Watermarking for Out-of-distribution Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Out-of-distribution (OOD) detection aims to identify OOD data based on representations extracted from well-trained deep models. However, existing methods largely ignore the *reprogramming* property of deep models and thus may not fully unleash their intrinsic strength: *without* modifying parameters of a well-trained deep model, we can reprogram this model for a new purpose via data-level manipulation (e.g., adding a specific feature perturbation). This property motivates us to reprogram a classification model to excel at OOD detection (a new task), and thus we propose a general methodology named *watermarking* in this paper. Specifically, we learn a unified pattern that is superimposed onto features of original data, and the model's detection capability is largely boosted after watermarking. Extensive experiments verify the effectiveness of watermarking, demonstrating the significance of the reprogramming property of deep models in OOD detection.

1 Introduction

Deep learning systems in the open world often encounter *out-of-distribution* (OOD) inputs whose label spaces are disjoint with that of training data, known as *in-distribution* (ID) data. For safety-critical applications, deep models should make reliable predictions for ID data, meanwhile detecting OOD data without prediction [2]. This leads to the OOD detection problem [28, 36, 40, 49], which has attracted intensive attention in reliable machine learning.

Identifying OOD data remains non-trivial since deep models can be overconfident with them [38]. As a promising technique, the *classification-based* OOD detection [57] relies on various scoring functions derived by the well-trained classification models, taking those inputs with small scores as OOD cases. In general, the scoring functions can be defined by logit outputs [18, 32], gradients [22], and embedding features [28, 43]. Without interfering with original models or requiring extra computation, they exploit the inherent capability of models learned from only ID sample. In general, these advantages can be critical in reality, where the cost of re-training is prohibitively high and the acquisition of real OOD data is largely limited [57].

Although promising progress has been achieved, previous works largely ignore the *reprogramming* property [10] of deep models: a well-trained model can be repurposed for a new task by a proper transformation of original inputs (e.g., a universal feature perturbation), without modifying any model parameter. For example, a model pre-trained on ImageNet [7] dataset can be reprogrammed for biomedical images [48]. In classification-based OOD detection, it indicates the possibility of making the target models adapt for effective detection given fixed methods and scoring strategies, motivating us the *first* attempt to investigate if the reprogramming property of deep models can help to address OOD detection, *i.e., can we reprogram well-trained deep models for OOD detection (a new task)?*

To this end, we propose a novel method, named **watermarking**, to *reprogram* the classification model for effective OOD detection. In our context, "reprogramming" is reflected by a *watermark* with its features on the shape of original inputs, which is defined as a static pattern that can be added for the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

test-time inputs (cf., Figure 1). After watermarking, the pre-defined scoring strategy is expected to be enhanced, with an improved gap of OOD scores between ID and OOD data (cf., Figure 2).

It is non-trivial to find the proper watermarks due to our lack of knowledge about unseen in advance. To address the issue, we propose a learning framework for effective watermarking. The insight is to make the model produce high scores for watermarked ID inputs meanwhile regularize the watermark such that the model will return low confidence without perceiving ID patterns. In this case, the model will have a relatively high score for a watermarked ID input, while the score remains low for an OOD case (cf., Figure 2). The reason is that the target model encounters a watermarked OOD input, without seeing any ID pattern.

Figure 1: Watermarking on CIFAR-10 [26] with free energy scoring [32]. The left figure is the learned watermark; the middle figure is an original input; the right figure is the watermarked result.

In our realization, we adopt several representative scoring strategies, devising specified learning objectives and proposing a reliable optimization algorithm. To understand our mechanism, Figure 1 depicts the watermark learned on CIFAR-10 [26] dataset, with the free energy scoring [32] (an advanced classification-based method). As we can see, the centre area of the learned watermark largely preserves the original input pattern, containing the semantic message that guides the detection primitively. By contrast, the edge area of the original input is superimposed by the specific pattern of the watermark, which may *encode* the knowledge once hidden by the model in boosting OOD detection. Overall, watermarking can preserve the meaningful pattern of original inputs in detection with further enhancement learned from the trained model. Therefore, the detection capability of the original model is largely improved after watermarking.

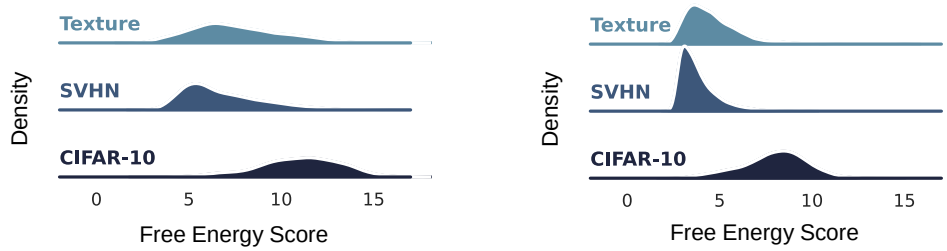
Figure 2 demonstrates the effect of our learned watermark, which is an example with the free-energy scoring. After watermarking, the scoring distributions are much concentrated, and the gap between ID (i.e., CIFAR-10) and OOD (i.e., SVHN [37] and Texture [5] datasets) cases is enlarged notably. We conduct extensive experiments for a wide range of OOD evaluation benchmarks in Section 6 and Appendix C, and the results verify the effectiveness of our proposal.

The success of watermarking takes roots in the following aspects: (1) a well-trained model on classification has the potential to be reprogrammed for OOD detection since they are two closely related tasks; (2) reprogramming has been widely studied, ranging from image classification to time series analysis [7, 48], making our proposal general across various domains; and (3) OOD detection suffers from the lack of knowledge about the real-world OOD distributions. Fortunately, with only data-level manipulation in low dimensions, watermarking can largely mitigate this issue of limited data. Overall, this data-level manipulation is orthogonal to existing methods, and thus provides a new road in OOD detection and can inspire more ways to design OOD detection methods in the future.

2 Related Works

To begin with, we briefly review the related works in OOD detection and model reprogramming. Please refer to Appendix A for the detailed discussion.

OOD Detection discerns ID and OOD data by their gaps regarding specified metrics/scores, and existing methods can mainly be attributed into three categories [57], namely, the *classification-based* methods, the *density-based* methods, and the *distance-based* methods. Specifically, the classification-based methods [18, 22, 32, 43] use representations extracted from the well-trained models in OOD scoring; and the distance-based methods [3, 21, 39, 59] measure the distance of inputs from class centers in the embedding space. Moreover, the density-based methods estimate input density with probabilistic models [28, 40, 44], identifying those OOD data with small likelihood values. Since distance-based and density-based methods may suffer from complexity in computation [28] and difficulty in optimization [61], recently, more researchers focus on developing classification-based methods and have made big progress on benchmark datasets [22, 32].



(a) before watermarking

(b) after watermarking

Figure 2: Experimental results before (a) /after (b) watermarking with CIFAR-10 being the ID dataset, SVHN and Texture being the OOD datasets. Data with large (small) OOD scores should be taken as ID (OOD) cases, and a larger distribution gap of scoring between ID and OOD cases ensures a better detection performance. After watermarking, the gap between ID and OOD cases is enlarged, demonstrating the improved capability of the original model in OOD detection. The horizontal axes are ignored for illustration, and please refer to Figure 4 for a formal manifestation.

Model Reprogramming repurposes the well-trained models for new tasks with only data-level manipulation [10], indicating that deep models are competent for different jobs without changing model parameters. In previous works, the data-level manipulation typically refers to a static padding pattern (different from our proposal) learned for the target task, which is added to the test-time data. The effectiveness of the model reprogramming is verified across image classification [10, 48] and time-series analysis [14, 56]. In this paper, we use the reprogramming property of deep models for effective OOD detection, which has been overseen previously.

3 Preliminary

Let $\mathcal{X} \subset \mathbb{R}^d$ be the input space and $\mathcal{Y} = \{1, \dots, c\}$ be the label space. We consider the ID distribution $D^{\text{ID}}_{\mathcal{X},\mathcal{Y}}$ defined over $\mathcal{X} \times \mathcal{Y}$, the training sample $S_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of size n independently drawn from $D^{\text{ID}}_{\mathcal{X},\mathcal{Y}}$, and a classification model $f : \mathcal{X} \to \mathbb{R}^c$ (with logit outputs) well-trained on S_n.

Based on the model $f(\cdot)$, the goal of the classification-based OOD detection is to design a detection model $g : \mathcal{X} \to \{0, 1\}$ that can distinguish test-time inputs given by the ID distribution $D^{\text{ID}}_{\mathcal{X}}$ from that of the OOD distribution $D^{\text{OOD}}_{\mathcal{X}}$. In general, $D^{\text{OOD}}_{\mathcal{X}}$ is defined as an irrelevant distribution of which the label set has no intersection with $\mathcal{Y}$, and thus should not be predicted by $f(\cdot)$. Overall, with 0 the OOD case and 1 the ID case, a detection model $g(\cdot)$ is defined as

$$g(\boldsymbol{x}; \tau) = \begin{cases} 1 & s(\boldsymbol{x}; f) \geq \tau \\ 0 & s(\boldsymbol{x}; f) < \tau \end{cases}, \tag{1}$$

where $\tau \in \mathbb{R}$ is a threshold and $s : \mathcal{X} \to \mathbb{R}$ is the scoring function defined by $f(\cdot)$, typically with its fixed parameters. Here, we focus on two representative methods in the classification-based OOD detection, namely, the *softmax scoring* and the *free energy scoring*.

Softmax Scoring Function [18] uses the maximum softmax prediction in OOD detection, of which the scoring function $s_{\text{SM}}(\cdot)$ is given by

$$s_{\text{SM}}(\boldsymbol{x}; f) = \max_k \ \texttt{softmax}_k \ f(\boldsymbol{x}), \tag{2}$$

where $\texttt{softmax}_k(\cdot)$ denotes the k-th element of the softmax outputs. In general, with a large (small) $s_{\text{SM}}(\boldsymbol{x}; f)$, the detection model will take the input $\boldsymbol{x}$ as an ID (OOD) case.

Free Energy Scoring Function [32] adopts the free energy function for scoring, defined by the logit outputs with a negative `logsumexp` operation, namely,

$$s_{\text{FE}}(\boldsymbol{x}; f) = \log \sum_k \exp f_k(\boldsymbol{x})/T, \tag{3}$$

where $T > 0$ is the temperature parameter fixed to 1 in our paper. It aligns with the density of inputs to some extent, and thus is less susceptible to the overconfidence issue mentioned above [32].

4 Watermarking Strategy

This section introduces the key concepts of watermarking for classification-based OOD detection.

Definition. A watermark $\boldsymbol{w} \in \mathbb{R}^d$ is a unified pattern with the exact shape as original inputs. It is added to test-time inputs statically, and we refer $\boldsymbol{w} + \boldsymbol{x}$ a *watermarked input* for $\forall \boldsymbol{x} \in \mathcal{X}$. In expectation, regarding the specified scoring function $s(\cdot)$, our watermarking should make the model excel at OOD detection for watermarked data.

Learning Strategy. Given the scoring function $s(\cdot)$, it is challenging to devise the exact watermark pattern by predefined rules. Therefore, for the proper watermarks in OOD detection, we need to devise learning objectives regarding watermarks, which consider both ID and OOD cases.

We generally have no information about the OOD distribution $D_{\mathcal{X},\mathcal{Y}}^{\text{OOD}}$, while we still want the model excels in discerning ID and OOD cases from scoring. For this challenge, we make the model produce high scores if watermarked ID data are observed; meanwhile, we regularize the watermark such that the model will return low scores when ID patterns do not exist. From the lens of our model, the scores should remain low if a watermarked OOD input is given since the watermark is not trained to perceive OOD data, of which the patterns are very different from the ID cases.

Benefits of Watermarking. Watermarking directly reprograms the model to make an adaptation to our specified task for scoring, such that the detection capability of the original model is largely improved. By contrast, previous methods typically adapt to their specified tasks by only the threshold τ as in Eq. (1). However, it requires the trade-off between false positive (ID) and false negative (OOD) rates when densities of scoring are non-separable (cf., Figure 2(a)).

