Enhancing Certified Robustness of Smoothed Classifiers via Weighted Model Ensembling

Chizhou Liu  
Center for Data Science  
Peking University  
liuchizhou@pku.edu.cn

Yunzhen Feng  
School of Mathematical Sciences  
Peking University  
fengyz@pku.edu.cn

Ranran Wang  
National School of Development  
Peking University  
ranranw@bibdr.org

Bin Dong  
Beijing International Center for Mathematical Research, Institute for Artificial Intelligence, and Center for Data Science  
Peking University  
donbin@math.pku.edu.cn

Abstract

Randomized smoothing has achieved state-of-the-art certified robustness against $l_2$-norm adversarial attacks. However, it also leads to accuracy drop compared to the normally trained models. In this work, we employ a Smoothed WEighted ENsembling (SWEEN) scheme to improve the performance of randomized smoothed classifiers. We characterize the optimal certified robustness attainable by SWEEN models. We show the accessibility of SWEEN models attaining the lowest risk w.r.t. a surrogate loss function. We also develop an adaptive prediction algorithm to reduce the prediction and certification cost of SWEEN models. Extensive experiments show that SWEEN models outperform the upper envelope of their corresponding base models by a large margin. Moreover, SWEEN models constructed using a few small models are able to achieve comparable performance to a single large model with notably reduced training time.

1 Introduction

Deep neural networks have shown great success in image classification tasks. However, they are vulnerable to adversarial examples, which are small imperceptible perturbations of their inputs that cause misclassification [3, 44]. To settle this issue, various defense methods have been proposed for training classifiers that are robust to adversarial perturbations. These defenses can be broadly categorized into empirical defenses and certified defenses. One of the most successful empirical defenses is adversarial training [24, 31], which optimizes the model by minimizing the loss over adversarial examples generated during training. Empirical defenses produce models robust to certain adversaries without theoretical guarantee. In fact, most of the empirical defenses are heuristic and subsequently broken by more sophisticated adversaries [1, 6, 47, 48]. Certified defenses, either exact or conservative, mitigate the deficiency in empirical defenses. In the context of $l_p$ norm-bounded perturbations, exact methods report whether an adversarial example exists within an $l_p$ ball with radius $r$ centered at a given input $x$. Exact methods are usually based on Satisfiability Modulo Theories [12, 20] or mixed integer linear programming [13, 30], which are computationally inefficient and not scalable [46]. Conservative methods are more computationally efficient, but might mistakenly flag a safe data point as vulnerable to adversarial examples [8, 10, 11, 14, 15, 33, 35, 36, 39, 42, 50].
However, both types of defenses are not scalable to practical networks that can perform well on modern machine learning problems like the ImageNet [9] classification task.

Recently, a new certified defense technique called randomized smoothing [7, 26] has been proposed. A (randomized) smoothed classifier is constructed from a base classifier, typically a deep neural network. It outputs the most probable class given by its base classifier under random noise perturbation of the input. Randomized smoothing is scalable due to its independency over architectures and has achieved state-of-the-art certified $l_2$-robustness. In theory, randomized smoothing can apply to any classifiers. However, naively applying randomized smoothing on standard-trained classifiers leads to poor robustness results. It is still not completely solved how to train a base classifier so that the corresponding smoothed classifier has good robustness properties. Recently, Salman et al. [38] employ adversarial training to train base classifiers and substantially improve the performance of randomized smoothing. This indicates that techniques originally proposed for empirical defenses can be useful in finding good base classifiers for randomized smoothing.

In this paper, we consider applying another empirical defense method to randomized smoothing, namely model ensembling. The idea of model ensembling has been used in various defenses against adversarial examples, and show promising results [29, 32, 34, 40, 43, 49]. Specifically, we employ a Smoothed WEighted ENsembling (SWEEN) scheme. We show that SWEEN substantially improves both the accuracy and robustness of smoothed classifiers. SWEEN does not limit how individual base classifiers are trained, and thus is compatible with most previously proposed training algorithms on randomized smoothing.

Our contributions are summarized as follows:

1. We employ a weighted ensembling scheme to substantially improve both the accuracy and robustness of smoothed classifiers. We theoretically demonstrate the best certified robustness attainable by SWEEN models. Furthermore, we prove that under certain conditions, we can obtain a SWEEN model which has near-optimal risk attainable by a broad set of functions.

2. We develop an adaptive prediction algorithm for the weighted ensembling, which effectively reduces the prediction and certification cost of smoothed ensemble classifiers.

3. We evaluate our proposed method through extensive experiments. On all tasks, weighted ensemble models consistently outperform the upper envelopes of their respective base models in terms of the approximated certified accuracy by a large margin.

2 Related Work

In the past few years, numerous defenses have been proposed to build classifiers robust to adversarial examples. Our work typically involves randomized smoothing and model ensembling.

**Randomized smoothing** Randomized smoothing constructs a smoothed classifier from a base classifier via convolution between the input distribution and certain noise distribution. It is first proposed as a heuristic defense by [5, 29]. Lecuyer et al. [26] first prove robustness guarantees for randomized smoothing utilizing tools from differential privacy. Subsequently, a stronger robustness guarantee is given by Li et al. [28]. Cohen et al. [7] provide a tight robustness bound for isotropic Gaussian noise in $l_2$-robustness setting. The theoretical properties of randomized smoothing in various norm and noise distribution settings have been further discussed in the literature [4, 23, 27, 45, 55]. Recently, a series of works [38, 56] further develop practical algorithms to train a base classifier for randomized smoothing. From another perspective, our work improves the performance of smoothed classifiers via weighted ensembling of pretrained base classifiers.

**Model ensembling** Model ensembling as a technique aiming to improve the generalization performance has been widely studied and applied in machine learning [16, 22]. Krogh and Vedelsby [22] showed that accurate and diverse networks produce better ensemble classifiers. Recently, simple averaging of multiple neural networks has been a success in ILSVRC competitions [17, 21, 31]. Model ensembling has also been used in defenses against adversarial examples [29, 32, 34, 40, 43, 49]. Wang et al. [49] have shown that jointly trained ensemble of noise injected ResNets can improve clean and robust accuracies. Recently, Meng et al. [32] find that an ensemble of many diverse weak models can be strong against adversarial attacks. Unlike the above works, which are empirical or heuristic, we employ ensembling in randomized smoothing to provide a theoretical robustness certification.
3 Preliminaries

Notation Let $\mathcal{Y} = \{e_1, e_2, \ldots, e_M\}$, where $e_k$ is the $M$-dimensional one-hot vector whose $k$-th entry is 1 for $k = 1, \ldots, M$. Sometimes we will use $e_k$ to refer to $e_k$ when there is no ambiguity. Let $\Delta_k = \{(p_1, p_2, \ldots, p_k) \mid p_i \geq 0, \sum_{i=1}^k p_i = 1\}$ be the $k$-dimensional probability simplex for $k \in \mathbb{N}_+$, and $\Delta = \Delta_M$. For an $M$-dimensional function $f$, we use $f_i$ to refer to its $i$-th entry, $i = 1, 2, \ldots, M$. We use $\mathcal{N}(0, \sigma^2 I)$ to denote the $d$-dimensional Gaussian distribution with mean 0 and variance $\sigma^2 I$. We use $\Phi^{-1}$ to denote the inverse of the standard Gaussian CDF, and use $\Gamma$ to denote the gamma function. We use $\mathbb{R}^n_+$ to denote the set of non-negative real numbers. For $x, a, b \in \mathbb{R}$, $a \leq b$, we define $\text{clip}(x; a, b) = \min\{\max\{x, a\}, b\}$. We use $\Omega(\cdot)$ to denote Big-Omega notation that suppresses multiplicative constants.

Neural network and classifier Consider a classification problem from $\mathcal{X} \subseteq \mathbb{R}^d$ to classes $\mathcal{Y}$. Assume the input space $\mathcal{X}$ has finite diameter $D = \sup_{x_1, x_2 \in \mathcal{X}} \|x_1 - x_2\|_2 < \infty$. The training set $\{(x_i, y_i)\}_{i=1}^p$ is i.i.d. drawn from data distribution $\mathcal{D}$. We call $f$ a probability function or a classifier if it is a mapping from $\mathbb{R}^d$ to $\Delta$ or $\mathcal{Y}$, respectively. For a probability function $f$, its induced classifier $f^*$ is defined such that $f^*(x) = e_k$ where $k = \arg\max_{1 \leq i \leq M} f_i(x)$. For simplicity, we will not distinguish between $f$ and $f^*$ when there is no ambiguity, and hence all definitions and properties for classifiers automatically apply to probability functions as well. $f(\cdot; \theta)$ denotes a neural network parameterized by $\theta \in \Theta$. Here $\Theta$ can include hyper-parameters, thus including diverse network architectures.

Certified robustness We call $x + \delta$ an adversarial example of a classifier $F$ if $F$ correctly classifies $x$ but $F(x + \delta) \neq F(x)$. Usually $\|\delta\|_2$ is small enough so $x + \delta$ and $x$ appear almost identical for the human eye. The $(l_2)$-robust radius of $F$ is defined as

$$r(x, y; F) = \inf_{F(x + \delta) \neq y} \|\delta\|_2, \quad (1)$$

which is the radius of the largest $l_2$ ball centered at $x$ within which $F$ consistently predicts the true label $y$ of $x$. Note that $r(x, y; F) = 0$ if $F(x) \neq y$. As mentioned above, we can extend the above definitions to the case when $F$ is a probability function by considering the induced classifier $F^*$. In most cases $r(x, y; F)$ is either too computationally expensive to solve exactly, or not tractable at all. A certified robustness method typically tries to find some lower bound $r_c(x, y; F)$ of $r(x, y; F)$, and we call $r_c$ a certified radius of $F$.

