Detecting Brittle Decisions for Free: Leveraging Margin Consistency in Deep Robust Classifiers.

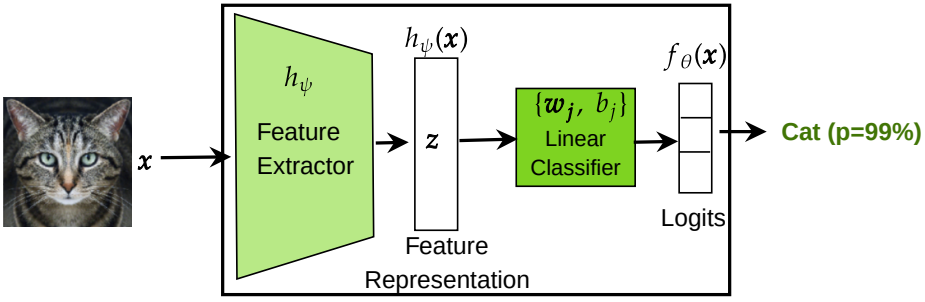
Jonas Ngnawé

{jonas.ngnawe.1}@ulaval.ca

In collaboration with:
Sabyasachi Sahoo, Yann Pequignot, Frederic Precioso, Christian Gagné

June 2024

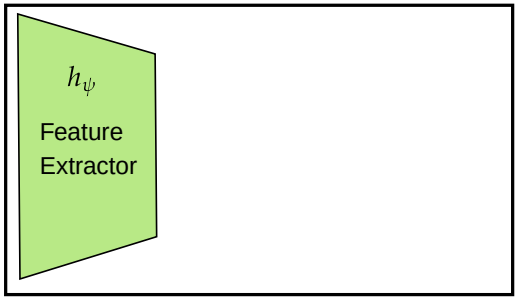
Deep Neural Classifiers



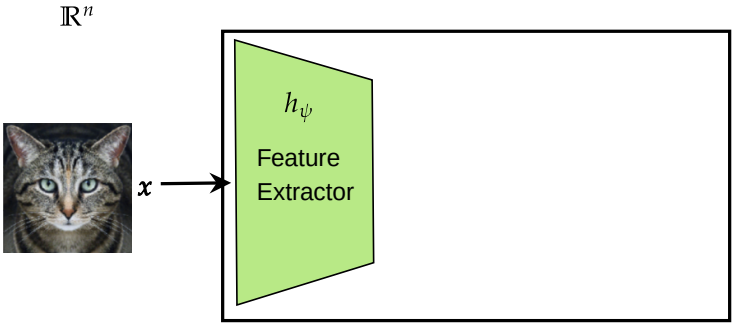
Deep Neural Classifiers

$\mathbb{R}^n$

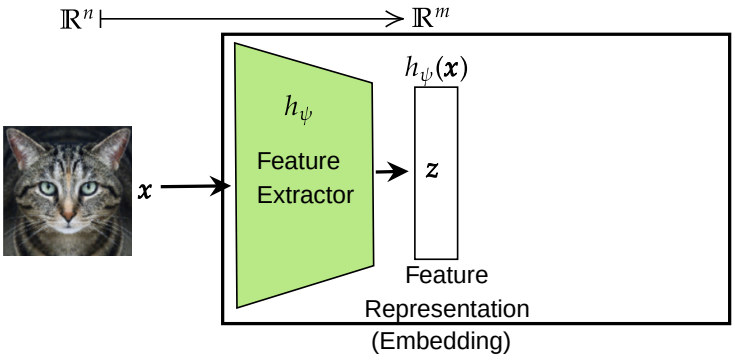
$\boldsymbol{x}$



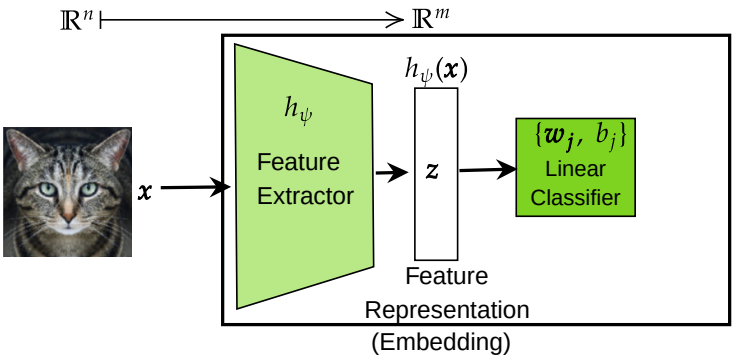
Deep Neural Classifiers



Deep Neural Classifiers



Deep Neural Classifiers



Deep Neural Classifiers

$\boldsymbol{x}$

h_ψ

Feature Extractor

$h_\psi(\boldsymbol{x})$

$\boldsymbol{z}$

Feature Representation (Embedding)

$\{\boldsymbol{w_j}, b_j\}$

Linear Classifier

$f_\theta(\boldsymbol{x})$

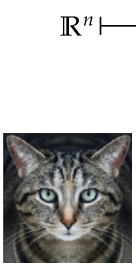
Logits

$\mathbb{R}^n \mapsto \mathbb{R}^m \mapsto \mathbb{R}^C$

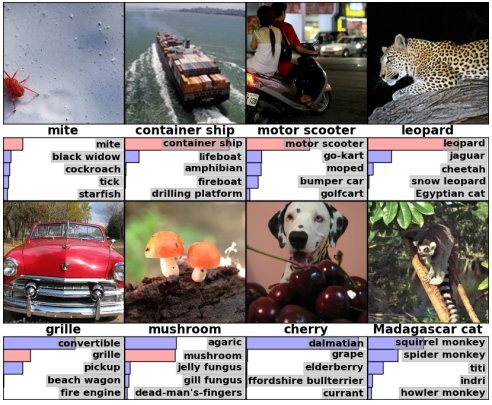
C: number of classes

$f^j_\theta(\boldsymbol{x}) = \boldsymbol{w_j^T x} + b_j$

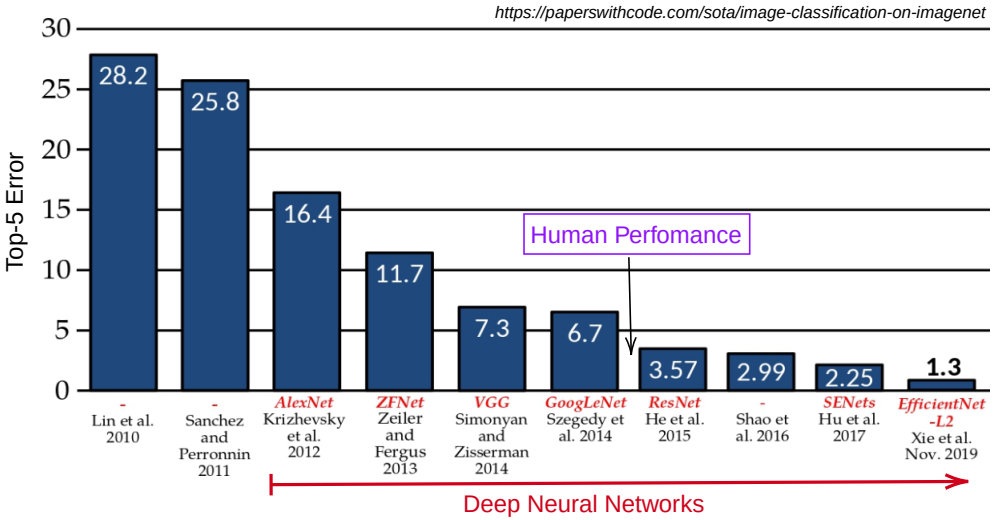
Deep Neural Classifiers



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



Great Performance (ILSVRC)



http://www2.ift.ulaval.ca/~pgiguere/cours/DeepLearning/2020/01-Introduction-2020.pdf

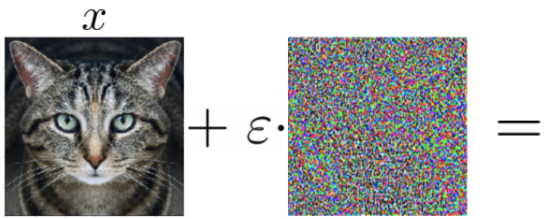
Vulnerability to Adversarial Examples

x

Cat
99%

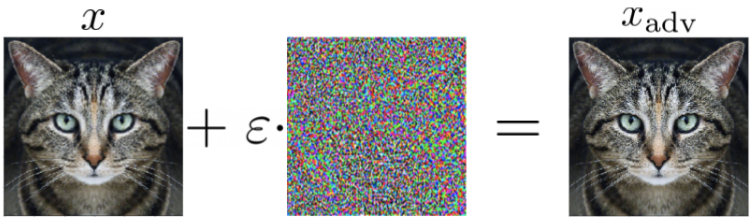
Dan Hendrycks, Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/

