INSTANCE ADAPTIVE ADVERSARIAL TRAINING: IMPROVED ACCURACY TRADEOFFS IN NEURAL NETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training is by far the most successful strategy for improving robustness of neural networks to adversarial attacks. Despite its success as a defense mechanism, adversarial training fails to generalize well to unperturbed test set. We hypothesize that this poor generalization is a consequence of adversarial training with uniform perturbation radius around every training sample. Samples close to decision boundary can be morphed into a different class under a small perturbation budget, and enforcing large margins around these samples produce poor decision boundaries that generalize poorly. Motivated by this hypothesis, we propose instance adaptive adversarial training – a technique that enforces sample-specific perturbation margins around every training sample. We show that using our approach, test accuracy on unperturbed samples improve with a marginal drop in robustness. Extensive experiments on CIFAR-10, CIFAR-100 and Imagenet datasets demonstrate the effectiveness of our proposed approach.

1 INTRODUCTION

A key challenge when deploying neural networks in safety-critical applications is their poor stability to input perturbations. Extremely tiny perturbations to network inputs may be imperceptible to the human eye, and yet cause major changes to outputs. One of the most effective and widely used methods for hardening networks to small perturbations is "adversarial training" (Madry et al., 2018), in which a network is trained using adversarially perturbed samples with a fixed perturbation size. By doing so, adversarial training typically tries to enforce that the output of a neural network remains nearly constant within an ℓ_p ball of every training input.

Despite its ability to increase robustness, adversarial training suffers from poor accuracy on clean (natural) test inputs. The drop in clean accuracy can be as high as 10% on CIFAR-10, and 15% on Imagenet (Madry et al., 2018; Xie et al., 2019), making robust models undesirable in some industrial settings. The consistently poor performance of robust models on clean data has lead to the line of thought that there may be a fundamental trade-off between robustness and accuracy (Zhang et al., 2019; Tsipras et al., 2019), and recent theoretical results characterized this tradeoff (Fawzi et al., 2018; Shafahi et al., 2018; Mahloujifar et al., 2019).

In this work, we aim to understand and optimize the tradeoff between robustness and clean accuracy. More concretely, our objective is to improve the clean accuracy of adversarial training for a chosen level of adversarial robustness. Our method is inspired by the observation that the constraints enforced by adversarial training are *infeasible*; for commonly used values of ϵ, it is not possible to achieve label consistency within an ϵ-ball of each input image because the balls around images of different classes overlap. This is illustrated on the left of Figure 1, which shows that the ϵ-ball around a "bird" (from the CIFAR-10 training set) contains images of class "deer" (that do not appear in the training set). If adversarial training were successful at enforcing label stability in an $\epsilon = 8$ ball around the "bird" training image, doing so would come at the *unavoidable* cost of misclassifying the nearby "deer" images that come along at test time. At the same time, when training images lie far from the decision boundary (eg., the deer image on the right in Fig 1), it is possible to enforce stability with large ϵ with no compromise in clean accuracy. When adversarial training on CIFAR-10, we see that $\epsilon = 8$ is too large for some images, causing accuracy loss, while being unnecessarily small for others, leading to sub-optimal robustness.

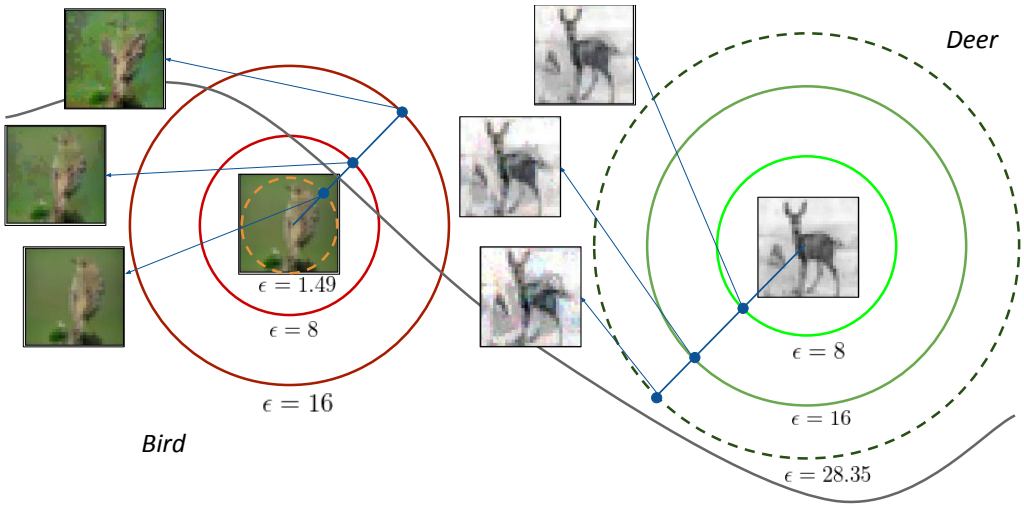


Figure 1: Overview of instance adaptive adversarial training. Samples close to the decision boundary (bird on the left) have nearby samples from a different class (deer) within a small L_p ball, making the constraints imposed by PGD-8 / PGD-16 adversarial training infeasible. Samples far from the decision boundary (deer on the right) can withstand large perturbations well beyond $\epsilon = 8$. Our adaptive adversarial training correctly assigns the perturbation radius (shown in dotted line) so that samples within each L_p ball maintain the same class.

The above observation naturally motivates adversarial training with *instance adaptive* perturbation radii that are customized to each training image. By choosing larger robustness radii at locations where class manifolds are far apart, and smaller radii at locations where class manifolds are close together, we get high adversarial robustness where possible while minimizing the clean accuracy loss that comes from enforcing overly-stringent constraints on images that lie near class boundaries. As a result, instance adaptive training significantly improves the tradeoff between accuracy and robustness, breaking through the pareto frontier achieved by standard adversarial training. Additionally, we show that the learned instance-specific perturbation radii are interpretable; samples with small radii are often ambiguous and have nearby images of another class, while images with large radii have unambiguous class labels that are difficult to manipulate.