Randomized smoothing Let $f$ be a probability function or a classifier. The (randomized) smoothed function of $f$ is defined as

$$g(x) = \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I)} [f(x + \delta)]. \quad (2)$$

The (randomized) smoothed classifier of $f$ is then defined as $g^*$. Cohen et al. [7] first provide a tight robustness guarantee for classifier-based smoothed classifiers, which is summarized in the following theorem:

Theorem 1. (Cohen et al. [7]) For any classifier $f$, denote its smoothed function by $g$. Then

$$r(x, y; g) \geq \frac{\sigma}{2} [\Phi^{-1}(g_y(x)) - \Phi^{-1}(\max_{k \neq y} g_k(x))]. \quad (3)$$

Later on, Salman et al. [38], Zhai et al. [56] prove that Theorem 1 holds for probability functions as well.

4 SWEEN: Smoothed weighted ensembling

In this section, we describe the SWEEN framework we use. We also present some theoretical results for SWEEN models. The proofs of the results in this section can be found in Appendix A.

4.1 SWEEN: Overview

To be specific, we adopt a data-dependent weighted average of neural networks as our ensemble model. Suppose we have some pre-trained neural networks $f(\cdot; \theta_1), \ldots, f(\cdot; \theta_K)$ as ensemble candidates. A
weighted ensemble model is then
\[ f_{\text{ens}}(\cdot; \theta, w) = \sum_{k=1}^{K} w_k f(\cdot; \theta_k), \] (4)
where \( \theta = (\theta_1, \ldots, \theta_K) \in \Theta^K \), and \( w \in \Delta_K \) is the ensemble weight. For a specific \( f_{\text{ens}} \), the corresponding SWEEN model is defined as the smoothed function of \( f_{\text{ens}} \), denoted by \( g_{\text{ens}} \). We have
\[ g_{\text{ens}}(x; \theta, w) = \mathbb{E}[\sum_{k=1}^{K} w_k f(x + \delta; \theta_k)] = \sum_{k=1}^{K} w_k g(x; \theta_k), \] (5)
where \( g(\cdot; \theta) \) is the smoothed function of \( f(\cdot; \theta) \). This result means \( g_{\text{ens}} \) is the weighted sum of the smoothed functions of the ensemble candidates under the same weight \( w \). We can minimize a surrogate loss of \( g_{\text{ens}} \) over the training set to obtain the value of some appropriate weight. This data-dependent weight makes the ensemble model robust to the presence of some biased base models, as they will be assigned with small weights.

### 4.2 Certified robustness of SWEEN models

For a smoothed function \( g \), the certified radius at \((x, y)\) provided by Theorem 1 is \( r_c(x, y; g) = \text{clip} \left( \frac{2}{\sqrt{\pi}} \Phi^{-1} \left( \gamma_p(x, y) \right) - \Phi^{-1} \left( \max_{k \neq y} g_k(x) \right); 0, D \right) \). We formally define \( \gamma \)-robustness index as a metric of certified robustness.

**Definition 1.** \((\gamma \text{-robustness index}).\) For \( \gamma : \mathbb{R}^* \rightarrow \mathbb{R}^* \) and a smoothed function \( g \), the \( \gamma \)-robustness index of \( g \) is defined as
\[ I_\gamma(g) = \mathbb{E}(x,y) \sim \mathcal{D}(r_c(x, y; g)). \] (6)

It can be easily observed that \( \gamma \)-robustness index is an extension of many frequently-used criteria of certified robustness of smoothed classifiers.

**Proposition 1.** Let \( \gamma_1(r) = 1 \{ r \geq R \} \), \( \gamma_2(r) = r \), \( \gamma_3(r) = \frac{\pi^d}{1(\pi^d + 1)} r^d \). Then, \( \gamma_1 \)-robustness index is the certified accuracy at radius \( R \) [7]; \( \gamma_2 \)-robustness index is the average certified radius [56]; \( \gamma_3 \)-robustness index is the average volume of the certified region.

We note that criteria considering the volumes of the certified region are sometimes more comprehensive than those only considering the certified radii, as they take the input dimension into account.

Now consider \( \mathcal{F} = \{ f(\cdot; \theta) : \mathbb{R}^d \rightarrow \Delta | \theta \in \Theta \} \), the set of neural networks parametrized over \( \Theta \). The corresponding set of smoothed functions is \( \mathcal{G} = \{ g(x; \theta) = \mathbb{E}_{\theta \sim \mathcal{N}(0, \sigma^2 I)} f(x + \delta; \theta) | \theta \in \Theta \} \). Suppose we rely on some training algorithm to produce base neural networks, which outputs \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from a fixed probability distribution \( p \) on \( \Theta \). The set of SWEEN models is then
\[ \mathcal{F}_\theta = \left\{ \phi(x) = \sum_{k=1}^{K} w_k g(x; \theta_k) | w_k \geq 0, \sum_{k=1}^{K} w_k = 1 \right\}. \] (7)

Similar to Rahimi and Recht [17], we consider mixtures of form \( \phi(x) = \int_{\Theta} w(\theta) g(x; \theta) d\theta \). For a mixture \( \phi \), we define \( \| \phi \|_p := \sup_{\theta} \frac{|w(\theta)|}{w_p(\theta)} \). Define
\[ \mathcal{F}_p = \left\{ \phi(x) = \int_{\Theta} w(\theta) g(x; \theta) d\theta | \| \phi \|_p < \infty, w(\theta) \geq 0, \int_{\Theta} w(\theta) d\theta = 1 \right\}, \] (8)

note that for any \( \phi \in \mathcal{F}_p \), \( \phi \) is a smoothed probability function. We have the following result:

**Theorem 2.** Suppose \( \gamma \) is a Lipschitz function. Given \( \eta > 0 \). For any \( \epsilon > 0 \), for sufficiently large \( K \), with probability at least \( 1 - \eta \) over \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from \( p \), there exists \( \phi \in \mathcal{F}_\theta \) which satisfies
\[ I_\gamma(\phi) > \sup_{\phi \in \mathcal{F}_p} I_\gamma(\phi) - \epsilon. \] (9)

Moreover, if there exists \( \phi_0 \in \mathcal{F}_p \) such that \( I_\gamma(\phi_0) = \sup_{\phi \in \mathcal{F}_p} I_\gamma(\phi) \), \( K = \Omega \left( \frac{1}{\epsilon^2} \right) \).

Theorem 2 states that, if \( \gamma \) is a Lipschitz function and \( K \) is large enough, the best \( \gamma \)-robustness index for a SWEEN model is near the largest \( \gamma \)-robustness index attainable by functions in the class \( \mathcal{F}_p \) with very high probability.
4.3 Risk attainable by minimizing the empirical risk

Solving for ensemble weight \( w \) over a training set \( \{(x_i, y_i)\}_{i=1}^n \) can be formulated as

\[
\min_{w \in \Delta_K} \frac{1}{n} \sum_{i=1}^n l \left( \sum_{k=1}^K w_k g(x_i; \theta_k), y_i \right),
\]

where \( l : \mathbb{R}^M \times \mathcal{Y} \to \mathbb{R} \) is a surrogate loss function penalizing the predicted probability gap between the true class and the other classes. However, this process typically involves Monte Carlo simulation since we only have access to \( f(\cdot, \theta_k), k = 1, \ldots, K \).

**Definition 2.** (Risk and empirical risk). For a surrogate loss function \( l : \mathbb{R}^M \times \mathcal{Y} \to \mathbb{R} \), the risk of a probability function \( \phi \) are defined as

\[
\mathcal{R}[\phi] = \mathbb{E}_{(x, y) \sim \mathcal{D}} l(\phi(x), y).
\]

If \( \phi(x) = \sum_{k=1}^K w_k g(x; \theta_k) \in \mathcal{F}_\theta \), for training set \( \{(x_i, y_i)\}_{i=1}^n \) and sample size \( s \), the empirical risk of \( \phi \) is defined as

\[
\mathcal{R}_\text{emp}[\phi] = \frac{1}{n} \sum_{i=1}^n \left( \sum_{k=1}^K w_k \frac{1}{s} \sum_{j=1}^s f(x_i + \delta_{ijk}; \theta_k), y_i \right),
\]

where \( \delta_{ijk} \sim \mathcal{N}(0, \sigma^2 I), 1 \leq i \leq n, 1 \leq j \leq s, 1 \leq k \leq K \).

Now solving for \( w \) is equivalent to find the minimizer of \( \mathcal{R}_\text{emp} \). When the loss function \( l \) is convex, this problem is a low-dimensional convex optimization, so we can obtain the global empirical risk minimizer using traditional convex optimization algorithms. Furthermore, we have the following theorem:

**Theorem 3.** Suppose for all \( y \in \mathcal{Y} \), \( l(\cdot, y) \) is a Lipschitz function with constant \( L \) which satisfies \( l(0, y) = 0 \). Given \( \eta > 0 \). For any \( \varepsilon > 0 \), for sufficiently large \( K \), if \( n = \Omega\left( \frac{\sigma^2}{\varepsilon^2} \right), s = \Omega\left( \frac{\log K n}{\varepsilon^2} \right) \), then with probability at least \( 1 - \eta \) over the training dataset \( \{(x_i, y_i)\}_{i=1}^n \) drawn i.i.d. from \( \mathcal{D} \) and the parameters \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from \( p \) and the noise samples drawn i.i.d. from \( \mathcal{N}(0, \sigma^2 I) \), the empirical risk minimizer \( \hat{\phi} \) over \( \mathcal{F}_\theta \) satisfies

\[
\mathcal{R}[\hat{\phi}] - \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi] < \varepsilon.
\]

Moreover, if there exists \( \phi_0 \in \mathcal{F}_p \) such that \( \mathcal{R}[\phi_0] = \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi], K = \Omega\left( \frac{1}{\varepsilon^2} \right) \).