Vulnerability to Adversarial Examples



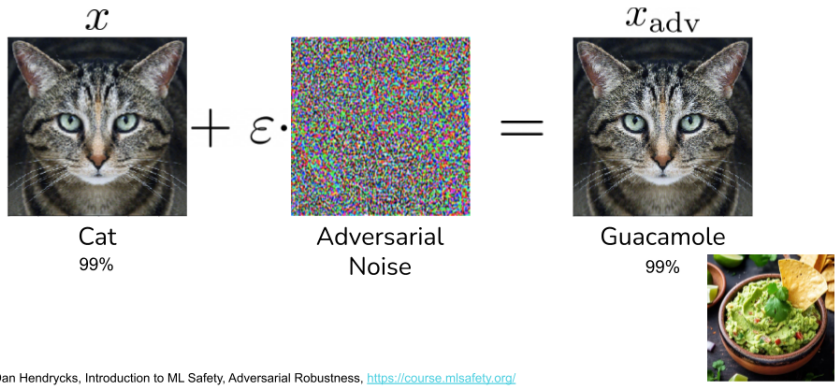
Dan Hendrycks, Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/

Vulnerability to Adversarial Examples



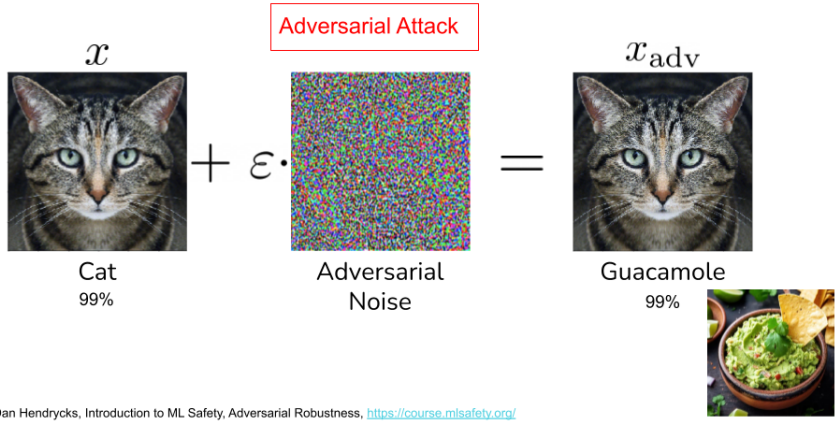
Dan Hendrycks, Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/

Vulnerability to Adversarial Examples



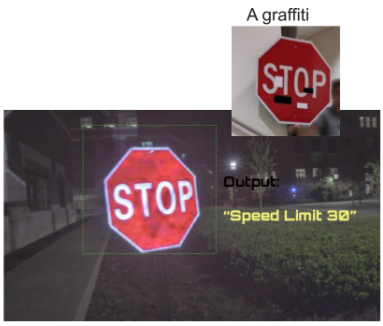
Dan Hendrycks, Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/

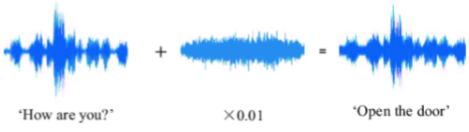
Vulnerability to Adversarial Examples



Dan Hendrycks, Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/

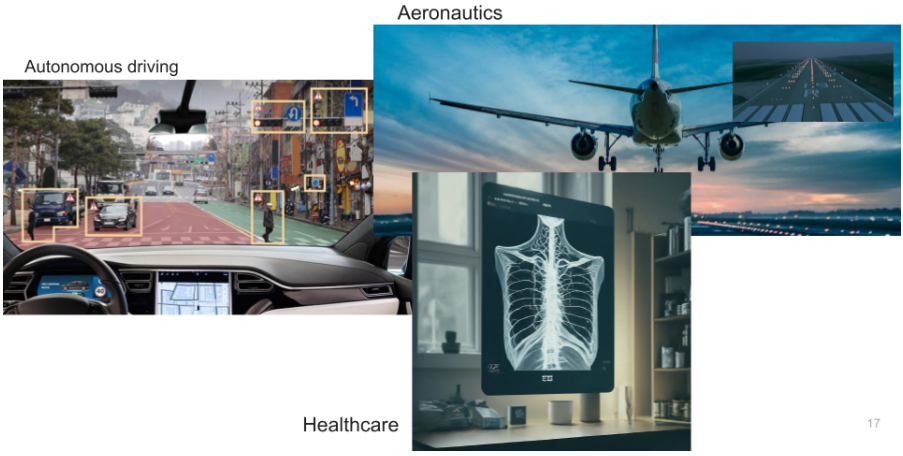
Vulnerability to Adversarial Examples





Dan Hendrycks Introduction to ML Safety, Adversarial Robustness, https://course.mlsafety.org/
Athalye, Anish, et al. "Synthesizing robust adversarial examples." *International conference on machine learning*. PMLR, 2018.

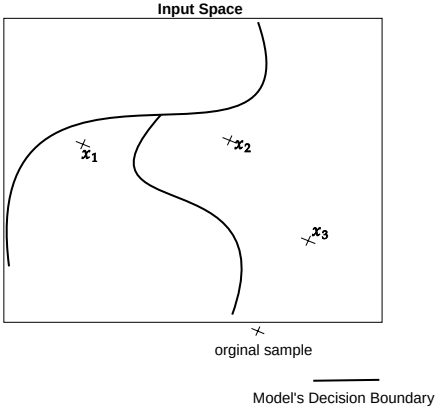
Robustness is required in sensitive and safety critical systems



Local Robustness

Definition (ℓ_p-robustness/ϵ-robustness)

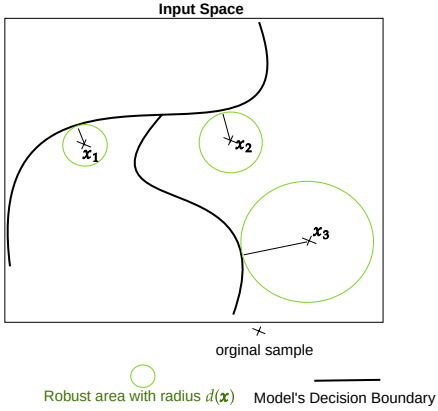
A model f_θ is **ϵ-robust** at point $\mathbf{x}$ if for any $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon \implies \hat{y}(\mathbf{x}') = \hat{y}(\mathbf{x})$



Local Robustness

Definition (ℓ_p-robustness/ϵ-robustness)

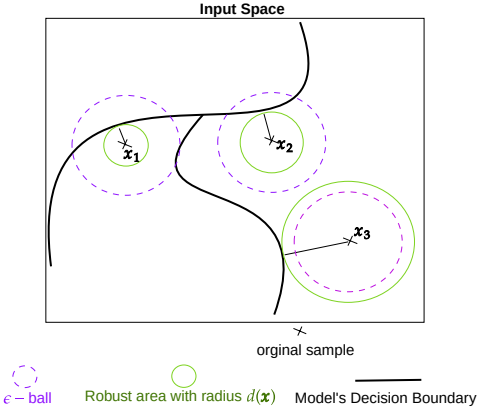
A model f_θ is **ϵ-robust** at point $\mathbf{x}$ if for any $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon \implies \hat{y}(\mathbf{x}') = \hat{y}(\mathbf{x})$



Local Robustness

Definition (ℓ_p-robustness/ϵ-robustness)

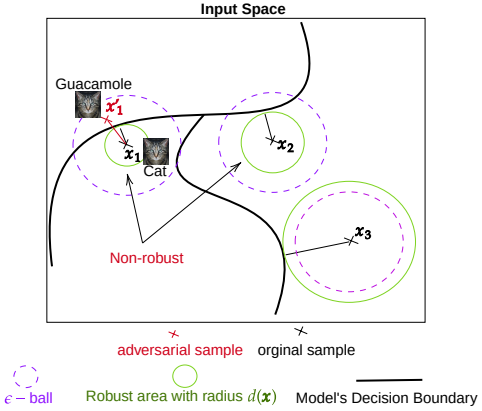
A model f_θ is **ϵ-robust** at point $\mathbf{x}$ if for any $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon \implies \hat{y}(\mathbf{x}') = \hat{y}(\mathbf{x})$



Local Robustness

Definition (ℓ_p-robustness/ϵ-robustness)

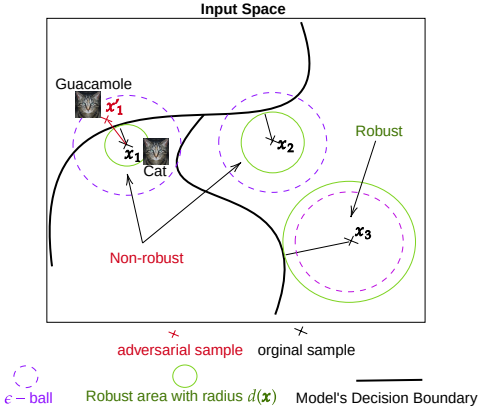
A model f_θ is **ϵ-robust** at point $\mathbf{x}$ if for any $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon \implies \hat{y}(\mathbf{x}') = \hat{y}(\mathbf{x})$



Local Robustness

Definition (ℓ_p-robustness/ϵ-robustness)

A model f_θ is **ϵ-robust** at point $\mathbf{x}$ if for any $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon \implies \hat{y}(\mathbf{x}') = \hat{y}(\mathbf{x})$



Robustness Evaluation

- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.

