MORA: Improving Ensemble Robustness Evaluation with Model-Reweighing Attack

Yunrui Yu[*]
University of Macau
Macau SAR, China
yb97445@um.edu.mo

Xitong Gao[*]
Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Shenzhen, China
xt.gao@siat.ac.cn

Cheng-Zhong Xu[†]
University of Macau
Macau SAR, China
czxu@um.edu.mo

Abstract

Adversarial attacks can deceive neural networks by adding tiny perturbations to their input data. Ensemble defenses, which are trained to minimize attack transferability among sub-models, offer a promising research direction to improve robustness against such attacks while maintaining a high accuracy on natural inputs. We discover, however, that recent state-of-the-art (SOTA) adversarial attack strategies cannot reliably evaluate ensemble defenses, sizeably overestimating their robustness. This paper identifies the two factors that contribute to this behavior. First, these defenses form ensembles that are notably difficult for existing gradient-based method to attack, due to gradient obfuscation. Second, ensemble defenses diversify sub-model gradients, presenting a challenge to defeat all sub-models simultaneously, simply summing their contributions may counteract the overall attack objective; yet, we observed that ensemble may still be fooled despite most sub-models being correct. We therefore introduce MORA, a model-reweighing attack to steer adversarial example synthesis by reweighing the importance of sub-model gradients. MORA finds that recent ensemble defenses all exhibit varying degrees of overestimated robustness. Comparing it against recent SOTA white-box attacks, it can converge orders of magnitude faster while achieving higher attack success rates across all ensemble models examined with three different ensemble modes (*i.e.*, ensembling by either softmax, voting or logits). In particular, most ensemble defenses exhibit near or exactly 0% robustness against MORA with ℓ^∞ perturbation within 0.02 on CIFAR-10, and 0.01 on CIFAR-100. We make MORA open source with reproducible results and pre-trained models; and provide a leaderboard of ensemble defenses under various attack strategies[1].

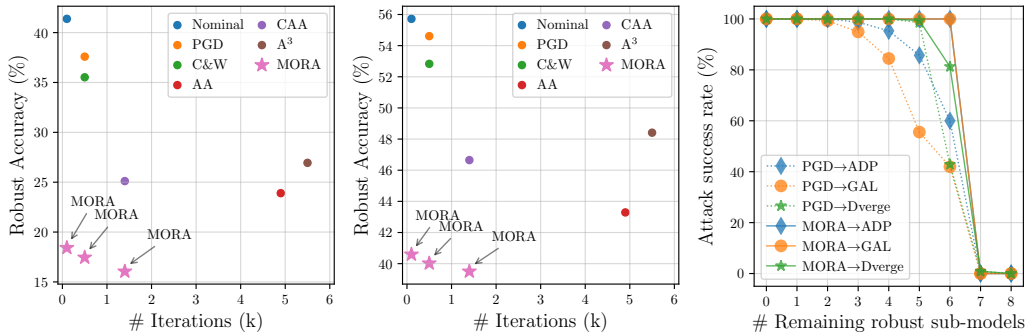
1 Introduction

Many safety-critical applications, such as autonomous robots [34], self-driving [8], search engines [24], *etc.* are becoming increasingly powerful and reliant on deep neural networks (DNNs). Despite the monumental success of DNNs on these applications, they are highly susceptible to adversarial examples — an attacker can add tiny delibrate perturbations to the input data, misleading the model into giving incorrect results [23, 9]. Such adversarial attacks could pose a significant threat to the safety and reliability of deep learning applications.

To mitigate this threat, many defense strategies [17, 33, 4] based on adversarial training [17] have been proposed to improve model robustness. Adversarial training, however, gains robustness at the expense of model accuracy on clean natural images [27]. Ensemble defenses [19, 14, 30, 31] have

[*]Xitong Gao and Yunrui Yu contributed equally to this work.
[†]Correspondence to: Cheng-Zhong Xu (cz.xu@um.edu.mo).
[1]https://github.com/lafeat/mora.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) GAL. (b) Dverge. (c) Robust sub-models *vs.* attacks.

Figure 1: (a,b) Existing attacks [17, 3, 5, 18, 16] with strong baselines are neither efficient in the number of model forward/backward passes, nor reliable in the estimation of ensemble robustness when compared with MORA. GAL [14] and Dverge [30] defenses are trained on CIFAR-10 with 8 sub-models. We used ℓ^∞ attacks within $\epsilon = 0.01$, "Nominal" is self-reported. (c) MORA can successfully fool logit-based ensembles (ADP [19], GAL and Dverge) even with the majority of their sub-models giving correct outputs ("$A \to B$" means using A to attack B for up to 100 iterations).

thus emerged to combine multiple predictions from independent sub-models. The intuition is that an ensemble of models can often lead to higher accuracy, while learning to stop adversarial example transfer among sub-models may improve robustness against adversarial attacks. This approach could potentially offer a promising research direction to improve model robustness while preserving high accuracy on natural inputs.

Yet surprisingly, under the white-box threat model, existing state-of-the-art (SOTA) adversarial attacks with strong performance on conventional DNN models performed poorly on ensemble models, sizeably overestimating their robustness (Figures 1a and 1b). This also suggests, to some extent, that ensemble defenses may rely on two potential design flaws below that cause obfuscated gradients [1], *i.e.*, they are either deliberately non-differentiable, or give no useful gradients, thus inducing overestimated robustness:

(a) *Gradient obfuscation via ensemble-forming strategy.* They typically form ensembles by averaging probability vectors (softmax) of sub-models, and softmax operations can easily cause gradient obfuscation. While the model's actual robustness is pertinent to the strategy used to form an ensemble, this indicates that gradient-based attacks have to *also* leverage this effectively.

(b) *Gradient diversification.* Motivated by the reasoning that a majority of sub-models may need to be fooled for successful attacks, they learn to reduce adversarial transferability among sub-models, often via gradient diversification. This intuitively causes sub-models to counteract each other, averaging to a small or inaccurate overall gradient. Attacking only the ensemble loss would fool most sub-models, but the ensemble may remain still correct; conversely, it is actually possible to fool an ensemble, despite the majority of its sub-models giving correct predictions (Figure 1c).

From the above observations, it is perceivable that the practical evaluation of ensemble robustness cannot be solely done by treating such models holistically. To this end, this paper introduces MORA, model-reweighing attack, to adaptively adjust the importance of sub-models in attack iterations. Sub-models are reweighed according to their respective "ease of attack", which is in turn evaluated by the gradient of the difference of ensemble classification outputs *w.r.t.* the ones of individual sub-models. Pushing the limits of the current SOTA in ensemble robustness evaluation, it draws inspiration from recent effective attack tactics, *e.g.*, momentum [7, 5], step size schedule [5, 16], loss normalization [32], and multiple targets [5, 25]. We summarize our contributions:

- This paper presents the first extensive study on the robustness of ensemble defenses under multiple ensemble-forming strategies.
- By reweighing the importance weights of sub-models to steer adversarial example synthesis, we show that gradient-based attacks on ensemble defenses can often be orders of magnitude faster, while enjoying a higher success rate.

- Empirical results on a wide variety of different ensemble defenses show that MORA outperforms competing attacks in both performance and convergence rate. Finally, this paper provides extensive ablation of its components and sensitivity analyses of hyperparameters.

To our best knowledge, MORA is currently the strongest attack against a wide range of ensemble defenses. We make MORA open source with reproducible results and pre-trained models; moreover, we maintain a leaderboard of ensemble defenses under various attack strategies.

2 Preliminaries & Related Work

2.1 Adversarial Attacks

An adversarial example adds a small perturbation, typically bounded a small value with ℓ^p norms, to the original image, such that the model under attack can be deceived into giving incorrect results. The advent of adversarial attacks [23] has piqued the interest of deep learning practitioners, and revealed security concerns of deep learning [26, 21], improved GAN training [2] transfer learning [28, 6], and DNN interpretability [22], *etc.* Formally, assuming a defending classifier $f : \mathcal{I} \to \mathbb{R}^K$, taking an input image $\mathbf{x} \in \mathcal{I} = [0, 1]^{C \times H \times W}$ as input, and $\arg\max f(\mathbf{x})$ predicts the correct class label $y \in \mathcal{C}$, then an attacker attempts to find an adversarial example $\hat{\mathbf{x}}$ in the set:

$$\{\hat{\mathbf{x}} \in \mathcal{A}_{\epsilon,\mathbf{x}} \colon \arg\max f(\hat{\mathbf{x}}) \neq y\}. \tag{1}$$

Here, $\hat{\mathbf{x}} \in \mathcal{A}_{\epsilon,\mathbf{x}}$ constrains the adversarial example $\hat{\mathbf{x}}$ to be within both the input space $\mathcal{I}$ and a small ϵ-ball of ℓ^p-distance from the original image $\mathbf{x}$, or equivalently $\|\mathbf{x} - \hat{\mathbf{x}}\|_p \leq \epsilon$. Satisfying the condition $\arg\max f(\mathbf{x}) \neq y$ means that $f(\mathbf{x})$ fails to give the expected correct classification y. We focus on the ℓ^∞ white-box threat model commonly considered by the defenses examined in this paper, which grants the attacker completely access to the internals of the defender, including, for instance, its model architecture, parameters, training algorithms, *etc.*

One of the popular and effective white-box attacks used by many defenders to evaluate their robustness is *projected gradient descent* (PGD) [17], which finds adversarial examples by maximizing the classification loss with gradient descent:

$$\hat{\mathbf{x}}_{i+1} = \mathcal{P}_{\epsilon,\mathbf{x}}(\hat{\mathbf{x}}_i + \alpha_i \operatorname{sign}(\nabla \mathcal{L}(f(\hat{\mathbf{x}}_i), y))), \tag{2}$$

where $\mathcal{L}$ is typically the softmax cross-entropy (SCE) loss used to train the model, α_i is the step size, and we let the initial $\hat{\mathbf{x}}_0 \triangleq \mathcal{P}_{\epsilon,\mathbf{x}}(\mathbf{x} + \boldsymbol{\mu})$. The projection function $\mathcal{P}_{\epsilon,\mathbf{x}}(\mathbf{v})$ constrains its input $\mathbf{v}$ to be within the feasible region $\mathcal{A}_{\epsilon,\mathbf{x}}$, and finally $\boldsymbol{\mu} \sim \mathcal{U}(-\epsilon, \epsilon)$ is a uniformly distributed noise bounded by $[-\epsilon, \epsilon]$. Besides PGD, C&W [3] is also a gradient-based attack which, instead of projection, indirectly constrains the search space by regularization.