Further, watermarking enjoys the benefits of previous classification-based methods in that we do not modify the original training procedure in classification, making our proposal easy to be deployed in real-world systems. Although the watermark also should be learned, the parameter space is in low dimension, and the learning procedure could be conducted *post-hoc* after the systems are deployed.

Comparison with Existing Works. In OOD detection, this paper is a first attempt in using the reprogramming property of deep models, leading to an effective learning framework named watermarking. At first glance, our methodology is seemingly similar to ODIN [31], which also conducts data-level perturbation for OOD detection. However, their instance-specified perturbation relies on extra backward-forward iterations during the test, which is not required in our method. Further, ODIN is designed for the softmax scoring, but our proposal is much general in OOD detection.

5 Realization of Watermarking Strategy

In this section, we discuss our learning framework of watermarking in detail.

Learning Objectives. As mentioned above, we need to consider the ID and OOD situations separately, with the associated objective functions denoted by $\ell^{\text{ID}}(\cdot)$ and $\ell^{\text{OOD}}(\cdot)$. For the ID case, the ID training sample is required, where we make the high scores for their watermarked counterparts. By contrast, since we typically lack knowledge about the test-time OOD data, only the watermark is used here, and we expect the model to produce the score as low as possible when perceiving the watermark.

Further, since only the watermark is adopted for training in the OOD case, the learned watermark is pretty sensitive regarding the detection model, i.e., the model may return different predictions when facing small perturbations. Thus, it hardly generalizes to real-world OOD cases, and the watermarked OOD inputs may not guarantee the low scores. To this end, the watermark is further perturbed during training. Here we adopt the Gaussian noise, leading to the perturbed watermark of the form $\boldsymbol{\epsilon} + \boldsymbol{w}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_1 \mathbf{I}_d)$ the (i.i.d.) Gaussian noise of d-dimension (the mean $\mathbf{0}$ and the standard deviation $\sigma_1 \mathbf{I}_d$). In summary, the overall risk is written as,

$$\mathcal{L}_n(\boldsymbol{w}) = \underbrace{\sum_n \ell^{\text{ID}}(\boldsymbol{x}_i + \boldsymbol{w}, y_i; f)}_{\mathcal{L}_n^{\text{ID}}(\boldsymbol{w})} + \beta \underbrace{\sum_n \ell^{\text{OOD}}(\boldsymbol{\epsilon}_j + \boldsymbol{w}; f)}_{\mathcal{L}_n^{\text{OOD}}(\boldsymbol{w})}, \tag{4}$$

with $\beta \geq 0$ the trade-off parameter, $\mathcal{L}_n^{\text{ID}}(\boldsymbol{w})$ the risk for ID data, and $\mathcal{L}_n^{\text{OOD}}(\boldsymbol{w})$ the risk for OOD data.

Optimization. To find the proper watermark, we use the first-order gradient update to iteratively update watermark's elements. However, data-level optimization remains difficult in deep learning, of which the results may get stuck at suboptimal points [51]. A common approach is to use the signum of first-order gradients, guiding the updating rule of the current watermark via

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \texttt{sign}(\nabla_{\boldsymbol{w}} \mathcal{L}_n(\boldsymbol{w})), \tag{5}$$

where $\texttt{sign}(\cdot)$ denotes the signum function and $\alpha > 0$ is the step size.

Further, for generality and insensibility, we prefer the solution that lies in the neighbourhood having uniformly low loss, i.e., with a smooth loss landscape [24]. Therefore, we adopt the *sharpness-aware minimization* (SAM) [11], an effective optimization framework in the seek of both the low loss value and the smooth loss landscape. Specifically, given the original risk $\mathcal{L}_n(\boldsymbol{w})$, the SAM problem is:

$$\mathcal{L}_n^{\text{SAM}}(w) = \max_{||\boldsymbol{\kappa}||_2 \leq \rho} \underbrace{[\mathcal{L}_n(\boldsymbol{w}+\boldsymbol{\kappa}) - \mathcal{L}_n(\boldsymbol{w})]}_{\text{sharpness}} + \mathcal{L}_n(\boldsymbol{w}) = \max_{||\boldsymbol{\kappa}||_2 \leq \rho} \mathcal{L}_n(\boldsymbol{w}+\boldsymbol{\kappa}) \tag{6}$$

where $\rho \geq 0$ is a constraint. For efficiency, the SAM makes the first-order Taylor expansion w.r.t. $\boldsymbol{\kappa}$ around $\mathbf{0}$, obtaining the approximated solution of the form [1]:

$$\boldsymbol{\kappa} = \rho \texttt{sign}(\nabla_{\boldsymbol{w}} \mathcal{L}_n(\boldsymbol{w})) \frac{|\nabla_{\boldsymbol{w}} \mathcal{L}_n(\boldsymbol{w})|^{q-1}}{(||\nabla_{\boldsymbol{w}} \mathcal{L}_n(\boldsymbol{w})||_q^q)^{1/p}}, \tag{7}$$

where $1/p + 1/q = 1$ and we set $p = q = 2$ for simplicity. Therefore, the final form of the SAM is written as $\mathcal{L}_n(\boldsymbol{w}+\boldsymbol{\kappa})$, with corresponding updating rule of

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \texttt{sign}(\nabla_{\boldsymbol{w}} \mathcal{L}_n(\boldsymbol{w}+\boldsymbol{\kappa})), \tag{8}$$

yielding an efficient optimization algorithm that induces the effective watermark.

The Overall Algorithm. In summary, we describe the overall learning framework. To begin with, the watermark is initialized by the (i.i.d.) Gaussian noise with the $\mathbf{0}$ mean and a small standard deviation $\sigma_2 \mathbf{I}_d$, and the learning procedure consists of three stages for each updating step:

- Negative sampling: a set of noise data ϵ is sampled, assuming be of the size m as that of the mini-batch regarding the ID sample;
- Risk calculating: the risk for ID and OOD cases are computed, and the overall risk is given by their sum with a trade-off parameter β as in Eq. (16);
- Watermark updating: the first-order gradient guides the pixel-level update of the watermark, using the signum of gradients and the SAM to make a reliable update as in Eq. (8).

The learned watermark is added to test-time inputs for OOD detection, and the detection model with the pre-defined scoring function is then deployed. Appendix B summarizes our learning framework of watermarking. Moreover, two specifications of watermarking are discussed in the following.

Two Realizations. Here, we focus on two representative methods in OOD detection, namely, the softmax scoring and the free energy scoring. For other representative methods in OOD detection, please refer to Appendix C for their descriptions and the experiments.

Softmax Scoring-based Watermarking. Following [19], we set $\ell_{\text{SM}}^{\text{ID}}(\cdot)$ to be the cross entropy loss and $\ell_{\text{SM}}^{\text{OOD}}(\cdot)$ to be the cross entropy regarding the uniform distribution, namely,

$$\ell_{\text{SM}}^{\text{ID}}(\boldsymbol{x}, y; f) = -\log \texttt{softmax}_y f(\boldsymbol{x}) \;\text{ and }\; \ell_{\text{SM}}^{\text{OOD}}(\boldsymbol{x}; f) = -\sum_k \frac{1}{c} \log \texttt{softmax}_k f(\boldsymbol{x}), \tag{9}$$

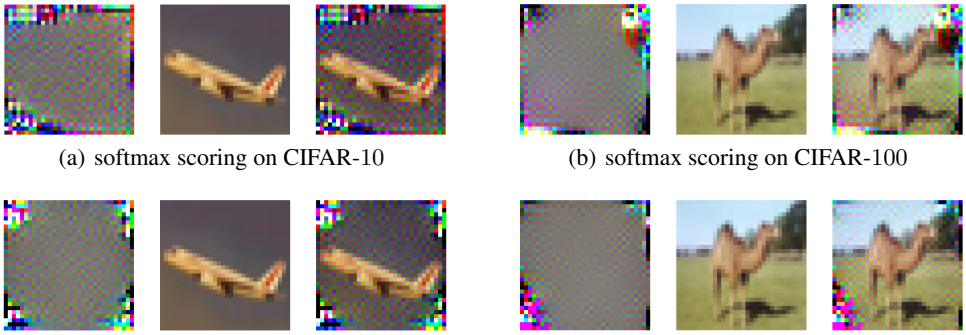
specifying the learning objectives in Eq. (16) for the softmax scoring-based watermarking.

Free-Energy Scoring-based Watermarking. [32] use a set of learning objectives for model re-training with free energy scoring. However, their `logsumexp` operation originating from the free energy function is difficult for optimization, posing notorious computing issues [34]. To this end, we drop the $\log$ operation and make the overall risk always positive by the following learning objectives:

$$\ell_{\text{FE}}^{\text{ID}}(\boldsymbol{x}; f) = \sum_k \exp -f_k(\boldsymbol{x})/T \;\text{ and }\; \ell_{\text{FE}}^{\text{OOD}}(\boldsymbol{x}; f) = \sum_k \exp f_k(\boldsymbol{x})/T, \tag{10}$$

realizing the learning objectives in Eq. (16) for the free-energy scoring.

[1] With an abuse of notation, we denote the estimated solution in the SAM as $\boldsymbol{\kappa}$ for simplicity.



(a) softmax scoring on CIFAR-10 (b) softmax scoring on CIFAR-100

(c) free energy scoring on CIFAR-10 (d) free energy scoring on CIFAR-100

Figure 3: The learned watermarks (left) and the example images with (middle) and without (right) the watermarks. All the pictures are clamped between 0 and 255 for the purpose of illustration.

6 Experiments

In this section, we conduct extensive experiments for watermarking in OOD detection. Specifically, we demonstrate the effectiveness of our method on a wide range of OOD evaluation benchmarks; we conduct ablation analysis for the important hyper-parameters in our learning framework; and we provide further experiments for an improved interpretation of our proposal.

Baselines results are achieved by our re-run of the publicly available codes. All the methods are realized by Pytorch 1.81 with CUDA 11.1, where we use several machines equipped with GeForce RTX 3090 GPUs and AMD Ryzen Threadripper 3960X Processors.

ID and OOD Datasets. We use CIFAR-10, CIFAR-100 [26], and ImageNet [41] datasets as three ID datasets, with data pre-processing including horizontal flip and normalization. Furthermore, for the OOD datasets, we adopt several commonly-used benchmarks, including Textures [5], SVHN [37], Places365 [62], LSUN [58], and iSUN [55].

Evaluation Metrics. The performance in OOD detection is measured via three representative metrics, which are all threshold-independent [6]: (1) the false positive rate of OOD sample when true positive rate of ID data is at 95% (FPR95); (2) the *area under the receiver operating characteristic curve* (AUROC), interpreted as the probability that an ID input has a greater score than an OOD one; and (3) the *area under the precision-recall curve* (AUPR), which further adjusts for different base rates.

Configuration. Following previous works [32], we employ WideResNet [60] (WRN-40-2) as the backbone model. For the CIFAR benchmarks, the models are trained for 200 epochs via the stochastic gradient descent, with the batch size 64, the momentum 0.9, and the initial learning rate 0.1. The learning rate is divided by 10 after 100 and 150 epochs. For the ImageNet Benchmark, the model is trained for 120 epochs via the stochastic gradient descent, with the batch size 32, the momentum 0.9 and the initial learning rate 0.05. The learning rate is divided by 10 after 60 and 90 epochs.