Robustness Evaluation

- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.
- Perform **adversarial attacks** (e.g., *AutoAttack*) on a test set.
- Determine if each sample $\mathbf{x}$ is **vulnerable** to attacks ($\implies$ *non-robust*).

Robustness Evaluation

- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.
- Perform **adversarial attacks** (e.g., *AutoAttack*) on a test set.
- Determine if each sample $\mathbf{x}$ is **vulnerable** to attacks ($\implies$ *non-robust*).

What if you want to know in a real-time deployment context?

Robustness Evaluation

- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.
- Perform **adversarial attacks** (e.g., *AutoAttack*) on a test set.
- Determine if each sample $\mathbf{x}$ is **vulnerable** to attacks ($\implies$ *non-robust*).

What if you want to know in a real-time deployment context?

- Reducing Risk
- Monitoring
- Resource Prioritization

Robustness Evaluation

- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.
- Perform **adversarial attacks** (e.g., *AutoAttack*) on a test set.
- Determine if each sample $\mathbf{x}$ is **vulnerable** to attacks ($\implies$ *non-robust*).

  What if you want to know in a real-time deployment context?

  - Reducing Risk
  - Monitoring
  - Resource Prioritization

- Attack again or use Formal Robustness Verification

Robustness Evaluation

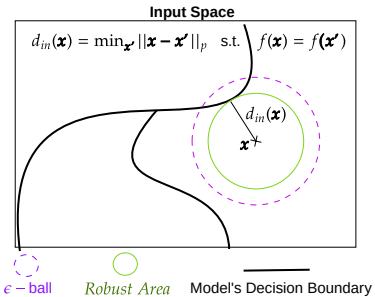
- Compute the $\underbrace{\textbf{robust accuracy}}_{\text{P(correct and robust)}}$ of the model.
- Perform **adversarial attacks** (e.g., *AutoAttack*) on a test set.
- Determine if each sample $\mathbf{x}$ is **vulnerable** to attacks ($\implies$ *non-robust*).

What if you want to know in a real-time deployment context?

- Reducing Risk
- Monitoring
- Resource Prioritization

- Attack again or use Formal Robustness Verification
- Both are (still) computationally expensive in that context

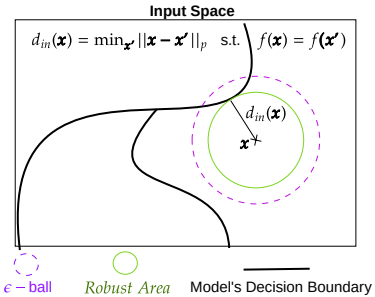
Input Space Margin and Non-Robustness Detection



The input margin $d_{in}(\mathbf{x})$ is the perfect discriminative score

$$g(\mathbf{x}) = \mathbb{1}_{[d_{in}(\mathbf{x}) \leq \epsilon]}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is non-robust} \\ 0 & \text{if } \mathbf{x} \text{ is robust} \end{cases}.$$

Input Space Margin and Non-Robustness Detection

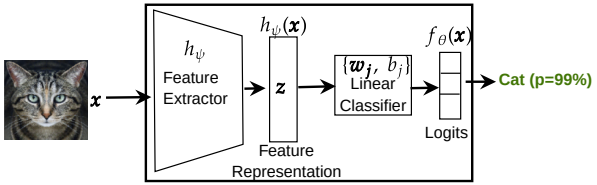


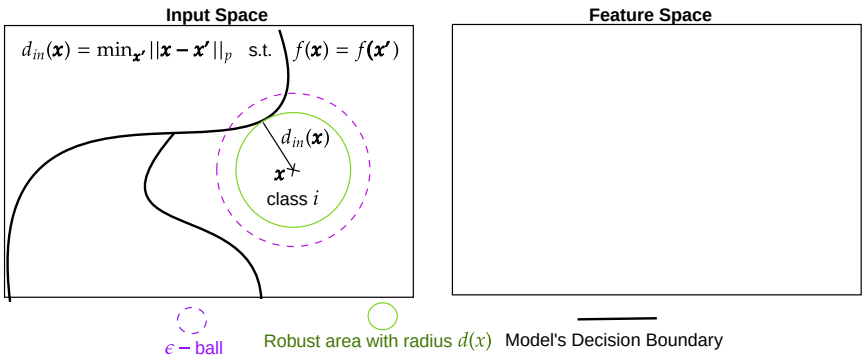
The input margin $d_{in}(\mathbf{x})$ is the perfect discriminative score

$$g(\mathbf{x}) = \mathbb{1}_{[d_{in}(\mathbf{x}) \leq \epsilon]}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is non-robust} \\ 0 & \text{if } \mathbf{x} \text{ is robust} \end{cases}.$$

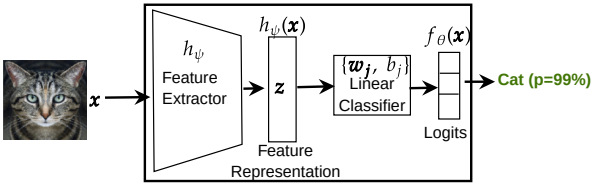
$d_{in}(\mathbf{x})$ is **intractable** for general deep nets.