As PGD gains popularity, many defense mechanisms rely on it to evaluate their robustness. Unfortunately, AutoAttack (AA) [5] finds that many of the defenses may inadvertently break PGD-based attacks, which result in drastic overestimation of their robustness, and proposes to combine an ensemble of diverse attacks to minimize robustness overestimation. LAFEAT [32] learns to leverage intermediate layers of the DNN, and shows that attacking multiple layers can produce stronger attacks, but unfortunately it cannot be applied to ensemble defenses. Adaptive Auto Attack (A^3) [16] improves attack success rates by using the gradient directions to prescribe a more effective initial random perturbation. As defenders may design mechanisms to circumvent existing attacks, Adaptive attacks [25] manually tailor specific attack strategies for an extensive set of defenses. Finally, Composite Adversarial Attacks (CAA) [18] further combine a large selection of attack methods, and use a genetic algorithm to learn an optimal attacking sequence. In comparison, MORA is a unified approach which uses only one attack algorithm, does not require a compute-intensive learning procedure, and yet it still achieves fast and SOTA estimation of ensemble robustness.

2.2 Defending Against Adversarial Attacks

Defending against adversarial attacks can be defined as a saddle-point problem to minimize the training loss on adversarial examples [17] with samples $(\mathbf{x}, y)$ drawn from the training set:

$$\min_{\boldsymbol{\theta}} \mathsf{E}_{(\mathbf{x},y)} \left[\max_{\hat{\mathbf{x}} \in \mathcal{A}_{\epsilon,\mathbf{x}}} \mathcal{L}(f(\hat{\mathbf{x}}), y)\right], \tag{3}$$

where $\mathcal{L}$ is the training loss, *e.g.*, the SCE loss. A direct optimization-based approach to approximately solving the above problem is *adversarial training* [17], *i.e.*, to train the DNN model with its own adversarial examples. Training DNNs to be robust is, however, a challenging endeavor. First, it may be much more compute intensive as training examples are typically generated with PGD [17], requiring a few forward/backward passes of the DNN. Second, to avoid overfitting, it requires stopping training early, a much larger size of the training set [4], and using improved data augmentation [20] or generated data [10]. Thirdly, as noted by other literatures [5, 32], currently no other design choices can rival the robustness provided by adversarial training, and notably, many defense strategies are considered harmful to model robustness [25]. Finally, the resulting models often cannot achieve high clean accuracy [27].

2.3 Ensemble Defenses & Ensemble-forming Strategies

Ensemble-based defense techniques may pave an alternative path to address the challenges of adversarial robustness, as they could potentially work around the above limitations of adversarial training. Adopting the theme of minimizing adversarial example transferability across sub-models, each ensemble defense proposed unique solutions. ADP [19] increases the orthogonality of non-maximal class logits among sub-models to encourage diversity. GAL [14] minimizes a gradient alignment loss, which directly reduces the cosine-similarity between sub-models. Building on top of this, TRS [31] further regularizes the smoothness of the loss function, as gradient orthogonality with smoothness may further diversify sub-models. Dverge [30] instead uses the adversarial examples of a sub-model to train another sub-model, thus lowering transferability. Ensemble defenses are also particularly interesting, as they are the last line of defense against even the strongest existing white-box attacks without resorting to adversarial training, showing a certain degree of robustness.

Besides the above mechanisms for training a successful ensemble defense, there exists different ways to combine sub-model predictions. Let us assume that an ensemble defense trains M sub-models, $f_m : \mathcal{I} \to \mathbb{R}^K$ for $m \in [1 : M]$, an ensemble $f_{\mathrm{E}} : \mathcal{I} \to \mathbb{R}^K$ thus forms a final classification result by combining individual decisions from the sub-models, namely:

$$f_{\mathrm{E}}(\mathbf{x}) = \tfrac{1}{M} \textstyle\sum_{m \in [1:M]} \mathrm{ens}(f_m(\mathbf{x})), \tag{4}$$

where ens is the ensemble-forming operator. In this paper, we investigate $\mathrm{ens} \in \{\mathrm{softmax}, \mathrm{wta}, \mathrm{id}\}$, where the potential choices respectively denoting forming an ensemble from sub-model outputs $f_m(\mathbf{x})$ by either summing predicted probabilities (evaluated with $\mathrm{softmax}$), or majority votes (using wta, the winner-take-all operator), or simply summing the logits (with id, the identity operator). Defending ensemble methods [19, 14, 30, 31] tested in this paper all employed the $\mathrm{softmax}$-based strategy to report their robustness. Methods that are exceptions to these options exist, for instance, ECOC [29] allows sub-models to produce binary predictions, and use error correcting codes based on the Hamming distance to combine the predictions into classification outputs. This approach is unfortunately not robust, and the added complexity is error-prone and may harm robustness [25].

Moreover, as the voting (wta) strategy is non-differentiable, an attacker can soften it approximately using a softmax operation with temperature τ, where we used $\tau = 0.1$ universally:

$$\mathrm{softwta}_\tau(\mathbf{x}) \triangleq \mathrm{softmax}(\mathbf{x}/\tau). \tag{5}$$

3 The Model-Reweighing Attack (MORA)

3.1 Problem Formulation & High-Level Overview

As discussed in Section 1, existing ensemble defenses may obfuscate gradients with the ensemble-forming mode and gradient diversification, such that the final loss of the ensemble model can no longer provide effective signals for gradient descent. It is therefore desirable to find an alternative $\mathcal{L}$ to the original SCE loss $\mathcal{L}^{\mathrm{sce}}$ on the ensemble, such that for a given number of iterations I, the original $\mathcal{L}^{\mathrm{sce}}$ loss can be maximized:

$$\begin{aligned} &\max_{\mathcal{L}} \mathcal{L}^{\mathrm{sce}}(\hat{\mathbf{x}}_I, y) \quad \text{where} \quad \hat{\mathbf{x}}_0 = \mathcal{P}_{\epsilon,\mathbf{x}}(\mathbf{x} + \boldsymbol{\mu}), \\ &\hat{\mathbf{x}}_{i+1} = \mathrm{PGD}(\mathcal{L}(f_E(\hat{\mathbf{x}}_i), f_{[1:M]}(\hat{\mathbf{x}}_i), y)), \end{aligned} \tag{6}$$

and $\mathrm{PGD}(\cdot)$ denotes a PGD step along the gradient of loss function $\mathcal{L}(f_{\mathrm{E}}(\hat{\mathbf{x}}_i), f_{[1:M]}(\hat{\mathbf{x}}_i), y)$, which not only takes the ensemble predictions $f_E(\hat{\mathbf{x}}_i)$ as input, but can further utilize sub-model predictions

$f_{[1:M]}$ to guide the PGD iterations. The challenge at hand is, therefore, to find a suitable $\mathcal{L}$ which can generate attacks on ensemble defenses efficiently and effectively.

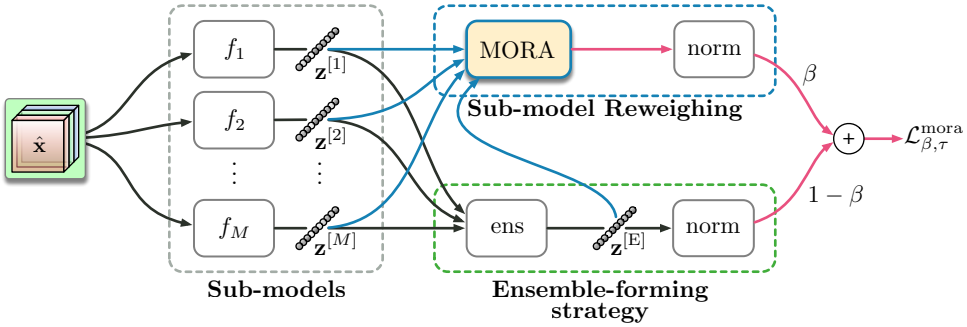


Figure 2: A high-level overview of MORA. Existing attack methods focus on maximizing the $\mathcal{L}^{\mathrm{sce}}$ of the ensemble predictions $\mathbf{z}^{[\mathrm{E}]}$. MORA further introduces a model-reweighing mechanism, which takes as input sub-models predictions $\mathbf{z}^{[1:M]}$, and the ensemble output $\mathbf{z}^{[\mathrm{E}]}$, forming a combined loss $\mathcal{L}^{\mathrm{mora}}_{\beta,\tau}$. MORA generates adversarial examples by maximizing the resulting loss.

MORA aims to provide a potential optimization route towards the above problem formulation. Namely, in addition to the original output of the ensemble $f_{\mathrm{E}}(\hat{\mathbf{x}})$, we leverage the sub-model predictions $f_{[1:M]}(\hat{\mathbf{x}})$ to facilitate the optimization. By way of illustration, Figure 2 shows a high-level overview of the model-reweighing attack, where we compliment the ensemble loss, with a newly added sub-model reweighing loss $\mathcal{L}^{\mathrm{mora}}_{\beta,\tau}$, as an auxiliary attack vector alongside the original objective. Not only can the new loss bypass the ensemble-forming strategy to work around its obfuscated gradients, but it further exploits information present in the individual sub-model and ensemble predictions to steer the direction of adversarial example synthesis.

3.2 Adaptive Sub-model Importance

Before we begin, assume that $\mathbf{z}^{[m]} \triangleq f_m(\mathbf{x})$ represents the mth sub-model output, and let $\mathbf{z}_t^{[m]}$ denote the corresponding logit of label t. We define the difference of logits (DL) [3] $k^{[m]} \triangleq \mathbf{z}_y^{[m]} - \mathbf{z}_{\hat{y}}^{[m]}$, which is the difference between the predictions of the ground truth $\mathbf{z}_y^{[m]}$ and the maximum of the remaining classes $\mathbf{z}_{\hat{y}}^{[m]} \triangleq \max_{i \in \mathcal{C}/y} \mathbf{z}_i^{[m]}$, where $\mathcal{C}/y$ is the set of all class labels except y. Similarly, we let $\mathbf{z}^{[\mathrm{E}]} \triangleq f_{\mathrm{E}}(\mathbf{x})$, and $\mathbf{z}_t^{[\mathrm{E}]}$ and $k^{[\mathrm{E}]}$ be the respective variants of the ensemble prediction. It is notable that a successful attack happens when $k^{[\mathrm{E}]} < 0$, and similarly $k^{[m]} < 0$ means the mth sub-model is producing incorrect classification.

Ensemble defenses tend to diversify sub-model gradients, for instance, ADP [19] minimizes the cosine-similarity $\langle \nabla\ell(\mathbf{z}^{[a]}), \nabla\ell(\mathbf{z}^{[b]}) \rangle$ among each loss function gradient pairs of sub-models $\ell(\mathbf{z}^{[a]})$ and $\ell(\mathbf{z}^{[b]})$. Their intuition is that it may lower transferability among these sub-models, such that attacks with the overall gradient of the ensemble, *i.e.*, $\nabla\ell(\mathbf{z}^{[\mathrm{E}]}) = \frac{1}{M}\sum_{m\in[1:M]} \nabla\ell(\mathbf{z}^{[m]})$, are becoming less effective in misleading all sub-models simultaneously, as individual gradients in $\nabla\ell(\mathbf{z}^{[m]})$ are encouraged to be orthogonal to each other. To this end, we propose to reweigh the importance of sub-models, by instead considering the modified gradient:

$$\widehat{\nabla\ell(\mathbf{z}^{[\mathrm{E}]})} = \frac{1}{M}\sum_{m\in[1:M]} \lambda_\tau^{[m]}(\mathbf{z}^{[m]}) \nabla\ell(\mathbf{z}^{[m]}), \tag{7}$$

where $\lambda_\tau^{[m]}$ assigns weights to important sub-models to contribute more heavily to the attack gradient.