Theorem 3 gives a guarantee that, for a large enough \( K, n, s \), the gap between risk of the empirical risk minimizer \( \hat{\phi} \) and \( \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi] \) can be arbitrarily small with high probability. Note that we can solve \( \hat{\phi} \) to any given precision when \( l \) is convex.

One may wonder whether \( \mathcal{F}_p \) is big enough to capture some good smoothed functions. Intuitively \( \mathcal{F}_p \) is quite a rich set when \( g(\cdot; \theta) \) is expressive enough. We note that at least the closure of \( \mathcal{F}_p \) contains \( g(\cdot; \theta) \) whenever \( p(\theta) > 0 \) under some mild assumptions.

**Proposition 2.** Assume \( \Theta = \mathbb{R}^q \) for some \( q \in \mathbb{N} \). We further assume \( g(x; \theta) \) is uniformly continuous on \( \Theta \) w.r.t. \( x \) and \( p(\theta) \) is continuous on \( \Theta \). Given \( \theta_0 \) such that \( p(\theta_0) > 0 \), for any \( \varepsilon > 0 \), there exists \( \phi \in \mathcal{F}_p \) which satisfies \( \|g(x; \theta_0) - \phi(x)\|_2 < \varepsilon, \forall x \in \mathbb{R}^d \).

4.4 Adaptive prediction algorithm

A major drawback of ensembling is the high execution cost during inference, which is consisted of prediction cost and certification cost for smoothed classifiers. The evaluation of smoothed classifiers relies on Monte Carlo simulation, which is computationally expensive. For instance, Cohen et al. \[7\] use 100 Monte Carlo samples for prediction and 100,100 samples for certification. If we use 100 base models for ensembling, then the certification of a single data point will require 10,010,000 local evaluations. Inoue \[19\] observes that ensembling does not make improvements for inputs predicted with high probabilities even when they are mispredicted. He proposes an adaptive ensemble prediction algorithm to reduce the execution cost of unweighted ensemble models. We modify the algorithm to make it applicable to weighted ensemble models, which is detailed in Algorithm 1. For a data point, classifiers are evaluated in descending order with respect to their weights. Whenever an early-exit condition is satisfied, we stop the evaluation and return the current prediction as the output.
Algorithm 1 Adaptive prediction for weighted ensembling

1: **Input:** Ensembling weight \( w \in \mathbb{R}^K \), base model parameters \( \theta \in \Theta^K \), significance level \( \alpha \), threshold \( T \), data point \( x \)
2: Compute \( z = \Phi^{-1}(1 - \frac{\alpha}{2}) \)
3: Set \( \pi \) as the permutation of indices that sorts \( w \) in descending order and \( i \leftarrow 0 \)
4: repeat
   5: Set \( i \leftarrow i + 1 \)
   6: Compute the \( w_{\pi_i} \)-th local prediction \( p_{\pi_i} \leftarrow (p_{\pi_i,1}, \cdots, p_{\pi_i,M}) \in \Delta \)
   7: Compute \( \hat{p}_{i,k} = \frac{\sum_{j=1}^{i} w_{\pi_j} p_{\xi_k}}{\sum_{j=1}^{i} w_{\pi_j}} \) for \( k = 1, 2, \cdots, M \)
   8: Compute \( k_i \leftarrow \arg\max_k \hat{p}_{i,k} \)
   9: until \( \hat{p}_{1,k_1} > T \) or \( \hat{p}_{i,k_i} > \frac{1}{2} + \frac{z}{2} \sqrt{\frac{\sum_{j=1}^{i} w_{\pi_j}^2}{\sum_{j=1}^{i} w_{\pi_j}} \left( \frac{\sum_{j=1}^{i} w_{\pi_j} (p_{\xi_k,j} - p_{\xi_k,k})^2}{\sum_{j=1}^{i} w_{\pi_j}} \right)} \), \( i > 1 \) or \( i = K \)
10: return \( k_i \) and \( \hat{p}_{i,k_i}, k = 1, 2, \cdots, M \)

5 Experiments

We empirically evaluate the performance of SWEEN models on standard image classification datasets.

5.1 Setup

Model setup We train different network architectures on CIFAR-10 and SVHN to serve as base models for ensembling, including LeNet [25], AlexNet [21], ResNet-20, ResNet-26, ResNet-32, ResNet-110, DenseNet [18] (depth=100), VGG-16 [41], VGG-19. We particularly evaluate two compositions of the SWEEN models. The first is a relatively rich set of models, including LeNet, AlexNet, ResNet-20, ResNet-110, DenseNet, VGG-16, VGG-19, denoted by the 7-model-ensemble. The second is a small set of small models, including ResNet-20, ResNet-26, ResNet-32, denoted by the 3-model-ensemble. The 7-model-ensemble and the 3-model-ensemble simulate how much a weighted ensemble scheme can help in scenarios when we have an adequate and limited number of base models, respectively. The noise level \( \sigma \) are fixed across the smoothed ensemble model and its base models, as well as during training and certification.

Base model Training We train base models using two training schemes, including Gaussian data augmentation training [7], which is denoted as standard training for simplicity, and MACER training [56]. Base models for the same ensemble model are trained with an identical training scheme. All hyper-parameters used in our experiments are listed in Appendix B.1.

Solving the ensembling weight From Section 4 we know that we can obtain the empirical risk minimizer by solving a convex optimization. However, this requires first to approximate the value of smoothed functions of base models at every data point, which can be very costly when the number of base models and training data points is large. Hence, we use Gaussian data augmented training to solve the ensembling weight, which is much faster and yields comparable results empirically.

Certification Following previous works, we use the approximated certified accuracy (ACA), which is defined as the fraction of the test set that can be certified to be robust at radius \( r \), as the metric of performance. We also report the average certified radius (ACR) following Zhai et al. [56]. All results were certified with \( N = 100,000 \) samples and failure probability \( \alpha = 0.001 \).

5.2 Results

Standard training on CIFAR-10 Table 1 displays the performance of two kinds of SWEEN models under noise levels \( \sigma \in \{0.25, 0.50, 1.00\} \). We also report the upper envelopes of ACA and ACR of their corresponding base models and the performance of a single ResNet-110 for comparison. In Figure 1 we display the radius-accuracy curves for the ensemble models and all their respective base models under \( \sigma = 0.50 \) on CIFAR-10. For all noise levels, the 7-model-ensemble and the 3-model-ensemble both outperform the upper envelope of their corresponding base models and the single ResNet-110 at all radii.
Table 1: ACA (%) and ACR on CIFAR-10. All models are trained via standard training. * means the upper envelope of base models.

| σ  | Model          | 0.00 | 0.25 | 0.5  | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | ACR |
|----|----------------|------|------|------|------|------|------|------|------|------|-----|
|    | ResNet-110     | 79.6 | 65.2 | 50.8 | 34.4 | 0    | 0    | 0    | 0    | 0    | 0.489 |
| 0.25| 3-model*      | 80.5 | 65.6 | 47.9 | 30.2 | 0    | 0    | 0    | 0    | 0    | 0.470 |
|    | 3-model        | 82.3 | 69.8 | 54.7 | 35.9 | 0    | 0    | 0    | 0    | 0    | 0.520 |
|    | 7-model*       | 80.5 | 67.9 | 52.2 | 36.1 | 0    | 0    | 0    | 0    | 0    | 0.506 |
|    | 7-model        | 84.2 | 72.0 | 58.7 | 43.0 | 0    | 0    | 0    | 0    | 0    | 0.560 |
|    | ResNet-110     | 68.7 | 58.6 | 46.7 | 35.4 | 25.0 | 17.0 | 9.0  | 4.6  | 0    | 0.573 |
| 0.50| 3-model*      | 69.6 | 58.3 | 45.8 | 33.7 | 23.0 | 15.8 | 9.2  | 4.7  | 0    | 0.556 |
|    | 3-model        | 70.9 | 61.4 | 50.8 | 38.3 | 27.7 | 20.1 | 12.8 | 6.7  | 0    | 0.630 |
|    | 7-model*       | 68.6 | 58.9 | 46.6 | 34.8 | 24.7 | 16.5 | 10.2 | 5.3  | 0    | 0.574 |
|    | 7-model        | 71.2 | 63.0 | 52.2 | 41.9 | 31.2 | 22.9 | 15.3 | 8.3  | 0    | 0.678 |
|    | ResNet-110     | 51.4 | 44.9 | 37.9 | 31.8 | 24.6 | 18.8 | 13.8 | 10.2 | 6.7  | 0.559 |
| 1.00| 3-model*      | 50.6 | 44.7 | 38.2 | 30.8 | 24.6 | 18.5 | 13.6 | 10.5 | 7.0  | 0.555 |
|    | 3-model        | 51.9 | 45.5 | 39.3 | 32.3 | 25.9 | 19.7 | 15.4 | 11.4 | 8.1  | 0.595 |
|    | 7-model*       | 52.0 | 45.7 | 37.9 | 31.9 | 25.1 | 19.2 | 13.9 | 10.1 | 7.2  | 0.557 |
|    | 7-model        | 52.7 | 46.3 | 39.8 | 34.0 | 27.6 | 22.7 | 17.9 | 12.6 | 9.2  | 0.631 |

Figure 1: Radius-accuracy curves under $\sigma = 0.50$ on CIFAR-10. All models are trained via standard training. (Left) The 7-model-ensemble and all its base models. (Middle) The 3-model-ensemble and all its base models. (Right) The 7-model-ensemble, the 3-model-ensemble and the ResNet-110.