Parallel to our work, we found that Ding et al. (2018) uses adaptive margins in a max-margin framework for adversarial training. Their work focuses on improving the adversarial robustness, which differs from our goal of understanding and improving the robustness-accuracy tradeoff. Moreover, our algorithm for choosing adaptive margins significantly differs from that of Ding et al. (2018).

2 Background

Adversarial attacks are data items containing small perturbations that cause misclassification in neural network classifiers (Szegedy et al., 2014). Popular methods for crafting attacks include the fast gradient sign method (FGSM) (Goodfellow et al., 2015) which is a one-step gradient attack, projected gradient descent (PGD) (Madry et al., 2018) which is a multi-step extension of FGSM, the C/W attack (Carlini & Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), and many more. All these methods use the gradient of the loss function with respect to inputs to construct additive perturbations with a norm-constraint. Alternative attack metrics include spatial transformer attacks (Xiao et al., 2018), attacks based on Wasserstein distance in pixel space (Wong et al., 2019), etc.

Defending against adversarial attacks is a crucial problem in machine learning. Many early defenses (Buckman et al., 2018; Samangouei et al., 2018; Dhillon et al., 2018), were broken by strong attacks. Fortunately, *adversarially training* is one defense strategy that remains fairly resistant to most existing attacks.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the set of training samples in the input dataset. In this paper, we focus on classification problems, hence, $y_i \in \{1, 2, \dots N_c\}$, where N_c denotes the number of classes. Let $f_\theta(\mathbf{x}) : \mathbb{R}^{c \times m \times n} \to \mathbb{R}^{N_c}$ denote a neural network model parameterized by θ. Classifiers are often trained by minimizing the cross entropy loss given by

$$\min_\theta \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i)) \big]$$

where $\tilde{\mathbf{y}}_i$ is the one-hot vector corresponding to the label y_i. In adversarial training, instead of optimizing the neural network over the clean training set, we use the adversarially perturbed training set. Mathematically, this can be written as the following *min-max* problem

$$\min_\theta \max_{\|\delta_i\|_\infty \leq \epsilon} \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i) + \delta_i) \big] \tag{1}$$

This problem is solved by an alternating stochastic method that takes minimization steps for θ, followed by maximization steps that approximately solve the inner problem using k steps of PGD. For more details, refer to Madry et al. (2018).

Algorithm 1 Adaptive adversarial training algorithm

Require: N_{iter}: Number of training iterations, N_{warm}: Warmup period
Require: $PGD_k(\mathbf{x}, y, \epsilon)$: Function to generate PGD-k adversarial samples with ϵ norm-bound
Require: ϵ_w: ϵ used in warmup
for t in $1 : N_{iter}$ **do**
  Sample a batch of training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{batch}} \sim \mathcal{D}$
  if $t < N_{warm}$ **then**
    $\epsilon_i = \epsilon_w$
  else
    Choose ϵ_i using Alg 2
  end if
  $\mathbf{x}_i^{adv} = PGD(\mathbf{x}_i, y_i, \epsilon_i)$
  $S_+ = \{i | f(\mathbf{x}_i)$ is correctly classified as $y_i\}$
  $S_- = \{i | f(\mathbf{x}_i)$ is incorrectly classified as $y_i\}$
  $\min_\theta \frac{1}{N_{batch}} \Big[\sum_{i \in S_+} L_{cls}(\mathbf{x}_i^{adv}, y^i) + \sum_{i \in S_-} L_{cls}(\mathbf{x}_i, y_i) \Big]$
end for

3 Instance Adaptive Adversarial Training

To remedy the shortcomings of uniform perturbation radius in adversarial training (Section 1), we propose *Instance Adaptive Adversarial Training* (IAAT), which solves the following optimization:

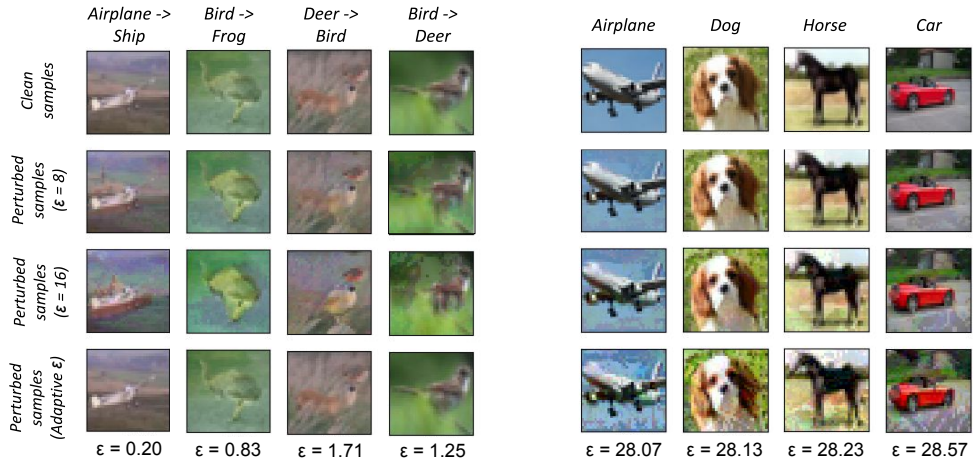
$$\min_\theta \max_{\|\delta_i\|_\infty < \epsilon_i} \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i) + \delta_i) \big] \tag{2}$$

Like vanilla adversarial training, we solve this by sampling mini-batches of images $\{\mathbf{x}_i\}$, crafting adversarial perturbations $\{\delta_i\}$ of size at most $\{\epsilon_i\}$, and then updating the network model using the perturbed images.

