

# GSmooth: Certified Robustness against Semantic Transformations via Generalized Randomized Smoothing

Zhongkai Hao<sup>1,2</sup> Chengyang Ying<sup>1</sup> Yinpeng Dong<sup>1,3</sup> Hang Su<sup>1,4,5</sup> Jian Song<sup>2,6</sup> Jun Zhu<sup>1,3,4,5</sup>

## Abstract

Certified defenses such as randomized smoothing have shown promise towards building reliable machine learning systems against  $\ell_p$ -norm bounded attacks. However, existing methods are insufficient or unable to provably defend against semantic transformations, especially those without closed-form expressions (such as defocus blur and pixelate), which are more common in practice and often unrestricted. To fill up this gap, we propose generalized randomized smoothing (GSmooth), a unified theoretical framework for certifying robustness against general semantic transformations via a novel dimension augmentation strategy. Under the GSmooth framework, we present a scalable algorithm that uses a surrogate image-to-image network to approximate the complex transformation. The surrogate model provides a powerful tool for studying the properties of semantic transformations and certifying robustness. Experimental results on several datasets demonstrate the effectiveness of our approach for robustness certification against multiple kinds of semantic transformations and corruptions, which is not achievable by the alternative baselines.

## 1. Introduction

Deep learning models are vulnerable to adversarial examples (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow

<sup>1</sup>Department of Computer Science & Technology, Institute for AI, BNRist Center, Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University <sup>2</sup>Department of Electronic Engineering, Tsinghua University <sup>3</sup>RealAI <sup>4</sup>Peng Cheng Laboratory <sup>5</sup>Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute <sup>6</sup>Key Laboratory of Digital TV System of Guangdong Province and Shenzhen City, Research Institute of Tsinghua University in Shenzhen. Correspondence to: Hang Su <suhangss@mail.tsinghua.edu.cn>, Jian Song <jsong@tsinghua.edu.cn>, Jun Zhu <dczsj@tsinghua.edu.cn>.

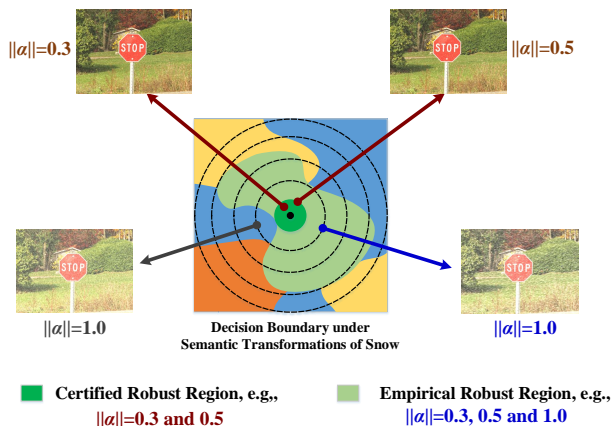


Figure 1. Illustration of certified defense against semantic transformations. We take snow as an example, where  $\alpha$  is the parameter corresponding to the severity. The model is certifiably robust with  $\|\alpha\| = 0.3$  and  $0.5$ . Nevertheless, the model may make an erroneous prediction when  $\|\alpha\| = 1.0$  although it might be empirically robust to other corrupted inputs with  $\|\alpha\| = 1.0$ . This implies that the empirical robustness may not be sufficient for safety-sensitive applications such as autonomous driving.

et al., 2015) as well as semantic transformations (Engstrom et al., 2019; Hendrycks & Dietterich, 2019), which can limit their applications in various security-sensitive tasks. For example, a small adversarial patch on the road markings can mislead the autonomous driving system (Jing et al., 2021), which raises severe safety concerns. Compared with the  $\ell_p$ -norm bounded adversarial examples, semantic transformations can occur more naturally in real-world scenarios and are often unrestricted, including image rotation, translation, blur, weather, etc., most of which are common corruptions (Hendrycks & Dietterich, 2019). Such transformations do not damage the semantic features of images that can still be recognized by humans, but they degrade the performance of deep learning models significantly. Therefore, it is imperative to improve model robustness against these semantic transformations.

Although various methods (Wang et al., 2019; Hendrycks et al., 2020; Xie et al., 2020; Calian et al., 2021) can empirically improve model robustness against semantic transfor-

mations on typical benchmarks (evaluated in average-case), these methods often fail to defend against adaptive attacks by generating adversarial semantic transformations (Hosseini & Poovendran, 2018; Engstrom et al., 2019; Wong et al., 2019), which are optimized over the parameter space of transformations for the worst-case. In contrast, the certified defenses aim to provide a certified region where the model is theoretically robust under any attack or perturbation (Wong & Kolter, 2018; Cohen et al., 2019; Goyal et al., 2018; Zhang et al., 2019). As an example in Fig. 1, we might encounter any levels and types of complex semantic transformations for autonomous driving cars in practice. An ideal safe model should tell users its safe regions that the model is certifiably robust under different transformations.

Several recent studies (Fischer et al., 2020; Mohapatra et al., 2020; Li et al., 2021) have attempted to extend the certified defenses to simple semantic transformations with good mathematical properties like translation and Gaussian blur. However, these methods are neither scalable nor capable of certifying robustness against complex image corruptions and transformations, especially the non-resolvable ones (Li et al., 2021). Typically, many non-resolvable transformations, such as zoom blur and pixelate, do not have closed-form expressions. This makes the theoretical analyses of these transformations difficult and sometimes impossible with the existing methods, although they are common in real-world scenarios. Therefore, it remains highly challenging to certify robustness against these complex and realistic semantic transformations.

In this paper, we propose **generalized randomized smoothing (GSmooth)**, a novel unified theoretical framework for certified robustness against general semantic transformations, including both the resolvable ones (e.g., translation) and non-resolvable ones (e.g., rotational blur). Specifically, we propose to use an image-to-image translation neural network to approximate these transformations. Due to the strong capacity of neural networks, our method is flexible and scalable for modeling these complex non-resolvable semantic transformations. Then, we can theoretically provide the certified radius for the proxy neural networks by introducing new augmented noise in the layers of the surrogate model, which can be used for certifying the original transformations. We further provide a theoretical analysis and an error bound for the approximation, which shows that the impact of the approximation error on the certified bound can be ignored in practice. Finally, we validate the effectiveness of our methods on several publicly available datasets. The results demonstrate that our method is effective for certifying complex semantic transformations. Our GSmooth achieves state-of-the-art performance in both certified accuracy and empirical accuracy for different types of blur or image quality corruptions.

## 2. Related Work

### 2.1. Robustness under Semantic Transformations

Unlike an  $\ell_p$  perturbation adding a small amount of noise to an image, semantic attacks are usually unrestricted. Representative works include the adversarial patches (Brown et al., 2017; Song et al., 2018), the manipulation based on spatial transformations such as rotation or translation (Xiao et al., 2018; Engstrom et al., 2019). Recently, Hendrycks & Dietterich (2019) show that a wide variety of image corruptions and perturbations degrade the performance of many deep learning models. Most of them such as types of blur, pixelate are hard to be analyzed mathematically, such that defending against them is highly challenging. Various data augmentation techniques (Cubuk et al., 2019; Hendrycks et al., 2020; 2021; Robey et al., 2020) have been developed to enhance robustness under semantic perturbations. Also, Calian et al. (2021) propose adversarial data augmentation that can be viewed as adversarial training for defending against semantic perturbations.

### 2.2. Certified Defense against Semantic Transformations

Beyond empirical defenses, several works (Singh et al., 2019; Balunović et al., 2019; Mohapatra et al., 2020) have attempted to certify some simple geometric transformations. However, these are deterministic certification approaches and their performance on realistic datasets is unsatisfactory. Randomized smoothing is a novel certification method that originated from differential privacy (Lecuyer et al., 2019). Cohen et al. (2019) then improve the certified bound and apply it to realistic settings. Yang et al. (2020) exhaustively analyze the bound by using different noise distributions and norms. Hayes (2020) and Yang et al. (2020) point out that randomized smoothing suffers from the curse of dimensionality for the  $l_1$  norm. Fischer et al. (2020) and Li et al. (2021) extend randomized smoothing to certify some simple semantic transformations, e.g., image translation and rotation. They show that randomized smoothing could be used to certify attacks beyond  $l_p$  norm. However, their methods are limited to simple semantic transformations, which are easy to analyze due to their resolvable mathematical properties. Scalable algorithms for certifying most non-resolvable and complex semantic transformations remain unexplored.

## 3. Proposed Method

In this section, we present generalized randomized smoothing (GSmooth) with theoretical analyses.

### 3.1. Problem Formulation

We first introduce the notations and problem formulation. Given the input  $x \in \mathbb{R}^n$  and the label of  $Y = \{1, 2, \dots, K\}$ ,

we denote the classifier as  $f(x) : \mathbb{R}^n \rightarrow [0, 1]^K$ , which outputs predicted probabilities over all  $K$  classes. The prediction of  $f$  is  $\arg \max_{i \in \mathcal{Y}} f(x)_i$ , where  $f(\cdot)_i$  denotes the  $i$ -th element of  $f(\cdot)$ . We denote  $\tau(\theta, x) : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  to be a semantic transformation of the raw input  $x$  with parameter  $\theta \in \mathbb{R}^m$ . We define the smoothed classifier  $G(x)$  as

$$G(x) = \mathbb{E}_{\theta \sim g(\cdot)} [f(\tau(\theta, x))], \quad (1)$$

which is the average prediction for the samples under a smooth distribution  $g(\theta) \propto \exp(-\psi(\theta))$ , here  $\psi : \mathbb{R}^m \rightarrow \mathbb{R}$  is a smooth function. We let  $y_A$  be the predicted label of the smoothed classifier  $G(x)$  for a clean image, and we use  $G(x)_A$  to denote the probability of the top-1 class  $y_A$  in the rest of the paper as follows. Similarly, we can define  $y_B$  as the runner-up class of the smoothed classifier  $G(x)$ .

$$y_A = \arg \max_{i \in \mathcal{Y}} G(x)_i, \quad y_B = \arg \max_{i \in \mathcal{Y} \setminus y_A} G(x)_i, \quad (2)$$

and we use  $G(x)_B$  to denote the probability of class  $y_B$ .

