Review on Adversarial Machine Learning

Adam Dziedzic
Postdoctoral Fellow

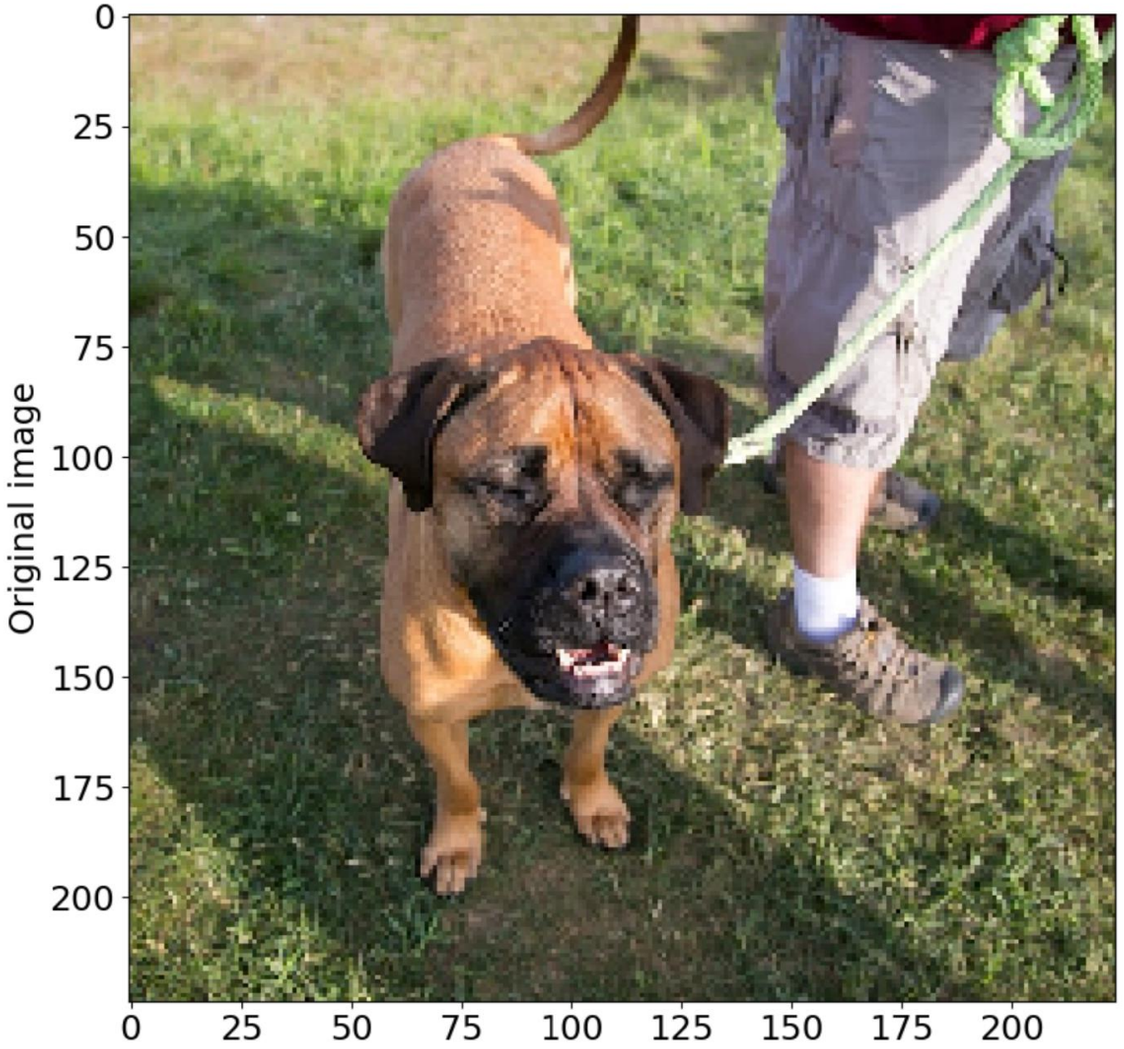
ady@vectorinstitute.ai
December 15th, 2020

VECTOR INSTITUTE

UNIVERSITY OF TORONTO

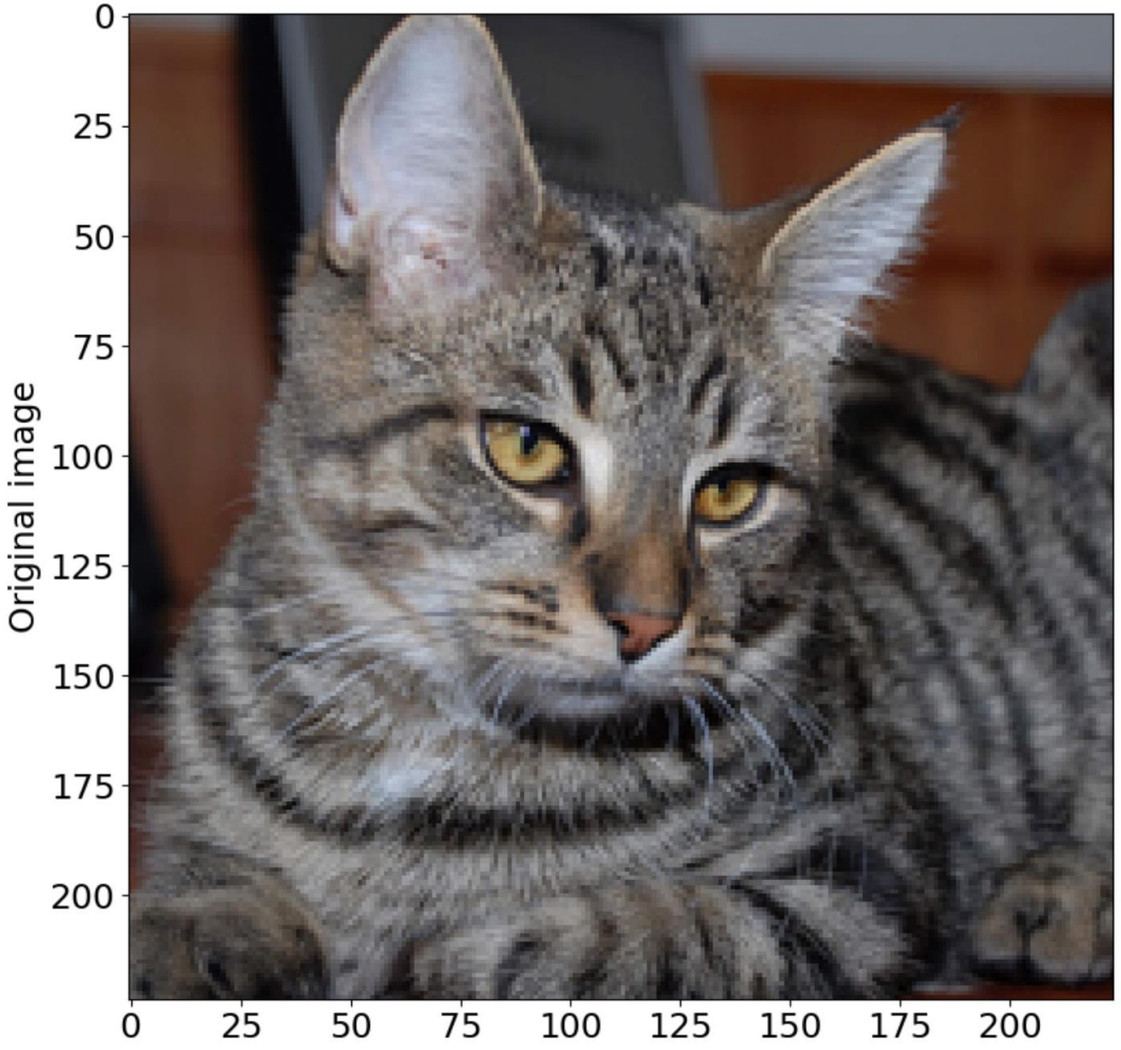
Dog

original label: bull mastiff
confidence: 0.9592
L2 distance: 0.0



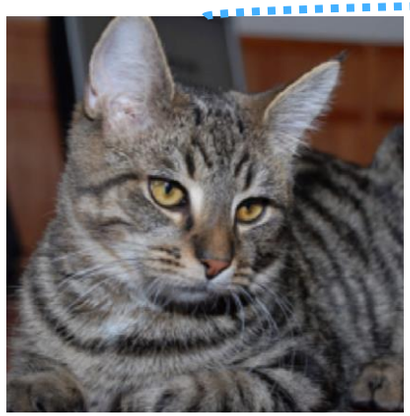
Cat

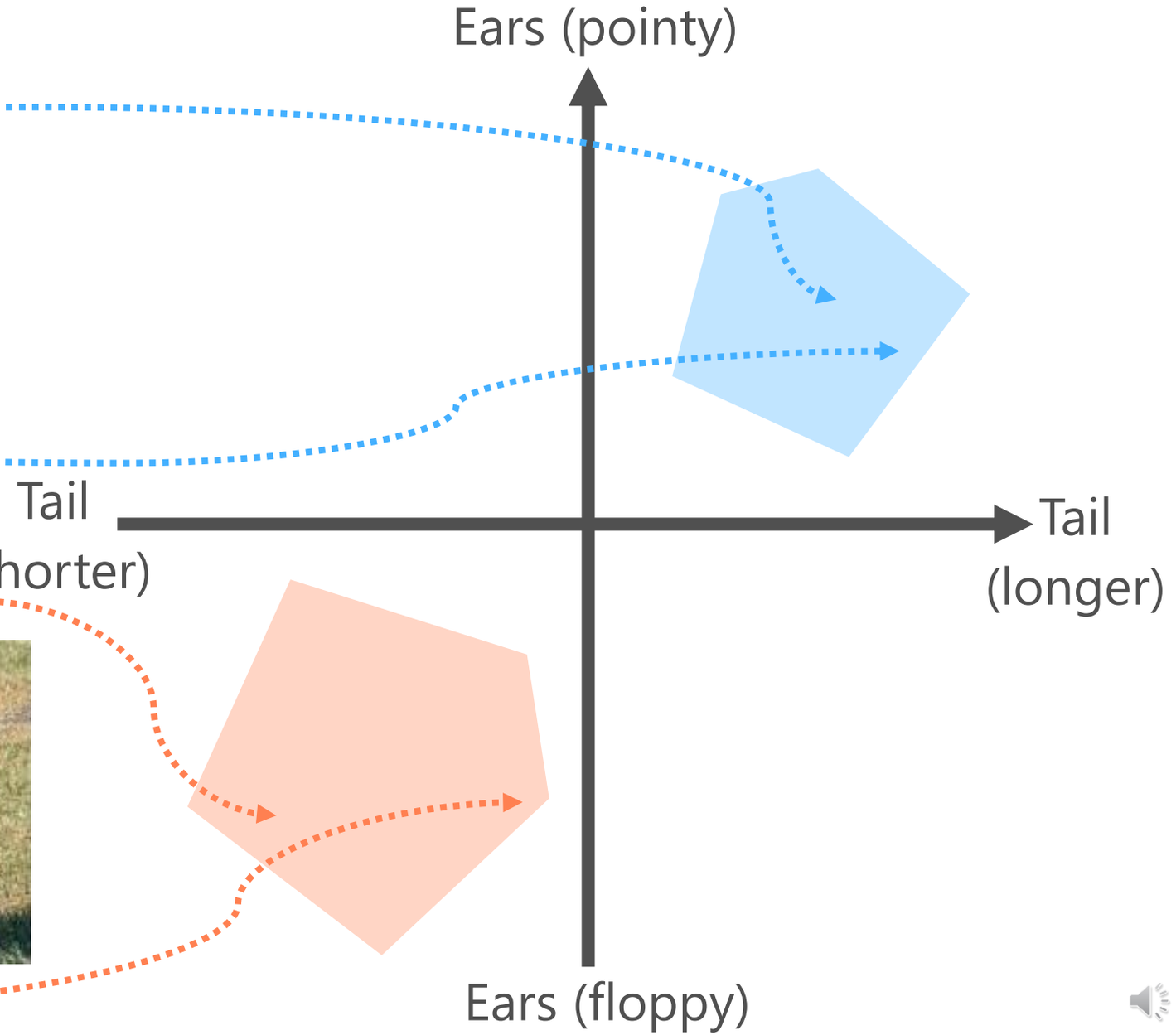
original label: tiger cat
confidence: 0.8539
L2 distance: 0.0



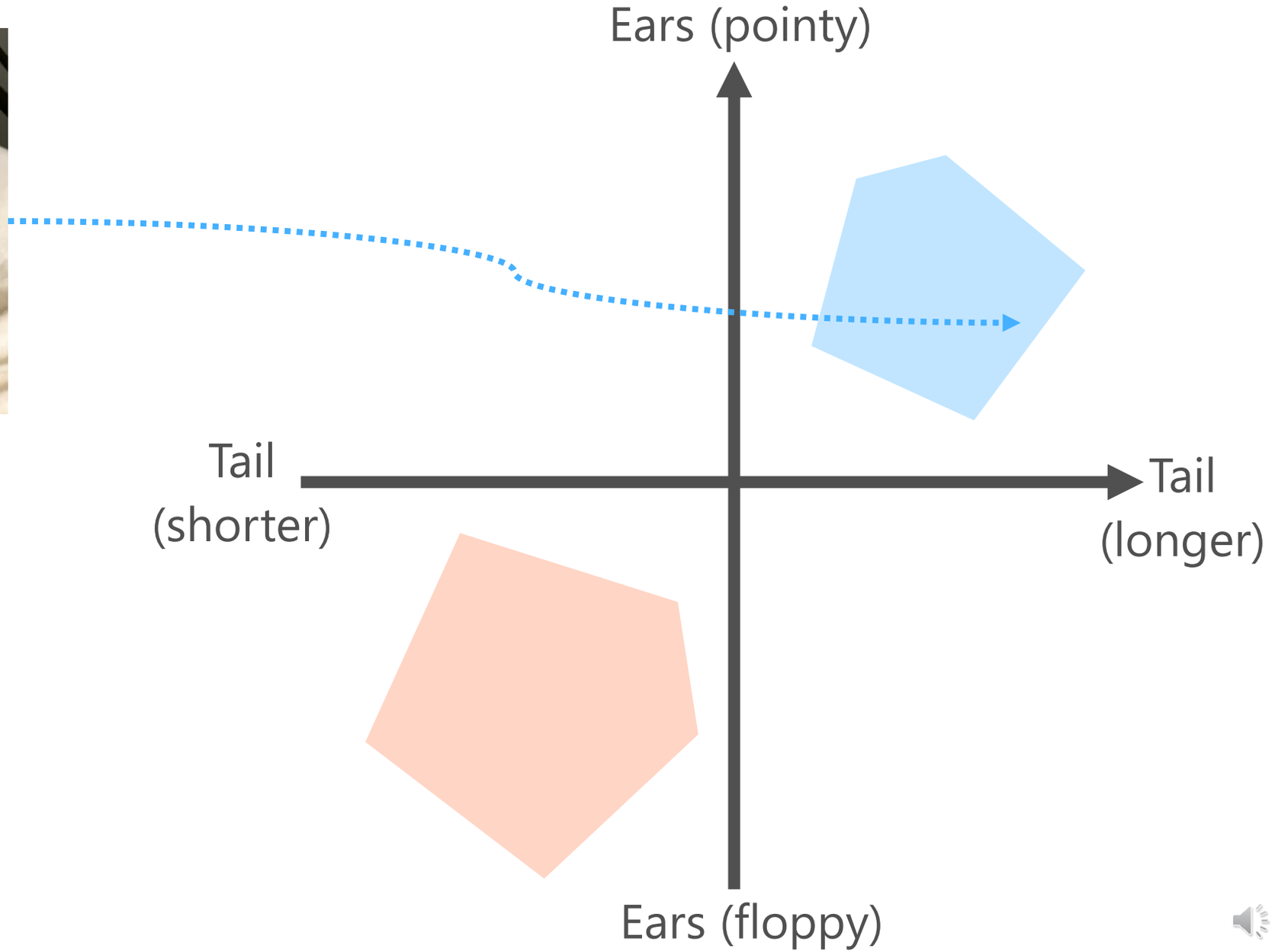
Machine Learning: train time

Machine Learning: train time

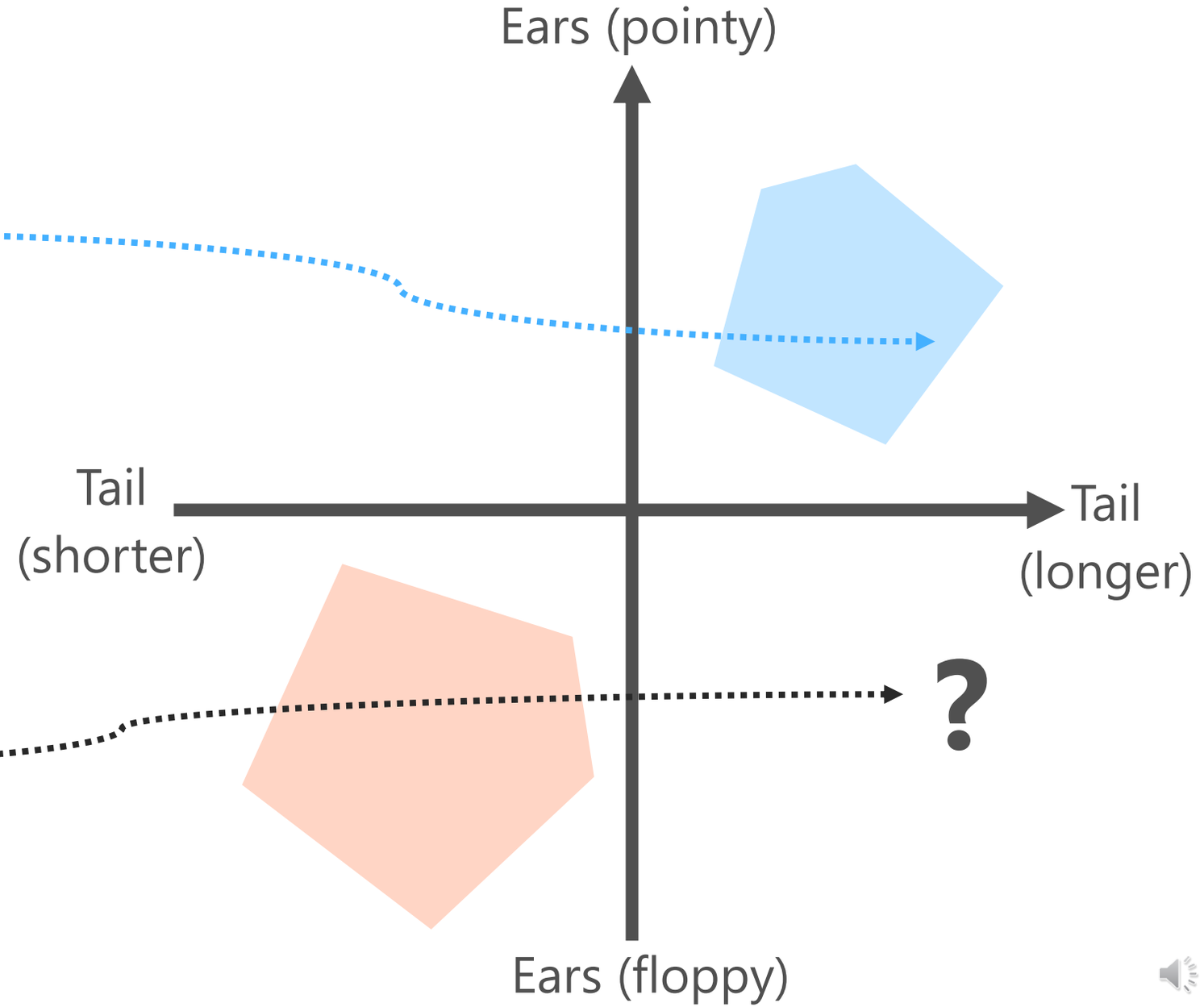




Machine Learning: inference time

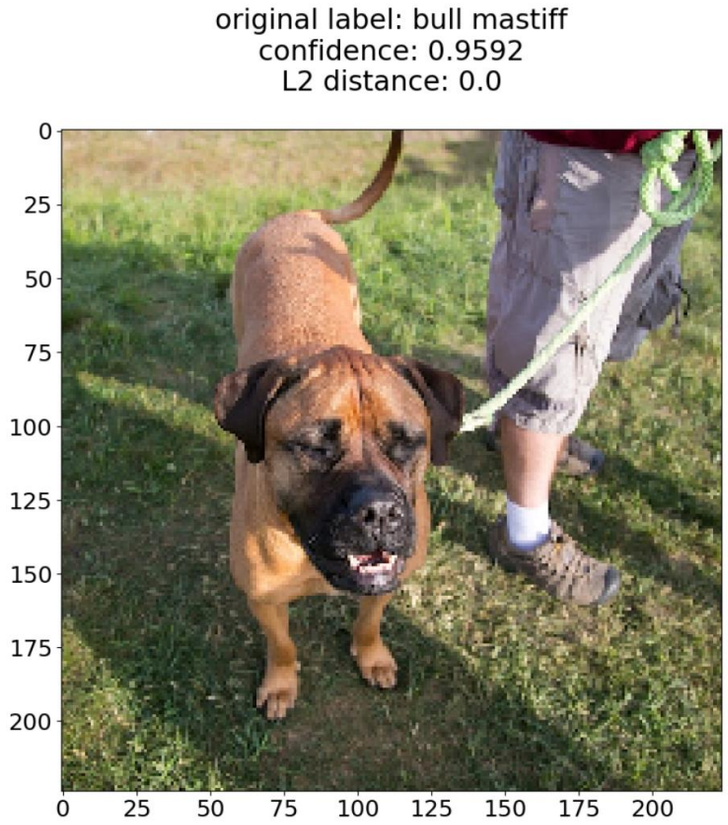


Machine Learning: inference time

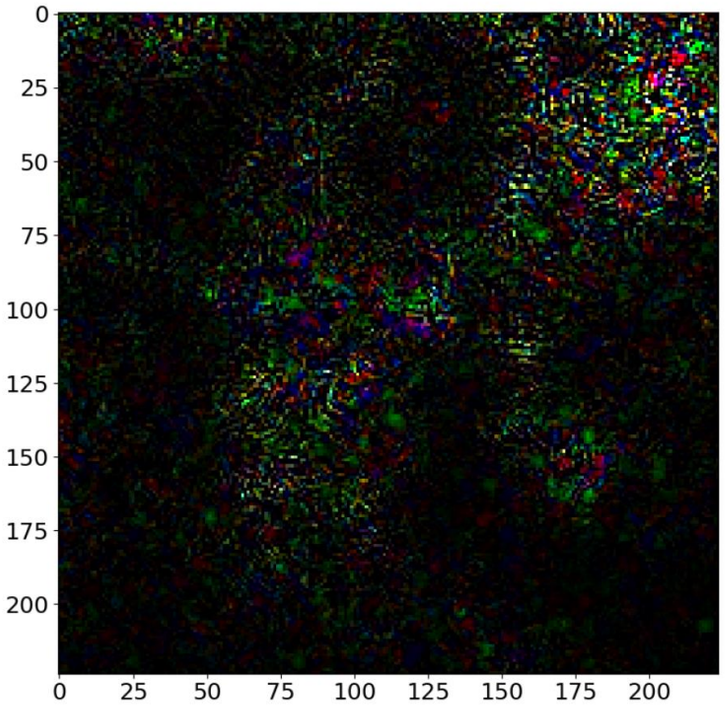


Adversarial perturbations: test time

[BS14]