The proposed algorithm is distinctive in that it uses a different ϵ_i for each image $\mathbf{x}_i$. Ideally, we would choose each ϵ_i to be as large as possible without finding images of a different class within the ϵ_i-ball around $\mathbf{x}_i$. Since we have no a-priori knowledge of what this radius is, we use a simple heuristic to update ϵ_i after each epoch. After crafting a perturbation for $\mathbf{x}_i$, we check if the perturbed image was a successful adversarial example. If PGD succeeded in finding an image with a different class label, then ϵ_i is too big, so we replace $\epsilon_i \leftarrow \epsilon_i - \gamma$. If PGD failed, then we set $\epsilon_i \leftarrow \epsilon_i + \gamma$.

Since the network is randomly initialized at the start of training, random predictions are made, and this causes $\{\epsilon_i\}$ to shrink rapidly. For this reason, we begin with a warmup period of a few (usually 10 epochs for CIFAR-10/100) epochs where adversarial training is performed using uniform ϵ for every sample. After the warmup period ends, we perform instance adaptive adversarial training.

A detailed training algorithm is provided in Alg. 1.



(a) Samples from bottom 1% ϵ

(b) Samples from top 1% ϵ

Figure 2: Visualizing training samples and their perturbations. The left panel shows samples that are assigned small ϵ (displayed below images) during adaptive training. These images are close to class boundaries, and change class when perturbed with $\epsilon \geq 8$. The right panel show images that are assigned large ϵ. These lie far from the decision boundary, and retain class information even with very large perturbations. All ϵ live in the range $[0, 255]$

Algorithm 2 ϵ selection algorithm

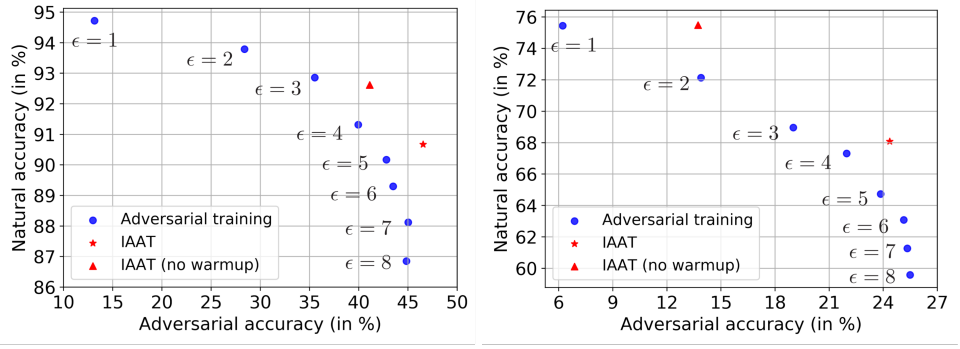
Require: i: Sample index, j: Epoch index
Require: β: Smoothing constant, γ: Discretization for ϵ search.
Set $\epsilon_1 = \boldsymbol{\epsilon}_{mem}[j-1, i] + \gamma$
Set $\epsilon_2 = \boldsymbol{\epsilon}_{mem}[j-1, i]$
Set $\epsilon_3 = \boldsymbol{\epsilon}_{mem}[j-1, i] - \gamma$
if $f_\theta(PGD_k(\mathbf{x}_i, y_i, \epsilon_1))$ predicts as y_i **then**
    Set $\epsilon_i = \epsilon_1$
else if $f_\theta(PGD_k(\mathbf{x}_i, y_i, \epsilon_2))$ predicts as y_i **then**
    Set $\epsilon_i = \epsilon_2$
else
    Set $\epsilon_i = \epsilon_3$
end if
$\epsilon_i \leftarrow (1-\beta)\boldsymbol{\epsilon}_{mem}[j-1, i] + \beta\epsilon_i$
Update $\boldsymbol{\epsilon}_{mem}[j, i] \leftarrow \epsilon_i$
Return ϵ_i

4 Experiments

To evaluate the robustness and generalization of our models, we report the following metrics: (1) test accuracy of unperturbed (natural) test samples, (2) adversarial accuracy of white-box PGD attacks, (3) adversarial accuracy of transfer attacks and (4) accuracy of test samples under common image corruptions (Hendrycks & Dietterich, 2019). Following the protocol introduced in Hendrycks & Dietterich (2019), we do not train our models on any image corruptions.

4.1 CIFAR

On CIFAR-10 and CIFAR-100 datasets, we perform experiments on Resnet-18 and WideRenset-32-10 models following (Madry et al., 2018; Zhang et al., 2019). All models are trained on PGD-10 attacks i.e., 10 steps of PGD iterations are used for crafting adversarial attacks during training. In the whitebox setting, models are evaluated on: (1) PGD-10 attacks with 5 random restarts, (2) PGD-100 attacks with 5 random restarts, and (3) PGD-1000 attacks with 2 random restarts. For



(a) CIFAR-10 (b) CIFAR-100

Figure 3: Tradeoffs between accuracy and robustness: Each blue dot denotes an adversarially trained model with a different ϵ used at training (Training ϵ is marked next to blue dots). Models trained using instance adaptive adversarial training are shown in red. Adaptive training breaks through the Pareto frontier achieved by plain adversarial training with a fixed ϵ. For all models, adversarial accuracy is reported on PGD-1000 attacks with a fixed test $\epsilon = 8$.

transfer attacks, an independent copy of the model is trained using the same training algorithm and hyper-parameter settings, and PGD-1000 adversarial attacks with 2 random restarts are crafted on the surrogate model. For image corruptions, following (Hendrycks & Dietterich, 2019), we report average classification accuracy on 19 image corruptions.