A classifier has a certified robust radius  $R$  if it satisfies that for any perturbation  $k\xi, k \in \mathbb{R}$  where  $k$  is any  $l_p$  norm without specification, we have

$$\arg \max_{i \in \mathcal{Y}} G(\tau(\xi, x))_i = y_A. \quad (3)$$

In this paper, we categorize semantic transformations into two classes: resolvable and non-resolvable. A semantic transformation is resolvable if the composition of two transformations with parameters belonging to a perturbation set  $\theta, \xi \in \mathbb{R}^m$ , is still a transformation with a new parameter  $\gamma = \gamma(\theta, \xi) \in \mathbb{R}^m$ , here  $\gamma(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  is a function depending on these parameters, i.e., satisfying

$$\tau(\theta, \tau(\xi, x)) = \tau(\gamma(\theta, \xi), x). \quad (4)$$

Otherwise, it is non-resolvable, e.g., zoom blur and pixelate. The theoretical properties of resolvable transformations make it much easier to derive the certified bound. Therefore, in next two subsections, we will discuss resolvable and non-resolvable cases in turn under a unified framework.

### 3.2. Certified Bound for Resolvable Transformations

We first discuss a class of transformations that are resolvable. We introduce a function  $\varphi_u(\cdot)$  which will be used in the certified bound. For any vector  $u$  with unit norm, i.e.,  $\|u\| = 1$ , we set  $\gamma_u = \langle hu, \tau(\delta) \rangle$  as a random variable, where  $\delta \sim g(\cdot)$  and  $\tau$  is the gradient operator. We further define the complementary Cumulative Distribution Function (CDF) of  $\gamma_u$  as  $\varphi_u(c) = \mathbb{P}[\gamma_u > c]$  and the inverse complementary CDF of  $\gamma_u$  as  $\varphi_u^{-1}(p) = \inf \{c \mid \mathbb{P}[\gamma_u > c] \leq p\}$ . Following Yang et al. (2020), we define a function as

$$\varphi(p) = \max_{\|u\|=1} \mathbb{E}[\gamma_u \mid \mathbb{P}[\gamma_u > \varphi_u^{-1}(p)]]. \quad (5)$$

For resolvable semantic transformations, we have the following theorem.

**Theorem 1 (Certified bound for Resolvable Transformations, proof in Appendix A.1).** *For any classifier  $f(x)$ , let  $G(x)$  be the smoothed classifier defined in Eq. (1). If there exists a function  $M(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  satisfying*

$$\frac{\partial \gamma(\theta, \xi)}{\partial \xi} = \frac{\partial \gamma(\theta, \xi)}{\partial \theta} M(\theta, \xi),$$

and there exist two constants  $\underline{p}_A, \overline{p}_B$  satisfying

$$G(x)_A > \underline{p}_A > \overline{p}_B > G(x)_B,$$

then  $\arg \max_{i \in \mathcal{Y}} G(\tau(\xi, x))_i = y_A$  holds for any  $k\xi, k \in \mathbb{R}$ , where

$$R = \frac{1}{2M} \int_{\underline{p}_B}^{\underline{p}_A} \frac{1}{\varphi(p)} dp, \quad (6)$$

and  $M = \max_{\xi, \theta \in \mathbb{R}^m} kM(\xi, \theta)k$ .

**Remark.** The settings of Theorem 1 are similar to those in Li et al. (2021) for resolvable semantic transformations. But here we adopt a different presentation and proof for this theorem, which could be easier to extend to our GSmooth framework for general semantic transformations. Specifically, we show two examples of the theorem: additive transformations and commutable transformations. A transformation is additive if  $\tau(\theta, \tau(\xi, x)) = \tau(\xi + \theta, x)$  for any  $\theta, \xi \in \mathbb{R}^m$ . It is commutable if  $\tau(\theta, \tau(\xi, x)) = \tau(\xi, \tau(\theta, x))$  for any  $\theta, \xi \in \mathbb{R}^m$ . For these two types of transformations, it is straightforward to verify that they satisfy the property proposed in Theorem 1 with  $M(\theta, \xi) = I$ . Consequently, we simply apply Theorem 1 for an isotropic Gaussian distribution  $g(\theta) = \mathcal{N}(0, \sigma^2 I)$  and get the certified radius as

$$R = \frac{\sigma}{2} \left( \varphi(\underline{p}_A) - \varphi(\overline{p}_B) \right), \quad (7)$$

where  $\varphi$  is the inverse CDF of the standard Gaussian distribution. These two kinds of transformations include image translation and Gaussian blur, which are basic semantic transformations and the results are consistent with previous work (Li et al., 2021; Fischer et al., 2020). The certification of these simple transformations only requires applying translation or Gaussian blur to the sample and we obtain the average classification score under the noise distribution.

### 3.3. Certified Bound for General Transformations

Translation and Gaussian blur are two specific cases of semantic transformations. In practice, most semantic transformations are not commutable or even not resolvable. Therefore, we need to develop better methods for certifying more types of semantic transformations. However, the existing methods like Semanify-NN (Mohapatra et al., 2020), based on convex relaxation, and TSS (Li et al., 2021), based on randomized smoothing, require developing a specific algorithm

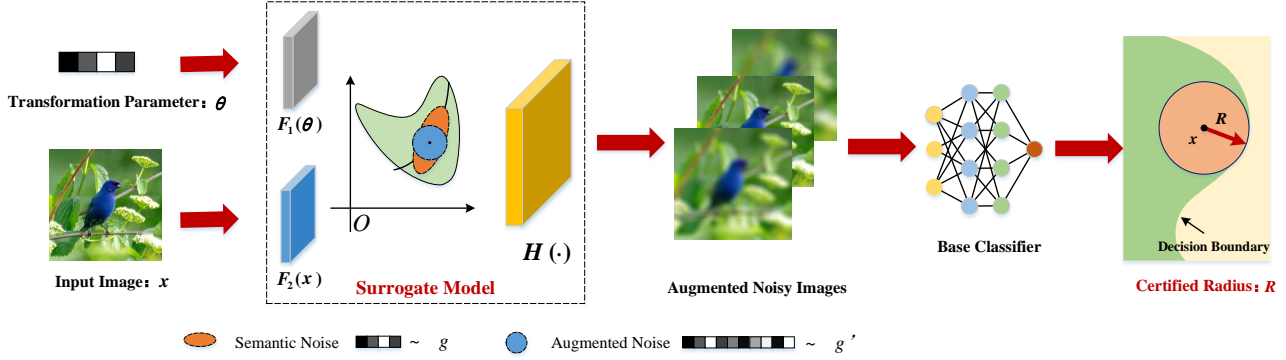


Figure 2. A graphical illustration of our GSmooth. We use a surrogate image-to-image translation network to accurately fit a semantic transformation. Then we add new augmented noise into the surrogate model and construct the GSmooth classifier. The augmented noise is sampled to ensure that the transformation is resolvable in the semantic space. We theoretically calculate the certified bound for the surrogate model to certify the original transformation.

or bound for each individual semantic transformation. This is not scalable and might be infeasible for more complicated transformations without explicit mathematical forms.

To address this challenge, we draw inspiration from the fact that neural networks are able to approximate functions including a complex and unknown semantic transformation (Zhu et al., 2017). Specifically, we propose to use a surrogate image-to-image translation network to accurately fit the semantic transformation. We theoretically prove that, by introducing an augmented noise in the layer of the surrogate model, as shown in Fig. 2, randomized smoothing can be extended to handle these transformations. Specifically, we define the surrogate model as the following form that will lead to a simple certified bound, as we shall see:

$$\tau(\theta, x) = H(F_1(\theta) + F_2(x)), \quad (8)$$

where  $F_1(\cdot)$ ,  $F_2(\cdot)$ , and  $H(\cdot)$  are three individual neural networks.  $F_1(\cdot)$  and  $F_2(\cdot)$  are encoders for transformation parameters and images respectively, and their encodings are added together in the semantic space, which is critical for the theoretical certification. We find that the surrogate neural network is much easier to analyze and can be certified by introducing a dimension augmentation strategy for both noise parameters and input images.

As illustrated in Fig. 2, the augmented noise is added to the semantic layer  $H(\cdot)$  in the surrogate model. Our key insight is that the transformation can be viewed as the superposition of a resolvable part and a non-resolvable residual part in the augmented semantic space. Then, we can use the augmented noise to control the non-resolvable residual part of the augmented dimension  $d > m + n$ . This dimension augmentation is the key step in our method. The augmentation for noise is from  $\mathbb{R}^m$  to  $\mathbb{R}^d$ . To keep the dimension consistent, we also augment data  $x$  to  $\mathbb{R}^d$  by padding 0 entries. By certifying the transformation using the surrogate model, we

are able to certify the original transformation if the approximation error is within an acceptable region (details are in Theorem 3). Our method is flexible and scalable because the surrogate neural network has a uniform mathematical form for theoretical analysis and is trained automatically. Next, we discuss the details of GSmooth.