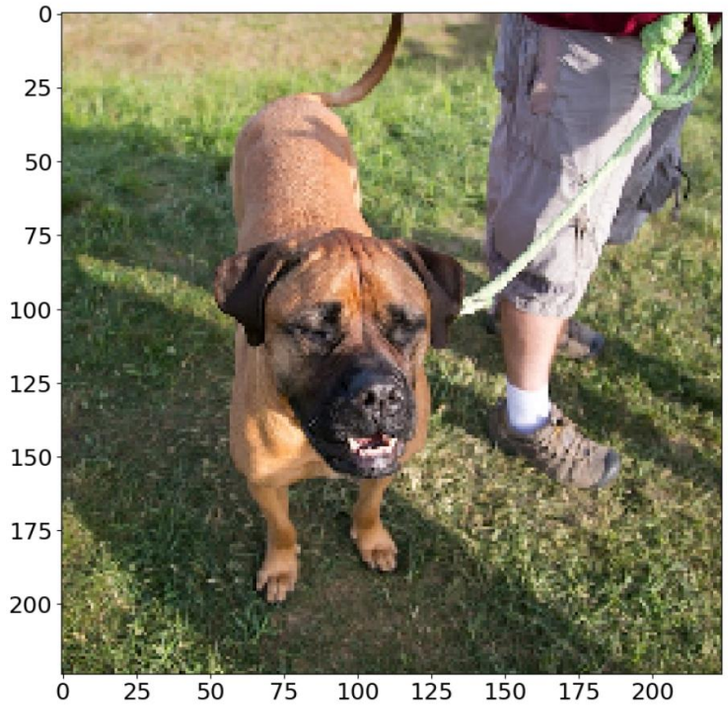
100 X enlarged difference between original & adversarial images



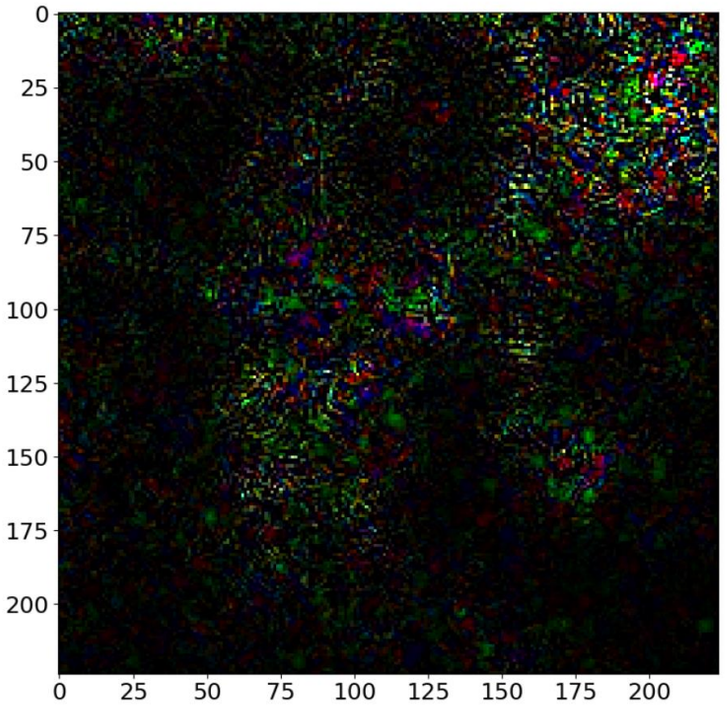
Adversarial perturbations: test time

[BS14]

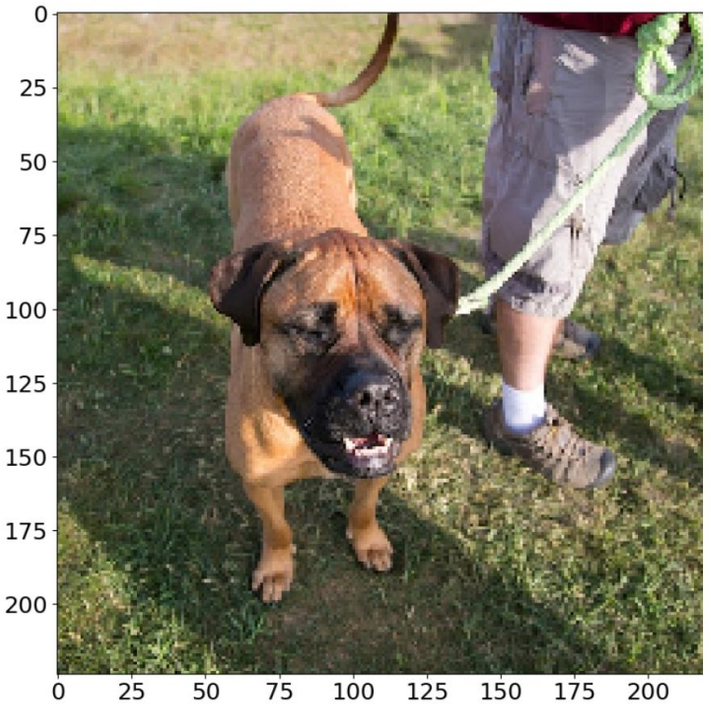
original label: bull mastiff
confidence: 0.9592
L2 distance: 0.0



100 X enlarged difference between original & adversarial images



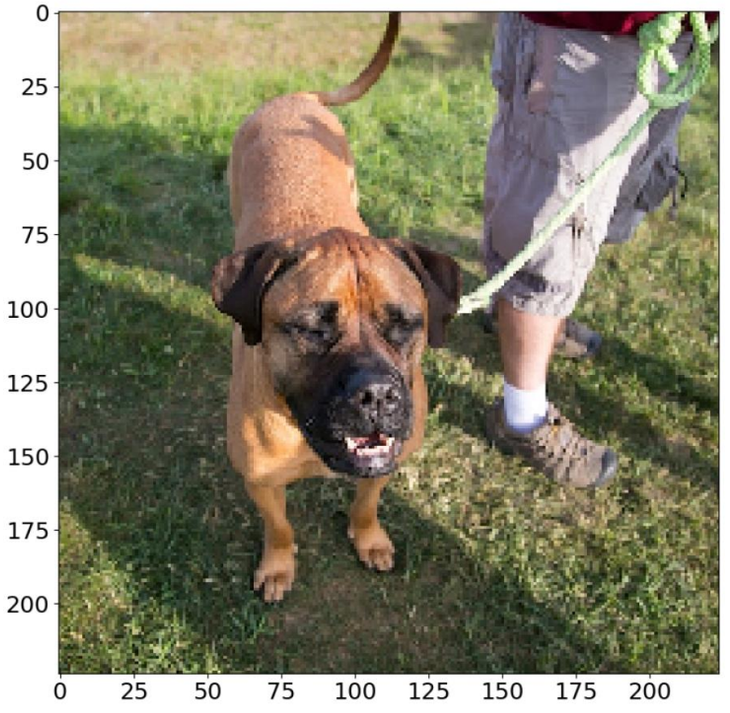
adv. label: tiger cat
confidence: 0.1929
L2 distance: 1.3815



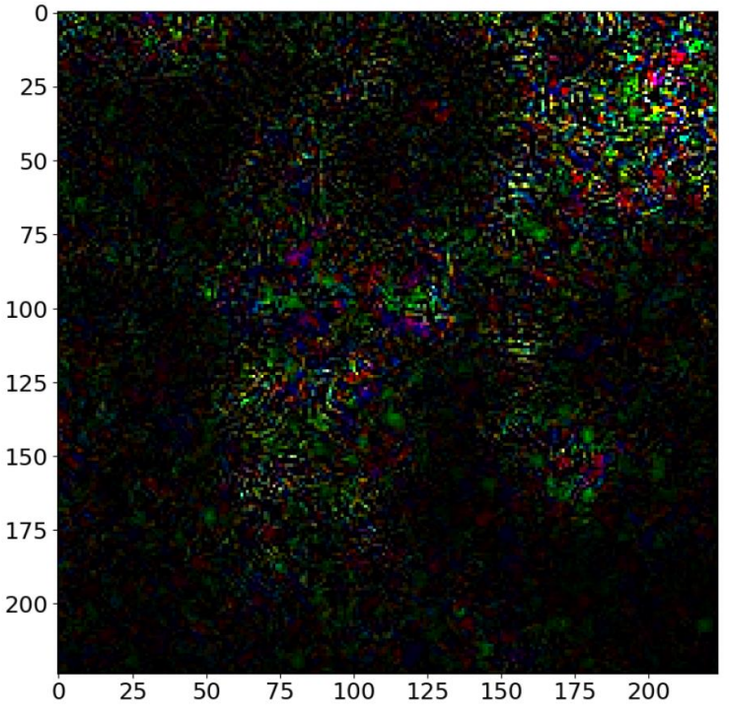
Adversarial examples: broadest definition

[GP17]

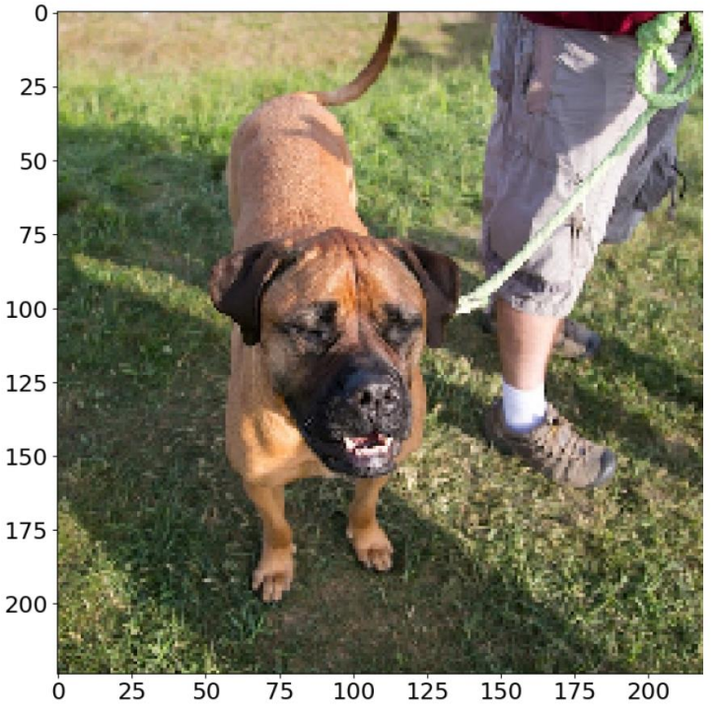
original label: bull mastiff
confidence: 0.9592
L2 distance: 0.0



100 X enlarged difference between original & adversarial images



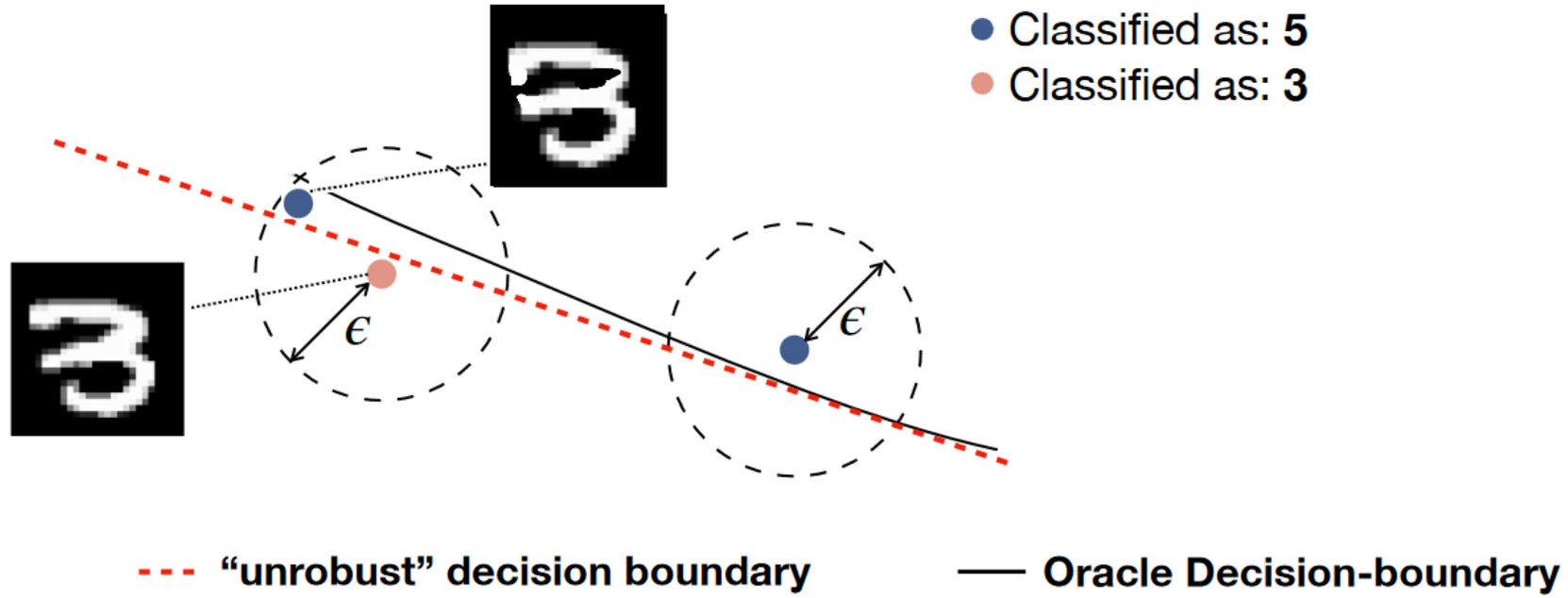
adv. label: tiger cat
confidence: 0.1929
L2 distance: 1.3815



An input to an ML model that is intentionally designed by an attacker to fool the model into producing an incorrect output.

Adversarial examples: ϵ-bounded

[TB20]



$$f(x^*) \neq f(x)$$

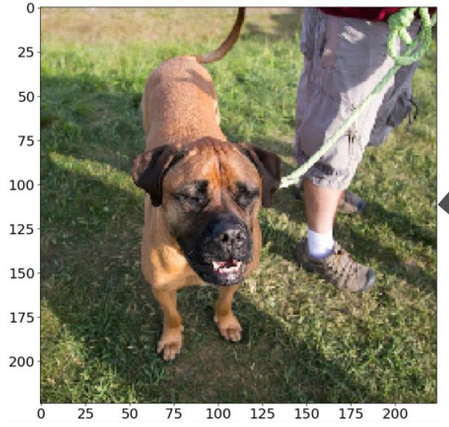
$$\|x^* - x\| \leq \epsilon$$

assumes: $O(x^*) = O(x)$

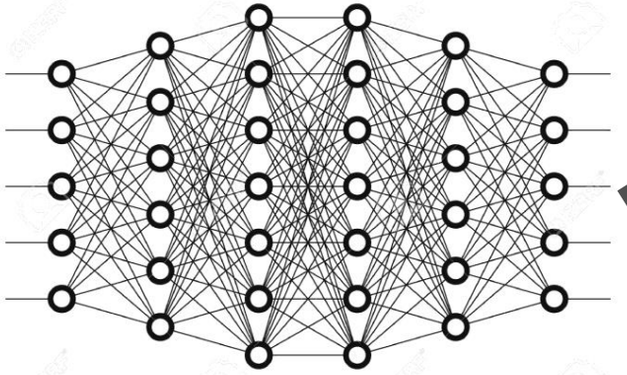
Threat model

[PM17]

Bull mastiff



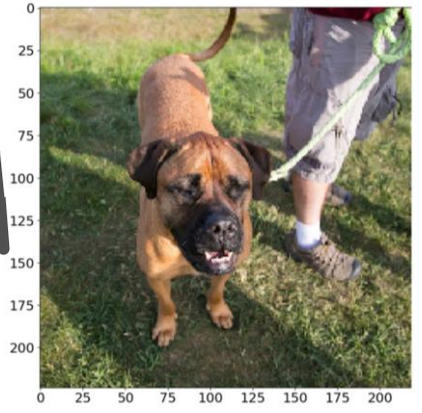
x



White-box

Black-box

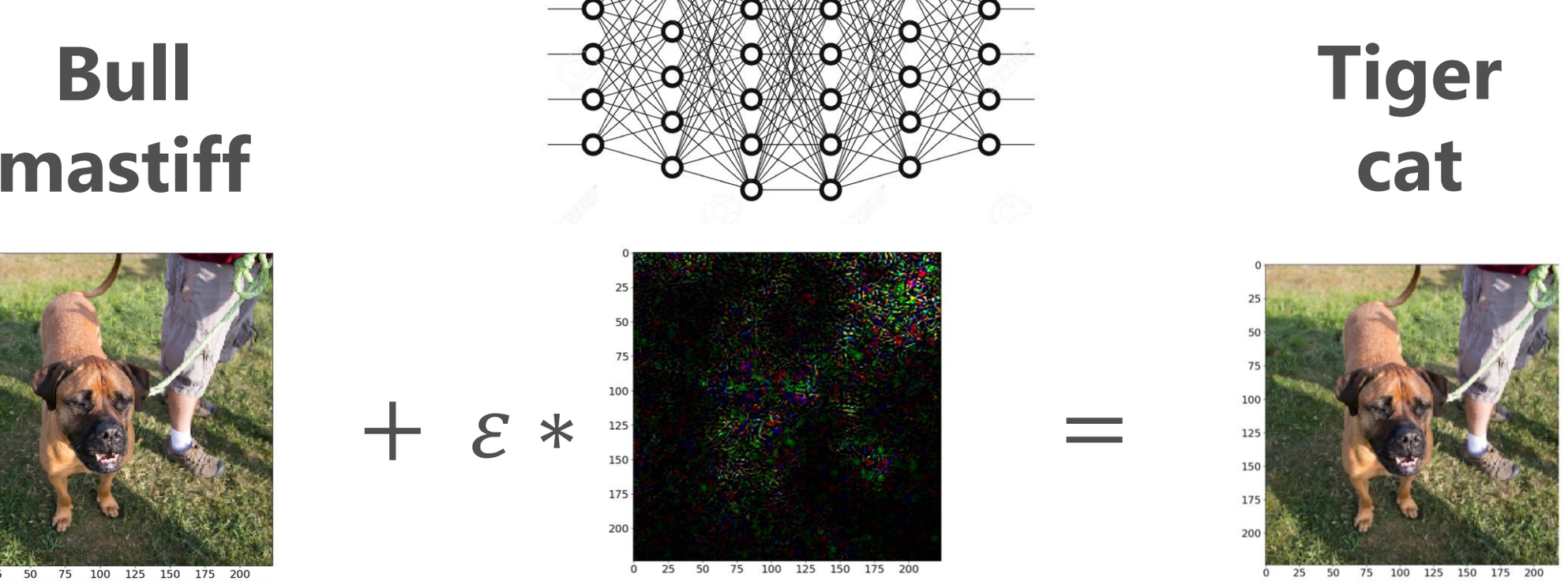
Tiger cat



x'

Crafting white-box attacks

Bull mastiff



Tiger cat

$+ \quad \varepsilon * \quad$

$=$

x $\qquad$ $\nabla_x Loss(f(x;\theta), y)$ $\qquad$ x'

Evolution of L_∞ based adversarial examples

Fast Gradient Sign Method (FGSM)-generate adversarial examples with a single gradient step.[GS15]

$$x = x + \varepsilon \operatorname{sgn}(\nabla_x L(x, y, \theta))$$

Evolution of L_∞ based adversarial examples

Fast Gradient Sign Method (FGSM)-generate adversarial examples with a single gradient step.[GS15]

$$x = x + \varepsilon \operatorname{sgn}(\nabla_x L(x, y, \theta))$$

Random Start (R+FGSM): $x = x + Uniform(-\epsilon, \epsilon)$ [TK18]

Evolution of L_∞ based adversarial examples

Fast Gradient Sign Method (FGSM)-generate adversarial examples with a single gradient step.[GS15]

$$x = x + \varepsilon \operatorname{sgn}(\nabla_x L(x, y, \theta))$$

Random Start (R+FGSM): $x = x + Uniform(-\epsilon, \epsilon)$ [TK18]

Basic Iterative Method (BIM) - multiple smaller FGSM steps:

$$x^{k+1} = Proj_{x+S}(x^k + \varepsilon\, sgn\left(\nabla_{x^k} L\left(x^k, y, \theta\right)\right)) \quad [KGB17]$$

Increasing the strength of the attacks

Fast Gradient Sign Method (FGSM)-generate adversarial examples with a single gradient step.[GS15]

$$x = x + \varepsilon \operatorname{sgn}(\nabla_x L(x, y, \theta))$$

Random Start (R+FGSM): $x = x + Uniform(-\epsilon, \epsilon)$ [TK18]

Basic Iterative Method (BIM) - multiple smaller FGSM steps:

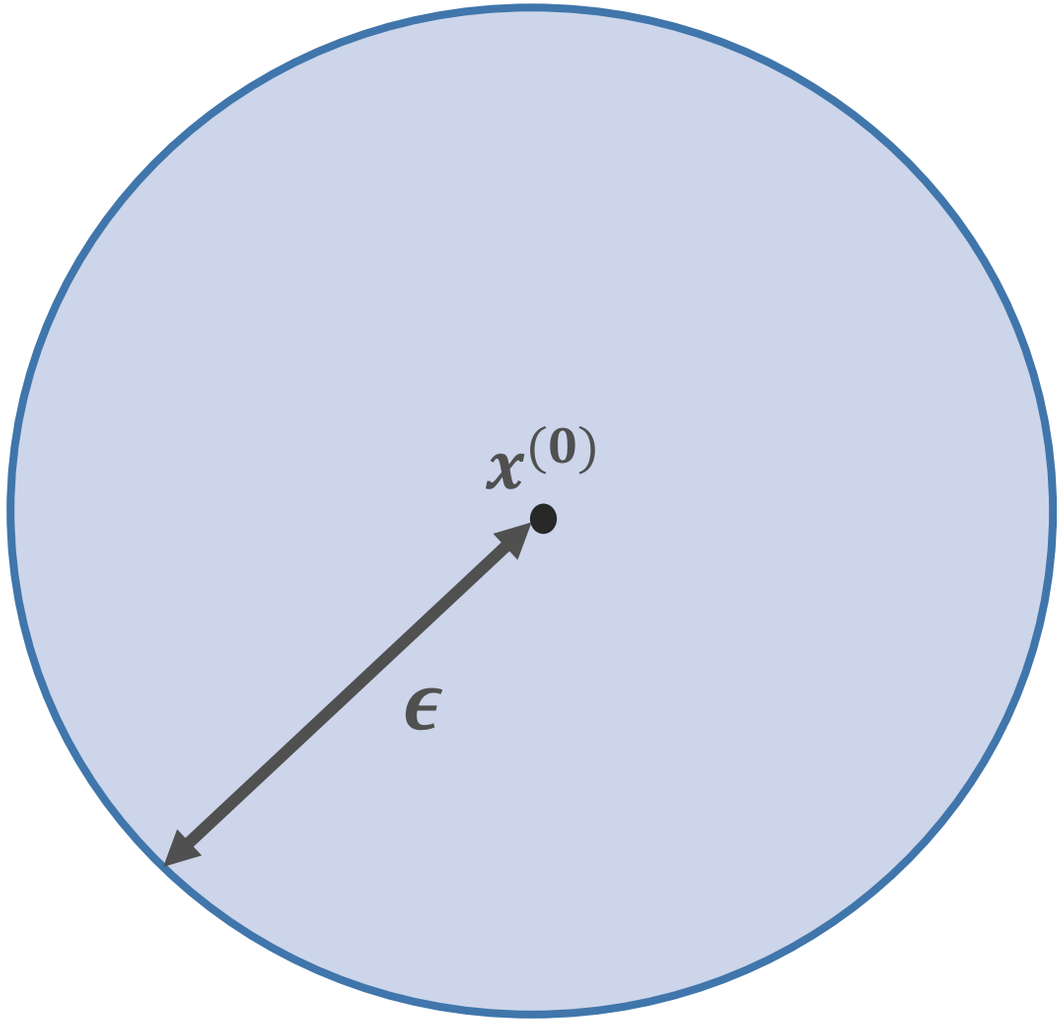
$$x^{k+1} = Proj_{x+S}(x^k + \varepsilon\, sgn\left(\nabla_{x^k} L\left(x^k, y, \theta\right)\right)) \quad [KGB17]$$

[MM18]

Projected Gradient Descent (PGD) - multiple random restarts.

