Dimension-independent Certified Neural Network Watermarks via Mollifier Smoothing

Jiaxiang Ren [1] **Yang Zhou** [1] **Jiayin Jin** [1] **Lingjuan Lyu** [2] **Da Yan** [3]

Abstract

Certified_Watermarks is the first to provide a watermark certificate against l_2-norm watermark removal attacks, by leveraging the randomized smoothing techniques for certified robustness to adversarial attacks. However, the randomized smoothing techniques suffer from hardness of certified robustness in high-dimensional space against l_p-norm attacks for large p ($p > 2$). The certified watermark method based on the randomized smoothing is no exception, i.e., fails to provide meaningful certificates in high-dimensional space against the l_p-norm watermark removal attacks ($p > 2$). By leveraging mollifier theory, this paper proposes a mollifier smoothing method with dimension-independent certified radius of our proposed smooth classifier, for conducting the certified watermark problem against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) for high parameter dimension d. Based on partial differential equation (PDE) theory, an approximation of mollifier smoothing is developed to alleviate the inefficiency of sampling and prediction in the randomized smoothing as well as numerical integration in the mollifier smoothing, while maintaining the certified watermark against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$).

1. Introduction

Watermarking neural networks has become the tendency in protecting intellectual property of deep neural networks (DNNs) by embedding watermarks into the DNNs to enable the ownership verification of deep learning models (Zhang et al., 2018; Adi et al., 2018; Rouhani et al., 2019). Despite achieving remarkable performance, recent studies have shown that many watermarking mechanisms are highly vulnerable to watermark removal attacks (Aiken et al., 2021; Shafieinejad et al., 2021). The watermark removal attacks aim to make the victim model forget the embedded watermarks but preserve accuracy on main task through fine-tuning the model with source training data or/and proxy data (Uchida et al., 2017; Wang & Kerschbaum, 2019; Yang et al., 2019; Liu et al., 2020; Chen et al., 2021b; Guo et al., 2021; Zhong et al., 2022; Yan et al., 2022).

Many encouraging empirical defense progresses have been made towards improving watermarking robustness against watermark removal attacks (Tartaglione et al., 2020; Sun et al., 2021; Uchida et al., 2017; Zhang et al., 2018; Adi et al., 2018; Rouhani et al., 2019; Namba & Sakuma, 2019; Wang & Kerschbaum, 2021; Yang et al., 2021; Wang et al., 2021; Liu et al., 2022b; Wu, 2022; Pagnotta et al., 2022; Anonymous, 2023). However, it is typically observed that empirically robust watermarking models are defeated by strong attacks (Carlini & Wagner, 2017; Athalye et al., 2018; Uesato et al., 2018; Tramèr et al., 2020). The watermarks can be detected and removed without compromising prediction accuracy of stolen model, by adding regularization, fine-tuning and pruning (Aiken et al., 2021; Shafieinejad et al., 2021), or leveraging both labeled and unlabeled data (Wang & Kerschbaum, 2019). Even if the watermarks appear empirically robust to currently known attacks, stronger attacks may eventually come along, prompting better watermark methods (Bansal et al., 2022).

Certified defense techniques guarantee the watermarks to be unremovable under the watermark removal attacks. In backdoor-based watermarking, the owner employs a trigger set of specially chosen images that has disjoint distribution compared to original dataset. A sufficiently high accuracy on this trigger set implies the model ownership with high probability. In other words, the model has a low accuracy on the trigger set if the watermark is removed. Goldberger et al. proposed to find the minimal modification required to remove a watermark in a neural network (Goldberger et al., 2020). But they did not propose new methods to embed a watermark that would be more robust. In addition, their approach is based on solving mixed integer linear programs and thus does not scale well to larger DNNs.

[1]Auburn University, USA [2]Sony AI, Japan [3]University of Alabama at Birmingham, USA. Correspondence to: Yang Zhou <yangzhou@auburn.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Certified_Watermarks (Bansal et al., 2022) is the first to provide a watermark certificate against l_2-norm watermark removal attacks, by leveraging the randomized smoothing techniques (Cohen et al., 2019; Chiang et al., 2020) for certified robustness to adversarial attacks. The Certified_Watermarks method takes a base trigger set accuracy function as input, outputs a smooth one by adding randomized noise to model parameters, and bounds the worst-case decrease of the smooth trigger set accuracy function in its trigger set accuracy, when the adversary is allowed to move the model parameters within a certain l_2-norm ball.

In the field of certified robustness to adversarial attacks, several recent studies have identified the inapplicability of randomized smoothing to the certified robustness in high-dimensional space against l_p-norm attacks ($p > 2$) (Kumar et al., 2020; Blum et al., 2020; Yang et al., 2020; Mohapatra et al., 2020). Concretely, the largest certified radius r_p by randomized smoothing against l_p-norm attacks is proportional to $O(1/d^{\frac{1}{2}-\frac{1}{p}})$, where d is the input dimension. It is observed that r_p decreases with d when $p > 2$ and a special case of $p = 2$ does not suffer from such dependency on d. We have witnessed many promising randomized smoothing methods for certifying l_0 (Lee et al., 2019; Levine & Feizi, 2020), l_1 (Lécuyer et al., 2019; Li et al., 2019; Teng et al., 2019), and l_2 robustness (Li et al., 2018; Cohen et al., 2019; Salman et al., 2019; Zhai et al., 2020; Levine et al., 2020). However, the randomized smoothing is incapable of producing large certified radius against l_p-norm attacks for high d and $p > 2$. In particular, when $p \to \infty$, $O(1/d^{\frac{1}{2}-\frac{1}{p}}) \to O(1/\sqrt{d})$. This results in the failure of certified robustness against l_p adversarial attacks ($p > 2$) in high-dimensional space. In the context of certified watermark, the input of randomized smoothing is millions or billions of model parameters, which has huge dimension d. Thus, Certified_Watermarks (Bansal et al., 2022) based on the randomized smoothing techniques (Cohen et al., 2019; Chiang et al., 2020) fails to provide meaningful watermark certificates in high-dimensional parameter space against the l_p-norm watermark removal attacks ($p > 2$).

To our best knowledge, this work is the first to leverage mollifier theory and partial differential equation theory for conducting the certified watermark problem in high-dimensional parameter space against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) with better applicability.

Based on the mollifier theory, we propose a mollifier smoothing method to solve the certified watermark problem in high-dimensional parameter space against l_p-norm watermark removal attacks ($1 \leq p \leq \infty$). It aims to solve an integral transform problem with the convolution of a base classifier $f(w)$ with the embedded watermarks and a smooth mollifier, where w is the model parameter. We theoretically derive that the Lipschitz constant and certified radius of our smooth classifier $g(w)$ are independent of the parameter dimension d and thus $g(w)$ can maintain the high classification accuracy on the trigger set (i.e. watermark maintenance) and provide certified watermark guarantees against l_p-norm adversarial attacks for high d and any p ($1 \leq p \leq \infty$).

In order to retain high-confidence certificates, certifying the watermarks with the randomized smoothing or mollifier smoothing method requires to sample massive noise points. For example, certifying the robustness with $1 - \alpha = 99.9\%$ confidence would require 10^5 samples in all experiments in the randomized smoothing paper (Cohen et al., 2019). In addition, the cost of sampling and prediction over a large number of points of the input within its neighborhood is non-trivial. In order to further improve the efficiency of the certified watermark, based on the partial differential equation (PDE) theory, we develop an approximation of our smooth classifier $g(w)$ to avoid time-consuming noise sampling and integral calculation, while maintaining the certified watermark unremovable against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$).

Empirical evaluation on real datasets demonstrates the superior performance of our mollifier smoothing appproach against several state-of-the-art empirical and certified defense methods on image classification. In addition, more experiments, implementation details, and hyperparameter selection and setting are presented in Appendices A.2-A.4.

2. Background and Problem Statement

2.1. Randomized Smoothing for Certified Robustness & Certified Watermarks

Randomized smoothing for certified robustness aims to build a smooth classifier g from a base classifier f that maps inputs $x \in \mathbb{R}^d$ to classes $c \in C$.

$$g(x) = \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \underset{\varepsilon \sim \mathcal{D}}{\mathbb{P}}(f(x+\varepsilon) = c) \tag{1}$$

where $\mathcal{D} = \mathcal{N}\left(0, \sigma^2 I\right)$ is a Gaussian probability distribution in $\mathbb{R}^d$. g returns whichever class f is most likely to return when x is perturbed by noise ε.

Let $p_c(x)$ be the output probability of f over class c, i.e., $p_c(x) = \underset{\varepsilon \sim \mathcal{D}}{\mathbb{P}}(f(x+\varepsilon) = c)$. Without loss of generality, we assume that $p_A(x)$ and $p_B(x)$ are the probabilities on the most probable class c_A and the runner-up class c_B respectively. If $\mathbb{P}\left(f(x+\varepsilon) = c_A\right) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c)$, where $\underline{p_A(x)}$ is a lower bound of $p_A(x)$ and $\overline{p_B(x)}$ is an upper bound of $p_B(x)$, then $g(x+\delta) = c_A$ for $\forall \delta \in \mathbb{R}^d, \|\delta\|_p \leq r_p$. In this case, the smooth classifier g can always output the correct prediction as long as the perturbation δ is within a certified l_p-norm radius of r_p for $p > 0$.

Theorem 1. *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$. Let g be defined as in (1). Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy (Cohen et al., 2019):*

$$\mathbb{P}\left(f(x+\varepsilon) = c_A\right) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{2}$$

Then $g(x + \delta) = c_A$ for all $\|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2}\left(\Phi^{-1}\left(\underline{p_A}\right) - \Phi^{-1}\left(\overline{p_B}\right)\right) \tag{3}$$

where Φ^{-1} is the inverse of the standard Gaussian CDF.

Several recent studirs report that the largest certified radius r_p by the randomized smoothing against l_p-norm threat model is proportional to $O(1/d^{\frac{1}{2}-\frac{1}{p}})$, where d is the input dimension (Kumar et al., 2020; Blum et al., 2020; Yang et al., 2020; Mohapatra et al., 2020). An explicit upper bound of r_p for a Gaussian distribution with variance σ^2 is given as follows (Kumar et al., 2020).

$$r_p = \frac{\sigma}{2d^{\frac{1}{2}-\frac{1}{p}}}\left(\Phi^{-1}\big(p_A(x)\big) - \Phi^{-1}\big(p_B(x)\big)\right) \tag{4}$$

where noise σ also serves as a hyperparameter to balance robustness and accuracy achieved by g.

It is observed that the upper bound of r_p decreases with d when $p > 2$. This makes the largest radius r_p that can be certified tiny for high d, and thus results in the failure of the certified robustness against the l_p-norm adversarial attacks ($p > 2$) in high-dimensional space.

In the context of certified watermark, given a base classifier $f(x, w)$, where x is the trigger set and w is the model parameters with the embedded watermark, x remains constant and the adversary can change w for watermark removal attacks. For ease of presentation, we rewrite $f(x, w)$ as $f(w)$ since x is constant. Randomized smoothing aims to build a smooth classifier $g(w)$ from $f(w)$ that outputs the probabilities of data samples on the trigger set x over classes.

Similar to the randomized smoothing for the certified robustness, we have $g(w + \delta) = g(w)$ for $\forall\delta$, $\|\delta\|_p \leq r_p$ for any p ($1 \leq p \leq \infty$), where δ denotes the watermark removal attack. Thus, $g(w)$ can maintain the high accuracy on the trigger set x against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) within a certified radius of r_p, guaranteeing the watermarks to be unremovable under the watermark removal attacks. However, the hardness of the certified watermark in high-dimensional space against the l_p-norm watermark removal attacks ($p > 2$) still remains unsolved.

2.2. Mollifier Theory

In mathematics, mollifiers (also known as approximations to the identity) are smooth functions with special properties, which are used to create a sequence of smooth functions approximating non-smooth (generalized) functions via convolution (Friederichs, 1944). More specifically, given a rather irregular function, by convolving it with a smooth mollifier, we obtain a smooth function which is close to the original non-smooth one, such that the sharp features of the original function are smoothed.

Definition 1. *[Vector Space of Differentiable Function] Let $C^\infty(\mathbb{R}^n)$ denote the set of all infinitely differentiable functions $\pi : \mathbb{R}^n \to \mathbb{R}^n$, then $C^\infty(\mathbb{R}^n)$ is a vector space, using the usual notions of addition and scalar multiplication for functions.*

Definition 2. *[Mollifier] A function $\psi(w) \in C^\infty(\mathbb{R}^n)$ for $w \in \mathbb{R}^n$ is a mollifier if it satisfies the three conditions:*

- *$\psi(w)$ is compactly supported;*
- *$\int_{\mathbb{R}^n} \psi(w)dw = 1$;*
- *$\lim_{\sigma \to 0} \psi_\sigma(w) = \tau(w)$.*

where $\psi_\sigma(w) = \sigma^{-n}\psi(w/\sigma)$ and $\tau(w)$ is the Dirac delta function and the limit must be understood in the space of Schwartz distributions.