Specifically, we introduce the augmented data  $\mathbf{x} \in \mathbb{R}^d$  and the augmented parameter  $\theta \in \mathbb{R}^d$  as

$$\mathbf{x} = \begin{bmatrix} \mathbf{0}_{d-n} \\ x \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta \\ \theta^\theta \end{bmatrix}, \quad (9)$$

where the additional parameters  $\theta^\theta \in \mathbb{R}^n$  are sampled from  $g^\theta(\theta^\theta)$ , and the joint distribution of  $\theta^\theta$  and  $\theta$  is  $\theta \sim g$  where  $g(\theta) = g^\theta(\theta^\theta)g(\theta)$ . We define the *generalized smoothed classifier* as

$$\mathcal{G}(\mathbf{x}) = \mathbb{E}_{\theta \sim g(\cdot)} \left[ \mathbb{1}_{f(\tau(\theta, \mathbf{x}))} \right], \quad (10)$$

where  $f$  is the ‘‘augmented target classifier’’ that is equivalent to the original classifier when constrained on the original input  $x$ , which means  $f(\mathbf{x}) = f(x)$ . Note that now all the functions are augmented for a  $d$ -dimensional input. We further augment our surrogate neural network to represent the augmented transformation  $\tau : \mathbb{R}^d \rightarrow \mathbb{R}^d$  as

$$\tau(\theta, \mathbf{x}) = H(F_1(\theta) + F_2(\mathbf{x})), \quad (11)$$

where  $H(\cdot)$ ,  $F_1(\cdot)$ ,  $F_2(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  are parts of the augmented surrogate model. By carefully designing the interaction between the augmented parameters and the original parameters, we could turn the transformation to a resolvable one. It does not change the original surrogate model when constraining to the original input  $x$  and  $\theta$ . Specifically, we

design  $F_1$ ,  $F_2$  and  $H$  as follows:

$$\begin{aligned} F_1(\theta) &= \begin{matrix} \theta \\ F_1(\theta) + \theta^\theta \end{matrix}, F_2(x) = \begin{matrix} \mathbf{0}_{d \times n} \\ F_2(x) \end{matrix}, \\ H(x) &= \begin{matrix} I_{d \times n} \\ H(x) \end{matrix}. \end{aligned} \quad (12)$$

Before stating our main theorem, to simplify the notations, we set

$$\begin{aligned} z_\xi &= F_1(\xi) + F_2(x), \quad z_\theta = F_1(\theta) + F_2(x), \\ y_\xi &= (y_\xi^\theta, y_\xi)^\top = H(F_1(\xi) + F_2(x)), \\ y_\theta &= (y_\theta^\theta, y_\theta)^\top = H(F_1(\theta) + F_2(x)). \end{aligned} \quad (13)$$

Then we provide our main theoretical result below, i.e., the GSmooth classifier is certifiably robust within a given range.

**Theorem 2 (Certified bound for Non-resolvable Transformations, proof in Appendix A.2).** *For any classifier  $f(x)$ , let  $\mathcal{G}(x)$  be the smoothed classifier defined in Eq. (10). If there exist  $\underline{p}_A$  and  $\overline{p}_B$  satisfying*

$$\mathcal{G}(x)_A > \underline{p}_A > \overline{p}_B > \mathcal{G}(x)_B,$$

then  $\arg \max_{i \in \mathcal{Y}} \mathcal{G}(\tau(\xi, x))_i = y_A$  holds for any  $k\xi k_2 \in R$ , where

$$R = \frac{1}{2M} \int_{\underline{p}_B}^{\underline{p}_A} \frac{1}{(p)} dp, \quad (14)$$

and the coefficient  $M$  is defined as

$$M = \max_{\xi, \theta \in \mathcal{P}} \frac{1}{1 + \frac{\partial F_2(y_\xi)}{\partial \xi} \frac{\partial F_1(\theta)}{\partial \theta}}. \quad (15)$$

We have several observations from the theorem. First, it is noted that the certified radius is similar to the result in Theorem 1. Second, compared with resolvable transformations, we need to introduce a new type of noise when constructing the GSmooth classifier. This isotropic noise has the same dimension as the data and is added to the intermediate layers of surrogate neural networks. The theoretical explanation behind this is that this isotropic noise allows the Jacobian matrix of the semantic transformation to be invertible, which is crucial for the proof. Third, we find that the coefficient  $M$  depends on the norm of the difference of two Jacobian matrices and is independent of the target classifier. Later we will discuss the meaning of this term in detail.

Before diving into our theoretical insight and proof of the theorem, we introduce a specific case of Theorem 2 which is convenient for practical usage. We empirically found that taking a linear transformation as  $F_1(\theta) = A_1\theta + b_1$  where  $A_1 \in \mathbb{R}^{d \times m}$ ,  $b_1 \in \mathbb{R}^d$  does not sacrifice the precision of the surrogate network. So, we have  $\frac{\partial F_1(\theta)}{\partial \theta} = A_1$ . After substituting the term in Eq. (15), we only need to

optimize  $\xi$  to calculate  $M$  which can make the bound tighter. Additionally, we use two Gaussian distributions as the noise distribution and the augmented noise distribution, i.e.,  $g(\theta) = \mathcal{N}(0, \sigma_1^2 I)$  and  $g^\theta(\theta^\theta) = \mathcal{N}(0, \sigma_2^2 I)$ . Formally, we have the following corollary.

**Corollary 1.** *Suppose  $f(x)$  is a classifier and  $\mathcal{G}(x)$  is the smoothed classifier defined in Eq. (10). If the layer  $F_1(\theta)$  in the surrogate neural network has the following form:*

$$F_1(\theta) = A_1\theta + b_1, \quad (16)$$

where  $A_1 \in \mathbb{R}^{d \times m}$ ,  $b_1 \in \mathbb{R}^d$  are the parameters; and if there exist  $\underline{p}_A$  and  $\overline{p}_B$  satisfying

$$\mathcal{G}(x)_A > \underline{p}_A > \overline{p}_B > \mathcal{G}(x)_B,$$

then  $\arg \max_{i \in \mathcal{Y}} \mathcal{G}(\tau(\xi, x))_i = y_A$  for any  $k\xi k_2 \in R$  where

$$R = \frac{1}{2M} \Phi^{-1} \left( \frac{\underline{p}_A - \overline{p}_B}{\sigma_1} \right), \quad (17)$$

where  $\Phi^{-1}(\cdot)$  is the inverse CDF of the standard Gaussian distribution, and the coefficient  $M$  is defined as

$$M = \max_{\xi \in \mathcal{P}} \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \frac{\partial F_2(y_\xi)}{\partial \xi} A_1. \quad (18)$$

## 4. Proof Sketch and Theoretical Analysis

In this section, we briefly summarize the main idea for proving Theorem 2 and provide the theoretical insights for our GSmooth further. More details of the proof can be found in Appendix A.2.

### 4.1. Proof Sketch of Theorem 2

The key idea is to prove that the gradient of the smoothed classifier can be bounded by a function of the classification confidence and the parameters of the noise distribution. Formally, we calculate the gradient to the perturbation parameter  $\xi$  for our augmented smoothed classifier as

$$\nabla_{\xi} \mathcal{G}(\tau(\xi, x)) = \nabla_{\xi} \mathbb{E}_{\theta \sim g(\theta)} [f(\tau(\theta, \tau(\xi, x)))]. \quad (19)$$

We further expand the expectation into an integral and find that

$$\nabla_{\xi} \mathcal{G}(\tau(\xi, x)) = \int_{\mathbb{R}^d} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \xi} g(\theta) d\theta. \quad (20)$$

The key step is to eliminate the gradient of  $\frac{\partial f(\tau(\theta, y_\xi))}{\partial \xi}$  and replace it with  $\frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta}$ . Then we integrate it by parts to get the following objective:

$$\nabla_{\xi} \mathcal{G}(\tau(\xi, x)) = \int_{\mathbb{R}^d} F(\tau(\theta, y_\xi)) \frac{\partial M(\xi, \theta) g(\theta)}{\partial \theta} d\theta. \quad (21)$$

After that, we can bound the gradient of the GSmooth classifier by using a technique similar to randomized smoothing (Yang et al., 2020).

## 4.2. Theoretical Insights

Next, we provide some theoretical insights for our augmentation scheme on transformation parameters.

- **Transformation space expansion.** The key purpose of the augmented noise is to construct a closed subspace using additional dimensions. In the augmented space, the Jacobian matrix of the semantic transformation becomes invertible, which is crucial for our proof.
- **Decomposition of the non-resolvable transformations.** As we can see in Eq. (18),  $M$  is influenced by two factors. One is the standard deviation of the two noise distributions. The other is the norm of the Jacobian matrix  $\frac{\partial F_2(y)}{\partial \xi} A_1$ . It can be viewed as the residual of the non-resolvable part of the transformation. Along these lines, our method decomposes the unknown semantic transformation into a resolvable part and a residual part. The non-resolvable residual part can be handled by introducing an additional noise with standard deviation  $\sigma_2$ .

## 4.3. Error Analysis for Surrogate Model Approximation

In this subsection, we theoretically analyze the effectiveness of certifying a real semantic transformation due to the existence of approximation error of surrogate neural networks.