While the adversarial examples of the ensemble could present a challenge to discover, individual sub-models are weak defenders which can be easily defeated. Based on this property, we propose to weigh sub-models importance with the rate of change in $k^{[\mathrm{E}]}$ *w.r.t.* that of $k^{[m]}$, *i.e.*, sub-models would be given higher weights if attacking it would bring a significant change to the ensemble's prediction. Following this idea, for all ensemble-forming strategies (softmax, voting, logits), we rewrite $k^{[\mathrm{E}]}$ as a function of $k^{[m]}$, where the term below can become a function of $k^{[m]}$:

$$\begin{aligned} k^{[\mathrm{E}]} = \mathrm{ens}(\mathbf{z}^{[m]})_y - \mathrm{ens}(\mathbf{z}^{[m]})_{\hat{y}} &= \mathrm{ens}(\mathbf{z}^{[m]} - \mathbf{z}_{\hat{y}}^{[m]})_y - \mathrm{ens}(\mathbf{z}^{[m]} - \mathbf{z}_{\hat{y}}^{[m]})_{\hat{y}} \\ &= \mathrm{ens}(k^{[m]}, \cdots)_y - \mathrm{ens}(k^{[m]}, \cdots)_{\hat{y}} \triangleq h_m(k^{[m]}). \end{aligned} \tag{8}$$

Algorithm 1 The MORA white-box robust evaluation for ensemble defenses.

function MORA_Attack($f_{[1:M]}, \mathbf{x}, y, \beta, \tau, \nu, \epsilon, I$)
  $\hat{\mathbf{x}}_0 \leftarrow \mathcal{P}_{\epsilon,\mathbf{x}}\left(\mathbf{x}+\mathbf{u}\right)$, where $\mathbf{u} \sim \mathcal{U}\left(-\epsilon, \epsilon\right)$ ▷ Random init
  $\boldsymbol{\mu}_0 \leftarrow 0$
  for $i \in [0 : I-1]$ **do**
    $\mathbf{z}^{[m]} \leftarrow f_m(\hat{\mathbf{x}}_i)$ for all $m \in [1:M]$ ▷ Sub-model predictions
    $\mathbf{z}^{[\mathrm{E}]} \leftarrow \frac{1}{M}\sum_{m\in[1:M]} \mathrm{ens}(\mathbf{z}^{[m]})$ ▷ Ensemble prediction
    $k^{[\mathrm{E}]} \leftarrow \mathbf{z}^{[\mathrm{E}]}_y - \mathbf{z}^{[\mathrm{E}]}_{\hat{y}}$
    if $k^{[\mathrm{E}]} \leq 0$ **then return** $\hat{\mathbf{x}}_i$ ▷ Successful attack
    $\boldsymbol{g}_{i+1} \leftarrow \mathrm{sign}(\nabla_{\hat{\mathbf{x}}_i} \mathcal{L}^{\mathrm{mora}}_{\beta,\tau}(\mathbf{z}^{[1:M]}, \mathbf{z}^{[\mathrm{E}]}, y))$ ▷ Sign-gradient with the MORA loss
    $\alpha \leftarrow \epsilon(1+\cos(i\pi/I))$ ▷ Cosine step-size schedule
    $\boldsymbol{\mu}_{i+1} \leftarrow \mathcal{P}_{\epsilon,\mathbf{x}}\left(\boldsymbol{\mu}_i + \alpha \boldsymbol{g}_{i+1}\right)$ ▷ Iterative update
    $\hat{\mathbf{x}}_{i+1} \leftarrow \mathcal{P}_{\epsilon,\mathbf{x}}\left(\hat{\mathbf{x}}_i + \nu\left(\boldsymbol{\mu}_{i+1} - \hat{\mathbf{x}}_i\right) + (1-\nu)\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{i-1}\right)\right)$ ▷ ... with momentum
  end for
  return $\hat{\mathbf{x}}_I$ ▷ Give up after I iterations
end function

The weights are thus defined as follows:

$$\lambda^{[m]}_{\tau}(\mathbf{z}^{[m]}) = \frac{\partial k^{[\mathrm{E}]}(k^{[m]})}{\partial k^{[m]}} = \frac{\partial}{\partial k^{[m]}}\left(\frac{1}{M}\sum_{m\in[1:M]}\frac{\partial h_m(k^{[m]})}{\partial k^{[m]}}\right) = \frac{1}{M}\frac{\partial h_m(k^{[m]})}{\partial k^{[m]}}. \tag{9}$$

While it is possible to compute the weights using gradient back-propagation, we can simply derive the following closed-form solution of the weights for each of the three ensemble-forming strategies. For wta, we use the softened version of wta as defined in (5) and can derive the weights as follows:

$$\lambda^{[m]}_{\tau}(\mathbf{z}^{[m]}) = 1[k^{[m]} > 0] \cdot \mathrm{detach}\big(\tfrac{1}{\tau M}\mathbf{s}_{\hat{y}}(1+\mathbf{s}_y - \mathbf{s}_{\hat{y}})\big), \quad \text{where } \mathbf{s} = \mathrm{softmax}\big(\mathbf{z}^{[m]}/\tau\big). \tag{10}$$

Here $1[k^{[m]} > 0]$ is the indicator function that equals 1 if $k^{[m]} > 0$, or 0 otherwise, effectively stopping the attack on the mth sub-model upon success, and the detach operator admits no backward propagation to its input. In the case of using sums of sub-model softmax outputs to form an ensemble decision, *i.e.*, $\mathrm{ens} = \mathrm{softmax}$, it is a special case of $\mathrm{softwta}_\tau$ where the temperature coefficient can be fixed at $\tau = 1$. Finally, when $\mathrm{ens} = \mathrm{id}$, *i.e.*, forming ensembles by summing logits, $\lambda^{[m]}_{\tau}(\mathbf{z}^{[m]})$ simply reduces to $1[k^{[m]} > 0]$ for the mth sub-model.

3.3 The MORA Loss

For reference, defenses mechanisms we examine in this paper aim to find $\hat{\mathbf{x}}$ which maximizes the SCE loss $\mathcal{L}^{\mathrm{sce}}(\mathbf{z}^{[\mathrm{E}]}, y)$, to evaluate the ensemble robustness. The MORA loss improves this further by proposing two additional modifications to the untargeted loss function used to attack ensembles:

$$\mathcal{L}^{\mathrm{mora}}_{\beta,\tau}(\mathbf{z}^{[1:M]}, \mathbf{z}^{[\mathrm{E}]}, y) \triangleq \mathcal{L}^{\mathrm{sce}}\Big(\beta\,\mathrm{norm}\Big(\sum_{m\in[1:M]}\lambda^{[m]}_{\tau}(\mathbf{z}^{[m]})\cdot\mathbf{z}^{[m]}\Big) + (1-\beta)\,\mathrm{norm}(\mathbf{z}^{[\mathrm{E}]}), y\Big). \tag{11}$$

First, it additionally introduces a sum of the $\lambda^{[m]}_{\tau}$-weighted variant of sub-model logits, in order to expose sub-model logits with adaptive reweighing described in Section 3.2. Second, β interpolates the importance of the newly added auxiliary logits and the original ensemble logits. Finally, inspired by the effective surrogate loss in [32], it further normalizes the logits by their respective DL using:

$$\mathrm{norm}(\mathbf{z}) \triangleq 1[\mathbf{z}_y - \mathbf{z}_{\hat{y}} > 0] \cdot \mathbf{z}/\mathrm{detach}(\mathbf{z}_y - \mathbf{z}_{\hat{y}}). \tag{12}$$

Finally, the targeted variant of the MORA loss simply replaces y with t where t is the intended target.

3.4 Improving the State-of-the-art

While the new $\mathcal{L}^{\mathrm{mora}}_{\beta,\tau}$ loss is highly effective against ensemble defenses we test in this paper, we strive for further advances in MORA's ability to generate faster and better adversarial examples. Inspired by recent publications, we borrow ideas from related adversarial attack tactics, which includes adopting a cosine step-size schedule [16], momentum [7, 5], random restarts [25] and multiple target attacks [5, 25]. We provide the overall algorithm in Algorithm 1, which computes an adversarial image $\hat{\mathbf{x}}_I$ as return, by taking as input the sub-models $f_{[1:M]}$, natural image $\mathbf{x}$, ground truth label y, β to interpolate between the auxiliary logits and the original, τ controls the temperature, momentum $\mu = 0.75$ following [32, 5], ϵ perturbation bound, and finally the maximum number of iterations I.

Table 1: Comparing accuracies among iterative methods [17, 3], learned attacks (**CAA** [18]), AutoAttack (**AA**) [5], adaptive auto attack (**A**^{3}) [16], and MORA across various ensemble defense strategies under 3 ensembling modes (softmax, voting and logits), and $\epsilon = 0.01$. The "Complexity" row shows the worst-case complexity in iteration counts. The "$\mathbf{\Delta}$" column shows the accuracy overestimation from self-reported/reproduced "**Nominal**" values to MORA^{mt}. Baselines with † are reproduced with source code. All results are re-run 5 times and within $\pm 0.05\%$ standard deviation.

