Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

CEAR: Certified Ensemble Adversarial Robustness in DNNs

Daniel Sadig*†, Mohammadreza Maleki†, Hamed Karimi†, Reza Samavi†‡
† Department of Electrical, Computer, and Biomedical Engineering,
Toronto Metropolitan University, Toronto, ON, Canada
‡ Vector Institute, Toronto, ON, Canada

Abstract

Deep Neural Networks (DNNs) are highly susceptible to adversarial perturbations, leading to extensive research on robustness for safety-critical applications. State-of-the-art empirical defense mechanisms improve the robustness of DNNs through the training phase, but still struggle against adaptive white-box attacks. On the other hand, certified defenses offer provable guarantees of robustness within a specified perturbation bound. These guarantees hold regardless of the level of perturbations, even if the attacker is given full knowledge of the model. In this paper, we propose *CEAR*, an ensemble-based robust method that utilizes a hybrid of empirical and certified defense mechanisms. CEAR trains each network within the ensemble using varying Gaussian noise and temperatures to obfuscate gradients and logits, making the model more resistant to stronger gradient-based attacks. We then use noisy logits and propose two different voting mechanisms to further improve robustness. Furthermore, we extend randomized smoothing to verify the robustness of ensemble-based classifiers. Our experimental evaluations on MNIST, CIFAR10, and TinyImageNet datasets demonstrate superior certified accuracy on average, increased robustness radius, and decreased transferability compared to baseline methods.

Keywords: Deep Neural Networks, Adversarial Robustness, Certified Defenses, Ensemble Methods, Randomized Smoothing, Gradient-Based Attacks

1. Introduction

Modern Deep Neural Networks (DNNs) achieve high clean accuracy on independently and identically distributed (i.i.d.) test sets [1], but remain highly vulnerable to adversarially crafted perturbations, small and often imperceptible input modifications that can drastically alter the model predictions. The threat of these adversaries has hindered the deployment of DNNs in safety-critical tasks [2, 3]. To counter these threats, researchers have proposed two broad categories of defense mechanisms. *Empirical defenses*, such as adversarial training [4] and defensive distillation [5], which modify the training process, often by incorporating adversarial examples or altering the training process to improve robustness. Nevertheless, these defense techniques do not offer a formal robustness guarantee and have been shown to fail under strong or adaptive attacks. On the other hand, *certified defenses* provide formal, provable guarantees that classifiers prediction remains invariant within a specific perturbation boundary [6]. Certified defense approaches such as convex relaxation methods [7] and randomized smoothing [8–10] all provide formal guarantees of robustness. Randomized smoothing verifies that the prediction of a smoothed classifier remains consistent within specified perturbation radii by aggregating predictions over perturbed inputs.

Despite advances in certified defense methodologies, enhancing the robustness of individual classifiers at larger perturbation radii remains a fundamental challenge. Recent investigations have demonstrated that Ensemble-based approaches have shown improved robustness through the aggregation of diverse model predictions [11]. However, they still exhibit a steep decline in certified accuracy under larger perturbation budgets.

In this paper, we propose CEAR, *Certified Ensemble Adversarial Robustness*, a defense method that improves the robustness of DNNs via a certified ensemble approach. CEAR

* daniel.sadig@torontomu.ca

is inspired by three existing defense approaches: Gaussian augmentation [8, 9], distillation with temperature scaling [5], and noisy logits [12]. Although each of these three approaches has individually demonstrated effectiveness in improving the robustness of DNNs, their combined impact remains unexplored.

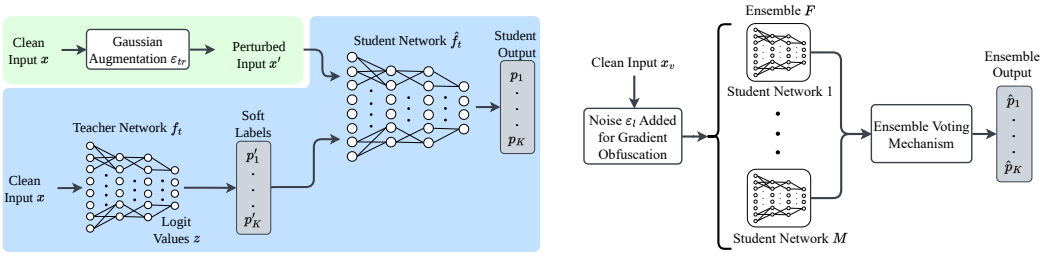
To leverage randomized smoothing, we train an ensemble of networks employing *Variable Gaussian Augmentation* (VGA), which augments each network with Gaussian noise sampled from distinct standard deviations. VGA enhances robustness while mitigating adversarial transferability, the phenomenon whereby adversarial examples crafted for one network successfully compromise other ensemble members [13]. Furthermore, we incorporate *Distillation with Temperature Scaling* (DTS) to smooth decision boundaries, thereby diminishing the effectiveness of gradient-based attacks. At inference time, we apply Gaussian noise to the input data to produce *noisy logits*, making it more difficult for stronger adversarial attacks (such as C&W [3]) to recover the original logits. Predictions across the ensemble are then aggregated via *Geometric Median* (GM) and *Robust Weighted Ensemble* (RW) voting mechanism, which unifies decision boundaries and improves robustness for varying confidence regimes. Finally, we verify the robustness of an ensemble by extending randomized smoothing to compute the robustness radius for each input in the dataset.