**Theorem 3 (Error Analysis, proof in Appendix A.3).** *Suppose the simulation of the semantic transformation has a small enough error*

$$k_{\bar{\tau}}(\xi, \mathbf{x}) - \bar{\tau}(\xi, \mathbf{x}) k_2 < \varepsilon,$$

where  $\bar{\tau}(\xi, \mathbf{x})$  is the real semantic transformation. Then there exists a constant ratio

$$A = A(kF_1^0(\xi)k_2, kF_2^0(y_\xi)k_2, kF_2^0(z_\xi)k_2) > 0,$$

which does not depend on the target classifier. The certified radius for the real semantic transformation satisfies:

$$R_r > R(1 - A\varepsilon), \quad (22)$$

where  $R$  is the certified radius in Eq. (14) for the surrogate neural network in Theorem 2.

We find that the reduction of the certified radius is influenced by two factors. The first one is the approximation error  $\varepsilon$  between the surrogate transformation and the real semantic transformation. The second one, the ratio  $A$ , is about the norm of the Jacobian matrix for some layers of the surrogate model. This is also an inherent property of the semantic transformation itself and does not depend on the target classifier.

## 5. Experiments

In this section, we conduct extensive experiments to show the effectiveness of our GSmooth on various types of semantic transformations.

### 5.1. Experimental Setup and Evaluation Metrics

We use MNIST (Lecun et al., 1998), CIFAR-10, and CIFAR-100 (Krizhevsky et al., 2009) datasets to verify our methods. We train a ResNet with 110 layers (He et al., 2016a) from scratch. Similar to prior work, we apply moderate data augmentation (Cohen et al., 2019) to improve the generalization of the classifier. For the surrogate image-to-image translation model for simulating semantic transformations, we adopt the U-Net architecture (Ronneberger et al., 2015) for  $H(\cdot)$  and several simple convolutional or linear layers for  $F_1(\cdot)$  and  $F_2(\cdot)$ . All models are trained by using an Adam optimizer (Kingma & Ba, 2015) with an initial learning rate of 0.001 that decays every 50 epochs until convergence. The algorithms for calculating  $M$  and other details in our experiments are listed in Appendix B due to limited space. The evaluation metric is the certified accuracy, which is the percentage of samples that are correctly classified with a larger certified radius than the given range. We use  $k\alpha k$  to indicate the preset certified radius. The results are evaluated on the real semantic transformations after considering the error correction in Theorem 3.

### 5.2. Main Results for Certified Robustness

To demonstrate the effectiveness of our GSmooth on certifying complex semantic transformations, we compare the results of our GSmooth with several baselines, including *randomized smoothing* for some simple semantic transformations of TSS (Li et al., 2021) and **IndivSPT/distSPT** (Fischer et al., 2020). Our GSmooth is a natural and powerful extension of their methods. We also compare our method with *deterministic certification* approaches, including **DeepG** (Balunović et al., 2019), which uses linear relaxations similar to Wong & Kolter (2018), **Interval** (Singh et al., 2019), which is based on interval bound propagation, **VeriVis** (Pei et al., 2017), which enumerates all possible outcomes for semantic transformations with finite values of parameters, and **Semanify-NN** (Mohapatra et al., 2020), which uses a new preprocessing layer to turn the problem into a  $\ell_p$  norm certification.

We measure the certified robust accuracy for different semantic transformations on different datasets in Table 1. We have the following observations based on the experimental results. First, only our method achieves non-zero accuracy in certifying some complex semantic transformations, including rotational blur, defocus blur, zoom blur, and pixelate. The results verify Theorem 2. This is a breakthrough that

Table 1. Our main results of certified robust accuracy on several datasets and multiple types of semantic transformations. – or 0.0% means the method fails to certify this type of semantic transformation.

Transformation	Type	Dataset	Certified Radius	Certified Accuracy (%)						
				GSmooth (Ours)	TSS	DeepG	Interval	VeriVis	Semanify-NN	IndivSPT/distSPT
Gaussian Blur	Additive	MNIST	$\ \alpha\ _2 < 6$	<b>91.0</b>	90.6	–	–	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 4$	<b>67.4</b>	63.6	–	–	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 4$	<b>22.1</b>	21.0	–	–	–	–	–
Translation	Additive	MNIST	$\ \alpha\ _2 < 8$	98.7	<b>99.6</b>	0.0	0.0	98.8	98.8	<b>99.6</b>
		CIFAR-10	$\ \alpha\ _2 < 20$	<b>82.2</b>	80.8	0.0	0.0	65.0	65.0	78.8
		CIFAR-100	$\ \alpha\ _2 < 20$	<b>42.2</b>	41.3	–	–	24.2	24.2	–
Brightness Contrast	Resolvable	MNIST	$\ \alpha\ _\infty < 0.5$	<b>97.7</b>	97.6	60.4	0.0	–	67.4	–
		CIFAR-10	$\ \alpha\ _\infty < 0.4$	<b>82.5</b>	82.4	0.0	0.0	–	–	–
		CIFAR-100	$\ \alpha\ _\infty < 0.4$	<b>42.3</b>	41.4	0.0	0.0	–	–	–
Rotation	Non-resolvable	MNIST	$\ \alpha\ _2 < 50^\circ$	95.7	<b>97.4</b>	685.8	66.0	–	692.48	676
		CIFAR-10	$\ \alpha\ _2 < 10^\circ$	65.6	<b>70.6</b>	62.5	20.2	–	649.37	634
		CIFAR-100	$\ \alpha\ _2 < 10^\circ$	33.2	<b>36.7</b>	0.0	0.0	–	621.7	618
Scaling	Non-resolvable	MNIST	$\ \alpha\ _2 < 0.3$	95.9	<b>97.2</b>	85.0	16.4	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 0.3$	54.3	<b>58.8</b>	0.0	0.0	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 0.3$	31.2	<b>37.8</b>	0.0	0.0	–	–	–
Rotational Blur	Non-resolvable	MNIST	$\ \alpha\ _2 < 10$	<b>95.9</b>	–	–	–	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 10$	<b>39.7</b>	–	–	–	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 10$	<b>27.2</b>	–	–	–	–	–	–
Defocus Blur	Non-resolvable	MNIST	$\ \alpha\ _2 < 5$	<b>89.2</b>	–	–	–	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 5$	<b>25.0</b>	–	–	–	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 5$	<b>13.1</b>	–	–	–	–	–	–
Zoom Blur	Non-resolvable	MNIST	$\ \alpha\ _2 < 0.5$	<b>93.9</b>	–	–	–	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 0.5$	<b>44.6</b>	–	–	–	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 0.5$	<b>14.2</b>	–	–	–	–	–	–
Pixelate	Non-resolvable	MNIST	$\ \alpha\ _2 < 0.5$	<b>87.1</b>	–	–	–	–	–	–
		CIFAR-10	$\ \alpha\ _2 < 0.5$	<b>45.3</b>	–	–	–	–	–	–
		CIFAR-100	$\ \alpha\ _2 < 0.5$	<b>30.2</b>	–	–	–	–	–	–

Table 2. Results of empirical accuracy under adaptive attack on CIFAR-10 (PGD using EoT). For comparison, we also list the results of the corresponding certified accuracy in Table 1.

Type	Certified Acc. (%)		Adaptive Attack Acc. (%)	
	GSmooth	Vanilla	GSmooth	Vanilla
Gaussian blur	67.4	68.1	3.4	3.4
Translation	82.2	87.5	4.2	4.2
Brightness	82.5	85.9	9.6	9.6
Rotation	65.6	68.4	65.4	65.4
Rotational blur	39.7	45.0	33.1	33.1
Defocus blur	25.0	25.5	16.6	16.6
Pixelate	45.3	49.2	38.2	38.2

greatly extends the range of applicability for methods based on randomized smoothing. Second, we see that the performance of GSmooth is similar to state-of-the-art randomized smoothing approaches (e.g., TSS) on several simple semantic transformations such as Gaussian blur and translation. This is a natural result because our method works similarly for resolvable transformations. For two specific non-resolvable transformations, i.e., rotation and scaling, our accuracy is slightly lower. The possible reason is that, in TSS (Li et al., 2021), they derive a more elaborate Lipschitz bound for rotation, which is better than us. Third, those inherently non-resolvable transformations such as image

blurring (except Gaussian blur) and pixelate are more difficult than resolvable or approximately resolvable (rotation) transformations, which makes their certified accuracy lower than the resolvable ones.

### 5.3. Results for Empirical Robustness

To demonstrate that the certified radius is tight enough for real applications, we conduct empirical robustness experiments by testing our method under two types of corruptions. First, we apply project gradient descent (PGD) (Madry et al., 2018) using Expectation over Transformation (EoT) (Athalye et al., 2018) and evaluate the accuracy under adversarial examples on CIFAR-10. Second, we measure the empirical performance of GSmooth under common image corruptions such as CIFAR-10-C (Hendrycks & Dietterich, 2019). We choose AugMix (Hendrycks et al., 2020) and TSS (Li et al., 2021) as baselines. For evaluation, we choose subsets of CIFAR-10-C that are attacked by deterministic corruptions.

The performance under adaptive attack is shown in Table 2. From the results, we find that the empirical accuracy is always higher than the certified accuracy, which indicates the tightness of the certified bound for GSmooth. Additionally, GSmooth also serves as a strong empirical defense

Table 3. Results of ablation study on the influence of standard deviation of smoothing distributions, i.e., transformation noise  $\sigma_1$  and augmented noise  $\sigma_2$  for certification accuracy on rotational blur.

