CY 7790 Lecture 7: Black Box Attacks and Adversarial Training as a Defense

Apra Gupta and Lisa Oakley

October 4, 2021

The topics covered in today's class were:

1. Black Box Adversarial Attacks, a class of adversarial attacks that do not require knowledge of model architecture, parameters or training data by the adversary. Such attacks are much more realistic compared to white box or perfect knowledge attacks. Our focus today test time evasion attacks where the goal of the adversary is to construct minimally perturbed test samples that are misclassified by the model.
2. A Theoretical Framework to define and discuss the notion of robustness of ML models against test time adversarial attacks. This framework tries to build the notion of a model robust to all adversarial attacks and gives us ways to think about how we can achieve this goal.

1 Guo et al. Simple Black Box Adversarial Attacks

Presented by Hye Sun Yun, Scribed by Apra Gupta

Figure 1: Simple Black-box Attack (SimBA)

Problem Statement. Black box attacks are much more realistic that white box attacks but are also challenging to construct due to the inherent lack of knowledge the adversary has about the model. Black Box Attacks discussed prior to this paper tended to focus on trying to reconstruct the model from scratch and then attack the reconstructed model, relying on the transferability of the attack from the reconstructed model to the targeted model. This approach tends to be expensive and time consuming (in the context of querying some unknown model API that charges per query). Thus, black box attacks introduce a new added constraint: budgeting the number of queries that one can make to the targeted model. At the time, the most efficient and complex attacks required upwards of 10k+ of such queries. In this paper, the authors introduce a novel simple yet highly efficient algorithm for the creation of adversarial images in the black box setting that requires much fewer queries to the target model to successfully create an adversarial example by taking advantage of the class-confidence scores outputted by the target model.

Threat Model. This paper constructs a test time evasion attack in the black box setting, meaning the **adversarial objectives** here are to add **imperceptible** perturbations to some test image that induce the targeted model to misclassify it to either a specific target class or some class that is not the one the model would otherwise classify this image as. Since this is a black box setting, the adversary has no knowledge of the training data used to train the target model, its architecture or parameters. The **adversarial capabilities** include querying the target model for the class confidences (as opposed to just the label) across some or all possible classes for any test image of their choice a limited number of times. Again, since query cost is high, the adversary would like to be able to do all this, while simultaneously **minimizing the number of queries** to the model.

$$\min_{\delta} \;\; \ell_y(\mathbf{x}+\delta) \text{ subject to: } \;\; \|\delta\|_2 < \rho, \text{queries} \leq B$$

Figure 2: Optimization objective for adversary: minimize adversarial loss on minimally perturbed input x with budget constraint B

Methodology. The authors take advantage of their assumption of access to the continuous valued confidence score ($p_h(y|x)$) of the original class $h(x) = y$ in the untargeted case and that of the targeted class in the targeted case. In the untargeted case, the goal is to find some small δ such that $h(x + \delta) \neq y$. Their algorithm works by picking a random (uniformly without replacement) direction $q \in Q$ and some step size ϵ and adding ϵq to the input x ,if doing so decreases $p_h(y|x)$, they update $\delta = \delta + \epsilon q$ and if it increases $p_h(y|x)$, they update $\delta = \delta - \epsilon q$. In the targeted case, instead of minimizing the probability on the original class, the algorithm proceeds by maximizing probability on targeted class. See figure 3 for the pseudocode of the proposed algorithm in the untargeted case.

They show that if Q is orthonormal and all queried $x + \epsilon q$ bring some change in $P_h(y|x)$, then $\|\delta\|_2 = \sqrt{T}\epsilon$. The derivation is as follows:

Let $\alpha_i \in \{-\epsilon, 0, \epsilon\}$ denote the sign of the search direction chosen at step t, so

$$\delta_{t+1} = \delta_t + \alpha_t q_t$$

We can recursively expand the above term to get the final perturbation δ_T after T a sum across each search directions

$$\delta_T = \textstyle\sum_{t=1}^{T} \alpha_t q_t$$

Since the directions q_t are orthogonal, $q_t^\top q_{t'} = 0$ for any $t \neq t'$. We can therefore compute the L_2-norm of the adversarial perturbation:

$$\|\delta_T\|_2^2 = \left\|\textstyle\sum_{t=1}^{T} \alpha_t q_t\right\|_2^2 = \textstyle\sum_{t=1}^{T} \|\alpha_i q_t\|_2^2 = \textstyle\sum_{t=1}^{T} \alpha_t^2 \|q_t\|_2^2 \leq T\epsilon^2.$$

Here, the second equality follows from the orthogonality of q_t and $q_{t'}$, and the last inequality is tight if all queries result in a step of either ϵ or $-\epsilon$. Thus the adversarial perturbation has L_2-norm at most $\sqrt{T}\epsilon$ after T iterations.