Defense Complexity	**#**	**Clean** 1	**Nominal** —	**PGD** 500	**CW** 500	**MORA** 500	**A**^{3} 12k	**AA** 4.9k	**CAA** 1.8k	**MORA**^{mt} 1.4k	$\mathbf{\Delta}$
Softmax											
ADP	3	92.88	29.12	5.98	7.72	0.59	2.12	0.98	3.34	**0.34**	28.78
	5	93.34	25.14	7.10	8.70	0.97	3.62	2.18	4.25	**0.67**	24.47
	8	93.48	20.20	9.22	9.59	1.70	4.84	3.94	6.04	**1.32**	18.88
Dverge	3	91.99	47.42	44.49	40.17	25.77	33.36	30.58	32.98	**25.26**	22.16
	5	92.38	55.72	54.61	52.83	40.02	48.41	43.29	46.65	**39.50**	16.22
	8	91.65	59.63	59.13	58.25	55.68	57.29	56.71	56.89	**55.57**	4.06
GAL	3	89.41	19.48	8.13	11.57	0.67	0.70	0.85	1.00	**0.51**	18.97
	5	90.93	41.38	37.59	35.52	17.45	26.94	23.90	25.11	**16.05**	25.33
	8	92.45	56.31	53.39	52.56	28.71	36.51	37.46	35.30	**27.44**	28.87
TRS†	3	70.02	19.71	14.01	10.87	8.11	8.72	8.46	9.75	**7.60**	12.11
	5	69.00	23.17	15.91	15.28	12.67	13.22	13.20	13.78	**12.47**	10.70
	8	73.01	23.64	18.02	17.59	15.90	16.22	16.51	16.73	**15.64**	8.00
Voting											
ADP	3	91.84	41.62†	9.32	11.84	0.64	3.06	6.13	8.29	**0.29**	41.33
	5	93.13	40.29†	12.42	12.05	1.17	6.03	10.13	0.67	**0.62**	39.67
	8	93.28	30.10†	12.53	10.50	3.16	6.11	9.21	1.69	**1.65**	28.45
Dverge	3	91.72	39.05†	31.48	28.00	23.57	24.95	24.98	27.65	**22.91**	16.14
	5	92.18	49.36†	44.28	42.28	35.06	39.15	39.20	40.85	**34.46**	14.90
	8	91.58	56.85†	53.72	52.35	47.12	50.58	50.04	51.15	**46.10**	10.75
GAL	3	89.09	21.48†	5.85	7.64	0.87	0.71	0.56	0.78	**0.35**	21.13
	5	90.77	37.32†	29.33	27.62	12.96	18.55	20.82	22.17	**12.25**	25.07
	8	92.37	55.39†	49.56	48.02	21.66	30.35	31.39	30.93	**20.16**	35.23
TRS†	3	68.95	13.79	10.19	8.71	5.73	11.89	6.69	8.08	**5.44**	8.35
	5	68.31	15.36	12.71	11.88	8.82	10.08	10.30	11.21	**8.38**	6.98
	8	72.05	17.00	14.57	13.48	11.39	11.99	11.85	12.80	**10.69**	6.31
Logits											
ADP	3	92.86	3.44†	0.87	2.05	0.48	0.25	0.22	0.31	**0.21**	3.23
	5	93.48	4.57†	1.97	4.24	1.12	1.00	0.97	1.09	**0.89**	3.68
	8	93.38	5.39†	3.57	4.77	2.13	2.20	2.05	2.11	**1.93**	3.46
Dverge	3	92.19	38.31†	37.99	38.60	36.89	36.94	36.96	37.07	**36.84**	1.47
	5	92.28	50.77†	50.57	51.28	49.65	49.72	49.66	49.75	**49.59**	1.18
	8	91.73	61.06†	60.95	61.51	60.52	60.59	60.52	60.55	**60.49**	0.57
GAL	3	89.50	15.47†	10.01	10.53	0.52	**0.02**	**0.02**	0.08	0.03	15.44
	5	90.93	36.36†	33.97	35.14	22.24	33.43	20.24	21.66	**19.40**	16.96
	8	92.54	56.08†	53.67	54.69	31.52	40.90	30.89	31.17	**30.66**	25.42
TRS†	3	69.72	13.31	13.06	13.80	12.11	12.13	12.16	12.21	**12.07**	1.24
	5	68.90	16.89	16.65	17.34	15.88	15.86	15.90	15.95	**15.82**	1.07
	8	72.24	19.40	19.20	19.67	18.20	18.18	18.27	18.34	**18.17**	1.23

4 Experimental Results

We compare MORA against SOTA attacks for a wide range of ensemble defenses under three ensemble-forming strategies (softmax, voting, and logits). We use pre-trained ResNet-20 [12] models from ADP [19], Dverge [30], GAL [14], and reproduced TRS [31] using the same architecture with official source code, as pre-trained models were unavailable. Our robustness evaluation considers the ℓ^∞ white-box attacks on the CIFAR-10 test set [15], with perturbation $\epsilon = 0.01$ unless specified. The full comparison results can be found in Table 1; larger ϵ comparisons, and similar results on CIFAR-100 models are in Appendix A. We provide our key observations below.

Traditional attacks may fail to break through gradient obfuscation. We reproduce two traditional white-box attacks, *i.e.*, projected gradient descent (PGD) [17] and C&W [3] with 5 random restarts, each with a maximum of 100 iterations, giving a total of 500 iterations. PGD uses a fixed step size of $\epsilon/4$. For a fair comparison, MORA with 500 iterations sweeps $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$, with each β up to 100 iterations. Even with a 500 iteration budget, it is clear that PGD and C&W may substantially overestimate robustness, especially when tested under the softmax and voting ensemble-forming options, and MORA can work around this obstacle thanks to its attacks on sub-models.

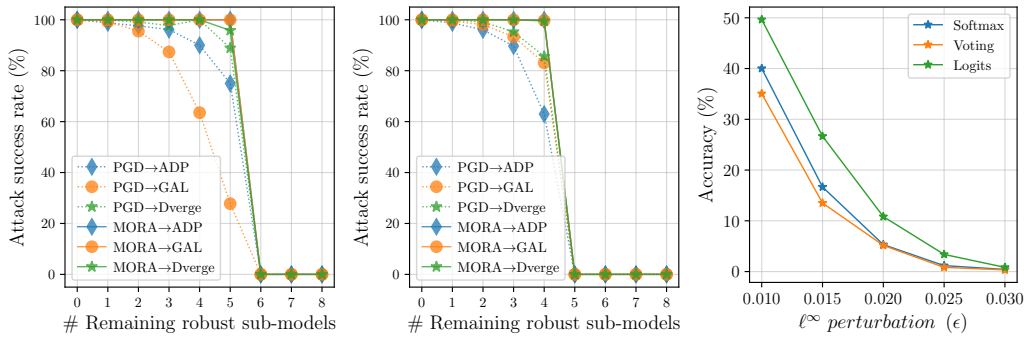
Diversified gradients can hamper even integrated attacks with large arsenals. Moreover, we test the defenses against recent integrated attacks with SOTA baselines on robustness evaluation, namely Adaptive Auto Attack (A^3) [16], AutoAttack (AA) [5], and Composite Adversarial Attacks (CAA) [18], which comprise large arsenals of various attack strategies. We reproduce CAA following [18] to search for the attack policy before evaluating the defending models. Note that its computational complexity is thus much higher than the other attacks, but we only report its test-time complexity. In particular, while they enjoyed much higher success rates than PGD and C&W, some defenses render their tactics ineffective. We observe, *e.g.*, sizeable robustness overestimation on ADP [19] under softmax and voting, which explicitly diversifies sub-model gradients. As MORA can dynamically re-adjust sub-model importance *w.r.t.* their "ease-of-attack", it performs substantially better with much fewer iterations. In addition to the earlier 500 iterations, the multi-targeted MORA^{mt} targets the remaining 9 class labels with 100 iterations for each label and β fixed at 0.5. Others also use multi-targeted attacks along with respective tactics.

Robustness of most sub-models *vs.* robustness of ensemble. We find that robustness of a majority of sub-models (fooling $3/8$ for softmax and $2/8$ for logits) usually do not translate to the overall robustness of the ensemble (Figures 1c and 3a). As voting requires breaking $1/2$ sub-models simultaneously (Figure 3b), it is perceivable that using voting may give rise to a higher overall robustness. Yet surprisingly, for most defending ensembles, voting performs worse than softmax and logits.

Ensemble-forming strategies may give a false sense of security. On one hand, softmax and voting strategies exhibit substantially larger overestimated robustness (up to 40%) than logits. On the other hand, in stark contrast to the proposed use of softmax from [14, 19, 30, 31], we find summing by logits can form ensembles that are notably more robust than the other two (Figure 3c), while attackers only needs to successfully deceive a few sub-models (referring back to Figure 1c).

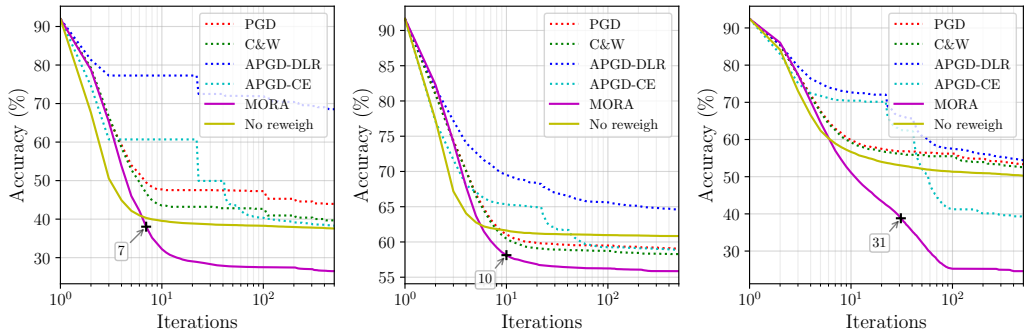
Up to $60\times$ faster convergence under 500 iterations. Figure 4 compares the convergence speed of MORA against AA losses, C&W, and PGD on defending ensembles. MORA converges substantially faster than the other attacks, using only up to 31 steps to match AA losses with 500 iterations.

Ensemble defense mechanisms may be at odds with robustness. In Table 2, we compare respective attacks on adversarially trained Dverge models. To our surprise, forming larger ensembles is actually harmful to the robustness of ensemble.



(a) Softmax. (b) Voting. (c) Dverge 5 *w.r.t.* ϵ bound.

Figure 3: (a, b) MORA can successfully fool ensemble-forming methods (ADP [19], GAL [14] and Dverge [30]) even with the majority of their sub-models giving correct outputs under softmax and logits (Figure 1c). While voting requires $\geq 1/2$ sub-models to be incorrect, it is unfortunately the least robust option in all defenses. "$A \rightarrow B$" means using A to attack B for up to 100 iterations. (c) Dverge with 5 sub-models *w.r.t.* the ϵ bound on ℓ^∞ perturbation. Contrary to existing literatures, we find logits to be the most robust option of the three ensemble-forming strategies.



(a) Dverge 3. (b) Dverge 8. (c) GAL 8.

Figure 4: (a, b, c) Comparing the convergence speed of MORA against a variant of MORA without sub-model reweighing, C&W, PGD and AA losses (APGD-{DLR,CE}) on Dverge and GAL with softmax. While "No reweigh" converge faster initially, it struggles to improve after 10 iterations; in contrast, adaptive reweighing (MORA) continues to converge at a much faster rate. The horizontal and vertical axes respectively show the iteration count used, and the percentage of unsuccessful images remaining. We annotated the number of iterations for MORA to overtake all with 500 iterations.

Table 2: Attacking adversarially trained Dverge [30] models under the same setting as Table 1, except we let $\epsilon = 0.03$. Notably, forming larger ensembles can actually be detrimental to robustness.

Dverge Complexity	**#**	**Clean** 1	**Nominal** —	**PGD** 500	**CW** 500	**MORA** 500	**A^3** 12k	**AA** 4.9k	**CAA** 1.8k	**MORA^{mt}** 1.4k	**Δ**
Softmax	3	83.78	45.09	44.85	44.21	42.91	42.66	42.70	42.69	**42.65**	2.44
	5	86.09	42.57	42.40	42.51	41.05	**40.74**	40.85	40.84	40.75	1.79
	8	86.69	40.80	40.58	40.94	39.49	**39.33**	39.40	39.39	39.35	1.41
Voting	3	83.67	59.13†	55.35	55.85	38.59	38.78	39.97	40.31	**38.24**	20.89
	5	86.05	47.72†	44.32	45.18	36.19	36.42	37.29	37.89	**36.01**	11.71
	8	86.54	38.93†	37.16	38.17	34.20	34.89	35.54	36.19	**34.03**	4.90
Logits	3	83.74	44.83†	44.69	44.24	42.83	42.66	42.70	42.70	**42.63**	2.20
	5	86.03	42.47†	42.22	42.63	41.00	40.91	40.92	40.92	**40.87**	1.60
	8	86.65	40.53†	40.33	41.14	39.53	**39.43**	39.50	39.47	**39.43**	1.09

Additional results, ablation, and sensitivity analyses. Due to the page limit, we provide full results of relevant figures in Appendix A, note that the above key observations still hold true for all ensemble defenses we test under different ensemble-forming strategies and ϵ perturbation bounds. In addition, we provide extensive ablation study on the design choices we made, and sensitivity analysis on the temperature constant τ.

