提下实现了水印容量的指数级增长。

我们编写并提交了以下三个核心脚本来展示和评估我们的方法：

- `t2i_64ID.py`: 这是一个稳定的实现版本，支持 64 个独立用户 ID。该版本经过参数调优，在多种攻击下表现出极高的鲁棒性。
- `t2i_256ID.py`: 这是一个实验性的高容量版本，支持 256 个 ID。该版本探索了频域编码的密度极限。
- `t2i_64ID_test.py`: 这是一个综合测试脚本。它会自动遍历 1000 个 Prompt 和所有 64 个 ID，生成统计报告并绘制可视化图表，用于评估不同攻击下的准确还原率。

2 理论框架与核心创新

2.1 背景：零空间水印 (Shallow Diffuse)

我们的工作建立在 Shallow Diffuse 方法之上。其核心思想是将水印信号注入到扩散模型去噪函数 Jacobian 矩阵的 **零空间 (Null Space)** 中。数学上，若 J 为去噪函数的 Jacobian 矩阵，则零空间投影算子为 $P_{null} = I - J^{\dagger}J$。任何经过 P_{null} 投影的信号 w (即 $w_{proj} = P_{null} \cdot w$)，在理论上对于图像生成的语义内容是 "不可见" 的，但物理上却保留在 Latent Code 中。这确保了水印的注入不会破坏图像的结构，这是相比于简单叠加水印的显著优势。

2.2 核心创新：螺旋相位与指数级容量增长

原版 Shallow Diffuse 及类似的 Tree-Ring 方法主要通过**径向切分 (Radial Slicing)** 来编码信息。例如切分 16 个圆环，只能代表 16 个 ID，容量随环数呈**线性增长** ($Capacity \propto N_{rings}$)，导致高容量需求下频域过于拥挤。

我们的创新在于引入了 **螺旋相位编码**，利用频域的角向维度 θ。构造水印图案 W 如下：

$$W(r, \theta) = M(r) \cdot e^{ik\theta} \tag{1}$$

其中 $M(r)$ 是径向掩码，k 是旋转频率。我们的创新意义在于：

1. **由线性增长转变为指数增长**：我们在每个圆环上叠加不同的螺旋频率 k。若有 R 个圆环，每个圆环可区分 S 种频率状态，则总容量为 S^R。例如在 64-ID 方案中，我们仅用了 3 个圆环，每个圆环 4 种状态，即实现了 $4^3 = 64$ 的容量。这是前人未曾尝试的高密度编码方式。

2. **正交性与隐蔽性**：螺旋相位 $e^{ik\theta}$ 在角向积分上是正交的。我们将这种特殊的物理信号与零空间投影相结合，发现螺旋纹理能很好地通过投影算子，被模型视为“纹理细节”而非“破坏性噪声”保留下来。也正是因为这些特性只有在 Shallow Diffuse 的零空间投影中，才不会破坏图案内容，因此前人在图片水印植入中并没有纳入这种图案模式，使得这种图案模式成为 Shallow Diffuse 独有的扩大容量的创新改进。

2.3 频率选择策略

我们特意从 **质数或奇数**集合中选择旋转频率 k（例如 $k \in \{7, 11, 13, 17, \dots\}$），并避开了 $2, 3, 4$ 等小整数。**原因**：自然图像通常包含大量的水平和垂直结构，这些结构在频域对应于低频偶数 k。选择高频质数可以使水印信号避开自然图像能量集中的区域，减少干扰。

3 实验分析与局限性讨论

3.1 64-ID 方案 (稳定版)

经过 `t2i_64ID_test.py` 的大规模测试，该方案在 JPEG 压缩、高斯模糊、裁剪等常规攻击下均表现出极高的准确率。但发现 $ID < 10$ 的用户在部分攻击下识别率略低。**原因分析**：较小的 ID 通常对应于较小的频率 k (如 $k = 7$) 或特定的相位组合。低频信号波长较长，更容易与图像本身的轮廓信息发生混叠，导致检测时的信噪比不如高频信号高。在针对最内侧的圆环单独增加权重后，可以在更准确的识别和轻微画质损失中权衡，实现不错的检验效果。（如图）

(a) 加入水印前

(b) 加入水印后

图 1: ID=2, 强度设为 1.2 设定下的水印植入效果。上右图可以在 diff 和 diffpure 以外的十种攻击下全部精确还原出水印 ID 为 2，且图片质量并无显著下降

3.2 256-ID 方案 (实验版)

该方案通过增加圆环数量实现了 256 ID 的编码。目前虽已能跑通，但存在画质与准确率的权衡问题。由于 Stable Diffusion 的 Latent 空间仅为 64×64，频域资源极其有限。强行塞入 4 层螺旋信号会导致空域出现可见差异，且不同环之间存在干扰，导致约 1-2 bit 的解码误差。