Related Work. Empirical defenses, such as adversarial training [1] and defensive distillation [5], attempt to smooth decision boundaries and reduce gradient sensitivity but remain vulnerable to strong gradient-based white-box attacks, such as C&W [3] and AutoAttack [14]. To protect against these attacks, we use noisy logits to further obfuscate the gradient [12]. However, empirical defenses that rely on gradient obfuscating techniques lead to a false sense of security [15]. Therefore, we incorporate certified defenses such as randomized smoothing [9], which have advanced robustness guarantees by optimizing sampling efficiency [16] and tightening robustness bounds [17, 18]. To further improve robustness, researchers proposed training an ensemble of independently smoothed classifiers paired with adaptive ensemble voting [11, 19, 20]. Most existing ensemble methods assume a uniform noise distribution across networks, limiting the network's diversity and increasing transferability [13]. To overcome this limitation, we trained each network in the ensemble with a distinct noise parameter, resulting in diverse gradient distributions and preventing the success of transfer-based attacks. Furthermore, classical weighted ensemble techniques [21, 22] have shown limited improvements in certified accuracy for larger perturbation budgets, which we mitigate by our proposed voting mechanisms. These improvements decrease transferability and increase robustness of the ensemble-based classifier (Section 4).

Contributions. We are making the following contributions: (1) We propose CEAR, a certified adversarial defense framework that unifies DTS and VGA in an ensemble training pipeline to promote decision-boundary smoothness and diversity. At inference, CEAR leverages noisy logits to hinder gradient-based attacks and applies two voting mechanisms to recover clean accuracy while improving robustness. (2) We extend the randomized smoothing verification method to verify the robustness of ensemble-based classifiers. (3) Our experimental evaluations demonstrate that CEAR achieves a higher certified accuracy on average at larger radii and is more resilient to gradient-based white-box attacks than existing state-of-the-art baselines.

2. Certified Ensemble Adversarial Robustness

An overview of CEAR method is presented in Figure 1. In the robust training phase (shown in Figure 1(a) and described in Section 2.1), we adapt Gaussian augmentation, which was initially designed for a single network (green-shaded area in Figure 1(a)), and expand this process across an ensemble of networks. In this phase, we also adapt distillation with temperature scaling (blue-shaded area in Figure 1(a)) to train a teacher network and



(a) Training: Use clean input x to train teacher network to generate soft labels; train student network using the sift labels plus perturbed input x'.

(b) Inference: Add asynchronous noise to ensemble for gradient obfuscation and generate output using a voting mechanism.

Figure 1. Overview of Certified Ensemble Adversarial Robustness

use the output of the network in conjunction with Gaussian augmented inputs to train several student networks. In the inference phase (shown in Figure 1(b) and described in Section 2.2), we adapt noisy logits [12] to obfuscate the gradients across all layers of the individual student networks in the ensemble and protect the model against stronger attacks such as C&W attack [3]. Since clean inputs are exposed to several stochastic perturbations (due to the addition of Gaussian noise during training and inference phases), degradation of clean accuracy is unavoidable [4]. Thus, we practically show that using the ensemble, along with our proposed voting mechanism, can effectively compensate for the clean accuracy deficiency.

2.1. **Robust Training**

Let $f : \mathcal{X} \to \mathcal{Y}$ denote a DNN, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the input space with dimension d, and $\mathcal{Y} = \{1, 2, \ldots, K\}$ is the output space consisting of K class labels. Given a clean input $x \in \mathcal{X}$, an adversarial example x' is defined as $f(x') \neq f(x)$ subject to $\|x' - x\| \leq \varepsilon$ where ε denotes the perturbation bound under a specified norm (e.g., ℓ_2 or ℓ_∞).

Variable Gaussian Augmentation (VGA). To improve the robustness of a single network, Gaussian noise is added to the input, $x' = x + \varepsilon_{\text{tr}}$, where $\varepsilon_{\text{tr}} \sim \mathcal{N}(0, \sigma_{\text{tr}}^2 I)$ and σ_{tr} is the standard deviation used during training (green-shaded area in Figure 1(a)) [9]. By training on noisy inputs, the network learns smoother decision boundaries and becomes provably less sensitive to small adversarial perturbations. In an ensemble, each network receives distinct Gaussian perturbations that are drawn from the same fixed standard deviation σ_{tr}. However, when all networks share the same noise level, transferability remains high due to overlapping decision boundaries. To address this vulnerability, we propose VGA, in which each ensemble network is assigned a different noise level by perturbing a base standard deviation σ_{tr}. Specifically, we sample a random variable β from a uniform distribution as,

$$\beta \sim \text{Uniform}\big(-\tfrac{1}{2}\sigma_{\text{tr}} \ , \ +\tfrac{1}{2}\sigma_{\text{tr}}\big) \, . \tag{2.1}$$

The effective standard deviation becomes $\sigma_{\text{tr}} + \beta$, and the training-time Gaussian noise is:

$$\varepsilon_{\text{tr}} \sim \mathcal{N}\big(0, \ (\sigma_{\text{tr}} + \beta)^2 I\big) \, . \tag{2.2}$$

By drawing perturbations from the distribution described in Equation 2.2, we generate diverse noise distributions in the ensemble and increase the diversity of the ensemble, resulting in decreasing adversarial transferability across all networks [23]. We also provide empirical evidence in Section 4 that drawing variable Gaussian noise per input improves certified accuracy. This improvement is especially notable under large perturbation magnitudes.