5 Conclusions

This paper identifies severe robustness overestimation in many ensemble defense techniques, and further investigates problem the robustness evaluation under three ensemble-forming strategies. To efficiently and accurately evaluate the robustness of ensembles, we introduce MORA, a new attack technique which reweighs sub-model importance adaptively by their respective "ease-of-attack" during attack iterations. MORA enjoys a much improved success rate and convergence rate compared with other SOTA attacks. Moreover, we found several surprising observations related to ensemble defenses, for instance, (1) misleading a minority of sub-models is sufficient to fool the ensemble, (2) summing by logits is the simplest yet most robust way to form ensembles, (3) with adversarial training, ensemble defenses may actually harm robustness, *etc.* We hope the above observations may help to guide future avenue on ensemble defenses, and provide a strong attack baseline for potential approaches. Finally, MORA is open source with reproducible results and pre-trained models; and we continually update a leaderboard of ensemble defenses under various attack strategies.

Acknowledgements

This work is supported in part by National Key R&D Program of China (No. 2019YFB2102100), Key-Area Research and Development Program of Guangdong Province (No. 2020B010164003), Science and Technology Development Fund of Macao S.A.R (FDCT) under No. 0015/2019/AKP, and Shenzhen Science and Technology Innovation Commission (No. JCYJ20190812160003719). This work was carried out in part at SICC which is supported by SKL-IOTSC, University of Macau.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018.

[2] Dina Bashkirova, Ben Usman, and Kate Saenko. Adversarial self-defense for cycle-consistent GANs. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[4] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, volume 32, pages 11192–11203, 2019.

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[6] Zhun Deng, Linjun Zhang, Kailas Vodrahalli, Kenji Kawaguchi, and James Y Zou. Adversarial training helps transfer learning via better representations. *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[8] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[10] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[11] Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for PGD-based adversarial testing. *arXiv 1910.09338*, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] Bo Huang, Zhiwei Ke, Yi Wang, Wei Wang, Linlin Shen, and Feng Liu. Adversarial defence by diversified simultaneous training of deep ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7823–7831, 2021.

[14] Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

[15] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 and CIFAR-100 datasets, 2014. Available at: `http://www.cs.toronto.edu/~kriz/cifar.html`.

[16] Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. Practical evaluation of adversarial robustness via adaptive auto attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[18] Xiaofeng Mao, Yuefeng Chen, Shuhui Wang, Hang Su, Yuan He, and Hui Xue. Composite adversarial attacks. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[19] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

[20] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

[22] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

[24] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5037–5046, 2019.

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

[26] Florian Tramèr, Pascal Dupré, Gili Rusak, Giancarlo Pellegrino, and Dan Boneh. AdVersarial: Perceptual Ad blocking meets adversarial machine learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2005–2021, 2019.

[27] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[28] Francisco Utrera, Evan Kravitz, N. Benjamin Erichson, Rajiv Khanna, and Michael W. Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *International Conference on Learning Representations*, 2021.

[29] Gunjan Verma and Ananthram Swami. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[30] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. DVERGE: Diversifying vulnerabilities for enhanced robust generation of ensembles. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5505–5515, 2020.

[31] Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin Rubinstein, Ce Zhang, and Bo Li. TRS: Transferability reduced ensemble via promoting gradient diversity and model smoothness. *Advances in Neural Information Processing Systems*, 34, 2021.

[32] Yunrui Yu, Xitong Gao, and Cheng-Zhong Xu. LAFEAT: Piercing through adversarial defenses with latent features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5735–5745, June 2021.

[33] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[34] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1945–1960, 2021.

A Additional Results

A.1 Larger Perturbations

Tables 3 and 4 compare the effectiveness of iterative methods [17, 3], learned attacks (**CAA** [18]), AutoAttack (**AA**) [5], MORA and MORA^{mt} across various ensemble defense strategies with ResNet-20 [12] sub-models under 3 ensemble-forming modes (softmax, voting and logits). Here, Table 3 uses an ℓ^∞ perturbation bound $\epsilon = 0.02$, whereas Table 4 uses $\epsilon = 0.03$. All results are re-run 5 times and within $\pm 0.05\%$ standard deviation. Note that under $\epsilon = 0.02$ most defense methods only retain up to 4% robust accuracies. Dverge [30] with 8 sub-models however surprisingly has 23.31% robustness against the strongest MORA^{mt} attack. We believe this can be attributed to the fact that similarly to adversarial training, Dverge diversifies sub-models with adversarial examples from each other, rather than via explicit regularization as carried out by the other defenses. It thus requires substantially higher training cost than the others ($\sim 5\times$ ADP [19]).

Finally, Figure 5 compares the different ensemble-forming strategies under an increasing ℓ^∞ perturbation bound $\epsilon \in \{0.01, 0.015, 0.02, 0.025, 0.03\}$. Note that across all configurations, logit-based ensembles are the most robust.

A.2 Convergence Speed Comparisons

Figures 6 and 7 compare the convergence speed of MORA against SOTA attacks for up to 500 iterations. As AA comprises multiple attack strategies, we extract its two gradient-based attacks (APGD-{CE,DLR}) to facilitate comparisons. Each attack uses 5 restarts, each with up to 100 iterations. The horizontal axes show the iteration counts used, and the vertical axes denote the percentages of remaining unsuccessful images. When forming ensembles with logits, MORA and "No reweighing" are both identical. Note that MORA substantially outperforms most existing attacks under 500 iterations, and generally requires (up to $70\times$) fewer iterations to achieve the same accuracies of other attacks with 500 iterations.

A.3 Ablation and Sensitivity Analyses

In Table 5, we perform ablation of the individual components used in MORA. We begin with the standard "PGD" attack [17] with 5 random restarts, each with 100 iterations, and a constant step size of $\epsilon/4$. Each row then consecutively adds a new component. "Momentum" introduces momentum $\nu = 0.75$ as used in Algorithm 1, "Cosine Step Size" then replaces the constant step size with a cosine schedule $\alpha_i = \epsilon(1 + \cos(i\pi/I))$. "Sub-model Logits" further exploits the sub-model logits directly following Section 3.1, and replaces random restarts with a $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$ grid search to match the cost of 500 iterations. "Logit Normalization" incorporates the normalization of logits as proposed in (12) of Section 3.3. "Adaptive Reweighing" adaptively adjusts sub-model weights for attacking ensemble defenses ((10) of Section 3.2). Finally, "Multiple Targets" [11] additionally uses 100 iterations of the targeted variant of $\mathcal{L}^{\text{mora}}_{\beta,\tau}$ on each of the remaining 9 class labels (Section 3.3).

Increasing the temperature coefficient τ from 1 as used in (11) can affect the convergence speed of MORA. To ensure a fair comparison in our results, we fix a constant $\tau = 5$ for softmax-based ensembles, and $\tau = 10$ for voting, as increasing τ may help improve convergence. In addition, and Figure 8 provides sensitivity analyses of τ on the three defending methods (ADP, GAL, Dverge).

Finally, Figure 9 shows the effect of varying $\beta \in [0, 1]$, the interpolation between $\mathcal{L}^{\text{mora}}_{\beta,\tau}$ and the ensemble's original loss $\mathcal{L}^{\text{sce}}$. Introducing $\mathcal{L}^{\text{mora}}_{\beta,\tau}$ substantially improves the strength of attack. Note that in our comparison results (Tables 1, 2, 3, 4 and 6), instead of 5 random restarts we use a $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$ schedule to further improve the final attack success rate.

A.4 CIFAR-100

Table 6 compares the attacks on PDD+DEG [13] defenses trained on CIFAR-100. Similar to GAL [14] and TRS [31], PDD+DEG also diversifies sub-models gradients by minimizing cosine-similarities of gradients via regularization, and further diversifies feature selection with adaptive dropouts. We report attacks on ensembles with three ensemble-forming methods (softmax, voting and logits). Note that we only included PDD+DEG as other methods did not train on CIFAR-100.

Table 3: Comparing the accuracies of SOTA attacks and MORA on various defenses. Please refer to Table 1 for a detailed explanation. This table uses $\epsilon = 0.02$ as the ℓ^∞ perturbation bound, under which most defenses $(27/48)$ have close to nil robustness ($\leq 2\%$).

Defense Complexity	**#**	**Clean** 1	**Nominal** —	**PGD** 500	**CW** 500	**MORA** 500	**AA** 4.9k	**CAA** 1.8k	**MORA^{mt}** 1.4k	**Δ**
					Softmax					
ADP	3	92.88	12.53	0.07	0.23	**0.00**	**0.00**	0.03	**0.00**	12.53
	5	93.34	12.75	0.31	0.49	**0.00**	0.01	0.10	**0.00**	12.75
	8	93.48	12.61	3.04	1.68	**0.00**	0.02	0.47	**0.00**	12.61
Dverge	3	91.99	12.78	10.16	8.09	0.95	6.33	6.77	**0.78**	12.00
	5	92.38	22.36	20.33	18.88	5.31	10.12	16.13	**4.73**	17.63
	8	91.65	28.20	26.66	26.65	17.29	22.98	25.40	**16.94**	11.26
GAL	3	89.41	1.75	0.12	0.35	**0.00**	**0.00**	**0.00**	**0.00**	1.75
	5	90.93	6.80	4.81	4.22	0.64	2.87	2.77	**0.49**	6.31
	8	92.45	12.22	10.75	10.80	3.43	8.71	6.94	**2.97**	9.25
TRS†	3	70.02	2.24	1.49	0.80	0.20	0.33	0.86	**0.18**	2.06
	5	69.00	3.56	1.91	1.75	0.72	1.07	1.68	**0.64**	2.92
	8	73.01	4.18	1.71	1.77	1.14	1.50	1.72	**0.97**	3.21
					Voting					
ADP	3	91.84	22.18†	0.35	0.39	**0.00**	0.19	0.41	**0.00**	22.18
	5	93.13	21.76†	0.80	0.96	**0.00**	0.58	0.67	**0.00**	21.76
	8	93.28	15.19†	3.47	2.07	0.02	0.90	1.15	**0.00**	15.19
Dverge	3	91.72	4.70†	2.61	1.54	1.11	1.47	2.38	**0.89**	3.81
	5	92.18	9.30†	7.68	6.38	5.15	7.15	8.33	**4.84**	4.46
	8	91.58	18.15†	16.82	15.81	11.99	15.56	17.16	**11.47**	6.68
GAL	3	89.09	3.57†	0.23	0.11	**0.00**	**0.00**	**0.00**	**0.00**	3.57
	5	90.77	3.26†	1.95	1.83	0.54	1.64	2.10	**0.48**	2.78
	8	92.37	9.36†	7.26	6.85	2.34	4.91	5.33	**1.90**	7.46
TRS†	3	68.95	0.83	0.53	0.25	0.16	0.59	0.34	**0.12**	0.71
	5	68.31	1.34	0.92	0.83	0.60	0.72	0.91	**0.53**	0.81
	8	72.05	1.45	1.17	0.87	0.86	0.71	1.00	**0.68**	0.77
					Logits					
ADP	3	92.86	0.13†	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.13
	5	93.48	0.20†	**0.00**	0.04	**0.00**	**0.00**	**0.00**	**0.00**	0.20
	8	93.38	0.04†	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.04
Dverge	3	92.19	4.39†	4.12	4.24	3.27	3.31	3.40	**3.17**	1.22
	5	92.28	12.66†	12.24	12.56	10.83	10.93	10.99	**10.80**	1.86
	8	91.73	24.75†	24.41	25.42	23.32	23.39	23.49	**23.31**	1.54
GAL	3	89.50	1.10†	0.25	0.22	**0.00**	**0.00**	**0.00**	**0.00**	1.10
	5	90.93	3.98†	3.14	3.93	1.01	0.47	0.64	**0.45**	3.53
	8	92.54	11.18†	9.81	10.81	3.86	3.52	3.79	**3.41**	7.77
TRS†	3	69.72	0.73	0.64	0.71	0.48	0.50	0.52	**0.47**	0.26
	5	68.90	1.60	1.51	1.63	1.35	1.31	1.33	**1.30**	0.30
	8	72.24	1.78	1.69	1.78	1.52	1.47	1.46	**1.42**	0.36