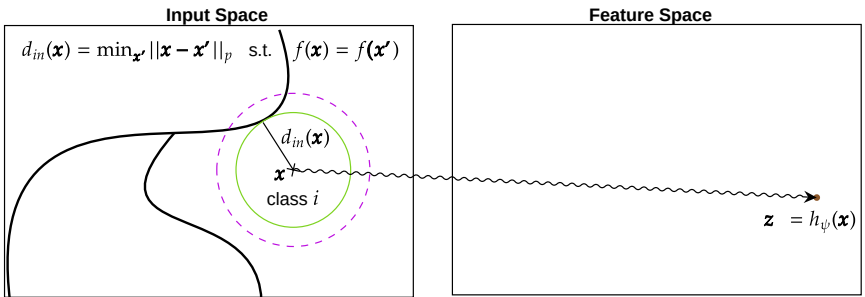
Margin in the feature space



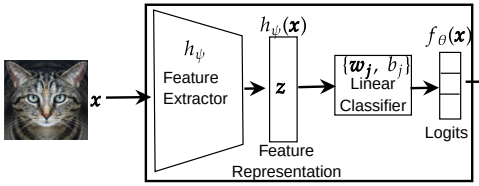


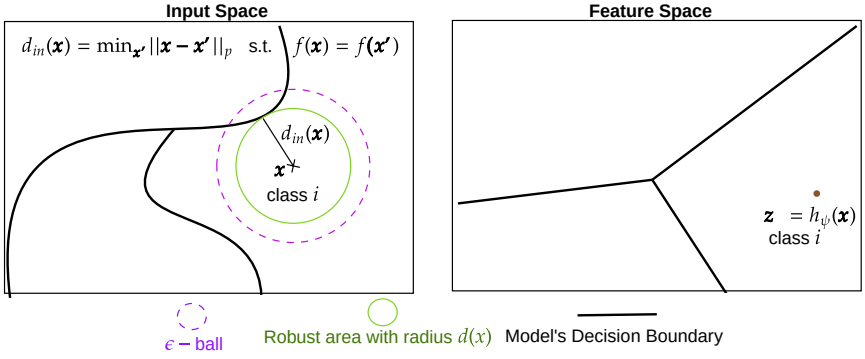
Margin in the feature space



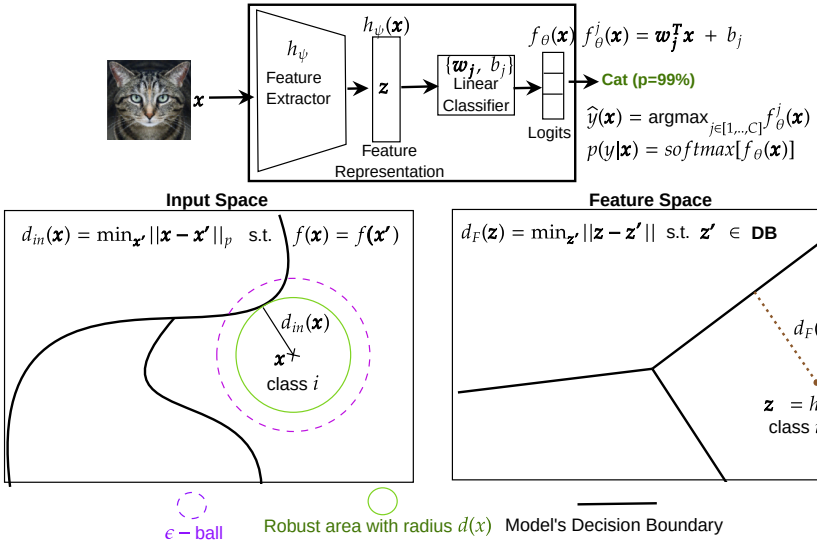


Margin in the feature space

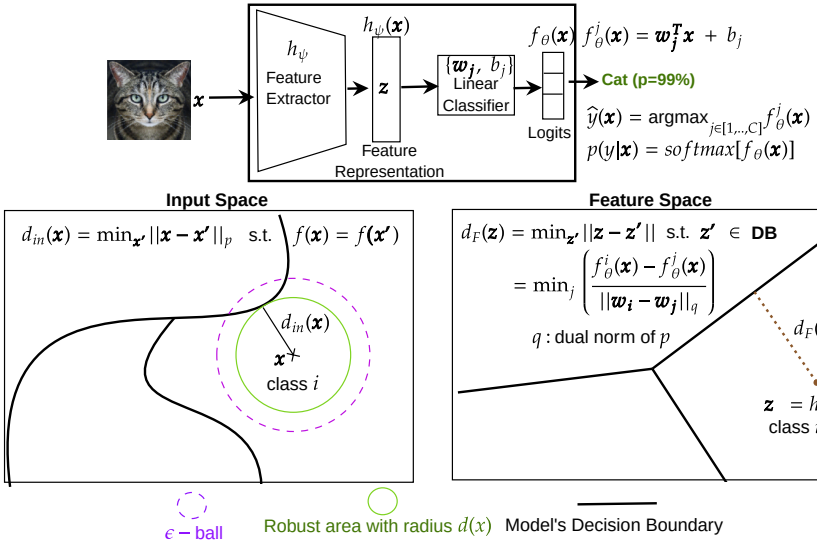




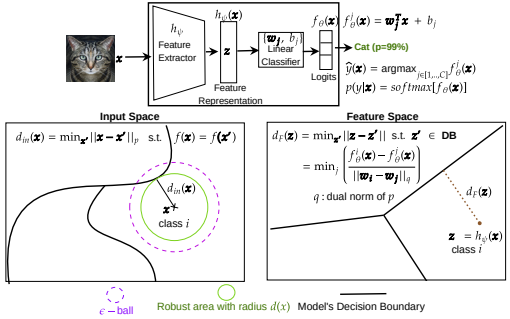
Margin in the feature space



Margin in the feature space

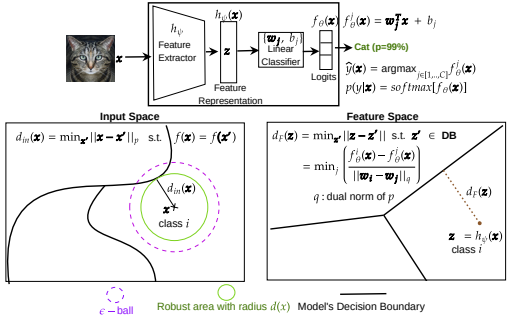


Margin in the feature space



- If the classifiers are equidistant ($\|\mathbf{w}_i - \mathbf{w}_j\|_q = \omega > 0, \forall i, j$)

Margin in the feature space



- If the classifiers are equidistant ($\|\mathbf{w}_i - \mathbf{w}_j\|_q = \omega > 0, \forall i, j$), the margin becomes:

$$\frac{1}{\omega}\Bigg(\underbrace{f_\theta^i(\mathbf{x}) - \max_{j, j\neq i} f_\theta^j(\mathbf{x})}_{d_{out}(x):=\text{logit margin}}\Bigg).$$

Contributions

We introduce **Margin Consistency**, a property of robust deep classifiers: rank correlation between input margin and logit margin.

Contributions

We introduce **Margin Consistency**, a property of robust deep classifiers: rank correlation between input margin and logit margin.

- We show that it is a necessary and sufficient condition to use the logit margin as a proxy score for non-robustness detection.

Contributions

We introduce **Margin Consistency**, a property of robust deep classifiers: rank correlation between input margin and logit margin.

- We show that it is a necessary and sufficient condition to use the logit margin as a proxy score for non-robustness detection.
- Experimentally show (on CIFAR10 and CIFAR100) that most robust deep classifiers are strongly margin consistent.

Contributions

We introduce **Margin Consistency**, a property of robust deep classifiers: rank correlation between input margin and logit margin.

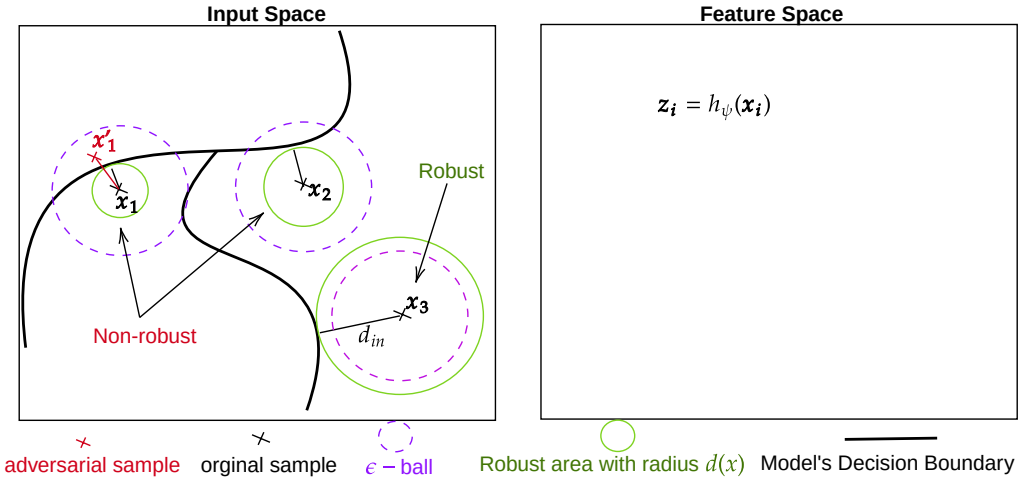
- We show that it is a necessary and sufficient condition to use the logit margin as a proxy score for non-robustness detection.
- Experimentally show (on CIFAR10 and CIFAR100) that most robust deep classifiers are strongly margin consistent.
- Experimentally show that the logit margin of strongly margin consistent models can confidently detect vulnerable samples and estimate robust accuracy at scale using only a small subset of test data.

Contributions

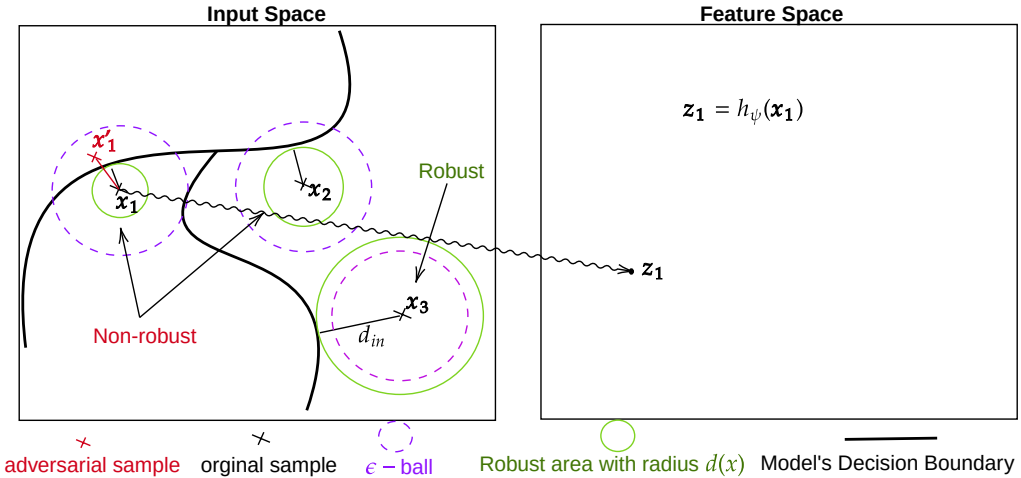
We introduce **Margin Consistency**, a property of robust deep classifiers: rank correlation between input margin and logit margin.

- We show that it is a necessary and sufficient condition to use the logit margin as a proxy score for non-robustness detection.
- Experimentally show (on CIFAR10 and CIFAR100) that most robust deep classifiers are strongly margin consistent.
- Experimentally show that the logit margin of strongly margin consistent models can confidently detect vulnerable samples and estimate robust accuracy at scale using only a small subset of test data.
- In weak margin consistency cases, we can learn a pseudo-margin that better correlates with the input margin.