Cert Acc. (%)	CIFAR-10					CIFAR-100				
	$\sigma_1 \backslash \sigma_2$	0.05	0.10	0.15	0.25	$\sigma_1 \backslash \sigma_2$	0.05	0.10	0.15	0.20
Rotational Blur	0.1	44.3	46.4	35.5	16.1	0.1	23.1	26.2	17.2	10.4
	0.25	45.7	47.1	38.3	18.9	0.25	23.2	<b>27.2</b>	20.3	11.3
	0.5	46.5	<b>48.4</b>	38.8	18.3	0.5	22.2	26.1	18.6	11.8
	0.75	42.1	45.5	36.6	17.0	0.75	24.0	25.5	18.3	13.2

Table 4. Empirical accuracy (%) on subsets of CIFAR-10-C. Results with the best performance are bolded.

Type	AugMix	TSS	GSmooth
Gaussian blur	67.4	75.8	<b>76.0</b>
Brightness	<b>82.4</b>	71.8	72.1
Defocus blur	72.2	75.6	<b>76.8</b>
Zoom blur	70.8	75.2	<b>77.1</b>
Motion blur	68.6	70.2	<b>70.5</b>
Pixelate	50.9	76.0	<b>76.7</b>

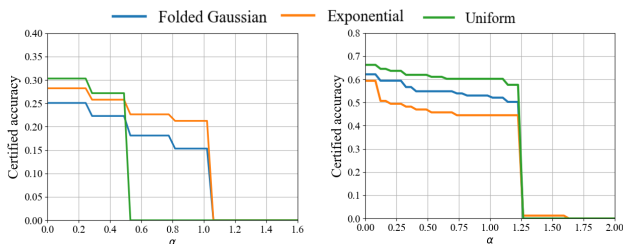


Figure 3. Results of ablation experiments on the influence of smoothing distribution for zoomed blur on the CIFAR-10 dataset. The horizontal axis  $\|\alpha\|_2$  is the certified radius.

compared with a vanilla neural network. The results on CIFAR-10-C are listed in Table 4. We see that our method achieves improvement on most types of corruptions. Since the corruptions in CIFAR-10-C are randomly produced and model-agnostic, the overall performance is better than adaptive attacks. In summary, these empirical results show that the certified bound is tight enough and that GSmooth can also be used as a strong empirical defense method.

#### 5.4. Ablation Studies

**Ablation study on the influence of noise distribution.** The choice of noise distribution is important for randomized smoothing based methods. Since our GSmooth can certify different types of semantic transformations that exhibit different properties, it is necessary to understand the influence of different smoothing distributions for different semantic transformations. We choose (folded) Gaussian, uniform, and exponential distribution and compare the certified accuracy on zoom blur transformation for both CIFAR-10 and CIFAR-100 datasets. As shown in Fig. 3, we found that the impact of smoothing distributions depends on the dataset.

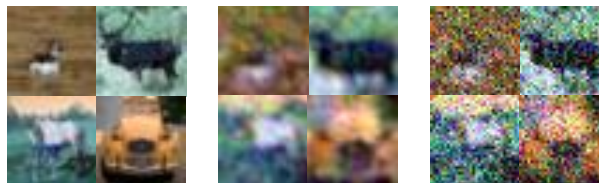


Figure 4. Visualization of difference between the augmented noise in the semantic layers and the noise on raw images. Left: original images from CIFAR-10. Middle: images with augmented noise of  $\sigma_2 = 0.2$ . Right: images with additive Gaussian noise  $\sigma = 0.2$ .

On average, uniform distribution is better for small radius certification, while exponential distribution is more suitable for certifying a large radius.

**Ablation study on the influence of noise variance for certification.** Since our GSmooth contains two different variances for controlling the resolvable part and the residual part for a non-resolvable semantic transformation, here we investigate the effect of different noise variance on the certified accuracy. The results are shown in Table 3. We found that using medium transformation noise and augmented noise achieves the best certified accuracy. This fact is consistent with results in Cohen et al. (2019). An explanation is that there is a trade-off: higher noise variance decreases the coefficient  $M$ , but it might also degrade the clean accuracy.

**Visualization experiments: comparison between augmented semantic noise and image noise.** Our GSmooth needs to add a new noise in the semantic layers of the surrogate model. Here we compare the difference between these two types of noise and visualize them. We randomly sample images from CIFAR-10 and add Gaussian noise with  $\sigma = 0.2$  to both the semantic layer of the surrogate model simulating zoomed blur transformation and raw images. Results are shown in Fig. 4. Both types of noise severely blur the images. But we found that the augmented semantic noise is more placid, which can therefore keep the holistic semantic features better, e.g., as shapes.

## 6. Conclusion

In this paper, we proposed a generalized randomized smoothing framework (GSmooth) for certifying robust-



ness against general semantic transformations. Considering that it is nontrivial to conduct theoretical analysis for non-resolvable transformations, we proposed a novel method using a surrogate neural network to fit semantic transformations. Then we theoretically provide a tight certified robustness bound for the surrogate model, which can be used for certifying semantic transformations. Extensive experiments on publicly available datasets verify the effectiveness of our method by achieving state-of-the-art performance on various types of semantic transformations especially for those complex transformations such as zoom blur or pixelate.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (2020AAA0106000, 2020AAA0104304, 2020AAA0106302, 2017YFA0700904), NSFC Projects (Nos. 62061136001, 61621136008, 62076147, U19B2034, U19A2081, U1811461), the major key project of PCL (No. PCL2021A12), Tsinghua-Alibaba Joint Research Program, Tsinghua-OPPO Joint Research Center, and Tiangong Institute for Intelligent Computing, NVIDIA NVAIL Program and the High Performance Computing Center, Tsinghua University.

## References

- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, pp. 274–283, 2018.
- Mislav Balunović, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. *Advances in Neural Information Processing Systems* 32, 2019.
- Battista Biggio, Iginio Corona, Davide Maiorca, Blaine Nelson, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 387–402, 2013.
- Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.
- Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvester-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021.
- Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.
- Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.
- Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162: 94–114, 2020.
- Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pp. 1802–1811. PMLR, 2019.
- Marc Fischer, Maximilian Baader, and Martin T Vechev. Certified defense to image transformations via randomized smoothing. In *NeurIPS*, 2020.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.
- Jamie Hayes. Extensions and limitations of randomized smoothing for robustness guarantees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 786–787, 2020.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.
- Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.
- Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix:

- A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.
- Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.
- Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1614–1619, 2018.
- Pengfei Jing, Qiyi Tang, Yuefeng Du, Lei Xue, Xiapu Luo, Ting Wang, Sen Nie, and Shi Wu. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.
- Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.
- Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification, 2021.
- Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Towards verifying robustness of neural networks against a family of semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–252, 2020.
- Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*, 2017.
- Alexander Robey, Hamed Hassani, and George J Pappas. Model-based robust deep learning: Generalizing to natural, out-of-distribution data. *arXiv preprint arXiv:2005.10247*, 2020.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.
- Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.
- Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (fWOOTg 18)*, 2018.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32:12635–12644, 2019.
- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295, 2018.
- Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pp. 6808–6817, 2019.
- Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 819–828, 2020.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

## A. Proofs of Theorems

In this section, we will provide detailed proofs of theorems in our paper.

First, we restate the theorem of randomized smoothing for additive noise and binary classifiers  $f(\cdot) : \mathbb{R}^n \rightarrow [0, 1]$ ,

**Theorem 4.** Let  $f(x)$  be any classifier and  $G(x)$  be the smoothed classifier defined as

$$G(x) = \mathbb{E}_{\theta} [f(x + \theta)], \quad (23)$$

if  $G(x) < \frac{1}{2}$ , then  $G(x + \delta) < \frac{1}{2}$  holds for any

$$\|\delta\| < \frac{Z^{-\frac{1}{2}}}{G(x)} \int \frac{1}{p} dp, \quad (24)$$

where  $\psi(\cdot)$  is a function about smoothing distribution defined in Eq. (5).

*Proof.* We first calculate the gradient of the smoothed classifier as

$$\begin{aligned} \nabla G(x) &= \int_{\mathbb{R}^n} \frac{\partial}{\partial x} f(x + \theta) g(\theta) d\theta \\ &= \int_{\mathbb{R}^n} \frac{\partial}{\partial x} f(x + \theta) g(\theta) d\theta \\ &= \int_{\mathbb{R}^n} \frac{\partial}{\partial \theta} f(x + \theta) g(\theta) d\theta. \end{aligned}$$

Then we multiply any vector with unit norm  $u \in B_n(1) = \{u : \|u\| = 1, u \in \mathbb{R}^n\}$ ,

$$\begin{aligned} \langle \nabla G(x), u \rangle &= \int_{\mathbb{R}^n} \frac{\partial}{\partial \theta} f(x + \theta) g(\theta) d\theta, u \\ &= \int_{\mathbb{R}^n} \frac{\partial}{\partial \theta} f(x + \theta), u g(\theta) d\theta \\ &= \int_{\mathbb{R}^n} \frac{\partial f(x + \theta)}{\partial \theta_i} u_i g(\theta) d\theta \\ &= \int_{\mathbb{R}^{n-1}} \int_1^{Z+1} \frac{\partial f(x + \theta)}{\partial \theta_i} u_i g(\theta) d\theta_i \prod_{j \neq i} d\theta_j \\ &= \int_{\mathbb{R}^{n-1}} \int_1^{Z+1} f(x + \theta) u_i \frac{\partial g(\theta)}{\partial \theta_i} d\theta_i \prod_{j \neq i} d\theta_j \\ &= \int_{\mathbb{R}^n} f(x + \theta) \frac{\partial g(\theta)}{\partial \theta_i} u_i d\theta \\ &= \int_{\mathbb{R}^n} f(x + \theta) \langle \nabla g(\theta), u \rangle d\theta \\ &= \int_{\mathbb{R}^n} f(x + \theta) g(\theta) \langle \nabla \psi(\theta), u \rangle d\theta \\ &= \mathbb{E}_{\theta} [f(x + \theta) \langle \nabla \psi(\theta), u \rangle], \end{aligned} \quad (25)$$

here equation 1 holds because of integration by parts. To bound the gradient of the smoothed classifier, we use the following inequality,

$$\langle \nabla G(x), u \rangle \leq \sup_{\mathbb{P}(\hat{G}(x)=G(x))} \mathbb{E}_{\theta} [f(x + \theta) \langle \nabla \psi(\theta), u \rangle]. \quad (26)$$