MACER training on CIFAR-10 The results are summarized in Table 2 and Table 3. For the approximate certified accuracy and ACR of the ResNet-110 model, we use the original numbers from Zhai et al. [56]. We can see that the performance of the 3-model-ensemble matches that of the ResNet-110. However, the total number of parameters of the 3-model ensemble is only approximately 64% of that of the ResNet-110. The total training time of the 3-model-ensemble (33.9 hours) is much less than that of the ResNet-110 (49.4 hours) as well.

Adaptive prediction ensembling We apply the previously mentioned adaptive prediction algorithm on the 7-model-ensemble models via standard training on CIFAR-10 to evaluate its effectiveness.

Table 2: ACA (%) and ACR on CIFAR-10. All models are trained via MACER training. * means the upper envelope of base models.

| σ  | Model          | 0.00 | 0.25 | 0.5  | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | ACR |
|----|----------------|------|------|------|------|------|------|------|------|------|-----|
| 0.25| ResNet-110     | 81   | 71   | 59   | 43   | 0    | 0    | 0    | 0    | 0    | 0.556 |
|     | 3-model*       | 77.4 | 66.9 | 56.8 | 41.9 | 0    | 0    | 0    | 0    | 0    | 0.529 |
|     | 3-model        | 77.7 | 68.7 | 60.3 | 46.6 | 0    | 0    | 0    | 0    | 0    | 0.558 |
| 0.50| ResNet-110     | 66   | 60   | 53   | 46   | 38   | 29   | 19   | 12   | 0    | 0.726 |
|     | 3-model*       | 64.9 | 57.1 | 49.7 | 41.1 | 34.1 | 26.2 | 20.2 | 11.7 | 0    | 0.685 |
|     | 3-model        | 64.7 | 58.4 | 51.8 | 43.9 | 37.2 | 29.2 | 22.8 | 14.6 | 0    | 0.725 |

7
Table 3: Training time for models under $\sigma = 0.50$ via MACER training.

| Model     | sec/epoch | #epochs | Total hrs |
|-----------|-----------|---------|-----------|
| ResNet-110| 404.2     | 440     | 49.4      |
| ResNet-20 | 72.2      | 440     | 8.8       |
| ResNet-26 | 92.9      | 440     | 11.3      |
| ResNet-32 | 113.2     | 440     | 13.8      |
| Weight    | 0.6       | 150     | 0.025     |
| Ensemble  | -         | -       | 33.9      |

Table 4: ACA (%) and ACR on CIFAR-10. All models are trained via standard training. * means the upper envelope of base models.

| $\sigma$ | Model    | 0.00 | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | ACR  | #evals/img |
|----------|----------|------|------|------|------|------|------|------|------|------|------------|
| 0.25     | Normal   | 84.2 | 72.0 | 58.7 | 43.0 | 0    | 0    | 0    | 0    | 0.560| 700,700    |
|          | Adaptive | 84.3 | 71.5 | 57.6 | 41.2 | 0    | 0    | 0    | 0    | 0.549| 283,727    |
|          | Normal   | 71.2 | 63.0 | 52.2 | 41.9 | 31.2 | 22.9 | 15.3 | 8.3  | 0.678| 700,700    |
|          | Adaptive | 70.9 | 62.8 | 52.3 | 41.6 | 31.1 | 22.8 | 14.5 | 7.8  | 0.672| 382,426    |

The results are summarized in Table 4. It can be observed that the adaptive prediction models require much fewer evaluations to certify an image. However, the performance of the adaptive prediction models is only slightly worse than their vanilla counterparts.

Results on SVHN To further evaluate our method, we also experiment on SVHN. The results show that SWEEN models outperform the upper envelopes of their corresponding base models as well. Figure 2 plots the results on CIFAR-10 and SVHN for comparison.

Figure 2: Comparing ensembles to the upper envelopes of their corresponding base models. All models are trained via standard training. (Left) The 3-model-ensemble on CIFAR-10. (Middle) The 7-model-ensemble on CIFAR-10. (Right) The 7-model-ensemble on SVHN.

6 Conclusions

In this work, we introduced the smoothed weighted ensembling (SWEEN) to improve randomized smoothed classifiers in terms of both the accuracy and robustness. We theoretically demonstrated the certified robustness and risk attainable by SWEEN models. Moreover, we developed an adaptive prediction algorithm to accelerate the prediction and certification process of SWEEN models. Our extensive experiments showed that an properly designed weighted ensemble model was able to consistently outperform all its base models by a significant margin. This suggested that SWEEN is a viable tool for improving the performance of randomized smoothing models.
Broader Impact

Randomized smoothing, as a provable defense against adversarial examples, helps to defend against potential adversaries who will attack the system. In this work we employ SWEEN to improve the performance of randomized smoothed classifiers, which may lead to more robust machine learning systems in the real world application. At the same time, SWEEN models often have more computation cost comparing to a single model on prediction and certification phase, which may have some negative consequences in the society. Furthermore, we should be cautious of the result of failure of the system which could cause a drop of the performance, or a waste of computational resources.

Acknowledgments

We would like to thank Dachao Lin and Haocheng Ju for useful discussions. Yunzhen Feng is partially supported by the elite undergraduate training program of School of Mathematical Sciences in Peking University. Bin Dong is supported in part by Beijing Natural Science Foundation (No. 180001) and Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In ICML 2018: Thirty-fifth International Conference on Machine Learning, pages 274–283, 2018.

[2] Peter Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. The Journal of Machine Learning Research, 3:224–240, 06 2001. doi: 10.1007/3-540-44581-1_15.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. European conference on machine learning, 8190:387–402, 2013.

[4] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random smoothing might be unable to certify $\ell_\infty$ robustness for high-dimensional images. arXiv preprint arXiv:2002.03517, 2020.

[5] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In Proceedings of the 33rd Annual Computer Security Applications Conference on, pages 278–287, 2017.

[6] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 3–14, 2017.

[7] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In ICML 2019: Thirty-sixth International Conference on Machine Learning, pages 1310–1320, 2019.

[8] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 2057–2066, 2019.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[10] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O’Donoghue, Jonathan Usato, and Pushmeet Kohli. Training verified learners with learned verifiers. arXiv preprint arXiv:1805.10265, 2018.

[11] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In UAI 2018: The Conference on Uncertainty in Artificial Intelligence (UAI), pages 550–559, 2018.

[12] Rüdiger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In International Symposium on Automated Technology for Verification and Analysis, pages 269–286, 2017.
[13] Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. *arXiv preprint arXiv:1712.06174*, 2017.

[14] Timon Gehr, Matthew Mirman, Dana Drachslr-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2018.

[15] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[16] L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[19] Hiroshi Inoue. Adaptive ensemble prediction for deep neural networks based on confidence level. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1284–1293, 2019.

[20] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117, 2017.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[22] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238, 1994.

[23] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. *arXiv preprint arXiv:2002.03239*, 2020.

[24] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR 2017: International Conference on Learning Representations 2017, 2017*.

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[26] 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)*, pages 656–672, 2019.

[27] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi S. Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 4911–4922, 2019.

[28] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. 2018.

[29] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 381–397, 2018.

[30] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR 2018: International Conference on Learning Representations 2018, 2018*.

[32] Ying Meng, Jianhai Su, Jason O’Kane, and Pooyan Jamshidi. Ensembles of many diverse weak defenses can be strong: Defending deep neural networks against adversarial attacks. *arXiv preprint arXiv:2001.00308*, 2020.
[33] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In ICML 2018: Thirty-fifth International Conference on Machine Learning, pages 3578–3586, 2018.

[34] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In ICML 2019: Thirty-sixth International Conference on Machine Learning, pages 4970–4979, 2019.

[35] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In ICLR 2018: International Conference on Learning Representations 2018, 2018.

[36] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems, pages 10877–10887, 2018.

[37] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In Advances in Neural Information Processing Systems 21, pages 1313–1320, 2008.

[38] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In NeurIPS 2019: Thirty-third Conference on Neural Information Processing Systems, pages 11292–11303, 2019.

[39] Jiaye Teng, Guang-He Lee, and Yang Yuan. \ell_1 adversarial robustness certificates: a randomized smoothing approach. 2019.

[40] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems, pages 10802–10813, 2018.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In ICLR 2014: International Conference on Learning Representations (ICLR) 2014, 2014.

[42] Jonathan Uesato, Brendan O’Donoghue, Pushmeet Kohli, and Aäron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In ICML 2018: Thirty-fifth International Conference on Machine Learning, pages 5025–5034, 2018.

[43] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. arXiv preprint arXiv:2002.08347, 2020.

[44] Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of formally robust neural networks. arXiv preprint arXiv:1811.02625, 2018.

[45] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems, pages 6367–6377, 2018.
Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, pages 5273–5282, 2018.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, pages 5283–5292, 2018.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial defenses. In *NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems*, pages 8400–8409, 2018.