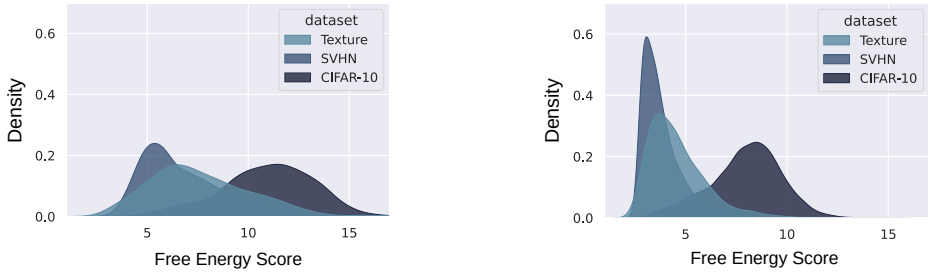
Hyper-parameters. For hyper-parameter setups in our experiments, we choose the proper σ_1 from the candidate parameter set $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, and the proper ρ from $\{0.0, 0.02, 0.05, 0.07, 0.1, 0.2, 0.5, 0.7, 1.0, 2.0, 5.0\}$. For the softmax scoring, β is chosen from $\{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$; and for the free energy scoring, β is chosen from $\{0.0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$. The above hyper-parameters are selected by random search (cf., Appendix C.7). We adopt validation OOD datasets that are separated from the original iSUN, Places365, Texture, SVHN, and LSUN datasets, and we choose the proper hyper-parameters from the candidates. On CIFAR benchmarks, our method is executed for 50 epochs. The initial learning rate $\alpha = 0.01$ divided by 10 after 25 epochs. For the softmax scoring, we set $\sigma_1 = 0.4$, $\rho = 1.0$, $\beta = 3.5$ in CIFAR-10 and $\sigma_1 = 1.0$, $\rho = 0.2$, $\beta = 2.5$ in CIFAR-100; for the free energy scoring, we set $\sigma_1 = 0.6$, $\rho = 0.7$, $\beta = 0.1$ in CIFAR-10 and $\sigma_1 = 1.0$, $\rho = 0.05$, $\beta = 1.2$ in CIFAR-100. On the ImageNet benchmark, our method is executed for 10 epochs and the initial learning rate $\alpha = 0.01$ is divided by 10 after 5 epochs. We set $\rho = 0.5$, $\sigma_1 = 0.2$, and $\beta = 1$ for both the softmax and the free energy scoring. Further, we fix $\sigma_2 = 0.001$ and $T = 1$.

Table 1: Softmax scoring with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**43.60** (55.55)	**93.53** (90.14)	**98.67** (97.84)
Places365	**60.75** (62.50)	**87.85** (87.41)	**96.98** (96.94)
Texture	**42.00** (59.30)	**92.83** (88.37)	**98.43** (97.14)
SVHN	**27.25** (49.10)	**96.00** (91.69)	**99.17** (96.54)
LSUN	**40.70** (52.05)	**94.36** (91.50)	**98.86** (98.16)
average	**42.86** (55.70)	**92.91** (89.82)	**98.42** (97.32)
CIFAR-100			
iSUN	**77.85** (83.35)	**79.91** (75.28)	**95.35** (94.00)
Places365	83.25 (**82.20**)	74.28 (**74.40**)	**93.47** (93.44)
Texture	**79.10** (83.80)	**77.14** (72.83)	**94.26** (92.81)
SVHN	**82.95** (85.05)	**76.92** (70.64)	**94.72** (92.61)
LSUN	**76.75** (80.45)	**79.60** (76.25)	**95.27** (94.32)
average	**79.98** (82.97)	**77.57** (73.88)	**94.61** (93.43)

Table 2: Free energy scoring with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**16.30** (32.10)	**96.97** (92.84)	**99.39** (98.33)
Places365	**36.25** (41.45)	**91.87** (89.65)	**97.94** (97.21)
Texture	**32.60** (52.05)	**93.14** (85.43)	**98.08** (95.52)
SVHN	**16.45** (35.25)	**97.11** (90.91)	**99.39** (97.68)
LSUN	**16.85** (27.50)	**96.97** (93.98)	**99.38** (98.59)
average	**23.69** (37.67)	**95.21** (90.56)	**98.83** (97.46)
CIFAR-100			
iSUN	**75.05** (81.80)	**83.07** (79.04)	**96.15** (94.98)
Places365	**80.45** (80.50)	**77.78** (74.99)	**94.45** (93.37)
Texture	**75.15** (80.20)	**79.55** (76.00)	**94.79** (93.53)
SVHN	**82.85** (85.10)	**75.26** (74.20)	**94.18** (93.70)
LSUN	**71.85** (80.45)	**84.01** (78.29)	**96.33** (94.69)
average	**77.07** (81.61)	**79.93** (76.50)	**95.18** (94.05)



(a) w/o watermarking (b) w/ watermarking

Figure 4: An illustration on CIFAR-10 dataset regarding the free energy scoring. (a) depicts the scoring distributions before watermarking and (b) is the scoring distributions after watermarking.

CIFAR Benchmarks. We depict the learned watermarks in Figure 3. As we can see, the centre areas maintain the pattern of original inputs, which are helpful in OOD detection primitively. By contrast, the edge areas of the watermarks distort the original features, superimposed with the pattern that may further boost the capability of the original models in OOD detection.

Then, we demonstrate the improvement of watermarking on CIFAR-10 and CIFAR-100 datasets. For the results of the softmax scoring in Table 1, our watermarking reduces the average FPR95 by $2.99 \sim 12.84$, boosts the average AUROC by $3.09 \sim 3.69$ and the average AUPR by $1.10 \sim 1.18$. Moreover, for each of the considered test-time OOD dataset, there also exist stable improvements after watermarking, except for the comparable results regarding Places365 on CIFAR-100 dataset.

For the results of the free energy scoring in Table 2, the improvement via watermarking is also substantial, with $4.54 \sim 13.98$, $3.43 \sim 4.65$, and $1.13 \sim 1.37$ better performance regarding the average FPR95, AUROC, and AUPR. Overall, Table 1 and Table 2 not only justify the effectiveness of watermarking and also demonstrate the generality of the proposed watermarking. Further, comparing between CIFAR-10 and CIFAR-100 datasets, the improvements of watermarking on CIFAR-10 is much greater than that of CIFAR-100, aligning with the previous observations [23] that a large semantic space can exaggerate the challenge in effective OOD detection.

Figure 4 illustrates the scoring distributions before (a) and after (b) watermarking on CIFAR-10, where we take the free energy scoring as an example. Due to the space limit, we only consider two test-time OOD datasets, namely, Texture and SVHN. As we can see, after watermarking, the distribution gap between the ID (i.e., CIFAR-10) and the OOD (i.e., Texture and SVHN) cases is enlarged, and thus the detection capability is improved.

Table 3: The softmax scoring with/without watermarking on ImageNet. The notion ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
iSUN	**11.54** (52.45)	**97.41** (92.52)	**99.45** (98.70)
Places365	**70.59** (73.25)	**82.03** (80.78)	**95.62** (94.58)
Texture	**61.20** (67.18)	**84.00** (82.27)	**98.60** (97.85)
SVHN	44.58 (**28.49**)	93.56 (**95.60**)	98.70 (**99.00**)
LSUN	**11.84** (54.62)	**97.97** (91.52)	**99.57** (98.25)
average	**40.50** (54.93)	**91.22** (88.57)	**98.42** (97.69)

Table 4: The free-energy scoring with/without watermarking on ImageNet. The notion ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
iSUN	**32.83** (45.40)	**94.35** (94.00)	**98.89** (98.20)
Places365	**71.85** (75.01)	**79.85** (78.54)	**94.65** (94.40)
Texture	**67.75** (68.77)	**80.80** (80.22)	**97.00** (96.51)
SVHN	**12.85** (27.60)	**97.68** (95.17)	**99.45** (99.00)
LSUN	**33.75** (46.47)	**93.71** (90.59)	**98.80** (97.94)
average	**43.23** (52.73)	**89.10** (86.14)	**97.73** (97.15)

ImageNet Benchmark. Huang and Li [23] show that many advanced methods developed on the CIFAR benchmarks can hardly work for the ImageNet dataset due to its large semantic space with 1k classes. In order to verify the power of watermarking in large semantic space, we conduct experiments with ImageNet being the ID dataset, and the results regarding the softmax scoring and the free-energy scoring are summarized in Table 3 and Table 4. For the softmax scoring, we decrease the average FPR95 from 54.93 to 40.50 after watermarking; for the free-energy scoring, we reduce the average FPR95 from 52.73 to 43.23. It demonstrates that our watermarking still works well in the case of the large semantic space, with considerable improvements in OOD detection power.

Near OOD Detection. The above experiments focus on the far OOD detection setups where ID and OOD data are different regarding semantics and stylish. Now, we further demonstrate the power of our watermarking strategy in a near OOD situation [53], covering a challenging situation where ID and OOD data are stylistically similar (i.e., near-OOD data). Besides the common learning setup in Section 5, we further consider the use of shifting augmentations [46], which are data augmentations that are harmful to the standard contrastive learning but can be used to construct near-OOD data similar to ID data. We consider two representative shifting augmentations: "perm" (permute evenly partitioned data) and "rotate" (rotate 90 degrees of original data). The shifting-augmented ID data are taken as OOD data fed into $\ell^{\text{OOD}}(\cdot)$ along with random Gaussian noise.

Table 5: The performance of our watermarking on near-OOD detection regarding the softmax scoring and the free energy scoring.

	FPR95	AUROC	AUPR
softmax scoring			
w/o watermark	90.10	55.47	86.16
common	88.25	53.16	84.75
permute	86.45	60.04	86.33
rotate	**81.50**	**65.69**	**88.67**
free-energy scoring			
w/o watermark	52.25	86.49	96.44
common	49.75	88.52	96.98
permute	48.55	88.40	97.03
rotate	**47.85**	**88.90**	**97.08**

The experimental results are summarized in Table 5, where we take CIFAR-10 as ID data and CIFAR-100 as OOD data. Here, the common watermark learning setup (common) can already lead to improved performance in near-OOD detection compared to the cases without watermarking (w/o watermark). Moreover, watermarking with shifting augmentations (perm and rotate) can further boost the detection power of the models, leading to at most 8.60 and 4.70 improvements in FPR95 for the softmax and the free energy scoring.

Effect of Hyper-parameters. To further interpret our learning framework, we compare the performance of the resultant watermarks regarding various setups of hyper-parameters, focusing on the standard deviation σ_1 in the Gaussian noise and the perturbation constraint ρ in the SAM, which are both critical for our proposal. As a case study, we conduct experiments regarding the softmax scoring on CIFAR-10. For detailed results about the ablation study, please refer to Appendix C.

Table 6 lists the results with various values of σ_1, ranging from 0.00 to 1.00. Note that σ_1 is the standard deviation of the Gaussian noise added to the watermark, with different values indicating various degrees of the perturbation. As we can see, a mild perturbation of the watermark (e.g., $\sigma_1 = 0.4$) can truly lead to improved results in detection, with 10.05, 1.60, and 0.34 improvements regarding the average FPR95, AUROC, and AUPR. It indicates that the learned watermark is sensitive when facing small perturbations, if the Gaussian noise is not applied during training. However, some extreme values (e.g., $\sigma_1 = 1$) may overwhelm the watermark pattern and thus be detrimental.

Table 6: The average performance of the softmax scoring on CIFAR-10 dataset with various values of the parameter σ_1. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

σ_1	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
2.00	42.11	92.01	98.24
1.80	42.84	91.89	98.22
1.60	41.41	92.14	98.25
1.40	42.20	91.82	98.20
1.20	41.98	91.91	98.20
1.00	42.76	91.95	98.23
0.80	43.38	91.89	98.21
0.60	39.16	92.89	98.41
0.40	**38.66**	**93.03**	**98.45**
0.20	43.88	92.80	98.42
0.00	48.71	91.43	98.11

Table 7: The average performance of the softmax scoring on CIFAR-10 dataset with various values of the parameter ρ. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

ρ	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
5.00	60.02	87.36	97.15
2.00	46.24	91.61	98.14
1.00	**39.12**	**92.96**	**98.42**
0.70	41.07	92.77	98.40
0.50	43.55	92.38	98.34
0.20	42.02	92.68	98.38
0.10	41.99	92.77	98.41
0.07	42.13	92.79	98.42
0.05	42.06	92.84	98.42
0.02	44.35	92.34	98.32
0.00	43.04	92.44	98.32

In Table 7, we summarize the experimental results given by various values of ρ for the SAM. Overall, a large value of ρ indicates that a wide range in the solution's neighbours should be smooth, and thus the stability of the result is expected to be improved. However, such a solution may consume too much capacity of the watermark, misleading the learning procedure to some unsatisfactory results. On the other side, our watermarking truly benefits from the SAM with a mild choice of the hyper-parameter. Specifically, comparing with the results without the SAM (i.e., $\rho = 0.00$), the detection capability of the watermark with a suitable ρ (i.e., $\rho = 1.0$) is largely improved, with 3.92, 0.52, and 0.10 better results regarding the average FPR95, AUROC, and AUPR.