When $\psi(w) = \mu(|w|)$ for some infinitely differentiable function $\mu : \mathbb{R}^+ \to \mathbb{R}$, then $\psi(w)$ becomes a radially symmetric mollifier, i.e., $\psi(w_1) = \psi(w_2)$ when $|w_1| = |w_2|$.

Definition 3. *[Mollification] Given a smooth mollifier $\psi(w) \in C^\infty(\mathbb{R}^n)$, a mollification $g(w)$ of a locally integrable function $f(w)$ is defined as the convolution of $f(w)$ and $\psi(w)$.*

$$g(w) = f(w) * \psi_\sigma(w) = \int f(u)\psi_\sigma(w - u)du \tag{5}$$

$g(w)$ contains the following three properties:

- *$g(w)$ is smooth on $C^\infty(\mathbb{R}^n)$;*
- *$g(w)$ converges to $f(w)$ when $\sigma \to 0$, i.e., $\lim_{\sigma \to 0} g(w) = \lim_{\sigma \to 0} f(w) * \psi_\sigma(w) = f(w)$;*
- *If $f(w)$ is continuous, then $g(w) \to f(w)$ uniformly in any compact set when $\sigma \to 0$.*

The mollification is essentially a smoothening operation from $f(w)$ to $g(w)$: $g(w) \in C^\infty(\mathbb{R}^n)$.

The certified watermark problem with mollifier smoothing is defined as producing a dimension-independent certified radius by convolving the base classifier with a smooth mollifier, such that the smooth classifier is close to the base one, for conducting the certified watermark problem against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) for high parameter dimension d.

3. Mollifier Smoothing for Certified Watermarks

In order to deal with the dilemma of certified watermarks in high-dimensional space against the l_p-norm watermark removal attacks ($p > 2$) by randomized smoothing, the idea of this paper is to utilize a well-designed mollifier $\psi(w)$ to get the smooth classifier $g(w)$, i.e., the mollification of the base classifier $f(w)$, make the Lipschitz constant of g independent of d, and thus produce the dimension-independent r_p by g. Thus, our mollifier smoothing method is able to conduct the certified watermark problem against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) for high parameter dimension d.

For the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$), the mollifier $\psi(w)$ for the model parameter $w \in \mathbb{R}^d$ is defined below.

$$\psi(w) = Ke^{-\|w\|_p} \tag{6}$$

where $K = \frac{1}{\int_{\mathbb{R}^d} e^{-\|w\|_p} dw}$ is a chosen constant such that $\int_{\mathbb{R}^d} \psi(w)dw = 1$, i.e., the second condition in Definition 2. We have $\psi_\sigma(w) = \sigma^{-d}\psi(w/\sigma)$ too.

Accordingly, a smooth classifier $g(w)$ is given as follows.

$$g(w) = f(w) * \psi_\sigma(w) \tag{7}$$

where σ is a hyperparameter to balance certified watermark robustness and accuracy achieved by our g. When $\sigma \to 0$, $\psi_\sigma(w)$ converges to the Dirac Delta function $\tau(w)$ and thus $g(w) \to f(w)$. According to Theorem 5, the certified radius r_p by g increases with σ, i.e., the certified watermark robustness increases with σ.

The following theoretical analyses quantify the correctness and applicability of our mollifier smoothing method for certified watermark robustness. Definitions 4, 5, and Lemma 1 are the preparation of the theoretical proofs. Lemma 2 and Theorem 2 validate that l_p-norm ($1 \leq p < \infty$) is Fréchet differentiable and derive the upper bound of its Fréchet derivative. Theorem 3 verifies that l_p-norm ($p = \infty$) is one-sided Gateaux differentiable. Based on the differentiable properties of l_p-norm, Theorem 4 derives the dimension-independent Lipschitz constant of our smooth classifier g. Theorem 5 deduces the dimension-independent certified radius r_p by g, according to the Lipschitz constant.

Definition 4. *[Fréchet Derivative] Let W and U be two Banach spaces and $F(w)$ is a function from W to U. $F(w)$ is Fréchet differentiable at $w \in W$ if there exists a bounded linear operator $DF : W \to U$ such that*

$$\lim_{\|h\|_W \to 0} \frac{\|F(w+h) - F(w) - DF(w)h\|_U}{\|h\|_W} = 0 \tag{8}$$

$DF(w)$ is also called the Fréchet derivative of $F(w)$.

Definition 5. *[Gateaux Derivative] Let W and U be two Banach spaces and $F(w)$ is a function from W to U. The Gateaux derivative $DF(w;h)$ of $F(w)$ at $w \in W$ in the direction $h \in W$ is defined as*

$$DF(w;h) = \frac{d}{dt}F(w+th) = \lim_{t \to 0} \frac{F(w+th) - F(w)}{t} \tag{9}$$

If the limit exists for all $h \in W$, then $F(w)$ is Gateaux differentiable at $w \in W$.

Lemma 1. *If $F : W \to U$ is Fréchet differentiable at w, then $F(w)$ is also Gateaux differentiable at w (Lang, 1995). In addition,*

$$DF(w;h) = DF(w)h \tag{10}$$

The Fréchet derivative is used to generalize the derivative from real-valued functions of a single real variable to functions on Banach spaces. The Gateaux derivative is a generalization of the directional derivative in differential calculus. Lemma 1 assesses the relationship between the Fréchet derivative and the Gateaux derivative (Lang, 1995).

For ease of presentation, we use a function $N_p(w)$ to replace the norm $\|w\|_p$ for model parameter $w \in \mathbb{R}^d$, i.e., $N_p(w) = \|w\|_p$ ($1 \leq p \leq \infty$).

Lemma 2. *The function $N_p(w)$ is Fréchet differentiable for $1 \leq p < \infty$ (Bonic & Frampton, 1966). More specifically,*

- *$N_p(w)$ is infinitely many times Fréchet differentiable when p is even;*
- *$N_p(w)$ is $(p-1)$-times Fréchet differentiable when p is odd.*

Theorem 2. *$\|DN_p(w)\|_{op} \leq 1$ for a function $N_p(w)$ with $w \in \mathbb{R}^d$ and norm p ($1 \leq p < \infty$), where $\|\cdot\|_{op}$ is the operator norm.*

Theorem 3. *Let $w = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^d$ with $w_i > 0$ for $\forall i, 1 \leq i \leq d$ and $w^* = N_\infty(w)$.*

- *If $w^* = w_j$ ($1 \leq j \leq d$) and $w^* > w_i$ for $\forall i, i \neq j$, then*

$$\frac{\partial N_\infty(w)}{\partial w_i} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} . \tag{11}$$

  Namely, $N_p(w)$ is Gateaux differentiable.

- *If there exists $j \neq k$ such that $w^* = w_j = w_k$, then*

$$\frac{\partial N_\infty(w)}{\partial w_i} = 0 \text{ if } w^* \neq w_i, \tag{12}$$

$$\frac{\partial N_\infty(w)}{\partial w_i^+} = 1, \ \frac{\partial N_\infty(w)}{\partial w_i^-} = 0 \text{ if } w^* = w_i. \tag{13}$$

  where $\frac{\partial N_\infty(w)}{\partial w_i^+}$ and $\frac{\partial N_\infty(w)}{\partial w_i^-}$ are the left and right partial derivatives of $N_\infty(w)$ at w_i respectively. In this case, $N_p(w)$ is one-sided Gateaux differentiable.

Theorems 2 and 3 verify the differentiability of l_p-norm. Given an arbitrary base classifier f, the following theorem derives the Lipschitz constant of our smooth classifier g with mollifier smoothing and demonstrates that the Lipschitz constant is independent of the parameter dimension d.

Theorem 4. *Given a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$ and $|f(w)| \leq M$, the Lipschitz constant of the smooth classifier g for $\forall p, 1 \leq p \leq \infty$ is $M\sigma^{-1}$, i.e.,*

$$|g(w+\delta) - g(w)| \leq M\sigma^{-1}\|\delta\|_p \tag{14}$$

Notice that $f(w)$ outputs the probabilities of samples on the trigger set over classes. Therefore, there must exist a upper bound $M \leq 1$ such that $|f(w)| \leq M \leq 1$.

According to the conclusion of Theorem 4, the following theorem derives the certified radius r_p by our proposed smooth classifier g and shows that r_p is irrelevant to the parameter dimension d.

Theorem 5. *Given a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$, $\|f(w)\|_\infty \leq M$, and our smooth classifier g with mollifier smoothing, let $p_A(w)$ and $p_B(w)$ be the probabilities on the most probable class c_A and the runner-up class c_B predicted by f respectively, then $g(w+\delta) = g(w)$ for $\forall \delta$, $\|\delta\|_p \leq r_p$ for any p $(1 \leq p \leq \infty)$, where*

$$r_p = \frac{p_A(w) - p_B(w)}{2}\sigma \tag{15}$$

Please refer to Appendix A.1 for detailed proof of Theorems 2-5.

In this case, the smooth classifier g can always output the correct prediction on the trigger set as long as the perturbation δ (i.e., watermark removal attacks) is within a certified l_p-norm radius of r_p for $p > 0$. Compared with the largest certified radius r_p by the randomized smoothing in Eq.(24), the largest certified radius r_p by our randomized smoothing in Eq.(15) is independent of the parameter dimension d. This demonstrates that our method provides strong certified watermark guarantees against l_p-norm attacks $(1 \leq p \leq \infty)$, even with large d, when using the same σ to balance robustness and accuracy by the smoothed classifiers.

4. Efficient Approximation of Mollifier Smoothing

The randomized smoothing method needs to sample and predict massive points for certifying large radius with high standard confidence. The cost of a large number of sampling and prediction operations is non-trivial. Our mollifier smoothing method consists of many integral calculations. Solving high-dimensional numerical integration has been notoriously difficult due to the curse of dimensionality (Bellman, 1957; Bakhvalov, 1959). In order to further improve the efficiency of the certified watermark, based on the partial differential equation (PDE) theory, this paper introduces a practical algorithm for approximating the mollifier smoothing process in high-dimension space.

The following lemmas and theorem offer the theoretical foundation of the approximation of our smooth classifier g. The Green's second identity captures the relationship between multiple integrals and line integrals (Strauss, 2007). We utilize it to translate the computation of difficult multiple integrals in high-dimensional space into the calculation of straightforward line integrals. Theorem 6 makes use of the following two lemmas to eliminate time-consuming numerical integration and derive the approximation of the mollifier smoothing.

In mathematics, the Green's second identity is an identity in vector calculus relating the bulk with the boundary of a region on which differential operators act (Strauss, 2007).

Lemma 3. *[Green's second identity] If a and b are both twice continuously differentiable functions on $W \subset \mathbb{R}^d$, then*

$$\int_U (a\Delta b - b\Delta a)dx = \oint_{\partial U} (a\frac{\partial b}{\partial \mathbf{n}} - b\frac{\partial a}{\partial \mathbf{n}})dS \tag{16}$$

where $\Delta a = \frac{\partial^2 a}{\partial x_1^2} + \frac{\partial^2 a}{\partial x_2^2} + \cdots + \frac{\partial^2 a}{\partial x_d^2}$ is the Laplacian operator and $\frac{\partial a}{\partial \mathbf{n}}$ is the directional derivative of a in the direction of the outward pointing normal $\mathbf{n}$ to the surface element dS (Strauss, 2007).

Definition 6. *[Harmonic Function] A function $q(w) : Q \subset \mathbb{R}^d \to \mathbb{R}$ is a harmonic function if $q(w)$ is twice continuously differentiable and satisfies Laplace's equation $\Delta q = 0$. Namely,*

$$\frac{\partial^2 q}{\partial w_1^2} + \frac{\partial^2 q}{\partial w_2^2} + \cdots + \frac{\partial^2 q}{\partial w_d^2} = 0 \tag{17}$$

everywhere on Q (Axler et al., 2001).

Lemma 4. *[Mean Value Property of Harmonic Functions] Let $B(w, \sigma)$ be a ball centering at x with radius σ in $\mathbb{R}^d$. If $q \in C^2(Q)$ and $B(w, \sigma) \subset Q$, then it holds that*

$$q(w) = \frac{1}{\omega_d \sigma^d}\int_{B(w,\sigma)} q(u)du = \frac{1}{d\omega_d \sigma^{d-1}}\oint_{\partial B(w,\sigma)} q(u)dS \tag{18}$$

where ω_d is the volume of the unit ball in $\mathbb{R}^d$ and u is the $(d-1)$-dimensional surface measure (Axler et al., 2001).

By the mean value property of harmonic function, one can see that, if a function f is harmonic, then $f * \psi = f$ for any radially symmetric mollifier ψ. In fact

$$\begin{aligned}\int f(w-u)\psi(u)du &= \int_0^\infty \int_{\partial B(0,r)} f(w-u)\psi(u)dSdr \\ &= \int_0^r f(w)\psi(|r|)dr = f(w)\int_0^r \psi(|r|)dr = f(w)\end{aligned} \tag{19}$$

Table 1: Certified accuracy (%) of embedded content watermark on CIFAR-10 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 1.00$	Certified_Watermarks	80	55	34	19	3	0	0	0
	Randomized Smoothing	**100**	96	35	2	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**54**	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	54	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	98	80
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.50$	Certified_Watermarks	**100**	**100**	**100**	**100**	98	97	94	89
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 2: Certified accuracy (%) of embedded content watermark on CIFAR-10 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	80	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**9**	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**56**	0

Eq.(19) implies that if a base classifier f is harmonic, then the smooth one g is itself. Therefore, it is naturally certified-watermark robust. We leverage this observation regarding the mean value property of harmonic function to prove Theorem 6 and generate a high-quality approximation of our g.