Beating the robustness-accuracy tradeoff: In adversarial training, the perturbation radius ϵ is a hyper-parameter. Training models with varying ϵ produces a robustness-accuracy tradeoff curve - models with small training ϵ achieve better natural accuracy and poor adversarial robustness, while models trained on large ϵ have improved robustness and poor natural accuracy. To generate this tradeoff, we perform adversarial training with ϵ in the range $\{1, 2, \dots 8\}$. Instance adaptive adversarial training is then compared with respect to this tradeoff curve in Fig. 3a, 3b. Two versions of IAAT are reported - with and without a warmup phase. In both versions, we clearly achieve an improvement over the accuracy-robustness tradeoff. Use of the warmup phase helps retain robustness with a drop in natural accuracy compared to its no-warmup counterpart.

Clean accuracy improves for a fixed level of robustness: On CIFAR-10, as shown in Table. 1, we observe that our instance adaptive adversarial training algorithm achieves similar adversarial robustness as the adversarial training baseline. However, the accuracy on clean test samples increases by 4.06% for Resnet-18 and 4.49% for WideResnet-32-10. We also observe that the adaptive training algorithm improves robustness to unseen image corruptions. This points to an improvement in overall generalization ability of the network. On CIFAR-100 (Table. 2), the performance gain in natural test accuracy further increases - 8.79% for Resnet-18, and 9.22% for Wideresnet-32-10. The adversarial robustness drop is marginal.

Maintaining performance over a range of test ϵ: Next, we plot adversarial robustness over a sweep of ϵ values used to craft attacks at test time. Fig. 4a, 4b shows an adversarial training baseline with $\epsilon = 8$ performs well at high ϵ regimes and poorly at low ϵ regimes. On the other hand, adversarial training with $\epsilon = 2$ has a reverse effect, performing well at low ϵ and poorly at high ϵ regimes. Our instance adaptive training algorithm maintains good performance over all ϵ regimes, achieving slightly less performance than the $\epsilon = 2$ model for small test ϵ, and dominating all models for larger test ϵ.

Interpretability of ϵ: We find that the values of ϵ_i chosen by our adaptive algorithm correlate well with our own human concept of class ambiguity. Figure 2 (and Figure 6 in Appendix B) shows that a sampling of images that receive small ϵ_i contains many ambiguous images, and these images are perturbed into a (visually) different class using $\epsilon = 16$. In contrast, images that receive a large ϵ_i have a visually definite class, and are not substantially altered by an $\epsilon = 16$ perturbation.

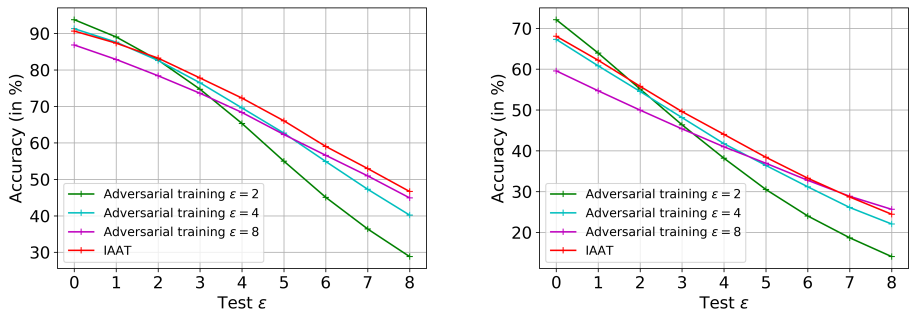
Table 1: Robustness experiments on CIFAR-10. PGD attacks are generated with $\epsilon = 8$. PGD_{10} and PGD_{100} attacks are generated with 5 random restarts, while PGD_{1000} attacks are generated with 2 random restarts

Method	Natural acc. (in %)	Whitebox acc. (in %) PGD_{10}	PGD_{100}	PGD_{1000}	Transfer (in %) acc. (PGD_{1000})	Corruption acc. (in %)
Resnet-18						
Clean	94.21	0.02	0.00	0.00	3.03	72.71
Adversarial	83.20	43.79	42.30	42.36	59.80	73.73
IAAT	87.26	43.08	41.16	41.16	59.87	78.82
WideResnet 32-10						
Clean	95.50	0.05	0.00	0.00	5.02	78.35
Adversarial	86.85	46.86	44.82	44.84	62.77	77.99
IAAT	91.34	48.53	46.50	46.54	58.20	83.13

Table 2: Robustness experiments on CIFAR-100. PGD attacks are generated with $\epsilon = 8$. PGD_{10} and PGD_{100} attacks are generated with 5 random restarts. PGD_{1000} attacks are generated with 2 random restarts

Method	Natural acc. (in %)	Whitebox acc. (in %) PGD_{10}	PGD_{100}	PGD_{1000}	Transfer acc. (in %) PGD_{1000}
Resnet-18					
Clean	74.88	0.02	0.00	0.01	1.81
Adversarial	55.11	20.69	19.68	19.91	35.57
IAAT	63.90	18.50	17.10	17.11	35.74
WideResnet 32-10					
Clean	79.91	0.01	0.00	0.00	1.20
Adversarial	59.58	26.24	25.47	25.49	38.10
IAAT	68.80	26.17	24.22	24.36	35.18

Robustness to other attacks: While our instance adaptive algorithm is trained on PGD attacks, we are interested to see if the trained model improves robustness on other adversarial attacks. As shown in Table. 3, IAAT achieves similar level of robustness as adversarial training on other gradient-based attacks, while improving the natural accuracy.