Transferability of watermarking. Further, we explore the transferability of the watermarking strategy learned with different scoring functions, e.g., we study the effect of the watermark learned with the softmax scoring when it is deployed with the free energy scoring. The results are summarized in Table 8. Note that, a "learn" with FE and "score" with SM indicates that the watermark is learned with the free-energy scoring and tested regarding the softmax scoring. As we can see, the watermarks learned with the free energy scoring can be reused for the softmax scoring (even with better results), while the reverse typically leads a deterioration. It may indicate that our learning objectives in Eq. (10) are general and effective for both the softmax scoring and the free energy scoring. However, we do not observe any transferability between different datasets. For example, for the softmax scoring, when the watermark learned on CIFAR-10 is adopted for CIFAR-100, there is a drop of performance with 51.93 (from 42.86 to 94.79), 35.29 (from 92.91 to 57.62) and 12.00 (98.42 to 86.42) regarding the average FPR95, AUROC, and AUPR.

Table 8: Transferability of watermarking across scoring functions. SM denotes softmax scoring and FE denotes free energy scoring.

learn	score	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
CIFAR-10				
SM	SM	42.86	92.91	98.42
FE	SM	**40.19**	**94.83**	**98.99**
FE	FE	**23.69**	**95.21**	**98.83**
SM	FE	28.22	94.70	98.81
CIFAR-100				
SM	SM	79.98	**77.57**	**94.61**
FE	SM	**77.07**	76.27	94.02
FE	FE	**77.07**	**79.93**	**95.18**
SM	FE	78.48	79.24	94.56

7 Conclusion

This paper demonstrates that the model's inherent capability in OOD detection can be largely improved with only data-level manipulation, where we propose a general and effective methodology named watermarking. Overall, we learn a discriminative pattern that could be superimposed onto original inputs, such that the OOD scores become much separable between ID and OOD situations. Our results indicate a promising direction in OOD detection that warrants our exploration in the future. Further investigation should focus on the stability in optimization and general learning objectives regarding advanced OOD detection methods.

References

[1] J. Bitterwolf, A. Meinke, and M. Hein. Certifiably adversarially robust detection of out-of-distribution data. In *NeurIPS*, 2020.

[2] S. Bulusu, B. Kailkhura, B. Li, P. Varshney, and D. Song. Anomalous instance detection in deep learning: A survey. Technical report, Lawrence Livermore National Lab., 2020.

[3] X. Chen, X. Lan, F. Sun, and N. Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *ECCV*, 2020.

[4] S. Choi and S. Chung. Novelty detection via blurring. In *ICLR*, 2020.

[5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[6] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *ICML*, 2006.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[10] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein. Adversarial reprogramming of neural networks. In *ICLR*, 2019.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[13] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[14] K. Hambardzumyan, H. Khachatrian, and J. May. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*, 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 2019.

[17] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, and J. Steinhardt. Improving and assessing anomaly detectors for large-scale settings. 2021.

[18] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

[19] D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

[20] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

[21] H. Huang, Z. Li, L. Wang, S. Chen, B. Dong, and X. Zhou. Feature space singularity for out-of-distribution detection. *arXiv preprint arXiv:2011.14654*, 2020.

[22] R. Huang, A. Geng, and Y. Li. On the importance of gradients for detecting distributional shifts in the wild. In *NeurIPS*, 2021.

[23] R. Huang and Y. Li. MOS: towards scaling out-of-distribution detection for large semantic space. In *CVPR*, 2021.

[24] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[25] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. In *NeurIPS*, 2020.

[26] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Technical Report TR-2009, University of Toronto*, 2009.

[27] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[28] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[29] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021.

[30] Y. Li, B. Wu, Y. Jiang, Z. Li, and S.-T. Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020.

[31] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[32] W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

[33] Y. Liu, X. Ma, J. Bailey, and F. Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020.

[34] T. P. Mann. Numerically stable hidden markov model implementation. *An HMM Scaling Tutorial*, pages 1–8, 2006.

[35] P. Morteza and Y. Li. Provable guarantees for understanding out-of-distribution detection. In *AAAI*, 2022.

[36] E. T. Nalisnick, A. Matsukawa, Y. W. Teh, D. Görür, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019.

[37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[38] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

[39] J. Ren, S. Fort, J. Liu, A. G. Roy, S. Padhy, and B. Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

[40] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *NeurIPS*, 2019.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[42] A. Saha, A. Subramanya, and H. Pirsiavash. Hidden trigger backdoor attacks. In *AAAI*, 2020.

[43] C. S. Sastry and S. Oore. Detecting out-of-distribution examples with gram matrices. In *ICML*, 2020.

[44] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *ICLR*, 2020.

[45] Y. Sun, C. Guo, and Y. Li. React: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021.

[46] J. Tack, S. Mo, J. Jeong, and J. Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 33:11839–11852, 2020.

[47] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, 2019.

[48] Y. Tsai, P. Chen, and T. Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *ICML*, 2020.

[49] S. Vernekar, A. Gaurav, V. Abdelzad, T. Denouden, R. Salay, and K. Czarnecki. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*, 2019.

[50] H. Wang, W. Liu, A. Bocchieri, and Y. Li. Can multi-label classification networks know what they don't know? In *NeurIPS*, 2021.

[51] Q. Wang, F. Liu, B. Han, T. Liu, C. Gong, G. Niu, M. Zhou, and M. Sugiyama. Probabilistic margins for instance reweighting in adversarial training. In *NeurIPS*, 2021.

[52] Y. Wang, B. Li, T. Che, K. Zhou, Z. Liu, and D. Li. Energy-based open-world uncertainty modeling for confidence calibration. In *ICCV*, 2021.

[53] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[54] Z. Xiao, Q. Yan, and Y. Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In *NeurIPS*, 2020.

[55] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[56] C. H. Yang, Y. Tsai, and P. Chen. Voice2series: Reprogramming acoustic models for time series classification. In *ICML*, 2021.

[57] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[58] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[59] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In *CVPR*, 2021.

[60] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.

[61] L. H. Zhang, M. Goldstein, and R. Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *ICML*, 2021.

[62] B. Zhou, À. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We use only standard datasets
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use only standard datasets
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Additional Related Works

We review the recent studies in OOD detection, model reprogramming, and backdoor attack.

A.1 OOD Detection

Following [57], we attribute existing works into three categories, namely, the *classification-based* methods, the *density-based* methods, and the *distance-based* methods. In general, these methods aim to maximize the gap between ID and OOD data regarding specified metrics in identifying OOD cases.

The classification-based methods use the representations extracted from the well-trained classification models in OOD scoring. For example, [18, 31, 32, 35, 45, 50] employ logit outputs from models in estimating the confidence of ID cases; [28, 43] adopt Mahalanobis distance and Gram Matrix to exploit models' detection capability from embedding features; [22, 31] further demonstrate the importance of gradient information, either perturbing inputs with its gradients or directly using the gradient norm in scoring. The classification-based methods are easy to be deployed without modifying the models [57], and thus it is the main focus in this paper.

The distance-based methods measure the distance regarding the embedding space, taking those data far away from the class prototypes as the OOD cases. Representative works adopt the Mahalanobis distance [22, 28, 39], the cosine similarity [3, 59], and the Euclidean distance [21]. Our methods can also be used in the distance-based methods. However, extra computation, such as calculating the precision matrix [28] and background splitting [39], may lead difficulty in devising proper learning objectives, out of the scope of our paper.

The density-based methods explicitly estimate the density of ID samples with various probabilistic models, identifying those OOD data based on the likelihood [28], the likelihood ratio [40, 44], and the likelihood regret [54]. Typically, the input density is modelled by the mixture of Gaussian models [28] and the flow-based methods [25, 36]. Although the density-based methods can directly characterize the properties of ID density, these methods are difficult to be trained and may make overconfident predictions, as demonstrated in previous works [35, 36].

Recent works also focus on the causes of key challenges in OOD detection, from the lens of the BatchNorm statistics [45], the density estimation [35], and the spurious correlation [43]. Improved methods, related to specified model architectures [8, 52], data perturbation [1, 4, 31], data augmentation [16, 20, 47], and outlier exposure [19], are also well-studied. However, these methods typically overlook the reprogramming property of deep models, which remain orthogonal to our proposal.

A.2 Reprogramming Property

The seminal work [10] introduces adversarial reprogramming as an attack method in the literature of adversarial learning [12], adversarially reprogramming the target model to perform a new task without changing the original model. The term "attack" lies in the fact that, by reprogramming, an attacker can easily steal public machine learning services, abusing their computational resources for tasks that violate their original purposes. Overall, [10] claim the reprogramming property of deep models—without modifying parameters of a well-trained model, we can reprogram it for a new purpose with only data-level manipulation. The reprogramming property of deep learning is preliminarily verified for various tasks, and its further applications are not limited to adversarial learning. Actually, advanced works often take reprogramming as an effective transfer learning technique in cases with limited data and resources. In the context of the image classification, [10] reprogram a model trained on ImageNet in solving vision-based counting tasks; and [48] further consider reprogramming a black-box system for biomedical image classification, which suffers from label scarcity issue. In the context of the natural language processing, well-trained models are reprogrammed for time-series classification [56] and sentiment analysis [14], where data scarcity issues frequently occur.

In this paper, we also employ the reprogramming property of deep models for transfer learning. However, instead of reprogramming across different datasets, we reprogram our original classification model for the task in OOD detection, considering the situation with the same (ID) dataset before/after reprogramming. Further, to preserve the benefits of previous classification-based detection methods,

Algorithm 1 Watermarking — the learning framework.

1: **Inputs:** trained model $f(\cdot)$ and ID training set S_n;
2: $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \sigma_2 \mathbf{I}_d)$; ▷ watermark initialization
3: **for** $t = 1$ **to** `num_step` **do**
4: mini-batch $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ and noise $\{\boldsymbol{\epsilon}_i\}_{i=1}^m$; ▷ data sampling
5: $\mathcal{L}_m(\boldsymbol{w}) = \mathcal{L}_m^{\text{ID}}(\boldsymbol{w}) + \beta \mathcal{L}_m^{\text{OOD}}(\boldsymbol{w})$; ▷ risk calculation
6: $\nabla_{\boldsymbol{w}} \mathcal{L}_m(\boldsymbol{w})$ and $\boldsymbol{\kappa}$; ▷ gradient calculation
7: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \nabla_{\boldsymbol{w}} \mathcal{L}_m(\boldsymbol{w})|_{\boldsymbol{w}+\boldsymbol{\kappa}}$; ▷ watermark updating
8: **Output:** learned watermark $\boldsymbol{w}$.

Table 9: Test accuracy before/after watermarking. w/o denotes the benchmark without watermarking, SE (FE) denotes the softmax (free energy) scoring with watermarking.

dataset	w/o	SM	FE	dataset	w/o	SM	FE
CIFAR-10	**94.84%**	91.85%	93.49%	CIFAR-100	**75.96%**	72.03%	74.08%

we adopt the perturbing pattern (i.e., the watermark) on the same shape as the original inputs, instead of reshaping original inputs and adding padding features as previous works [10, 56].

A.3 Backdoor Attack

The model reprogramming is related to the *backdoor attack*, which also change models' behaviour during the test. Overall, when a backdoor is *embedded* during training and the *trigger* is activated during the test, model predictions will be modified to the *attacker-specified* labels deliberately [13]. Nowadays, the *data poisoning* [13, 29, 33, 42] is among the best to realize the backdoor attack regarding deep models—a portion of the training sample is modified with the attacker-specified pattern (i.e., pre-defined trigger) and the attacker-specified labels. The target models are trained on these poisoned data, and the resultant models will suffer from the backdoor attack when the trigger is activated. Please refer to [30] for a comprehensive survey.