In this section, we use the following symmetric mollifier $\psi(w) \in C^\infty(\mathbb{R}^d)$ for the approximate mollification $\hat{g}(w)$ of a base classifier $f(w)$.

$$\psi(w) = \begin{cases} Ke^{-1/(1-|w|^2)}, & \text{if } |w| < 1, \\ 0, & \text{if } |w| \geq 1. \end{cases} \tag{20}$$

where $K = \frac{1}{\int_{\mathbb{R}^d} \psi(w)dw}$ is a chosen constant such that $\int_{\mathbb{R}^d} \psi(w)dw = 1$.

We also have $\psi_\sigma(w) = \sigma^{-d}\psi(w/\sigma)$. For ease of representation, we use a function $\mu(|w|)$ to denote the symmetric representation of $\psi_\sigma(w)$, i.e., $\psi_\sigma(w) = \mu(|w|)$.

Theorem 6. *Assuming that a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$ is continuous everywhere, $f(w)$ is C^∞-smooth, and the second derivatives of $f(w)$ are all zero almost everywhere except on a measure zero set Γ, where Γ consists of several planes, then the mollification of $f(w)$*

$$g(w) = \int_{B(w,\sigma)} f(u)\psi_\sigma(w-u)du \tag{21}$$

can be approximated as

$$g(w) \approx \hat{g}(w) = d\omega_d\sigma^{d-1}\nu'(\sigma)f(w) - \omega_d\sigma^{d-1}\alpha\nabla f(w) \tag{22}$$

where α is a hyperparameter vector that is used to balance the accuracy of approximation. $\phi_\sigma(w)$ is the solution of the Poisson equation $\Delta\phi_\sigma(w) = \psi_\sigma(w)$ and $\nu(|w|)$ is the symmetric representation of $\phi_\sigma(w)$, i.e., $\phi_\sigma(w) = \nu(|w|)$. Please refer to Appendix A.1 for detailed proof.

Since Eq.(84) and Theorem 6 give an explicit solution of $g(w)$, this helps avoid time-consuming numerical integration.

5. Experimental Evaluation

In this section, we have evaluated the empirical and certified defense of our Mollifier Smoothing model and other comparison methods against l_2, l_3, and l_∞-norm watermark removal attacks over three standard image classification datasets: MNIST (Deng, 2012), CIFAR-10 (Krizhevsky, 2009), and CIFAR-10 (Krizhevsky, 2009). The experiments exactly follow the same settings described by the original paper of Certified_Watermarks (Bansal et al., 2022). We train the classifiers on the training set and test them on the test set for three datasets. We train a convolutional neural network (CNN) on MNIST for handwritten digit recognition. We apply ResNet-20 and ResNet-18 over CIFAR-10 and CIFAR-100 for image classification respectively.

Table 3: Empirical accuracy (%) on CIFAR-10 under three attack methods

Method/Attack	Finetuning	Hard-Label Distillation	Soft-Label Distillation
Certified_Watermarks	**100**	98	**100**
PTYNet	32	68	70
SAC	57	55	0
CosWM	33	55	78
RPV	73	67	**100**
Watermark Embedding	14	29	81
DeepMarks	71	80	99
NO-stealing-LTH	15	18	31
Mollifier Smoothing	**100**	**100**	**100**

Table 4: Certified accuracy (%) by Mollifier Smoothing variants on CIFAR-10 under two l_p-norm attacks

Watermark	Norm	Noise Level σ	1.00	1.25	1.50	1.75
Embedded Content	l_2	Mollifier Smoothing-A	**100**	**100**	90	56
		Mollifier Smoothing	**100**	**100**	**100**	**100**
Noise	l_2	Mollifier Smoothing-A	**100**	**100**	17	92
		Mollifier Smoothing	**100**	**100**	**100**	**100**
Unrelated	l_2	Mollifier Smoothing-A	**100**	**100**	**100**	97
		Mollifier Smoothing	**100**	**100**	**100**	**100**
Embedded Content	l_∞	Mollifier Smoothing-A	**100**	99	98	64
		Mollifier Smoothing	**100**	**100**	**100**	**100**
Noise	l_∞	Mollifier Smoothing-A	99	**100**	79	18
		Mollifier Smoothing	**100**	**100**	**100**	**100**
Unrelated	l_∞	Mollifier Smoothing-A	**100**	**100**	**100**	96
		Mollifier Smoothing	**100**	**100**	**100**	**100**

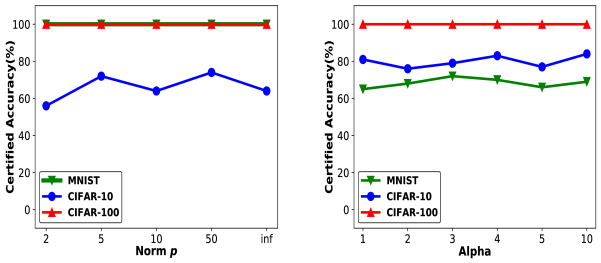
Baselines. We compare the Mollifier Smoothing model with two representative certified defense and seven state-of-the-art empirical defense models. **Randomized Smoothing** is the original randomized smoothing method, which certifies robustness against l_2-norm adversarial attacks by smoothing with the isotropic Gaussian distribution (Cohen et al., 2019). We modified and applied it to the setting of certified defense against watermark removal attacks. **Certified_Watermarks** is the first to provide a certified defense against l_2-norm watermark removal attacks, by leveraging the randomized smoothing techniques (Cohen et al., 2019; Chiang et al., 2020) for certified robustness to adversarial attacks (Bansal et al., 2022). **PTYNet** is a plug-and-play watermarking scheme for DNN models by injecting an independent proprietary model into the target model to serve the watermark embedding and ownership verification. (Wang et al., 2022). **SAC** is a model stealing detection method based on SAmple Correlation, by considering the pairwise relationship between samples (Guan et al., 2022). **CosWM** is a watermarking technique named CosWM to achieve outstanding model watermarking performance against ensemble distillation by embedding a watermark as a cosine signal within the output of a teacher model (Charette et al., 2022). **Reducing Parametric Vulnerability (RPV)** is a minimax formulation to find existing watermark-removed models in the vicinity and recover their watermark behaviors (Anonymous, 2023). **Watermark Embedding** is a watermark implanting approach to infuse watermark into deep learning models, and designs a remote verification mechanism to determine the model ownership (Zhang et al., 2018). **DeepMarks** is the first end-to-end collusion-secure fingerprinting framework that enables the owner to retrieve model authorship information and identification of unique users in the context of deep learning (Chen et al., 2019). **NO-stealing-LTH** is a mask embedding method to embed ownership signatures into lottery tickets' typologies without much affecting its performance. (Chen et al., 2021a). Existing efforts focus on certified watermarks against l_2-norm watermark removal attacks. To our best knowledge, this work is the first to conduct the certified watermark problem in high-dimensional space against l_p-norm watermark removal attacks ($1 \leq p \leq \infty$).

Attack methods. By following the same options in the paper of Certified_Watermarks (Bansal et al., 2022), we evaluate the performance of certified watermarks with three representative watermark attack methods: hard-label distillation, soft-label distillation, and finetuning. Three attack methods have been shown to be very effective in (Shafieinejad et al., 2021; Aiken et al., 2021). In the distillation threat model, the adversary initializes their watermarking model with benign model, and then trains their model with distillation using unlabeled data that comes from the same distribution. Soft-label distillation takes the probability distribution of the original model as labels, whereas hard-label distillation takes only the label with maximum probability. In the finetuning threat model, the adversary has its own labeled dataset from the original data-generating distribution. This adversary is strictly stronger compared to the distillation threat model.

Variants of Mollifier Smoothing model. We evaluate two versions of Mollifier Smoothing to show the strengths of different techniques. Mollifier Smoothing leverages the mollifier theory and numerical integration for the certified defense against l_p-norm watermark removal attacks. Mollifier Smoothing-A utilizes the PDE theory to generate an approximation of Mollifier Smoothing with better efficiency.

Evaluation metrics. We use a popular measure in certified watermark to verify the performance of different methods: **certified accuracy** at radius r_p (Bansal et al., 2022). It is defined as the certified accuracy on the trigger set with watermark removal attacks on the model. We also report **test accuracy** (i.e., the accuracy on the test dataset) (Bansal et al., 2022). A larger certified accuracy or test accuracy indicates a better result.

Certified accuracy against l_2 attacks Table 1 exhibits the certified accuracy obtained by three certified watermark methods with varying noise level σ and radius r_p based on the embedded content watermark on CIFAR-10. It is



(a) Norm p (b) Approx. parameter α

Figure 1: Certified accuracy with varying parameters

observed that among three approaches, no matter how large the radii are, the Mollifier Smoothing method achieves the highest certified accuracy in most tests, showing the effectiveness of Mollifier Smoothing to the certified watermark. Compared to the highest certified accuracy by other certified watermark methods, Mollifier Smoothing, on average, achieves at least 10.6% improvement of certified accuracy on MNIST. In addition, the promising performance of Mollifier Smoothing over MNIST, CIFAR-10, and CIFAR-100 datasets (Please see Appendix A.2 for the experiments on other datasets) implies that Mollifier Smoothing has great potential as a general certified watermark solution to other image datasets, which is desirable in practice.

Certified accuracy against l_∞ attacks. Table 2 show the certified accuracy against l_∞ attacks by varying r_p and σ respectively. We have observed Mollifier Smoothing substantially improves the certified accuracy, but the certified accuracy of most baseline methods quickly drop to zero. This demonstrates that the randomized smoothing methods are hard to achieve very promising certified accuracy against large-norm attacks. In addition, the certified accuracy by our Mollifier Smoothing is less sensitive to the radius. This shows that our mollifier smoothing method can certify the watermark in both high and low-dimensional space against large-norm attacks.

Empirical accuracy against watermark removal attacks. Table 3 presents the empirical accuracy with two certified defense and seven empirical defense approaches against three watermark removal attacks of hard-label distillation, soft-label distillation, and finetuning. It is obvious that Mollifier Smoothing achieves the highest empirical accuracy (100%), which is much better than the other eight methods under all three attack methods in most tests. Especially, on the CIFAR-100 dataset, our Mollifier Smoothing method significantly outperforms Certified_Watermarks on MNIST in Table 32. This demonstrates the superior performance of our Mollifier Smoothing method in both certified defense and empirical defense.

Ablation study. Table 4 exhibits the certified accuracy by two variants of the Mollifier Smoothing model with varying noise level on CIFAR-10 against l_2 and l_∞ attacks.

Table 5: Runtime (seconds) of Certified Watermark

Method/Dataset	MNIST	CIFAR-10	CIFAR-100
Randomized Smoothing	208	2,200	2,503
Certified_Watermark	294	347	443
Mollifier Smoothing	206	400	451
Mollifier Smoothing-A	**45**	**150**	**343**

We have observed the exact Mollifier Smoothing achieves the highest certified accuracy (100%) for three watermark schemes, while Mollifier Smoothing-A also obtains good performance. This shows Mollifier Smoothing-A utilize the partial differential equation theory to generate an approximation of Mollifier Smoothing with better efficiency while maintaining the quality.

Impact of norm p and approximation parameter α. Figures 1 (a) and (b) measure the performance effect of p and α in the certified accuracy of our Mollifier Smoothing model. It is observed that when increasing p and α, the certified accuracy of the Mollifier Smoothing model keeps relatively stable. This demonstrates that our mollifier smoothing method derives a dimension/norm-independent certified watermark solution, such that it can certify the watermarks against l_p-norm watermark removal attacks for large d and any p. α is used to balance the the accuracy of approximation in the Mollifier Smoothing-A model. A reasonable explanation is that α is associated with $\nabla f(w)$ in the second term Eq.(22), instead of $f(w)$ itself. However, $f(w)$ is often larger than $\nabla f(w)$ in most cases. This substantially decreases the impact of α towards the certified robustness by Mollifier Smoothing-A.

Running time. Table 5 reports the running time achieved by two comparison methods and two versions of our Mollifier Smoothing over three datasets respectively. Mollifier Smoothing scales well with different image datasets and shows good efficiency for certified watermarks. Notice that Mollifier Smoothing is comparable to the Certified_Watermarks method but achieves significantly better accuracy than the latter. In addition, Mollifier Smoothing-A performs the best in all experiments. A reasonable explanation is that Mollifier Smoothing-A is able to directly generate a smooth classifier without time-consuming noise sampling and integral calculation.