A.5 Failure Modes in Ensemble Defenses

Figure 10 provides a visualization of the loss surfaces of ADP [19] under 3 different ensemble-forming strategies. In this section, we continue the discussion of the two failure modes in ensemble defense that induce overestimated robustness as introduced in Section 1:

(a) *Gradient obfuscation via ensemble-forming strategies.* It is evident that under PGD-10 attacks, both softmax- and voting-based ensembles (Figures 10a and 10b respectively) exhibit to some extent gradient obfuscation as they result in flatter loss surfaces in the adversarial direction $\boldsymbol{g}$, whereas the logits-based variant does not (Figure 10c).

(b) *Gradient diversification.* As sub-model gradients counteract, PGD attacks on softmax- and voting-based ensembles may result in an averaged gradient direction $\boldsymbol{g}$ that experience difficulty in

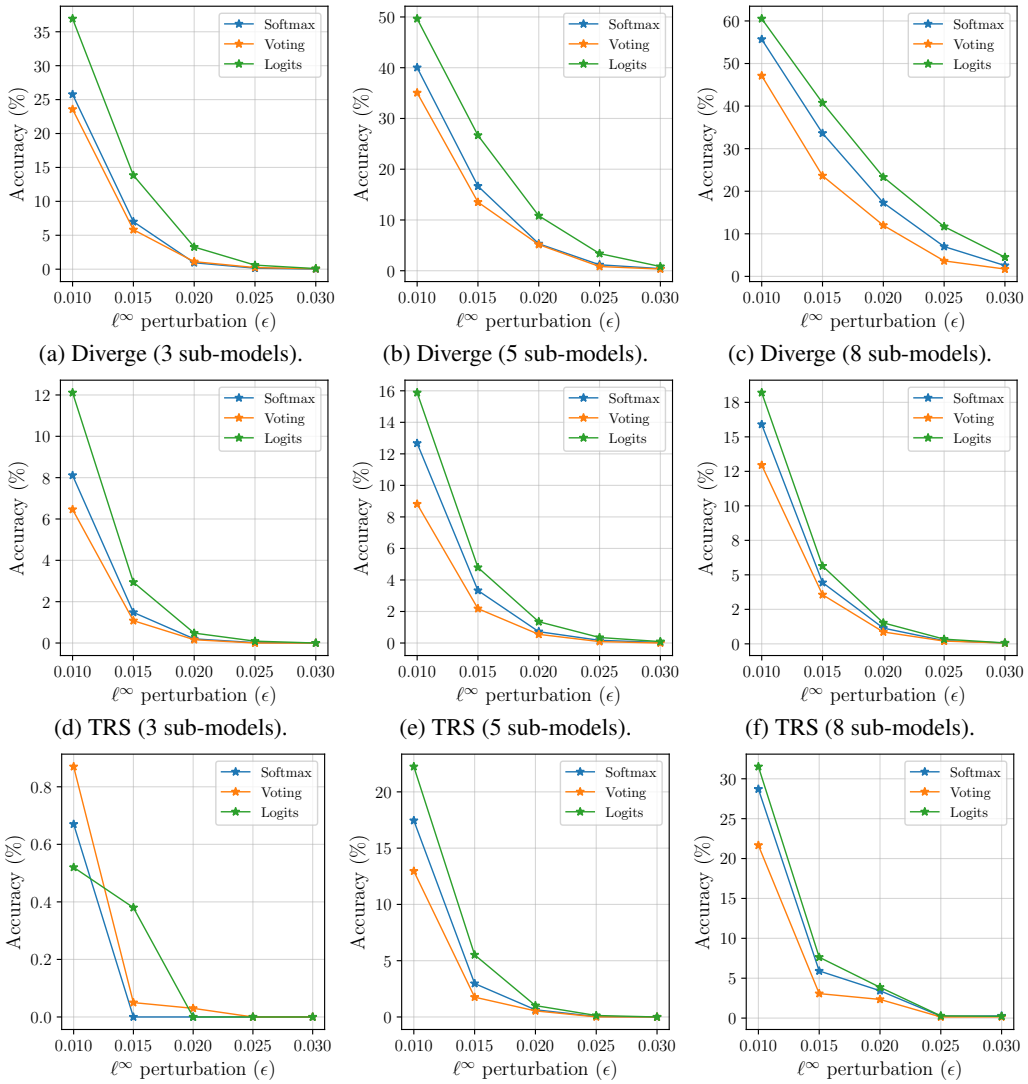
Table 4: Comparing the accuracies of SOTA attacks and MORA on various defenses. Please refer to Table 1 for a detailed explanation. This table uses $\epsilon = 0.03$ as the ℓ^∞ perturbation bound, under which almost all defenses exhibit 0% robustness, and all defenses fail to give 5% robustness.

Defense Complexity	**#**	**Clean** 1	**Nominal** —	**PGD** 500	**CW** 500	**MORA** 500	**AA** 4.9k	**CAA** 1.8k	**MORA^{mt}** 1.4k	**Δ**
					Softmax					
ADP	3	92.88	7.29	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	7.29
	5	93.34	8.52	0.04	0.03	**0.00**	**0.00**	**0.00**	**0.00**	8.52
	8	93.48	9.64	1.75	0.89	**0.00**	**0.00**	0.05	**0.00**	9.64
Dverge	3	91.99	2.91	1.43	1.47	0.02	1.59	1.10	**0.00**	2.91
	5	92.38	5.90	4.69	4.27	1.13	2.98	4.05	**0.23**	5.67
	8	91.65	9.17	8.28	8.46	2.54	6.84	8.21	**2.34**	6.83
GAL	3	89.41	1.26	0.01	0.06	**0.00**	**0.00**	**0.00**	**0.00**	1.26
	5	90.93	0.68	0.34	0.22	**0.00**	0.20	0.10	**0.00**	0.68
	8	92.45	1.18	0.92	0.88	0.27	1.19	0.79	**0.17**	1.01
TRS†	3	70.02	0.71	0.11	0.04	**0.00**	0.01	0.06	**0.00**	0.71
	5	69.00	2.24	0.18	0.17	0.04	**0.03**	0.14	**0.03**	2.21
	8	73.01	2.32	0.12	0.17	0.06	0.09	0.15	**0.02**	2.30
					Voting					
ADP	3	91.84	15.18†	0.03	0.09	**0.00**	0.02	0.03	**0.00**	15.18
	5	93.13	14.97†	0.33	0.17	**0.00**	0.03	0.04	**0.00**	14.97
	8	93.28	11.26†	1.91	1.01	**0.00**	0.09	0.17	**0.00**	11.26
Dverge	3	91.72	0.99†	0.19	0.07	0.06	0.05	0.10	**0.00**	0.99
	5	92.18	0.98†	0.55	0.44	0.33	0.50	0.71	**0.23**	0.75
	8	91.58	3.42†	2.72	2.27	1.72	2.30	2.92	**1.56**	1.86
GAL	3	89.09	1.90†	0.02	0.04	**0.00**	**0.00**	**0.00**	**0.00**	1.90
	5	90.77	0.27†	0.10	0.07	**0.00**	0.04	0.07	**0.00**	0.27
	8	92.37	0.66†	0.44	0.38	0.15	0.36	0.45	**0.13**	0.53
TRS†	3	68.95	0.03	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.03
	5	68.31	0.16	0.10	0.04	0.06	0.01	0.03	**0.00**	0.09
	8	72.05	0.15	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.15
					Logits					
ADP	3	92.86	0.01†	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.01
	5	93.48	0.05†	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05
	8	93.38	**0.00**†	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
Dverge	3	92.19	0.18†	0.15	0.15	**0.09**	**0.09**	**0.09**	**0.09**	0.09
	5	92.28	1.25†	1.13	1.13	0.83	**0.79**	0.83	**0.79**	0.46
	8	91.73	6.01†	5.77	5.95	4.51	4.54	4.63	**4.42**	1.59
GAL	3	89.50	0.09†	**0.00**	0.02	**0.00**	**0.00**	**0.00**	**0.00**	0.09
	5	90.93	0.24†	0.16	0.13	**0.00**	**0.00**	**0.00**	**0.00**	0.24
	8	92.54	0.66†	0.55	0.70	0.22	**0.17**	0.18	**0.17**	0.48
TRS†	3	69.72	0.01	0.01	0.02	**0.00**	0.01	0.01	**0.00**	0.01
	5	68.90	0.11	0.10	0.12	0.09	0.07	0.06	**0.04**	0.07
	8	72.24	0.13	0.12	0.15	0.07	0.08	0.07	**0.06**	0.07

increasing loss (Figures 10a and 10b respectively). Adopting sub-model reweighing (bottom row in Figure 10) alleviates this difficulty, and allows the attack to succeed more reliably.

B Limitations and Potential Societal Impacts

The ℓ^∞ white-box threat model assumes the availability of the models' gradients, which could present a challenge as such information may not be available to the attacker. It is thus critical to evaluate white-box robustness accurately, as it provides the lower bounds on the robustness of ensemble defenses in practical scenarios.



(a) Diverge (3 sub-models). (b) Diverge (5 sub-models). (c) Diverge (8 sub-models).

(d) TRS (3 sub-models). (e) TRS (5 sub-models). (f) TRS (8 sub-models).

