EntropyMark: Towards More Harmless Backdoor Watermark via Entropy-based Constraint for Open-source Dataset Copyright Protection

Supplementary Material

Nomenclature

Data

$\mathcal{D}$	original dataset
$\mathcal{D}_s$	selected subset to be watermarked
$\mathcal{D}_m$	modified watermarked subset
$\mathcal{D}_b$	benign subset
$\mathcal{D}_w$	watermarked dataset
$\mathcal{D}_f$	full-watermarked dataset
$\mathcal{D}_v$	verification set

Loss Function

$\mathcal{L}_{ce}$	cross-entropy loss
$\mathcal{L}_d$	dispersion loss
$\mathcal{L}_{ver}$	verification loss
$\mathcal{L}_{train}$	fine-tuning loss

Module

G_ψ	watermark generator
S_θ	surrogate model
F	suspicious model

Metric

$\mathcal{C}_w$	prediction consistency indices on $\mathcal{D}_f$
$\mathcal{C}_b$	prediction consistency indices on $\mathcal{D}$
ΔE	confidence score

Others

R	iteration rounds
γ	watermarking rate
β	dispersion loss hyper-parameter
$\mathcal{H}$	entropy

A. Dataset Settings

We conduct experiments on two benchmark datasets: CIFAR-10 [18] and a 12-class subset of ImageNet [23]. The specific details of the datasets used in our experiments are summarized in Tab. 7.

A.1. CIFAR-10

CIFAR-10 comprises 10 classes with a total of 50,000 training samples (5,000 images per class) and 10,000 validation samples (1,000 images per class). Each sample is resized to 32×32 pixels by default.

A.2. Sub-ImageNet-12

Sub-ImageNet-12 is a commonly used subset of ImageNet, consisting of 12 classes with 12,480 training samples (1,040 images per class) and 3,120 validation samples (260 images per class). Each sample is resized as 64×64 by default.

dataset	classes	size	samples
CIFAR-10	10	32×32×3	50000+10000
Sub-ImageNet	12	64×64×3	12480+3120

Table 7. Details of the datasets in our experiments.

B. Detailed Settings For EntropyMark

In this section, we detailed the settings of EntropyMark across four phases: generator pre-training, dataset watermarking, dataset ownership verification, and defenses.

B.1. Threat Model

Defenders (*i.e.*, dataset owners) typically lack knowledge of the model architecture and training components and can only detect unauthorized use based on the model's prediction behavior. In contrast, attackers (*i.e.*, dataset stealers) have access to the protected dataset and can manipulate the training model and its components.

B.2. Image Steganography Settings

We adopt the configuration established by StegaStamp [35], utilizing a U-Net-style network as the encoder and a spatial transformer network as the decoder. The watermark size is set to 32×32 for the CIFAR-10 dataset and 64×64 for the Sub-ImageNet-12 dataset.

During the generator pre-training phase, both the encoder and decoder are trained for the first 2 epochs, using

only the code reconstruction loss. This is followed by joint training of the encoder and decoder for 20 epochs with the full set of loss functions. The secret length is fixed at 3, with Adam as the optimizer and an initial learning rate of 0.0001. The scaling hyper-parameters for the code reconstruction loss, LIPIS perceptual loss, l_2 residual regularization loss, and critic loss are 1.5, 1.5, 2.0, and 0.5, respectively.

B.3. Dataset Watermarking Settings

Algorithm 1 outlines the dataset watermarking process employed in EntropyMark. For both CIFAR-10 and Sub-ImageNet-12, the process alternates between training the surrogate model and fine-tuning the watermark generator over 2 rounds. The final fine-tuned watermark generator is then used to modify a portion of the dataset.

The surrogate model is trained for 100 epochs using the current watermarked dataset with a batch size of 128. The training utilizes the SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The initial learning rate is set to 0.1 and is reduced by a factor of 10 at the 50^{th} and 80^{th} epochs.

Fine-tuning the watermark generator also spans 100 epochs and is performed using a modified subset of the watermarked dataset with a batch size of 128. The Adam optimizer is employed with an initial learning rate of 0.0001, which is reduced by a factor of 10 at the 37^{th}, 62^{th}, and 87^{th} epochs. All samples from the source class are designated as verification samples for gradient matching. An 8×8 black and white square is used as the trigger, randomly attached to the verification samples.