6. Related Work

Watermark Removal Attacks. Trustworthy machine learning has attracted active research in recent years (Palanisamy et al., 2018; Zhou et al., 2020; Zhang et al., 2020; Wu et al., 2021; Zhou et al., 2021; Zhao et al., 2021; Ren et al., 2021; Zhang et al., 2021c;b;a; Liu et al., 2022a; Zhou et al., 2022; Jin et al., 2022b; Zhang et al., 2022; Jin et al., 2022a; Su et al., 2013; Zhou & Liu, 2013; Su et al., 2015; Bao et al., 2015; Zhou et al., 2015; Zhou, 2017; Zhou et al., 2018b;a; Zhou & Liu, 2019). Water-

marking techniques are proposed to protect the intellectual property of machine learning models. Adversaries attempt to subvert watermarking mechanisms by designing watermark removal attacks, which have attracted active research in recent years. Two recent works show that by adding regularization, fine-tuning and pruning, their watermarks can be removed without compromising the prediction accuracy of the stolen model (Aiken et al., 2021; Shafieinejad et al., 2021). A recent study show that the watermark signals embedded by (Uchida et al., 2017) can be easily detected and overwritten (Wang & Kerschbaum, 2019). Yang et al. proposed a distillation attack technique to remove watermark embedded by existing algorithms (Yang et al., 2019). WILD is a backdoor-based watermark removal framework using only a small portion of training data (Liu et al., 2020). REFIT is a unified watermark removal framework based on fine-tuning, which is agnostic to watermark embedding schemes (Chen et al., 2021b). Guo et al. designed a simple yet powerful transformation algorithm by combining imperceptible pattern embedding and spatial-level transformations, which can effectively and blindly destroy the memorization of watermarked mod- els to the watermark samples (Guo et al., 2021). Attention Distraction is a source data-free watermark removal attack, to make the model selectively forget the embedded watermarks by customizing continual learning (Zhong et al., 2022). Yan et al. presentED an effective removal attack which cracks almost all the existing white-box watermarking schemes with provably no performance overhead and no required prior knowledge (Yan et al., 2022).

Defense against Watermark Removal Attacks. Existing techniques on robust watermarking can be broadly classified into two categories below.

(1) Empirical Defenses. Empirical defense is gaining attention in recent years (Tartaglione et al., 2020; Sun et al., 2021). Several early works have been proposed for building empirically robust watermarking models that are resistant to watermark removal attacks, such as fine-tuning (Uchida et al., 2017; Zhang et al., 2018; Adi et al., 2018; Rouhani et al., 2019). Namba and Sakuma proposed a novel robust watermark method for neural networks that resists against both model modification and query modification (Namba & Sakuma, 2019). RIGA is a white-box watermarking algorithm whose watermark extracting function is also a DNN and which is trained using an adversarial network (Wang & Kerschbaum, 2021). Yang et al. proposed a bi-level optimization framework that optimizes the robustness of the embedded watermarks (Yang et al., 2021). Characteristic Examples effectively fingerprint deep neural networks, featuring high-robustness to the base model against model pruning as well as low-transferability (Wang et al., 2021). Watermark Vaccine is a novel defense mechanism by injecting a watermark-agnostic perturbation on host images before adding watermark just like vaccination in reality (Liu et al., 2022b). PMI is a pooled membership inference technique to modify the DNN parameters through an imperceptible way, which increases the difficulty of locating and further removing the watermark (Wu, 2022). TATTOOED is a robust and efficient DNN watermarking technique based on spread-spectrum channel coding against watermark removals (Pagnotta et al., 2022). RPV is a minimax formulation to find existing watermark-removed models in the vicinity and recover their watermark behaviors (Anonymous, 2023).

(2) Certified Defenses. Certified defense techniques provide guarantees that the watermark is guaranteed to be unremovable under the watermark removal attacks. Goldberger et al. proposed to find the minimal modification required to remove watermark in a neural network (Goldberger et al., 2020). They did not propose methods to embed a watermark that would be more resilient, rather they simply find the minimal change required to remove a watermark. In addition, their approach is based on solving mixed integer linear programs and thus does not scale well to larger networks. Certified_Watermarks (Bansal et al., 2022) is the first to provide a certified defense against l_2-norm watermark removal attacks, by leveraging the randomized smoothing techniques (Cohen et al., 2019; Chiang et al., 2020) for certified robustness to adversarial attacks. As shown in the above analysis of randomized smoothing for certified robustness, the randomized smoothing methods suffer from hardness of certified robustness in high-dimensional space against l_p-norm attacks ($p > 2$), especially l_∞-norm attacks. Thus, Certified_Watermarks based on the randomized smoothing techniques fails to provide a meaningful certificate in high-dimensional space against the l_p-norm watermark removal attacks ($p > 2$).

To our best knowledge, this work is the first to leverage mollifier theory and partial differential equation theory for conducting the certified watermark problem in high-dimensional parameter space against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) with better applicability.

7. Conclusions

In this work, we have studied the certified watermark problem against l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) for high d. First, we proposed a mollifier smoothing method to certify the watermarks against l_p-norm attacks, with the dimension-independent Lipschitz constant of the proposed smooth classifier. Second, we developed an approximation of mollifier smoothing for avoiding time-consuming noise sampling and numerical integration. Finally, our mollifier smoothing model achieves superior certified watermark performance against several representative algorithms, in terms of both effectiveness and efficiency.

References

Adi, Y., Baum, C., Cissé, M., Pinkas, B., and Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In Enck, W. and Felt, A. P. (eds.), *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pp. 1615–1631. USENIX Association, 2018.

Aiken, W., Kim, H., Woo, S. S., and Ryoo, J. Neural network laundering: Removing black-box backdoor watermarks from deep neural networks. *Comput. Secur.*, 106:102277, 2021.

Anonymous. Towards robust model watermark via reducing parametric vulnerability. In *Submission to 11th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, Conference Track Proceedings*, 2023.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 274–283, 2018.

Axler, S., Bourdon, P., and Ramey, W. *Harmonic Function Theory*. Wiley, 2001.

Bakhvalov, N. S. On approximate computation of integrals. *Vestnik MGU Ser. Math. Mech. Astron. Phys. Chem (In Russian)*, 4:3–18, 1959.

Bansal, A., Chiang, P., Curry, M. J., Jain, R., Wigington, C., Manjunatha, V., Dickerson, J. P., and Goldstein, T. Certified neural network watermarks with randomized smoothing. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1450–1465. PMLR, 2022.

Bao, X., Liu, L., Xiao, N., Zhou, Y., and Zhang, Q. Policy-driven autonomic configuration management for nosql. In *Proceedings of the 2015 IEEE International Conference on Cloud Computing (CLOUD'15)*, pp. 245–252, New York, NY, June 27-July 2 2015.

Bellman, R. E. *Dynamic programming*. Princeton Univisity Press, 1957.

Blum, A., Dick, T., Manoj, N., and Zhang, H. Random smoothing might be unable to certify $l\infty$ robustness for high-dimensional images. *J. Mach. Learn. Res.*, 21:211:1–211:21, 2020.

Bonic, R. and Frampton, J. Smooth functions on banach manifolds. *Journal of Mathematics and Mechanics*, 15 (5):877–898, 1966.

Carlini, N. and Wagner, D. A. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pp. 3–14, 2017.

Charette, L., Chu, L., Chen, Y., Pei, J., Wang, L., and Zhang, Y. Cosine model watermarking against ensemble distillation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 9512–9520. AAAI Press, 2022.

Chen, H., Rouhani, B. D., Fu, C., Zhao, J., and Koushanfar, F. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In El-Saddik, A., Bimbo, A. D., Zhang, Z., Hauptmann, A. G., Candan, K. S., Bertini, M., Xie, L., and Wei, X. (eds.), *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019*, pp. 105–113. ACM, 2019.

Chen, X., Chen, T., Zhang, Z., and Wang, Z. You are caught stealing my winning lottery ticket! making a lottery ticket claim its ownership. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 1780–1791, 2021a.

Chen, X., Wang, W., Bender, C., Ding, Y., Jia, R., Li, B., and Song, D. Refit: a unified watermark removal framework for deep learning systems with limited data. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp. 321–335, 2021b.

Chiang, P., Curry, M. J., Abdelkader, A., Kumar, A., Dickerson, J., and Goldstein, T. Detection as regression: Certified object detection with median smoothing. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In

Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pp. 1310–1320, 2019.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, 2009.

Deng, L. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.*, 29(6):141–142, 2012.

Friederichs, K. O. The identity of weak and strong extensions of differential operators. *Transactions of the American Mathematical Society*, 55(1):132–151, 1944.

Goldberger, B., Katz, G., Adi, Y., and Keshet, J. Minimal modifications of deep neural networks using verification. In Albert, E. and Kovács, L. (eds.), *LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Alicante, Spain, May 22-27, 2020*, volume 73 of *EPiC Series in Computing*, pp. 260–278. EasyChair, 2020.

Guan, J., Liang, J., and He, R. Are you stealing my model? sample correlation for fingerprinting deep neural networks. *CoRR*, abs/2210.15427, 2022.

Guo, S., Zhang, T., Qiu, H., Zeng, Y., Xiang, T., and Liu, Y. Fine-tuning is not enough: A simple yet effective watermark removal attack for DNN models. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 3635–3641. ijcai.org, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034, 2015.

Jin, J., Ren, J., Zhou, Y., Lv, L., Liu, J., and Dou, D. Accelerated federated learning with decoupled adaptive optimization. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pp. 10298–10322, Baltimore, MD, July 17-23 2022a.

Jin, J., Zhang, Z., Zhou, Y., and Wu, L. Input-agnostic certified group fairness via gaussian parameter smoothing. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pp. 10340–10361, Baltimore, MD, July 17-23 2022b.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Kumar, A., Levine, A., Goldstein, T., and Feizi, S. Curse of dimensionality on randomized smoothing for certifiable robustness. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pp. 5458–5467, 2020.

Lang, S. *Differential and Riemannian Manifolds*. Springer, 1995.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pp. 656–672, 2019.

Lee, G., Yuan, Y., Chang, S., and Jaakkola, T. S. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 4911–4922, 2019.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4585–4593, 2020.

Levine, A., Kumar, A., Goldstein, T., and Feizi, S. Tight second-order certificates for randomized smoothing. *CoRR*, abs/2010.10549, 2020.

Li, B., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. *CoRR*, abs/1809.03113, 2018.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 9459–9469, 2019.

Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., and Dou, D. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems (KAIS)*, 64(4):885–917, 2022a.

Liu, X., Li, F., Wen, B., and Li, Q. Removing backdoor-based watermarks in neural networks with limited data. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 10149–10156. IEEE, 2020.

Liu, X., Liu, J., Bai, Y., Gu, J., Chen, T., Jia, X., and Cao, X. Watermark vaccine: Adversarial attacks to prevent watermark removal. In Avidan, S., Brostow, G. J., Cissé, M., Farinella, G. M., and Hassner, T. (eds.), *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV*, volume 13674 of *Lecture Notes in Computer Science*, pp. 1–17. Springer, 2022b.

Mohapatra, J., Ko, C., Weng, T., Chen, P., Liu, S., and Daniel, L. Higher-order certification for randomized smoothing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Namba, R. and Sakuma, J. Robust watermarking of neural network with exponential weighting. In Galbraith, S. D., Russello, G., Susilo, W., Gollmann, D., Kirda, E., and Liang, Z. (eds.), *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, AsiaCCS 2019, Auckland, New Zealand, July 09-12, 2019*, pp. 228–240. ACM, 2019.

Pagnotta, G., Hitaj, D., Hitaj, B., Pérez-Cruz, F., and Mancini, L. V. TATTOOED: A robust deep neural network watermarking scheme based on spread-spectrum channel coding. *CoRR*, abs/2202.06091, 2022.

Palanisamy, B., Liu, L., Zhou, Y., and Wang, Q. Privacy-preserving publishing of multilevel utility-controlled graph datasets. *ACM Transactions on Internet Technology (TOIT)*, 18(2):24:1–24:21, 2018.

Ren, J., Zhang, Z., Jin, J., Zhao, X., Wu, S., Zhou, Y., Shen, Y., Che, T., Jin, R., and Dou, D. Integrated defense for resilient graph matching. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pp. 8982–8997, Virtual Event, July 18-24 2021.

Rouhani, B. D., Chen, H., and Koushanfar, F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In Bahar, I., Herlihy, M., Witchel, E., and Lebeck, A. R. (eds.), *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2019, Providence, RI, USA, April 13-17, 2019*, pp. 485–497. ACM, 2019.

Salman, H., Li, J., Razenshteyn, I. P., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 11289–11300, 2019.

Shafieinejad, M., Lukas, N., Wang, J., Li, X., and Kerschbaum, F. On the robustness of backdoor-based watermarking in deep neural networks. In Borghys, D., Bas, P., Verdoliva, L., Pevný, T., Li, B., and Newman, J. (eds.), *IH&MMSec '21: ACM Workshop on Information Hiding and Multimedia Security, Virtual Event, Belgium, June, 22-25, 2021*, pp. 177–188. ACM, 2021.

Strauss, W. A. *Partial Differential Equations: An Introduction, 2nd Edition*. Wiley, 2007.