However, the backdoor attack and the model reprogramming exploit different aspects of deep models. In general, the backdoor attack utilises the *excessive* learning ability in memorizing noise features [30], while the reprogramming property states that the well-trained models can be reprogrammed for new tasks without modifying the original models.

B The Overall Algorithm

The overall learning framework is summarized in Algorithm 1, optimizing in a stochastic manner with `num_step` iterations. The watermark is initialized by the Gaussian noise with the $\mathbf{0}$ mean and a small standard deviation $\sigma_2 \mathbf{I}_d$ (Step 2), and the learning procedure consists of three stages for each update: (1) a set of Gaussian noise data is sampled, assuming be of the size m as that of the mini-batch regarding the ID sample (Step 4); (2) the risk for ID and OOD cases are computed and the overall risk is given by their sum with a trade-off parameter β (Step 5); (3) the first-order gradient guides the pixel-level update of the watermark, using the signum of gradients and the SAM to make a reliable update (Step 7). After watermark training, the learned watermark is added to test-time inputs for OOD detection and the detection model with the pre-defined scoring function is deployed.

C Further Experiments

This section conducts further experiments about our proposal.

C.1 Impact on Test Accuracy

To begin with, we study the impact of watermarking on the classification accuracy in Table 9, comparing with the results without watermarking. As we can see, watermarking has a negative impact on the test accuracy, dropping from 94.84% to 91.85% and 93.49% on CIFAR-10; and from 75.96%

Table 10: OE with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**2.70** (2.75)	**99.54** (99.55)	**99.91** (99.90)
Places365	**35.05** (37.75)	**94.22** (93.12)	**98.70** (98.39)
Texture	**25.80** (27.95)	**95.58** (95.30)	**98.99** (98.92)
SVHN	**30.25** (35.85)	**95.40** (94.31)	**99.83** (99.04)
LSUN	1.50 (**0.50**)	99.71 (**99.81**)	**99.94** (99.94)
average	**19.06** (20.96)	**96.89** (96.41)	**99.39** (99.31)
CIFAR-100			
iSUN	**22.85** (40.55)	**95.27** (90.60)	**98.92** (97.84)
Places365	**71.75** (73.75)	**78.60** (77.89)	**94.66** (94.48)
Texture	**66.70** (68.20)	81.30 (**81.50**)	95.44 (**95.49**)
SVHN	89.95 (**84.80**)	68.20 (**72.40**)	92.24 (**93.45**)
LSUN	**19.75** (34.20)	**96.17** (92.56)	**99.15** (98.32)
average	**54.20** (60.30)	**83.90** (82.99)	**96.08** (95.91)

Table 11: MaxLogit with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**24.40** (34.90)	**96.07** (92.54)	**99.21** (98.28)
Places365	50.60 (**43.70**)	89.03 (**89.41**)	97.28 (**96.81**)
Texture	**31.35** (51.10)	**93.72** (85.88)	**98.36** (95.72)
SVHN	**18.80** (35.70)	**96.81** (91.25)	**99.33** (97.75)
LSUN	**22.65** (29.10)	**96.28** (93.73)	**99.23** (98.63)
average	**29.56** (38.90)	**94.38** (90.40)	**98.68** (97.91)
CIFAR-100			
iSUN	**76.20** (78.45)	**83.18** (79.50)	**96.23** (95.06)
Places365	**79.20** (80.55)	**77.19** (75.24)	**94.20** (93.42)
Texture	**67.75** (78.00)	**83.89** (77.15)	**96.14** (93.89)
SVHN	**81.10** (84.00)	**80.75** (73.66)	**95.79** (93.53)
LSUN	**72.60** (78.35)	**84.20** (78.95)	**96.49** (94.84)
average	**75.37** (79.87)	**81.85** (76.90)	**95.77** (94.15)

to 72.03% and 74.08% on CIFAR-100. Further, after watermarking, the classification accuracy with the free energy scoring is much better than that of the softmax scoring, with only 1.35% to 1.88% decrease in classification accuracy. Therefore, we suggest using the free energy scoring in watermarking as a default setup, which leads to better detection capability and largely preserves the original capability in classification.

C.2 Other Scoring Strategies with Watermarking

Note that watermarking is orthogonal to much of the existing methods and the watermarking strategy can boost many other advanced OOD detection methods. To further verify the generality and the effectiveness of our proposal, we utilize watermarking for three representative OOD detection methods, namely, OE [19], MaxLogit [17], ODIN [31], and ReAct [45].

For OE, Hendrycks et al. [19] state that the target model can benefit from fine-tuning with extra OOD training data, in the context of the softmax scoring. In general, OE requires to re-train the target model, which will be prohibitively expensive for many real-world applications. However, since the model has seen some kinds of OOD data during training, it typically reveals superior results than many other advanced detection methods.

We are interested in whether our watermarking can improve the detection capability of the models that have been fine-tuned with OE. Typically, we assume that the training-time OOD data are different from that of the test time. Therefore, following previous works [19, 32], we adopt the tiny-ImageNet [27] as the training-time OOD data for model fine-tuning. The learning objectives regarding the model parameters in OE is similar to Eq. (9) and we follow the default hyper-parameter setups as in [19]. We learn the watermarks for the fixed OE-trained models with the softmax scoring, and the results on CIFAR benchmarks are summarized in Table 10. Overall, the experimental results suggest that our watermarking can still benefit OE in effective OOD detection, which may require our in-depth exploration in the future.

MaxLogit, ODIN, and ReAct can be viewed as the improved versions of the softmax scoring in Eq. (2). Specifically, MaxLogit takes the maximal logit outputs in OOD scoring, which is better than softmax scoring when facing large-class setting; ODIN clamps embedding features from the second-last layer of model outputs, which can attenuate the overconfidence issue caused by the out-sized activation of abnormal hidden units; and ReAct observes that temperature scaling and adversarial feature perturbation can improve model capability in discerning OOD data from ID cases.

For MaxLogit, we directly use the learning objective of the softmax scoring-based watermarking, with the corresponding scoring function of the form:

$$s_{\text{MaxLogit}}(\boldsymbol{x}; f) = \max_k \; f_k(\boldsymbol{x}), \tag{11}$$

Table 12: ReAct scoring with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**27.90** (63.65)	**95.73** (87.54)	**99.13** (97.17)
Places365	**62.55** (62.65)	**85.90** (86.98)	**96.50** (96.74)
Texture	**39.85** (58.90)	**93.68** (87.32)	**98.63** (96.86)
SVHN	**40.40** (43.35)	**93.47** (93.27)	**98.61** (98.60)
LSUN	**23.35** (59.40)	**96.29** (88.85)	**9.24** (97.51)
average	**38.91** (57.59)	**93.01** (88.79)	**80.42** (97.37)
CIFAR-100			
iSUN	**68.05** (86.40)	**83.91** (75.31)	**96.35** (94.04)
Places365	**82.65** (87.70)	**73.18** (71.20)	**93.00** (92.50)
Texture	**74.65** (86.35)	**78.35** (71.33)	**94.72** (92.39)
SVHN	**85.95** (77.50)	**75.47** (72.79)	**94.34** (92.33)
LSUN	**66.95** (86.85)	**84.02** (74.71)	**96.39** (93.90)
average	**75.65** (84.96)	**78.98** (73.06)	**94.96** (93.03)

Table 13: ODIN scoring with/without watermarking on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**25.05** (35.15)	**96.21** (93.09)	**99.23** (98.44)
Places365	**59.60** (55.95)	**87.74** (85.84)	**97.07** (96.22)
Texture	**36.35** (49.50)	**93.83** (86.72)	**98.64** (96.18)
SVHN	**40.55** (43.20)	**93.29** (91.34)	**98.56** (97.95)
LSUN	**23.75** (29.40)	**96.45** (94.06)	**99.27** (98.64)
average	**37.06** (42.64)	**93.50** (90.21)	**98.55** (97.49)
CIFAR-100			
iSUN	**69.60** (70.80)	**83.70** (81.32)	**96.32** (95.39)
Places365	**82.10** (88.50)	**74.65** (72.07)	**93.64** (92.74)
Texture	**75.95** (82.40)	**78.80** (71.87)	**94.86** (92.50)
SVHN	**86.35** (94.65)	**76.04** (59.40)	**94.52** (89.21)
LSUN	**67.80** (71.20)	**84.61** (81.18)	**96.60** (95.47)
average	**76.36** (81.51)	**79.56** (73.17)	**95.18** (93.06)

which directly use the logit outputs (instead of softmax outputs) in discerning ID and OOD data.

The scoring function for ODIN is

$$s_{\text{ODIN}}(\boldsymbol{x}; f) = \max_k \ \texttt{softmax}_k \ f(\tilde{\boldsymbol{x}})/T, \tag{12}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \xi \texttt{sign}(-\nabla_{\boldsymbol{x}} \log \texttt{softmax}_y f(\boldsymbol{x}))$ is the perturbed data point and ξ is the perturbation magnitude. For the watermark training, the learning objectives are of the form:

$$\ell_{\text{SM}}^{\text{ID}}(\boldsymbol{x}, y; f) = -\log \texttt{softmax}_y f(\tilde{\boldsymbol{x}}) \ \text{ and } \ \ell_{\text{SM}}^{\text{OOD}}(\boldsymbol{x}; f) = -\sum_k \frac{1}{c} \log \texttt{softmax}_k f(\tilde{\boldsymbol{x}}), \tag{13}$$

following the same definition of $\tilde{\boldsymbol{x}}$ as in Eq. (12).

For ReAct, we assume the feature extractor defined by the second-last of model outputs by $f_{\texttt{FEA}}$ and the above classifier by $f_{\texttt{CLA}}$, i.e., $f(\boldsymbol{x}) = f_{\texttt{CLA}}(f_{\texttt{FEA}}(x))$. Then, the rectified model output is

$$f_{\texttt{ReAct}}(\boldsymbol{x}) = f_{\texttt{CLA}}(\min(f_{\texttt{FEA}}(\boldsymbol{x}), \tau)), \tag{14}$$

truncating values of hidden units from the second-last layer that are above τ. The corresponding learning objectives in watermark training can be written as

$$\ell_{\text{SM}}^{\text{ID}}(\boldsymbol{x}, y; f) = -\log \texttt{softmax}_y f_{\texttt{ReAct}}(\boldsymbol{x}) \ \text{ and } \ \ell_{\text{SM}}^{\text{OOD}}(\boldsymbol{x}; f) = -\sum_k \frac{1}{c} \log \texttt{softmax}_k f_{\texttt{ReAct}}(\boldsymbol{x}), \tag{15}$$

which is similar to that of the softmax scoring in Eq. (9).

Tables 11 to 13 summarize the experimental results on CIFAR-10 and CIFAR-100 datasets, where T is fixed to 1000, ξ is set to 0.0014, and τ is chosen such that 10% of the activation values are clamped, following the setups of the original papers. The performance improvements are illustrious regarding the performance of MaxLogit, ReAct, and ODIN with and without watermarking, largely confirming the fact that our proposed watermarking is orthogonal to existing works. Further, since we directly use the training strategy of softmax scoring-based watermarking for MaxLogit, our results in Table 11 demonstrate that watermarking can also benefit from improved choices of scoring strategies.