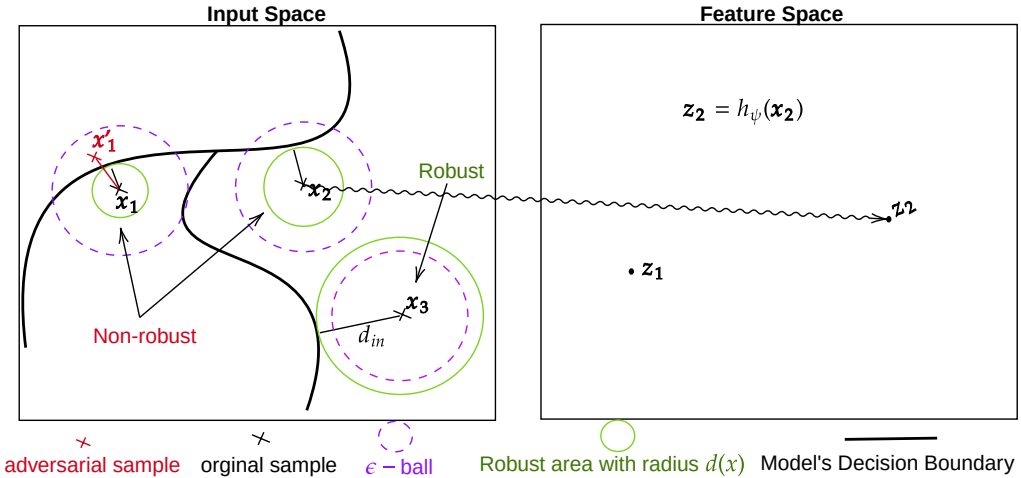
Margin Consistency



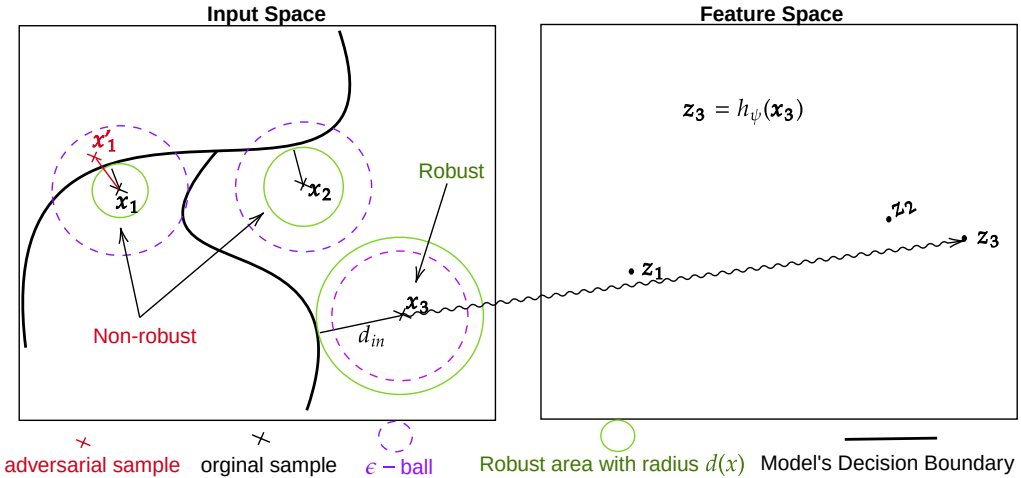
Margin Consistency



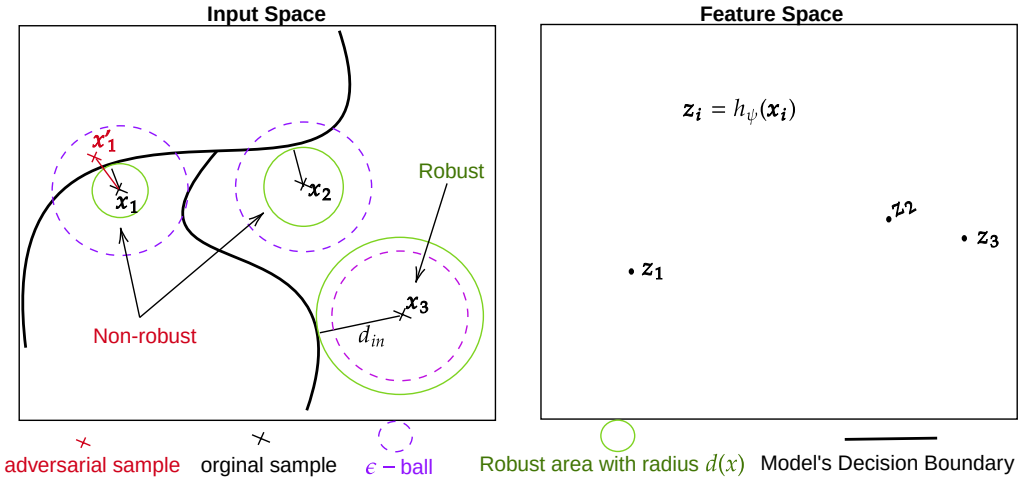
Margin Consistency



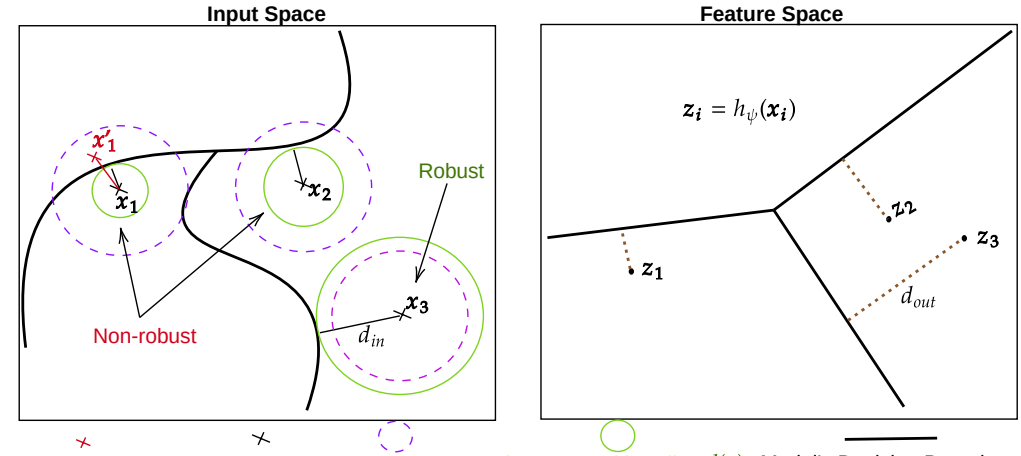
Margin Consistency



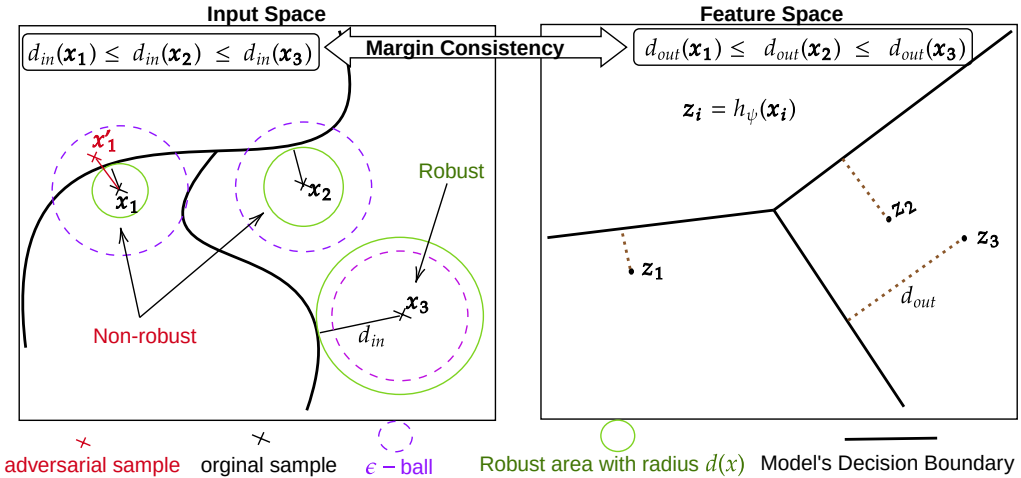
Margin Consistency



Margin Consistency

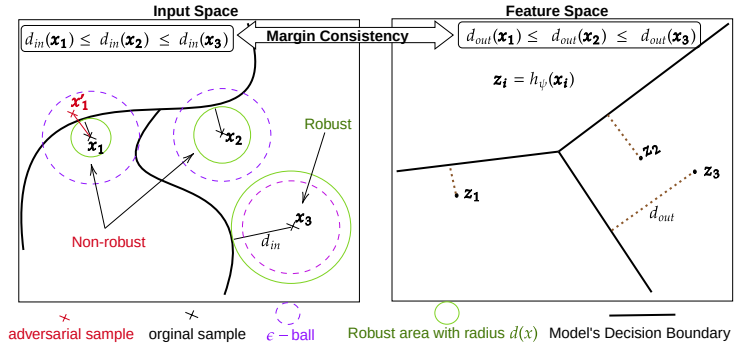


Margin Consistency



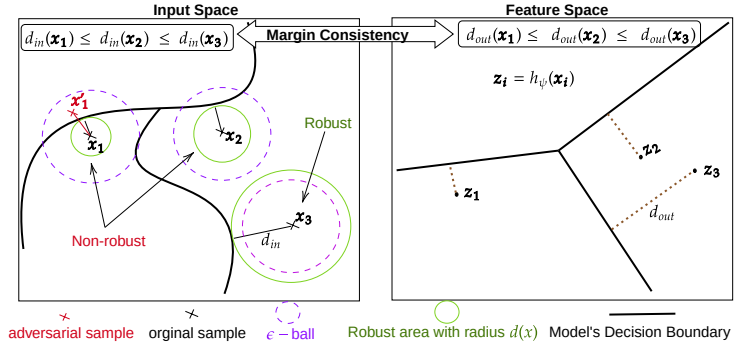
Definition (Margin Consistency - MC)

A model is **margin-consistent** if there is a monotonic relationship between the input space margin and the logit margin, i.e., $d_{in}(\mathbf{x}_1) \leq d_{in}(\mathbf{x}_2) \Leftrightarrow d_{out}(\mathbf{x}_1) \leq d_{out}(\mathbf{x}_2)$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$.



Definition (Margin Consistency - MC)

A model is **margin-consistent** if there is a monotonic relationship between the input space margin and the logit margin, i.e., $d_{in}(\mathbf{x}_1) \leq d_{in}(\mathbf{x}_2) \Leftrightarrow d_{out}(\mathbf{x}_1) \leq d_{out}(\mathbf{x}_2)$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$.



Kendall Rank Correlation τ measures margin consistency ($|\tau| \in [0, 1]$).

Margin Consistency

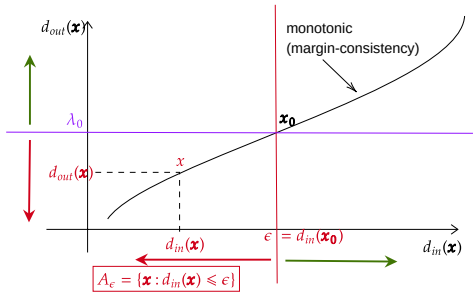
Theorem

Margin consistency is a necessary and sufficient condition for the logit margin to perfectly separate non-robust samples from robust samples.

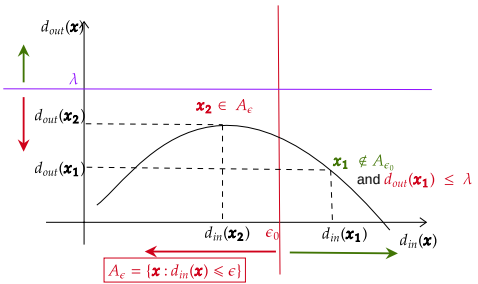
Margin Consistency

Theorem

Margin consistency is a necessary and sufficient condition for the logit margin to perfectly separate non-robust samples from robust samples.



MC $\implies$ logit margin can perfectly discriminate non-robust samples.



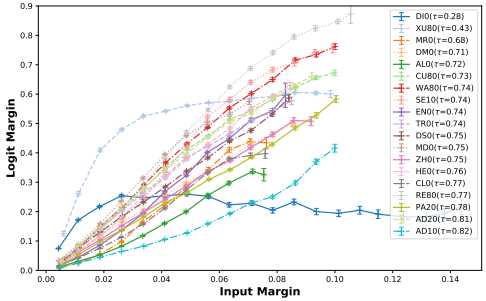
No MC $\implies$ logit margin cannot perfectly discriminate non-robust samples.

Experimental Setup

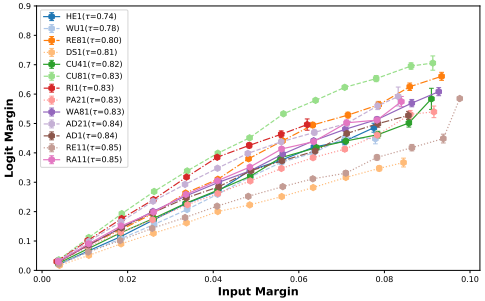
- **Datasets:** CIFAR10 and CIFAR100 datasets.
- **Robustness:** ℓ_∞ norm, threshold $\epsilon = 8/255 \approx 0.031$
- **Models:**
  - Loaded from the *RobustBench* model zoo[1] (Croce et al., 2021)
  - ResNet-18 models trained on CIFAR10 with Standard Adversarial Training (Madry et al., 2018), TRADES (Zhang et al., 2019), Logit Pairing (ALP and CLP, Kannan et al. (2018)), and MART (Wang et al., 2020).
- **Input margin estimation:** $d_{in}(x) = \|\mathbf{x} - \mathbf{x}'\|$
  - $\mathbf{x}'$ is the FAB attack (Croce & Hein, 2020) adversarial example.
  - The estimation of the input margins over the $10,000$ test samples
  - Pool of vulnerable samples that can be successfully attacked at threshold ϵ: $\{\mathbf{x} : d_{in}(\mathbf{x}) \leq \epsilon\}$.

[1] https://github.com/RobustBench/robustbench