Su, Z., Liu, L., Li, M., Fan, X., and Zhou, Y. Servicetrust: Trust management in service provision networks. In *Proceedings of the 10th IEEE International Conference on Services Computing (SCC'13)*, pp. 272–279, Santa Clara, CA, June 27-July 2 2013.

Su, Z., Liu, L., Li, M., Fan, X., and Zhou, Y. Reliable and resilient trust management in distributed service provision networks. *ACM Transactions on the Web (TWEB)*, 9(3): 1–37, 2015.

Sun, S., Wang, H., Xue, M., Zhang, Y., Wang, J., and Liu, W. Detect and remove watermark in deep neural networks via generative adversarial networks. In Liu, J. K., Katsikas, S. K., Meng, W., Susilo, W., and Intan, R. (eds.), *Information Security - 24th International Conference, ISC 2021, Virtual Event, November 10-12, 2021, Proceedings*, volume 13118 of *Lecture Notes in Computer Science*, pp. 341–357. Springer, 2021.

Tartaglione, E., Grangetto, M., Cavagnino, D., and Botta, M. Delving in the loss landscape to embed robust watermarks into neural networks. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 1243–1250. IEEE, 2020.

Teng, J., Lee, G.-H., and Yuan, Y. l_1 adversarial robustness certificates: a randomized smoothing approach. *OpenReview*, 2019.

Tramèr, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Uchida, Y., Nagai, Y., Sakazawa, S., and Satoh, S. Embedding watermarks into deep neural networks. In Ionescu, B., Sebe, N., Feng, J., Larson, M. A., Lienhart, R., and Snoek, C. (eds.), *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR 2017, Bucharest, Romania, June 6-9, 2017*, pp. 269–277. ACM, 2017.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against

weak attacks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 5032–5041, 2018.

Wang, R., Ren, J., Li, B., She, T., Lin, C., Fang, L., Chen, J., Shen, C., and Wang, L. Free fine-tuning: A plug-and-play watermarking scheme for deep neural networks. *CoRR*, abs/2210.07809, 2022.

Wang, S., Wang, X., Chen, P., Zhao, P., and Lin, X. Characteristic examples: High-robustness, low-transferability fingerprinting of neural networks. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 575–582. ijcai.org, 2021.

Wang, T. and Kerschbaum, F. Attacks on digital watermarks for deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 2622–2626. IEEE, 2019. doi: 10.1109/ICASSP.2019.8682202. URL https://doi.org/10.1109/ICASSP.2019.8682202.

Wang, T. and Kerschbaum, F. RIGA: covert and robust white-box watermarking of deep neural networks. In Leskovec, J., Grobelnik, M., Najork, M., Tang, J., and Zia, L. (eds.), *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 993–1004. ACM / IW3C2, 2021.

Wu, H. Robust and lossless fingerprinting of deep neural networks via pooled membership inference. *CoRR*, abs/2209.04113, 2022.

Wu, S., Li, Y., Zhang, D., Zhou, Y., and Wu, Z. Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended topic fact. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*, pp. 3766–3772, Online, January 7-15 2021.

Yan, Y., Pan, X., Wang, Y., Zhang, M., and Yang, M. Cracking white-box DNN watermarks via invariant neuron transforms. *CoRR*, abs/2205.00199, 2022.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I. P., and Li, J. Randomized smoothing of all shapes and sizes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pp. 10693–10705, 2020.

Yang, P., Lao, Y., and Li, P. Robust watermarking for deep neural networks via bi-level optimization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 14821–14830. IEEE, 2021.

Yang, Z., Dang, H., and Chang, E. Effectiveness of distillation attack and countermeasure on neural network watermarking. *CoRR*, abs/1906.06046, 2019.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C., and Wang, L. MACER: attack-free and scalable robust training via maximizing certified radius. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Zhang, G., Zhou, Y., Wu, S., Zhang, Z., and Dou, D. Cross-lingual entity alignment with adversarial kernel embedding and adversarial knowledge translation. *CoRR*, abs/2104.07837, 2021a.

Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M. P., Huang, H., and Molloy, I. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 159–172, 2018.

Zhang, Z., Zhang, Z., Zhou, Y., Shen, Y., Jin, R., and Dou, D. Adversarial attacks on deep graph matching. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 (NeurIPS'20)*, Virtual, December 6-12 2020.

Zhang, Z., Jin, J., Zhang, Z., Zhou, Y., Zhao, X., Ren, J., Liu, J., Wu, L., Jin, R., and Dou, D. Validating the lottery ticket hypothesis with inertial manifold theory. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021 (NeurIPS'21)*, pp. 30196–30210, Virtual, December 6-14 2021b.

Zhang, Z., Zhang, Z., Zhou, Y., Wu, L., Wu, S., Han, X., Dou, D., Che, T., and Yan, D. Adversarial attack against cross-lingual knowledge graph alignment. In *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pp. 5320–5337, Virtual Event / Punta Cana, Dominican Republic, November 7-11 2021c.

Zhang, Z., Zhou, Y., Zhao, X., Che, T., and Lyu, L. Prompt certified machine unlearning with randomized gradient smoothing and quantization. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS'22)*, New Orleans, LA, November 28-December 9 2022.

Zhao, X., Zhang, Z., Zhang, Z., Wu, L., Jin, J., Zhou, Y., Jin, R., Dou, D., and Yan, D. Expressive 1-lipschitz neural networks for robust multiple graph learning against

adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pp. 12719–12735, Virtual Event, July 18-24 2021.

Zhong, Q., Zhang, L. Y., Hu, S., Gao, L., Zhang, J., and Xiang, Y. Attention distraction: Watermark removal through continual learning with selective forgetting. In *IEEE International Conference on Multimedia and Expo, ICME 2022, Taipei, Taiwan, July 18-22, 2022*, pp. 1–6. IEEE, 2022.

Zhou, Y. *Innovative Mining, Processing, and Application of Big Graphs*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2017.

Zhou, Y. and Liu, L. Social influence based clustering of heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'13)*, pp. 338–346, Chicago, IL, August 11-14 2013.

Zhou, Y. and Liu, L. Approximate deep network embedding for mining large-scale graphs. In *Proceedings of the 2019 IEEE International Conference on Cognitive Machine Intelligence (CogMI'19)*, pp. 53–60, Los Angeles, CA, December 12-14 2019.

Zhou, Y., Liu, L., Lee, K., Pu, C., and Zhang, Q. Fast iterative graph computation with resource aware graph parallel abstractions. In *Proceedings of the 24th ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC'15)*, pp. 179–190, Portland, OR, June 15-19 2015.

Zhou, Y., Amimeur, A., Jiang, C., Dou, D., Jin, R., and Wang, P. Density-aware local siamese autoencoder network embedding with autoencoder graph clustering. In *Proceedings of the 2018 IEEE International Conference on Big Data (BigData'18)*, pp. 1162–1167, Seattle, WA, December 10-13 2018a.

Zhou, Y., Wu, S., Jiang, C., Zhang, Z., Dou, D., Jin, R., and Wang, P. Density-adaptive local edge representation learning with generative adversarial network multi-label edge classification. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM'18)*, pp. 1464–1469, Singapore, November 17-20 2018b.

Zhou, Y., Ren, J., Dou, D., Jin, R., Zheng, J., and Lee, K. Robust meta network embedding against adversarial attacks. In *Proceedings of the 20th IEEE International Conference on Data Mining (ICDM'20)*, pp. 1448–1453, Sorrento, Italy, November 17-20 2020.

Zhou, Y., Zhang, Z., Wu, S., Sheng, V., Han, X., Zhang, Z., and Jin, R. Robust network alignment via attack signal scaling and adversarial perturbation elimination. In *Proceedings of the 30th Web Conference (WWW'21)*, pp. 3884–3895, Virtual Event / Ljubljana, Slovenia, April 19-23 2021.

Zhou, Y., Ren, J., Jin, R., Zhang, Z., Zheng, J., Jiang, Z., Yan, D., and Dou, D. Unsupervised adversarial network alignment with reinforcement learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(3): 50:1–50:29, 2022.

A. Appendix

A.1. Proof of Theorems

Theorem 1. *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$. Let g be defined as in (1). Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:*

$$\mathbb{P}\left(f(x+\varepsilon) = c_A\right) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{23}$$

Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2}\left(\Phi^{-1}\left(\underline{p_A}\right) - \Phi^{-1}\left(\overline{p_B}\right)\right) \tag{24}$$

Proof. Please refer to the paper (Cohen et al., 2019) for detailed proof.

Lemma 1. *If $F : W \to U$ is Fréchet differentiable at w, then $F(w)$ is also Gateaux differentiable at w. In addition,*

$$DF(w;h) = DF(w)h \tag{25}$$

Proof. Please refer to the book (Lang, 1995) for detailed proof.

Lemma 2. *The function $N_p(w)$ is Fréchet differentiable for $1 \leq p < \infty$. More specifically,*

- *$N_p(w)$ is infinitely many times Fréchet differentiable when p is even;*
- *$N_p(w)$ is $(p-1)$-times Fréchet differentiable when p is odd.*

Proof. Please refer to the paper (Bonic & Frampton, 1966) for detailed proof.

Theorem 2. *$\|DN_p(w)\|_{op} \leq 1$ for a function $N_p(w)$ with $w \in \mathbb{R}^d$ and norm p $(1 \leq p < \infty)$, where $\|\cdot\|_{op}$ is the operator norm.*

Proof. According to the conclusion of Lemma 2, $N_p(w)$ is Fréchet differentiable for $1 \leq p < \infty$. Based on the definition of the Fréchet derivative, when $\|h\|_p \to 0$, we have

$$\frac{|N_p(w+h) - N_p(w) - DN_p(w)h|}{\|h\|_p} = 0 \tag{26}$$

In terms of the reverse triangle inequality of l_p norm, we get

$$|N_p(w+h) - N_p(w)| \leq \|h\|_p \tag{27}$$

Without loss of generality, we assume that $\bar{h}$ is a direction such that $\|DN_p(w)\|_{op}$ achieves its operator norm, i.e.,

$$\|DN_p(w)\|_{op} = \sup_{h \in \mathbb{R}^d} \frac{\|DN_p(w)h\|_p}{\|h\|_p} = \frac{|DN_p(w)\bar{h}|}{\|\bar{h}\|_p} \tag{28}$$

Notice the linearity of $DN_p(w)$, we have

$$\frac{|DN_p(w)\bar{h}|}{\|\bar{h}\|_p} = \frac{|DN_p(w)\lambda\bar{h}|}{\|\lambda\bar{h}\|_p} \text{ for any scalar} \lambda > 0 \tag{29}$$

Therefore, we can always select a small enough λ such that $\|\lambda\bar{h}\|_p$ is arbitrarily small. We replace h in Eq.(26) with $\lambda\bar{h}$ and use the reverse triangle inequality of l_p norm to get

$$\|DN_p(w)\|_{op} = \frac{|DN_p(w)\lambda\bar{h}|}{\|\lambda\bar{h}\|_p} \leq \frac{|N_p(w+\lambda\bar{h}) - N_p(w)|}{\|\lambda\bar{h}\|_p} \leq 1 \tag{30}$$

Thus, the proof is concluded.

Theorem 3. *Let $w = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^d$ with $w_i > 0$ for $\forall i, 1 \leq i \leq d$ and $w^* = N_\infty(w)$.*

- *If $w^* = w_j$ $(1 \le j \le d)$ and $w^* > w_i$ for $\forall i, i \ne j$, then*

$$\frac{\partial N_\infty(w)}{\partial w_i} = \begin{cases} 1, \text{ if } i = j, \\ 0, \text{ if } i \ne j. \end{cases} . \tag{31}$$

  Namely, $N_p(w)$ is Gateaux differentiable.
- *If there exists $j \ne k$ such that $w^* = w_j = w_k$, then*

$$\frac{\partial N_\infty(w)}{\partial w_i} = 0 \text{ if } w^* \ne w_i, \tag{32}$$

$$\frac{\partial N_\infty(w)}{\partial w_i^+} = 1, \ \frac{\partial N_\infty(w)}{\partial w_i^-} = 0 \text{ if } w^* = w_i. \tag{33}$$

  where $\frac{\partial N_\infty(w)}{\partial w_i^+}$ and $\frac{\partial N_\infty(w)}{\partial w_i^-}$ are the left and right partial derivatives of $N_\infty(w)$ at w_i respectively. In this case, $N_p(w)$ is one-sided Gateaux differentiable.

Proof. The proof of this theorem can be divided into two cases.

• If $w^* = w_j$ $(1 \le j \le d)$ and $w^* > w_i$ for $\forall i, i \ne j$, let $\{e_j\}_{j=1}^d$ be the standard basis of $\mathbb{R}^d$.

In terms of the definition of Gateaux Derivative in Eq.(9), we have

$$\frac{\partial N_\infty(w)}{\partial w_j} = \lim_{t \to 0} \frac{N_\infty(w + te_j) - w^*}{t} \tag{34}$$

Since $N_\infty(w) = w^* = w_j$, $N_\infty(w + te_j) = w^* + t$ for $i = j$ and $N_\infty(w + te_i) = w^*$ for $i \ne j$ when $|t|$ is sufficiently small. This implies Eq.(31).