(a) CIFAR-10 (b) CIFAR-100

Figure 4: Plot of adversarial robustness over a sweep of test ϵ

Table 3: Robustness results on other attacks for models trained using PGD-10 for WRN-32-10 model on CIFAR-10 dataset. Accuracies are reported in $\%$

Algorithm	Natural acc.	PGD-1000	DeepFool	MIFGSM	CW40
Adversarial training	86.85	44.84	65.28	54.66	55.62
IAAT	91.34	46.54	66.58	53.99	56.80

Table 4: Robustness experiments on Imagenet. All adversarial attacks are generated with PGD-1000. (↑) indicates higher numbers are better, while (↓) indicates lower numbers are better

Method	Natural acc. (in $\%$) (↑)	Whitebox acc. (in $\%$) (↑)				Corruption mCE (↓)
		$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 12$	$\epsilon = 16$	
Resnet-50						
Clean training	75.80	0.64	0.18	0.00	0.00	76.69
Adversarial training	50.99	50.89	49.11	44.71	35.82	95.48
IAAT	62.71	61.52	54.63	39.90	22.72	85.21
Resnet-101						
Clean training	77.10	0.83	0.12	0.00	0.00	70.37
Adversarial training	55.42	55.11	53.07	48.35	39.08	91.45
IAAT	65.29	63.83	56.62	41.51	23.91	79.52
Resnet-152						
Clean training	77.60	0.57	0.08	0.00	0.00	69.27
Adversarial training	57.26	56.77	54.75	49.86	40.40	89.31
IAAT	67.44	65.97	59.28	45.01	27.85	78.53

4.2 IMAGENET

Following the protocol introduced in Xie et al. (2019), we attack Imagenet models using random targeted attacks instead of untargeted attacks as done in previous experiments. During training, adversarial attacks are generated using 30 steps of PGD. As a baseline, we use adversarial training with a fixed ϵ of $16/255$. This is the setting used in Xie et al. (2019). Adversarial training on Imagenet is computationally intensive. To make training practical, we use distributed training with synchronized SGD on $64/128$ GPUs. More implementation details can be found in Appendix E.

At test time, we evaluate the models on clean test samples and on whitebox adversarial attacks with $\epsilon = \{4, 8, 12, 16\}$. PGD-1000 attacks are used. Additionally, we also report normalized mean corruption error (mCE), an evaluation metric introduced in Hendrycks & Dietterich (2019) to test the robustness of neural networks to image corruptions. This metric reports mean classification error of different image corruptions averaged over varying levels of degradation. Note that while accuracies are reported for natural and adversarial robustness, mCE reports classification errors, so lower numbers are better.

Our experimental results are reported in Table. 4. We observe a huge drop in natural accuracy for adversarial training (25%, 22% and 20% drop for Resnet-50, 101 and 152 respectively). Adaptive adversarial training significantly improves the natural accuracy - we obtain a consistent performance gain of $10+\%$ on all three models over the adversarial training baseline. On whitebox attacks, IAAT outperforms the adversarial training baseline on low ϵ regimes, however a drop of 13% is observed at high ϵ's ($\epsilon = 16$). On the corruption dataset, our model consistently outperforms adversarial training.

5 ABLATION EXPERIMENTS

5.1 EFFECT OF WARMUP

Recall from Section 3 that during warmup, adversarial training is performed with uniform norm-bound constraints. Once the warmup phase ends, we switch to instance adaptive training. From

Table 5: Ablation: Effect of warmup on CIFAR-10

Method	Natural acc. (%)	Whitebox acc. (in %) PGD_{10}	PGD_{100}	PGD_{1000}	Transfer acc.(%) PGD_{1000}	Corruption acc. (in %)
Resnet-18						
IAAT (no warm)	89.62	40.55	38.15	38.08	58.89	81.10
IAAT (warm)	87.26	43.08	41.16	41.16	59.87	78.82
WideResnet 32-10						
IAAT (no warm)	92.62	45.12	41.08	41.11	53.08	84.92
IAAT (warm)	90.67	48.53	46.50	46.54	58.20	83.13

Table 6: Ablation: Effect of warmup on CIFAR-100

Method	Natural acc. (in %)	Whitebox acc. (in %) PGD_{10}	PGD_{100}	PGD_{1000}	Transfer acc.(%) PGD_{1000}
Resnet-18					
Adaptive (no warm)	68.34	14.76	13.29	13.30	32.39
Adaptive (warm)	63.90	18.50	17.10	17.11	35.74
WideResnet 32-10					
Adaptive (no warm)	75.48	18.14	13.78	13.71	24.00
Adaptive (warm)	68.80	26.17	24.22	24.36	35.18

Table 7: Ablation: Comparison of IAAT with exact line search. Accuracies are reported in % for Resnet-18 model trained on CIFAR-10 dataset.

Algorithm	Natural acc.	PGD-10	PGD-1000
Full line search	88.67	43.26	41.37
IAAT	87.26	43.08	41.16

Table 5 and 6, we observe that when warmup is used, adversarial robustness improves with a small drop in natural accuracy, with more improvements observed in CIFAR-100. However, as shown in Fig. 3a and 3b, both these settings improve the accuracy-robustness tradeoff.

5.2 Other heuristics

We are interested in estimating instance-specific perturbation radius ϵ_i such that predictions are consistent within the chosen ϵ_i-ball. To obtain an exact estimate of such an ϵ_i, we can perform a line search as follows: Given a discretization η and a maximum perturbation radius ϵ_{max}, generate PGD attacks with radii $\{i\eta\}_{i=1}^{\epsilon_{max}/\eta}$. Choose the desired ϵ_i as the maximum $i\eta$ for which the prediction remains consistent as that of the ground-truth label. We compare the performance of exact line search with that of IAAT in Table 7. We observe that exact line search marginally improves compared to IAAT. However, exact line search is computationally expensive as it requires performing ϵ_{max}/η additional PGD computations, whereas IAAT requires only 2.

6 Conclusion

In this work, we focus on improving the robustness-accuracy tradeoff in adversarial training. We first show that realizable robustness is a sample-specific attribute: samples close to the decision boundary can only achieve robustness within a small ϵ ball, as they contain samples from a different class beyond this radius. On the other hand samples far from the decision boundary can be robust on a relatively large perturbation radius. Motivated by this observation, we develop *instance adaptive adversarial training*, in which label consistency constraints are imposed within sample-specific perturbation radii, which are in-turn estimated. Our proposed algorithm has empirically been shown to improve the robustness-accuracy tradeoff in CIFAR-10, CIFAR-100 and Imagenet datasets.