Margin Consistency



CIFAR10



CIFAR100

Figure: Correlation between Input Margins and Logit margins

Vulnerable Samples Detection

	Model ID	Kendall τ (↑)	AUROC (↑)	AUPR (↑)	FPR@95 (↓)	Acc	Rob. Acc	Architecture
CIFAR10	**DI0** (Wu et al., 2020)	0.28	67.49	70.91	82.56	84.36	41.44	WideResNet-28-4
	XU80 (Xu et al., 2023)	0.43	83.30	80.50	83.42	93.69	63.89	WideResNet-28-10
	MR0 (Wang et al., 2020)	0.68	92.95	94.92	29.76	79.69	39.12	ResNet-18
	DM0 (Debenedetti et al., 2023)	0.71	94.31	93.20	32.76	91.30	57.27	XCiT-M12
	AL0 (Kannan et al., 2018)	0.72	94.67	95.98	24.93	80.38	40.21	ResNet-18
	CU80 (Cui et al., 2023)	0.73	96.87	94.42	17.90	92.16	67.73	WideResNet-28-10
	WA80 (Wang et al., 2023)	0.74	96.82	94.33	17.60	92.44	67.31	WideResNet-28-10
	SE10 (Sehwag et al., 2021)	0.74	96.03	94.66	19.13	84.59	55.54	ResNet-18
	EN0 (Engstrom et al., 2019)	0.74	95.16	95.07	24.10	87.03	49.25	ResNet-50
	TR0 (Zhang et al., 2019)	0.74	94.63	96.13	30.93	80.72	42.23	ResNet-18
	DS0 (Debenedetti et al., 2023)	0.75	95.80	95.08	24.65	90.06	56.14	XCiT-S12
	MD0 (Madry et al., 2018)	0.75	95.36	97.00	23.23	81.85	36.91	ResNet-18
	ZH0 (Zhang et al., 2019)	0.75	95.86	95.65	24.91	84.92	53.08	WideResNet-34-10
	HE0 (Hendrycks et al., 2019)	0.76	96.35	95.68	20.01	87.11	54.92	WideResNet-28-10
	CL0 (Kannan et al., 2018)	0.77	95.93	96.98	20.01	81.12	40.08	ResNet-18
	RE80 (Rebuffi et al., 2021)	0.77	97.33	95.70	13.87	87.33	60.73	WideResNet-28-10
	PA20 (Pang et al., 2022)	0.78	97.65	96.39	14.40	88.61	61.04	WideResNet-28-10
	AD20 (Addepalli et al., 2022)	0.81	97.67	97.46	13.42	85.71	52.48	ResNet-18
	AD10 (Addepalli et al., 2021)	0.82	97.86	97.68	13.26	80.24	51.06	ResNet-18
CIFAR100	**HE1** (Hendrycks et al., 2019)	0.74	94.43	97.39	30.40	59.23	28.42	WideResNet-28-10
	WU1 (Wu et al., 2020)	0.78	95.81	98.00	23.34	60.38	28.86	WideResNet-34-10
	RE81 (Rebuffi et al., 2021)	0.80	96.87	98.30	18.06	62.41	32.06	WideResNet-28-10
	DS1 (Debenedetti et al., 2023)	0.81	96.78	98.30	19.18	67.34	32.19	XCiT-S12
	CU41 (Cui et al., 2023)	0.82	97.07	98.48	17.21	64.08	31.65	WideResNet-34-10
	CU81 (Cui et al., 2023)	0.83	97.41	98.24	15.62	73.85	39.18	WideResNet-28-10
	RI1 (Rice et al., 2020)	0.83	96.61	99.05	18.14	53.83	18.95	PreActResNet-18
	PA21 (Pang et al., 2022)	0.83	97.66	98.82	13.83	63.66	31.08	WideResNet-28-10
	WA81 (Wang et al., 2023)	0.83	97.51	98.28	14.96	72.58	38.83	WideResNet-28-10
	AD21 (Addepalli et al., 2022)	0.84	97.46	98.92	16.00	65.45	27.67	ResNet-18
	AD1 (Addepalli et al., 2021)	0.84	97.65	98.99	13.88	62.02	27.14	PreActResNet-18
	RE11 (Rebuffi et al., 2021)	0.85	97.97	99.05	13.21	56.87	28.50	PreActResNet-18
	RA11 (Rade & Moosavi-Dezfooli, 2021)	0.85	98.01	99.08	12.36	61.50	28.88	PreActResNet-18