C.3 Other Learning Strategies with Watermarking

Also, we consider the situation where a set of OOD data are available for watermark training, where we adopt the tiny-ImageNet [27] dataset as training-time OOD dataset. Here, we replace the Gaussian noise in the OOD learning objective ℓ^{OOD} to be the randomly selected sample from the tiny-ImageNet dataset. Then, the learning objective with training-time OOD data is of the form

$$\mathcal{L}_n(\boldsymbol{w}) = \sum_n \ell^{\text{ID}}(\boldsymbol{x}_i + \boldsymbol{w}, y_i; f) + \beta \sum_n \ell^{\text{OOD}}(\boldsymbol{o}_j + \boldsymbol{w}; f), \tag{16}$$

Table 14: Softmax scoring with tiny-ImageNet and Gaussian noise on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	tiny-ImageNet (Gaussian Noise)		
CIFAR-10			
iSUN	**22.80** (43.60)	**95.88** (93.53)	**99.16** (98.67)
Places365	**61.95** (60.75)	86.47 (**87.85**)	96.87 (**96.94**)
Texture	**41.45** (42.00)	**92.93** (92.83)	**98.51** (98.43)
SVHN	67.15 (**27.25**)	88.43 (**96.00**)	97.59 (**99.17**)
LSUN	**22.40** (40.70)	**96.16** (94.36)	**99.22** (98.86)
average	43.15 (**42.86**)	91.97 (**92.91**)	98.27 (**98.41**)
CIFAR-100			
iSUN	**72.85** (77.85)	**81.22** (79.91)	**95.66** (95.35)
Places365	**79.75** (83.25)	**75.55** (74.53)	**93.77** (93.47)
Texture	**73.35** (78.10)	**78.88** (77.14)	**94.61** (94.26)
SVHN	83.00 (**82.95**)	76.15 (**76.92**)	94.52 (**94.72**)
LSUN	**71.65** (76.75)	**82.93** (79.60)	**96.16** (95.27)
average	**76.20** (76.54)	**79.10** (78.99)	**95.08** (94.96)

Table 15: Free energy scoring with tiny-ImageNet and Gaussian noise on CIFAR benchmarks. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	tiny-ImageNet (Gaussian Noise)		
CIFAR-10			
iSUN	23.65 (**16.30**)	95.89 (**96.97**)	99.16 (**99.39**)
Places365	**36.15** (36.25)	**91.89** (91.87)	**97.94** (97.94)
Texture	**31.20** (32.60)	**93.34** (93.14)	**98.39** (98.08)
SVHN	**13.80** (16.45)	**97.53** (97.11)	**99.49** (99.39)
LSUN	23.15 (**16.85**)	86.20 (**96.97**)	99.20 (**99.38**)
average	25.59 (**23.69**)	92.97 (**95.21**)	**98.83** (98.83)
CIFAR-100			
iSUN	**77.10** (75.05)	**83.62** (83.07)	**96.40** (96.15)
Places365	**79.25** (80.50)	77.67 (**77.78**)	94.22 (**94.45**)
Texture	**68.70** (75.15)	**81.36** (79.55)	**95.10** (94.79)
SVHN	**82.00** (82.85)	**77.38** (75.26)	**94.80** (94.18)
LSUN	**71.35** (71.85)	**84.14** (84.01)	**96.55** (96.33)
average	**75.68** (77.08)	**80.83** (79.93)	**95.41** (95.18)

Table 16: Softmax scoring with learning from Gaussian noise and "perm" augmentation. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	perm (common)		
CIFAR-10			
iSUN	**38.00** (41.50)	**93.99** (93.98)	**98.79** (98.77)
Places365	**55.20** (56.30)	**89.13** (89.03)	**97.49** (97.32)
Texture	**42.20** (43.80)	92.30 (**93.06**)	98.28 (**98.46**)
SVHN	33.35 (**27.00**)	94.75 (**96.07**)	98.93 (**99.20**)
LSUN	**36.40** (37.85)	94.17 (**94.57**)	98.80 (**98.89**)
average	**41.03** (41.29)	92.87 (**93.34**)	98.46 (**98.53**)
CIFAR-100			
iSUN	**77.90** (84.00)	**79.11** (75.98)	**95.11** (94.37)
Places365	**79.90** (83.60)	**75.21** (73.20)	**93.68** (93.22)
Texture	**75.80** (83.00)	**77.13** (72.45)	**94.17** (92.78)
SVHN	**85.45** (87.20)	**73.22** (72.45)	**93.65** (93.45)
LSUN	**76.85** (81.05)	**79.37** (77.40)	**95.21** (94.75)
average	**79.18** (83.77)	**76.81** (74.30)	**94.36** (93.71)

Table 17: Softmax scoring with learning from Gaussian noise and "rotate" augmentation. ↓ (↑) indicates smaller (larger) values are preferred.

	FPR95 ↓	AUROC ↑	AUPR ↑
	rotate (common)		
CIFAR-10			
iSUN	**40.25** (41.50)	93.44 (**93.98**)	98.63 (**98.77**)
Places365	**56.15** (56.30)	88.12 (**89.03**)	97.11 (**97.32**)
Texture	**41.15** (43.80)	92.66 (**93.06**)	98.34 (**98.46**)
SVHN	29.65 (**27.00**)	95.42 (**96.07**)	99.07 (**99.20**)
LSUN	37.90 (**37.85**)	93.95 (**94.57**)	98.76 (**98.89**)
average	**41.02** (41.29)	92.72 (**93.34**)	98.38 (**98.53**)
CIFAR-100			
iSUN	**77.45** (84.00)	**78.73** (75.98)	**94.95** (94.37)
Places365	**79.35** (83.60)	**75.26** (73.20)	**93.67** (93.22)
Texture	**76.35** (83.00)	**76.88** (72.45)	**94.16** (92.78)
SVHN	**84.90** (87.20)	**72.98** (72.45)	**93.58** (93.45)
LSUN	**78.10** (81.05)	**78.59** (77.40)	**95.00** (94.75)
average	**79.23** (83.77)	**76.49** (74.30)	**94.27** (93.71)

where $\boldsymbol{o}_j$ denotes the randomly selected sample from the tiny-ImageNet dataset. The experimental results on CIFAR-10 and CIFAR-100 datasets are summarized in Table 14 and Table 15. Unfortunately, for our current realization, we observe that watermark training with extra OOD data fails to induce a large performance improvement in OOD detection. Even worse, in some cases, learning with extra OOD data can impair the power of the resultant watermarks. We conjecture that the inductive bias introduced by the training-time OOD data may deviate from the test-time cases, severely misleading the resultant watermarks in showing results even lower than that of the simple Gaussian noise.

We also list the detection performance with "perm" and "rotate" in Section 6, demonstrating the effectiveness of the resultant watermarks on far OOD cases. Here, the training objective is:

$$\mathcal{L}_n(\boldsymbol{w}) = \sum_n \ell^{\text{ID}}(\boldsymbol{x}_i + \boldsymbol{w}, y_i; f) + \beta \sum_n \ell^{\text{OOD}}(\boldsymbol{\epsilon}_j + \boldsymbol{w}; f) + \beta \sum_n \ell^{\text{OOD}}(\tilde{\boldsymbol{x}}_i + \boldsymbol{w}; f), \tag{17}$$

where $\tilde{\boldsymbol{x}}$ is an augmentation of the original $\boldsymbol{x}$ with either "perm" or "rotate". The results are summarized from Tables 16 to 19. As we can see, the results with "perm" and "rotate" is comparable with (even better than) the original learning setup with only the Gaussian noise (common).

Table 18: Free energy scoring with learning from Gaussian noise and "perm" augmentation. ↓ (↑) prefers smaller (larger) values.

	FPR95 ↓	AUROC ↑	AUPR ↑
	perm (common)		
CIFAR-10			
iSUN	25.60 (**24.20**)	95.81 (**96.10**)	99.14 (**99.22**)
Places365	39.70 (**39.45**)	**91.96** (91.46)	**98.01** (97.89)
Texture	39.15 (**38.95**)	92.13 (**92.65**)	97.93 (**98.16**)
SVHN	**16.95** (18.75)	**97.01** (96.54)	**99.37** (99.27)
LSUN	22.10 (**21.80**)	**96.38** (96.27)	**99.27** (99.24)
average	28.70 (**28.63**)	**94.66** (94.61)	98.74 (**98.76**)
CIFAR-100			
iSUN	77.05 (**75.30**)	83.49 (**84.51**)	96.32 (**96.78**)
Places365	80.40 (**78.05**)	77.11 (**78.15**)	94.25 (**94.28**)
Texture	**68.85** (70.80)	**81.31** (81.14)	95.14 (**95.02**)
SVHN	81.95 (**80.50**)	**78.27** (77.27)	**95.14** (94.70)
LSUN	77.20 (**75.10**)	**83.58** (83.53)	96.35 (**97.05**)
average	77.09 (**75.95**)	80.75 (**81.12**)	95.44 (**95.57**)

Table 19: Free energy scoring with learning from Gaussian noise and "rotate" augmentation. ↓ (↑) prefers smaller (larger) values.

	FPR95 ↓	AUROC ↑	AUPR ↑
	rotate (common)		
CIFAR-10			
iSUN	23.85 (**24.20**)	96.04 (**96.10**)	99.20 (**99.22**)
Places365	**38.60** (39.45)	**91.94** (91.46)	**97.97** (97.89)
Texture	**35.20** (38.95)	**92.93** (92.65)	**98.26** (98.16)
SVHN	**16.85** (18.75)	**97.00** (96.54)	**99.37** (99.27)
LSUN	22.80 (**21.80**)	96.10 (**96.27**)	99.19 (**99.24**)
average	**27.46** (28.63)	**94.80** (94.61)	98.80 (**98.76**)
CIFAR-100			
iSUN	85.45 (**75.30**)	81.39 (**84.51**)	95.85 (**96.78**)
Places365	80.10 (**78.05**)	76.53 (**78.15**)	93.79 (**94.28**)
Texture	71.55 (**70.80**)	80.27 (**81.14**)	94.67 (**95.02**)
SVHN	**80.20** (80.50)	**79.27** (77.27)	**95.29** (94.70)
LSUN	81.25 (**75.10**)	81.85 (**83.53**)	95.97 (**97.05**)
average	79.71 (**75.95**)	79.86 (**81.12**)	95.12 (**95.57**)

Table 20: Comparison of watermarking and different OOD scoring functions on CIFAR benchmarks.

	w/o watermarking							w/ watermarking				
	Softmax [18]	Energy [32]	ReAct [45]	ODIN [31]	Mahalanobis [28]	GradNorm [5]	OE [19]	Softmax	Energy	ReAct	ODIN	OE
CIFAR-10												
FPR95	55.70	37.67	57.59	42.64	34.18	40.51	20.96	42.86	23.69	38.91	37.06	**19.06**
AUROC	89.82	90.56	88.79	90.21	93.23	90.10	96.41	92.91	95.21	93.01	93.50	**96.89**
AUPR	97.32	97.46	97.37	97.49	98.41	97.35	99.39	98.42	98.83	80.42	98.55	**99.31**
CIFAR-100												
FPR95	82.97	81.61	84.96	81.51	55.63	83.68	60.30	79.98	77.07	75.65	76.36	**54.20**
AUROC	73.88	76.50	73.06	73.17	82.26	72.93	82.99	77.57	79.93	78.98	79.56	**83.90**
AUPR	93.43	94.05	93.03	93.06	95.56	93.00	95.91	94.61	95.18	94.96	95.18	**96.08**

C.4 Comparison with State-of-the-art Methods

For the concreteness of our discussion, this section compares our proposal with state-of-the-art methods in OOD detection. In particular, we compare with softmax scoring (Softmax) [18], free energy scoring (Energy) [32], ReAct [45], ODIN [31], Mahalanobis [28], GradNorm [22], and OE [19]. The experimental results on CIFAR benchmarks are summarized in Table 20. The average performance on iSUN, Places365, Texture, SVHN, and LSUN is reported. As we can see, watermarking can boost the performance of various scoring methods in OOD detection, achieving the best detection performance compared with all other advanced methods.

C.5 Experiments with Mean and Standard Deviation

This section further verifies the results from Table 1 to Table 4 with five individual trails (random seeds). In Table 21, Table 22, and Table 23, we summarize the average results and the standard deviation for the softmax scoring and the free energy scoring, respectively. In Figure 5, we also depict the learned watermarks for each trial on CIFAR-10 and CIFAR-100. Overall, we observe that the learned watermarks preserve some similar pattern (e.g., the shape of areas with large values) given the same dataset and the same scoring strategy, and the improvement of watermarking is stable across different datasets and scoring methods.