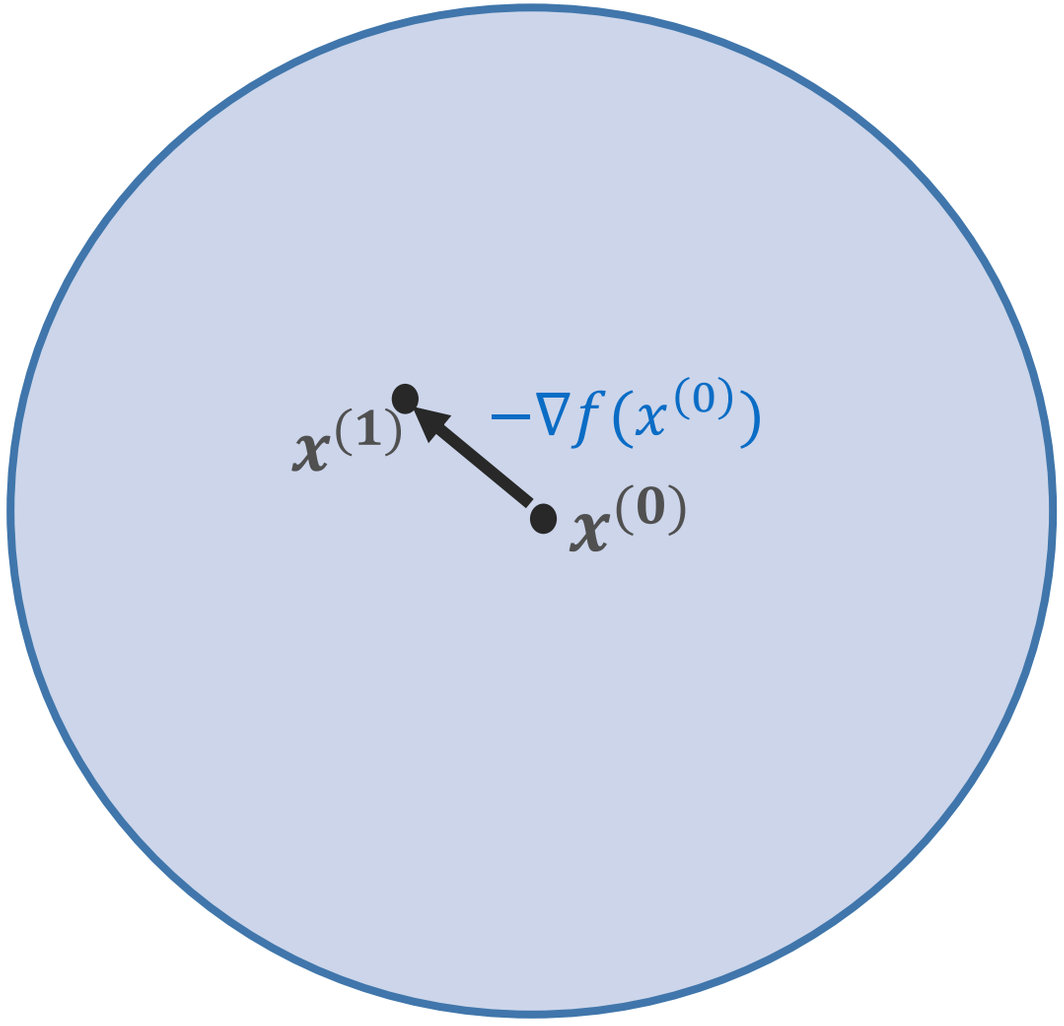
PGD: Projected Gradient Descent

[MM18]



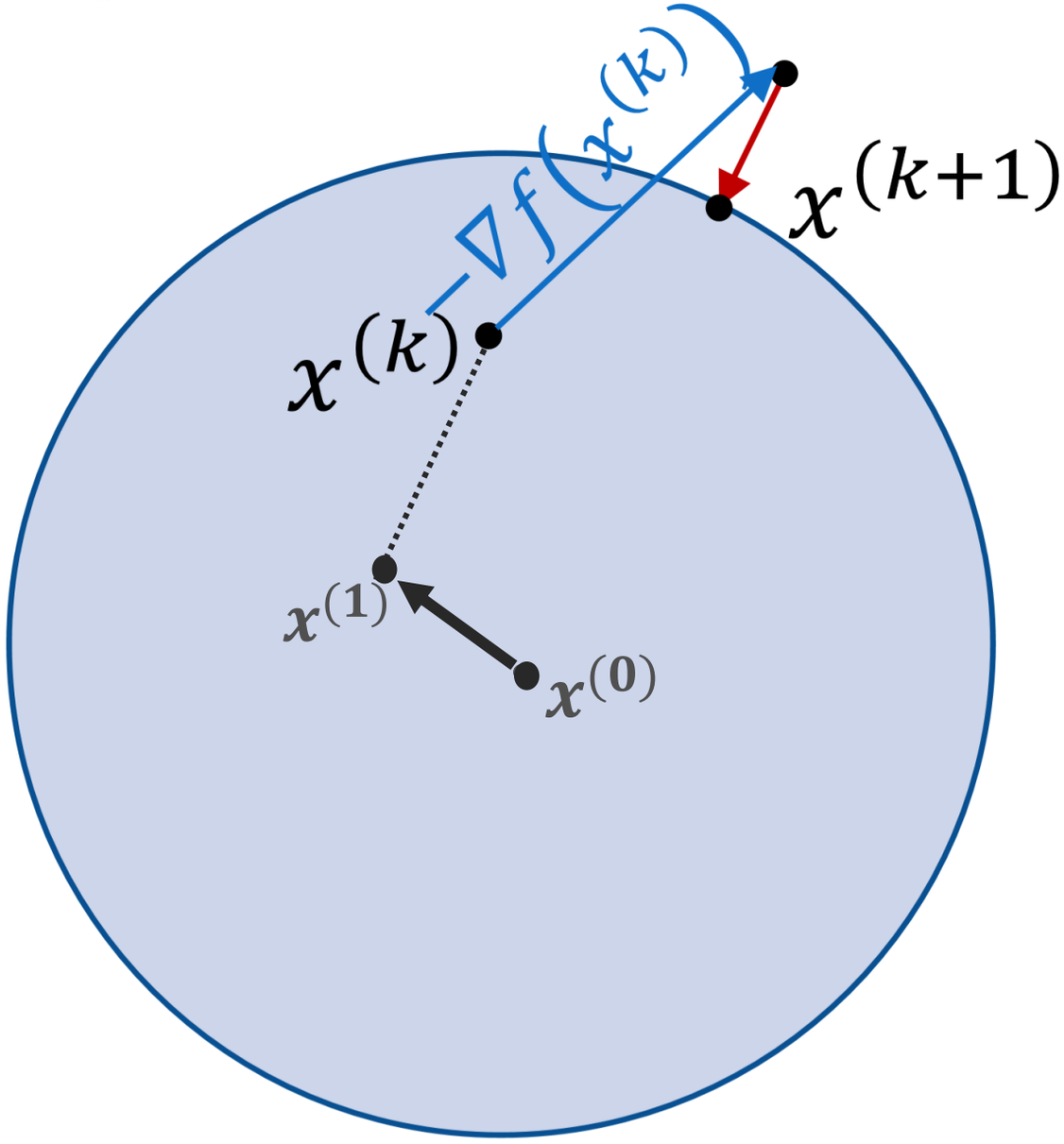
PGD: Projected Gradient Descent

[MM18]



PGD: Projected Gradient Descent

[MM18]

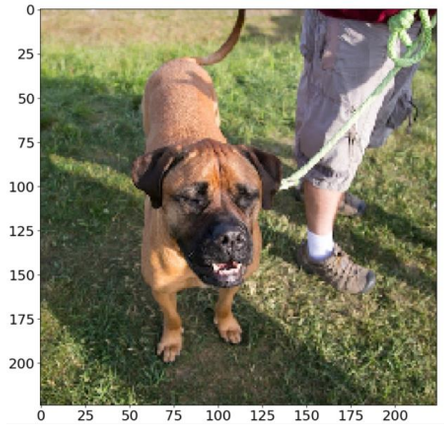


Crafting black-box attacks

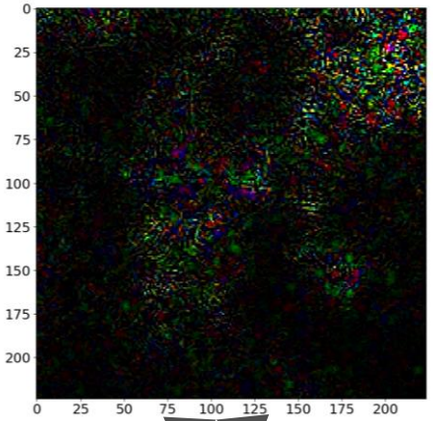
Bull mastiff

ML model

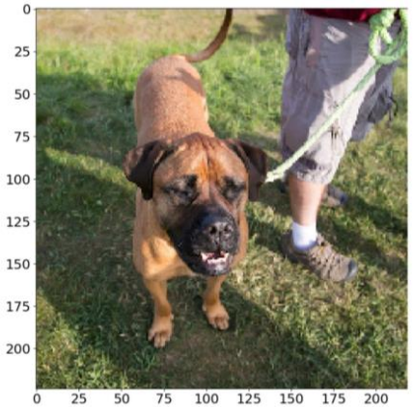
Tiger cat



+



=



x

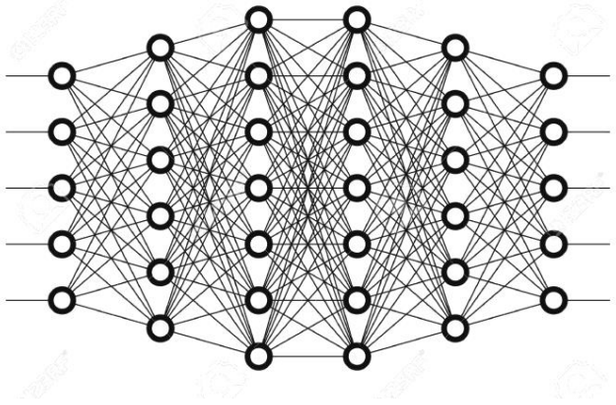
x'

Score -based

Decision -based

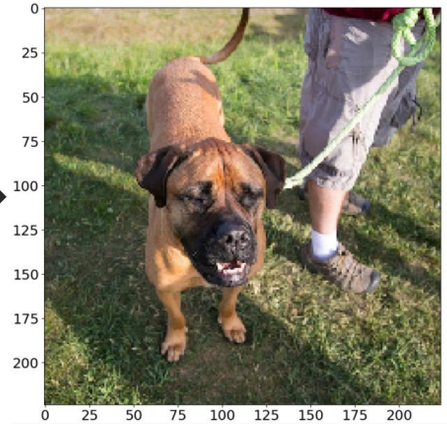
Transfer-based black-box attacks

[PM17]



Surrogate Model

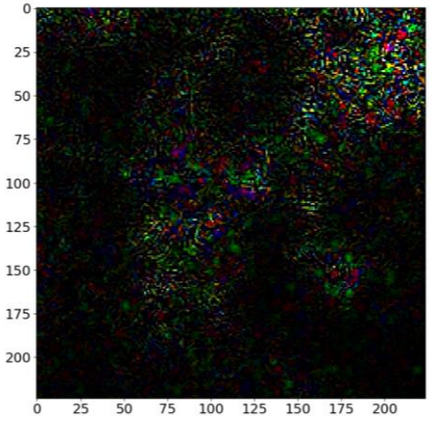
Bull mastiff



x

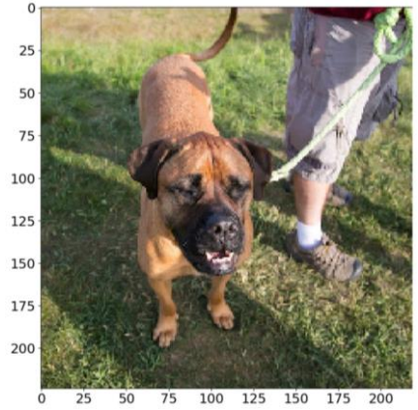
ML model

+



=

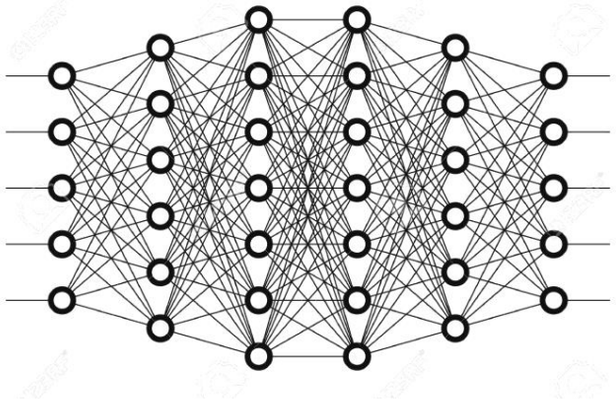
Tiger cat



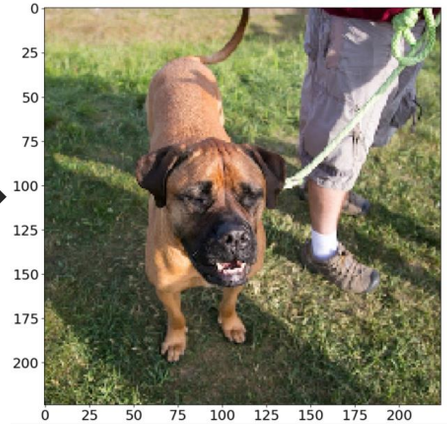
x'

Transfer-based black-box attacks

[PM17]



Surrogate Model



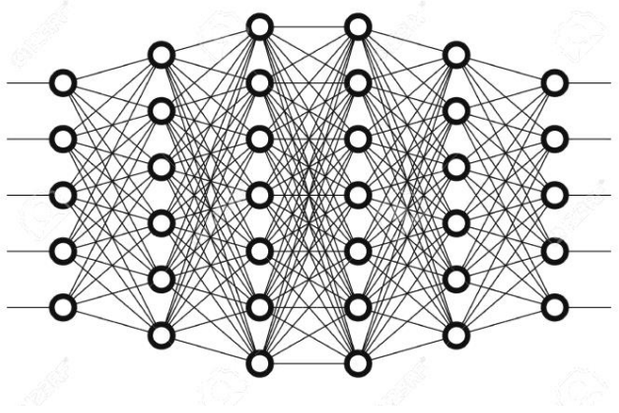
x

ML model

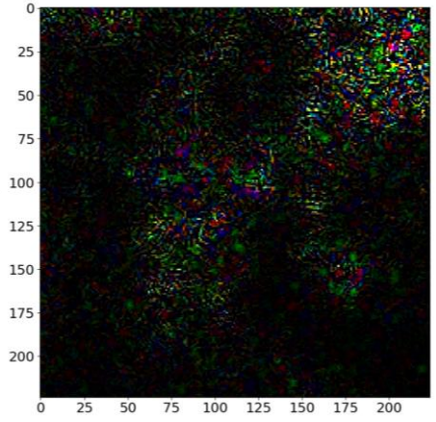
Attacked Model

Transfer-based black-box attacks

[PM16, PM17]

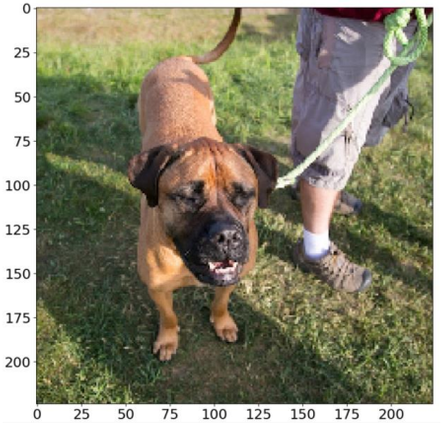


Surrogate Model



$$\nabla_x Loss(f(x;\theta), y)$$

+



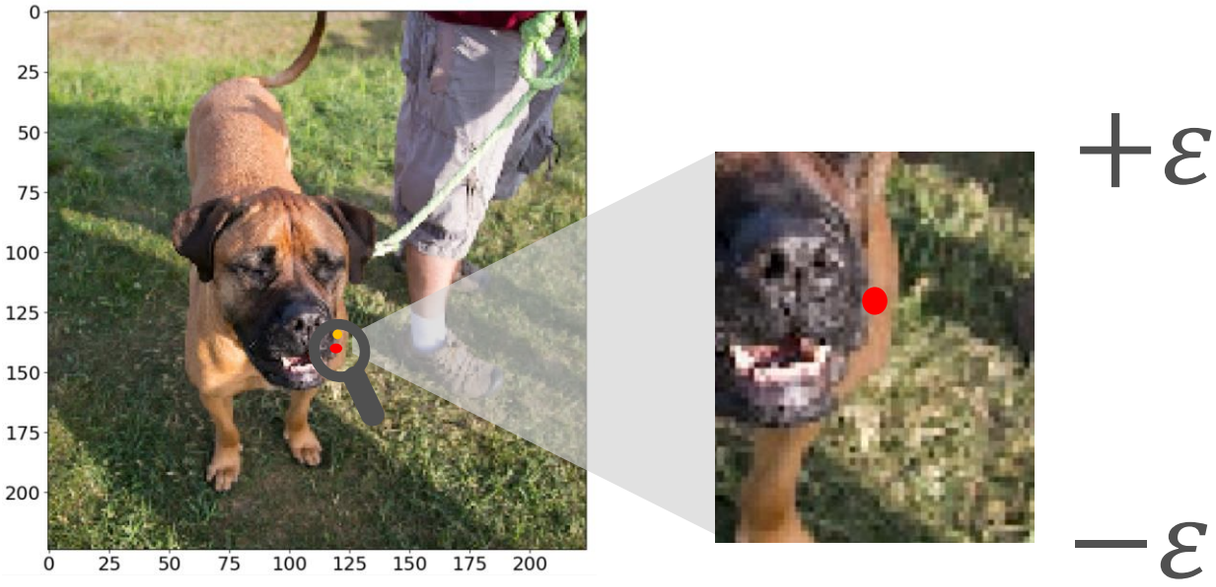
$$x$$

ML model

Attacked Model

SimBA: Simple Black-box Attack

[GG19]

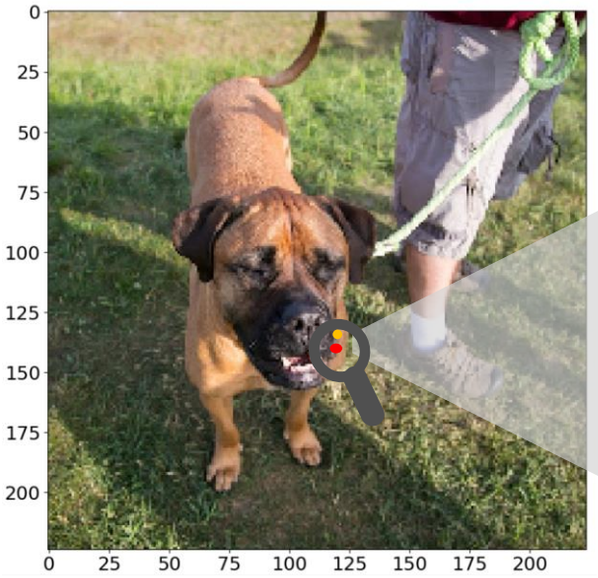


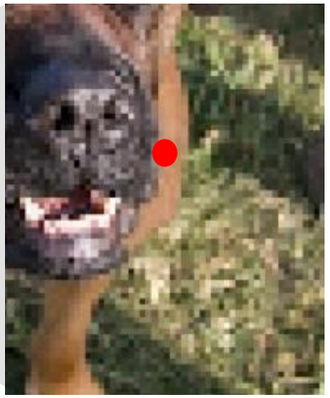
$+\varepsilon$

$-\varepsilon$

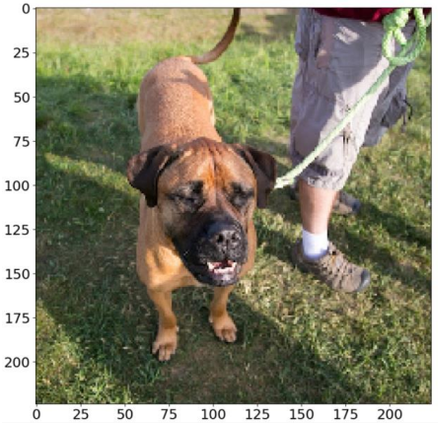
SimBA: Simple Black-box Attack

[GG19]





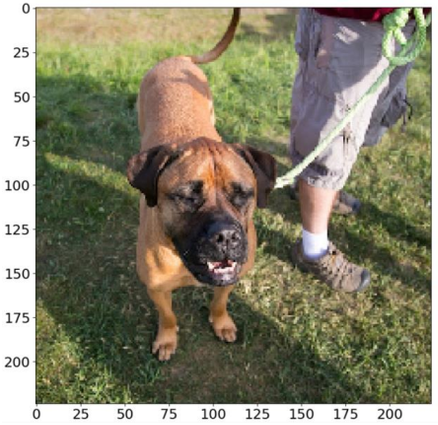
$+\varepsilon$



x_+

Classifier $p_y(x_+)$

$-\varepsilon$

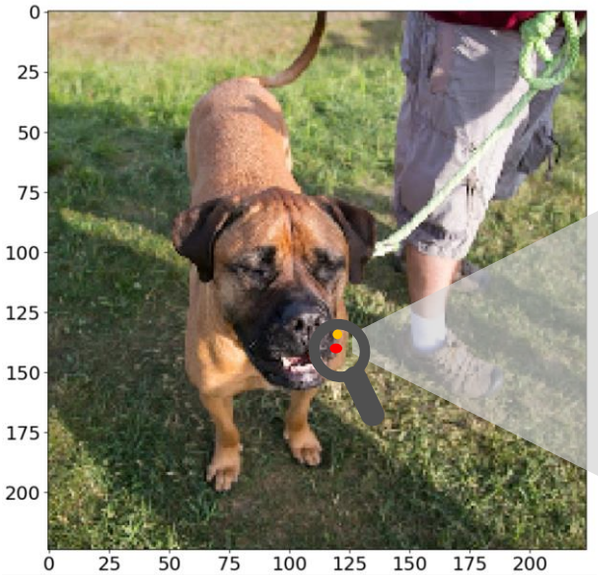


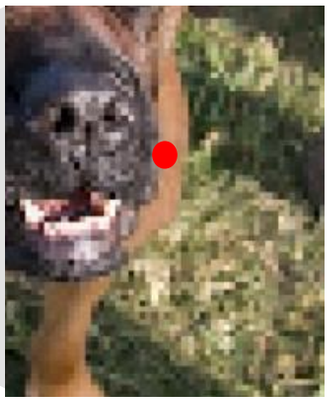
x_-

Classifier $p_y(x_-)$

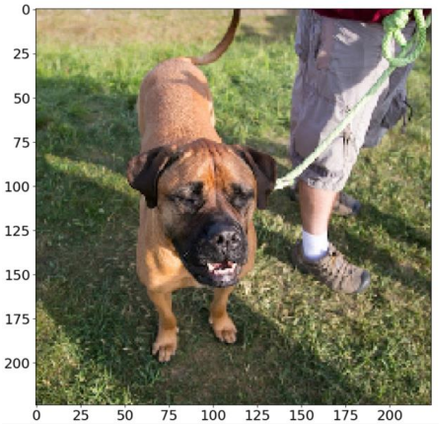
SimBA: Simple Black-box Attack

[GG19]