3.3 对抗 Diff 和 DiffPure 攻击的局限性

实验数据显示，我们的方法在面对 **Diff (Diffusion Attack)** 和 **DiffPure (Diffusion Purification)** 攻击时表现不佳，还原率较低。**原因分析**：

- 这两种攻击的本质是利用扩散模型重新生成图像（或加噪后去噪）。
- 我们的水印信号主要寄宿在**零空间的高频分量**中。
- 扩散模型的生成过程本身就是一个“去噪”过程，它倾向于保留语义信息而抹除高频残差。当攻击者强行进行重生成时，模型会将我们的螺旋水印视为“非语义噪声”并将其清洗掉，从而导致水印丢失。这是基于零空间水印方法的共同短板。

4 运行指南

以下是复现我们实验的推荐参数指令：

1. 64-ID 生成 (推荐稳定版)

```
python t2i_64ID.py --run_name 64ID_Final --model_id /root/autodl-tmp/sd-model --start
    0 --end 1 --edit_time_list 0.3 --w_channel 3 --w_pattern spiral --w_measurement
    accuracy --w_injection complex2
```

2. 256-ID 生成 (高容量实验版)

```
python t2i_256ID.py --run_name 256ID_Final --model_id /root/autodl-tmp/sd-model --
    start 0 --end 1 --edit_time_list 0.3 --w_channel 3 --w_pattern spiral --
    w_measurement accuracy --w_injection complex2
```

3. 综合测试 (遍历 1000 Prompts, 64 IDs)

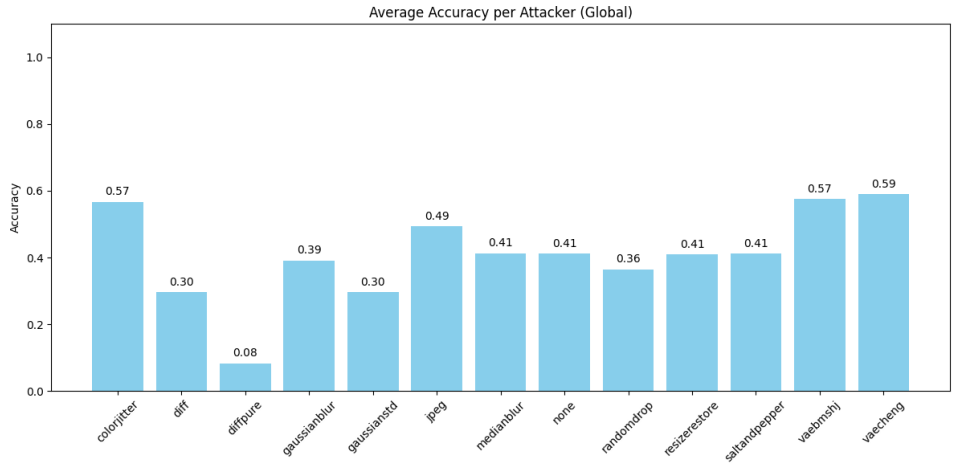
```
# 注意: 该测试较为耗时, 将生成统计热力图
python t2i_64ID_test.py --run_name 64ID_Test_Report --model_id /root/autodl-tmp/sd-
    model --start 0 --end 1000 --edit_time_list 0.3 --w_channel 3 --w_pattern spiral
    --w_measurement accuracy --w_injection complex2
```

5 实验结果

分别在 64ID 和 256ID 的容量下，对 1000 个 prompt 进行图片生成，对应不同的水印 ID，随后对攻击的图片进行还原，对还原情况进行可视化：

- 对于 ID 容量 64 的情况，在 diff 和 diffpure 以外的攻击中可以还原大部分 ID。如图 (a) 中三个环中加信号位的强度通常显著强于其他位置，从而可以准确得出水印编号，外侧的环针对攻击鲁棒性更好。
- 对于 ID 容量 256 的情况，面对各种攻击的准确还原率通常不足一半，如图 (b)。查看结果后发现通常只差一位，多由于内侧环出错所致。因为能精确还原全部 k 的圆环需要有足够宽度，因此能容纳的环数量有限。如果继续增大强度可以提升还原成功率，但是图片局部会出现不正常颜色。
- 综上认为已经基本可以承担 64 的 ID 容量，如果要进一步扩大容量至 256 甚至更多，可以进一步权衡调整各参数进行尝试。更可行的方法或是引入冗余位和纠错位，用信息论的方法在允许少量出错的前提下精确还原。

(a) ID 容量 64 时，水印内容对于大多种类攻击鲁棒



(b) ID 容量 256 时，对于各种攻击的还原能力有限

图 2: 不同 ID 容量下进行攻击实验结果可视化