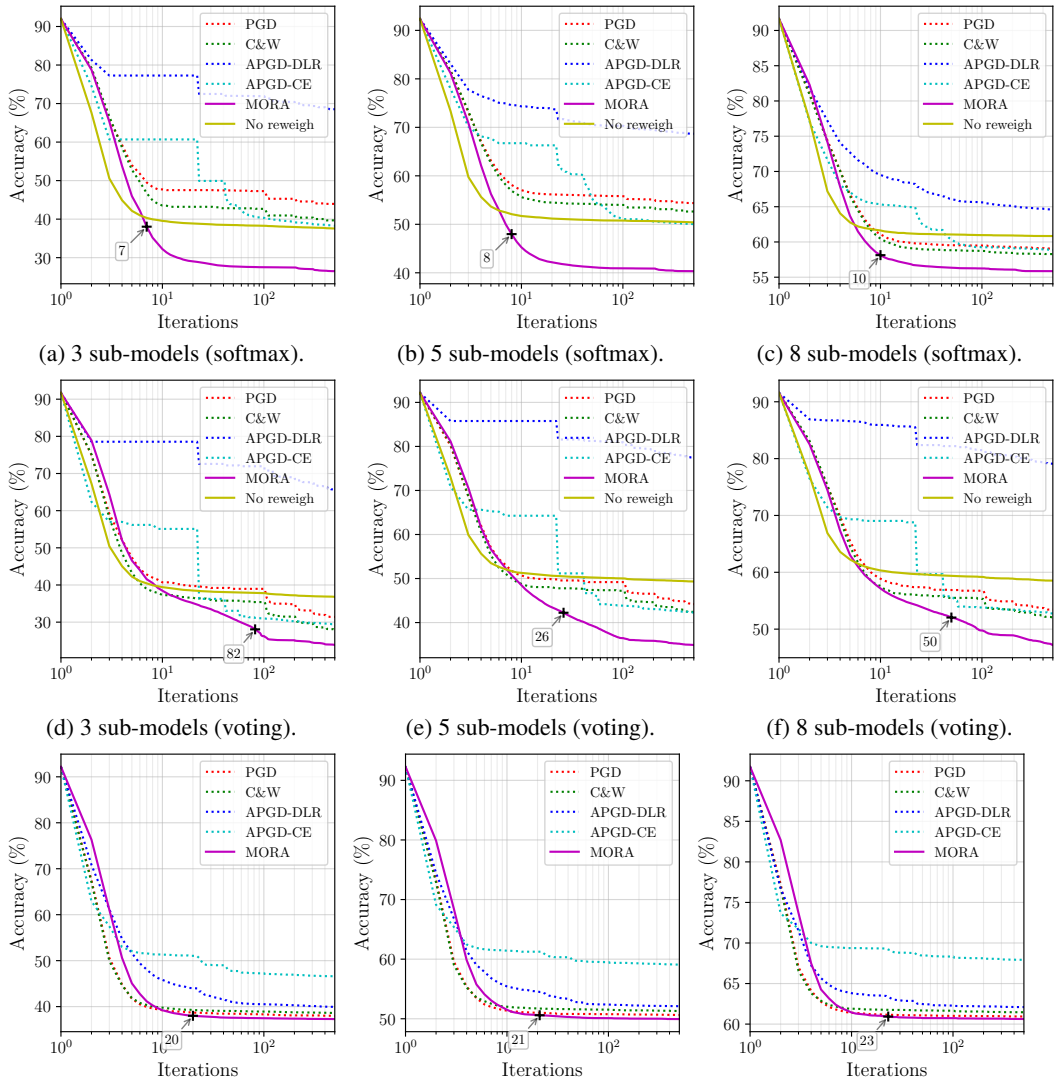
(g) GAL (3 sub-models). (h) GAL (5 sub-models). (i) GAL (8 sub-models).

Figure 5: Ensemble defenses with 5 sub-models *w.r.t.* the ϵ bound on ℓ^∞ perturbation. Contrary to existing literatures that propose "softmax"-based ensembles [14, 30, 31], we generally find "logits" to be the most robust option across the three ensemble-forming strategies. All defenses are evaluated with MORA (500 iterations). Note that the only exceptions to this rule, *i.e.*, GAL ensembles with 3 sub-models, exhibit very low robust accuracies.

We acknowledge that adversarial attacks may have the potential to be used by a malicious party, but we believe defending against such attacks is critically pertinent to the accurate evaluation of robustness. We hope this paper furthers the understanding of ensemble robustness, and accurately evaluating adversarial robustness can help improve future defenses. It is also noteworthy that the white-box threat model has applications in the context of advancements in deep learning, and can improve *e.g.*, transfer learning [28, 6], GAN training [2], interpretability [22], and *etc.*

C Computational Resources

On NVIDIA Tesla V100 GPUs, MORA with 500 iterations uses up to 1.0 GPU-hours on each ensemble defense, and MORA^{mt} uses up to 2.8 GPU-hours on the CIFAR-10 test set. The run time depends on the number of sub-models in an ensemble and its attack difficulty (Table 7).



(a) 3 sub-models (softmax). (b) 5 sub-models (softmax). (c) 8 sub-models (softmax).

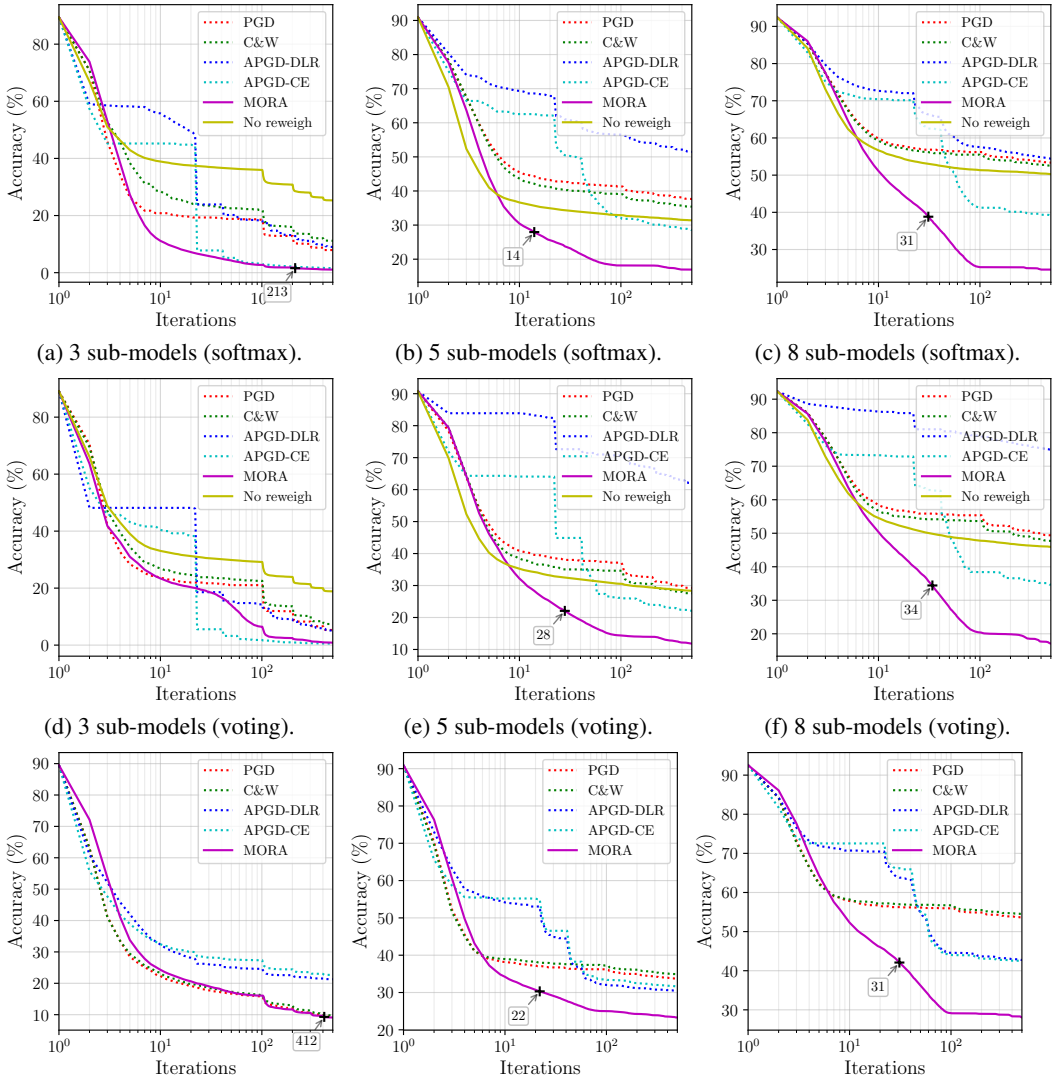
(d) 3 sub-models (voting). (e) 5 sub-models (voting). (f) 8 sub-models (voting).

(g) 3 sub-models (logits). (h) 5 sub-models (logits). (i) 8 sub-models (logits).

Figure 6: Comparing the convergence speed on Dverge [30] models with three ensemble-forming strategies (softmax, voting, logits) of MORA against a variant of MORA without sub-model reweighing, C&W, PGD and AA losses (APGD-{DLR,CE}). For logit-based ensembles (g, h, i), MORA and "No reweighing" are identical. The horizontal and vertical axes respectively show the iteration count used, and the percentage of unsuccessful images remaining. We annotated the number of iterations for MORA to overtake competition with 500 iterations.

D Licenses

Table 8 lists the relevant resources used in this paper and their respective licenses.

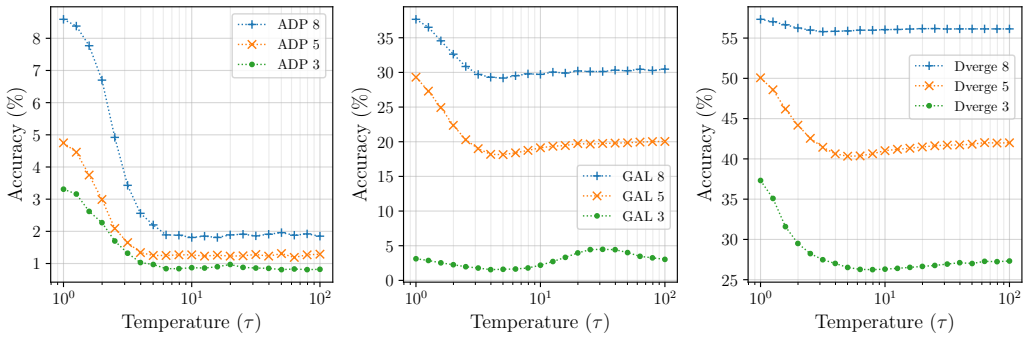


(a) 3 sub-models (softmax). (b) 5 sub-models (softmax). (c) 8 sub-models (softmax).

(d) 3 sub-models (voting). (e) 5 sub-models (voting). (f) 8 sub-models (voting).