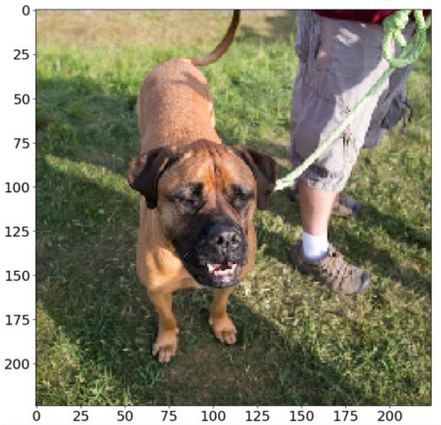
$+\varepsilon$



x_+

Classifier $p_y(x_+)$

$-\varepsilon$



x_-

Classifier $p_y(x_-)$

$$x' \leftarrow argmin\{\, p_y(x_+), p_y(x_-), p_y(x) \,\}$$

Efficient attack on Google Cloud Vision API

[GG19]

origin_54.BMP

Camera Accessory	87%
Product	82%
Hardware	67%
Optical Instrument	66%
Camera Lens	61%
Gun	61%
Product	58%
Weapon	53%

after_54.BMP

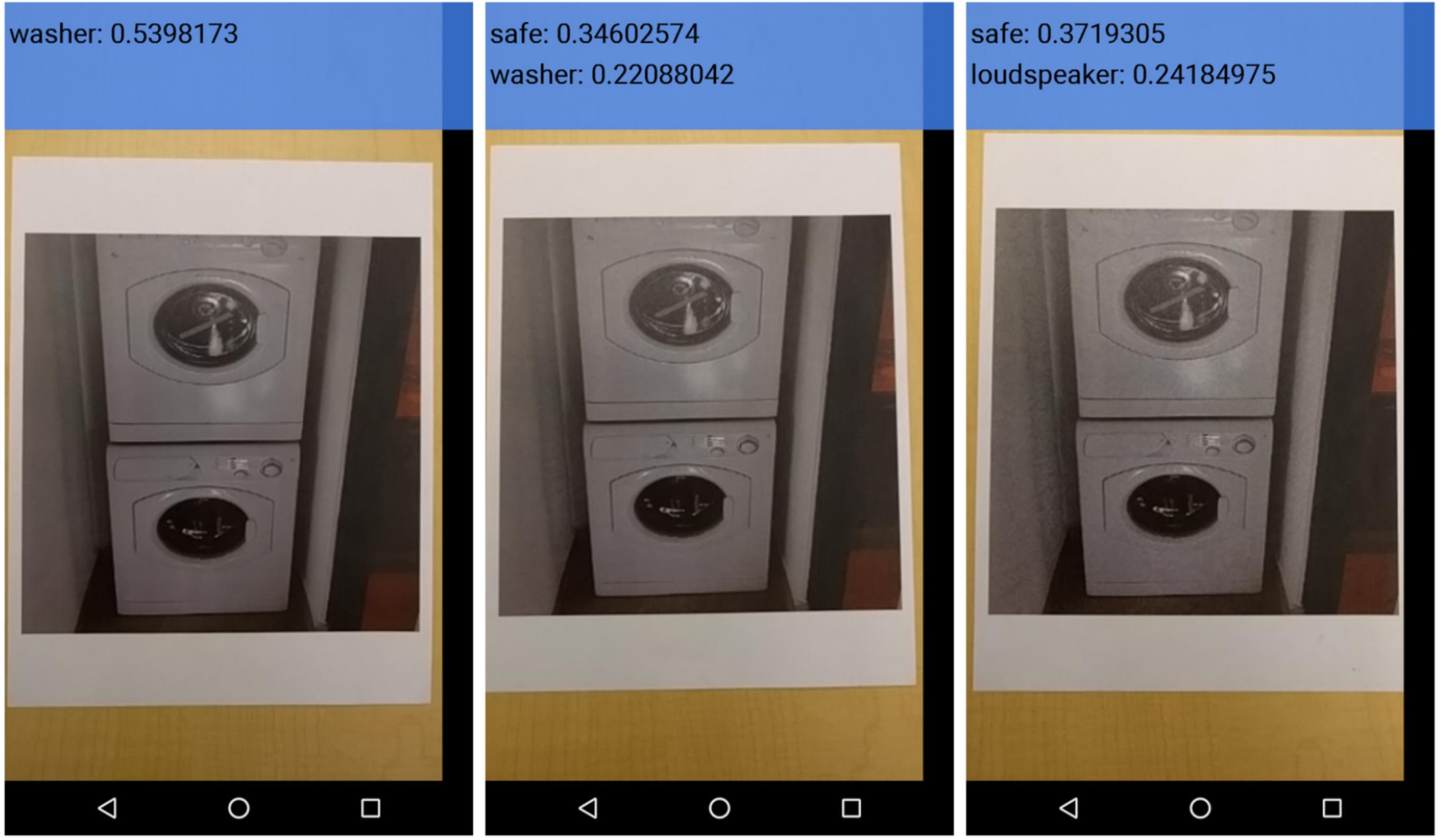
Weapon	94%
Gun	94%
Firearm	76%
Air Gun	65%
Trigger	63%
Optical Instrument	59%
Airsoft Gun	58%
Rifle	51%

SimBA: 70% success rate with max 5000 API calls (~$7.5)

Other best attack: 25% success rate under the same budget

Adversarial examples in the physical world

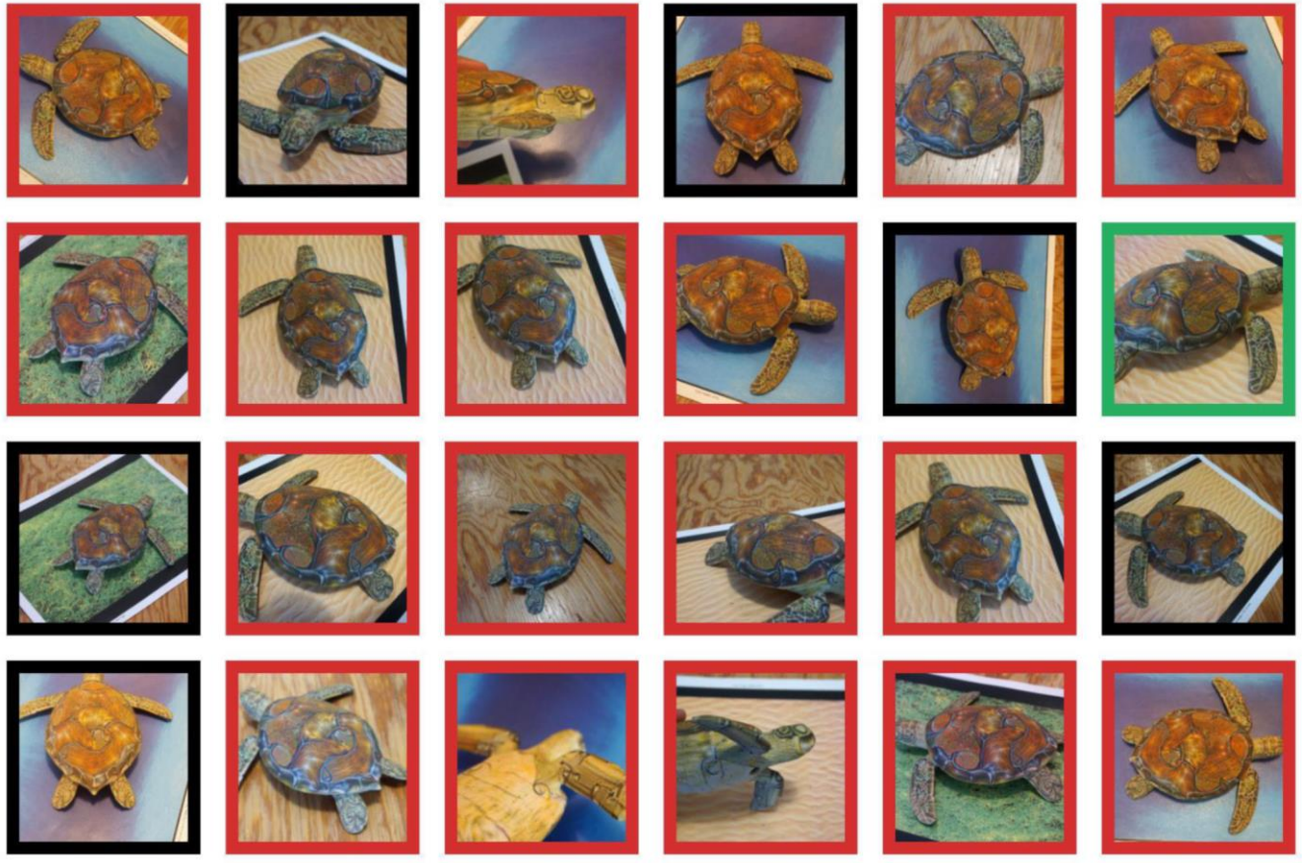
[KGB17]



Clean image | Adv. image, $\epsilon = 4$ | Adv. image, $\epsilon = 8$

Synthesizing Robust Adversarial Examples

[AE18]

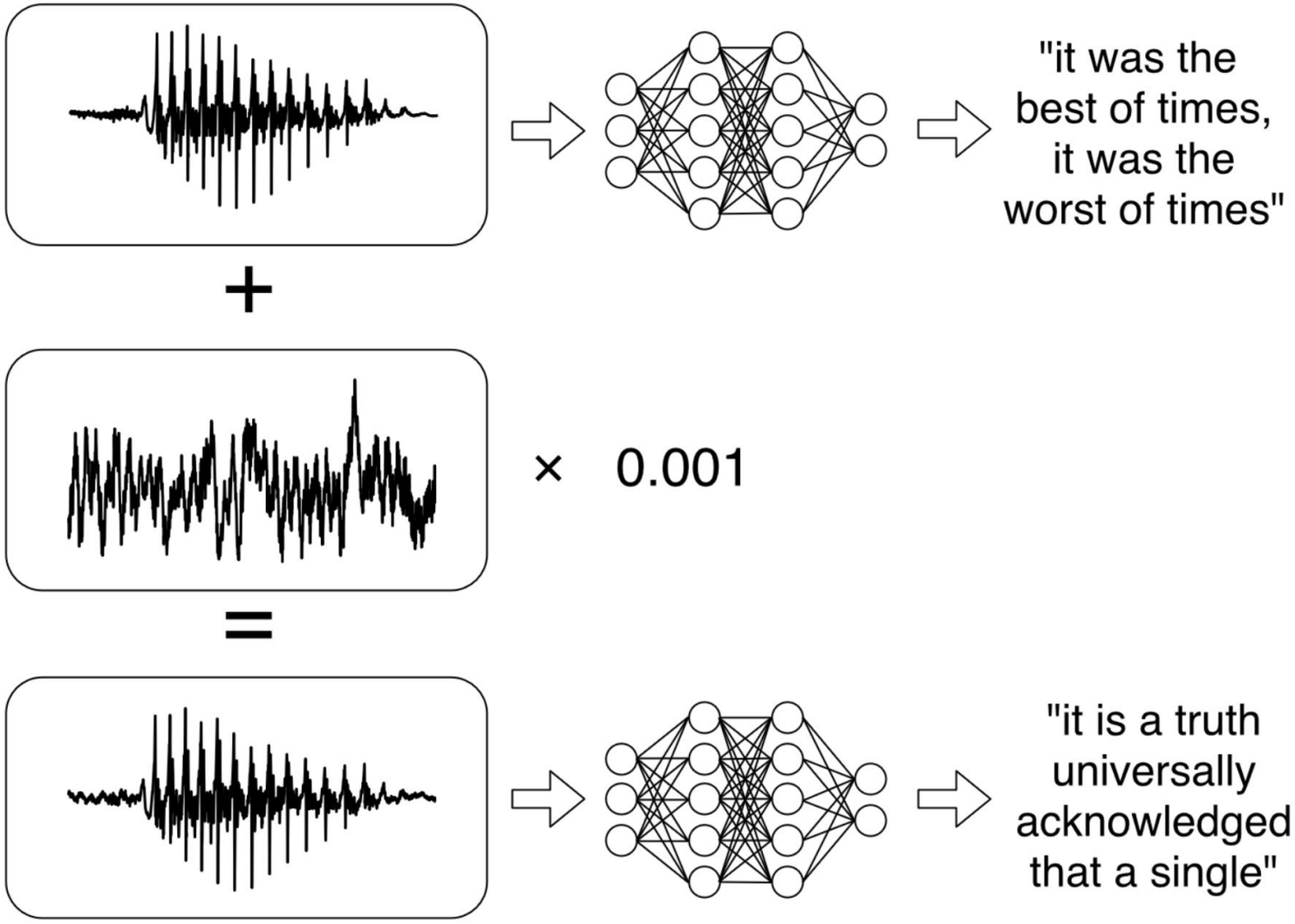


classified as turtle classified as rifle

classified as other

Audio adversarial examples

[CW18, SE20 (over-the-air)]



Four main categories of defenses

- **Preprocessing**: inputs are quantized, projected into different representations, compresses, perturbed, etc.
- **Stochasticity**: inputs and internal layers are randomized (usually with Gaussian or Uniform noise).
- **Manifold Projections**: input sample projected onto a learned manifold (uses generative models).
- **Adversarial Training**: augment training data with adversarial examples.

Four main categories of defenses

- **Preprocessing**: inputs are quantized, projected into different representations, compresses, perturbed, etc.
- **Stochasticity**: inputs and internal layers are randomized (usually with Gaussian or Uniform noise).
- **Manifold Projections**: input sample projected onto a learned manifold (uses generative models).
- **Adversarial Training**: augment training data with adversarial examples.

Four main categories of defenses

- **Preprocessing**: inputs are quantized, projected into different representations, compresses, perturbed, etc.
- **Stochasticity**: inputs and internal layers are randomized (usually with Gaussian or Uniform noise).
- **Manifold Projections**: input sample projected onto a learned manifold (uses generative models).
- **Adversarial Training**: augment training data with adversarial examples.

Four main categories of defenses

- **Preprocessing**: inputs are quantized, projected into different representations, compresses, perturbed, etc.
- **Stochasticity**: inputs and internal layers are randomized (usually with Gaussian or Uniform noise).
- **Manifold Projections**: input sample projected onto a learned manifold (uses generative models).
- **Adversarial Training**: augment training data with adversarial examples.

Four main categories of defenses

- **Preprocessing**: inputs are quantized, projected into different representations, compresses, perturbed, etc.
- **Stochasticity**: inputs and internal layers are randomized (usually with Gaussian or Uniform noise).
- **Manifold Projections**: input sample projected onto a learned manifold (uses generative models).
- **Adversarial Training**: augment training data with adversarial examples.

Defend models via adversarial training (AT)

1. Traditional training: minimize the loss with respect to θ

$$\min_{\theta} E_{(x,y)\sim D}[L(x, y, \theta)]$$

Adversarial training on adversarial inputs

1. Traditional training: minimize the loss with respect to θ

$$\min_{\theta} E_{(x,y)\sim D}[L(x, y, \theta)]$$

2. Adversarial example: maximize the loss using x and δ

$$\max_{\delta \in S} L(x + \delta, y, \theta)]$$

Train robust models: adversarial training

1. Traditional training: minimize the loss with respect to θ

$$\min_{\theta} E_{(x,y)\sim D}[L(x, y, \theta)]$$

2. Adversarial example: maximize the loss using x and δ

$$\max_{\delta \in S} L(x + \delta, y, \theta)]$$

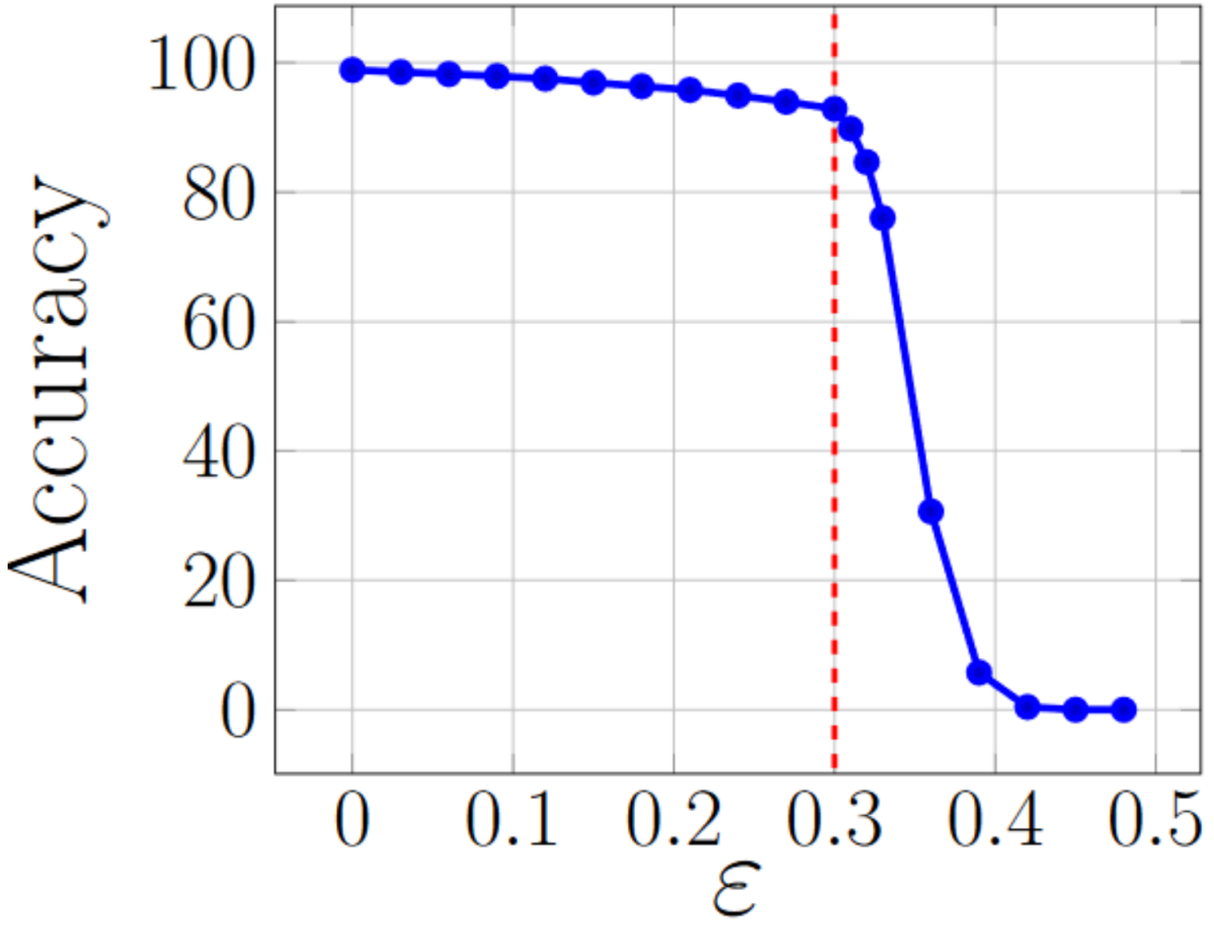
3. Adversarial training:

$$\min_{\theta} E_{(x,y)\sim D}\left[\max_{\delta \in S} L(x + \delta, y, \theta)\right]$$

Robustness of Adversarially Trained models

[MM18]

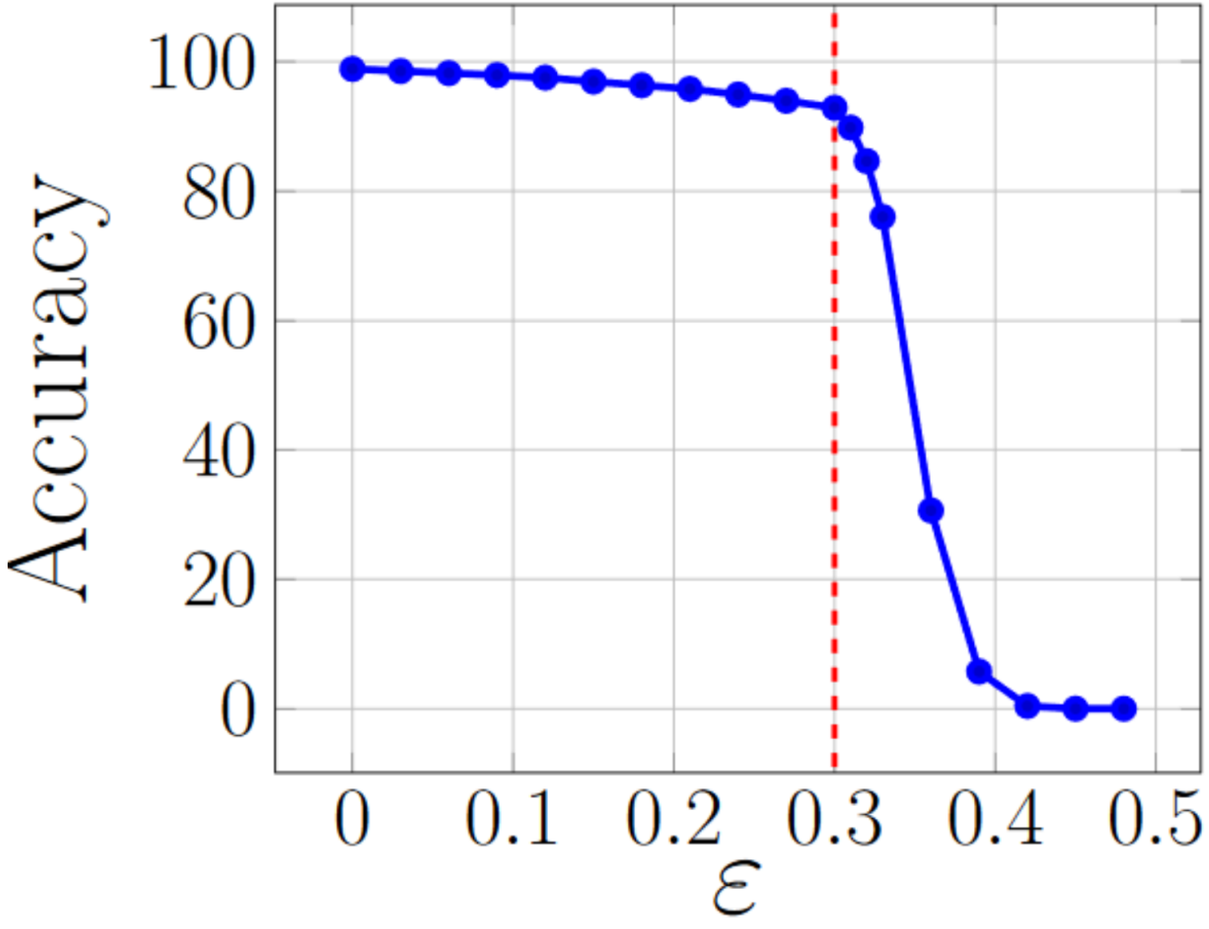
MNIST



Strength of the PGD attack

Robustness of Adversarially Trained models

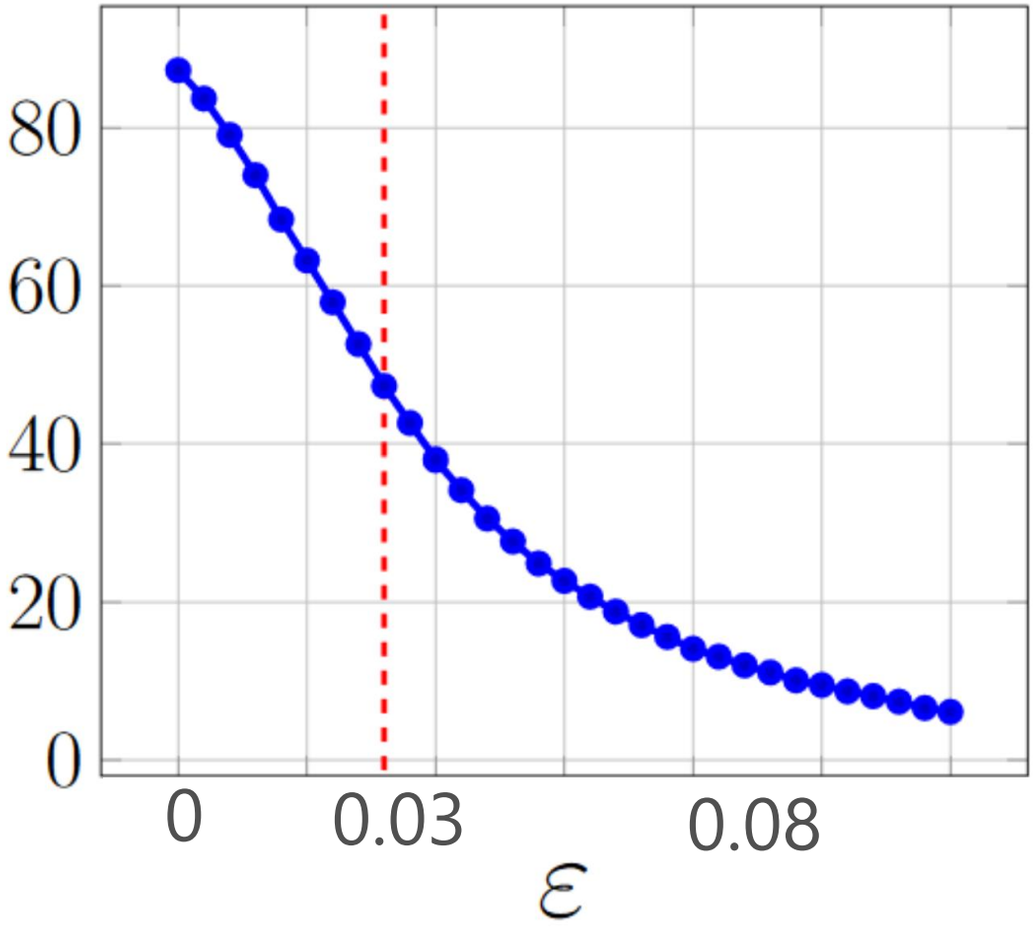
MNIST



Strength of the PGD attack

CIFAR10

[MM18]