As shown in Yang et al. (2020), the optimal  $\hat{f}(x)$  achieves at,

$$\hat{f}(x + \theta) = \begin{cases} 1, & \text{if } h(u, r\psi(\theta)) > \varphi_u^{-1}(G(x)) \\ 0, & \text{else.} \end{cases} \quad (27)$$

Then, we have

$$\mathbb{E}_{\theta \sim g} \left[ \hat{f}(x + \theta) h(u, r\psi(\theta)) \right] = \mathbb{E} \left[ \gamma_u | \gamma_u > \varphi_u^{-1}(G(x)) \right] g(G(x)), \quad (28)$$

which means that,

$$h(u, r\psi(\theta)) \geq \varphi_u^{-1}(G(x)). \quad (29)$$

Since it holds for all  $u \in B_n(1)$ , i.e.,  $\|u\| = 1$ , we have,

$$\max_{u \in B_n(1)} h(u, r\psi(\theta)) \geq \varphi_u^{-1}(G(x)). \quad (30)$$

Consider a path from  $\xi_t : [0, k\delta k] \rightarrow \mathbb{R}^d$  with  $\xi_0 = x$ ,  $\xi_{k\delta k} = x + \delta$  and  $\xi_t^0 = \frac{\delta}{k\delta k}$ , we have

$$\frac{dG(\xi_t)}{dt} = h(u, r\psi(\theta)) \geq \varphi_u^{-1}(G(\xi_t)). \quad (31)$$

If the norm of  $\delta$  satisfies that,

$$k\delta k < \int_{G(x)}^{G(x) + \frac{1}{2}} \frac{1}{\varphi(p)} dp. \quad (32)$$

If the right hand side of eq (32) exists, we name it  $k\delta_0 k = \int_{G(x)}^{G(x) + \frac{1}{2}} \frac{1}{\varphi(p)} dp$ . WLOG, we assume that  $G(\xi_t)$  is increasing in  $t$ , then we could calculate the minimal  $t$  when  $G(\xi_t)$  increase to  $\frac{1}{2}$ ,

$$T = \int_{G(\xi_0)}^{G(\xi_0) + \frac{1}{2}} \frac{1}{\varphi(p)} dp = k\delta_0 k. \quad (33)$$

By the generality of  $\delta$  we have  $G(x + \delta) < \frac{1}{2}$  holds for any  $\delta < k\delta_0 k$ .  $\square$

This theorem can be naturally extended to problems with  $p > 2$  classes by considering the two top classes which are  $G(x)_A$  and  $G(x)_B$ . This turns the problem into a binary classification problem.

### A.1. The Proof of Theorem 1

In this part, we will prove Theorem 1.

**Theorem 1.** For any classifier  $f(x)$ , let  $G(x)$  be the smoothed classifier defined in Eq. (1). If there exists a function  $M(\cdot, \cdot) : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$  satisfying

$$\frac{\partial \gamma(\theta, \xi)}{\partial \xi} = \frac{\partial \gamma(\theta, \xi)}{\partial \theta} M(\theta, \xi),$$

and there exist two constants  $\underline{p}_A, \overline{p}_B$  satisfying

$$G(x)_A > \underline{p}_A > \overline{p}_B > G(x)_B,$$

then  $y_A = \arg \max_{i \in \mathcal{Y}} G(\tau(\xi, x))_i$  holds for any  $k\xi k \in R$ , where

$$R = \frac{1}{2M} \int_{\underline{p}_A}^{\overline{p}_B} \frac{1}{\varphi(p)} dp, \quad (34)$$

and  $M = \max_{\xi, \theta \in \mathcal{P}} kM(\xi, \theta)k$ .

*Proof.* WLOG, we only prove it for binary cases that  $f(\cdot) : \mathbb{R}^n \rightarrow [0, 1]$ . Recall that  $G(x) = \mathbb{E}_{\theta} g[f(\tau(\theta, x))]$ , we have

$$\begin{aligned} \nabla_{\xi} G(\tau(\gamma(\theta, \xi), x)) &= \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \tau(\gamma(\theta, \xi), x)} \frac{\partial \tau(\gamma(\theta, \xi), x)}{\partial \xi} g(\theta) d\theta \\ &= \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \tau(\gamma(\theta, \xi), x)} \frac{\partial \tau(\gamma(\theta, \xi), x)}{\partial \gamma(\theta, \xi)} \frac{\partial \gamma(\theta, \xi)}{\partial \xi} g(\theta) d\theta \\ &= \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \tau(\gamma(\theta, \xi), x)} \frac{\partial \tau(\gamma(\theta, \xi), x)}{\partial \gamma(\theta, \xi)} \frac{\partial \gamma(\theta, \xi)}{\partial \theta} M(\theta, \xi) g(\theta) d\theta \\ &= \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \theta} M(\theta, \xi) g(\theta) d\theta. \end{aligned} \quad (35)$$

For any  $u \in \mathbb{R}^d$  satisfying  $\|u\| = 1$ , we have

$$\nabla_{\xi} G(\tau(\gamma(\xi, x)), u) = \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \theta} M(\theta, \xi) u g(\theta) d\theta \quad (36)$$

$$\leq M \max_{\|v\|=1} \int \frac{\partial f(\tau(\gamma(\theta, \xi), x))}{\partial \theta} v g(\theta) d\theta \quad (37)$$

$$\stackrel{1}{=} M \max_{\|v\|=1} \int f(\tau(\gamma(\theta, \xi), x)) \frac{\partial g(\theta)}{\partial \theta} v d\theta, \quad (38)$$

here  $M = \max_{\theta, \xi} \|M(\theta, \xi)\|$  and equality 1 holds because of partial integral. Moreover, we assume that  $g(\theta) = \exp(\psi(\theta))$  and can further prove that

$$\nabla_{\xi} G(\tau(\gamma(\xi, x)), u) \leq M \max_{\|v\|=1} \int f(\tau(\gamma(\theta, \xi), x)) \frac{\partial g(\theta)}{\partial \theta} v d\theta \quad (39)$$

$$= M \max_{\|v\|=1} \int f(\tau(\gamma(\theta, \xi), x)) g(\theta) \nabla \psi(\theta) v d\theta \quad (40)$$

$$= M \max_{\|v\|=1} \int \mathbb{E}_{\theta} g[f(\tau(\gamma(\theta, \xi), x))] \nabla \psi(\theta) v d\theta \quad (41)$$

$$\leq M \max_{\|v\|=1} \sup_{\mathbb{P}_{\hat{G}}(\tau(\xi, x)) = G(\tau(\xi, x))} \mathbb{E}_{\theta} g[\hat{f}(\tau(\gamma(\theta, \xi), x))] \nabla \psi(\theta) v d\theta. \quad (42)$$

Similar with the proof of Theorem 4, the optimal  $\hat{f}$  achieves at

$$\hat{f}(\tau(\gamma(\theta, \xi), x)) = \begin{cases} 1, & \text{if } \nabla \psi(\theta) \cdot u > \varphi_u^{-1}(G(\tau(\xi, x))), \\ 0, & \text{else.} \end{cases} \quad (43)$$

Then we have

$$\nabla_{\xi} G(\tau(\xi, x), u) \leq G(\tau(\xi, x)). \quad (44)$$

Consider a path from  $\zeta_t : [0, k\delta k] \rightarrow \mathbb{R}^d$  with  $\zeta_0 = x$ ,  $\zeta_{k\delta k} = \tau(\xi, x)$  and  $\zeta_t^{\theta} = \frac{\delta}{k\delta k}$ , we have

$$\frac{dG(\zeta_t)}{dt} = \nabla G(\zeta_t) \cdot u \leq G(\zeta_t). \quad (45)$$

The last part of proof is the same with Theorem 4.  $\square$

## A.2. The proof of Theorem 2

In this part, we will prove Theorem 2, which is the main result in this paper.