Table: Correlations and vulnerable points detection performance at $\epsilon = 8/255$

Vulnerable Samples Detection

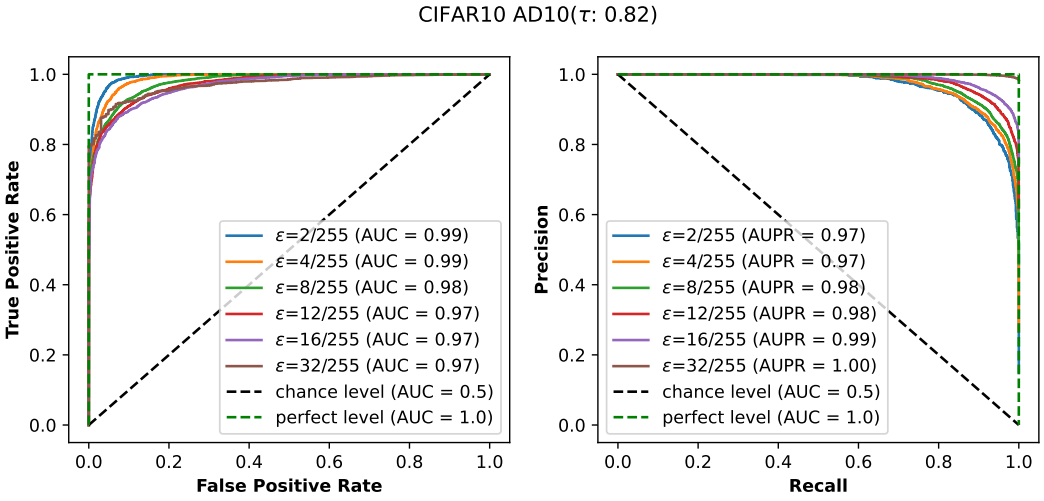


Figure: ROC and PR curves at different threshold levels

Sample Efficient Robustness Estimation

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

- Select uniformly at random a subset X_n of n samples from X.

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

- Select uniformly at random a subset X_n of n samples from X.
- Compute the estimations of the input margins on X_n, $D_n = \{\hat{d}_{in}(\mathbf{x}) : \mathbf{x} \in X_s\}$

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

- Select uniformly at random a subset X_n of n samples from X.
- Compute the estimations of the input margins on X_n, $D_n = \{\hat{d}_{in}(\mathbf{x}) : \mathbf{x} \in X_s\}$
- Create ground truth labels for vulnerability at threshold ϵ i.e. $\mathbb{1}_{[\hat{d}_{in}(\mathbf{x}) \leq \epsilon]}(\mathbf{x})$

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

- Select uniformly at random a subset X_n of n samples from X.
- Compute the estimations of the input margins on X_n, $D_n = \{\hat{d}_{in}(\mathbf{x}) : \mathbf{x} \in X_s\}$
- Create ground truth labels for vulnerability at threshold ϵ i.e. $\mathbb{1}_{[\hat{d}_{in}(\mathbf{x}) \leq \epsilon]}(\mathbf{x})$
- Determine the threshold λ of d_{out} that gives best approximation of robust accuracy on X_s.

Sample Efficient Robustness Estimation

Algorithm Sample Efficient Robustness Estimation

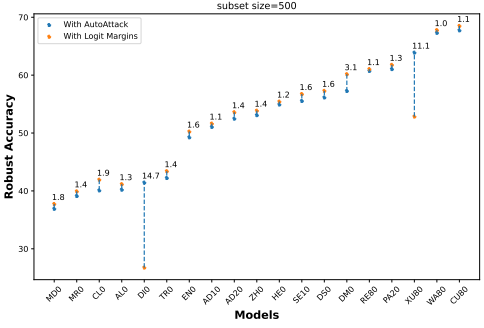
Input: Test Dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^N$, Robustness threshold $\epsilon > 0$, Subset size $n \ll N$.

Output: Robust Accuracy Estimation $\mathcal{A}_r$.

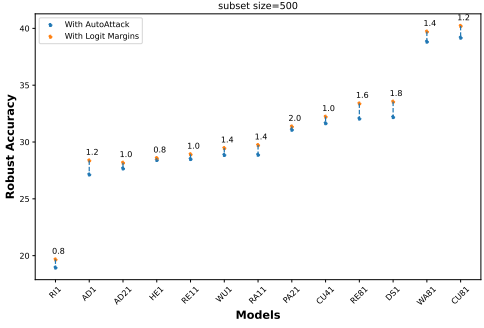
- Select uniformly at random a subset X_n of n samples from X.
- Compute the estimations of the input margins on X_n, $D_n = \{\hat{d}_{in}(\mathbf{x}) : \mathbf{x} \in X_s\}$
- Create ground truth labels for vulnerability at threshold ϵ i.e. $\mathbb{1}_{[\hat{d}_{in}(\mathbf{x}) \leq \epsilon]}(\mathbf{x})$
- Determine the threshold λ of d_{out} that gives best approximation of robust accuracy on X_s.

Return: $\mathcal{A}_r = |\{\mathbf{x} \in X : d_{out}(\mathbf{x}) > \lambda \text{ and } \hat{y}(\mathbf{x}) = y\}|/N$

Sample Efficient Robustness Estimation



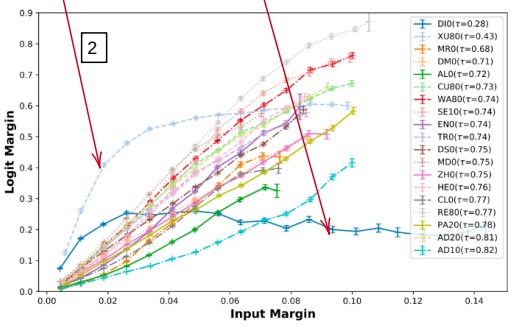
CIFAR10



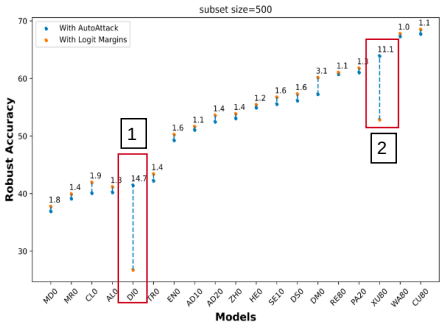
CIFAR100

What about weakly margin consistent models?