Strength of the PGD attack

Adversarial Training vs Other Defenses

[PM17 (gradient masking), AC18 (obfuscated gradients)]

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

* *Propose combining adversarial training*

** *Imperfect Projected Gradient Descent*

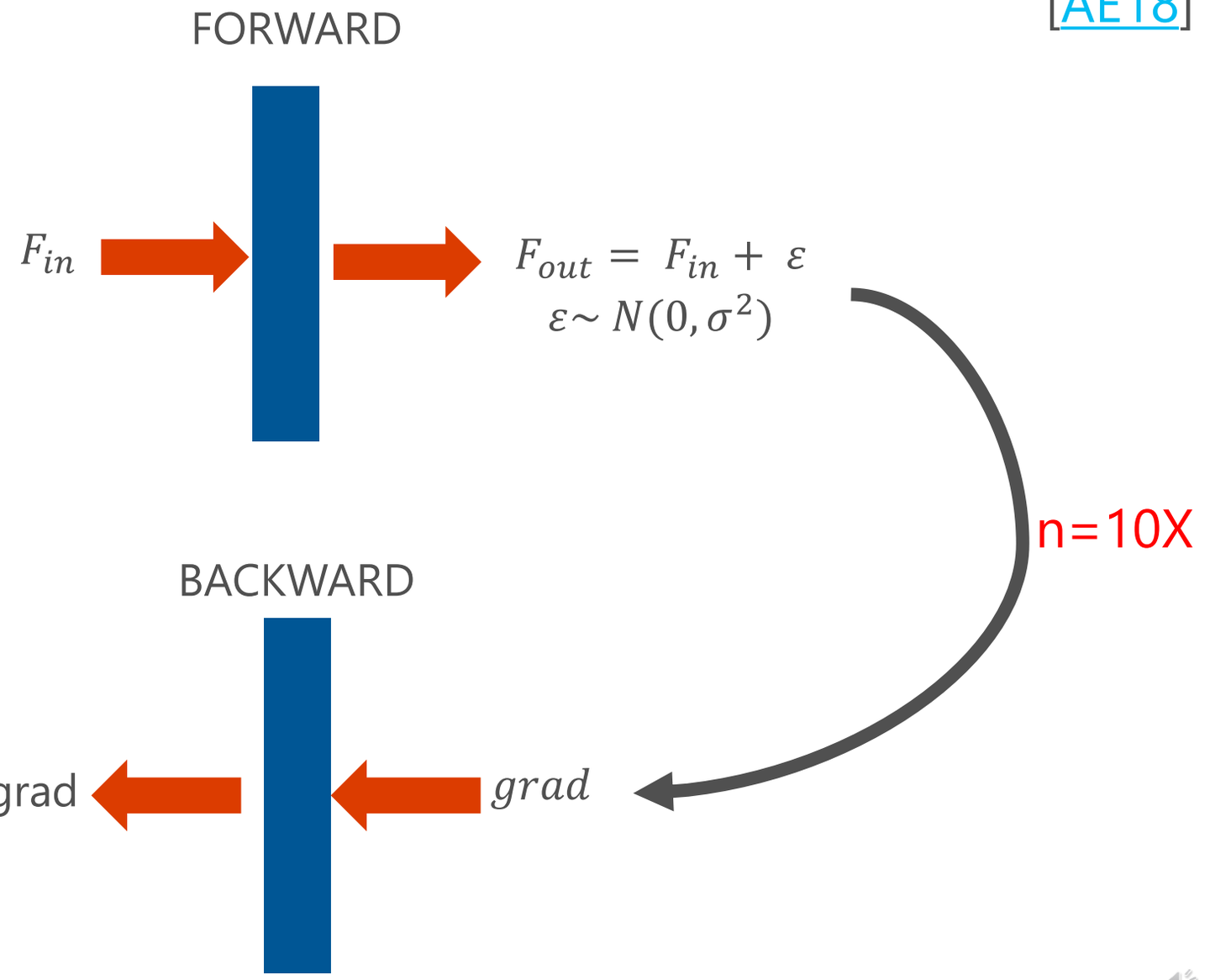
Adaptive attack with more accurate gradient

[AE18]

EOT

(Expectation Over Transformation)

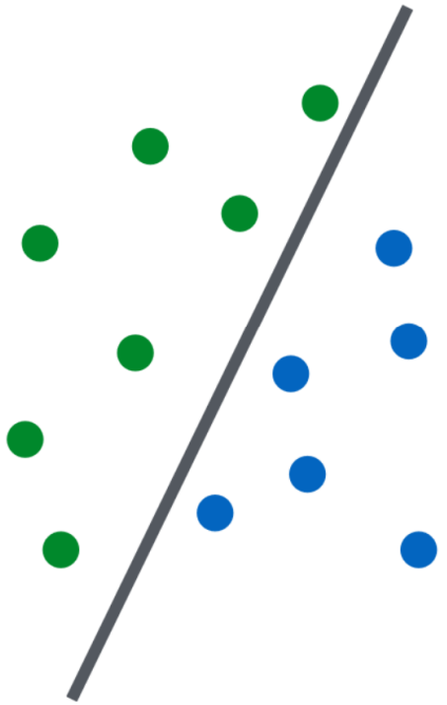
$$\nabla \mathbb{E}_{t\sim T} f\big(t(x)\big) = \mathbb{E}_{t\sim T} \nabla f(t(x))$$



$$\frac{\partial L}{\partial F_{in}} = \frac{grad\_noise}{10}$$

Natural vs Adversarial Decision Boundaries

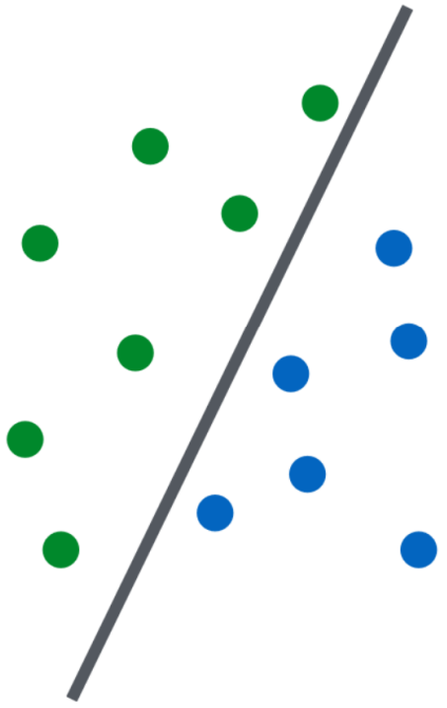
[MM18]



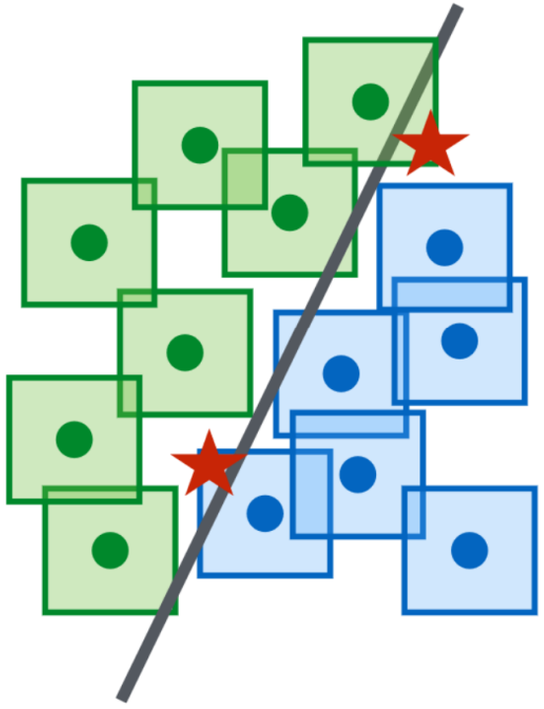
1) Simple decision boundary

Natural vs Adversarial Decision Boundaries

[MM18]



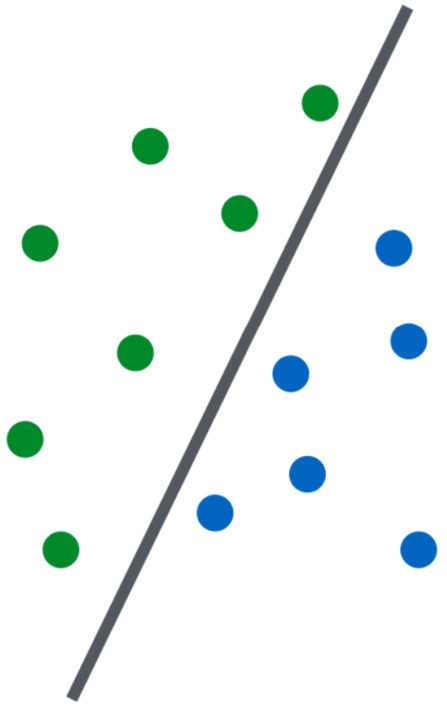
1) Simple decision boundary



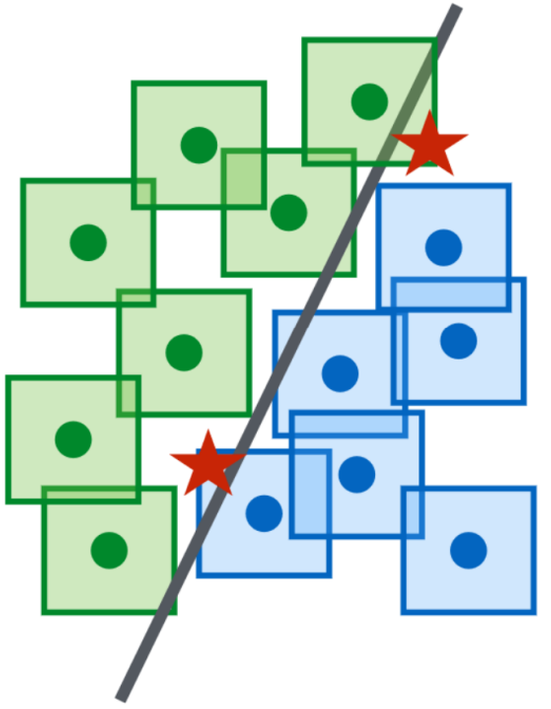
2) Adversarial examples ★ within L_∞ balls

Natural vs Adversarial Decision Boundaries

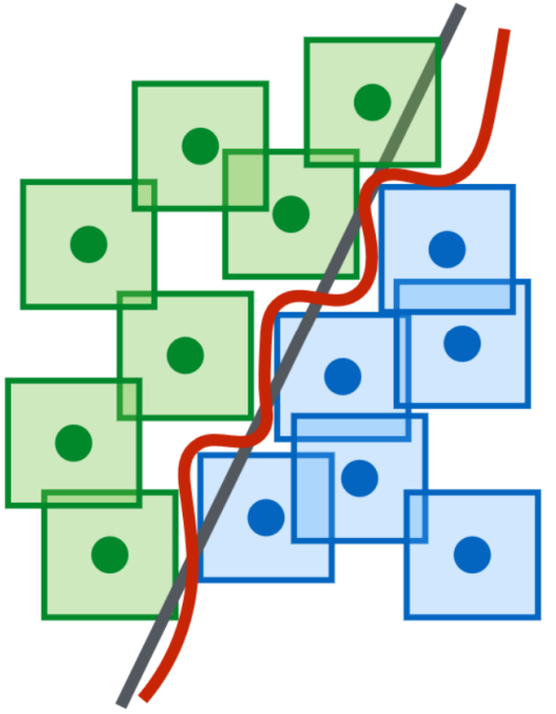
[MM18]



1) Simple decision boundary



2) Adversarial examples ★ within L_∞ balls

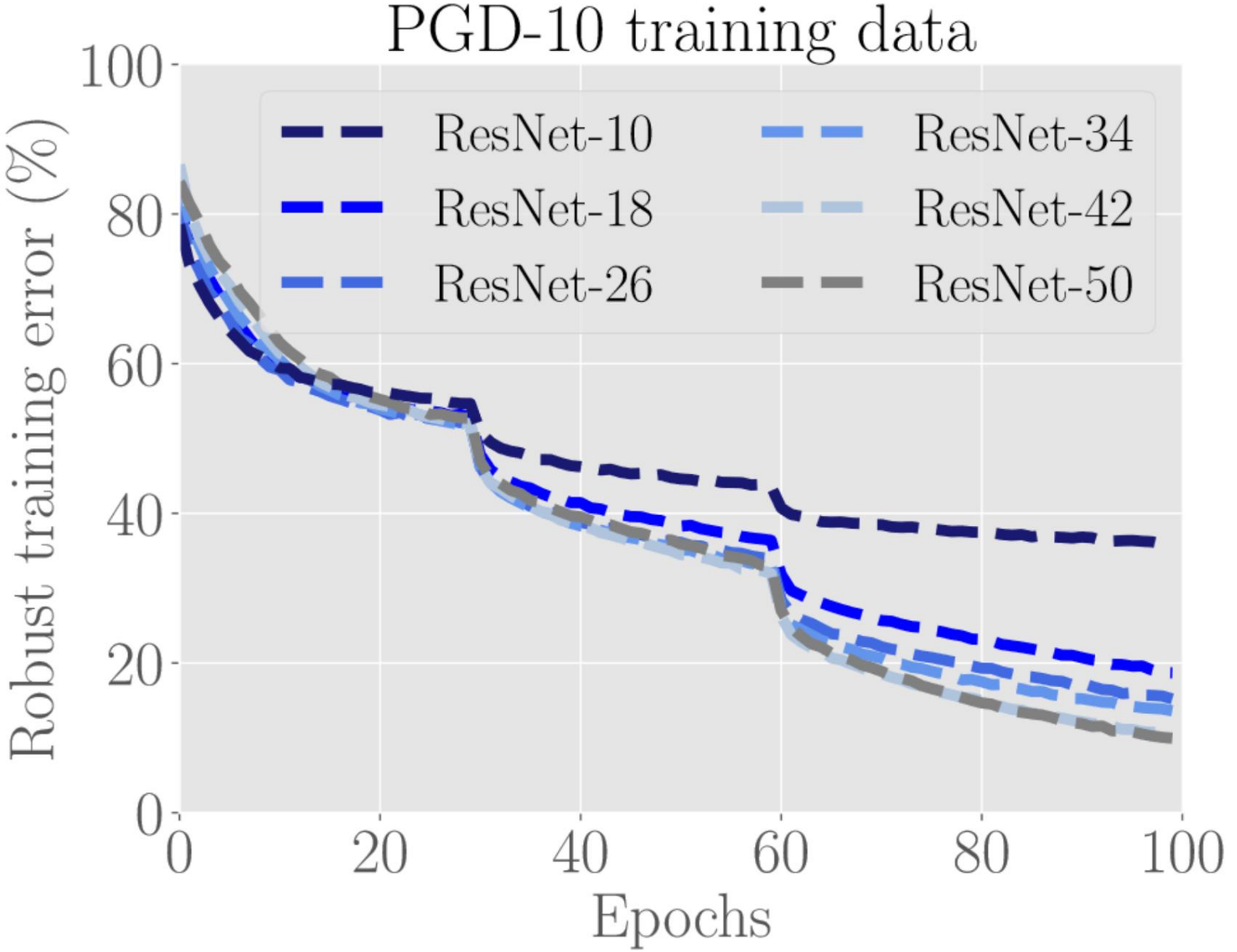


3) Training on ★ requires a complex decision boundary

Higher network capacity enables more robust classification

Diminishing returns for larger networks

[GA21]



Over-parameterized deep networks for natural data.

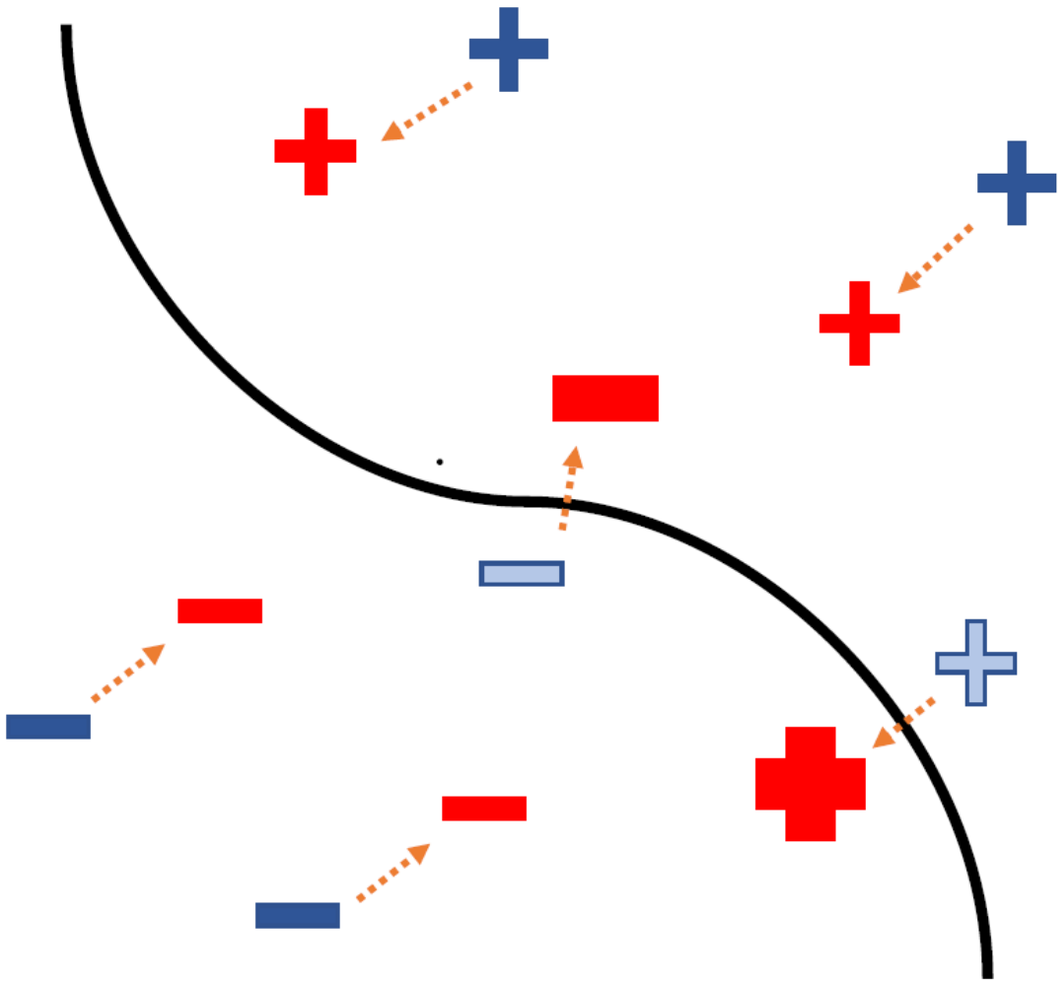
Not sufficient for adversarial training.

Adversarial data smooth the neighborhood of natural data - it consumes much more of model capacity.

How can we better utilize provided model capacity?

