PlugMark: A Plug-in Zero-Watermarking Framework for Diffusion Models

Supplementary Material

A. Time Cost

To assess the efficiency of PlugMark in generating boundary representations (BRoK), we compare average time costs for BRoKs with different numbers (β) of boundary sample pairs. As shown in Table 1, if each BRoK is composed of only $\beta = 1$ sample pair, the generation time is approximately 5 seconds. As β increases, the time grows approximately linearly. In other words, for generating the 10 BRoKs as watermarks, the total time required is about 540 seconds. This is a reasonable and feasible time cost for generating a highly robust zero-distortion watermark for a diffusion model.

Sample number (β)	Average time(s)
1	5.31
5	23.89
10	54.33
15	77.23

Table 1. Time cost of a BRoK with different number of samples

B. Generalization

To test the generalization capability of PlugMark, we conduct experiments on different versions of diffusion models. Specifically, we evaluated Stable-Diffusion-v1-5,Stable-Diffusion-v2-1, and Stable-Diffusion-v3-5. As shown in Table 2, PlugMark achieves excellent matching accuracy and TPR(FPR=10^{-6}) for post-processed models, demonstrating strong robustness against fine-tuning and pruning across these models. Theoretically, as long as a model can extract intermediate layer features, it can be used for zero-watermarking with this approach. We see this as an interesting future work.

Model	**Origin**	**Fine-tuning (2000 steps)**	**Pruning**($\rho = 0.1$)
Stable diffusion-v-1.5	100.00/1.00	98.50/0.967	98.33/0.995
Stable diffusion-v-2.1	100.00/1.00	98.66/0.971	98.41/0.995
Stable diffusion-v-3.5	100.00/1.00	98.00/0.965	97.51/0.990

Table 2. Matching accuracy (%) and TPR(FPR=10^{-6})

C. Model Editing (Soups)

We evaluate PlugMark's robustness against model editing by merging Stable-Diffusion-v2 models fine-tuned on different downstream tasks for 2000 steps using DreamBooth. We average their weights as an edited version[59]. As shown in Table 3, for models merged from 2 fine-tuned versions, and 3 fine-tuned versions, matching accuracy remains a quite remarkable performance, demonstrating our method's robustness against model editing.

Model	**Combinations**			
Model 1	✓	✓		✓
Model 2	✓		✓	✓
Model 3		✓	✓	✓
Matching accuracy	95.32	94.89	94.27	93.36

Table 3. Matching accuracy (%) of models edited by merging different models fine-tuned for different downstream tasks.

D. Fine-tuning setup

D.1. DreamBooth fine-tuning

The models are fine-tuned using DreamBooth with a learning rate of 10^{-5} ,and the personalized data are shown in Fig.1. We adhere to the recommended hyperparameters for other settings[45].

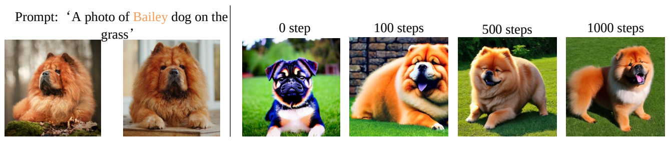

Figure 1. Personalized data for DreamBooth fine-tuning.

D.2. LoRA fine-tuning

We attempt to employ more fine-tuning approaches to evaluate PlugMark's robustness. We utilize the Pokemon dataset [35] for LoRA fine-tuning with a rank of 128 and the learning rate is 10^{-5}.As shown in Table.4, although fine-tuning using LoRA is specially designed for partial weights for attention layers, which is more influential to the upblock features of UNet, PlugMark still maintains a relative strong robustness against it.

Model	**100 steps**	**200 steps**	**2000 steps**
Stable diffusion-v-1.5	1.000	0.994	0.960
Stable diffusion-v-2.1	0.996	0.995	0.971
Stable diffusion-v-3.5	0.997	0.990	0.957

Table 4. TPR(FPR=10^{-6}) of Diffusion models using LoRA

E. Impact of knowledge extractor choice

The results are in Table 5. Since decision boundaries are required, we set different classifier architectures as the knowledge extractor (DKe). When the classification precision surpasses 93%, For any DKe structure, the generated BRoK can effectively represent the boundaries and result in high accuracy.