The following are some key observations that the authors make that enable their method to be successful:

Algorithm 1 SimBA in Pseudocode

```
1:  procedure SIMBA($\mathbf{x}, y, Q, \epsilon$)
2:      $\delta = \mathbf{0}$
3:      $\mathbf{p} = p_h(y \mid \mathbf{x})$
4:      while $\mathbf{p}_y = \max_{y'} \mathbf{p}_{y'}$ do
5:          Pick randomly without replacement: $\mathbf{q} \in Q$
6:          for $\alpha \in \{\epsilon, -\epsilon\}$ do
7:              $\mathbf{p}' = p_h(y \mid \mathbf{x} + \delta + \alpha\mathbf{q})$
8:              if $\mathbf{p}'_y < \mathbf{p}_y$ then
9:                  $\delta = \delta + \alpha\mathbf{q}$
10:                 $\mathbf{p} = \mathbf{p}'$
11:                 break
        return $\delta$
```

Figure 3: Psuedocode for SimBA in targeted case

1. Continuous-valued confidence scores serve as a strong proxy for adversarial loss to guide the search for adversarial perturbations

2. If the distance to a decision boundary is small, the exact direction of a perturbation does not matter.

3. The density of adversarial examples is much higher in low frequency domains.

Building on the third point, the authors use the Discrete Cosine Transformation as one of the methods used to construct the orthonormal basis Q (they call the DCT extracted basis as Q_{DCT}). DCT is an orthonormal transformation used in signal processing and data compression. It maps a 2-dimensional image feature matrix into frequency coefficients corresponding to magnitudes of cosine wave functions, and preserves only a fraction ($\frac{1}{r}$) of the lowest frequencies since critical and content-defining image information has been found to live in the low end of the frequency domain, with $r = 8$ found to be optimal. The other method the authors use to construct Q is simply $Q = I$ or the standard basis: at every iteration they simple increase or decrease one color of a single randomly chosen pixel.

When the the random direction q is sampled from the standard basis, the algorithm is referred to as SimBA, and as SimBA-DCT when the DCT space is used.

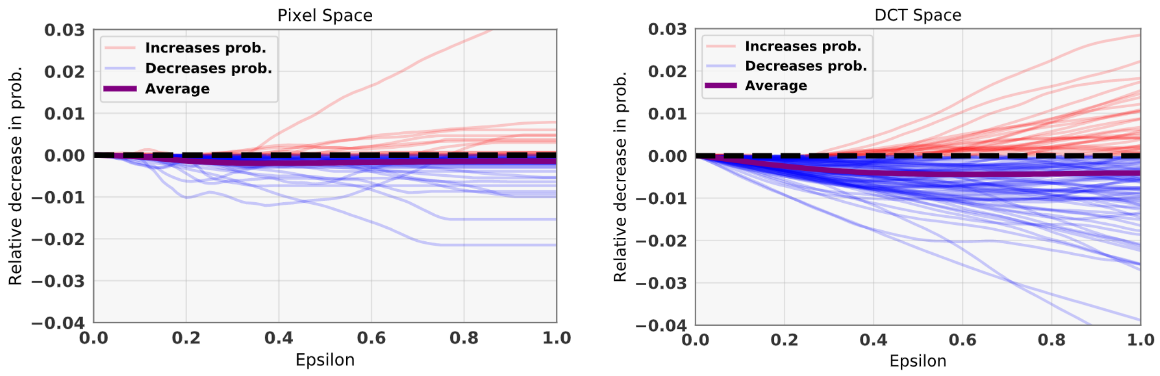


Figure 4: Relative decrease in $p_h(y|x + \epsilon q)$ as a function of ϵ for randomly sampled search directions q

The following is a brief discussion of the experiments conducted by the authors along with key results:

1. The authors plot the relative decrease in output probability $p_h(y|x + \epsilon q)$ as a function of ϵ for randomly sampled search directions q in pixel and DCT space by running perturbed ImageNet validation samples on a ResNet-50 model (4). They find that $p_h(y|x + \epsilon q)$ consistently decreases monotonically in ϵ, meaning their algorithm is not overly sensitive to the choice of ϵ. They also find that search in DCT space tends to lead to steeper descent directions than pixel space.