Distillation with Temperature Scaling (DTS). As shown in the blue-shaded region of Figure 1(a), we first train a teacher network f_t to generate the predictive probability vector

$\mathbb{P}\big(f_t(x)\big)$ for each input $x \in \mathcal{X}$ as,

$$\mathbb{P}\big(f_t(x) = c\big) = \text{softmax}\left(\frac{z_c}{t}\right) , \tag{2.3}$$

where z_c denotes the logit value associated with the class $c \in \mathcal{Y}$, and $t \in \mathbb{R}^{[1,+\infty)}$ is a temperature parameter that smoothens the softmax distribution, resulting in softened class probabilities known as *soft labels* [24]. Then, the soft labels are used as ground truth to train a student network on the Gaussian augmented input x' via the loss function $\mathcal{L}$ as,

$$\mathcal{L}(x') = \text{CE}\Big(\mathbb{P}\big(\hat{f}(x')\big), \mathbb{P}\big(f_t(x)\big)\Big) , \tag{2.4}$$

where $\text{CE}(\cdot)$ is cross-entropy function and $\hat{f}(x')$ denotes the logit vector associated with the student network outputs on the perturbed input x'. The advantage of using soft labels as training labels, comes from the valuable information they contain compared to the correct class. Instead of solely indicating the correct class, soft probability distributions capture the relative similarities between classes. Furthermore, utilizing DTS effectively smoothens the decision boundaries, making the transitions between different class regions in the input space more gradual. Therefore, DTS improves generalization and robustness of DNNs against unforeseen adversarial examples [5, 25].

In temperature-scaled distillation, we adjust the temperature within a modest range (typically 1 to 5) to foster diversity across the ensemble while preserving informative gradients. Moderate temperatures allow the student to learn inter-class information without flattening the distribution. According to Equation 2.3, if t is too low ($t \approx 1$), $\mathbb{P}\big(f_t(x)\big)$ approaches the true label, losing significance of class similarities. If t is too high, the distribution becomes nearly uniform (i.e., $\mathbb{P}\big(f_t(x)\big) \approx 1/K$), thus flattening away informative gradients conveyed.

2.2. **Robust Inference**

Noisy Logits. The C&W attack [3] revealed that defensive distillation can be circumvented when adversaries have iterative access to model logits. Building on this insight, CEAR adopts input-level randomization at inference time by injecting Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ into the input rather than perturbing the logits directly [12]. While logit-level noise can be averaged out through repeated querying, input-level noise induces inherently stochastic logits without modifying model parameters, making adversarial reconstruction substantially harder. Thus, in CEAR the prediction $\hat{y}_x$ is obtained by applying the network layers to the perturbed input x' as $\hat{y}_x = f_i \circ f_{i+1} \circ \cdots \circ f_l(x')$, where f_i denotes the intermediary logits and $\circ$ denotes the composition of the logits. This stochastic forward process enhances robustness to adaptive attacks, albeit at the cost of reduced predictive confidence.

Geometric Median Ensemble (GM). The geometric median is a classical robust estimator of multivariate location. Unlike the arithmetic mean, it is insensitive to a minority of extreme inputs, ensuring that a small number of corrupted or adversarially perturbed predictions cannot dominate the aggregate. In the ensemble setting, this implies that the aggregate prediction remains close to the majority consensus even when several student networks produce distorted outputs.

Let $p_i(x) = \mathbb{P}\big(\hat{f}_i(x + \varepsilon_l)\big) \in \Delta^K$ denote the K-class probability vector produced by the i-th network for input x. We aggregate the ensemble predictions using the geometric median of the networks' softmax outputs defined as,

$$\hat{q}(x) \;=\; \arg\min_{q \in \Delta^K} \sum_{i=1}^{M} \big\| q - p_i(x) \big\|_2 , \tag{2.5}$$

where the optimization is constrained to the probability simplex Δ^K to ensure a valid predictive distribution. This estimator corresponds to the GM, a classical notion of multivariate

location [26]. This optimization finds the point $\hat{q}(x)$ whose total Euclidean distance to all member probabilities $\{p_i(x)\}$ is minimal. The final class decision is $c_A = \arg\max_c \hat{q}_c(x)$. In practice, the geometric median can be efficiently computed using iterative procedures such as Weiszfeld's algorithm [27]. Importantly, aggregation is performed at the level of softmax probabilities rather than logits or labels, preserving the probabilistic interpretation of each model's output.

Robust Weighted Ensemble (RW). To improve robustness at larger perturbation budgets while compensating for accuracy degradation, we propose RW that accounts for the relative contribution of each student network by assigning greater influence to networks with higher certified accuracy. This approach effectively unifies the decision regions induced by the individual ensemble members. Unlike the distinct training set $\mathcal{X}_i$ used for each student network $\hat{f}_i$ in the ensemble, a shared validation set $\mathcal{X}_v$ is used for all networks where $\mathcal{X}_i \cap \mathcal{X}_v = \emptyset$ for all $i \in \mathbb{N}^{[1,M]}$. In the ensemble $F = \{\hat{f}_i\}_{i=1}^{M}$ of M student networks, we first perturb the input $x_v \in \mathcal{X}_v$ with the Gaussian noise of a fixed standard deviation, resulting in perturbed inputs. Then, we pass the perturbed validation data to each individual student network $\hat{f}_i$ separately to compute its corresponding certified accuracy a_i. Finally, we compute the ensemble's predictive probability vector $\hat{p}$ using the weighted average of the individual network probability vector p_i as,

$$\hat{p}(x) = \sum_{i=1}^{M} \lambda_i p_i(x) \qquad \text{such that} \qquad \lambda_i = \frac{a_i}{\sum_{j=1}^{M} a_j}\ , \tag{2.6}$$

where $\lambda_i \in \mathbb{R}^{[0,1]}$ denotes the *network contribution factor* proportional to a_i of each $\hat{f}_i$. Therefore, networks with higher certified accuracy within the given perturbation level have a stronger vote in the final predictive probability $\hat{p}$ in Equation 2.6. By aggregating the networks' predictions in the ensemble with our robust weighted ensemble, the robustness of DNNs is improved.

Although both aggregation schemes improve robustness, they are effective in different confidence regions as validated in Section 4. When individual student networks produce confident and consistent predictions, we employ GM aggregation, which sharpens the ensemble decision and yields higher certified accuracy at small radii. In contrast, as the perturbation budget increases and the model confidence degrades, we adopt RW, which explicitly down-weights less reliable networks based on their certified accuracy.