Greg Yang, Tony Duan, Edward Hu, Hadi Salman, Ilya P. Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. *arXiv preprint arXiv:2002.08118*, 2020.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *ICLR 2020: Eighth International Conference on Learning Representations*, 2020.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems*, pages 4939–4948, 2018.
A Proofs

A.1 Proof of Theorem 2

Define
\[
\mathcal{F}_p' = \left\{ \phi(x) = \int_\Theta w(\theta)g(x; \theta)d\theta \mid \|\phi\|_p < \infty, w(\theta) \geq 0 \right\},
\]
\[
\mathcal{F}_\theta' = \left\{ \phi(x) = \sum_{k=1}^K w_k g(x; \theta_k) \mid w_k \geq 0 \right\}.
\]

We have \(\mathcal{F}_p \subseteq \mathcal{F}_p', \mathcal{F}_\theta \subseteq \mathcal{F}_\theta'.\)

**Lemma 1.** Let \(\mu\) be any probability measure on \(\mathbb{R}^d\). For \(\phi : \mathbb{R}^d \rightarrow \mathbb{R}^M\), define the norm \(\|\phi\|_\mu^2 \triangleq \int_{\mathbb{R}^d} \|\phi(x)\|_2^2 d\mu(x)\). Fix \(\phi \in \mathcal{F}_p',\) then for any \(\eta > 0\), with probability at least \(1 - \eta\) over \(\theta_1, \ldots, \theta_K\) drawn i.i.d. from \(\mu\), there exists \(\hat{\phi}(x) = \sum_{k=1}^K c_k g(x; \theta_k) \in \mathcal{F}_\theta'\) which satisfies
\[
\|\hat{\phi} - \phi\|_\mu \leq \frac{\|\phi\|_p}{\sqrt{K}} \left(1 + \sqrt{\frac{2 \log \frac{1}{\eta}}{K}}\right).
\]

**Proof.** Since \(\phi \in \mathcal{F}_p',\) we can write \(\phi(x) = \int_\Theta w(\theta)g(x; \theta)d\theta,\) where \(w(\theta) \geq 0.\) Construct \(\phi_k = \beta_k g(\theta_k), k = 1, 2, \ldots, K,\) where \(\beta_k = \frac{w(\theta_k)}{p(\theta_k)},\) then \(\mathbb{E}\phi_k = \phi, \|\phi_k\|_\mu = \sqrt{\int_{\mathbb{R}^d} \beta_k^2 \|g(x; \theta_k)\|_2^2 d\mu(x)} \leq \|\beta_k\| \leq \|\phi\|_p.\) We then define
\[
u(\theta_1, \ldots, \theta_K) = \left| \frac{1}{K} \sum_{k=1}^K \phi_k - \phi \right|_\mu.
\]
First, by using Jensen’s inequality and the fact that \(\|\phi_k\|_\mu \leq \|\phi\|_p,\) we have
\[
\mathbb{E}[\nu(\theta)] \leq \sqrt{\mathbb{E}[\nu^2(\theta)]} = \sqrt{\mathbb{E} \left[ \left| \frac{1}{K} \sum_{k=1}^K \phi_k - \mathbb{E}\phi_k \right|_\mu^2 \right]} = \sqrt{\frac{1}{K} (\mathbb{E}[\|\phi_k\|_\mu^2 - \|\mathbb{E}\phi_k\|_\mu^2])} \leq \frac{\|\phi\|_p}{\sqrt{K}}.
\]
Next, for \(\theta_1, \ldots, \theta_M\) and \(\tilde{\theta}_i,\) we have
\[
\left| \nu(\theta_1, \ldots, \theta_M) - \nu(\theta_1, \ldots, \tilde{\theta}_i, \ldots, \theta_M) \right| = \left| \frac{1}{K} \sum_{k=1}^K \phi_k - \phi \right|_\mu - \left| \frac{1}{K} \left( \sum_{k=1, k \neq i}^M \phi_k + \phi_i \right) - \phi \right|_\mu \leq \left| \frac{1}{K} \sum_{k=1}^M \phi_k - \frac{1}{K} \left( \sum_{k=1, k \neq i}^M \phi_k + \phi_i \right) \right|_\mu = \frac{\|\phi_i - \hat{\phi}_i\|_\mu}{K} \leq \frac{2 \|\phi\|_p}{K}.
\]
Now we can use McDiarmid’s inequality to bound \(\nu(\theta),\) which gives
\[
\mathbb{P}[\nu(\theta) - \frac{\|\phi\|_p}{\sqrt{K}} \geq \epsilon] \leq \mathbb{P} u(\theta) - \mathbb{E} u(\theta) \geq \frac{\epsilon}{2} \leq \exp\left( - \frac{K \epsilon^2}{2 \|\phi\|_p^2} \right).
\]
The theorem follows by setting \(\delta\) to the right hand side and solving \(\epsilon.\)
Lemma 2. Let $\mu$ be any probability measure on $\mathbb{R}^d$. For $\phi : \mathbb{R}^d \to \mathbb{R}^M$, define the norm $\| \phi \|^2_\mu \triangleq \int_{\mathbb{R}^d} \| \phi(x) \|^2_2 \, d\mu(x)$, then for any $\eta > 0$, for $K \geq M \| \phi \|^2_p (1 + \sqrt{2 \log \frac{1}{\eta}})^2$, with probability at least $1 - \eta$ over $\theta_1, \ldots, \theta_K$ drawn i.i.d. from $p$, there exists $\hat{\phi}(x) = \sum_{k=1}^K c_k g(x; \theta_k) \in \mathcal{F}_\theta$ which satisfies

$$\| \hat{\phi} - \phi \|_\mu < 2 \sqrt{\| \phi \|_p} \sqrt{\frac{M}{K}} (1 + \sqrt{2 \log \frac{1}{\eta}})^\frac{1}{2}. \quad (19)$$

Proof. Fix $\phi \in \mathcal{F}_p \subseteq \mathcal{F}_p'$, by using Lemma 1 we have that for any $\delta > 0$, with probability at least $1 - \eta$ over $\theta_1, \ldots, \theta_K$ drawn i.i.d. from $p$, there exists $\tilde{\phi}(x) = \sum_{k=1}^K c_k g(x; \theta_k) \in \mathcal{F}_\theta'$ which satisfies

$$\| \tilde{\phi} - \phi \|_\mu < \frac{\| \phi \|_p}{\sqrt{K}} (1 + \sqrt{2 \log \frac{1}{\eta}}) \triangleq B(K). \quad (20)$$

Denote $C = \sum_{k=1}^K c_k$, and define $s(t) \triangleq \sum_{i=1}^M t_i$ as the sum of all elements of $t \in \mathbb{R}^M$. Then $s(g(x; \theta)) = 1, \forall x \in \mathbb{R}^d, \theta \in \Theta$. Thus,

$$s(\phi(x)) = \sum_{i=1}^M \phi_i(x) = \sum_{i=1}^M \int_\Theta w(\theta) g_i(x; \theta) d\theta = \int_\Theta w(\theta) \sum_{i=1}^M g_i(x; \theta) d\theta = \int_\Theta w(\theta) d\theta = 1,$n

$$s(\tilde{\phi}(x)) = \sum_{i=1}^M \tilde{\phi}_i(x) = \sum_{i=1}^M \sum_{k=1}^K c_k g_i(x; \theta_k) = \sum_{k=1}^K \sum_{i=1}^M c_k g_i(x; \theta_k) = \sum_{k=1}^K c_k = C.$n

Now we have

$$B(K)^2 > \| \tilde{\phi} - \phi \|^2_\mu = \int_{\mathbb{R}^d} \| \tilde{\phi}(x) - \phi(x) \|^2_2 \, d\mu(x) \geq \int_{\mathbb{R}^d} \frac{(s(\tilde{\phi}(x) - \phi(x)))^2}{M} \, d\mu(x) = \frac{(C - 1)^2}{M} \int_{\mathbb{R}^d} \, d\mu(x) = \frac{(C - 1)^2}{M},$$

which gives $1 - \sqrt{MB(K)} < C < 1 + \sqrt{MB(K)}$. Construct $\hat{\phi}(x) = \frac{\tilde{\phi}(x)}{C}$, then $\hat{\phi} \in \mathcal{F}_\theta$ and

$$\| \hat{\phi} - \phi \|^2_\mu = \int_{\mathbb{R}^d} \| \hat{\phi}(x) - \phi(x) \|^2_2 \, d\mu(x) \geq \int_{\mathbb{R}^d} \| \tilde{\phi}(x) - \phi(x) \|^2_2 \, d\mu(x) \geq \int_{\mathbb{R}^d} \| (\hat{\phi}(x) - \phi(x)) + (C^{-1} - 1)\hat{\phi}(x) \|^2_2 \, d\mu(x) = \int_{\mathbb{R}^d} \| (\hat{\phi}(x) - \phi(x)) \|^2_2 + \| (C^{-1} - 1)\hat{\phi}(x) \|^2_2 + 2(C^{-1} - 1)\langle \hat{\phi}(x) - \phi(x), \hat{\phi}(x) \rangle \, d\mu(x) = \int_{\mathbb{R}^d} \| \hat{\phi}(x) - \phi(x) \|^2_2 + (C^{-2} - 1)\| \hat{\phi}(x) \|^2_2 + 2(C^{-1} - 1)\langle \phi(x), \hat{\phi}(x) \rangle \, d\mu(x).$$

Sine we have $\frac{C^2}{M} \leq \| \hat{\phi}(x) \|^2_2 \leq C^2, |\langle \phi(x), \hat{\phi}(x) \rangle| \leq \sqrt{\| \phi(x) \|^2_2 \| \hat{\phi}(x) \|^2_2} \leq C$, it holds that
The desired result follows from Lemma 2.

Let \( p \) with respect to \( m \)

Given any \( \text{Corollary of Proposition 1 in Zhai et al. [56]} \)

Lemma 4.

Proof.

there exists \( \geq \)

Lemma 3.

Suppose \( 1 \geq \cdots \geq 1 \).