**Theorem 2.** Suppose  $f(x)$  is any classifier and  $\hat{G}(x)$  is the smoothed classifier defined in Eq. (10), if there exists  $\underline{p}_A, \overline{p}_B$  that satisfies that

$$\hat{G}(x)_A > \underline{p}_A > \overline{p}_B > \hat{G}(x)_B,$$

then  $y_A = \arg \max_{i \in \mathcal{Y}} \hat{G}(\tau(\xi, x))_i$  for any  $\|x\| \leq R$  where

$$R = \frac{1}{2M} \int_{\underline{p}_B}^{\underline{p}_A} \frac{1}{p} dp, \quad (46)$$

and the coefficient  $M$  is defined as

$$M = \max_{\xi, \theta \in \mathcal{Z}^P} \frac{\mathbb{S}}{1 + \frac{\partial F_2(y_\xi)}{\partial \xi} \frac{\partial F_1(\theta)}{\partial \theta}^2} \quad (47)$$

*Proof.* WLOG, we prove it for binary cases where  $f(\cdot) : \mathbb{R}^n \rightarrow [0, 1]$ . First, we will calculate the gradient of  $G(\tau(\xi, x))$  to  $\xi$  as

$$\nabla_{\xi} G(\tau(\xi, x)) = \nabla_{\xi} \mathbb{E}_{\theta \sim g(\theta)} [f(\tau(\theta, \tau(\xi, x)))]. \quad (48)$$

We expand the expectation into integral and use chain rule to see that

$$\nabla_{\xi} G(\tau(\xi, x)) = \int_{\mathbb{R}^{n+d}} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \tau(\theta, y_\xi)} \frac{\partial \tau(\theta, y_\xi)}{\partial y_\xi} \frac{\partial y_\xi}{\partial \xi} g(\theta) d\theta. \quad (49)$$

Similar to previous proof, the key step is to eliminate the gradient of  $\frac{\partial f(\tau(\theta, y_\xi))}{\partial \tau(\theta, y_\xi)}$  and replace it with  $\frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta}$ . Since

$$\frac{\partial \tau(\theta, y_\xi)}{\partial y_\xi} = \begin{bmatrix} \sigma_1^0(z_\theta^0) & F_{21}^0(y_\xi^0) \\ \sigma_2^0(z_\theta) & F_{22}^0(y_\xi) \end{bmatrix}, \quad (50)$$

$$\frac{\partial y_\xi}{\partial \xi} = \begin{bmatrix} H_1^0(z_\xi^0) & I_d \\ H_2^0(z_\xi) & F_1^0(\xi) \end{bmatrix} I_n. \quad (51)$$

We have

$$\begin{aligned} \nabla_{\xi} G(\tau(\xi, x)) &= \int_{\mathbb{R}^{n+d}} \frac{\partial F(\tau(\theta, y_\xi))}{\partial \tau(\theta, y_\xi)} \begin{bmatrix} H_1^0(z_\theta^0) & F_{21}^0(y_\xi^0) \\ H_2^0(z_\theta) & F_{22}^0(y_\xi) \end{bmatrix} \\ &\quad \begin{bmatrix} H_1^0(z_\xi^0) & I_d \\ H_2^0(z_\xi) & F_1^0(\xi) \end{bmatrix} I_n g(\theta) d\theta \end{aligned} \quad (52)$$

$$\begin{aligned} &= \int_{\mathbb{R}^{n+d}} \frac{\partial F(\tau(\theta, y_\xi))}{\partial \tau(\theta, y_\xi)} \begin{bmatrix} H_1^0(z_\theta^0) & I_d & I_d \\ H_2^0(z_\theta) & F_1^0(\theta) & F_1^0(\theta) \end{bmatrix} I_n \\ &\quad \begin{bmatrix} F_{21}^0(y_\xi^0) & H_1^0(z_\xi^0) & I_d \\ F_{22}^0(y_\xi) & H_2^0(z_\xi) & F_1^0(\xi) \end{bmatrix} I_n g(\theta) d\theta \end{aligned} \quad (53)$$

$$\begin{aligned} &= \int_{\mathbb{R}^{n+d}} \frac{\partial F(\tau(\theta, y_\xi))}{\partial \theta} \begin{bmatrix} I_d & F_{21}^0(y_\xi^0) \\ F_1^0(\theta) & I_n \end{bmatrix} F_{22}^0(y_\xi) \\ &\quad \begin{bmatrix} H_1^0(z_\xi^0) & I_d \\ H_2^0(z_\xi) & F_1^0(\xi) \end{bmatrix} I_n g(\theta) d\theta \end{aligned} \quad (54)$$

$$= \int_{\mathbb{R}^{n+d}} \frac{\partial F(\tau(\theta, y_\xi))}{\partial \theta} M(\xi, \theta) g(\theta) d\theta, \quad (55)$$

here we define

$$M(\xi, \theta) = \begin{bmatrix} I_d & F_{21}^0(y_\xi^0) & H_1^0(z_\xi^0) & I_d \\ F_1^0(\theta) & I_n & F_{22}^0(y_\xi) & H_2^0(z_\xi) & F_1^0(\xi) & I_n \end{bmatrix}. \quad (56)$$

We consider the unit enlargement, which means that

$$H_1(z^0) = z^0, F_{21}(x^0) = x^0, \quad (57)$$

thus we have

$$M(\xi, \theta) = \begin{bmatrix} I_d & O_{d \times n} \\ F_{21}^0(y_\xi) & F_1^0(\theta) & F_{22}^0(y_\xi) & H_2^0(z_\xi) \end{bmatrix}. \quad (58)$$

Since  $\theta^0$  is the virtual parameter introduced, which can be taken as 0 in case of actual disturbance. Thus we only need to consider the projection of  $r_\xi \mathcal{G}(y_\xi)$  in the space of  $\xi$ , i.e., we set

$$u = \begin{matrix} u \\ O_{n-1} \end{matrix}, \quad (59)$$

here  $u \in \mathbb{R}^d$  satisfying  $\|u\| = 1$ . Assume

$$\begin{aligned} g(\theta) &= \exp(-\psi(\theta)), \\ \frac{\partial g(\theta)}{\partial \theta} &= -g(\theta) r \psi(\theta). \end{aligned} \quad (60)$$

And we have

$$h r_\xi \mathcal{G}(y_\xi), u_i = \int_{\mathbb{R}^{n+d}} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta} M(\xi, \theta) u g(\theta) d\theta \quad (61)$$

$$= \int_{\mathbb{R}^{n+d}} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta} \begin{matrix} I_d, & O_{d-n} \\ F_{22}^0(y_\xi) & F_1^0(\theta), & O_{n-n} \end{matrix} u g(\theta) d\theta \quad (62)$$

$$= \int_{\mathbb{R}^{n+d}} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta} M(\xi, \theta) u g(\theta) d\theta \quad (63)$$

$$\leq M \max_{\|u\|=1} \int_{\mathbb{R}^{n+d}} \frac{\partial f(\tau(\theta, y_\xi))}{\partial \theta} u g(\theta) d\theta \quad (64)$$

$$\stackrel{1}{=} M \max_{\|u\|=1} \int_{\mathbb{R}^{n+d}} f(\tau(\theta, y_\xi)) \frac{\partial g(\theta)}{\partial \theta} u d\theta \quad (65)$$

$$= M \max_{\|u\|=1} \int_{\mathbb{R}^{n+d}} f(\tau(\theta, y_\xi)) g(\theta) r \psi(\theta) u d\theta \quad (66)$$

$$= M \max_{\|u\|=1} E_{\theta \sim g} \left[ f(\tau(\theta, \tau(\xi, x))) h r \psi(\theta), u_i \right], \quad (67)$$

here equality 1 holds because of partial integral. Moreover, we can bound the right hand side by

$$h r_\xi \mathcal{G}(y_\xi), u_i \leq \sup_{\mathbb{P}: \mathcal{G}(\tau(\xi, x)) = G(\tau(\xi, x))} M \max_{\|u\|=1} E_{\theta \sim g} \left[ \hat{f}(\tau(\theta, \tau(\xi, x))) h r \psi(\theta), u_i \right], \quad (68)$$

and the optimal  $\hat{f}$  is

$$\hat{f}(\tau(\theta, \tau(\xi, x))) = \begin{cases} 1, & \text{if } h r \psi(\theta), u_i > \varphi_u^{-1}(G(\tau(\xi, x))) \\ 0, & \text{else} \end{cases}, \quad (69)$$

here

$$M(\xi, \theta) = \begin{matrix} I_d, & O_{d-n} \\ F_{22}^0(y_\xi) & F_1^0(\theta), & O_{n-n} \end{matrix}, \quad (70)$$

and  $M$  is

$$\begin{aligned} M &= \max_{\xi, \theta \in \mathcal{P}} \begin{matrix} I_d, & O_{d-n} \\ F_{22}^0(y_\xi) & F_1^0(\theta), & O_{n-n} \end{matrix} \\ &= \max_{\xi, \theta \in \mathcal{P}} \frac{\partial F_{22}(y)}{\partial \xi} \frac{\partial F_1(\theta)}{\partial \theta} \\ &= \max_{\xi, \theta \in \mathcal{P}} \frac{1 + \frac{\partial F_{22}(y_\xi)}{\partial \xi} \frac{\partial F_1(\theta)}{\partial \theta}}{2}. \end{aligned} \quad (71)$$



Notice that here  $F_{22}(\cdot)$  is the same as  $F_2(\cdot)$  in the main text. Then we could apply the techniques used in Theorem 1 and Theorem 4 here and have:

$$R = \frac{1}{2M} \int_{\frac{p_A}{p_B}}^Z \frac{1}{(p)} dp. \quad (72)$$

Thus we have proven this theorem.  $\square$

### A.3. The Proof of Theorem 3

In this part, we will prove Theorem 3.

**Theorem 3.** *Suppose the simulation of the semantic transformation has a small enough error*

$$k_{\mathcal{T}}(\xi, \mathbf{x}) - \bar{\tau}(\xi, \mathbf{x}) k_2 < \varepsilon,$$

where  $\bar{\tau}(\xi, \mathbf{x})$  is the real semantic transformation. Then there exists a constant ratio

$$A = A(kF_1^0(\xi)k_2, kF_2^0(y_\xi)k_2, kF_2^0(z_\xi)k_2) > 0,$$

which does not depend on the target classifier. The certified radius for the real semantic transformation satisfies:

$$R_r > R(1 - A\varepsilon), \quad (73)$$

where  $R$  is the certified radius in Eq. (14) for the surrogate neural network in Theorem 2.