The closer to boundary the larger weight

[GA21]



Large-weight adversarial data

Small-weight adversarial data

Guarded data

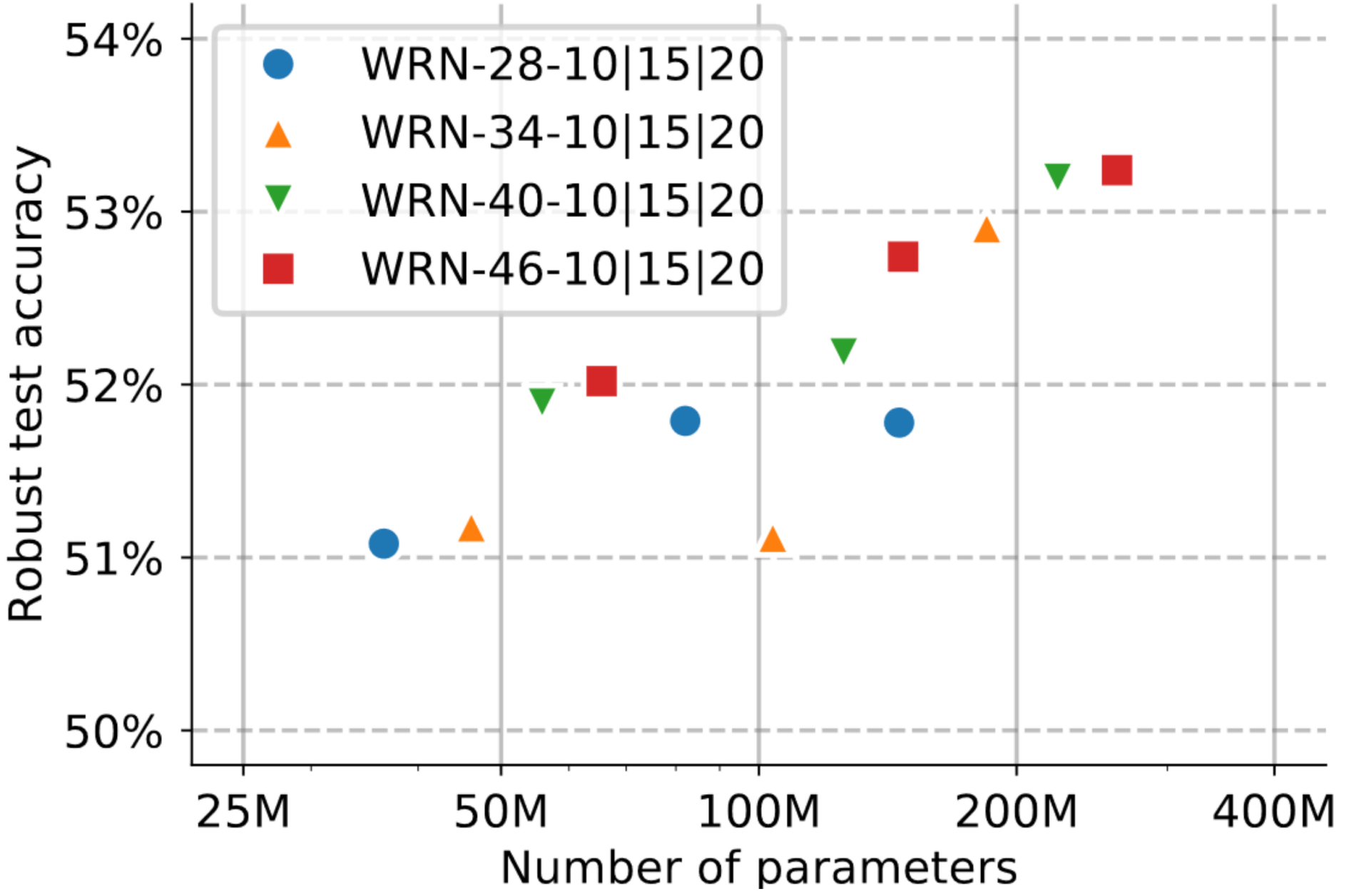
Attackable data

Decision boundary

Adversary direction

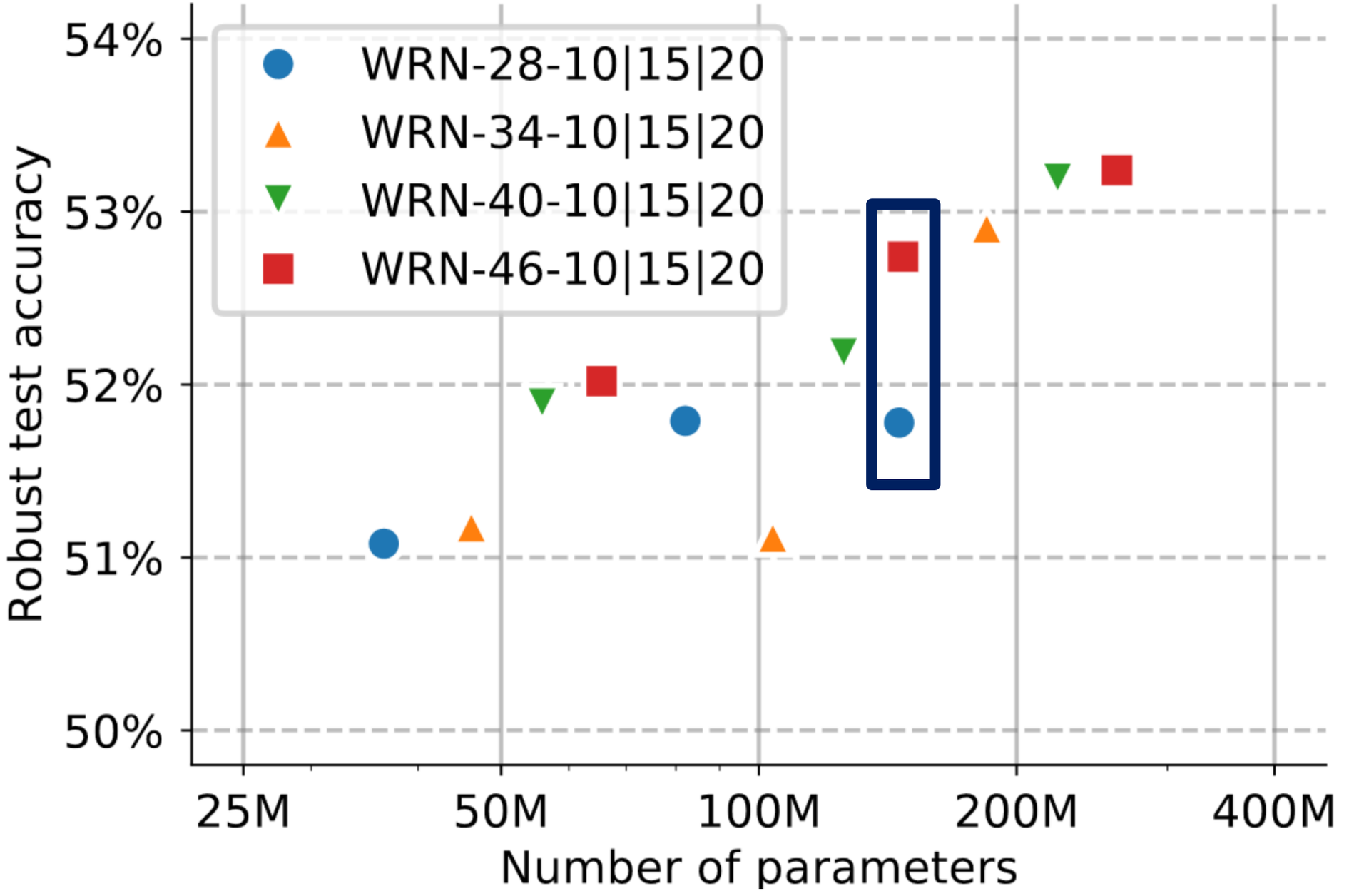
Deeper models can perform better [GQ20]

CIFAR10



Deeper models can perform better [GQ20]

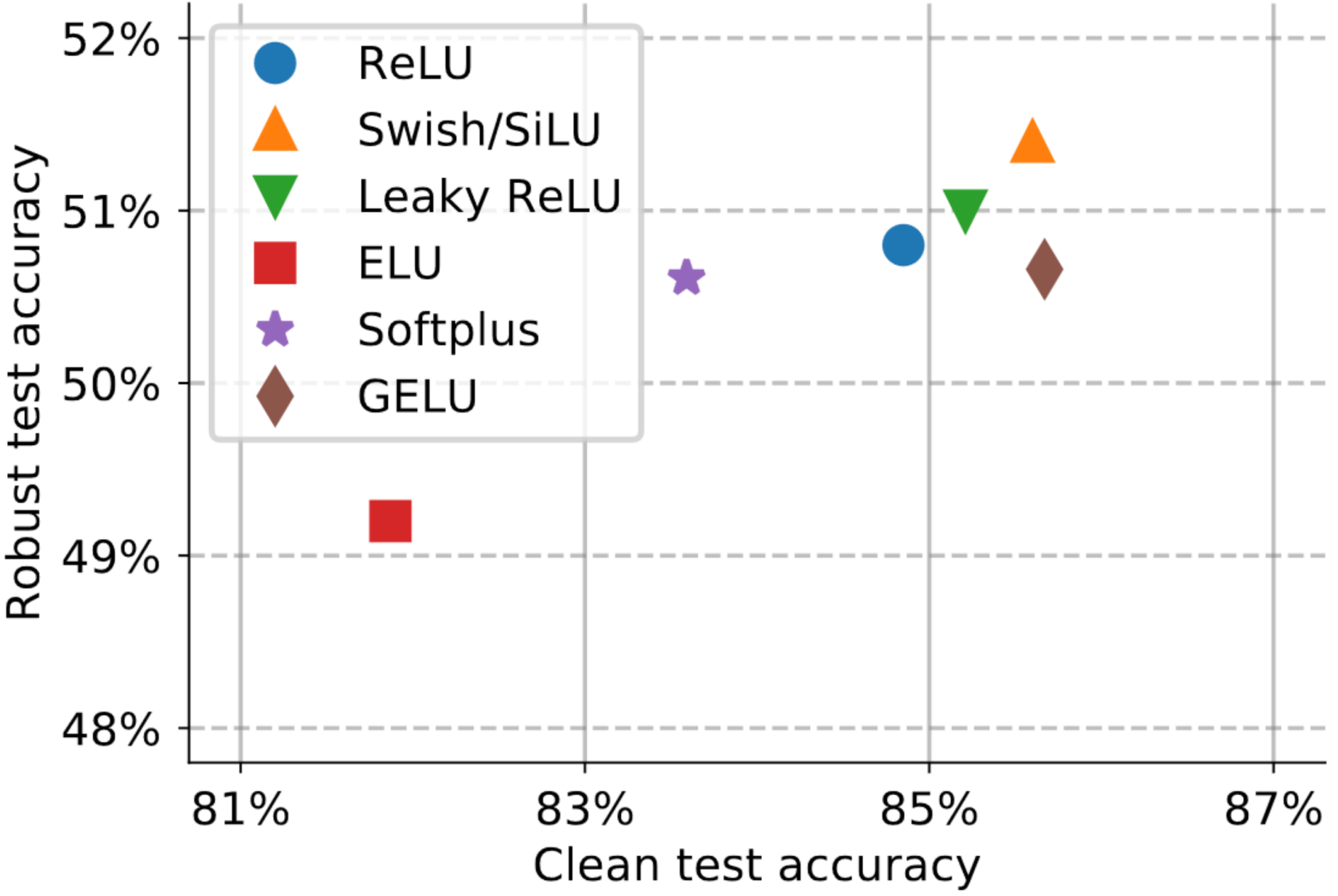
CIFAR10



Smooth activations can boost robustness

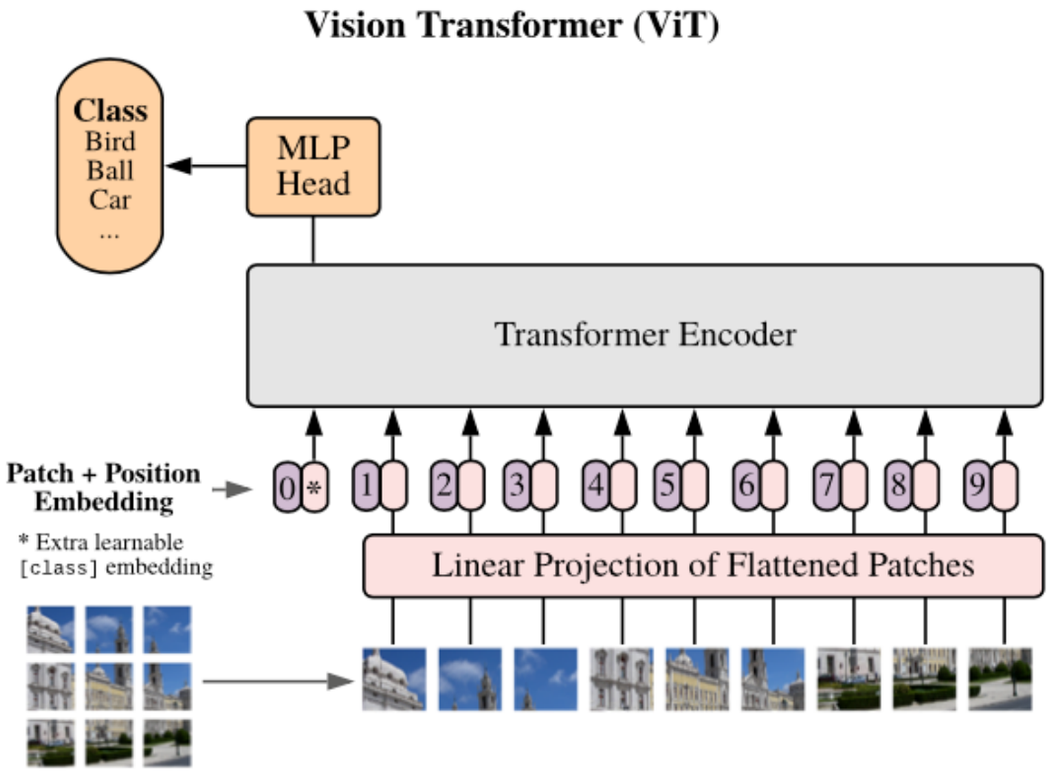
[GQ20]

CIFAR10



New architectures needed for Adv. Training

[AI21,GA21]



Deep or wide ConvNets - tremendous number of parameters make the decision boundary complicated but inhibit the optimization.

New **network architectures catering to local smoothing via adversarial training** are needed (e.g., Vision Transformer).

Robust generalization requires more data

- Additional data from ImageNet can improve robust accuracy of models trained on CIFAR10. [SS18] [HL19]
- Augment CIFAR10 with 200K and 500K unlabeled images sourced from 80 Million Tiny Images. [UB19, CR20]

Robust generalization requires more data

- Additional data from ImageNet can improve robust accuracy of models trained on CIFAR10. [HL19] [SS18]
- Augment CIFAR10 with 200K and 500K unlabeled images sourced from 80 Million Tiny Images. [UB19, CR20]

CIFAR10, ℓ_∞ based attack $\epsilon = 0.031$

Quantity of Data	Clean Accuracy (%)	Robust Accuracy (%)
200K	90.95	57.29
500K	90.68	59.12
1M	91.00	58.89

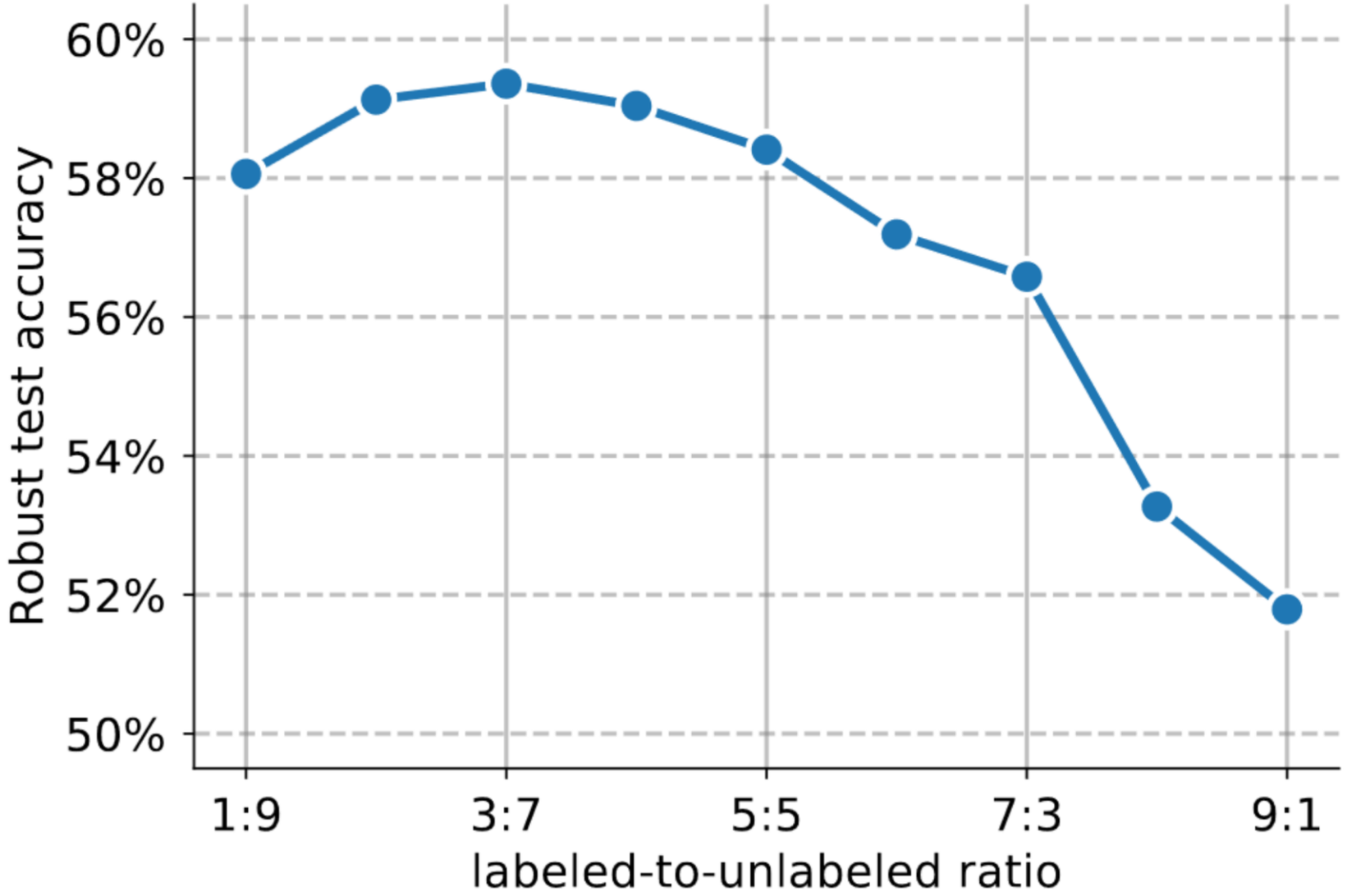
[GQ20]

How to use the additional data?

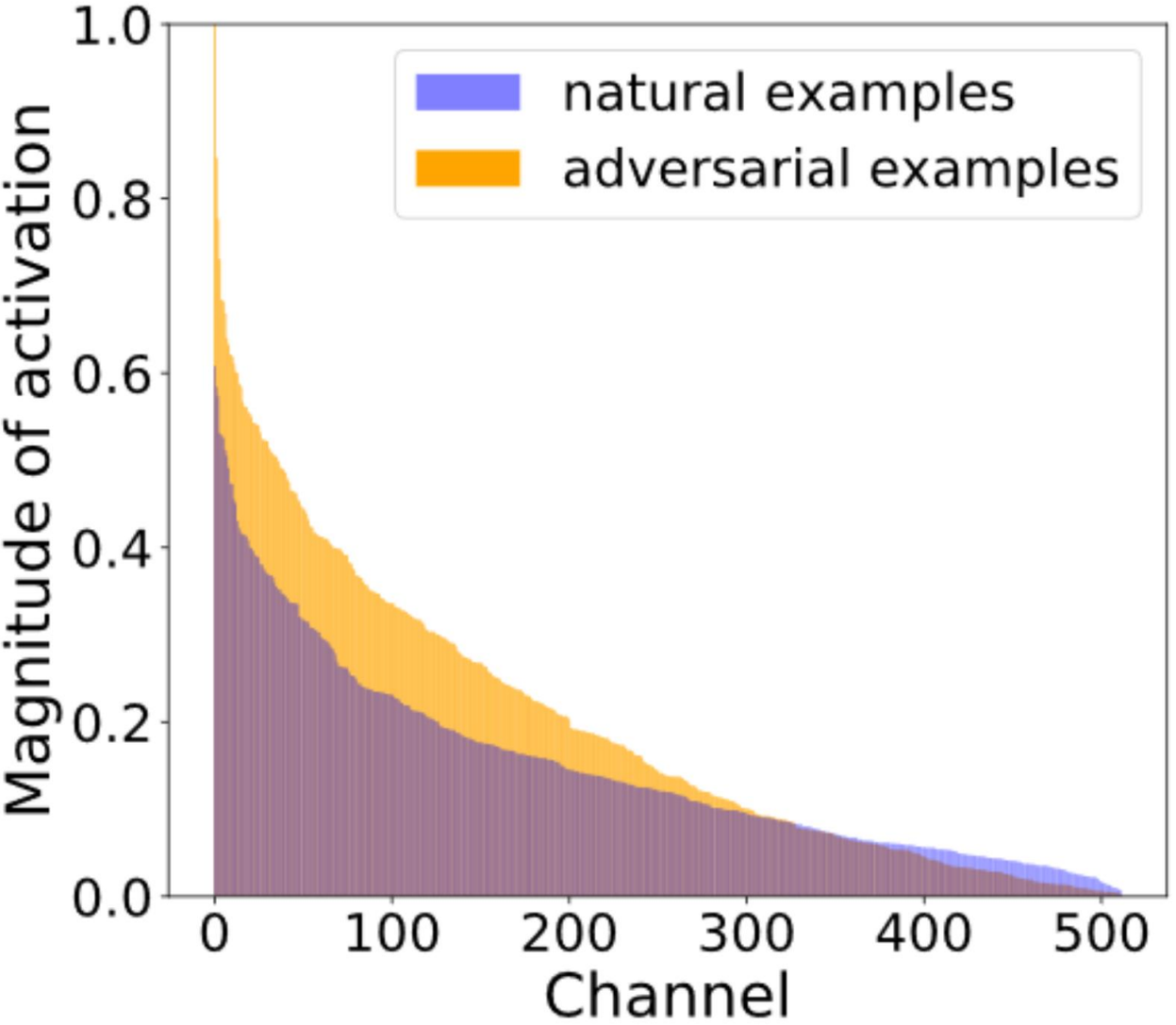
[GQ20]

How to select the best ratio of labeled-to-unlabeled data per batch?