Computational Overhead. As an ensemble-based defense, CEAR increases runtime and memory by requiring M student networks, but this deliberate overhead is often acceptable in safety-critical deployments where improved robustness and verifiable guarantees outweigh additional compute. When analyzing CEAR's computational complexity, we suppress constant per-sample costs (e.g., those for forward/backward passes or logit perturbation in a single DNN) since they remain uniform across method components, and asymptotic analysis inherently ignores such implementation-level constants. Instead, we focus on how runtime grows with the main input variables in CEAR: the ensemble size M, training dataset size $|\mathcal{X}|$, validation dataset size $|\mathcal{X}_v|$, and test sample size $|\mathcal{X}_{\text{test}}|$.

During training, CEAR independently trains each of the M networks in the ensemble using DTS (teacher and student forward-backward passes as described in Section 2.1), and VGA is applied on-the-fly. As VGA involves generating input-dependent noise ε_{tr} per sample via Gaussian sampling, this noise augmentation introduces a negligible constant per-sample overhead, i.e., $\mathcal{O}(1)$; thus, the total time to train all networks on dataset $\mathcal{X}$ scales as $\mathcal{O}\big(M.|\mathcal{X}|\big)$. This reflects linear growth in both ensemble and dataset size.

After training, each network's contribution factor is computed once on a held-out validation set $\mathcal{X}_v$ based on its certified accuracy (Equation 2.6), incurring a one-time cost of

Algorithm 1 Ensemble-based Robustness Radius Measurement

Require: Clean input x; ensemble $F = \{\hat{f}_i\}_{i=1}^{M}$ of student networks; λ_i: The contribution factor for the i^{th} student network; N_x: The number of perturbations of x for ensemble prediction; N: Number of perturbations of x for Monte-Carlo sampling; σ_v: Verification standard deviation; α: Abstinence threshold;

Ensure: Predicted class c_A with certified radius R, or abstain (-1)

$c_0 \leftarrow [\,]$ ▷ Empty list of predictions
for $j \leftarrow 1$ **to** N_x **do**
  $\varepsilon_v \sim \mathcal{N}(0, \sigma_v^2 I)$
  $s_c \leftarrow \sum_{i=1}^{M} \lambda_i \, \mathbb{P}\Big(\hat{f}_i(x + \varepsilon_v) = c\Big) \quad \forall c \in \mathcal{Y}$
  $c \leftarrow \arg\max_{c \in \mathcal{Y}} \; s_c$
  $\textsc{Append}(c_0, c)$ ▷ Ensemble prediction (Eq. 3.2)
end for
$c_A \leftarrow \textsc{Mode}(c_0)$ ▷ Most frequent prediction
$P_A \leftarrow \textsc{SampleWithNoise}(F, x, c_A, N, \sigma_v)$
$\underline{P_A} \leftarrow \textsc{LowerConfBound}(P_A \cdot N, N, 1 - \alpha)$
if $\underline{P_A} > \frac{1}{2}$ **then**
  return (c_A, R) ▷ Return class and radius (Eq. 3.6)
end if
return -1 ▷ Abstain

$\mathcal{O}(M.|\mathcal{X}_v|)$. These weights are then fixed and reused during inference. For each test sample, CEAR performs noisy forward passes through the M pretrained students and applies RW voting via a single linear scan over the ensemble, adding only $\mathcal{O}(M)$ overhead. Thus, per-sample inference remains $\mathcal{O}(M)$ and scales as $\mathcal{O}(M.|\mathcal{X}_{\text{test}}|)$ over the full test set. For CEAR with geometric median aggregation, the same M noisy forward passes are used, but logits are combined via element-wise log-probability averaging across K classes before prediction, incurring an additional $\mathcal{O}(M.K)$ aggregation cost. Nevertheless, the overall inference complexity remains linear in M, preserving CEAR's computational efficiency.

3. CEAR Robustness Verification

Prior to deployment, robust verification is essential to guarantee that a model's decisions are invariant to perturbations within a given budget. We adapt randomized smoothing to assess the robustness of an ensemble model. Specifically, we use the *robustness radius* as the maximum perturbation under a given ℓ_p-norm that the model can tolerate without changing its prediction for a given input. A larger radius means that the model's decision remains invariant under larger noise perturbations. We establish the robustness radius by first defining a smoothed classifier $g : \mathcal{X} \to \mathcal{Y}$, formed by adding the Gaussian noise ε_v to the inputs of base classifier $f : \mathcal{X} \to \mathcal{Y}$ to generate a set of perturbed instances of each validation input $x \in \mathcal{X}_v$. For N_x perturbed inputs $x + \varepsilon_v$, the smoothed classifier g returns the class with the highest probability that is obtained by:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \; \mathbb{P}\big(f(x + \varepsilon_v) = c\big) \;, \tag{3.1}$$

where $c \in \mathcal{Y}$ denotes the class labels. We then extend Equation 3.1 to smooth ensemble-based classifiers under l_2-norm. Let $\varepsilon_v \sim \mathcal{N}(0, \sigma_v^2 I)$ be the Gaussian noise and $F : \mathcal{X} \to \mathcal{Y}$

be a function of an ensemble that contains a set of M base networks $\{f_i\}_{i=1}^{M}$ as,

$$F(x) = \arg\max_{c \in \mathcal{Y}} \sum_{i=1}^{M} \lambda_i \cdot \mathbb{P}\big(f_i(x) = c\big) \ , \tag{3.2}$$

where λ_i denotes the contribution factor associated with each network f_i in the ensemble. We apply randomized smoothing to each individual network f_i, resulting in a set of smoothed classifiers constituting the smoothed ensemble G on the set of N_x perturbed inputs as,

$$G(x) = \arg\max_{c \in \mathcal{Y}} \ \mathbb{P}\big(F(x + \varepsilon_v) = c\big) \ . \tag{3.3}$$

Since it is infeasible to *exactly* compute the smoothed prediction $G(x)$ and verify its robustness, we employ Monte Carlo sampling algorithm to ensure reliable estimates of the predicted class and robustness radius with high probability [9]. We compute the robustness radius for the smoothed ensemble G as described in the following corollary.