• If there exists $j \ne k$ such that $w^* = w_j = w_k$, when $w_i \ne w^*$, we have

$$\frac{\partial N_\infty(w)}{\partial w_i} = 0 \text{ if } w^* \ne w_i, \tag{35}$$

by following the above same proof method.

When $w_i = w^* = w_j = w_k$, it is clear that $N_\infty(w + te_i) = w^* + t$ for $t > 0$ and $N_\infty(w + te_i) = w^*$ for $t < 0$. Then the desired results is directly derived.

$$\frac{\partial N_\infty(w)}{\partial w_i^+} = 1, \ \frac{\partial N_\infty(w)}{\partial w_i^-} = 0 \text{ if } w^* = w_i. \tag{36}$$

Theorem 4. *Given a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$ and $|f(w)| \le M$, the Lipschitz constant of the smooth classifier g for $\forall p, 1 \le p \le \infty$ is $M\sigma^{-1}$, i.e.,*

$$|g(w + \delta) - g(w)| \le M\sigma^{-1}\|\delta\|_p \tag{37}$$

Proof. The proof of this theorem can be divided into two cases: $1 \le p < \infty$ and $p = \infty$.

• If $1 \le p < \infty$, then we have

$$\begin{aligned} |g(w + \delta) - g(w)| \le & \int_{\mathbb{R}^d} |f(u)||\psi_\sigma(w + \delta - u) - \psi_\sigma(w - u)|du \\ \le & \|f(w)\|_\infty \int_{\mathbb{R}^d} |\psi_\sigma(w + \delta - u) - \psi_\sigma(w - u)|du \\ \le & \|f(w)\|_\infty \int_{\mathbb{R}^d} |\psi_\sigma(\delta - u) - \psi_\sigma(-u)|du \\ \le & M \int_{\mathbb{R}^d} |\psi_\sigma(\delta - u) - \psi_\sigma(-u)|du \end{aligned} \tag{38}$$

Notice that

$$\psi_\sigma(\delta-u)-\psi_\sigma(-u)=\int_0^1\frac{d}{dt}\psi_\sigma(t\delta-u)dt \tag{39}$$

Based on the Fubini's theorem, we have

$$\begin{aligned}\int_{\mathbb{R}^d}|\psi_\sigma(\delta-u)-\psi_\sigma(-u)|du&=\int_{\mathbb{R}^d}\Big|\int_0^1\frac{d}{dt}\psi_\sigma(t\delta-u)dt\Big|du\\&\le\int_{\mathbb{R}^d}\int_0^1\big|\frac{d}{dt}\psi_\sigma(t\delta-u)\big|dtdu\\&=\int_0^1\int_{\mathbb{R}^d}\big|\frac{d}{dt}\psi_\sigma(t\delta-u)\big|dudt\end{aligned} \tag{40}$$

In terms of Definition 5 and Lemma 1, we have

$$\frac{d}{dt}N_p(t\delta-u)=DN_p(-u)\delta \tag{41}$$

and

$$|DN_p(t\delta-u)\delta|\le\|DN_p(t\delta-u)\|\|\delta\|_p \tag{42}$$

where DN_p is the Fréchet derivative of N_p.

Based on the conclusion of Theorem 2, we get

$$\big|\frac{d}{dt}N_p(t\delta-u)\big|\le\|\delta\|_p \tag{43}$$

Now, we compute

$$\begin{aligned}\int_{\mathbb{R}^d}\big|\frac{d}{dt}\psi_\sigma(t\delta-u)\big|du=&\sigma^{-d-1}K\int_{\mathbb{R}^d}e^{-\|t\delta-u\|_p/\sigma}\big|\frac{d}{dt}N_p(t\delta-u)\big|du\\\le&\sigma^{-d-1}K\|\delta\|_p\int_{\mathbb{R}^d}e^{-\|t\delta-u\|_p/\sigma}du\\=&\sigma^{-d-1}K\|\delta\|_p\int_{\mathbb{R}^d}e^{-\|u\|_p/\sigma}du\end{aligned} \tag{44}$$

According to the definition of Mollifier in Definition 2 and Eq.(6), we get

$$\sigma^{-d}K\int_{\mathbb{R}^d}e^{-\|u\|_\infty/\sigma}du=1 \tag{45}$$

Thus, we have

$$\int_{\mathbb{R}^d}\big|\frac{d}{dt}\psi_\sigma(t\delta-u)\big|du\le\sigma^{-1}\|\delta\|_p \tag{46}$$

In conclusion, we get

$$|g(w+\delta)-g(w)|\le M\sigma^{-1}\|\delta\|_p \tag{47}$$

• *If $p=\infty$, then the above proof method can not be directly applied.*

According to the conclusion of Theorem 3, l_p-norm ($p=\infty$) not always Gateaux differentiable if there exists $j\neq k$ such that $w^*=w_j=w_k$. However, the set contains all such points w_j, w_k, and so on is a measure zero set, on which the integral is equal to zero.

Based on the above observation, we derive the conclusion of this case.

Concretely, let

$$A=\{w=(w_1,\cdots,w_d):\exists i\neq j,w^*=N_\infty(w)=w_i=w_j\} \tag{48}$$

and

$$B = \bigcup_{i,j\in\mathbb{R}^d, i\neq j} \{w = (w_1, \cdots, w_d) : w_i = w_j\} \tag{49}$$

It is clear that

$$A \subset B \tag{50}$$

It is obvious that the set $\{w = (w_1, \cdots, w_d) : i \neq j, w_i = w_j\}$ has dimension $d-1$. Hence it is a measure zero set. Since the union of finitely many measure zero sets is still a measure zero set, B is a measure zero set too. This implies A is also measured zero. Thus, we divide the integral into the one on the measure zero set A and another one on $\mathbb{R}^d - A$.

$$\begin{aligned} |g(w+\delta) - g(w)| \leq & M \int_{\mathbb{R}^d} |\psi_\sigma(\delta - u) - \psi_\sigma(-u)| du \\ = & M \int_{\mathbb{R}^d - A} |\psi_\sigma(\delta - u) - \psi_\sigma(-u)| du + M \int_A |\psi_\sigma(\delta - u) - \psi_\sigma(-u)| du \\ = & I + II = I + 0 = I \end{aligned} \tag{51}$$

Since A is a measure zero set, the integral on A is equal to zero, i.e., $II = 0$.

Again, we have

$$\psi_\sigma(\delta - u) - \psi_\sigma(-u) = \int_0^1 \frac{d}{dt} \psi_\sigma(t\delta - u) dt \tag{52}$$

It is obvious that for any $u \in \mathbb{R}^d - A$, there exists at most finitely many values $t \in [0, 1]$, such that $t\delta - u \in A$, denote them by $0 = t_0 < t_1 < t_2 < \cdots < t_k \leq t_{k+1} = 1$, where k is an integer.

The above integral is improper since the integrand is not defined at $t_1, \cdots, t_k$, and it is integrable in the standard improper integral sense.

$$\begin{aligned} \psi_\sigma(\delta - u) - \psi_\sigma(-u) = & \sum_{i=0}^{k} \psi_\sigma(t_{i+1}\delta - u) - \psi_\sigma(t_i\delta - u) \\ = & \sum_{i=0}^{k} \int_{t_i}^{t_{i+1}} \frac{d}{dt} \psi_\sigma(t\delta - u) dt \end{aligned} \tag{53}$$

where each improper integral in the summation is defined in the standard limiting sense.

Based on the conclusion of Theorem 3, we get

$$\big|\frac{d}{dt} N_\infty(t\delta - u)\big| \leq \|\delta\|_\infty \text{ for any } t\delta - u \in \mathbb{R}^d - A \tag{54}$$

Now, we compute

$$\begin{aligned} \int_{\mathbb{R}^d - A} |\psi_\sigma(\delta - u) - \psi_\sigma(-u)| du = & \int_{\mathbb{R}^d - A} \big| \sum_{i=0}^{k} \int_{t_i}^{t_{i+1}} \frac{d}{dt} \psi_\sigma(t\delta - u) dt \big| du \\ \leq & \sigma^{-d-1} K \int_{\mathbb{R}^d - A} \|\delta\|_\infty \sum_{i=0}^{k} \int_{t_i}^{t_{i+1}} e^{-\|t\delta - u\|_\infty / \sigma} dt du \\ = & \sigma^{-d-1} K \|\delta\|_\infty \int_{\mathbb{R}^d - A} \int_0^1 e^{-\|t\delta - u\|_\infty / \sigma} dt du \\ \leq & \sigma^{-d-1} K \|\delta\|_\infty \int_{\mathbb{R}^d} \int_0^1 e^{-\|t\delta - u\|_\infty / \sigma} dt du \\ = & \sigma^{-d-1} K \|\delta\|_\infty \int_0^1 \int_{\mathbb{R}^d} e^{-\|u\|_\infty / \sigma} du dt \\ = & \sigma^{-1} \|\delta\|_\infty \end{aligned} \tag{55}$$

Therefore, the proof is concluded.

Theorem 5. *Given a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$, $\|f(w)\|_\infty \leq M$, and our smooth classifier g with mollifier smoothing, let $p_A(w)$ and $p_B(w)$ be the probabilities on the most probable class c_A and the runner-up class c_B predicted by f respectively, then $g(w+\delta) = g(w)$ for $\forall \delta$, $\|\delta\|_p \leq r_p$ for any p ($1 \leq p \leq \infty$), where*

$$r_p = \frac{p_A(x) - p_B(x)}{2}\sigma \tag{56}$$

Proof. According to the conclusion of Theorem 4, we have $|g(x+\delta) - g(x)| \leq M\sigma^{-1}\|\delta\|_p$.

Notice that $f(w)$ outputs the probabilities of samples on the trigger set over classes. Therefore, $\|f(x)\|_\infty \leq M \leq 1$ and we get

$$|g(x+\delta) - g(x)| \leq \sigma^{-1}\|\delta\|_p \tag{57}$$

Thus, we have

$$p_A(x+\delta) \geq p_A(x) - \sigma^{-1}\|\delta\|_p \tag{58}$$

and

$$p_B(x+\delta) \leq p_B(x) + \sigma^{-1}\|\delta\|_p \tag{59}$$

where $p_A(x+\delta)$ and $p_B(x+\delta)$ represent the probabilities predicted by $g(x+\delta)$ on c_A and c_B respectively.

If $p_A(x+\delta) \geq p_A(x) - \sigma^{-1}\|\delta\|_p \geq p_B(x) + \sigma^{-1}\|\delta\|_p \geq p_B(x+\delta)$, then $g(x+\delta) = c_A$ for $\forall \delta \in \mathbb{R}^d$, $\|\delta\|_p \leq \frac{p_A(x)-p_B(x)}{2}\sigma$. Therefore,

$$r_p = \frac{p_A(x) - p_B(x)}{2}\sigma \tag{60}$$

Lemma 3. *[Green's second identity] If a and b are both twice continuously differentiable functions on $W \subset \mathbb{R}^d$, then*

$$\int_U (a\Delta b - b\Delta a)dx = \oint_{\partial U} (a\frac{\partial b}{\partial \mathbf{n}} - b\frac{\partial a}{\partial \mathbf{n}})dS \tag{61}$$

where $\Delta a = \frac{\partial^2 a}{\partial x_1^2} + \frac{\partial^2 a}{\partial x_2^2} + \cdots + \frac{\partial^2 a}{\partial x_d^2}$ is the Laplacian operator and $\frac{\partial a}{\partial \mathbf{n}}$ is the directional derivative of a in the direction of the outward pointing normal $\mathbf{n}$ to the surface element dS.

Proof. Please refer to the book (Strauss, 2007) for detailed proof.

Lemma 4. *[Mean Value Property of Harmonic Functions] Let $B(w,\sigma)$ be a ball centering at x with radius σ in $\mathbb{R}^d$. If $q \in C^2(Q)$ and $B(w,\sigma) \subset Q$, then it holds that*

$$q(w) = \frac{1}{\omega_d \sigma^d}\int_{B(w,\sigma)} q(u)du = \frac{1}{d\omega_d\sigma^{d-1}}\oint_{\partial B(w,\sigma)} q(u)dS \tag{62}$$

where ω_d is the volume of the unit ball in $\mathbb{R}^d$ and u is the $(d-1)$-dimensional surface measure.

Proof. Please refer to the book (Axler et al., 2001) for detailed proof.

Theorem 6. *Assuming that a base classifier f with watermark-embedded model parameter $w \in \mathbb{R}^d$ is continuous everywhere, $f(w)$ is C^∞-smooth, and the second derivatives of $f(w)$ are all zero almost everywhere except on a measure zero set Γ, where Γ consists of several planes, then the mollification of $f(w)$*

$$g(w) = \int_{B(w,\sigma)} f(u)\psi_\sigma(w-u)du \tag{63}$$

can be approximated as

$$g(w) \approx \hat{g}(w) = d\omega_d\sigma^{d-1}\nu'(\sigma)f(w) - \omega_d\sigma^{d-1}\alpha\nabla f(w) \tag{64}$$

where α is a hyperparameter vector that is used to balance the accuracy of approximation. $\phi_\sigma(w)$ is the solution of the Poisson equation $\Delta\phi_\sigma(w) = \psi_\sigma(w)$ and $\nu(|w|)$ is the symmetric representation of $\phi_\sigma(w)$, i.e., $\phi_\sigma(w) = \nu(|w|)$.