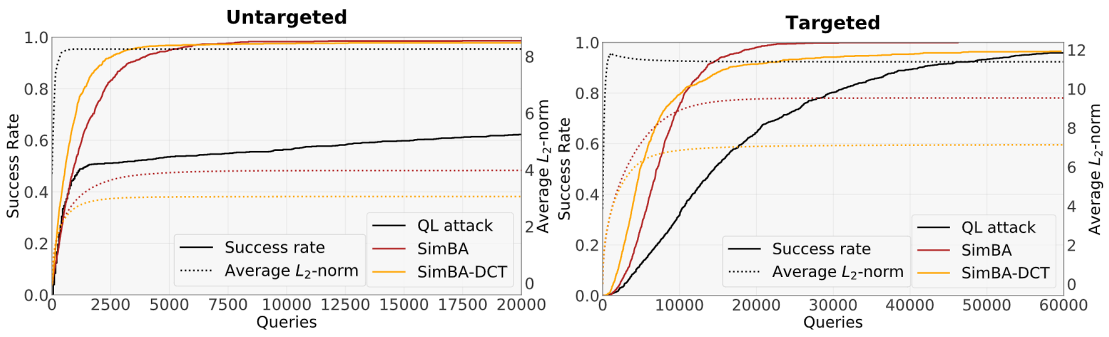


Figure 5: Comparison of Success Rate and no. of Queries of SimBA and SimBA-DCT against QL-attack

2. They evaluate their method on 1000 correctly classified ImageNet validation samples for both the targeted and untargeted case against state of the art models (QL-attack, LFBA-etc). They find that both SimBA and SimBA-DCT achieves a faster increase in success rate as well as much lower query counts than the QL-attack. While SimBA-DCT requires significantly lower query counts than SimBA to construct an adversarial sample, it fails to construct an adversarial sample within 60,000 queries in 2.5 % of the cases. (See figure 5 for comparison against QL-attack and figure 6 for comparison of query counts between SimBA and SimBA-DCT). Figure 7 shows aggregate statistics comparing SimBA & SimBA-DCT to other black-box attacks.

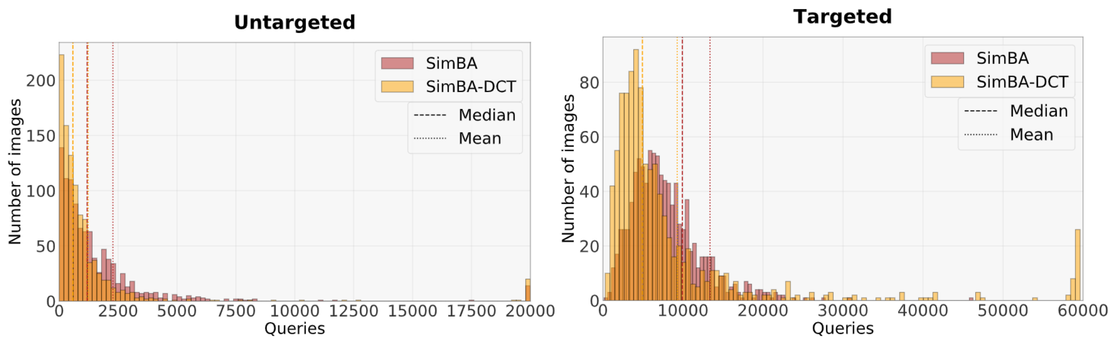


Figure 6: Histogram of queries required until a successful attack for SimBA and SimBA-DCT

3. Authors evaluate SimBA and SimBA-DCT against DenseNet-121 and Inception v3. They find that while their methods can successfully construct adversarial examples against all models with high probability in under 10,000 queries, complex models such as Inception v3 are noticeably more difficult to attack. Figure 8 plots this comparison between architectures.

4. To demonstrate efficacy in a real world setting, authors attack the Google Cloud Vision API: which only returns the top few class probabilities associated with an image. Their targeted attack thus aims to remove the top 3 classes. See figure 9 for the results. Their attack beats the state of the art and is the first adversarial attack result on Google Cloud Vision that has a high reported success rate (70%) within very limited number of queries.