Corollary 1. *Given the most probable class $c_A \in \mathcal{Y}$ with the lower bound probability of $\underline{P_A}$ and the upper bound probability of the runner-up class $\overline{P_B}$ such that $\underline{P_A}$, $\overline{P_B} \in \mathbb{R}^{[0,1]}$, the following conditions are satisfied:*

$$\mathbb{P}\big(F(x + \varepsilon_v) = c_A\big) \ \geq \ \underline{P_A} \ \geq \ \overline{P_B} \geq \ \max_{c \neq c_A} \mathbb{P}\big(F(x + \varepsilon_v) = c\big) \tag{3.4}$$

such that $\overline{P_B} = 1 - \underline{P_A}$. Then, the ensemble F is robust at x within the radius R if

$$G(x + \delta) = c_A \qquad \forall \ \|\delta\|_2 \leq R \ , \tag{3.5}$$

where

$$R = \frac{\sigma_v}{2}\Big(\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})\Big) \ , \tag{3.6}$$

in which Φ^{-1} denotes the inverse standard Gaussian CDF.

The proof of Corollary 1 follows that of [9]; however, the base classifier is an ensemble rather than a single network. Corollary 1 shows that the robustness radius R increases with larger verification noise σ_v and higher confidence in the top class c_A, and diverges as $\underline{P_A} \to 1$. Since the Gaussian distribution has full support on $\mathcal{X}_v$, the condition $\mathbb{P}(F(x+\varepsilon_v) = c_A) = 1$ implies that $F(\cdot) = c_A$ almost everywhere in the input space.

Algorithm 1 shows how to measure the robustness radius for a given input $x \in \mathcal{X}_v$. We first apply randomized smoothing to each base classifier f in the ensemble F by convolving f with ε_v. We generate a set of N_x perturbed instances of x by adding ε_v (Line 3) and collecting all the votes of the smoothed ensemble G in c_0 (Line 4). The ensemble prediction c_A is then obtained by selecting the class prediction that the ensemble is most likely to predict under the given noise ε_v (Line 7). We then compute the probability P_A that the ensemble predicts the class c_A across N different perturbed instances of x drawn from the same noise level ε_v, using the *SampleWithNoise* procedure (Line 8). Next, we compute a lower confidence bound on P_A using the *LowerConfBound* procedure, which estimates the minimum probability that the ensemble predicts the class c_A under noise level ε_v (Line 9). Here, $1-\alpha$ denotes the confidence level, e.g., a 95% confidence corresponds to $\alpha = 0.05$. The model is certifiably robust at x if this lower bound $\underline{P_A}$ exceeds the threshold 0.5 (Line 10). If the bound is greater than 0.5, we return c_A along with a certified robustness radius R, within which the prediction remains provably unchanged (Line 11). Otherwise, the ensemble model abstains, indicating insufficient confidence for certification (Line 13).

4. Experimental Evaluation

We evaluated the robustness of CEAR under two voting mechanisms: GM voting denoted by CEAR(GM), and RW voting denoted by CEAR(RW), through three different

experiments: (1) examine the certified accuracy for varying radii, (2) compute the robustness radius, and (3) determine which networks are less susceptible to AutoAttack under ℓ_2 and ℓ_∞ norms. We further extended our evaluations by examining the ablated variant of CEAR without the VGA, denoted as CEAR$^-$.

Datasets and Baselines. We evaluated CEAR against state-of-the-art baselines on MNIST [28], CIFAR10 [29] and TinyImageNet [30] datasets. These datasets provide a diverse evaluation setting, ranging from low-dimensional grayscale digits to high-resolution natural images with a large number of classes. The baselines are denoted by *RandSmooth* for the randomized smoothing method using a single network [9], and *SWEEN* for the smooth weighted ensemble method [20].

Threat Models. We evaluated robustness under white-box attacks, where the adversaries have full access to each network's architecture, weights, and gradients. We used a subset of *AutoAttack* [14] which is a parameter-free and reliable evaluation framework comprised of three attacks, APGD-CE, APGD-DLR, and FAB. These attacks effectively probe decision boundaries and detect gradient masking. Making it an effective tool to evaluate adversarial transferability across ensemble configurations.

Implementation Details. We implemented the proposed method[1] using the TensorFlow framework [31] in Python and conducted the experiments using a Tesla T4 GPU. For consistent comparative evaluations on MNIST and CIFAR10 with baseline methods, we used three ensembles, each consisting of five student networks individually trained with soft labels produced by teacher networks at temperatures $t = \{2, 3, 4, 5, 6\}$. The student networks were trained and tested under varying Gaussian noise levels $\{0.25,\ 0.5,\ 1.0\}$ for CIFAR10 and MNIST, and $\{0.125,\ 0.25\}$ for TinyImageNet. The use of lower-noise variants on TinyImagNet is motivated by the dataset's lower test accuracy, which makes the classification task significantly more challenging and renders the model highly sensitive to larger noise perturbations. For MNIST, we used LeNet5 [32] to train both the teacher and student networks using SGD Optimizer and Gaussian noise $\varepsilon_l = 0.3$. For CIFAR10 and TinyImageNet, we used the same hyperparameters with Gaussian noise $\varepsilon_l = 0.03$ on ResNet110 [33] and ResNet18 models, respectively. Furthermore, on 10,000 CIFAR10 instances, RandSmooth takes 33(s), SWEEN takes 164(s),CEAR(RW) takes 162(s), and CEAR(GM) takes 171(s), while on MNIST, RandSmooth takes 2(s), SWEEN takes 10(s), CEAR(RW) takes 7(s), and CEAR(GM) takes 14(s).