Algorithm 1 Backdoor Watermarking

Input: Benign Dataset $\mathcal{D}$, Pre-trained Watermark Generator G_ψ, Surrogate Model S_θ, Iteration Rounds R
Output:Watermarked Dataset $\mathcal{D}_w$

Randomly initialize the watermarked sample list l
for $i = 1, \ldots, R$ **do**
  Watermark $\mathcal{D}$ with G_ψ according to l to get $\mathcal{D}_w$.
  Train S_θ on watermarked dataset $\mathcal{D}_w$ using Cross-Entropy Loss $\mathcal{L}_{ce}$.
  Fine-tune G_ψ only on modified samples $\mathcal{D}_s$ using Dispersion Loss $\mathcal{L}_d$.
  Update l according to adaptive data selection.
end for
Watermark $\mathcal{D}$ with G_ψ according to l to get $\mathcal{D}_w$.
return $\mathcal{D}_w$

B.4. Dataset Ownership Verification Settings

We evaluate the verification effectiveness under three scenarios: 'Watermark', 'Benign', and 'Other'. In the 'Watermark' scenario, the suspicious model is queried using our patched verification samples. In the 'Benign' scenario, the same verification samples are used to query a benign model. In the 'Other' scenario, the suspicious model is queried with verification samples containing different trigger patterns.

The source label is set to 0, and all test samples from the source class are used for the hypothesis testing. The margin τ is set to 0.005 for CIFAR-10 and -0.002 for Sub-ImageNet-12.

B.5. Defenses Settings

We apply model pruning and fine-tuning defenses on the CIFAR-10 dataset, as described in the main manuscript.

For the model pruning defense, the second layer of the watermarked model is pruned using 20% of the benign training samples. The watermark success rate (WSR) and benign accuracy (BA) are measured at pruning rates ranging from 0% to 100%, recorded at 10% increments.

For the fine-tuning defense, the convolutional layers are frozen, and the fully connected layers of the watermarked models are fine-tuned for 100 epochs using 10% of the benign training samples. The WSR and BA are recorded every 10 epochs. This process utilizes 10% of the benign training samples with a fixed learning rate of 0.01.

C. Watermarking Baselines Settings

We train all watermarking baselines for 200 epochs using ResNet-18. The training is conducted with an SGD optimizer, starting with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 5×10^{-4}. The learning rate is reduced by a factor of 10 at the 150^{th} and 180^{th} epochs. The batch size is set to 128 for both CIFAR-10 and Sub-ImageNet-12.

We implement BadNets, Blend, and Label Consistent using the open-source toolbox BACKDOORBOX [25]. The experiments of the Sleeper Agent, UBW, and domain watermark are conducted using their official codes.

C.1. BadNets Settings

We use an 8×8 black and white square as the trigger for both CIFAR-10 and Sub-ImageNet-12, fixed in the bottom right corner of the original images. The target label is set to 0 by default.

C.2. Blend Settings

The random Gaussian noise is used as the trigger on CIFAR-10 and Sub-ImageNet-12, with a blend ratio of 0.2 for both datasets. The target label is set to 0 by default.

C.3. Label Consistent Attack Settings

We use Projected Gradient Descent (PGD) to generate adversarial perturbations within the l_∞ ball for preprocessing. The number of training steps is set to 100, with a step size of 1.5 and a maximum perturbation size of 8. Afterward,

the BadNets trigger is applied to create poisoned samples. The target label is set to 0 and 50% of the training samples from the target label are poisoned.

C.4. Sleeper Agent Settings

Following the original settings, we poison 1% of the training images, with each perturbation constrained within an l_∞ ball of radius $16/255$. During poison crafting, an Adam optimizer is used with an initial learning rate of 0.025, which is reduced by a factor of 10 at the 93^{rd}, 156^{th}, and 218^{th} epochs. Poison crafting is performed over 250 epochs, with the surrogate model retrained every 50 epochs. The target label is set to 1 and the source label is set to 2. We select 5,000 samples from the source class of CIFAR-10 and 1,248 of Sub-ImageNet-12 as source samples, all of which are embedded with the BadNets trigger.

C.5. UBW Settings

For UBW-P, we poison 10% of the training samples using the BadNets triggers, with the target label set to 0. For UBW-C, we follow the poison crafting settings of the Sleeper Agent. The non-negative trade-off hyper-parameter of UBW-C is set to 2.0.