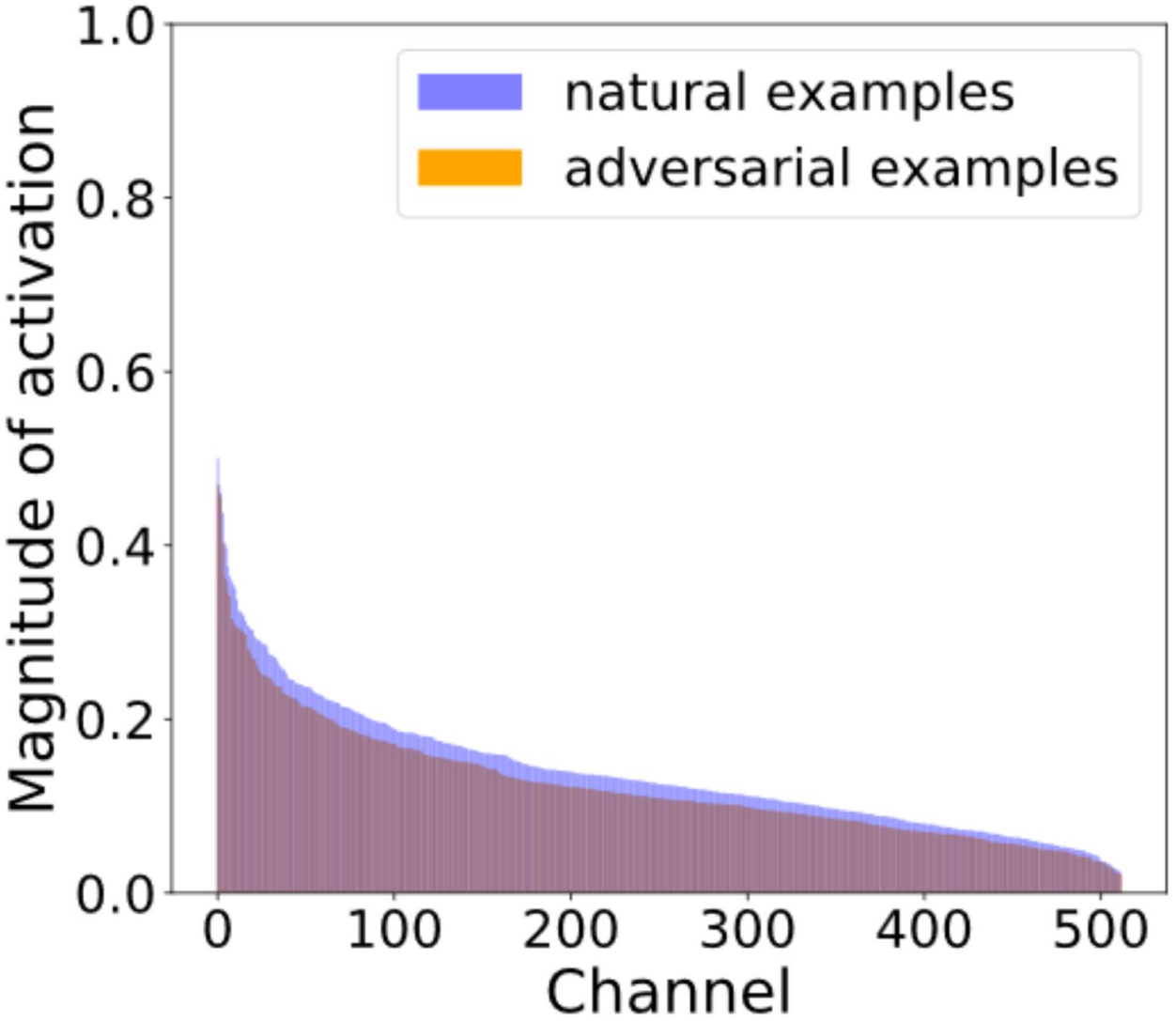
CIFAR10, ℓ_∞based attack $\epsilon = 0.031$, 500K unlabeled, 50K labeled images



Channel-wise activation magnitude [IA21]



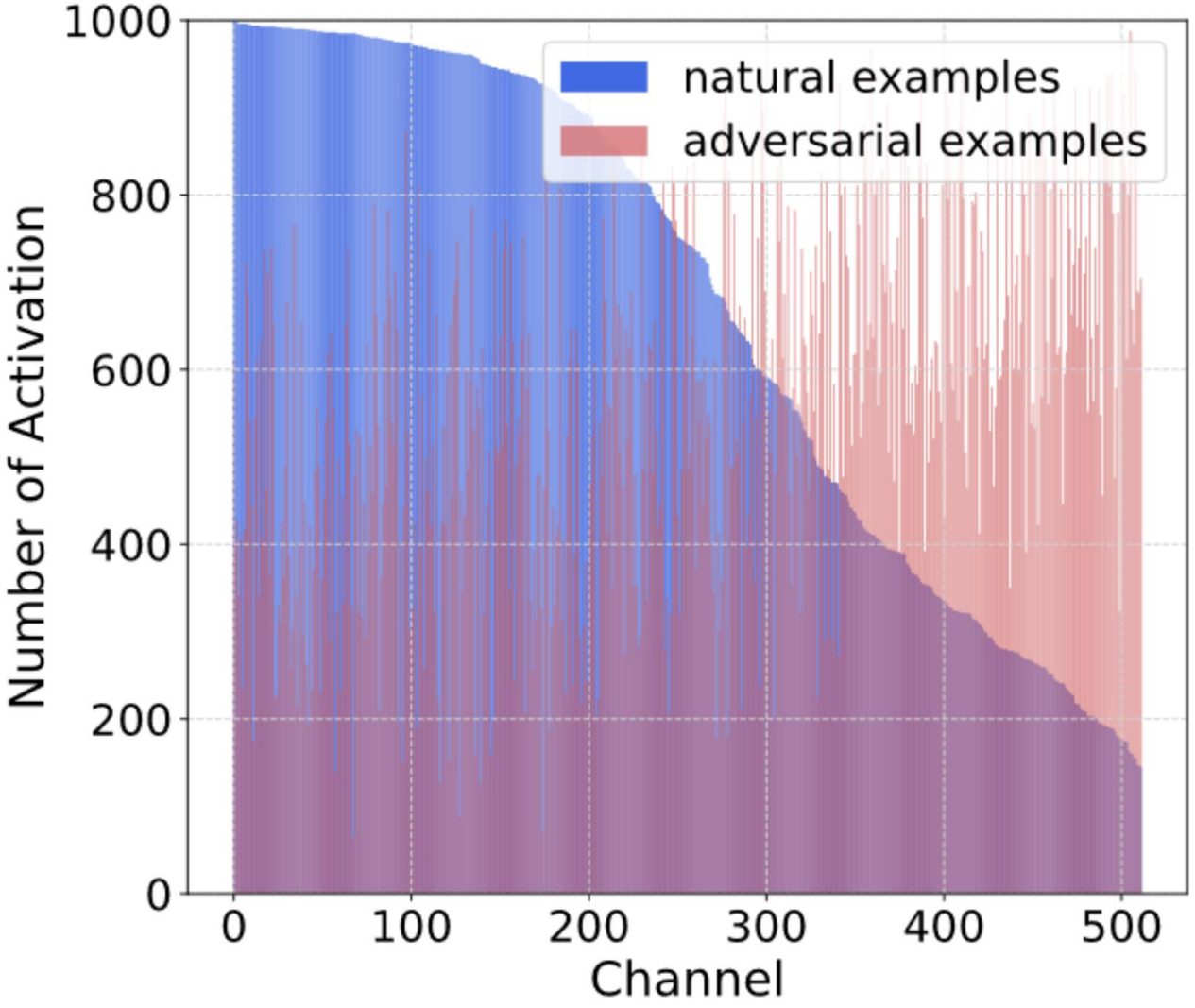
(a) ResNet-18 (STD)



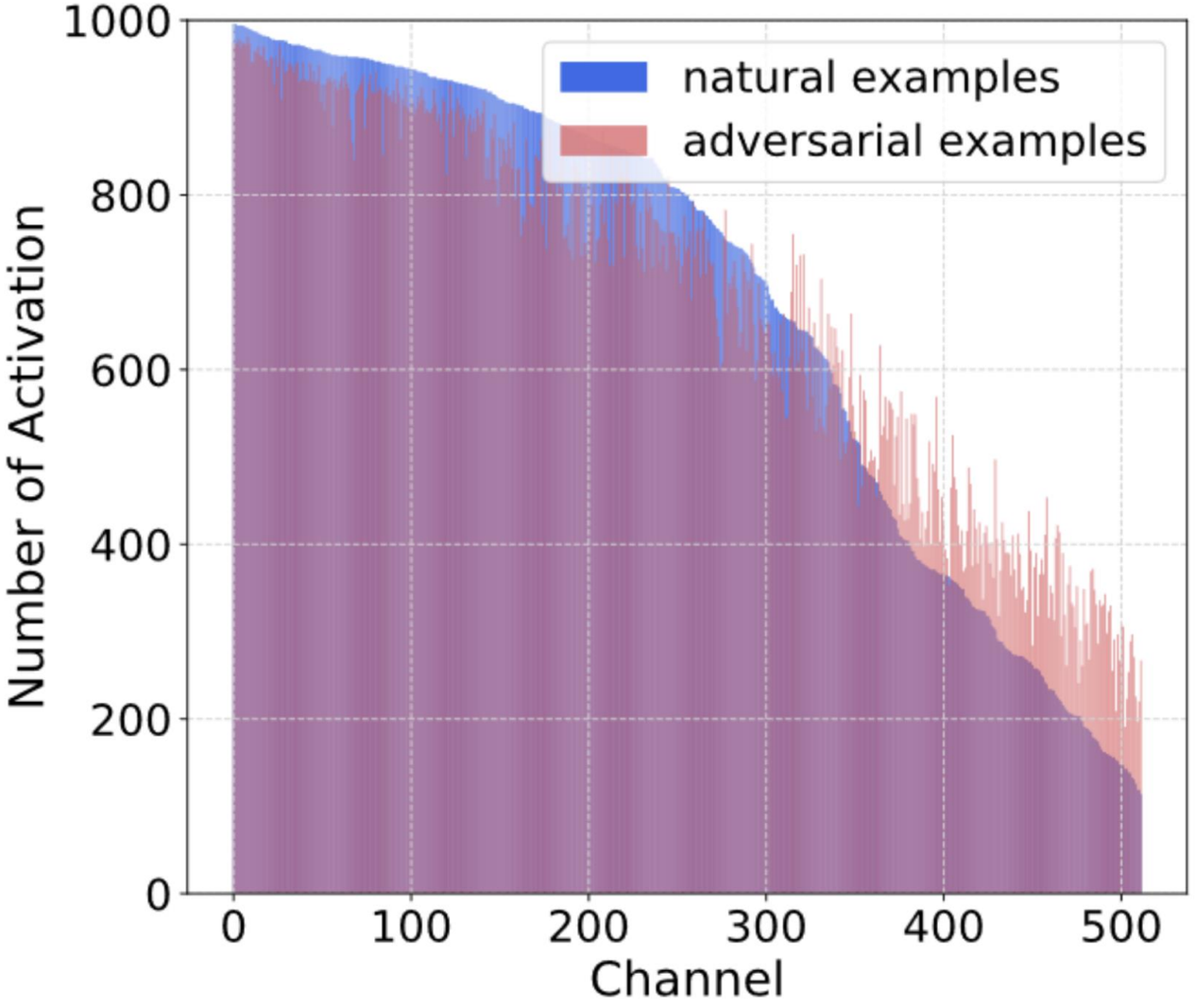
(b) ResNet-18 (ADV)

Channel-wise activation frequency

[IA21]



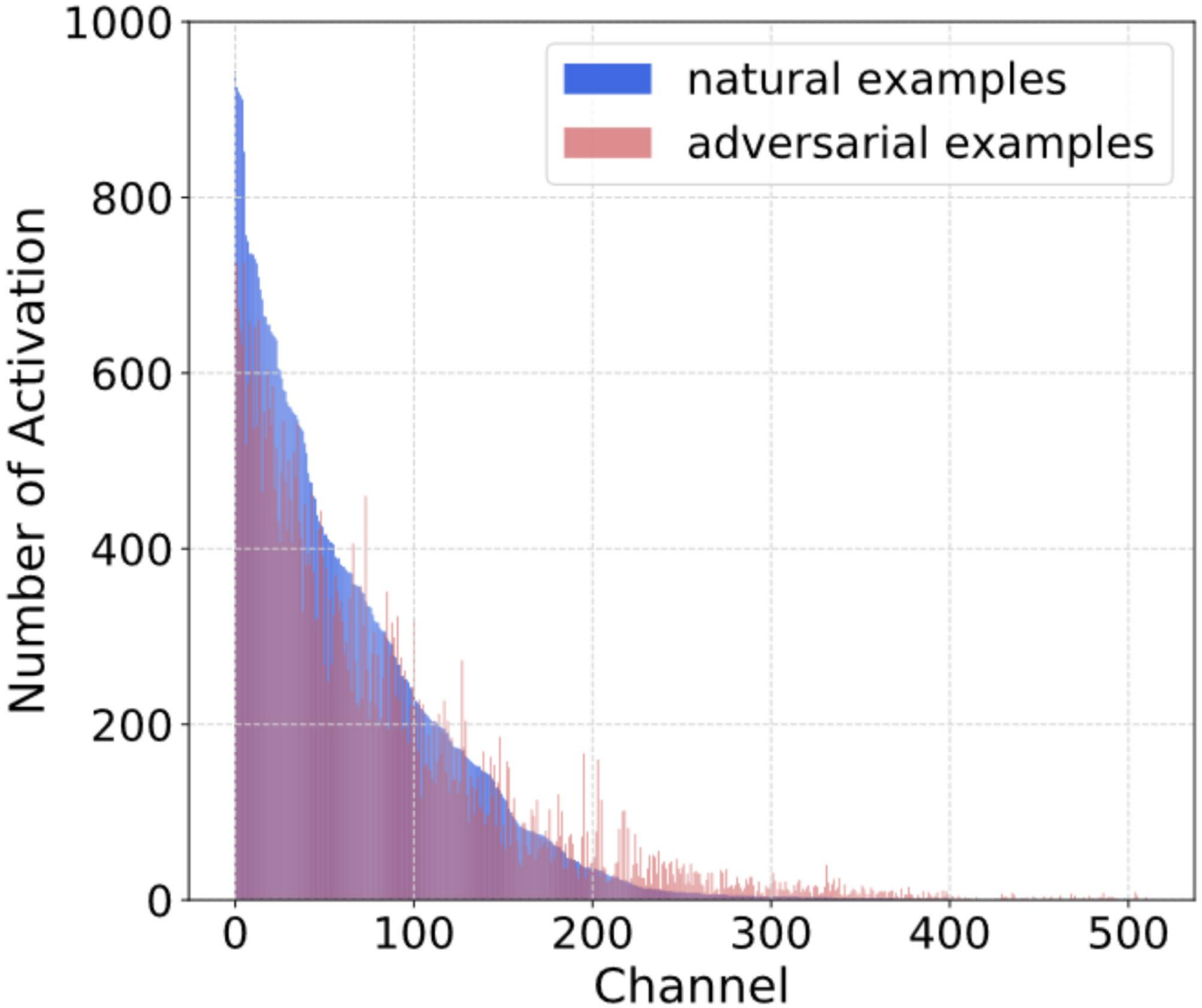
(a) ResNet-18 (STD)



(b) ResNet-18 (ADV)

Channel-wise activation suppressing (CAS)

[IA21]



FREE adversarial training improves efficiency

[SM19]

- **Re-use:** Update model parameters and adversarial perturbations during a backward pass.
- **Re-play:** train on the same mini-batch multiple during multiple consecutive iterations.
  - Increases the strength of the attack with each iterations.
  - Lowers that statistical efficiency of the training process.
- **Warm-start:** perturbation δ generated on the previous mini-batch re-used to warm-start a new mini-batch.

Adversarial Training: Standard vs FREE

Wide ResNet 32-10 trained on CIFAR10

[SM19, MM18]

Training	Evaluated Against					Train Time (min)
	Nat. Images	PGD-20	PGD-100	CW-100	10 restart PGD-20	
Natural	**95.01%**	0.00%	0.00%	0.00%	0.00%	**780**
Free $m = 2$	91.45%	33.92%	33.20%	34.57%	33.41%	816
Free $m = 4$	87.83%	41.15%	40.35%	41.96%	40.73%	800
Free $m = 8$	85.96%	**46.82%**	**46.19%**	**46.60%**	**46.33%**	785
Free $m = 10$	83.94%	46.31%	45.79%	45.86%	45.94%	785
7-PGD trained	87.25%	45.84%	45.29%	46.52%	45.53%	5418

~7X faster but 4% lower robust accuracy

FREE adversarial training improves efficiency

[SM19]

- **Re-use:** Update model parameters and adversarial perturbations during a backward pass.
- **Re-play:** train on the same mini-batch multiple during multiple consecutive iterations.
  - Increases the strength of the attack with each iterations.
  - Lowers that statistical efficiency of the training process.
- **Warm-start:** perturbation δ generated on the previous mini-batch re-used to warm-start a new mini-batch.

From FREE to FAST Adversarial Training

[SM19 (free), WR20 (fast)]

Warm-start: perturbation δ generated on the previous mini-batch is used to warm-start a new mini-batch.

"… there is little reason to believe that an adversarial perturbation for a previous minibatch is a reasonable starting point for the next minibatch. As a result, we hypothesize that the **main benefit comes** from simply **starting from a non-zero random initial perturbation**."

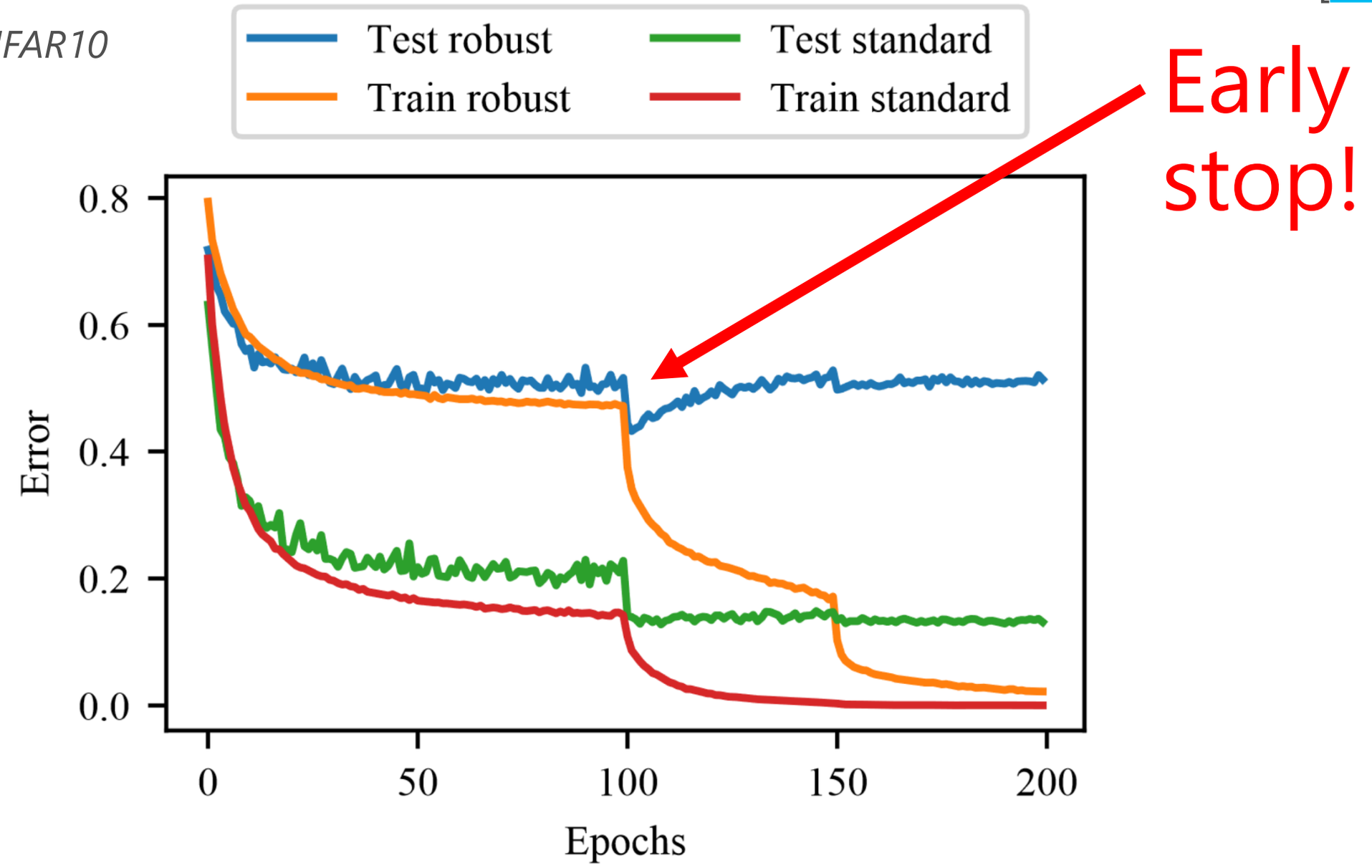
FREE vs FAST adversarial training

- FREE adversarial training [WR20]
  - Re-play min-batch multiple times (usually 8) and simultaneously compute gradients for inputs and parameters.
  - The first replay provides clean images for training (Mądry's approach uses only perturbed images).
  - It uses FGSM many times with smaller step size 2/255 but accumulates the perturbations δ.
- FAST adversarial training:
  - Generate gradients for inputs using **larger FGSM step size (10/255)** with **random initialization** and use the adversarial inputs as training samples to update θ.
  - Use mini-batch only once (no-replay).
  - Cannot simultaneously compute gradients for inputs and parameters.

Early stopping in Adversarial Training with FGSM

[RW20]

Model trained on CIFAR10



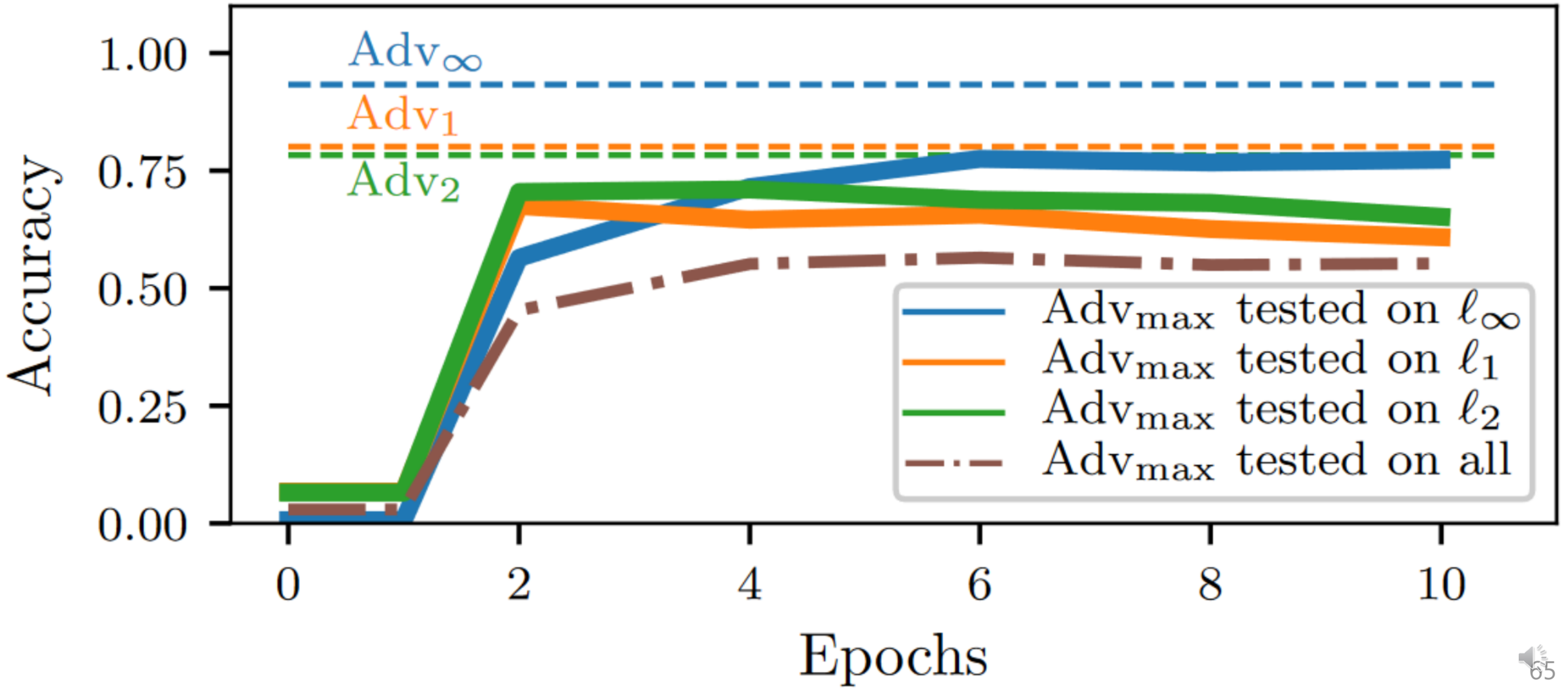
Standard & Robust Performance on CIFAR-10

Training time for $\varepsilon = \frac{8}{255}$ PGD adversary with step size $\frac{2}{255}$ and 10 random restarts on ResNet18 [WR20]

Method	Standard accuracy	PGD ($\epsilon = 8/255$)	Time (min)
FGSM + DAWNBench			
+ zero init	85.18%	0.00%	12.37
+ early stopping	71.14%	38.86%	7.89
+ previous init	86.02%	42.37%	12.21
+ random init	85.32%	44.01%	12.33
+ $\alpha = 10/255$ step size	83.81%	46.06%	12.17
+ $\alpha = 16/255$ step size	86.05%	0.00%	12.06
+ early stopping	70.93%	40.38%	8.81
"Free" ($m = 8$) (Shafahi et al., 2019)[1]	85.96%	46.33%	785
+ DAWNBench	78.38%	46.18%	20.91
PGD-7 (Madry et al., 2017)[2]	87.30%	45.80%	4965.71
+ DAWNBench	82.46%	50.69%	68.8

Robustness against multiple ℓ_p norm attacks

[TB19] c



Robustness against multiple ℓ_p norm attacks

[TB19,SB19]

- Training robust MNIST model might require expensive gradient-free attacks.
- Efficient scaling of current defenses to multiple perturbations remains an open problem.
- The max-strategy models achieve 100% train accuracy, thus multi-perturbation robustness increases the gap between accuracy on clean data vs robust accuracy.
- Unrecognized images are classified with high certainty.