Experiment 1 (Certified Accuracy). We use certified accuracy (CA) [9] at given radius L as a metric to measure the proportion of instances that are correctly classified by the smoothed classifier and are provably robust to any adversarial perturbation as,

$$\mathrm{CA} = \frac{1}{|\mathcal{X}_v|} \sum_{j \in \mathcal{X}_v} \mathbb{I}\{R_j \geq L\}\ , \tag{4.1}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function and R_j is the robustness radius for each sample j that is correctly classified.

Tables 1 and 2 report CA as a function of the perturbation radius R and noise level σ_v for RandSmooth, SWEEN, and CEAR variants, providing empirical support for our main theoretical claim that the optimal aggregation strategy depends critically on model confidence and perturbation budget of the smoothed classifier. Across MNIST and CIFAR10, we observe that in high confidence regions (small R), CEAR$^-$(GM) is optimal; it consistently yields the highest robust accuracy, achieving 99% at $R = 0$ on MNIST ($\sigma_v = 0.25$) and 51% at $R = 0.5$ on CIFAR10 ($\sigma_v = 0.25$), outperforming both RandSmooth and SWEEN. However, as R increases and predictive uncertainty grows, this advantage vanishes, and fixed-weight aggregation (RW) with VGA becomes more reliable for low-confidence regimes:

[1]The source codes are available at https://github.com/tailabTMU/CEAR.

Table 1. Certified accuracy (%) at varying radii and σ_v on MNIST (M) and CIFAR10 (C).

Radius R		0.00		0.25		0.50		0.75		1.00		1.25		1.50		1.75		2.00	
σ_v	**Model**	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C
0.25	RandSmooth	97	76	95	59	93	37	87	22	0	0	0	0	0	0	0	0	0	0
	SWEEN	98	77	97	61	94	46	90	30	0	0	0	0	0	0	0	0	0	0
	CEAR$^-$(RW)	98	**79**	97	**65**	95	47	**93**	31	0	0	0	0	0	0	0	0	0	0
	CEAR$^-$(GM)	**99**	**79**	**98**	**65**	**96**	**51**	**93**	31	0	0	0	0	0	0	0	0	0	0
	CEAR(RW)	98	76	97	63	93	49	90	**36**	0	0	0	0	0	0	0	0	0	0
	CEAR(GM)	98	72	97	58	93	45	90	33	0	0	0	0	0	0	0	0	0	0
0.50	RandSmooth	96	64	95	55	92	38	86	28	77	15	66	9	39	5	18	2	0	0
	SWEEN	97	68	**97**	56	93	**44**	86	34	84	20	72	14	50	8	20	5	0	0
	CEAR$^-$(RW)	97	68	94	56	92	42	87	31	75	22	66	14	51	9	26	4	0	0
	CEAR$^-$(GM)	**98**	**70**	**97**	**58**	**94**	**44**	**89**	31	**85**	**25**	**72**	16	55	11	25	6	0	0
	CEAR(RW)	94	68	92	56	87	**44**	83	**36**	75	**23**	67	14	**58**	10	**43**	**8**	0	5
	CEAR(GM)	96	63	93	53	90	39	84	29	75	17	61	10	44	8	25	0	0	0
1.00	RandSmooth	87	42	80	31	71	23	59	15	42	11	28	7	15	6	9	3	5	1
	SWEEN	88	43	80	34	71	26	64	20	50	16	37	12	28	9	15	6	9	4
	CEAR$^-$(RW)	87	**47**	79	35	69	26	62	20	58	15	49	12	32	8	22	6	15	4
	CEAR$^-$(GM)	**91**	**47**	83	37	**76**	27	69	22	57	16	42	13	31	11	17	8	13	5
	CEAR(RW)	91	**47**	**87**	**37**	**76**	**30**	**69**	**25**	**60**	**20**	**50**	**18**	**38**	**15**	**34**	**10**	**20**	**8**
	CEAR(GM)	88	46	80	35	72	29	58	24	45	18	36	15	22	12	12	10	7	8

Table 2. Certified accuracy (%) at varying radii and σ_v on TinyImageNet.

σ_v	**Model**	**Radius (R)**								
		0.00	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.125	RandSmooth	38	28	19	13	8	0	0	0	0
	SWEEN	45	33	24	18	13	0	0	0	0
	CEAR$^-$(RW)	45	35	30	24	16	0	0	0	0
	CEAR$^-$(GM)	46	**36**	29	25	16	0	0	0	0
	CEAR(RW)	**48**	35	**30**	**26**	**20**	0	0	0	0
	CEAR(GM)	46	36	29	25	17	0	0	0	0
0.25	RandSmooth	28	25	18	13	7	9	4	3	3
	SWEEN	30	23	17	14	10	9	6	5	5
	CEAR$^-$(RW)	37	33	29	23	20	16	13	10	8
	CEAR$^-$(GM)	37	33	27	20	17	12	9	7	6
	CEAR(RW)	**42**	**35**	**31**	**24**	**20**	**17**	**13**	**10**	**9**
	CEAR(GM)	32	29	26	21	15	11	9	7	5

CEAR(RW) attains 58% at $R = 1.50$ on MNIST ($\sigma_v = 0.50$) and 20% at $R = 1.0$ on CIFAR10 ($\sigma_v = 1.00$), surpassing both baselines. This pattern is even stronger on TinyImageNet, where inherently lower model confidence renders adaptive weighting less effective and fixed RW consistently dominates (e.g., 26% at $R = 0.3$, $\sigma_v = 0.125$).