Untargeted

Attack	Average queries	Average L_2	Success rate
	Label-only		
Boundary attack	123,407	5.98	100%
Opt-attack	71,100	6.98	100%
LFBA	30,000	6.34	100%
	Score-based		
QL-attack	28,174	8.27	85.4%
Bandits-TD	5,251	5.00	80.5%
SimBA	1,665	3.98	98.6%
SimBA-DCT	**1,283**	3.06	97.8%

Targeted

Attack	Average queries	Average L_2	Success rate
	Score-based		
QL-attack	20,614	11.39	98.7%
AutoZOOM	13,525	26.74	100%
SimBA	**7,899**	9.53	100%
SimBA-DCT	8,824	7.04	96.5%

Figure 7: Aggregate Statistics comparing SimBA and SimBA-DCT to other black-box attacks

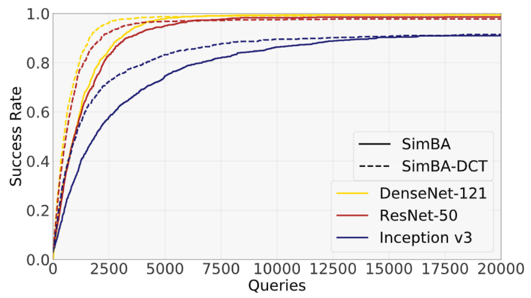


Figure 8: Comparison of Success Rate and Query Count of SimBA and SimBA-DCT across different model architectures

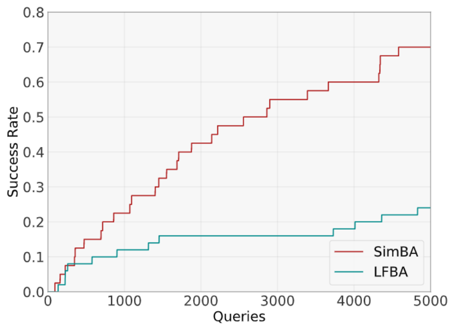


Figure 9: Success rate across number of model queries for Google Cloud Vision attack

Class Discussion.

Prior Work This paper is a follow-up to the paper Low Frequency Adversarial Perturbation by Guo et al. (2018) that dicusses the application of Discrete Cosine Transform in adversarial example construction.

Loss Function This paper does not use a common loss function such as cross-entropy loss. Rather, it defines the 'loss' function to minimize over as the confidence score outputted by the targeted model for the original class in the untargeted attack.

Effect of random perturbation directions Do all directions of perturbation have an equal impact on the probability outputted by the targeted model? No, but most of them seem to decrease the probability of the original class for the test image.

Size of orthonormal basis Q The maximum size of Q is $d \times d$ (where d is the number of features), but we can reduce the total size of Q based on some constraints to limit sampling of random directions q to the most useful ones. For example, we chop off the highest frequencies of the DCT transform and only preserve the lowest $\frac{1}{r}$ frequencies.

Lowest Frequencies of DCT Transfrom This space has been shown to carry the most useful information and thus by moving away from this space, we are most likely to be able to construct adversarial examples.

General Basis Why did the authors not use the general basis? Likely because of the high dimensionality of the space. The size of this space would be $number^2_{pixels} \times 3$, too high to run some form of decomposition on.

Defensive Distillation Will defensive distillation defend against this attack? It likely will, since if the probabilities of the classes are closer to each other it will make it harder to attack.Defensive distillation smooths the outputs likely having this effect since you don't get as much information about the gradients of the target model.

Effect of Defensive Distillation If confidence scores are closer, wouldn't' it be easier to attack since lesser queries would be require to change the class probabilities to beat the original class? This depends on objective: it's probably easier to switch the top confidence classes, but not change all the top confidence classes to different ones since they will all change a bit together

Is is realistic to expect confidence scores? This depends on the application: in the case of Google API, the answer is yes, but in more practical applications that are fully automated in a pipeline (like in a car vision API) we might only have access to the predicted label in the end with only internal access to confidence scores. The follow up paper HopSkipJumpAttack: A Query-Efficient Decision-Based Attack discusses an effective attack where the adversary only has access to the labels.

API's perspective From the API's perspective you might be able to detect the fact that the queries are really close a signal which may point to adversarial examples. Many defenses exist in the same vain that try to exploit this fact about black-box attack such as IP tracking, checking L2 norms etc-.

Targeted Case In the targeted case, wouldn't the probabilities of other classes also go up when maximizing a particular classes probabilities? Since there is nothing stopping this from happening, that is probably the case and the reason why targeted attacks have a higher query requirement and failure rate.