Proof. Since $\phi_\sigma(w)$ is radially symmetric and $\phi_\sigma(w) = \nu(|w|)$, $\psi_\sigma(w)$ is also radially symmetric and $\psi_\sigma(w) = \mu(|w|)$ and we have

$$\frac{d-1}{r}\nu'(r)+\nu''(r)=\psi_\sigma(|r|),\ \text{for } \nu(r)\to 0 \text{ as } r\to\infty \tag{65}$$

We use set Γ to divide $B(w,\sigma)$ into multiple subsets:

$$B(w,\sigma)=\bigcup_i \Omega_i \tag{66}$$

such that the second derivatives of $f(w)$ are 0 in each Ω_i.

We define another set Γ_w as follows.

$$\Gamma_w=\Gamma\cap B(w,\sigma)=\sum_j \Gamma_w^j \tag{67}$$

We also define the following symbol $\oint_{\Gamma_w^{j,+,-}}\frac{\partial z}{\partial \mathbf{n}}dS$ as the summation of surface integrals (flux) of a function z in two normal directions on Γ_w^j.

$$\oint_{\Gamma_w^{j,+,-}}\frac{\partial z}{\partial \mathbf{n}}dS=\oint_{\Gamma_w^{j,+}}\frac{\partial z}{\partial \mathbf{n}}dS+\oint_{\Gamma_w^{j,-}}\frac{\partial z}{\partial \mathbf{n}}dS \tag{68}$$

Then we let

$$\oint_{\Gamma_w^{+,-}}\frac{\partial z}{\partial \mathbf{n}}dS=\sum_j\oint_{\Gamma_w^{j,+,-}}\frac{\partial z}{\partial \mathbf{n}}dS \tag{69}$$

By using Green's second identity, we compute

$$\begin{aligned}
g(w)=&\int_{B(w,\sigma)} f(u)\psi_\sigma(w-u)du\\
=&\sum_i\int_{\Omega_i} f(u)\Delta\phi_\sigma(w-u)du\\
=&\sum_i\int_{\Omega_i}\Delta f(u)\phi_\sigma(w-u)du+\sum_i\oint_{\partial\Omega_i} f(u)\frac{\partial\phi_\sigma}{\partial\mathbf{n}}(w-u)dS-\sum_i\oint_{\partial\Omega_i}\phi_\sigma(w-u)\frac{\partial f}{\partial\mathbf{n}}(u)dS\\
=&\Big(\oint_{\partial B(w,\sigma)}+\oint_{\Gamma_w^{+,-}}\Big)f(u)\frac{\partial\phi_\sigma}{\partial\mathbf{n}}(w-u)dS-\Big(\oint_{\partial B(w,\sigma)}+\oint_{\Gamma_w^{+,-}}\Big)\phi_\sigma(w-u)\frac{\partial f}{\partial\mathbf{n}}(u)dS\\
=&I+II+III+IV
\end{aligned} \tag{70}$$

Notice that $\sum_i\int_{\Omega_i}\Delta f(u)\phi_\sigma(w-u)du=0$ since the second derivatives of $f(w)$ are all zero almost everywhere, i.e., $\Delta f=0$ in Ω_i.

***Term II**: Since $f(w)$ is continuous everywhere, it is clear that*

$$II=\oint_{\Gamma_w^{+,-}} f(u)\frac{\partial\phi_\sigma}{\partial\mathbf{n}}(w-u)dS=0 \tag{71}$$

***Term I**: Since $\phi_\sigma(w)$ is radially symmetric and $\phi_\sigma(w)=\nu(|w|)$, on the sphere $\partial B(w,\sigma)$, we have*

$$\frac{\partial\phi_\sigma}{\partial\mathbf{n}}(u)=\nabla\phi_\sigma(u)\cdot\frac{u}{|u|} \tag{72}$$

Since $\phi_\sigma(w)=V(|w|)$, we get

$$\nabla\phi_\sigma(u)=\nu'(|u|)\frac{u}{|u|} \tag{73}$$

Therefore, for $u\in\partial B(w,\sigma)$, we have

$$\frac{\partial\phi_\sigma}{\partial\mathbf{n}}(u)=\nu'(\sigma) \tag{74}$$

Hence, I is rewritten as follows.

$$\begin{aligned} I = \oint_{\partial B(w,\sigma)} f(u)\frac{\partial \phi_\sigma}{\partial \mathbf{n}}(w-u)dS =& \nu'(\sigma)\oint_{\partial B(w,\sigma)} f(u)dS \\ =& d\omega_d \sigma^{d-1}\nu'(\sigma)f(w) + \nu'(\sigma)\oint_{\partial B(w,\sigma)} f(u)-f(w)dS \end{aligned} \tag{75}$$

Notice that

$$f(u)-f(w) = \int_0^1 \nabla f(w+\tau(u-w))\cdot(u-w)d\tau \tag{76}$$

By utilizing the Fubini's theorem, we compute

$$\begin{aligned} \oint_{\partial B(w,\sigma)} f(u)-f(w)dS =& \oint_{\partial B(w,\sigma)} \int_0^1 \nabla f(w+\tau(u-w))\cdot(u-w)d\tau dS \\ =& \sigma\int_0^1 \oint_{\partial B(w,\sigma)} \frac{\partial f(w+\tau(u-w))}{\partial \mathbf{n}} dS d\tau \end{aligned} \tag{77}$$

By employing the Divergence theorem, we have

$$\begin{aligned} & \int_0^1 \oint_{\partial B(w,\sigma)} \frac{\partial f(w+\tau(u-w))}{\partial \mathbf{n}} dS d\tau \\ =& \int_0^1 \int_{B(w,\sigma)} \Delta f(w+\tau(u-w))du d\tau - \int_0^1 \int_{\Gamma^{+,-}_{w,\tau}} \frac{\partial f(w+\tau(u-w))}{\partial \mathbf{n}} dS d\tau \\ =& -\int_0^1 \int_{\Gamma^{+,-}_{w,\tau}} \frac{\partial f(w+\tau(u-w))}{\partial \mathbf{n}} dS d\tau \end{aligned} \tag{78}$$

where $\Gamma_{w,\tau} = \Gamma \cap B(w,\tau\sigma)$ *and* $\int_{\Gamma^{+,-}_{w,\tau}}$ *is defined similarly to* $\int_{\Gamma^{+,-}_{w}}$.

By combining the above all formulae together, we have

$$I = d\omega_d\sigma^{d-1}\nu'(\sigma)f(w) - \sigma\nu'(\sigma)\int_0^1 \int_{\Gamma^{+,-}_{w,\tau}} \frac{\partial f(w+\tau(u-w))}{\partial \mathbf{n}} dS d\tau \tag{79}$$

Term III*: By using Divergence Theorem, we have*

$$\begin{aligned} III =& -\oint_{\partial B(w,\sigma)} \phi_\sigma(w-u)\frac{\partial f}{\partial \mathbf{n}}(u)dS \\ =& -\nu(\sigma)\int_{B(w,\sigma)} \Delta f(w-u)du + \nu(\sigma)\int_{\Gamma^{+,-}_w} \frac{\partial f(u)}{\partial \mathbf{n}} dS \\ =& \nu(\sigma)\int_{\Gamma^{+,-}_w} \frac{\partial f(u)}{\partial \mathbf{n}} dS \end{aligned} \tag{80}$$

Term IV*:*

$$IV = -\oint_{\Gamma^{+,-}_w} \phi_\sigma(w-u)\frac{\partial f(u)}{\partial \mathbf{n}} dS \tag{81}$$

Common Term $\int_{\Gamma^{+,-}_w} \frac{\partial f(u)}{\partial \mathbf{n}} dS$ ***in Eqs.(79)-(81)****: Since the second derivatives of* $f(w)$ *are zero in each* Ω_i, $\nabla f(ua)$ *is a constant vector in each* Ω_i. *In addition, as* Γ_i *is a plane, its normal vector is also a constant vector. Therefore,*

$$\oint_{\Gamma^{i,+}_w} \frac{\partial f(u)}{\partial \mathbf{n}} dS = Area(\Gamma^i_w)\nabla f(u^{i,+})\cdot \mathbf{n}^{i,+} \tag{82}$$

where $u^{i,+}$ is any point in on Γ_w^i and $\mathbf{n}^{i,+}$ is the normal vector. Similar identity holds for integrals on $\Gamma_B^{i,-}$.

To get a reasonable approximation, we use $\nabla f(w)$ to approximate all $\nabla f(u^{i,+,-})$. Since Γ_w^i has a scale $\omega_d \sigma^{d-1}$, we have

$$\oint_{\Gamma_w^{+,-}} \frac{\partial f(u)}{\partial \mathbf{n}} dS \approx \omega_d \sigma^{d-1} \nabla f(w) \beta \tag{83}$$

where $\beta \in \mathbb{R}^d$ is a constant.

Since the second term in I in Eq.(79), III in Eq.(80), and IV in Eq.(81) contain this common term $\int_{\Gamma_w^{+,-}} \frac{\partial f(u)}{\partial \mathbf{n}} dS$, we use the same method in Eq.(83) to estimate it. In addition, as all other factors in the second term in I in Eq.(79), III in Eq.(80), and IV are irrelevant to the corresponding integrals and thus can be treated as constants.

Therefore, we obtain the following approximation of $g(w)$.

$$g(w) \approx \hat{g}(w) = d\omega_d \sigma^{d-1} \nu'(\sigma) f(w) - \omega_d \sigma^{d-1} \alpha \nabla f(w) \tag{84}$$

where α is a parameter that aggregates β in Eq.(83) and all other factors in the second term in I in Eq.(79), III in Eq.(80), and IV that are irrelevant to the integrals.

Therefore, the proof is concluded.

A.2. Additional Experiments

Watermark schemes. By following the same options in two representative watermark papers (Zhang et al., 2018; Bansal et al., 2022), we use three watermark schemes to produce the trigger sets: images with embedded content (superimposed text), images with random noise, or images from an unrelated dataset (CIFAR-10 for MNIST and vice versa). While we generated certificates for all three schemes, we focus on embedded content watermark for empirical persistency evaluation.

Certified accuracy against l_2, l_3, and l_∞ attacks. Tables 6-30 exhibit the certified accuracy of three watermark schemes obtained by three certified watermark methods with varying noise level σ and radius r_p on three datasets. Similar trends are observed for the certified accuracy comparison in these tables: our Mollifier Smoothing method achieves the highest certified accuracy values against l_2, l_3, and l_∞ attacks on MNIST, CIFAR-10, and CIFAR-100, which are much better than other baseline methods in most tests. Notice that especially the noise level σ is very high or the norm $p > 2$, such as 1.75 and $p > 3$ or $p > \infty$, our Mollifier Smoothing method can achieve sustainable certified accuracy improvement. It demonstrates that Mollifier Smoothing is relatively robust to both l_2, l_3, and l_∞ watermark removal attacks. This advantage is very important for the usage of deep learning models in the scenarios of intellectual property protection.