C.6 Experiments with different models

We demonstrate the effectiveness of our watermarking across various model architectures, including ResNet-18 [15], WRN-40-2, and ViT-B/16 [9]. We conduct experiments on the ImageNet benchmark and summarized the results in Table 24. As we can see, in both the softmax and free energy scoring cases, our watermarking can improve the detection performance across various models. However, the

Table 21: The softmax scoring with/without watermarking on CIFAR benchmarks. Five individual trails (mean $\pm$ std) are conducted. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**44.68** $\pm$ **1.49** (55.43 $\pm$ 0.29)	**93.38** $\pm$ **0.28** (90.10 $\pm$ 0.22)	**97.80** $\pm$ **0.07** (97.80 $\pm$ 0.07)
Places365	**59.21** $\pm$ **0.97** (60.53 $\pm$ 1.31)	**88.83** $\pm$ **0.34** (87.83 $\pm$ 0.17)	**97.08** $\pm$ **0.11** (97.06 $\pm$ 0.06)
Texture	**42.07** $\pm$ **1.23** (59.37 $\pm$ 1.55)	**93.03** $\pm$ **0.23** (88.56 $\pm$ 0.37)	**98.47** $\pm$ **0.07** (97.20 $\pm$ 0.09)
SVHN	**29.25** $\pm$ **2.17** (48.07 $\pm$ 0.97)	**95.69** $\pm$ **0.35** (91.80 $\pm$ 0.13)	**99.11** $\pm$ **0.07** (97.24 $\pm$ 0.03)
LSUN	**40.45** $\pm$ **2.14** (52.23 $\pm$ 1.04)	**94.11** $\pm$ **0.45** (91.50 $\pm$ 0.10)	**98.79** $\pm$ **0.11** (98.15 $\pm$ 0.04)
average	**43.13** $\pm$ **0.16** (55.12 $\pm$ 0.10)	**93.00** $\pm$ **0.33** (89.95 $\pm$ 0.19)	**98.25** $\pm$ **0.08** (97.49 $\pm$ 0.05)
CIFAR-100			
iSUN	**78.70** $\pm$ **1.48** (82.40 $\pm$ 0.81)	**78.30** $\pm$ **0.57** (75.62 $\pm$ 0.33)	**94.92** $\pm$ **0.17** (94.11 $\pm$ 0.10)
Places365	**82.55** $\pm$ **0.65** (82.97 $\pm$ 0.86)	**74.69** $\pm$ **0.50** (74.29 $\pm$ 0.29)	**93.77** $\pm$ **0.13** (93.41 $\pm$ 0.12)
Texture	**77.83** $\pm$ **1.47** (83.48 $\pm$ 0.70)	**76.42** $\pm$ **0.18** (73.37 $\pm$ 0.37)	**94.06** $\pm$ **0.08** (92.95 $\pm$ 0.11)
SVHN	**83.71** $\pm$ **2.27** (84.72 $\pm$ 0.73)	**76.16** $\pm$ **0.78** (71.29 $\pm$ 0.63)	**94.51** $\pm$ **0.17** (92.88 $\pm$ 0.22)
LSUN	**78.57** $\pm$ **1.09** (81.67 $\pm$ 0.78)	**78.37** $\pm$ **0.54** (75.77 $\pm$ 0.33)	**94.92** $\pm$ **0.16** (94.18 $\pm$ 0.11)
average	**88.27** $\pm$ **1.39** (83.04 $\pm$ 0.77)	**76.78** $\pm$ **0.51** (74.06 $\pm$ 0.39)	**94.43** $\pm$ **0.14** (93.50 $\pm$ 0.13)

Table 22: The free energy scoring with/without watermarking on CIFAR benchmarks. Five individual trails (mean $\pm$ std) are conducted. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
	w/ (w/o) watermark		
CIFAR-10			
iSUN	**18.84** $\pm$ **2.76** (33.66 $\pm$ 0.75)	**96.28** $\pm$ **0.62** (92.62 $\pm$ 0.31)	**99.23** $\pm$ **0.13** (98.27 $\pm$ 0.10)
Places365	**38.89** $\pm$ **1.74** (40.67 $\pm$ 0.91)	**91.92** $\pm$ **0.41** (89.62 $\pm$ 0.20)	**98.01** $\pm$ **0.11** (97.16 $\pm$ 0.12)
Texture	**34.60** $\pm$ **2.16** (52.67 $\pm$ 1.10)	**93.36** $\pm$ **0.26** (85.19 $\pm$ 0.35)	**98.31** $\pm$ **0.06** (95.40 $\pm$ 0.13)
SVHN	**14.96** $\pm$ **0.93** (35.60 $\pm$ 0.50)	**97.12** $\pm$ **0.15** (91.08 $\pm$ 0.22)	**99.39** $\pm$ **0.03** (97.71 $\pm$ 0.07)
LSUN	**16.63** $\pm$ **2.12** (27.12 $\pm$ 0.85)	**96.43** $\pm$ **0.43** (94.32 $\pm$ 0.07)	**99.26** $\pm$ **0.09** (98.70 $\pm$ 0.02)
average	**24.78** $\pm$ **1.94** (37.94 $\pm$ 0.82)	**95.02** $\pm$ **0.37** (90.56 $\pm$ 0.23)	**98.84** $\pm$ **0.08** (97.44 $\pm$ 0.08)
CIFAR-100			
iSUN	**74.62** $\pm$ **1.97** (81.85 $\pm$ 1.14)	**84.30** $\pm$ **0.85** (78.78 $\pm$ 0.40)	**96.53** $\pm$ **0.24** (94.90 $\pm$ 0.13)
Places365	**77.79** $\pm$ **0.27** (80.27 $\pm$ 1.02)	**78.13** $\pm$ **0.78** (76.46 $\pm$ 0.54)	**94.40** $\pm$ **0.31** (93.92 $\pm$ 0.16)
Texture	**68.96** $\pm$ **1.51** (79.47 $\pm$ 0.27)	**82.07** $\pm$ **0.62** (76.34 $\pm$ 0.34)	**95.38** $\pm$ **0.24** (93.64 $\pm$ 0.11)
SVHN	**80.30** $\pm$ **0.75** (85.80 $\pm$ 0.87)	**78.55** $\pm$ **0.63** (73.61 $\pm$ 0.37)	**95.11** $\pm$ **0.18** (93.51 $\pm$ 0.10)
LSUN	**71.25** $\pm$ **2.00** (79.26 $\pm$ 1.43)	**84.94** $\pm$ **0.54** (79.34 $\pm$ 0.40)	**96.65** $\pm$ **0.12** (94.99 $\pm$ 0.11)
average	**74.58** $\pm$ **1.30** (81.33 $\pm$ 0.94)	**81.59** $\pm$ **0.68** (76.90 $\pm$ 0.41)	**95.61** $\pm$ **0.21** (94.19 $\pm$ 0.12)

improvements after watermarking on the large-scale models (i.e., ViT-B/16) are not as remarkable as that of the small models (e.g., ResNet-18). It is because that the large-scale models themselves can already excel at OOD detection (better results without watermarking than that of ResNet-18 and WRN-40-2), so there does not remain a large space for their further improvements.

C.7 Hyper-parameter Setups

For the hyper-parameter setups in our experiments, we use random search to choose the proper σ_1 from the candidate parameter set $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, and the proper ρ from $\{0.0, 0.02, 0.05, 0.07, 0.1, 0.2, 0.5, 0.7, 1.0, 2.0, 5.0\}$. For softmax scoring, β is chosen from $\{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$; and for free energy scoring, β is chosen from $\{0.0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$. The model performance is tested on a set of validation OOD datasets that are separated from the original iSUN, Places365, Texture, SVHN, and LSUN datasets. Accordingly, for softmax scoring, we set $\sigma_1 = 0.4$, $\rho = 1.0$, $\beta = 3.5$ in CIFAR-10 and $\sigma_1 = 1.0$, $\rho = 0.2$, $\beta = 2.5$ in CIFAR-100; for free energy scoring, we set $\sigma_1 = 0.6$, $\rho = 0.7$, $\beta = 0.1$ and $\sigma_1 = 1.0$, $\rho = 0.05$, $\beta = 1.2$. We adopt the random search with many trials by the following three steps. Step 1: we randomly select a hyperparameter (e.g., β) and fix the values of all other hyperparameters to be their current optimal values. Step 2: we choose the best β from the candidate set. Step 3: do Steps 1-2 again. We repeat Steps 1 and 2 for 50 times in our experiments.

Table 23: Softmax and free energy scoring with/without watermarking on ImageNet. Five individual trails (mean $\pm$ std) are conducted. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
	w/ (w/o) watermark		
Softmax Scoring			
iSUN	**12.69** $\pm$ **1.55** (52.38 $\pm$ 2.07)	**97.74** $\pm$ **1.23** (92.62 $\pm$ 1.31)	**99.60** $\pm$ **0.17** (98.27 $\pm$ 0.20)
Places365	**70.80** $\pm$ **2.27** (73.35 $\pm$ 3.61)	**81.74** $\pm$ **1.34** (80.52 $\pm$ 0.85)	**95.32** $\pm$ **0.17** (94.88 $\pm$ 0.31)
Texture	**60.59** $\pm$ **2.65** (67.71 $\pm$ 2.83)	**83.46** $\pm$ **2.37** (82.47 $\pm$ 1.77)	**98.35** $\pm$ **0.06** (98.20 $\pm$ 0.17)
SVHN	44.81 $\pm$ 2.03 (**28.69** $\pm$ **1.49**)	93.72 $\pm$ 1.29 (**95.55** $\pm$ **1.13**)	98.76 $\pm$ 0.33 (**99.14** $\pm$ **0.20**)
LSUN	**11.63** $\pm$ **1.47** (54.43 $\pm$ 2.15)	**97.85** $\pm$ **1.07** (91.96 $\pm$ 2.57)	**99.57** $\pm$ **0.17** (98.39 $\pm$ 0.25)
average	**40.10** $\pm$ **1.99** (55.31 $\pm$ 2.43)	**90.90** $\pm$ **1.46** (88.62 $\pm$ 1.52)	**98.32** $\pm$ **0.18** (97.77 $\pm$ 0.22)
Free Energy Scoring			
iSUN	**32.61** $\pm$ **2.21** (45.41 $\pm$ 2.84)	**93.59** $\pm$ **1.45** (93.96 $\pm$ 1.85)	**98.74** $\pm$ **0.20** (98.23 $\pm$ 0.28)
Places365	**72.64** $\pm$ **1.37** (74.99 $\pm$ 2.50)	**79.55** $\pm$ **0.87** (78.83 $\pm$ 0.90)	**94.67** $\pm$ **0.32** (94.26 $\pm$ 0.15)
Texture	**67.36** $\pm$ **1.51** (67.39 $\pm$ 1.62)	**80.60** $\pm$ **1.70** (80.52 $\pm$ 1.14)	**97.00** $\pm$ **0.15** (96.90 $\pm$ 0.25)
SVHN	**12.92** $\pm$ **2.20** (25.85 $\pm$ 2.47)	**97.49** $\pm$ **1.00** (95.26 $\pm$ 1.51)	**99.45** $\pm$ **0.10** (99.00 $\pm$ 0.10)
LSUN	**33.53** $\pm$ **2.00** (46.68 $\pm$ 2.33)	**93.50** $\pm$ **1.50** (90.59 $\pm$ 1.70)	**98.59** $\pm$ **0.15** (97.96 $\pm$ 0.27)
average	**43.81** $\pm$ **1.85** (52.06 $\pm$ 2.35)	**88.94** $\pm$ **1.30** (87.83 $\pm$ 1.42)	**97.69** $\pm$ **0.18** (97.27 $\pm$ 0.21)

Table 24: The softmax scoring and the free energy scoring with/without watermarking on the ImageNet dataset, where we adopt different models including ResNet-18, WRN-40-2, and ViT-B/16. The notion $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

	Softmax Scoring			Free Energy Scoring		
	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
ResNet-18	**41.85** (55.60)	**90.98** (86.64)	**98.22** (97.80)	**42.87** (53.26)	**89.50** (86.42)	**97.83** (97.75)
WRN-40-2	**40.50** (54.93)	**91.22** (88.57)	**98.42** (97.69)	**43.23** (52.73)	**89.10** (86.14)	**97.73** (97.15)
ViT-B/16	**31.63** (34.95)	**92.52** (91.31)	**86.77** (85.13)	**20.61** (21.64)	**95.10** (94.95)	**90.87** (90.58)

Further, from Table 25 to 36, we list the performance of watermarking with different hyper-parameter settings for reference, where we fix the values of all other hyper-parameters (except for the considered one) to be their optimal values. Finally, we list the results on CIFAR benchmarks regarding the free energy scoring with different values of T in Table 37.