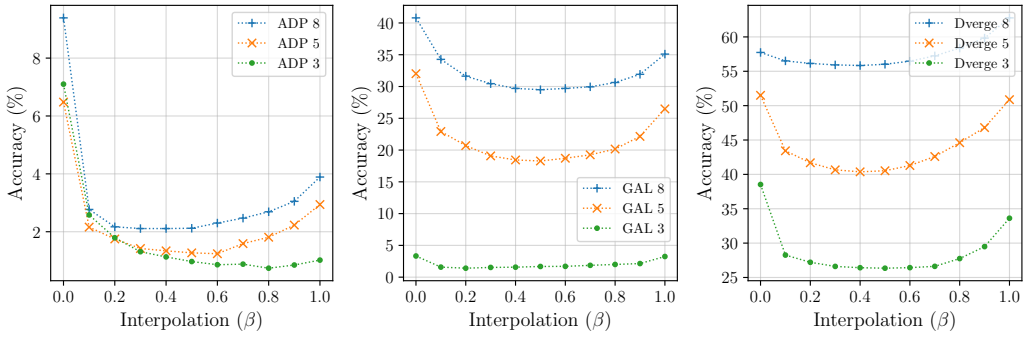
(g) 3 sub-models (logits). (h) 5 sub-models (logits). (i) 8 sub-models (logits).

Figure 7: Similar comparisons of convergence rates on GAL [14] models. Please refer to Figure 6 for a detailed description of the setup.



(a) ADP. (b) GAL. (c) Dverge.

Figure 8: Sensitivity analyses of the temperature coefficient τ as introduced in (10) and used in $\mathcal{L}^{\text{mora}}_{\beta,\tau}$. All ensembles use softmax values of sub-models to form outputs, as this is used by ADP [19], GAL [14] and Dverge [30]. All defending ensembles share similar flat regions of optimal τ with low sensitivity.



(a) ADP. (b) GAL. (c) Dverge.

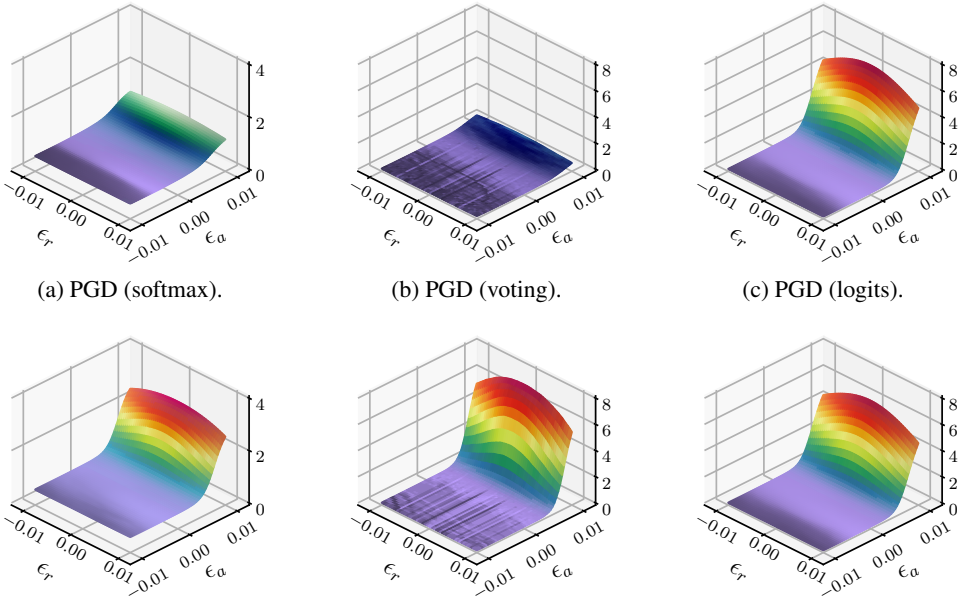
Figure 9: Sensitivity analyses of the β interpolation between the original ensemble loss $\mathcal{L}^{\text{sce}}$ and $\mathcal{L}^{\text{mora}}_{\beta,\tau}$, as introduced in (11). All ensembles use softmax values of sub-models to form outputs, as this is used by ADP [19], GAL [14] and Dverge [30]. All defending ensembles share similar flat regions of optimal β with low sensitivity.

Table 5: Ablation study of individual components used by MORA. "PGD" is the standard PGD attack [17] with 5 restarts each with 100 iterations. New components are added in consecutive rows. All rows use 500 iterations at most per image, except "Multiple Targets" which additionally uses 100 iterations each for the remaining 9 class labels.

Defenses	**ADP** [19]			**Dverge** [30]			**GAL** [14]			**TRS** [31]		
#	3	5	8	3	5	8	3	5	8	3	5	8
Softmax												
PGD	8.98	9.07	10.85	49.59	58.13	61.31	10.44	43.16	54.60	15.72	17.92	19.62
+ Momentum	5.98	7.10	9.22	44.49	54.61	59.13	8.13	37.59	53.39	14.01	15.91	18.02
+ Early Stop	5.04	6.57	8.89	40.18	53.53	58.84	3.70	35.34	50.53	13.21	15.63	17.97
+ Cosine Step Size	4.14	5.08	7.65	38.09	50.71	58.36	1.34	28.60	37.17	12.26	14.88	17.32
+ Sub-model Logits	0.69	1.13	1.84	28.51	42.91	56.95	1.78	20.73	30.51	9.15	13.52	16.80
+ Logit Normalization	0.66	0.91	1.56	27.42	41.02	55.78	2.39	19.13	29.71	9.21	13.49	16.78
+ Adaptive Reweighing	0.63	0.96	1.72	25.87	39.98	55.61	0.64	17.07	28.50	8.24	12.67	15.93
+ Multiple Targets	**0.34**	**0.67**	**1.32**	**25.26**	**39.50**	**55.57**	**0.51**	**16.05**	**27.44**	**7.60**	**12.47**	**15.64**
Voting												
PGD	13.31	13.87	13.72	36.06	49.59	57.32	7.23	34.71	51.37	13.38	15.56	17.06
+ Momentum	9.32	12.42	12.53	31.48	44.28	53.72	5.85	29.33	49.56	10.19	12.71	14.57
+ Early Stop	5.04	8.19	10.59	28.45	41.14	51.65	1.38	23.68	44.92	8.24	11.44	13.53
+ Cosine Step Size	4.54	7.55	9.17	28.14	40.85	51.43	0.55	19.21	29.36	8.07	10.94	12.87
+ Sub-model Logits	0.56	1.25	2.01	27.98	41.32	51.78	1.21	19.41	27.90	7.67	11.14	13.25
+ Logit Normalization	0.73	1.15	1.81	29.79	43.87	54.30	3.82	21.08	29.27	7.68	11.13	13.29
+ Adaptive Reweighing	0.59	1.11	2.51	23.47	34.94	47.24	0.85	12.88	21.62	5.66	8.82	11.36
+ Multiple Targets	**0.29**	**0.62**	**1.65**	**22.91**	**34.46**	**46.10**	**0.35**	**12.25**	**20.16**	**5.44**	**8.38**	**10.69**
Logits												
PGD	1.55	3.29	5.32	39.20	51.28	61.22	4.39	38.27	54.64	14.17	17.78	20.39
+ Momentum	0.87	1.97	3.57	39.20	50.57	60.95	10.01	33.97	53.67	13.06	16.65	19.20
+ Early Stop	0.82	1.77	3.27	37.91	50.53	60.89	8.12	31.24	50.68	12.96	16.51	19.13
+ Cosine Step Size	0.48	1.19	2.19	37.22	49.93	60.66	0.85	22.80	31.37	12.36	16.08	18.44
+ Sub-model Logits	0.52	1.21	2.24	37.22	49.95	60.65	0.81	22.79	31.25	12.36	16.05	18.48
+ Logit Normalization	0.45	1.15	2.12	36.90	49.63	60.53	0.35	22.21	31.28	12.35	16.09	18.48
+ Adaptive Reweighing	0.47	1.14	2.07	36.88	49.65	60.52	0.52	22.12	31.57	12.10	15.85	18.22
+ Multiple Targets	**0.21**	**0.89**	**1.93**	**36.84**	**49.59**	**60.49**	**0.03**	**19.40**	**30.66**	**12.07**	**15.82**	**18.17**

Table 6: Comparing the accuracies of SOTA attacks and MORA on PDD+DEG [13] defenses trained with CIFAR-100. Please refer to Table 1 for a detailed explanation. This table uses $\epsilon = 0.01$ as the ℓ^∞ perturbation bound, and evaluates its results on the CIFAR-100 test dataset.

PDD+DEG [13]	**#**	**Clean**	**Nominal**	**PGD**	**CW**	**MORA**	A^3	**AA**	**CAA**	**MORA^{mt}**	**Δ**
		1	—	500	500	500	12k	4.9k	1.8k	1.4k	
Softmax	3	79.30	22.77	11.32	14.19	1.33	9.44	9.19	11.46	**0.62**	22.15
Voting	3	79.28	6.55†	2.65	2.38	0.91	0.66	1.16	1.29	**0.55**	6.00
Logits	3	78.94	7.25†	3.29	1.30	0.90	0.63	0.63	0.67	**0.56**	6.69



(a) PGD (softmax). (b) PGD (voting). (c) PGD (logits).

(d) MORA (softmax). (e) MORA (voting). (f) MORA (logits).

Figure 10: The averaged loss surfaces across all samples that resisted PGD-10 attacks under all ensemble-forming modes of a 3 sub-model ensemble trained with ADP [19] in the image space $\mathbf{x} + \boldsymbol{g}\epsilon_{\mathrm{a}} + \boldsymbol{g}^{\perp}\epsilon_{\mathrm{r}}$. Here, $\boldsymbol{g}$ denotes the normalized adversarial direction after accumulating 10 initial iterations of gradient updates at the natural input $\mathbf{x}$, and $\boldsymbol{g}^{\perp}$ is its uniformly randomized orthogonal. The top row uses standard PGD attacks, and the bottom row then replaces the SCE loss used in the top row with the MORA loss and uses $\beta = 0.5$.

Table 7: Run time of MORA (up to 500 iterations) and MORA^{mt} (up to 1.4k iterations) for ADP, Dverge and GAL ensemble defenses on the CIFAR-10 test set. Easier ensembles consume less time with early stopping.

Run time (min)	#	MORA	MORA^{mt}
ADP [19]	3	2.0	3.7
	5	3.4	5.8
	8	6.5	11.4
Dverge [30]	3	12.3	31.1
	5	29.7	71.2
	8	61.9	169.3
GAL [14]	3	2.6	4.3
	5	15.4	35.2
	8	37.6	91.8

Table 8: Open-source resources used in this paper.

Name	**License**	**URL**
PyTorch	BSD	GitHub: pytorch/pytorch
Dverge	—	GitHub: zjysteven/DVERGE
TRS	—	GitHub: AI-secure/Transferability-Reduced-Smooth-Ensemble
CIFAR-10	—	`https://www.cs.toronto.edu/~kriz/cifar.html`
CIFAR-100	—	`https://www.cs.toronto.edu/~kriz/cifar.html`