REFERENCES

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S18Su--CW.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, 2017.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1uR4GZRZ.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems*, pp. 1178–1187, 2018.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing accuracy. *CoRR*, abs/1906.06784, 2019.

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. URL https://openreview.net/forum?id=rJzIBfZAb.

Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4536–4543, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2574–2582, 2016.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected Sinkhorn iterations. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 6808–6817. PMLR, 2019.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/zhang19p.html.

Table 8: Comparison with Mixup

Method	Natural acc. (in %)	Whitebox acc. (in %)			Transfer attack (in %)
		PGD_{10}	PGD_{100}	PGD_{1000}	PGD_{1000}
Resnet-18					
Mixup	89.47	42.60	38.42	38.49	59.48
IAAT	87.26	43.08	41.16	41.16	59.87
WideResnet 32-10					
Mixup	92.57	45.01	36.6	36.44	63.57
IAAT	90.67	48.53	46.50	46.54	58.20

A Appendix

A.1 Comparison with Mixup

A recent paper that addresses the problem of improving natural accuracy in adversarial training is mixup adversarial training (Lamb et al., 2019), where adversarially trained models are optimized using mixup loss instead of the standard cross-entropy loss. In this paper, natural accuracy was shown to improve with no drop in adversarial robustness. However, the robustness experiments were not evaluated on strong attacks (experiments were reported only on PGD-20). We compare our implementation of mixup adversarial training with IAAT on stronger attacks in Table. 8. We observe that while natural accuracy improves for mixup, drop in adversarial accuracy is much higher than IAAT.

B Sample visualization

A visualization of samples from CIFAR-10 dataset with the corresponding ϵ value assigned by IAAT is shown in Figure. 5. We observe that samples for which low ϵ's are assigned are visually confusing (eg., top row of Figure. 5), while samples with high ϵ distinctively belong to one class.

In addition, we also show more visualizations of samples near decision boundary which contain samples from a different class within a fixed ℓ_∞ ball in Figure. 6. The infeasibility of label consistency constraints within the commonly used perturbation radius of $\ell_\infty = 8$ is apparent in this visualization. Our algorithm effectively chooses an appropriate ϵ that retains label information within the chosen radius.

B.1 Visualizing ϵ progress

Next, we visualize the evolution of ϵ over epochs in adaptive adversarial training. A plot showing the average ϵ growth, along with the ϵ progress of 3 randomly picked samples are shown in Fig. 7a and 7b. We observe that average ϵ converges to around 11, which is higher than the default setting of $\epsilon = 8$ used in adversarial training. Also, each sample has a different ϵ profile - for some, ϵ increases well beyond the commonly use radius of $\epsilon = 8$, while for others, it converges below it. In addition, a plot showing the histogram of ϵ's at different snapshots of training is shown in Fig. 8. We observe an increase in spread of the histogram as the training progresses.

C ImageNet sweep over PGD iterations

Testing against a strong adversary is crucial to assess the true robustness of a model. A popular practice in adversarial robustness community is to attack models using PGD with many attack iterations (Xie et al., 2019). So, we test our instance adaptive adversarially trained models on a sweep of PGD iterations for a fixed ϵ level. Following (Xie et al., 2019), we perform the sweep upto 2000 attack steps fixing $\epsilon = 16$. The resulting plot is shown in Figure. 9. For all three Resnet models, we observe a saturation in adversarial robustness beyond 500 attack iterations.