Apply Lemma 3, we have that for \( \geq \)

and \( \geq \)

Thus, with probability at least \( 1 - \eta \) over \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from \( p \),

\[
\| \hat{\phi} - \phi \|_\mu < 2 \sqrt{MB(K)} = 2 \sqrt{\|\phi\|_p \sqrt{M \over K}} (1 + \sqrt{2 \log \frac{1}{\eta}})^{1/2}.
\]

Lemma 3. Suppose \( \ell(\cdot, \cdot) \) is \( L \)-Lipschitz in its first argument. Fix \( \phi \in \mathcal{F}_p \), then for any \( \eta > 0 \), for \( K \geq M \|\phi\|_p^2 (1 + \sqrt{2 \log \frac{1}{\eta}})^2 \), with probability at least \( 1 - \eta \) over \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from \( p \), there exists \( \hat{\phi} \in \mathcal{F}_\theta \) which satisfies

\[
|E_{(x, y) \sim D}[l(\hat{\phi}(x), y)] - E_{(x, y) \sim D}[l(\phi(x), y)]| < 2L \sqrt{\|\phi\|_p \sqrt{M \over K}} (1 + \sqrt{2 \log \frac{1}{\eta}})^{1/2}.
\]

Proof.

\[
|E[l(\hat{\phi}(x), y)] - E[l(\phi(x), y)]| \leq E|c(\phi(x), y) - c(\hat{\phi}(x), y)| \leq L E\|\phi(x) - \hat{\phi}(x)\|_2 \leq L \sqrt{E\|\phi(x) - \hat{\phi}(x)\|_2^2} = L \|\phi - \hat{\phi}\|_{D|x}.
\]

The desired result follows from Lemma 2.

Lemma 4. (Corollary of Proposition 1 in Zhai et al. [56]) Given any \( p_1, p_2, \ldots, p_M \) satisfies \( p_1 \geq p_2 \geq \cdots \geq p_M \geq 0 \) and \( p_1 + p_2 + \cdots + p_M = 1 \). The derivative of \( \text{clip} \left( \frac{\Phi^{-1}(p_1) - \Phi^{-1}(p_2)}{2}, D \right) \) with respect to \( p_1 \) and \( p_2 \) is bounded.

Now we can prove Theorem 2.

Proof of Theorem 2. Let \( \phi_0 \in \mathcal{F}_p \) such that \( I_\gamma(\phi_0) > \sup_{\phi \in \mathcal{F}_p} I_\gamma(\phi) - \frac{\delta}{2} \). From Lemma 4 we know that \( q(p, y) \) defined by \( \text{clip} \left( \frac{\Phi^{-1}(p_1) - \Phi^{-1}(p_2)}{2}, D \right) \) is Lipschitz in its first argument. Since \( m \) is Lipschitz, \( c(p, y) \) also Lipschitz in its first argument with some constant \( L \). Hence, we have that for \( K \geq M \|\phi\|_p^2 (1 + \sqrt{1 + 2 \log \frac{1}{\delta}})^2 \), with probability at least \( 1 - \eta \) over \( \theta_1, \ldots, \theta_K \) drawn i.i.d. from \( p \), there exists \( \phi \in \mathcal{F}_\theta \) which satisfies

\[
I_\gamma(\phi_0) - I_\gamma(\phi) = E_{(x, y) \sim D}[l(\phi_0(x), y)] - E_{(x, y) \sim D}[l(\phi(x), y)] < 2L \sqrt{\|\phi_0\|_p \sqrt{M \over K}} (1 + \sqrt{2 \log \frac{1}{\eta}})^{1/2}.
\]
When \( K > \frac{256L^2\|\phi_0\|_p^2M(1+\sqrt{2\log \frac{d}{2}})^2}{\epsilon^4} \), we have
\[
\sup_{\phi \in \mathcal{F}_{\rho}} \mathcal{I}_\gamma(\phi) - \mathcal{I}_\gamma(\hat{\phi}) = (\sup_{\phi \in \mathcal{F}_{\rho}} \mathcal{I}_\gamma(\phi) - \mathcal{I}_\gamma(\phi_0)) + (\mathcal{I}_\gamma(\phi_0) - \mathcal{I}_\gamma(\hat{\phi})) < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.
\]
If \( \mathcal{I}_\gamma(\phi_0) = \sup_{\phi \in \mathcal{F}_{\rho}} \mathcal{I}_\gamma(\phi) \), which means \( \|\phi_0\|_p \) is independent of \( \epsilon, K = \Omega(\frac{1}{\epsilon^2}) \).

**A.2 Proof of Theorem 3**

First we introduce some results from statistical learning theory.

**Definition 3.** (Gaussian complexity) Let \( \mu \) be a probability distribution on a set \( \mathcal{X} \) and suppose that \( x_1, \ldots, x_n \) are independent samples selected according to \( \mu \). Let \( \mathcal{F} \) be a class of functions mapping from \( \mathcal{X} \) to \( \mathbb{R} \). The Gaussian complexity of \( \mathcal{F} \) is
\[
G_n[\mathcal{F}] \triangleq \mathbb{E}[\sup_{f \in \mathcal{F}} 2n \sum_{i=1}^n \xi_i f(x_i) | x_1, \ldots, x_n, \xi_1, \ldots, \xi_n]
\]
where \( \xi_1, \ldots, \xi_n \) are independent \( \mathcal{N}(0, 1) \) random variables.

**Definition 4.** (Rademacher complexity) Let \( \mu \) be a probability distribution on a set \( \mathcal{X} \) and suppose that \( x_1, \ldots, x_n \) are independent samples selected according to \( \mu \). Let \( \mathcal{F} \) be a class of functions mapping from \( \mathcal{X} \) to \( \mathbb{R} \). The Rademacher complexity of \( \mathcal{F} \) is
\[
R_n[\mathcal{F}] \triangleq \mathbb{E}[\sup_{f \in \mathcal{F}} 2n \sum_{i=1}^n \sigma_i f(x_i) | x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n]
\]
where \( \sigma_1, \ldots, \sigma_n \) are independent uniform \( \{-1, 1\} \)-valued random variables.

**Lemma 5.** (Part of Lemma 4 in Bartlett and Mendelson [2]) There are absolute constants \( \beta \) such that for every class \( \mathcal{F} \) and every integer \( n \), \( R_n(\mathcal{F}) \leq \beta G_n(\mathcal{F}) \).

**Lemma 6.** (Corollary of Theorem 8 in Bartlett and Mendelson [2]) Consider a loss function \( c : \mathcal{A} \times \mathcal{Y} \to [0, 1] \). Let \( \mathcal{F} \) be a class of functions mapping from \( \mathcal{X} \) to \( \mathcal{A} \) and let \( (x_i, y_i)_{i=1}^n \) be independently selected according to the probability measure \( \mu \). Then, for any integer \( n \) and any \( 0 < \eta < 1 \), with probability at least \( 1 - \eta \) over samples of length \( n \), every \( f \in \mathcal{F} \) satisfies
\[
\mathbb{E}_{(x,y) \sim \mu} [c(f(x), y)] \leq \frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) + R_n[\hat{c} \circ \mathcal{F}] + \sqrt{\frac{8 \log \frac{2}{\eta}}{n}},
\]
where \( \hat{c} \circ \mathcal{F} = \{(x, y) \mapsto c(f(x), y) - c(0, y) | f \in \mathcal{F} \} \)

**Lemma 7.** (Corollary of Theorem 14 in Bartlett and Mendelson [2]) Let \( \mathcal{A} = \mathbb{R}^M \) and let \( \mathcal{F} \) be a class of functions mapping from \( \mathcal{X} \) to \( \mathcal{A} \). Suppose that there are real-valued classes \( \mathcal{F}_1, \ldots, \mathcal{F}_M \) such that \( \mathcal{F} \) is a subset of their Cartesian product. Assume further that \( c : \mathcal{A} \times \mathcal{Y} \to \mathbb{R} \) is such that, for all \( y \in \mathcal{Y} \), \( c(\cdot, y) \) is a Lipschitz function with constant \( L \) which passes through the origin and is uniformly bounded. Then
\[
G_n(c \circ \mathcal{F}) \leq 2L \sum_{i=1}^M G_n(\mathcal{F}_i).
\]

Now we prove the following lemma:

**Lemma 8.** Let \( c, \mathcal{F}, (x_i, y_i)_{i=1}^n, \hat{c} \circ \mathcal{F} \) be as in Lemma 5. Then, for any integer \( n \) and any \( 0 < \eta < 1 \), with probability at least \( 1 - \eta \) over samples of length \( n \), every \( f \in \mathcal{F} \) satisfies
\[
\frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) - \mathbb{E}_{(x,y) \sim \mu} [c(f(x), y)] \leq \sup_{h \in \hat{c} \circ \mathcal{F}} (\hat{E}_n h - E h) + \frac{8 \log \frac{2}{\eta}}{n}.
\]

**Proof.**
\[
\frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) - \mathbb{E}_{(x,y) \sim \mu} [c(f(x), y)] \leq \sup_{h \in \hat{c} \circ \mathcal{F}} (\hat{E}_n h - E h) = \sup_{h \in \hat{c} \circ \mathcal{F}} (\hat{E}_n h - E h) + \hat{E}_n c(0, y) - E c(0, y).
\]
When an \((x_i, y_i)\) pair changes, the random variable \( \sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) \) can change by no more than \(\frac{2}{n}\). McDiarmid’s inequality implies that with probability at least \(1 - \frac{\eta}{2}\),

\[
\sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) \leq \mathbb{E} \sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) + \sqrt{\frac{2 \log \frac{2}{\eta}}{n}}.
\]