2 Madry et al. Towards Deep Learning Models Resistant to Adversarial Attacks

Presented by Jaydeep Borkar, Scribed by Lisa Oakley

Problem Statement. This paper considers two main problems. The first is to formally define the notion of an "adversarially robust" neural network which is robust to adversarial examples generated using any attack under a specific threat model. The second problem is to find an adversarial training method which results in a trained neural network guaranteed to be adversarially robust according to this definition.

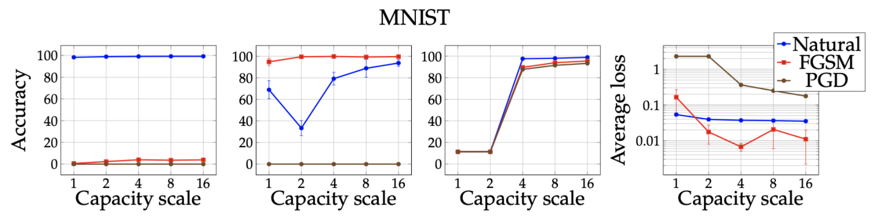
Threat Model. This paper focuses on the scenario of evasion attacks against neural network, in which an adversary is able to modify data points in the test set after the model has been trained. In particular, they analyze robustness with respect to an adversary who is able to perturb the features of an existing data point, bounded by an ℓ_∞ ball around their original values. They further argue that robustness against "first order adversaries", optimization-based adversaries who rely only on first-order information, is sufficient to call the method "universally robust", given that optimization over first order information is the state-of-the-art technique in machine learning. They assume a white-box attacker with full knowledge of the model and its parameters, as well as black-box scenarios.

Methodology. Madry et al. consider a notion of adversarially robust training which relies on a min-max optimization. Formally,

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \max_{x' \in P(x_i)} L\left(f_\theta\left(x'\right), y_i\right). \tag{1}$$

In this optimization, the inner (maximization) problem finds the most impactful adversarial perturbation of data point x (i.e. the inner maximization finds an adversarial example). The outer (minimization) problem finds a set of parameters which minimizes the loss of this adversarial example using Stochastic Gradient Descent (SGD). This procedure can be described as

1. Sample labeled training data point $(\mathbf{x}, y)$
2. Compute maximizer $\mathbf{x}^*$ of the robust loss $\phi_{(\mathbf{x},y)}(\theta)$ (using some attack method)
3. compute gradient $g = \nabla_\theta L(f_\theta(\mathbf{x}^*, y))$



CIFAR10

	Simple	Wide	Simple	Wide	Simple	Wide	Simple	Wide
Natural	92.7%	95.2%	87.4%	90.3%	79.4%	87.3%	0.00357	0.00371
FGSM	27.5%	32.7%	90.9%	95.1%	51.7%	56.1%	0.0115	0.00557
PGD	0.8%	3.5%	0.0%	0.0%	43.7%	45.8%	1.11	0.0218

(a) Standard training (b) FGSM training (c) PGD training (d) Training Loss

Figure 10: Figure 4 from Madry et al.

4. Update θ with gradient g (as in SGD).
5. repeat 1-4 until convergence

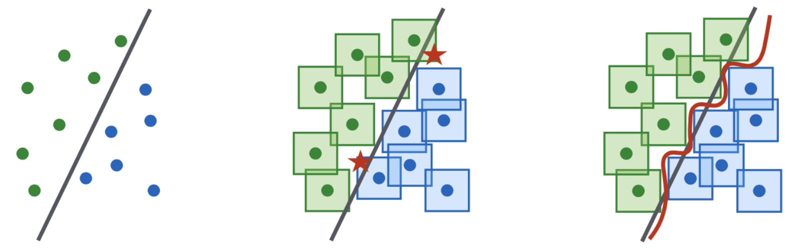


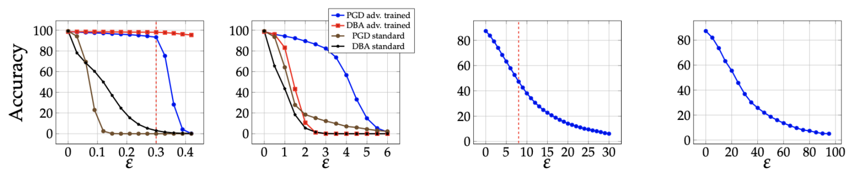
Figure 3: A conceptual illustration of standard vs. adversarial decision boundaries. Left: A set of points that can be easily separated with a simple (in this case, linear) decision boundary. Middle: The simple decision boundary does not separate the ℓ_∞-balls (here, squares) around the data points. Hence there are adversarial examples (the red stars) that will be misclassified. Right: Separating the ℓ_∞-balls requires a significantly more complicated decision boundary. The resulting classifier is robust to adversarial examples with bounded ℓ_∞-norm perturbations.