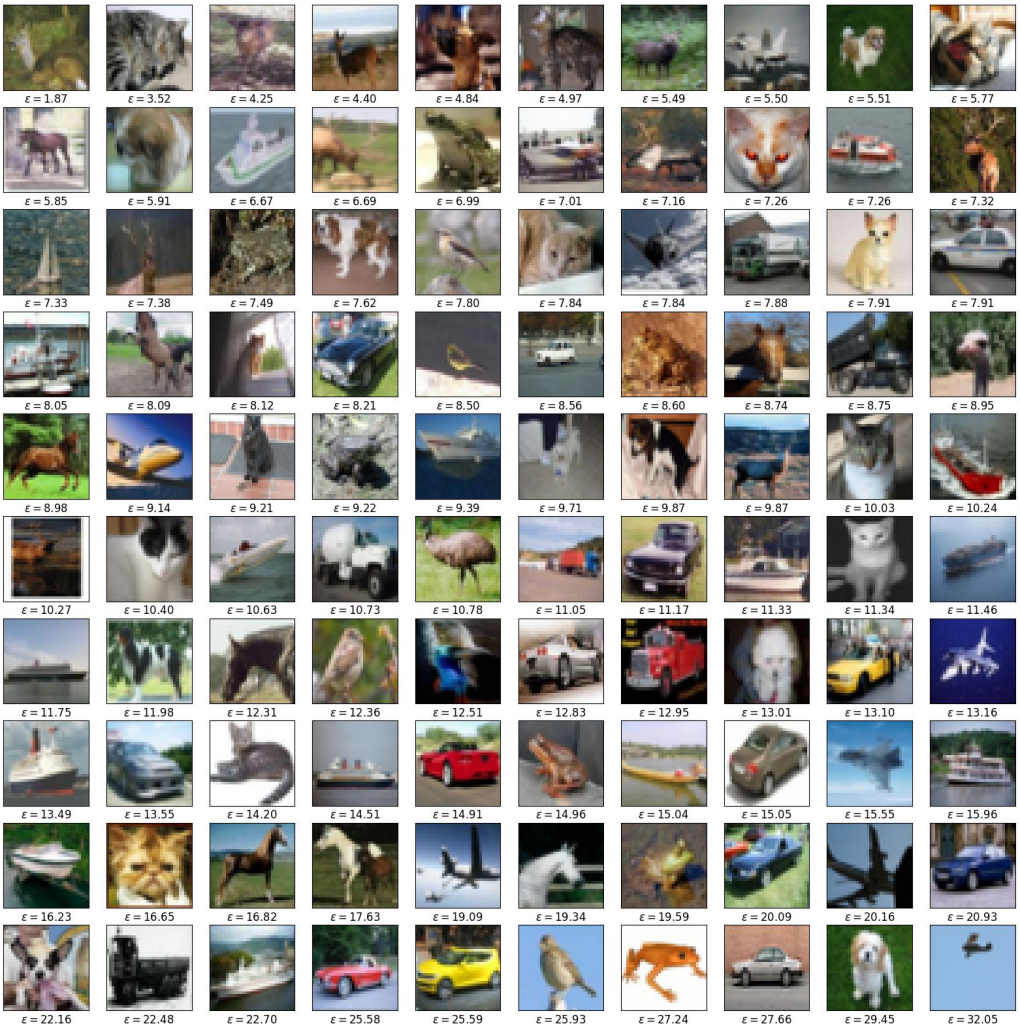


Figure 5: Visualizing training samples with their corresponding perturbation. All ϵ live in the range $[0, 255]$

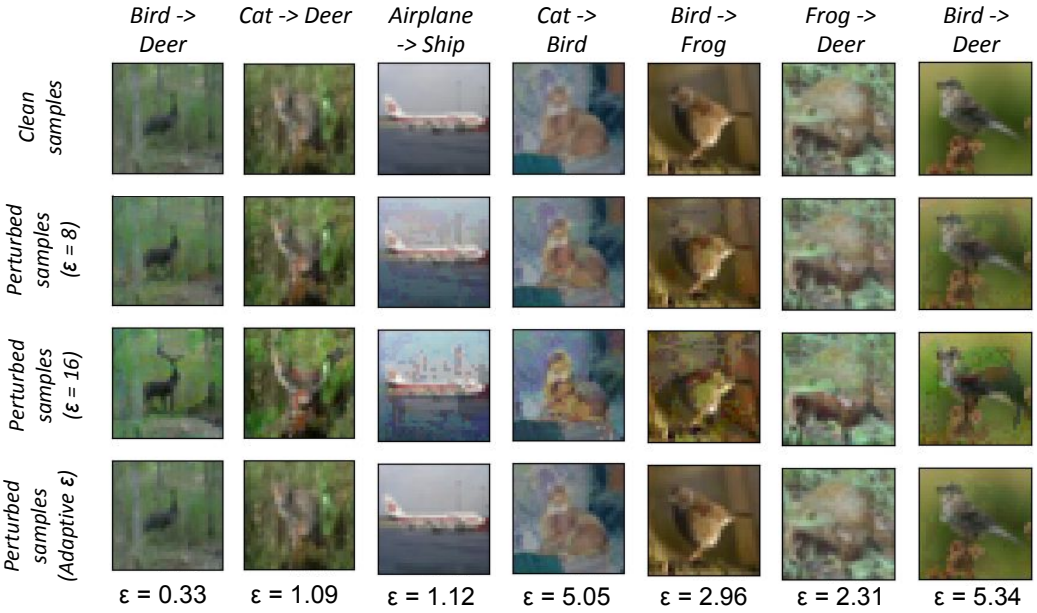
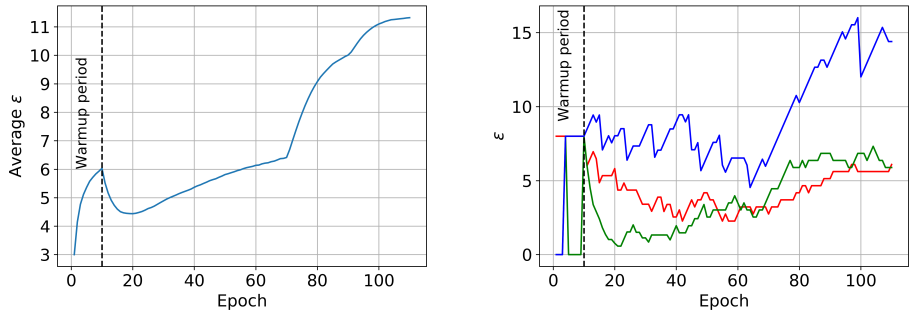


Figure 6: Visualizations of samples for which low ϵ's are assigned by instance adaptive adversarial training. These samples are close to the decision boundary and change class when perturbed with $\epsilon \geq 8$. Perturbing them with ϵ assigned by IAAT retains the class information.



(a) Average *epsilons*

(b) Individual ϵ for 3 random samples

Figure 7: Visualizing ϵ progress of instance adaptive adversarial trianing. Plot on the left shows average ϵ of samples over epochs, while the plot on the right shows ϵ progress of three randomly chosen samples.