Model ID	Kendall τ (↑)	AUROC (↑)	AUPR (↑)	FPR@95 (↓)	Acc	Rob. Acc	Architecture
DI0 (Wu et al., 2020)	0.28	67.49	70.91	82.56	84.36	41.44	WideResNet-28-4
XU80 (Xu et al., 2023)	0.43	83.30	80.50	83.42	93.69	63.89	WideResNet-28-10



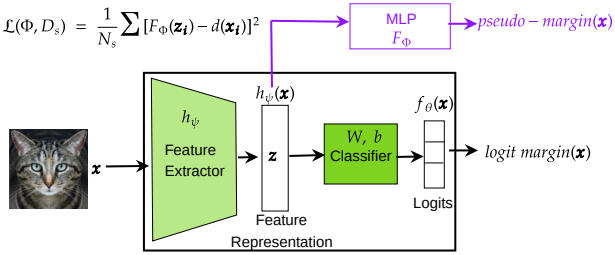
CIFAR10



CIFAR10

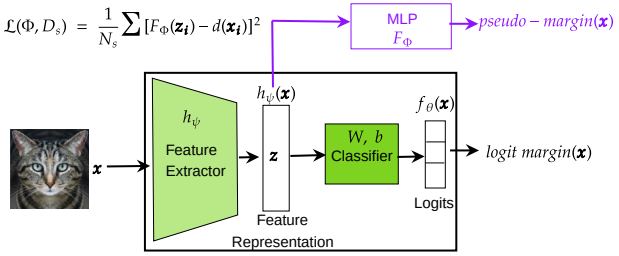
Learning a pseudo-margin (for weak margin consistent models)

Learning a pseudo-margin (for weak margin consistent models)

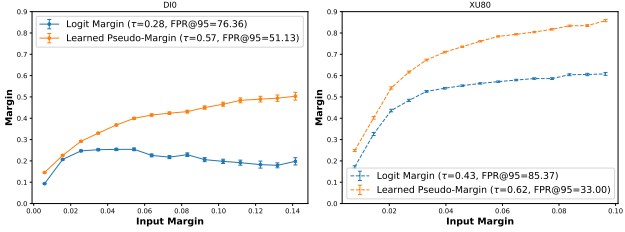


Corbière et al. "Addressing failure prediction by learning model confidence." Neurips 2019

Learning a pseudo-margin (for weak margin consistent models)

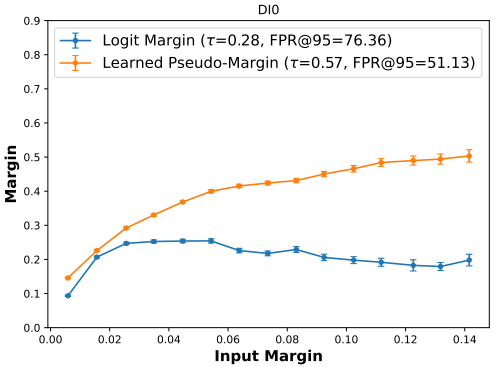


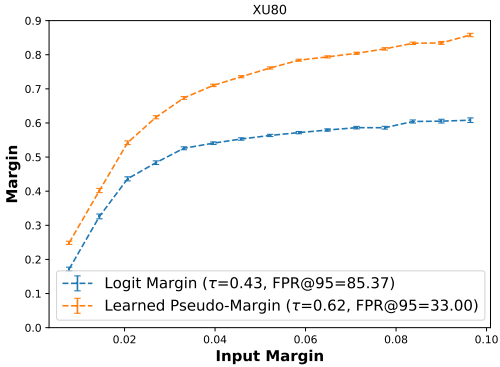
Corbière et al. "Addressing failure prediction by learning model confidence." Neurips 2019



Learning a pseudo-margin (for weak margin consistent models)

Model ID	Margin	Kendall τ (↑)	AUROC (↑)	AUPR (↑)	FPR@95 (↓)	Acc.	Rob. Acc
DI0 (Ding et al., 2020)	Logit margin	0.28	67.49	70.91	82.56	84.36	41.44
	Learned pseudo-margin	**0.57**	**88.49**	**89.04**	**51.13**		
XU80 (Xu et al., 2023)	Logit margin	0.43	83.30	80.50	83.42	93.69	63.89
	Learned pseudo-margin	**0.62**	**93.66**	**90.22**	**33.00**		





Margin Consistency vs Robustness/Lipschitz Smoothness

- L-Lipschitz: $\underbrace{\|f(x) - f(x')\|}_{\text{output}} \leq L \underbrace{\|x - x'\|}_{\text{input}}$

Margin Consistency vs Robustness/Lipschitz Smoothness

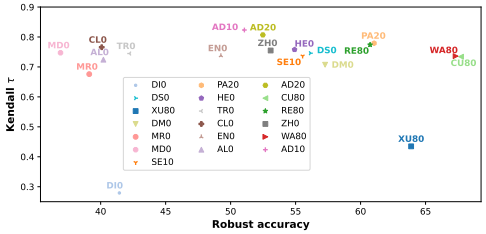
- L-Lipschitz: $\underbrace{\|f(x) - f(x')\|}_{\text{output}} \leq L \underbrace{\|x - x'\|}_{\text{input}}$ (smaller $L \implies$ higher robustness)

Margin Consistency vs Robustness/Lipschitz Smoothness

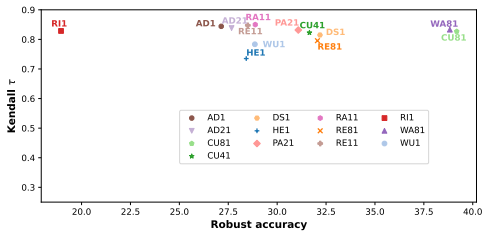
- L-Lipschitz: $\underbrace{\|f(x) - f(x')\|}_{\text{output}} \leq L \underbrace{\|x - x'\|}_{\text{input}}$ (smaller $L \implies$ higher robustness)
- Robustness/Lipschitz smoothness does not imply margin consistency

Margin Consistency vs Robustness/Lipschitz Smoothness

- L-Lipschitz: $\underbrace{\|f(x) - f(x')\|}_{\text{output}} \leq L \underbrace{\|x - x'\|}_{\text{input}}$ (smaller $L \implies$ higher robustness)
- Robustness/Lipschitz smoothness does not imply margin consistency



CIFAR10

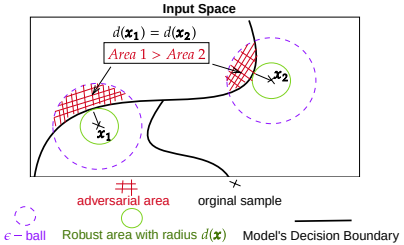


CIFAR100

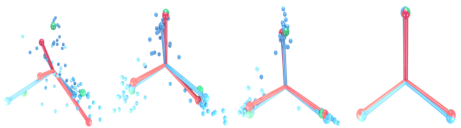
Figure: Strength of the correlation independent of the robust accuracy.

Perspectives

Vulnerability Scope: Worst vs Avg.



Influence of Neural Collapse



Papyan et al. "Prevalence of neural collapse during the terminal phase of deep learning training." Proceedings of the National Academy of Sciences 117.40 (2020): 24652-24663.

Conclusion

- We introduce **margin consistency**, order preservation property of input margins in the feature representation space

Conclusion

- We introduce **margin consistency**, order preservation property of input margins in the feature representation space
- Margin consistency is a necessary and sufficient condition to use the logit margin for non-robustness detection

Conclusion

- We introduce **margin consistency**, order preservation property of input margins in the feature representation space
- Margin consistency is a necessary and sufficient condition to use the logit margin for non-robustness detection
- Majority of Robust Deep Classifiers are strongly margin consistent

Conclusion

- We introduce **margin consistency**, order preservation property of input margins in the feature representation space
- Margin consistency is a necessary and sufficient condition to use the logit margin for non-robustness detection
- Majority of Robust Deep Classifiers are strongly margin consistent
- The logit margin of these classifiers can detect vulnerable samples and predict robust accuracy at scale using few samples

Conclusion

- We introduce **margin consistency**, order preservation property of input margins in the feature representation space
- Margin consistency is a necessary and sufficient condition to use the logit margin for non-robustness detection
- Majority of Robust Deep Classifiers are strongly margin consistent
- The logit margin of these classifiers can detect vulnerable samples and predict robust accuracy at scale using few samples
- It is possible to learn a better-correlated pseudo-margin in case of weak margin consistency

Thank you.

Questions?

Bibliography I

Addepalli, S., Jain, S., Sriramanan, G., Khare, S., and Radhakrishnan, V. B. Towards achieving adversarial robustness beyond perceptual limits. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL https://openreview.net/forum?id=SHB_znlW5G7.

Addepalli, S., Jain, S., Sriramanan, G., and Venkatesh Babu, R. Scaling adversarial training to large perturbation bounds. In *European Conference on Computer Vision*, pp. 301–316. Springer, 2022.

Croce, F. and Hein, M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020.

Bibliography II

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL `https://openreview.net/forum?id=SSKZPJCt7B`.

Cui, J., Tian, Z., Zhong, Z., Qi, X., Yu, B., and Zhang, H. Decoupled kullback-leibler divergence loss. *arXiv preprint arXiv:2305.13948*, 2023.

Debenedetti, E., Sehwag, V., and Mittal, P. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 225–253. IEEE, 2023.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkeryxBtPB`.

Bibliography III

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019.

Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Pang, T., Lin, M., Yang, X., Zhu, J., and Yan, S. Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, pp. 17258–17277. PMLR, 2022.

Bibliography IV

Rade, R. and Moosavi-Dezfooli, S.-M. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL `https://openreview.net/forum?id=BuD2LmNaU3a`.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021.

Bibliography V

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rklOg6EFwS.

Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., and Yan, S. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning (ICML)*, 2023.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Xu, Y., Sun, Y., Goldblum, M., Goldstein, T., and Huang, F. Exploring and exploiting decision boundary dynamics for adversarial robustness. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=aRTKuscKByJ.

Bibliography VI

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.