Figure 11: Fig. 3 from Madry et al.

To determine which attack method to use to solve the inner maximization (step 2), Madry et al. posit that Projected Gradient Descent (PGD) attacks are the strongest of the first-order attacks, and thus adversarial training with respect to a PGD attacker is sufficient to protect against any first-order attack. They provide initial intuition for this conjecture by showing that, over MNIST and CIFAR10 datasets, the local minima of the loss values are well-concentrated. They further support the conjecture by showing empirically that models trained adversarially with respect to a PGD attacker are also robust against assumed weaker attack techniques such as Fast Gradient Sign Method (FGSM) and randomized FGSM attacks. Fig. 10 present results which show that adversarially training with an FGSM attacker results in a model resilient to FGSM, but not PGD attacks, but adversarially training with a PGD attacker results in a model resilient to all attacks.

Fig. 10 also presents results on the impact of model capacity on robustness. Models with higher capacity are more expressive, and can draw a decision boundary which accounts for adversarial noise, as is shown in Fig. 11. In fact, Madry et al. argue that achieving a high enough capacity is "crucial" for robustness.

Madry et al. compare their results to a baseline adversarial training method, as well as the non-adversarially trained counterparts. In Fig. 12, we see that the PGD adversarial training method outperforms the standard training on MNIST and CIFAR-10, and performs similarly to the baseline for the MNIST data set. There is a sharp dropoff after ε exceeds the value for which the model was adversarially trained.



(a) MNIST, ℓ_∞-norm (b) MNIST, ℓ_2-norm (c) CIFAR10, ℓ_∞-norm (d) CIFAR10, ℓ_2-norm

Figure 6: Performance of our adversarially trained networks against PGD adversaries of different strength. The MNIST and CIFAR10 networks were trained against $\varepsilon = 0.3$ and $\varepsilon = 8$ PGD ℓ_∞ adversaries respectively (the training ε is denoted with a red dashed lines in the ℓ_∞ plots). In the case of the MNIST adversarially trained networks, we also evaluate the performance of the Decision Boundary Attack (DBA) [4] with 2000 steps and PGD on standard and adversarially trained models. We observe that for ε less or equal to the value used during training, the performance is equal or better. For MNIST there is a sharp drop shortly after. Moreover, we observe that the performance of PGD on the MNIST ℓ_2-trained networks is poor and significantly overestimates the robustness of the model. This is potentially due to the threshold filters learned by the model masking the loss gradients (the decision-based attack does not utilize gradients).

Figure 12: Fig. 6 from Madry et al.

Class Discussion.

Original vs. Adversarial: Is the attack pushing points away from each other or close to decision boundary? The original classifier will not train considering balls around the points, just about the points themselves. Adversarial training classifies points, and decision boundary is aware of the balls around the point that the adversarial examples will exist in. This does not work for less expressive (linear) models.

Overfitting: The iterative process helps it not overfit to a particular adversarial example. However for the inner maximization: PGD samples overfit. The authors say that there is some label leaking, but there is nothing definite.

Standard vs. Robust Accuracy: There is some tradeoff between adversarial and standard accuracy. Follow up work: Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, Percy Liang, *Understanding and Mitigating the Tradeoff Between Robustness and Accuracy*.

Inner Maximization: Why just one adversarial example per step? If you maximize it, they will have similar outcome on loss. It is good enough according to experiments, but it is not a consistent method because they do just choose one.

Other Attack Methods: Data augmentation during training is something that has been used for a long time (for generalization). At this point: very strong attacks, but can we use them to make our model resistant. PGD is a reasonable computation to do in the inner minimization, however other attacks might be a good future direction. Many similar defenses proposed at the same time, this one appears to be the most impactful among those attempts.

Capacity: Capacity in this paper is on order of doubling number of layers (repeating same arch. twice), not small modifications to number of convolutional or other layers.

Computationally Expensive: Computing adversarial example at every iteration of SGD can become very expensive. Follow up work addresses this: Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, Tom Goldstein, *Adversarial Training for Free!*