Why is it hard to build a robust defense? [GP17]

- Robust adversarial training has a very high computational cost.
- No theoretical model of adversarial example crafting process.
- Adversarial examples require models to produce good outputs for every possible input.
- Current defenses are not adaptive:
  - Adversary can switch to a black-box attack when a defense masks gradients.
  - Most defenses close some vulnerabilities but leave others open.

Why do adversarial examples exist?

[GS15,CH20]

"Quirks" of the models.

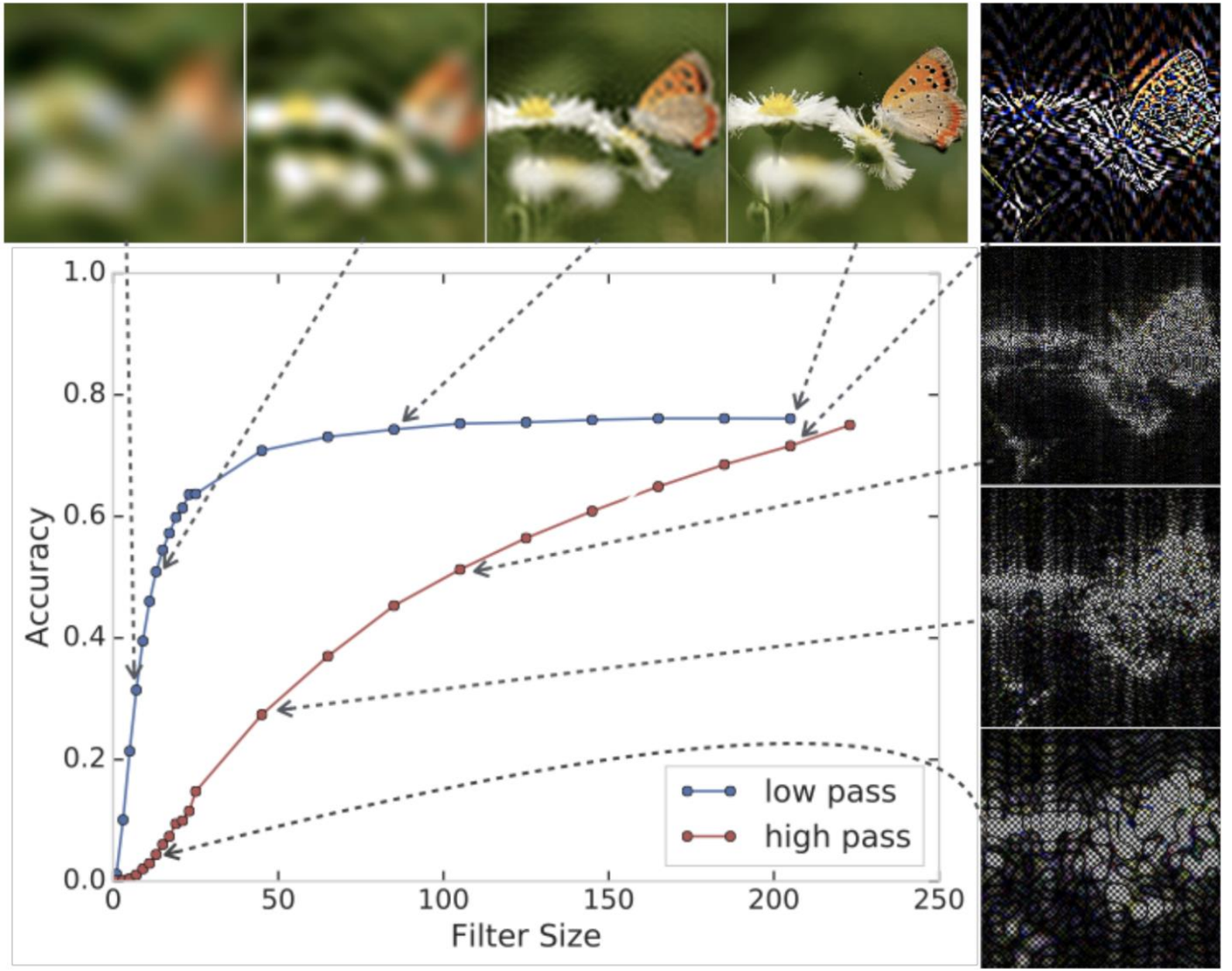
Consequence of high-dimensional input space.

Existence of highly-predictive but non-robust features that are brittle and incomprehensible to humans.

Open problem.

High accuracy on high frequency features

[YG19]

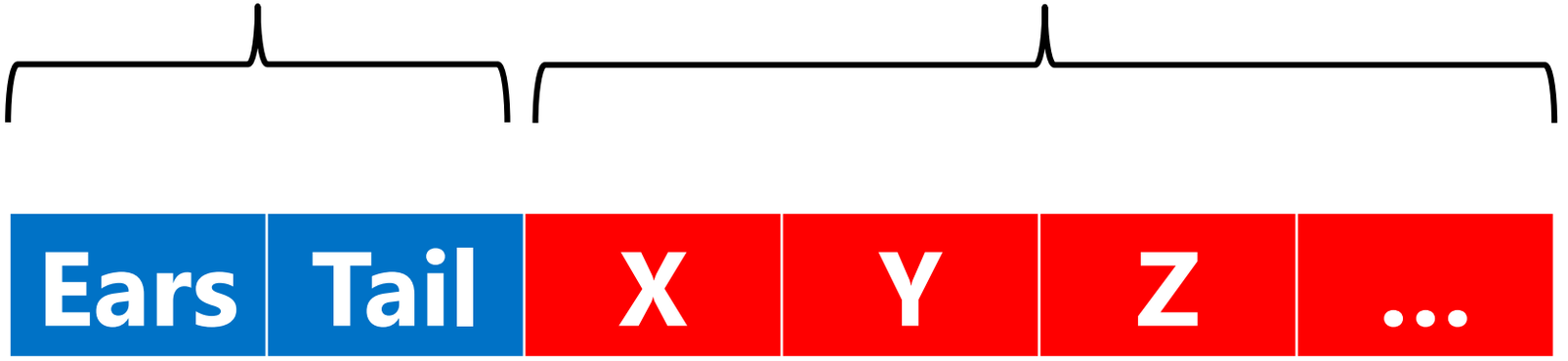


Robust & non-robust features

[IS20]

Robust features correlated with labels even when an adversary is present

Non-robust features correlated with labels on average but manipulated by an adversary

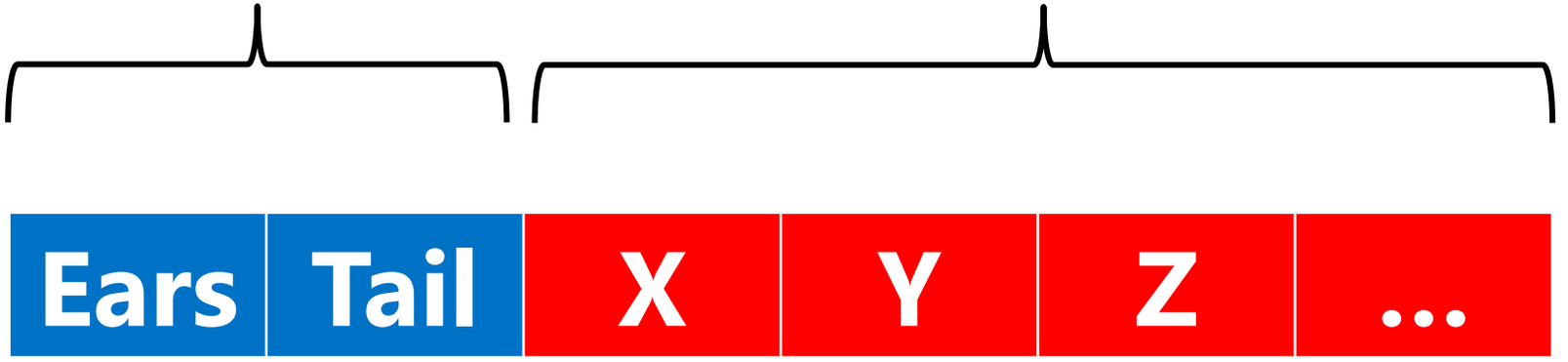


Robust & non-robust features

[IS20]

Robust features correlated with labels even when an adversary is present

Non-robust features correlated with labels on average but manipulated by an adversary



When maximizing (test) accuracy:

1. All features are useful.
2. Non-robust features are often very helpful.

Result: Models use the non-robust features and become vulnerable to adversarial examples.

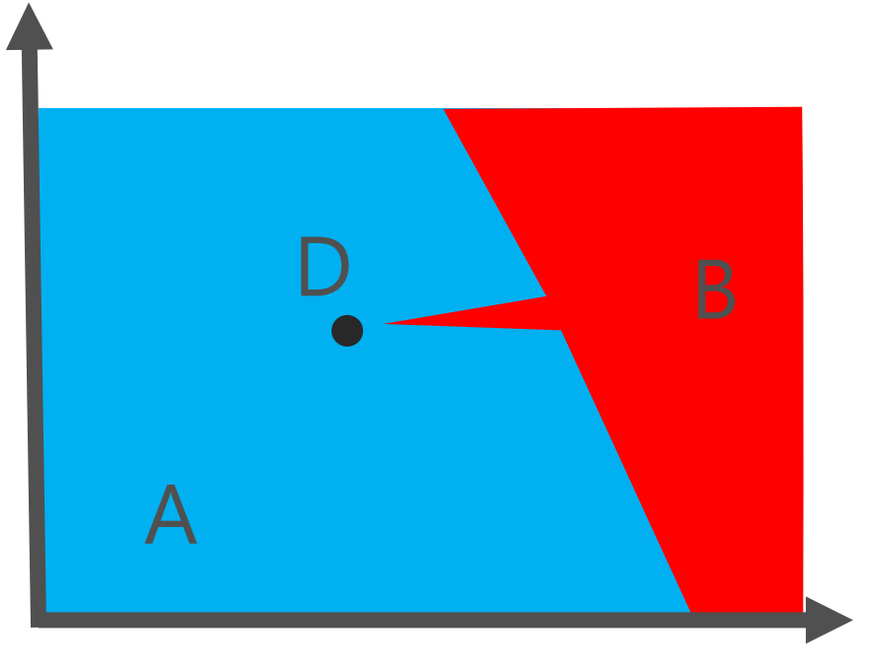
What causes adversarial examples?

[RK19]

A - correct class

B - incorrect class

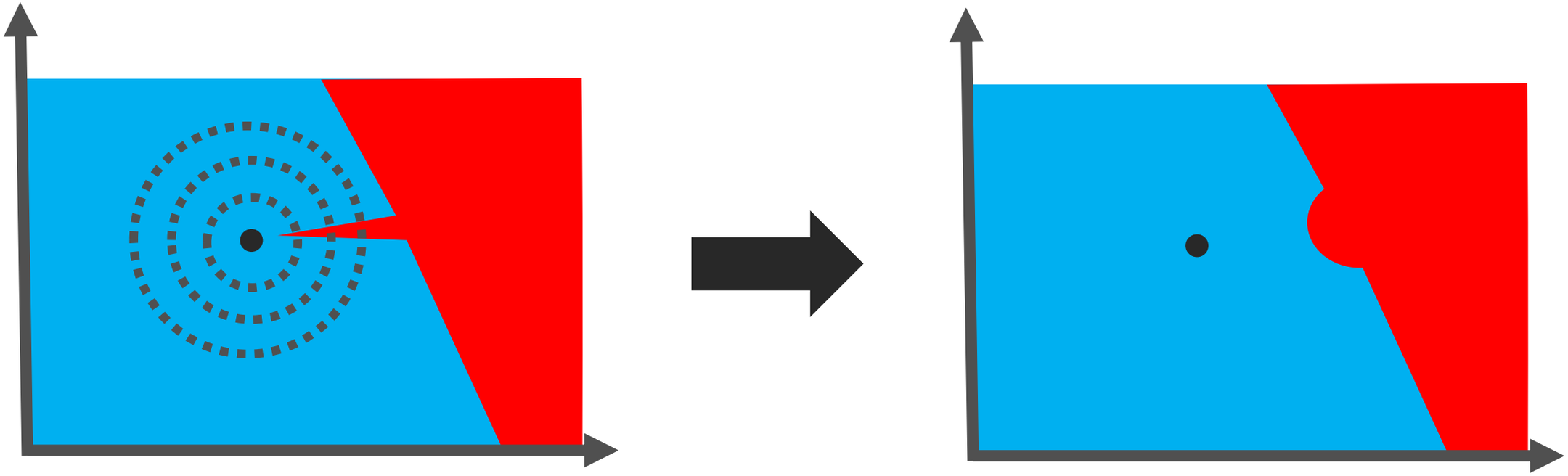
D - data point



Caused by a small adversarial cone jutting into a correct decision region

Certified robustness via randomized smoothing

[CR19,LA19,LC19]



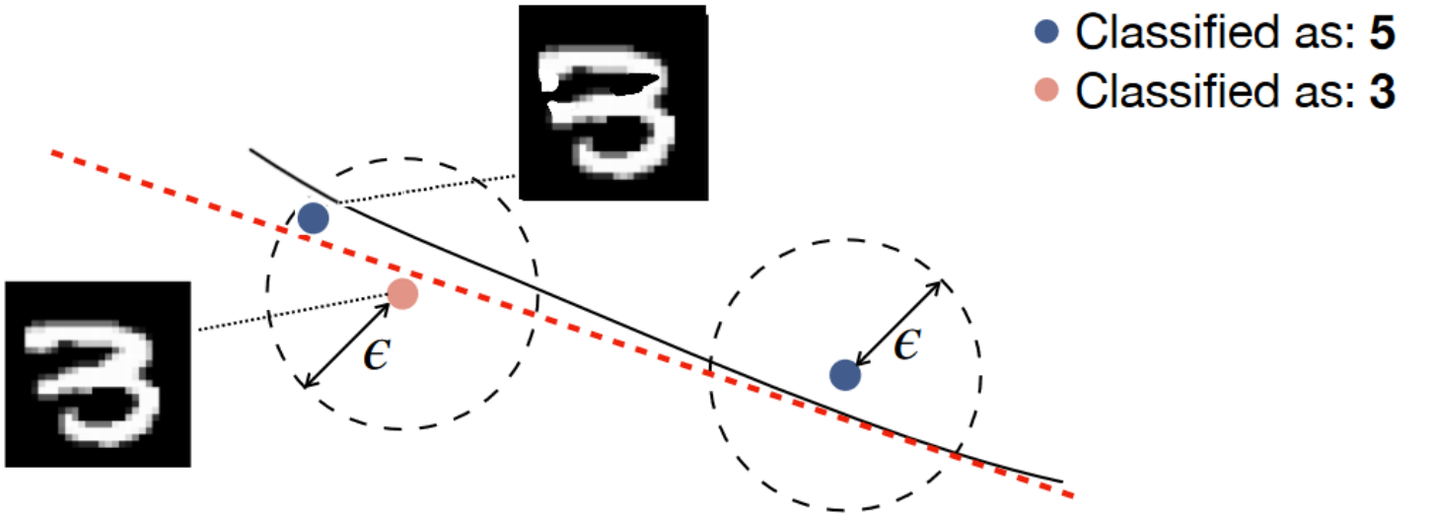
$f(x)$

$$g(x) = \underset{y}{\operatorname{argmax}}\, P_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[f(x+\epsilon) = y]$$

Smooth decision regions

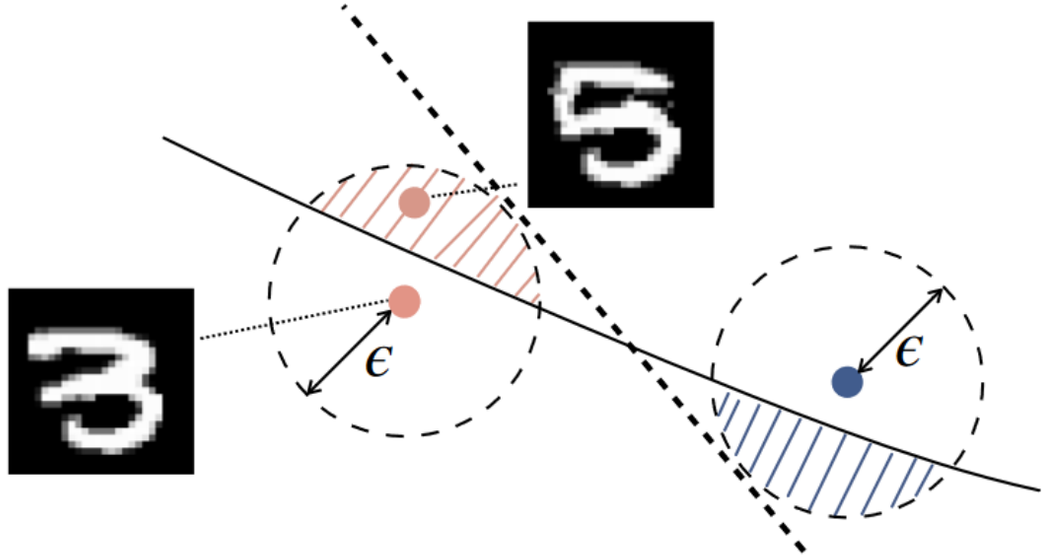
Adversarial examples

Excessively Sensitive Model



Excessively Invariant Model

[TB20]



--- "unrobust" decision boundary **—— Oracle Decision-boundary** **--- "robust" decision boundary**

ϵ-bounded sensitivity-based adversarial example

$f(x^*) \neq f(x)$

$\|x^* - x\| \leq \epsilon$

assumes: $O(x^*) = O(x)$

ϵ-bounded invariance-based adversarial example

$f(x^*) = f(x)$

$\|x^* - x\| \leq \epsilon$

$O(x^*) \neq O(x), O(x^*) \neq \perp$

Open problems

- How to define adversarial examples?
- What causes the adversarial examples?
- How can we create more robust models? How much capacity or additional data do they need? What should be their architecture and activation functions? How to defend against multiple perturbations that use different norms?
- Do we have to sacrifice the performance for legitimate users to train robust models?
- Many of the most important problems remain open.

Thank you

Bibliography

[BS14] Joan Bruna, Christian Szegedy, Ilya Sutskever, Ian Goodfellow, Wojciech Zaremba, Rob Fergus, Dumitru Erhan, "Intriguing properties of neural networks", ICLR 2014.

[GS15] Ian Goodfellow, Jonathon Shlens, Christian Szegedy, "Explaining and Harnessing Adversarial Examples", ICLR 2015.

[HX15] Ruitong Huang, Bing Xu, Dale Schuurmans, Csaba Szepesvari, "Learning with a Strong Adversary", arXiv 2015.

[PM16] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples", arXiv 2016.

[GP17] Ian Goodfellow and Nicolas Papernot, "Is attacking machine learning easier than defending it?", cleverhans blog post 2017.

[KG17] Alexey Kurakin, Ian Goodfellow, Samy Bengio, "Adversarial Machine Learning at Scale", ICLR 2017.

[KGB17] Alexey Kurakin, Ian Goodfellow, Samy Bengio, "Adversarial examples in the physical world", arXiv 2017.

[PM17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, Ananthram Swami, "Practical Black-Box Attacks against Machine Learning", ASIA CCS 2017.

[AC18] Anish Athalye, Nicholas Carlini, David Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples", ICML 2018.

[AE18] Anish Athalye, Logan Engstrom, Andrew Ilyas, Kevin Kwok, "Synthesizing Robust Adversarial Examples", ICML 2018.

[CW18] Nicholas Carlini, David Wagner, "Audio Adversarial Examples: Targeted Attacks on Speech-to-Text", arXiv 2018.

[MM18] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu, "Towards deep learning models resistant to adversarial attacks", ICLR 2018.

[SS18] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, Aleksander Madry, "Adversarially Robust Generalization Requires More Data", NeurIPS 2018.

[TK18] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick McDaniel, "Ensemble Adversarial Training: Attacks and Defenses", ICLR 2018.

[CR19] Jeremy Cohen, Elan Rosenfeld, Zico Kolter, "Certified Adversarial Robustness via Randomized Smoothing", ICML 2019.

[CR19] Jeremy Cohen, Elan Rosenfeld, Zico Kolter, "Certified Adversarial Robustness via Randomized Smoothing", ICML 2019.

[GG19] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, Kilian Weinberger, "Simple Black-box Adversarial Attacks", ICML 2019.

[HL19] Dan Hendrycks, Kimin Lee, Mantas Mazeika, "Using Pre-Training Can Improve Model Robustness and Uncertainty", ICML 2019.

[LA19] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Suman Jana, "Certified Robustness to Adversarial Examples with Differential Privacy", IEEE 2019.

[LC19] Bai Li, Changyou Chen, Wenlin Wang, Lawrence Carin, "Certified Adversarial Robustness with Additive Noise", NeurIPS 2019.

[RK19] Kevin Roth, Yannic Kilcher, Thomas Hofmann "The Odds are Odd: A Statistical Test for Detecting Adversarial Examples", ICML 2019.

[SB19] Lukas Schott, Jonas Rauber, Matthias Bethge, Wieland Brendel, "Towards the first adversarially robust neural network model on MNIST", ICLR 2019.

[SM19] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, Tom Goldstein, "Adversarial training for free!", NeurIPS 2019.

[TB19] Florian Tramer, Dan Boneh, "Adversarial Training and Robustness for Multiple Perturbations", NeurIPS 2019.

[UB19] Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli, "Are Labels Required for Improving Adversarial Robustness?", arXiv 2019.

[YG19] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, Justin Gilmer, "A Fourier Perspective on Model Robustness in Computer Vision", NeurIPS 2019.

[CH20] Francesco Croce, Matthias Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks", ICML 2020.

[CR20] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, Percy S. Liang, "Unlabeled Data Improves Adversarial Robustness", NeurIPS 2020.

[GQ20] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, Pushmeet Kohli, "Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples", arXiv 2020.

[IS20] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, Aleksander Madry, "Adversarial Examples Are Not Bugs, They Are Features", NerurIPS 2020.

[RW20] Leslie Rice, Eric Wong, Zico Kolter, "Overfitting in adversarially robust deep learning", ICML 2020.

[SE20] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, Dorothea Kolossa, "Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems", ACSAC 2020.

[TB20] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, Jörn-Henrik Jacobsen, "Fundamental Tradeoffs between Invariance andSensitivity to Adversarial Perturbations", ICML 2020.

[WR20] Eric Wong, Leslie Rice, J. Zico Kolter, "Fast is better than free: Revisiting adversarial training", ICLR 2020.

[AI21] "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", under submission to ICLR 2021.

[GA21] "Geometry-aware Instance-reweighted Adversarial Training", under submission to ICLR 2021.

[IA21] "Improving Adversarial Robustness via Channel-wise Activation Suppressing", under submission to ICLR 2021.