*Proof.* We set

$$\mathbf{u} = \begin{matrix} u \\ O_{n-1} \end{matrix}, \quad (74)$$

here  $u \in \mathbb{R}^d$  satisfying  $\|u\| = 1$ . Then we have

$$\begin{aligned} \int_{\xi} \mathcal{G}(\mathcal{T}(\xi, \mathbf{x})) - \int_{\xi} \mathcal{G}(\tau(\xi, \mathbf{x})), \mathbf{u} &= \int \frac{\partial f(\mathcal{T}(\theta, \mathcal{T}(\xi, \mathbf{x})))}{\partial \xi} - \frac{\partial f(\mathcal{T}(\theta, \tau(\xi, \mathbf{x})))}{\partial \xi} \mathbf{u} g(\theta) d\theta \\ &= \int_{\xi} \mathcal{T}(\xi, \mathbf{x}) - \tau(\xi, \mathbf{x}) \frac{\partial^2 f(\mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x})))}{\partial \xi \partial \hat{\tau}} \mathbf{u} g(\theta) d\theta. \end{aligned} \quad (75)$$

Set  $L = \frac{\partial^2 f(\mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x})))}{\partial \xi \partial \hat{\tau}}$ , and we have

$$L = \frac{\partial}{\partial \xi} \frac{\partial f(\mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x})))}{\partial \hat{\tau}} = \frac{\partial}{\partial \xi} \frac{\partial f(\mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x})))}{\partial \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))} \frac{\partial \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))}{\partial \hat{\tau}}. \quad (76)$$

Set  $y_\xi = \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))$ , and we have:

$$\begin{aligned} \frac{\partial \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))}{\partial \hat{\tau}} &= \frac{\partial H(F_1(\theta) + F_2(\hat{\tau}(\xi, \mathbf{x})))}{\partial \hat{\tau}} \\ &= H'(F_1(\theta) + F_2(\hat{\tau}(\xi, \mathbf{x}))) \frac{\partial F_2(\hat{\tau}(\xi, \mathbf{x}))}{\partial \hat{\tau}} \\ &= H'(F_1(\theta) + F_2(\hat{\tau}(\xi, \mathbf{x}))) \begin{matrix} I_d & I_d \\ F_1^0(\theta) & F_1^0(\theta) \end{matrix} \begin{matrix} I_n & I_n \\ F_1^0(\theta) & F_1^0(\theta) \end{matrix} \frac{\partial F_2(\hat{\tau}(\xi, \mathbf{x}))}{\partial \hat{\tau}} \\ &= \frac{\partial \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))}{\partial \theta} \begin{matrix} I_d & I_d \\ F_1^0(\theta) & F_1^0(\theta) \end{matrix} F_2^0(\hat{\tau}(\xi, \mathbf{x})) \\ &= \frac{\partial \mathcal{T}(\theta, \hat{\tau}(\xi, \mathbf{x}))}{\partial \theta} A_1, \end{aligned} \quad (77)$$

here  $A_1 = \frac{I_d}{F_1^0(\theta)} \cdot I_n \cdot F_2^0(\hat{\tau}(\xi, x))$ . By the proof above, we have  $\frac{\partial}{\partial \xi} = \frac{\partial}{\partial \theta} A_2$ , thus we have

$$\begin{aligned} L &= \frac{\partial}{\partial \xi} \frac{\partial f(\tau(\theta, \hat{\tau}(\xi, x)))}{\partial \tau(\theta, \hat{\tau}(\xi, x))} \frac{\partial \tau(\theta, \hat{\tau}(\xi, x))}{\partial \hat{\tau}} \\ &= \frac{\partial}{\partial \theta} \frac{\partial f(\tau(\theta, \hat{\tau}(\xi, x)))}{\partial \tau(\theta, \hat{\tau}(\xi, x))} \frac{\partial \tau(\theta, \hat{\tau}(\xi, x))}{\partial \theta} A_1 \cdot A_2 \\ &= \frac{\partial}{\partial \theta} \frac{\partial f(\tau(\theta, \hat{\tau}(\xi, x)))}{\partial \theta} A_1 \cdot A_2. \end{aligned} \quad (78)$$

Furthermore, we can prove that

$$\begin{aligned} & \int_{\xi} \int_{\tau(\xi, x)} \frac{\partial^2 f(\tau(\theta, \hat{\tau}(\xi, x)))}{\partial \xi \partial \hat{\tau}} u g(\theta) d\theta \\ &= \int_{\xi} \int_{\tau(\xi, x)} \frac{\partial}{\partial \theta} \frac{\partial f(\tau(\theta, \hat{\tau}(\xi, x)))}{\partial \theta} A_1 \cdot A_2 u g(\theta) d\theta \\ & \leq A \int_{\xi} \int_{\tau(\xi, x)} f(\tau(\theta, \hat{\tau}(\xi, x))) g(\theta) h r \psi(\theta), u d\theta, \end{aligned} \quad (79)$$

where  $A$  is a constant depending on  $kF_1^0(\xi)k$ ,  $kF_2^0(y_\xi)k$  and  $kF_2^0(z_\xi)k$ . Then there exists  $A$  and we have

$$R_r > R(1 - \epsilon A), \quad (80)$$

where

$$R = \frac{1}{2M} \int_{p_A}^1 \frac{1}{p} dp.$$

Thus we have proven this theorem.  $\square$

## B. Implementation Details and Experimental Settings

### B.1. Practical Algorithms for Calculating $M$

For resolvable transformations in Theorem 1, the  $M$  is defined as

$$M = \max_{\xi, \theta \in \mathcal{P}} kM(\xi, \theta)k. \quad (81)$$

Since if we have verified that the semantic transformation is resolvable, most of time we have a closed form of  $M$  like contrast/brightness transformation and we are able to calculate it analytically as shown in Li et al. (2021).

For non-resolvable transformations in Corollary 1,  $M$  is defined as

$$M = \max_{\xi \in \mathcal{P}} \frac{1}{\sigma_1^2 + \sigma_2^2} \frac{\partial F_2(y_\xi)}{\partial \xi} A_1 \cdot A_2. \quad (82)$$

This ratio is similar with the Lipschitz bound for a semantic transformation in Li et al. (2021). For low dimensional semantic transformations, we are able to interpolate the domain to find a maximum  $M$  and corresponding  $\xi$ . But this remains a challenge for high dimensional semantic transformations. Specifically, for a given  $\xi$ , we need to compute the norm of  $\frac{\partial F_2(y_\xi)}{\partial \xi} A_1$ . The Jacobian matrix is  $n \times n$ . Calculating it requires  $n$  times of backpropagation. Thus it is inefficient to store the matrix or directly compute its norm. To solve the problem, we notice that

$$\frac{\partial F_2(y_\xi)}{\partial \xi} A_1 \cdot A_2 \cdot u = \max_{\|u\|_2=1} \frac{\partial F_2(y_\xi)}{\partial \xi} A_1 \cdot u. \quad (83)$$

And then we have

$$\frac{\partial F_2(y_\xi)}{\partial \xi} A_1 u = \frac{\partial}{\partial \xi} hF_2(y_\xi) A_1, u. \tag{84}$$

Since tranposing a matrix does not change its norm, we could calculate its norm by optimizing  $u$  that,

$$\max_{\|u\|_2=1} \frac{\partial}{\partial \xi} hF_2(y_\xi) A_1, u \tag{85}$$

Using this formulation, we only need to multiply the output with an unit vector and perform one backpropagation. This is a simple convex optimization problem. Then we could use any iterative algorithm to find its solution which is very fast to compute. This trick is crucial and it makes the matrix norm computation to be scalable in practice.

### B.2. Other Experimental Details

Our GSmooth requires to train two neural networks. First we randomly generate corrupted images to train a image-to-image neural network. The training process of classifiers and certification for semantic transformations are done on 2080Ti GPUs. We use a U-Net (Ronneberger et al., 2015) for the surrogate model and replace all BatchNorm layers with GroupNorm (Wu & He, 2018) since we might use the model in low batchsize settings. The U-Net could be replace by other networks used in image segmentation or superresolution like Res-UNet (Diakogiannis et al., 2020) or EDSR (Lim et al., 2017). We use L1-loss to train the surrogate model which achieves better accuracy than others which is also reported (Lim et al., 2017).

After training a surrogate model to simulate the semantic transformation, we then train the base classifier for certification with a moderate data augmentation (Li et al., 2021; Cohen et al., 2019) to ensure that training and testing of the classifier is performed on the same distribution. There are two types of data augmentation, one is the semantic transformation and the other is the augmented noise introduced only in our work. Data augmentation based semantic transformation could be done using both the surrogate model or the raw semantic transformation. We can only use the surrogate model to add the augmented noise because this noise is a type of semantic noise in the layers of surrogate model. In our experiments the standard deviation of the augmented noise is chosen from 0.1 - 0.4 depending on the performance. The basic network architectures for these datasets are kept the same with Li et al. (2021). On CIFAR-100 datasets, we use a PreResNet (He et al., 2016b) re-implement the method by Li et al. (2021). We also adopt the progressive sampling trick mentioned in TSS (Li et al., 2021) which is useful to reduce computational cost and certify larger radius. The details could also be found in Li et al. (2021).