C.6. Domain Watermark Settings

We use the official model to test performance on CIFAR-10 and reproduce the domain-transfer method to generate the corresponding domain-watermarked samples from Sub-ImageNet-12. Following the original settings, we optimize the Upper-level Sub-problem for 50 iterations and optimize the Lower-level Sub-problem for 100 iterations. These domain-watermarked samples are then applied to the official backdoor training code.

D. Discussion of Additional Parameters

D.1. Effect of Dispersion Loss Hyper-parameter

We investigate the impact of varying the dispersion loss hyper-parameter β on EntropyMark. As shown in Tab. 8, both WSR and BA remain largely stable across different values of β. However, when β is too small, the loss function exerts minimal influence on prediction confidence, making it difficult to distinguish between benign and watermarked models based on entropy inconsistency. This can lead to verification failure.

D.2. Effect of Patch Size

We also examine the effects of varying patch sizes. As indicated in Tab. 9, WSR increases with larger patch sizes, while BA remains relatively stable. In addition, we observe that as the patch size decreases, the p-value gap between the 'Watermark' scenario and other scenarios becomes more pronounced. Notably, even a 2×2 patch can still achieve effective verification.

β	WSR (%)	BA (%)	Watermark	Benign	Other
1.0	83.02	88.56	0.9994	1.0000	0.9989
2.0	82.47	87.79	0.0017	1.0000	1.0000
3.0	82.05	88.32	0.0285	1.0000	0.9940

Table 8. The effect of trade-off parameter β.

Size	WSR (%)	BA (%)	Watermark	Benign	Other
2	71.78	87.17	0.2104	1.0000	0.9948
8	82.47	87.79	0.0017	1.0000	1.0000

Table 9. The effect of patch size.

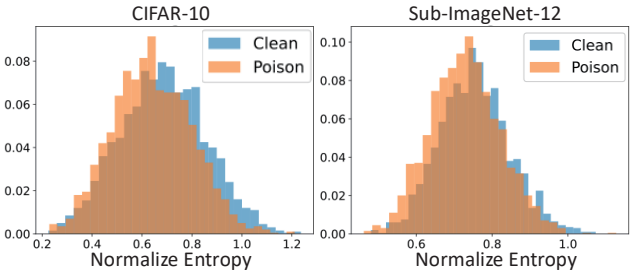

Figure 5. Experimental results of STRIP.

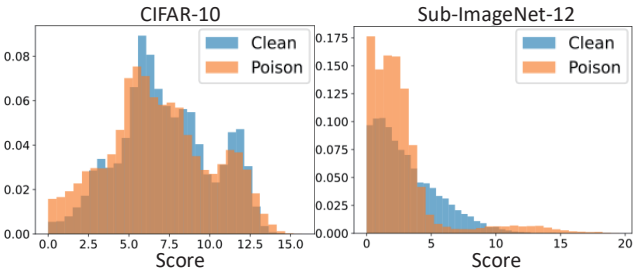

Figure 6. Experimental results of Spectral.

E. Resistance to Additional Defenses

E.1. Resistance to STRIP

STRIP [6] detects backdoored samples based on the assumption that a backdoored model will produce stable predictions for malicious inputs. It calculates the entropy of prediction probabilities after randomly overlaying benign samples onto potentially malicious inputs. For detection, we randomly select 2,000 malicious and benign samples, using the other 10,000 benign samples as the overlay background. As shown in Fig. 5, there is a significant overlap in the entropy distributions of benign and malicious samples, indicating that STRIP cannot effectively distinguish between these two types.

E.2. Resistance to Signature Spectral

Signature Spectral [37] detects malicious samples by analyzing detectable traces in the feature covariance spectrum.

It calculates feature correlations and uses the top singular value as an outlier score for each sample. As illustrated in Fig. 6, the distributions of outlier scores from benign and malicious samples show significant overlap. This overlap makes it challenging to establish a clear threshold for differentiation, demonstrating the resistance of EntropyMark to detection by Signature Spectral.

E.3. Resistance to CLP

Channel Lipschitzness-based Pruning (CLP) [43] is a data-free model pruning method designed to repair backdoored models through simple channel pruning. We implement this defense using the official code and follow the recommended parameter settings. As shown in Tab. 10, while CLP mitigates the implanted backdoor, it significantly degrades the model's performance on benign samples. This result demonstrates that EntropyMark effectively resists CLP.