A similar argument, together with the fact that \(\mathbb{E}\hat{E}_n c(0, y) = \mathbb{E}c(0, y)\), shows that with probability at least \(1 - \eta\),

\[
\mathbb{E}_{\text{emp}}[f] \leq \mathbb{E}[f] + \mathbb{E} \sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) + \sqrt{\frac{8 \log \frac{2}{\eta}}{n}}.
\]

It’s left to show that \(\mathbb{E} \sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) \leq \mathcal{R}_n[\hat{c} \circ \mathcal{F}]\). Let \((x'_1, y'_1), \ldots, (x'_n, y'_n)\) be drawn i.i.d. from \(\mu\) and independent from \((x_i, y_i)_{i=1}^n\), then

\[
\mathbb{E} \sup_{h \in \mathcal{G}} (\hat{E}_n h - \mathbb{E}h) = \mathbb{E} \sup_{h \in \mathcal{G}} \mathbb{E}[\hat{E}_n h - \frac{1}{n} \sum_{i=1}^n h(x'_i, y'_i)]
\]

\[
\leq \mathbb{E} \mathbb{E} \sup_{h \in \mathcal{G}} \hat{E}_n h - \frac{1}{n} \sum_{i=1}^n h(x'_i, y'_i)
\]

\[
= \mathbb{E} \sup_{h \in \mathcal{G}} \frac{1}{n} \left( \sum_{i=1}^n h(x_i, y_i) - \sum_{i=1}^n h(x'_i, y'_i) \right)
\]

\[
\leq 2 \mathbb{E} \sup_{h \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n \sigma_i h(x_i, y_i)
\]

\[
\leq \mathcal{R}_n[\hat{c} \circ \mathcal{F}].
\]

We can prove the following result:

**Theorem 4.** Let \(\mathcal{A} = \mathbb{R}^M\) and let \(\mathcal{F}\) be a class of functions mapping from \(X\) to \(\mathcal{A}\). Suppose that there are real-valued classes \(\mathcal{F}_1, \ldots, \mathcal{F}_M\) such that \(\mathcal{F}\) is a subset of their Cartesian product. Assume further that the loss function \(c: \mathcal{A} \times Y \to \mathbb{R}\) is such that, for all \(y \in Y\), \(c(\cdot, y)\) is a Lipschitz function with constant \(L\) which passes through the origin and is uniformly bounded. Let \(\{(x_i, y_i)\}_{i=1}^n\) be independently selected according to the probability measure \(\mu\). Then, for any integer \(n\) and any \(0 < \eta < 1\), there is a probability of at least \(1 - \eta\) that every \(f \in \mathcal{F}\) has

\[
|\frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) - \mathbb{E}_{(x,y) \sim \mu}[c(f(x), y)]| \leq \beta L \sum_{j=1}^M G_j[\mathcal{F}_j] + \sqrt{\frac{8 \log \frac{1}{\eta}}{n}},
\]

where \(\beta\) is a constant.

**Proof.** From Lemma 6 and 8 we have that with probability at least \(1 - \eta\) over samples of length \(n\), every \(f\) in \(\mathcal{F}\) satisfies

\[
|\frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) - \mathbb{E}_{(x,y) \sim \mu}[c(f(x), y)]| \leq \mathcal{R}_n[\hat{c} \circ \mathcal{F}] + \sqrt{\frac{8 \log \frac{1}{\eta}}{n}},
\]

it follows by applying Lemma 5 and 7.

**Lemma 9.** Let \(c(\cdot, \cdot)\), \(\beta\) be as in Theorem 4. Let \((x_i, y_i)_{i=1}^n\) be independently selected according to the probability measure \(\mathcal{D}\). For any integer \(n\) and any \(0 < \eta < 1\), there is a probability of at least \(1 - \eta\) that every \(f \in \mathcal{F}_0\) has

\[
|\frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) - \mathbb{E}_{(x,y) \sim \mathcal{D}}[c(f(x), y)]| \leq \frac{2\beta L M K}{\sqrt{n}} + \sqrt{\frac{8 \log \frac{1}{\eta}}{n}},
\]
We have that \( \hat{\mathbb{F}}_\theta(i) = \left\{ \phi(x) = \sum_{k=1}^{K} w_k g_i(x; w_k) \middle| w_k \geq 0, \sum_{k=1}^{K} w_k = 1 \right\} \), \( 1 \leq i \leq M \).

Theorem 5. Suppose for all \( \hat{\mathbb{F}}_\theta \), the desired result follows by applying Theorem 4 to \( \beta \) where \( \phi \) drawn \( i.i.d. \) least through the origin and is uniformly bounded. Fix \( \phi \) training set \( \{ (Semi-empirical risk) \). For a surrogate loss function \( l(\cdot, \cdot) : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R} \) and training set \( \{(x_i, y_i)\}_{i=1}^n \), the semi-empirical risk of \( \phi(x) = \sum_{k=1}^{K} w_k g_i(x; \theta_k) \in \hat{\mathbb{F}}_\theta \) are defined as

\[
\hat{\mathcal{R}}_{\text{se}}[\phi] = \frac{1}{n} \sum_{i=1}^{n} l(\sum_{k=1}^{K} w_k g_i(x_i; \theta_k), y_i).
\]  

(21)

We can use Lemma 3 and 9 to prove the following result:

Theorem 5. Suppose for all \( y \in \mathcal{Y} \), \( l(\cdot, \cdot) \) is a Lipschitz function with constant \( L \) which passes through the origin and is uniformly bounded. Fix \( \phi \in \mathbb{F}_p \), then for any \( \eta > 0 \), with probability at least \( 1 - \eta \) over the training dataset \( \{(x_i, y_i)\}_{i=1}^n \) drawn i.i.d. from \( \mathcal{D} \) and the parameters \( \theta_1, ..., \theta_K \) drawn i.i.d. from \( p \), the semi-empirical risk minimizer \( \hat{\phi} \) over \( \hat{\mathbb{F}}_\theta \) satisfies

\[
\mathcal{R}[\hat{\phi}] - \mathcal{R}[\phi] < \frac{4\beta LMK + 4 \sqrt{2 \log \frac{2}{\eta}}}{\sqrt{n}} + 2L \sqrt{||\phi||_p} \left( \frac{M}{K} \right) (1 + \sqrt{2 \log \frac{2}{\eta}})^{\frac{1}{2}},
\]

where \( \beta \) is a constant.
Proof. Let $\phi^*$ be the minimizer of $\mathcal{R}$ over $\mathcal{F}_\theta$. Combine Lemma 3 and 9, we derive that, with probability at least $1 - 2\delta$ over the training dataset and the choice of the parameters $\theta_1, \ldots, \theta_K$,

\[
\mathcal{R}[\hat{\phi}] - \mathcal{R}[\phi] = (\mathcal{R}[\hat{\phi}] - \mathcal{R}_{ac}[\hat{\phi}]) + (\mathcal{R}_{ac}[\hat{\phi}] - \mathcal{R}_{ac}[\phi^*]) + (\mathcal{R}_{ac}[\phi^*] - \mathcal{R}[\phi^*]) + (\mathcal{R}[\phi^*] - \mathcal{R}[\phi]) \\
\leq \frac{2\beta L MK}{\sqrt{n}} + \frac{2\beta L MK + 2\sqrt{2\log \frac{4}{\eta}}}{\sqrt{n}} + 2L \sqrt{\|\phi\|_p} \sqrt{\frac{M}{K}} (1 + \sqrt{2\log \frac{1}{\eta}}) + \frac{4\beta L MK}{\sqrt{n}} + 2L \sqrt{\|\phi\|_p} \sqrt{\frac{M}{K}} (1 + \sqrt{2\log \frac{1}{\eta}}) + \frac{2n}{n}.
\]

Lemma 10. Let $\mu$ be a probability distribution on $\Delta$. For any $\eta > 0$, with probability at least $1 - \eta$ over $x_1, \ldots, x_s$ drawn i.i.d. from $\mu$, it holds that

\[
\| \frac{1}{s} \sum_{i=1}^{s} x_i - \mathbb{E}_{x \sim \mu}[x] \|_2 \leq \frac{1}{\sqrt{s}} (1 + \sqrt{2\log \frac{1}{\eta}})
\] (22)

Proof. Define $u(x_1, \cdots, x_s) = \| \frac{1}{s} \sum_{i=1}^{s} x_i - \mathbb{E}[x] \|_2$. By using Jensen’s inequality, we have

\[
\mathbb{E}[u(x)] \leq \sqrt{\mathbb{E}[u^2(x)]} = \sqrt{\mathbb{E}[\| \frac{1}{s} \sum_{i=1}^{s} x_i - \mathbb{E}[x] \|_2^2]} = \sqrt{\frac{1}{s} (\mathbb{E}[\|x\|_2^2] - \|\mathbb{E}[x]\|_2^2)} \leq \frac{1}{\sqrt{s}}.
\]

Next, for $x_1, \cdots, x_M$ and $\bar{x}_k$, we have

\[
|u(x_1, \cdots, x_s) - u(x_1, \cdots, \bar{x}_k, \cdots, x_s)| \\
= \| \frac{1}{s} \sum_{i=1}^{s} x_i - \mathbb{E}[x] \|_2 - \| \frac{1}{s} (\sum_{i=1, i \neq k}^{s} x_i + \bar{x}_k) - \mathbb{E}[x] \|_2 \\
\leq \| \frac{1}{s} \sum_{i=1}^{s} x_i - \frac{1}{s} \left( \sum_{i=1, i \neq k}^{s} x_i + \bar{x}_k \right) \|_2 \\
= \| \bar{x}_k - \bar{x}_k \|_2 \\
\leq \frac{2}{s}.
\]

Now we can use McDiarmid’s inequality to bound $u(x)$, which gives

\[
\mathbb{P}[u(x) - \frac{1}{\sqrt{s}} \geq \varepsilon] \leq \mathbb{P}[u(x) - \mathbb{E}u(x) \geq \varepsilon] \leq \exp(-\frac{s\varepsilon^2}{2}).
\] (23)

The result follows by setting $\eta$ to the right hand side and solving $\varepsilon$. \qed

Now we are ready to prove Theorem 3.