Experiment 2 (Radius of Robustness). Figure 2 illustrates the certified robustness radius achieved by each method under increasing noise levels σ_v. Across all three datasets, CEAR(RW) consistently achieves a larger robustness radius than baselines and other voting mechanisms, indicating improved robustness in lower-confidence regions under higher perturbations. In particular, Figures 2(d), 2(h), and 2(l) demonstrate that CEAR(RW) maintains a greater CA at $R > 1.0$ whereas RandSmooth and SWEEN rapidly deteriorate.

On MNIST in Figure 2(c), CEAR(RW) exhibits a smoother decline in CA as the radius increases. For instance, at $\sigma_v = 1.0$ and $R = 2.5$, both RandSmooth and SWEEN fall to near-zero CA, whereas CEAR(RW) continues to dampen the effects of the perturbations and maintain its accuracy. A similar trend is observed on CIFAR10 and TinyImageNet in

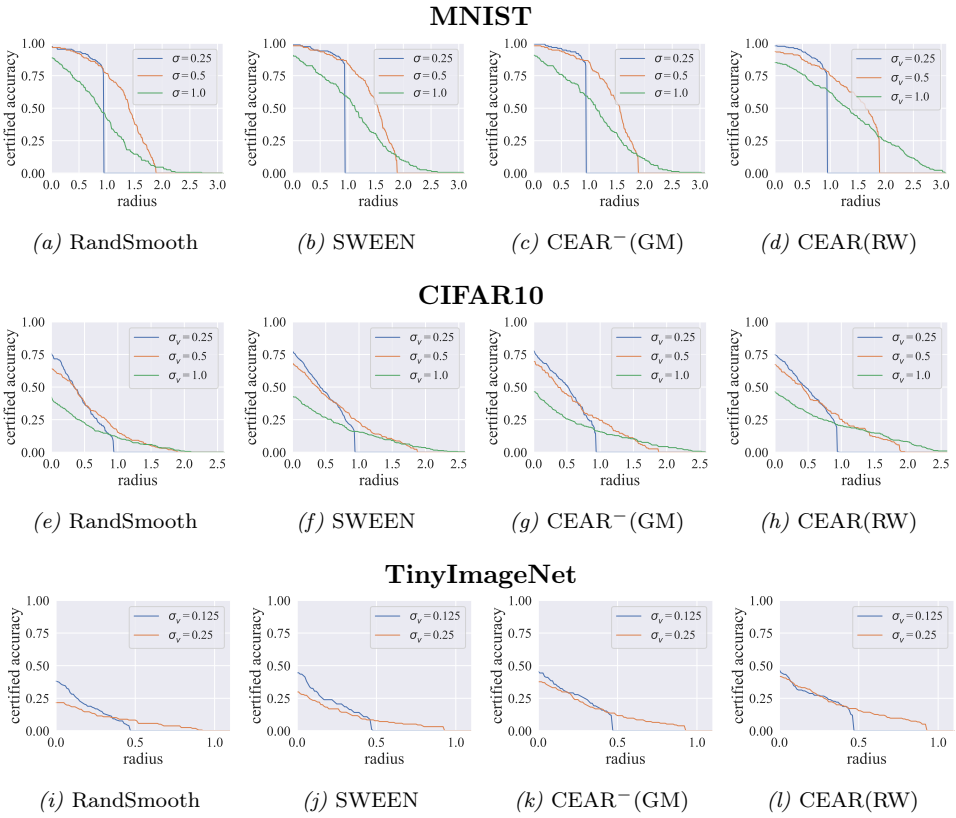


Figure 2. The average certified accuracy under varying radii and noise σ_v

Figures 2(h) and 2(l), confirming that robust weighted aggregation is particularly effective in the large-radius regime. In contrast, Figure 2 shows that CEAR$^-$(GM) achieves the highest CA at small radii but degrades more rapidly beyond $R > 1.0$. While this method still outperforms baselines at both lower and higher radii, it demonstrates that using a varied noise distribution in training in conjunction with an RW aggregation is better in low-confidence regions.

Experiment 3 (Mitigating Transferability Impact). We evaluated the transferability by generating attacks targeting one network at a time and measuring attacker success rate across the ensemble. Networks with low transferability achieve a lower attacker success rate. In contrast, to provide a non-transferable comparison, RandSmooth employs a single network where adversarial examples are generated and tested on the same network. As shown in Table 3, for instance, on CIFAR10 under the ℓ_∞ norm ($\varepsilon = 0.03$), CEAR(RW) substantially reduces the attacker success rate to 46%, outperforming RandSmooth (95.56%) and SWEEN (60.10%), demonstrating markedly improved robustness to transferable attacks.

5. Conclusion

CEAR is an ensemble-based robustness framework that integrates empirical and certified defense mechanisms across training and inference. During training, CEAR couples DTS and VGA to diversify ensemble members, smooth decision boundaries, and reduce adversarial transferability. At inference time, noisy logits are used to obfuscate gradients, while ensemble aggregation recovers clean accuracy and improves CA. We study two aggregation strategies: an optimized geometric median (GM) and a fixed robust weighted ensemble (RW). Experiments show that GM achieves higher certified accuracy in high-confidence,

Table 3. The attacker success rates under ℓ_2 and ℓ_∞ norms

Model	**MNIST**			**CIFAR10**		
	Clean Acc.	ℓ_2 ($\varepsilon = 1.00$)	ℓ_∞ ($\varepsilon = 0.20$)	Clean Acc.	ℓ_2 ($\varepsilon = 0.35$)	ℓ_∞ ($\varepsilon = 0.03$)
RandSmooth	99.12%	12.20%	99.82%	75.00%	33.18%	96.56%
SWEEN	99.16%	11.18%	56.28%	80.20%	45.29%	60.10%
CEAR$^-$(GM)	99.10%	**9.07%**	50.13%	84.6%	19.9%	48.31%
CEAR(RW)	98.29%	10.60%	**44.60%**	81.00%	**16.79%**	**46.00%**

low-perturbation regions, whereas RW coupled with varying noise is more reliable under larger perturbation budgets and lower-confidence settings, yielding larger certified radii and improved robustness compared to prior ensemble baselines. These results indicate that optimized aggregation is preferable when ensemble predictions are confident, while fixed robust weighting is superior in high-uncertainty adversarial regimes. For future work, we plan to systematically investigate how the choice of temperature in DTS influences ensemble diversity, calibration, and certified robustness across different perturbation regimes.