Dataset	Backdoored		CLP Pruned	
	ASR(%)	BA(%)	ASR(%)	BA(%)
CIFAR-10	82.47	87.79	48.66	55.66
Sub-ImageNet-12	66.70	71.35	47.53	53.30

Table 10. Experimental results of CLP.

E.4. Resistance to Adaptive Training

We introduce adaptive training, incorporating a negative entropy constraint in the training loss to prevent over-fitting. As shown in Tab. 11, while adaptive training reduces the p-value gaps between the 'Watermark' and other scenarios, EntropyMark still achieves successful verification.

Defense	WSR (%)	BA (%)	Watermark	Benign	Other
Adaptive Training	89.54	92.51 $_{-0.45}$	0.3323	1.0000	0.9934
Adaptive Training +	88.96	92.61 $_{-0.35}$	0.4293	1.0000	0.6285

Table 11. Resistance to adaptive training. '+' indicates a stronger negative entropy constraint. The decline in BA compared to clean models is highlighted in red.

F. Resistance to Data Augmentation

Dataset stealers often use various data augmentation methods during model training, making it crucial to verify the resilience of EntropyMark against strong data augmentations. Specifically, we evaluate two common data augmentation strategies: random cropping and center cropping. For random cropping, images are resized to 40×40 and then randomly cropped to 32×32 with a 4-pixel padding. For center cropping, images are resized to 36×36 and then cropped to 32×32 with a 2-pixel padding.

As shown in Tab. 12, while WSR decreases when these augmentation methods are applied, EntropyMark remains effective for dataset ownership verification. However, data augmentation does have some impact on verification effectiveness, which we identify as a key area for future research.

Cropping	WSR (%)	BA (%)	Watermark	Benign	Other
Random	79.32	85.39	0.4933	0.6059	0.7213
Center	68.54	88.17	0.1651	1.0000	0.7537

Table 12. The effect of trade-off parameter β.

G. Stealthiness Evaluation

We adopt PSNR, l_∞ norm, and FID [13] to quantitatively evaluate the stealthiness of different backdoor watermarks. Peak Signal-to-Noise Ratio (PSNR) is calculated based on mean squared error and assesses the quality difference between the original and modified images. A higher PSNR value indicates better image quality and less distortion. The l_∞ norm, commonly used in image processing, measures the maximum pixel difference between two images. Fréchet Inception Distance (FID) evaluates the differences between the mean and covariance of the feature representations of modified and original images, typically extracted using a pre-trained Inception-v3 [34] model. FID better captures the structural information of images, providing a more accurate reflection of image quality. As shown in Tab. 13, EntropyMark outperforms all other methods across all metrics, demonstrating its superior stealthiness.

Metric $\rightarrow$	PSNR (dB)	l_∞ norm	FID
BadNets	26.7323	227.1596	2.6498
Blend	22.9924	62.1561	215.1331
EntropyMark	**38.9022**	**24.9548**	**2.0589**

Table 13. Stealthiness evaluation results.

H. Limitation

We propose a novel BW-DOV method that leverages higher prediction confidence as a signal for dataset usage detection. However, the current method has several limitations:

Over-reliance on verification samples: EntropyMark computes reference gradients using verification samples and employs them in gradient matching. Therefore, the verification effectiveness largely depends on the selection of verification samples.

Limited validation reliability: Compared to most dirty-label BW-DOV methods, EntropyMark exhibits a more narrow p-value gap between the 'Watermark' scenario and other scenarios, suggesting reduced verification reliability.

Only supports probability-based verification: The current method is limited to probability-based verification. En-

tropyMark, designed to be more harmless without affecting predictions, follows the same approach.

These issues will be addressed in future research, with a focus on improving verification reliability, reducing dependency on sample selection, and exploring label-only verification. These improvements aim to enhance the overall robustness and effectiveness of the method.

I. Social Impact

This paper addresses the challenge of protecting open-source datasets. While existing methods aim to mitigate the targeted nature of backdoors, they often compromise dataset functionality by altering the original prediction results of watermarked models. In contrast, EntropyMark offers a higher degree of harmlessness, maintaining dataset utility. In addition, EntropyMark provides high stealthiness and effectively bypasses most existing defenses, making it particularly well-suited for real-world protection scenarios.

Nevertheless, dataset stealers may develop sophisticated techniques to counteract our BW-DOV method. Future research should focus on monitoring emerging defensive strategies and refining our approach to ensure continued effectiveness in protecting open-source datasets.