Table 6: Certified accuracy (%) of noise watermark on CIFAR-10 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 1.00$	Certified_Watermarks	88	38	5	0	0	0	0	0
	Randomized Smoothing	55	3	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	2	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	92	63	27	5	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	92	60	4	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**45**
$\sigma = 1.50$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 7: Certified accuracy (%) of unrelated watermark on CIFAR-10 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 1.00$	Certified_Watermarks	31	13	2	0	0	0	0	0
	Randomized Smoothing	16	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**98**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	97	77	59	23	11	2	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	99	69	22	1
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**59**
$\sigma = 1.50$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 8: Certified accuracy (%) of embedded content watermark on CIFAR-10 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	80	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**2**	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	2	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	6	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	**100**	0	0	0	0	0	0
	Randomized Smoothing	**100**	96	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**51**	0

Table 9: Certified accuracy (%) of noise watermark on CIFAR-10 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	88	0	0	0	0	0	0	0
	Randomized Smoothing	55	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**4**	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	**100**	0	0	0	0	0	0
	Randomized Smoothing	**100**	**100**	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**90**	0

Table 10: Certified accuracy (%) of unrelated watermark on CIFAR-10 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	31	0	0	0	0	0	0	0
	Randomized Smoothing	16	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	4	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	25	0	0	0	0	0	0
	Randomized Smoothing	**100**	24	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0

Table 11: Certified accuracy (%) of noise watermark on CIFAR-10 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	88	0	0	0	0	0	0	0
	Randomized Smoothing	55	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**2**	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**91**	0

Table 12: Certified accuracy (%) of unrelated watermark on CIFAR-10 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 1.00$	Certified_Watermarks	31	0	0	0	0	0	0	0
	Randomized Smoothing	16	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	0	0	0	0	0
$\sigma = 1.50$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	2	0	0
$\sigma = 1.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**10**	0

Table 13: Certified accuracy (%) of embedded content watermark on MNIST under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.5$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 14: Certified accuracy (%) of noise watermark on MNIST under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.5$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 15: Certified accuracy (%) of unrelated watermark on MNIST under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
$\sigma = 1.5$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 16: Certified accuracy (%) of embedded content watermark on MNIST under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	98	88	0	0	0	0	0
	Randomized Smoothing	**100**	**100**	80	0	0	0	0 0	
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	97	75	0	0	0	0
	Randomized Smoothing	**100**	**100**	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	98	91	75	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	4	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.5$	Certified_Watermarks	**100**	**100**	**100**	97	89	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 17: Certified accuracy (%) of noise watermark on MNIST under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	80	0	0	0	0	0
	Randomized Smoothing	**100**	**100**	91	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	36	0	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	20	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	97	0	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	92	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 18: Certified accuracy (%) of unrelated watermark on MNIST under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	89	0	0	0	0	0
	Randomized Smoothing	**100**	**100**	80	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	98	42	0	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	2	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	**100**	**100**	97	42	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.5$	Certified_Watermarks	**100**	**100**	**100**	98	94	0	0	0
	Randomized Smoothing	**100**	**100**	**100**	**100**	98	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 19: Certified accuracy (%) of embedded content watermark on MNIST under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.5$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 20: Certified accuracy (%) of noise watermark on MNIST under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.5$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 21: Certified accuracy (%) of unrelated watermark on MNIST under l_∞-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0
$\sigma = 1.25$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0
$\sigma = 1.5$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0

Table 22: Certified accuracy (%) of embedded content watermark on CIFAR-100 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	98	93
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	1	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 23: Certified accuracy (%) of noise watermark on CIFAR-100 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	96	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 24: Certified accuracy (%) of unrelated watermark on CIFAR-100 under l_2-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	96	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Randomized Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 25: Certified accuracy (%) of embedded content watermark on CIFAR-100 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**2**	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**88**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0

Table 26: Certified accuracy (%) of noise watermark on CIFAR-100 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**2**	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**6**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0

Table 27: Certified accuracy (%) of unrelated watermark on CIFAR-100 under l_3-norm attacks

Noise Level	Method/Radius	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	0	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	3	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**99**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	**100**	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	0	0	0

Table 28: Certified accuracy (%) of embedded content watermark on CIFAR-100 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**91**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0	0

Table 29: Certified accuracy (%) of noise watermark on CIFAR-100 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**2**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**100**	**100**	0	0	0

Table 30: Certified accuracy (%) of unrelated watermark on CIFAR-100 under l_∞-norm attacks

Noise Level	Method/Radius	0	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$\sigma = 0.25$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	0	0	0	0	0	0	0	0	0
$\sigma = 0.5$	Certified_Watermarks	0	0	0	0	0	0	0	0	0
	Randomized Smoothing	0	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	0	0	0	0	0	0	0
$\sigma = 0.75$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	**100**	**100**	**100**	**100**	**98**	0	0	0	0
$\sigma = 1.0$	Certified_Watermarks	**100**	0	0	0	0	0	0	0	0
	Randomized Smoothing	**100**	0	0	0	0	0	0	0	0
	Mollifier Smoothing	99	**99**	**99**	**99**	**99**	**99**	0	0	0

Table 31: Empirical accuracy (%) on MNIST under three attack methods

Method/Attack	Finetuning	Hard-Label Distillation	Soft-Label Distillation
Certified_Watermarks	85	**100**	**100**
PTYNet	14	23	56
SAC	0	25	97
CosWM	55	66	77
RPV	24	46	99
Watermark Embedding	50	42	96
DeepMarks	99	99	99
NO-stealing-LTH	2	2	3
Mollifier Smoothing	**100**	**100**	**100**

Table 32: Empirical accuracy (%) on MNIST under three attack methods

Method/Attack	Finetuning	Hard-Label Distillation	Soft-Label Distillation
Certified_Watermarkss	95	82	85
PTYNet	5	31	16
SAC	32	65	77
CosWM	**100**	67	33
RPV	96	93	99
Watermark Embedding	18	78	96
DeepMarks	98	99	99
NO-stealing-LTH	2	2	3
Mollifier Smoothing	**100**	**100**	**100**

A.3. Parameter Sensitivity

In this section, we conduct more experiments to validate the sensitivity of various parameters in our Mollifier Smoothing method for the certified robustness task.

Influence of sample numbers of mollifier smoothing. Figure 2 presents the influence of sample numbers of mollifier smoothing in the Mollifier Smoothing model with the number between 50 and 2,000. It is consistent with our earlier analysis that certifying the watermarks with the mollifier smoothing method requires to sample massive noise points, in order to retain high-confidence certificates. However, the cost of sampling and prediction over a large number of points of the input within its neighborhood is non-trivial. We suggest choosing the sample numbers of mollifier smoothing between 500 and 1,500 for well balancing the effectiveness and efficiency.

Sensitivity of training epochs of base classifiers. Figure 3 exhibits the sensitivity of training epochs of base classifiers in our Mollifier Smoothing model by varying them from 10 and 160. We have observed that the performance curves continuously increase with increasing training epochs of base classifiers. This is consistent with the fact that more training epochs makes the image classification models be resilient to watermark removal attacks. Later on, the performance curves keep relatively stable when the training epochs continuously increases, which demonstrates a good convergence of base classifier training.

Influence of learning rates. Figure 4 shows the influence of learning rate in our Mollifier Smoothing model by varying it from 0.001 to 0.1. As we can see, the certified accuracy values keep increasing or stable when we iteratively increase learning rate. The performance curves becomes relatively stable or even drop quickly when the learning rates go beyond a certain threshold, say 0.01. A reasonable explanation is that a too large learning rate may miss the optimal solution with large step size in the search process. Thus, it is important to determine the optimal learning rate for the certified watermark.

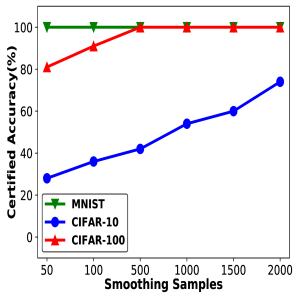


Figure 2: Certified accuracy with varying sample numbers of mollifier smoothing

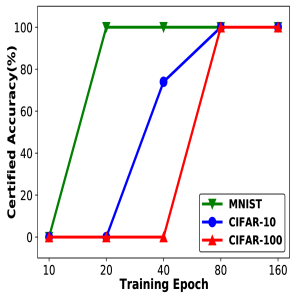


Figure 3: Certified accuracy with varying training epochs of base classifiers

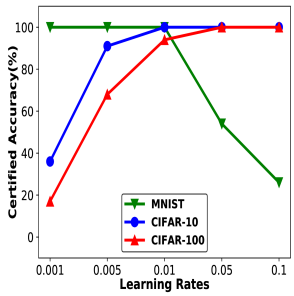


Figure 4: Certified accuracy with varying learning rates of base classifiers

A.4. Experimental Details

Environment. The experiments were conducted on a compute server running on Red Hat Enterprise Linux 7.2 with 2 CPUs of Intel Xeon E5-2650 v4 (at 2.66 GHz) and 8 GPUs of NVIDIA GeForce GTX 2080 Ti (with 11GB of GDDR6 on a 352-bit memory bus and memory bandwidth in the neighborhood of 620GB/s), 256GB of RAM, and 1TB of HDD. Overall, the experiments took about 5 days in a shared resource setting. We expect that a consumer-grade single-GPU machine (e.g., with a 2080 Ti GPU) could complete the full set of experiments in around 9-10 days, if its full resources were dedicated. The codes were implemented in Python 3.7.3 and PyTorch 1.0.14. We also employ Numpy 1.16.4 and Scipy 1.3.0 in the implementation. Since the datasets used are all public datasets and our methodologies and the hyperparameter settings are explicitly described in Section 3, 4, 5, and A.4, our codes and experiments can be easily reproduced on top of a GPU server. We promise to release our open-source codes on GitHub and maintain a project website with detailed documentation for long-term access by other researchers and end-users after the paper is accepted.

Training. We study image classification networks on three standard image datasets: MNIST [1], CIFAR-10 [2], and CIFAR-100 [3]. The above three image datasets are both public datasets, which allow researchers to use for non-commercial research and educational purposes. We train the base classifiers on the training sets of MNIST, CIFAR-10, and CIFAR-100 and test it on the corresponding test sets. We train a convolutional neural network (CNN) on MNIST. We train ResNet-20 over CIFAR-10. We apply the ResNet-18 architecture on CIFAR-100.The neural networks are trained with Kaiming initialization (He et al., 2015) using SGD for 160 epochs with an initial learning rate of 0.1 and batch size 100. The learning rate is decayed by a factor of 0.1 at 1/2 and 3/4 of the total number of epochs. In addition, we run each experiment for 3 trials for obtaining more stable results.

Implementation. For two representative certified defense methods [4] and Certified_Watermarks [5], we used the open-source implementation and default parameter settings by the original authors for our experiments. For six state-of-the-art empirical defense models of PTYNet [6], SAC [7], RPV [8], Watermark Embedding [9], DeepMarks [10], and NO-stealing-LTH [11], we utilized the same model architecture as the official open-source implementation and default parameter settings provided by the original authors for certified robustness in all experiments. To our best knowledge, there is no publicly available open-source implementation for CosWM on the Internet. We tried our best to implement the approach in terms of the algorithm

[1] http://yann.lecun.com/exdb/mnist/
[2] https://www.cs.toronto.edu/~kriz/cifar.html
[3] https://www.cs.toronto.edu/~kriz/cifar.html
[4] https://github.com/locuslab/smoothing
[5] https://github.com/arpitbansal297/certified_watermarks
[6] https://github.com/antigonerandy/ptynet
[7] https://github.com/guanjiyang/sac
[8] https://openreview.net/forum?id=wysXxmukfCA
[9] https://github.com/dnn-security/Watermark-Robustness-Toolbox
[10] https://github.com/dnn-security/Watermark-Robustness-Toolbox
[11] https://github.com/vita-group/no-stealing-lth

description from the original paper. All hyperparameters are standard values from reference codes or prior works. The above open-source codes from the GitHub are licensed under the MIT License, which only requires preservation of copyright and license notices and includes the permissions of commercial use, modification, distribution, and private use.

For our Mollifier Smoothing model, we performed hyperparameter selection by performing a parameter sweep on sample numbers $\in \{50, 100, 500, 1,000, 1,500, 2,000\}$ in the integral calculation, approximation parameter $\alpha \in \{1, 2, 3, 4, 5\}$ in the Mollifier Smoothing-A model, training epochs of the base classifiers $\in \{50, 100, 150, 200, 250\}$, batch size for training the base classifiers $\in \{48, 64, 128, 256, 512\}$, and learning rate $\in \{0.001, 0.005, 0.01, 0.05, 0.1\}$. We select the best parameters over 50 epochs of training and evaluate the model at test time.

Hyperparameter settings.

Unless otherwise explicitly stated, we used the following default parameter settings in the experiments.

Table 33: Hyperparameter Settings

Parameter	Value
Training data ratio on MNIST	60K/10K
Training data ratio on CIFAR-10	50K/10K
Training data ratio on CIFAR-100	50K/10K
Training epochs of the base classifiers	100
Noisy level σ	1.0
Approximation parameter α in the Mollifier Smoothing-A model	2
Sample numbers in the integral calculation	50
Batch size for training the base classifiers	64
Batch size for training the watermarks	256
Learning rate	0.05

A.5. Potential Negative Societal Impacts and Limitations

In this work, the three image datasets are all open-released datasets (Krizhevsky, 2009; Deng et al., 2009), which allow researchers to use for non-commercial research and educational purposes. These three datasets are widely used in training/evaluating the image classification. All baseline codes are open-accessed resources that are from the GitHub and licensed under the MIT License, which only requires preservation of copyright and license notices and includes the permissions of commercial use, modification, distribution, and private use.

To our best knowledge, this work is the first to leverage mollifier theory and partial differential equation theory for conducting the certified watermark problem in high-dimensional space against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) with better applicability. This paper proposes a mollifier smoothing method with dimension-independent Lipschitz constant of our proposed smooth classifier, for conducting the certified watermark problem against the l_p-norm watermark removal attacks ($1 \leq p \leq \infty$) for high parameter dimension d. Our framework can play an important building block for a wide variety of intellectual-property-critical applications that usually require near-zero tolerance of model collapse. This paper is primarily of a theoretical nature. We expect our findings to produce positive impact on intellectual property protection, i.e, significantly improve the ownership verification of real-world deep learning models with better applicability and efficiency. To our best knowledge, we do not envision any immediate negative societal impacts of our results, such as security, privacy, and fairness issues.

An important product of this paper is to develop an effective certified watermark method against large norm adversarial attacks in high-dimensional space. The cost of integral calculation in our original mollifier smoothing method is non-trivial. By leveraging the partial differential equation theory, an approximate model is developed for avoiding time-consuming integral calculation, while maintaining the certified watermark. Our theoretical framework can inspire further improved development and implementations on certified robustness with better applicability and efficiency from the academic institutions and industrial research labs.