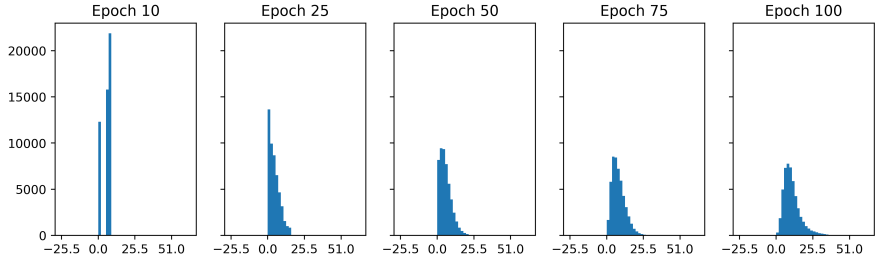


Figure 8: Histogram of ϵ of training samples at different training epochs

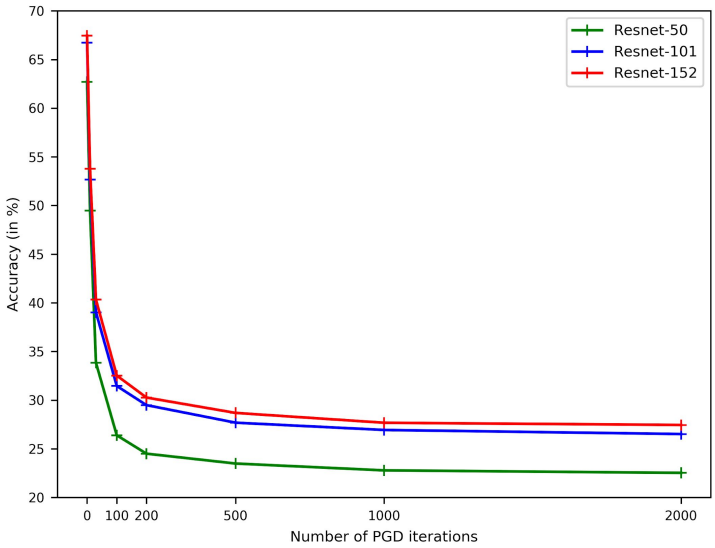


Figure 9: Imagenet robustness of IAAT over the number of PGD iterations

Table 9: Sensitivity of IAAT performance to hyperparameters β and γ. Models are trained on CIFAR-10 dataset using Wideresnet-32-10.

γ	β	Natural accuracy (in %)	Adversarial accuracy (in %)
0.00375 * 255	0.05	92.10	45.13
0.00375 * 255	0.1	90.27	46.32
0.00375 * 255	0.2	89.73	47.53
0.0075 * 255	0.05	92.15	46.34
0.0075 * 255	**0.1**	**91.34**	**48.53**
0.0075 * 255	0.2	89.95	48.72
0.011 * 255	0.05	90.47	46.11
0.011 * 255	0.1	90.52	46.17
0.011 * 255	0.2	89.99	46.32

D SENSITIVITY ANALYSIS

As shown in Alg. 2, IAAT algorithm has two hyper-parameters - smoothing constant β and discretization γ. In this section, we perform a sensitivity analysis of natural and robust accuracies by varying these hyper-parameters. Results are reported in Table. 9. We observe that the algorithm is not too sensistive to the choice of hyper-parameters. But the best performance is obtained for $\gamma = 1.9$ and $\beta = 0.1$.

E IMPLEMENTATION DETAILS

E.1 CIFAR

On CIFAR-10 and CIFAR-100 datasets, our implementation follows the standard *adversarial training* setting used in Madry et al. (2018). During training, adversarial examples are generated using PGD-10 attacks, which are then used to update the model. All hyperparameters we used are tabulated in Table. 10.

Table 10: Hyper-parameters for experiments on CIFAR-10 and CIFAR-100

Hyperparameters	Resnet-18	WideResnet-32-10
Optimizer	SGD	SGD
Start learning rate	0.1	0.1
Weight decay	0.0002	0.0005
Number of epochs trained	200	110
Learning rate annealing	Step decay	Step decay
Learning rate decay steps	[80, 140, 170]	[70, 90, 100]
Learning rate decay factor	0.1	0.2
Batch size	128	128
Warmup period	5 epochs	10 epochs
ϵ used in warmup (ϵ_w)	8	8
Discretization γ	1.9	1.9
Exponential averaging factor β	0.1	0.1
Attack parameters during training		
Attack steps	10	10
Attack ϵ (for adv. training only)	8	8
Attack learning rate	2/255	2/255

Table 11: Hyper-parameters for experiments on Imagenet

Hyperparameters	Imagenet
Optimizer	SGD
Start learning rate	0.1 $\times$ (effective batch size / 256)
Weight decay	0.0001
Number of epochs trained	110
Learning rate annealing	Step decay with LR warmup
Learning rate decay steps	[35, 70, 95]
Learning rate decay factor	0.1
Batch size	32 per GPU
Warmup period	30 epochs
ϵ used in warmup (ϵ_w)	16
Discretization γ	4
Exponential averaging factor β	0.1
Attack parameters during training	
Attack steps	30
Attack ϵ (for adv. training only)	16
Attack learning rate	1/255

E.2 IMAGENET

For Imagenet implementation, we mimic the setting used in Xie et al. (2019). During training, adversaries are generated with PGD-30 attacks. This is computationally expensive as every training update is followed by 30 backprop iterations to generate the adversarial attack. To make training feasible, we perform distributed training using synchronized SGD updates on 64 / 128 GPUs. We follow the training recipe introduced in Goyal et al. (2017) for large batch training. Also, during training, adversarial attacks are generated with FP-16 precision. However, in test phase, we use FP-32.

We further use two more tricks to speed-up instance adaptive adversarial training: (1) A weaker attacker(PGD-10) is used in the algorithm for selecting ϵ (Alg. 2). (2) After ϵ_i is selected per Alg. 2, we clip it with a lower-bound i.e., $\epsilon_i \leftarrow max(\epsilon_i, \epsilon_{lb})$. $\epsilon_{lb} = 4$ was used in our experiments.

Hyperparameters used in our experiments are reported in Table 11. All our models were trained on PyTorch.

Table 12: Training time for Imagenet experiments

Model	Number of GPUs used	Training time
Resnet-50	64	92 hrs
Resnet-101	128	78 hrs
Resnet-152	128	94 hrs

Resnet-50 model was trained on 64 Nvidia V100 GPUs, while Resnet-101 and Resnet-152 models were trained on 128 GPUs. Time taken for instance adaptive adversarial training for all models is reported in Table. 12.