Acknowledgements

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund at Toronto Metropolitan University and Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery grants (#348100).

References

[1] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges. "Adversarial example detection for DNN models: A review and experimental comparison". In: *Artificial Intelligence Review* 55.6 (2022), pp. 4403–4462.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014*. 2014.

[3] N. Carlini and D. Wagner. "Towards evaluating the robustness of neural networks". In: *2017 ieee symposium on security and privacy (sp)*. Ieee. 2017, pp. 39–57.

[4] Y. Wang and L. Liu. "Failure Cases Are Better Learned But Boundary Says Sorry: Facilitating Smooth Perception Change for Accuracy-Robustness Trade-Off in Adversarial Training". In: *arXiv preprint arXiv:2508.02186* (2025).

[5] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *symp. on security and privacy*. IEEE. 2016, pp. 582–597.

[6] L. Li, T. Xie, and B. Li. "Sok: Certified robustness for deep neural networks". In: *2023 IEEE symposium on security and privacy (SP)*. IEEE. 2023, pp. 1289–1310.

[7] A. Raghunathan, J. Steinhardt, and P. S. Liang. "Semidefinite relaxations for certifying robustness to adversarial examples". In: *NeurIPS* 31 (2018).

[8] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. "Certified robustness to adversarial examples with differential privacy". In: *symp. on security and privacy (SP)*. IEEE. 2019, pp. 656–672.

[9] J. Cohen, E. Rosenfeld, and Z. Kolter. "Certified adversarial robustness via randomized smoothing". In: *international conference on machine learning*. PMLR. 2019, pp. 1310–1320.

[10] M. Zühlke and D. Kudenko. "Adversarial robustness of neural networks from the perspective of lipschitz calculus: A survey". In: *ACM Computing Surveys* 57.6 (2025), pp. 1–41.

[11] Z. Yang, L. Li, X. Xu, B. Kailkhura, T. Xie, and B. Li. "On the Certified Robustness for Ensemble Models and Beyond". In: *Int. Conf. on Learning Representations*. 2022.

[12] Y. Liang and R. Samavi. "Advanced defensive distillation with ensemble voting and noisy logits". In: *Applied Intelligence* 53.3 (2023), pp. 3069–3094.

[13] X. Chen, W. Huang, Z. Peng, W. Guo, and F. Zhang. "Diversity supporting robustness: Enhancing adversarial robustness via differentiated ensemble predictions". In: *Computers & Security* 142 (2024), p. 103861.

[14] F. Croce and M. Hein. "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". In: *ICML*. PMLR. 2020, pp. 2206–2216.

[15] A. Athalye, N. Carlini, and D. Wagner. "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 274–283.

[16] R. Zhai, C. Dan, D. He, H. Zhang, B. Gong, P. Ravikumar, C.-J. Hsieh, and L. Wang. "MACER: attack-free and scalable robust training via maximizing certified radius". In: *International Conference on Learning Representations (ICLR)*. 2020.

[17] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang. "Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers". In: *NeurIPS*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[18] B. Li, Y. Wang, Z. Ren, and J. Z. Kolter. "Double-boosted randomized smoothing: A sharp certified defense against adversarial attacks". In: *NeurIPS*. 2022.

[19] L. Huang, Q. Huang, P. Qiu, S. Wei, and C. Gao. "FASTEN: fast ensemble learning for improved adversarial robustness". In: *IEEE Transactions on Information Forensics and Security* 19 (2023), pp. 2565–2580.

[20] C. Liu, Y. Feng, R. Wang, and B. Dong. "Enhancing Certified Robustness via Smoothed Weighted Ensembling". In: *ICML 2021 Workshop on Adv. Machine Learning*. 2020.

[21] D. Jiménez. "Dynamically weighted ensemble neural networks for classification". In: *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*. Vol. 1. IEEE. 1998, pp. 753–756.

[22] K. Zhang, H. Zhu, X. Li, and D. Evans. "Towards certified robustness under label noise". In: *International Conference on Learning Representations (ICLR)*. 2021.

[23] M. Yazdani, H. Karimi, and R. Samavi. "DENL: Diverse Ensemble and Noisy Logits for Improved Robustness of Neural Networks". In: *ACML*. PMLR. 2024, pp. 1574–1589.

[24] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330.

[25] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[26] J. Haldane. "Note on the median of a multivariate distribution". In: *Biometrika* 35.3-4 (1948), pp. 414–417.

[27] E. Weiszfeld. "Sur le point pour lequel la somme des distances de n points donnés est minimum". In: *Tohoku Mathematical Journal, First Series* 43 (1937), pp. 355–386.

[28] Y. LeCun, C. Cortes, and C. J. C. Burges. *The MNIST Database of Handwritten Digits*. http://yann.lecun.com/exdb/mnist/. 1998.

[29] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Technical Report, University of Toronto. CIFAR-10 dataset. 2009. URL: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[31] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. "TensorFlow: a system for Large-Scale machine learning". In: *12th USENIX symposium on OSDI 16*. 2016, pp. 265–283.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (2002), pp. 2278–2324.

[33] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proc. of the IEEE conf. on computer vision and pattern recognition*. 2016, pp. 770–778.