DKe structure	ResNet-18	MobileNetV2	EfficientNet-B0	ResNet-34
DKe precision(%)	97.06	93.16	95.13	97.49
Matching accuracy(%)	100.00	98.78	100.00	100.00

Table 5. Matching accuracy (%) with different DKe

F. Selection of diffusion timestep t.

As shown in Table 6, compared to later timesteps, earlier ones indeed perform better. However, if all selected timesteps are earlier, it may lead to insufficient coverage of the model's knowledge, resulting in reduced robustness against fine-tuning. The entire range guarantees full coverage, and uniform sampling further ensures that every stage is adequately covered.

Timestep Range	Origin		Fine-tuning (1000 steps)	
	uniform	random	uniform	random
0-100	99.82	99.20	95.03	94.85
100-500	99.71	98.13	95.12	93.89
500-1000	99.18	98.05	94.83	93.13
0-1000	100.00	99.33	98.83	95.74

Table 6. Matching accuracy (%) with different timestep ranges

G. Additional references on diffusion-native watermarking methods.

We add comparative experiments in Table 7. PlugMark significantly outperforms other methods, as they are not designed for fine-tuning.

Method	**FID↓**	**Matching accuracy(%)↑/TPR↑(FPR=10^{-6})**	
		Origin	Fine-tuning 1000 steps
Gaussian Shading [65]	24.63	100.00/1.000	52.13 /0.002
WMAdapter [5]	24.78	98.00/0.999	-
PRC-watermark [18]	24.31	99.55/1.000	51.22/0.003
PlugMark	**24.25**	**100.00/1.000**	**98.83/0.995**

Table 7. Additional comparison with diffusion-native watermarking methods

H. Discussion

In the downstream fine-tuning tasks of diffusion models, it is common to fine-tune not only the UNet but also other components such as the VAE decoder, text encoder, and other modules. Since Plugmark primarily targets the uniqueness of features within the UNet, fine-tuning these other components has minimal impact on the watermark verification process. Consequently, we believe this algorithm can exhibit strong adaptability to various fine-tuning tasks. Furthermore, this plug-and-play zero-watermarking mechanism can be applied to various models without introducing additional perturbations or disrupting the model's inherent structure, making it suitable for complex models composed of multiple components.

Additionally, we believe that the idea of treating intermediate features with unique representational capabilities as a form of distinctive knowledge and using boundary samples to uniquely characterize them can be extended to various models beyond diffusion models. Extensions, particularly for large language models with unique parametric architectures, will be quite meaningful. We consider these ideas as interesting avenues for future work.

I. TPR with a Fixed FPR

We assume there are a total of γ sample pairs $s = \{(image_i, text_i)_i\}^\gamma$ to be verified for a model in PlugMark. For other methods, we assume a γ-bit binary watermark $s \in \{0,1\}^\gamma$ is embedded. All watermarking techniques are regarded as single-bit schemes using a fixed watermark s. A threshold τ is preset within the range 0 to γ. A model is identified as watermarked if the accuracy score $\text{Acc}(s, s')$ reaches or exceeds τ. Previous research[67] assumed that the extracted watermark results $s'_1, \ldots, s'_\gamma$ from negative models follow a random and uniform distribution, where each bit s'_i can be described by a Bernoulli process with a success probability of 0.5. Consequently, the accuracy $\text{Acc}(s, s')$ conforms to a binomial distribution with parameters $(k, 0.5)$. Based on this distribution, the false positive rate (FPR) is defined as the likelihood that $\text{Acc}(s, s')$ for a negative model surpasses the threshold τ. This probability can be represented through the regularized incomplete beta function $B_x(a; b)$:

$$\text{FPR}(\tau) = P(\text{Acc}(s, s') > \tau) = \sum_{i=\tau+1}^{k} \binom{k}{i} \frac{1}{2^k} = B_{\frac{1}{2}}(\tau+1, k-\tau). \tag{1}$$

To ensure an FPR of 10^{-6}, we determine the appropriate threshold τ and assess the true positive rate (TPR) using 300 post-processed models for PlugMark and 1000 watermarked images for the compared black-box models. As shown in Table 7, our method demonstrates remarkable performance.