Proof of Theorem 3. Let $\phi_0 \in \mathcal{F}_p$, such that $\mathcal{R}[\phi_0] < \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi] + \frac{\varepsilon}{4}$. By Lemma 10, with probability at least $1 - \frac{\eta}{3}$,

\[
\| \frac{1}{s} \sum_{j=1}^{s} f(x_i + \delta_{ijk}; \theta_k) - g(x_i; \theta_k) \|_2 \leq \frac{1 + \sqrt{2\log \frac{3Kn}{\eta}}}{\sqrt{s}}, 1 \leq i \leq n, 1 \leq k \leq K.
\] (24)
hold simultaneously. So with probability at least $1 - \frac{\eta}{3}$, for every $\phi = \sum_{k=1}^{K} w_k g(x; \theta_k) \in \mathcal{F}_\theta$, it holds that

\[
|\mathcal{R}_{\text{emp}}[\phi] - \mathcal{R}_{\text{se}}[\phi]| \leq \frac{L}{n} \left( \sum_{i=1}^{n} \left( \sum_{k=1}^{K} w_k \left[ \frac{1}{s} \sum_{j=1}^{s} f(x_i + \delta_{ijk}; \theta_k) \right] - l(\sum_{k=1}^{K} w_k g(x_i; \theta_k), y_i) \right) \right) \\
\leq \frac{L}{n} \sum_{i=1}^{n} \left( \sum_{k=1}^{K} w_k \left[ \frac{1}{s} \sum_{j=1}^{s} f(x_i + \delta_{ijk}; \theta_k) - g(x; \theta_k) \right] \right) \\
= \frac{L(1 + \sqrt{2 \log \frac{3Kn}{n}})}{\sqrt{s}} \triangleq \varepsilon_1.
\]

By Lemma 9 with probability at least $1 - \frac{\eta}{3}$, for every $\phi \in \mathcal{F}_\theta$, it holds that

\[
|\mathcal{R}_{\text{se}}[\phi] - \mathcal{R}[\phi]| \leq \frac{2\beta LMK}{\sqrt{n}} + \sqrt{\frac{8 \log \frac{2r}{n}}{n}} \triangleq \varepsilon_2.
\]

Let $\phi^*$ be the minimizer of $\mathcal{R}$ over $\mathcal{F}_\theta$. By Lemma 3 with probability at least $1 - \frac{\eta}{3}$, for $K \geq M\|\phi_0\|_p^2(1 + \sqrt{2 \log \frac{3}{n}})^2$,

\[
\mathcal{R}[\phi^*] - \mathcal{R}[\phi_0] < 2L\sqrt{\|\phi\|_p^4} \sqrt{\frac{M}{K}} (1 + \sqrt{2 \log \frac{3}{n}})^2 \triangleq \varepsilon_3.
\]

So with probability at least $1 - \eta$, it holds that

\[
\mathcal{R}[\hat{\phi}] - \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi] = (\mathcal{R}[\hat{\phi}] - \mathcal{R}_{\text{se}}[\hat{\phi}]) + (\mathcal{R}_{\text{se}}[\hat{\phi}] - \mathcal{R}_{\text{emp}}[\hat{\phi}]) + (\mathcal{R}_{\text{emp}}[\hat{\phi}] - \mathcal{R}_{\text{emp}}[\phi^*]) + (\mathcal{R}_{\text{emp}}[\phi^*] - \mathcal{R}_{\text{se}}[\phi^*]) + (\mathcal{R}_{\text{se}}[\phi^*] - \mathcal{R}[\phi^*]) + (\mathcal{R}[\phi^*] - \mathcal{R}[\phi_0]) + (\mathcal{R}[\phi_0] - \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi]) < \varepsilon_2 + \varepsilon_1 + 0 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \frac{\varepsilon}{4}
\]

\[
= 2\varepsilon_2 + 2\varepsilon_3 + \varepsilon + \frac{\varepsilon}{4}
\]

When $K > \frac{256L^4\|\phi_0\|_p^4 M(1 + \sqrt{2 \log \frac{3}{n}})^2}{\varepsilon^2}$, $n > \frac{64(2\beta LMK + \sqrt{8 \log \frac{12}{n}})^2}{\varepsilon^2}$, $s > \frac{64L^2(1 + \sqrt{2 \log \frac{3Kn}{n}})^2}{\varepsilon^2}$, we have

\[
\mathcal{R}[\hat{\phi}] - \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi] < \varepsilon. \quad (25)
\]

If $\mathcal{R}[\phi_0] = \inf_{\phi \in \mathcal{F}_p} \mathcal{R}[\phi]$, which means $\|\phi_0\|_p$ is independent of $\varepsilon$, $K = \Omega(\frac{1}{\varepsilon^2})$.

A.3 Proof of Proposition 2

Proof of Proposition 2 For $a > 0$, define $\zeta_a(\theta) = e^{-a\|\theta - \theta_0\|^2} p(\theta)$ and $u_a(\theta) = \frac{\zeta_a(\theta)}{\int_{\mathcal{F}_p} \zeta_a(\theta) d\theta}$. Then $\phi_a(x) = \int_{\mathcal{F}_p} u_a(\theta) g(x; \theta) d\theta \in \mathcal{F}_p$. For any $\varepsilon > 0$, there exists $\delta_1, \delta_2 > 0$, s.t. $\|g(x; \theta_0) - g(x; \theta)\|_2 \leq \frac{\varepsilon}{2}$, $\forall \|\theta - \theta_0\|_2 \leq \delta_1$ and $p(\theta) > \frac{p(\theta_0)}{2}$, $\forall \|\theta - \theta_0\|_2 \leq \delta_2$. Let $\delta = \min(\delta_1, \delta_2)$, we can...
choose a sufficiently large so that \( \int_{B(\theta_0, \delta)} w_a(\theta)d\theta \geq 1 - \frac{\varepsilon^2}{8} \). Then for \( \forall x \in \mathbb{R}^d \), we have

\[
\|g(x; \theta_0) - \phi(x)\|_2^2 = \| \int_{\Theta} w_a(\theta)[g(x; \theta_0) - g(x; \theta)]d\theta \|_2^2 \\
\leq \int_{\Theta} w_a(\theta)\|g(x; \theta_0) - g(x; \theta)\|_2^2 d\theta \\
= \left( \int_{B(\theta_0, \delta)} + \int_{\Theta \setminus B(\theta_0, \delta)} \right) w_a(\theta)\|g(x; \theta_0) - g(x; \theta)\|_2^2 d\theta \\
\leq \int_{B(\theta_0, \delta)} \frac{\varepsilon^2 w_a(\theta)}{4} d\theta + \int_{\Theta \setminus B(\theta_0, \delta)} 4w_a(\theta)d\theta \\
\leq \frac{\varepsilon^2}{4} + \frac{\varepsilon^2}{2} < \varepsilon^2.
\]

\[\blacksquare\]

\section*{B Supplementary material for experiments}

\subsection*{B.1 Detailed settings and hyper-parameters}

We perform all experiments on a single GeForce GTX 1080 Ti GPU.

For training the SWEEN models, we divide the training set into two parts, one for training base models, and the other for solving weights. We employ 2,000 images for solving weights on CIFAR-10 and 3000 images for solving weights on SVHN. Models not for ensembling are trained on the full training set.

For Gaussian data augmentation training, all the models are trained for 400 epochs using SGD. The learning rate is initialized set as 0.01, and decayed by 0.1 at the 150th/300th epoch.

For MACER training, we use the same hyper-parameters as Zhai et al. \cite{56}, i.e., we use \( k = 16, \beta = 16.0, \gamma = 8.0 \), and we use \( \lambda = 12.0 \) for \( \sigma = 0.25 \) and \( \lambda = 4.0 \) for \( \sigma = 0.50 \). We train the models for 440 epochs, the learning rate is initialized set as 0.01, and decayed by 0.1 at the 200th/400th epoch.

\subsection*{B.2 MACER training on CIFAR-10}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image.png}
\caption{Radius-accuracy curves for ensembles with base classifiers trained by MACER on CIFAR-10. (Left) \( \sigma = 0.25 \). (Right) \( \sigma = 0.50 \).}
\end{figure}

\subsection*{B.3 Results on SVHN}
Table 5: Certified accuracy (%) and ACR on SVHN. All models are trained via standard training. * means the upper envelope of base models.

| σ   | Model     | 0.00 | 0.25 | 0.5  | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2    | ACR |
|-----|-----------|------|------|------|------|------|------|------|------|------|-----|
| 0.25| 7-model*  | 90.3 | 74.6 | 53.4 | 29.5 | 0    | 0    | 0    | 0    | 0    | 0.517 |
|     | 7-model   | 91.0 | 76.4 | 56.8 | 34.7 | 0    | 0    | 0    | 0    | 0    | 0.547 |
| 0.50| 7-model*  | 72.7 | 58.1 | 42.1 | 28.4 | 17.7 | 11.0 | 6.5  | 3.1  | 0    | 0.498 |
|     | 7-model   | 72.8 | 59.2 | 43.9 | 29.4 | 19.8 | 11.9 | 7.3  | 3.7  | 0    | 0.524 |