Table 25: Softmax scoring on CIFAR-10 with various σ_1.

	FPR95	AUROC	AUPR
2.00	42.11	92.01	98.24
1.80	42.84	91.89	98.22
1.60	41.41	92.14	98.25
1.40	42.20	91.82	98.20
1.20	41.98	91.91	98.20
1.00	42.76	91.95	98.23
0.80	43.38	91.89	98.21
0.60	39.16	92.89	98.41
0.40	**38.66**	**93.03**	**98.45**
0.20	43.88	92.80	98.42
0.00	48.71	91.43	98.11

Table 26: Softmax scoring on CIFAR-10 with various ρ.

	FPR95	AUROC	AUPR
5.00	60.02	87.36	97.15
2.00	46.24	91.61	98.14
1.00	**39.12**	**92.96**	**98.42**
0.70	41.07	92.77	98.40
0.50	43.55	92.38	98.34
0.20	42.02	92.68	98.38
0.10	41.99	92.77	98.41
0.07	42.13	92.79	98.42
0.05	42.06	92.84	98.42
0.02	44.35	92.34	98.32
0.00	43.04	92.44	98.32

Table 27: Softmax scoring on CIFAR-10 with various β.

	FPR95	AUROC	AUPR
5.00	40.11	92.48	98.34
4.50	41.30	92.39	98.32
4.00	41.21	92.39	98.33
3.50	**38.65**	**92.55**	**98.34**
3.00	41.01	92.55	98.35
2.50	39.66	92.61	98.35
2.00	38.95	92.98	98.47
1.50	43.89	92.35	98.33
1.00	40.47	93.08	98.49
0.50	44.51	92.35	98.30
0.00	49.91	91.46	98.08

Table 28: Energy scoring on CIFAR-10 with various σ_1.

	FPR95	AUROC	AUPR
2.00	28.49	94.61	98.74
1.80	28.61	94.56	98.71
1.60	25.36	95.15	98.85
1.40	28.74	94.85	98.77
1.20	27.48	94.91	98.78
1.00	26.47	95.02	98.83
0.80	24.99	95.08	98.85
0.60	**24.50**	**95.29**	**98.93**
0.40	26.40	94.84	98.78
0.20	27.21	94.73	98.71
0.00	27.97	94.85	98.75

Table 29: Energy scoring on CIFAR-10 with various ρ.

	FPR95	AUROC	AUPR
5.00	44.20	91.18	97.92
2.00	29.82	94.32	98.69
1.00	24.56	95.10	98.86
0.70	**24.38**	**95.20**	**98.87**
0.50	25.96	95.19	98.85
0.20	25.06	95.32	98.90
0.10	29.21	94.49	98.69
0.07	28.67	94.78	98.75
0.05	28.72	94.82	98.81
0.02	27.58	94.93	98.58
0.00	25.38	94.22	98.78

Table 30: Energy scoring on CIFAR-10 with various β.

	FPR95	AUROC	AUPR
1.00	50.71	91.23	98.10
0.80	42.48	92.42	98.30
0.60	42.57	92.78	98.42
0.40	33.74	94.07	98.69
0.20	30.12	94.57	98.74
0.10	**23.68**	**95.35**	**98.90**
0.08	27.08	94.87	98.76
0.06	25.63	95.15	98.84
0.04	24.58	95.06	98.79
0.02	25.47	94.86	98.73
0.00	33.31	92.37	98.08

Table 31: Softmax scoring on CIFAR-100 with various σ_1.

	FPR95	AUROC	AUPR
2.00	82.37	76.45	82.37
1.80	79.72	77.25	94.50
1.60	79.50	77.69	94.57
1.40	79.22	77.64	94.58
1.20	78.28	78.45	94.74
1.00	**76.57**	**79.06**	**94.94**
0.80	77.88	78.84	94.91
0.60	76.84	79.13	94.99
0.40	80.47	76.59	94.31
0.20	81.80	75.87	94.14
0.00	83.07	73.81	93.48

Table 32: Softmax scoring on CIFAR-100 with various ρ.

	FPR95	AUROC	AUPR
5.00	84.19	72.43	92.78
2.00	76.44	79.63	95.08
1.00	77.89	78.58	94.84
0.70	76.39	78.98	94.91
0.50	78.00	78.82	94.89
0.20	**75.43**	**78.38**	**94.73**
0.10	77.68	78.63	94.84
0.07	78.07	78.01	96.60
0.05	76.26	77.77	94.53
0.02	77.79	78.80	94.81
0.00	79.14	79.14	94.91

Table 33: Softmax scoring on CIFAR-100 with various β.

	FPR95	AUROC	AUPR
5.00	78.55	76.57	94.13
4.50	84.54	72.69	92.99
4.00	79.27	76.01	93.95
3.50	77.56	76.55	94.11
3.00	79.98	76.33	94.12
2.50	**76.43**	**77.28**	**94.29**
2.00	79.84	77.23	94.38
1.50	77.38	78.85	94.89
1.00	76.91	78.45	94.76
0.50	76.75	79.14	94.97
0.00	83.25	73.73	93.45

Table 34: Energy scoring on CIFAR-100 with various σ_1.

	FPR95	AUROC	AUPR
2.00	77.26	80.38	95.32
1.80	76.55	80.56	95.31
1.60	78.09	79.68	95.12
1.40	76.68	80.82	95.41
1.20	75.69	80.64	95.34
1.00	**75.57**	**80.80**	**95.40**
0.80	75.87	80.46	95.25
0.60	78.11	80.12	95.23
0.40	76.18	80.73	95.36
0.20	76.17	80.11	95.22
0.00	75.01	80.78	95.39

Table 35: Energy scoring on CIFAR-100 with various ρ.

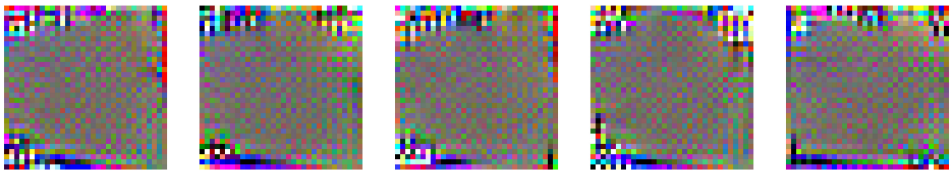
	FPR95	AUROC	AUPR
5.00	77.39	79.91	95.17
2.00	81.39	78.07	94.70
1.00	76.58	80.64	95.37
0.70	78.42	78.40	94.76
0.50	76.52	80.54	95.33
0.20	75.32	91.08	95.48
0.10	75.48	80.63	95.35
0.07	75.72	79.45	95.03
0.05	**75.18**	**79.54**	**95.06**
0.02	75.83	79.45	95.02
0.00	76.49	79.10	94.98

Table 36: Energy scoring on CIFAR-100 with various β.

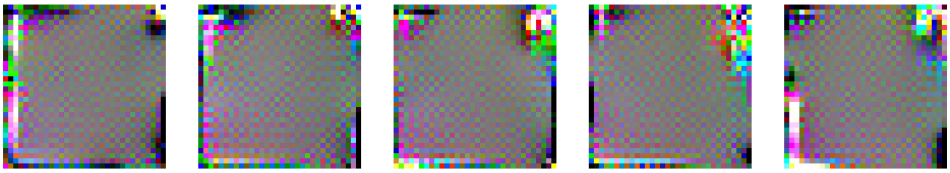
	FPR95	AUROC	AUPR
2.00	82.20	78.18	94.77
1.80	79.30	80.01	95.20
1.60	75.19	80.53	95.37
1.40	78.39	80.22	95.26
1.20	**74.78**	**81.51**	**95.57**
1.00	77.68	80.65	95.36
0.80	77.84	90.11	95.20
0.60	75.25	81.47	95.60
0.40	78.39	78.67	94.86
0.20	80.46	77.32	94.45
0.00	93.23	73.73	93.39

Table 37: Energy scoring on CIFAR benchmarks with various value of the hyper-parameter T.

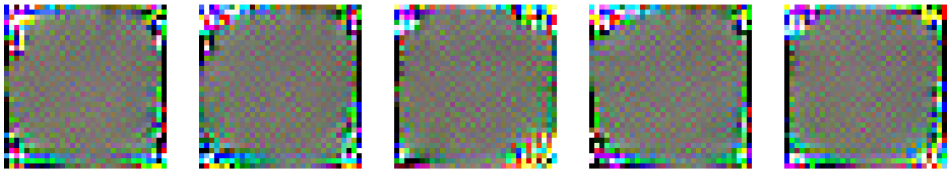
T	CIFAR-10							CIFAR-100						
	1	5	10	50	100	500	1000	1	5	10	50	100	500	1000
FPR95	**25.9**	27.8	27.7	28.9	28.4	28.0	31.2	**74.1**	80.4	77.7	82.3	80.4	87.0	89.3
AUROC	**95.0**	94.3	94.0	93.5	93.5	93.7	93.4	**81.9**	76.2	76.4	72.4	73.5	70.8	68.5
AUPR	**98.7**	98.5	98.4	98.2	98.2	98.2	98.1	**95.7**	94.1	94.1	92.7	93.0	91.9	90.9



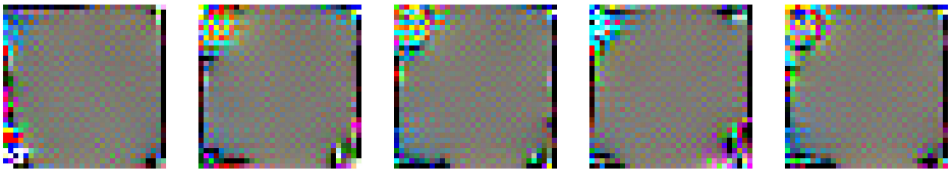
(a) softmax scoring on CIFAR-10



(b) softmax scoring on CIFAR-100



(c) free energy scoring on CIFAR-10



(d) free energy scoring on CIFAR-100

Figure 5: Illustrations of the learned watermarks with 5 individual trails.

(a) (b)

Figure 6: Masked watermarks for the free energy scoring, with (a) $\chi_1 = 1$ and (b) $\chi_2 = 1$.

C.8 Different Areas in the Watermark

After watermarking, the edge area of the image is overwhelmed by the watermark's pattern, while the centre part is not much affected. However, it does not mean that the centre area is not important. Instead, under the premise of maintaining the original features, the centre area also encodes useful information in OOD detection. Table 38 is a verification of this conclusion on CIFAR-10 dataset with the free energy scoring, where we employ various masks in only preserving the watermark's features with their absolute values larger (smaller) than a threshold χ_1 (χ_2). As we can see, even if only a small portion of the watermark is masked (e.g., $\chi_1 = 0.10$ or $\chi_2 = 10.0$), there is a large drop in performance, even lower than the results without any watermarking. It indicates that both areas of the watermark contribute, and the overall watermark works as a whole for effective OOD detection.

Table 38: The average performance of the free energy scoring on CIFAR-10 with masking. $\downarrow$ ($\uparrow$) indicates smaller (larger) values are preferred.

	FPR95 $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
w/ watermark	**23.69**	**95.21**	**98.83**
w/o watermark	37.67	90.56	97.46
χ_1			
0.10	30.46	94.41	98.68
1.00	37.72	93.00	98.52
10.0	37.66	91.87	97.94
χ_2			
0.10	42.06	89.88	97.38
1.00	51.28	87.54	96.86
10.0	36.72	93.08	98.39