

Robustness in Deep learning

Grigorios Chrysos

Assistant Professor
chrysos@wisc.edu



August 11, 2025

Machine Learning Summer School

Golden era for Machine Learning (ML)

You

Make a proof of the following property: Upper triangular matrices with all elements in their diagonal being non-zero and every second element in the diagonal is equal to 1, are full rank.

The proof should be in markdown format such that it is parsed correctly here. Make the proof brief to fit in 15 lines

Copilot

Sure, here's a brief proof in Markdown format:

```
==Proof==

Consider an upper triangular matrix  $A$  of size  $n \times m$  with the following properties:
1. All diagonal elements are non-zero.
2. Every second element in the diagonal is equal to 1.

We need to show that  $A$  is full rank, i.e.,  $\text{rank}(A) = n$ .

1. ==Upper Triangular Matrix==: By definition, all elements below the main diagonal are zero.
2. ==Non-zero Diagonal Elements==: Since all diagonal elements are non-zero,  $A$  has no zero rows or columns.
3. ==Determinant==: The determinant of an upper triangular matrix is the product of its diagonal elements.
4. ==Diagonal Elements==: Given that every second diagonal element is 1, the product of the diagonal elements is non-zero.
5. ==Non-zero Determinant==: Since the product of the diagonal elements is non-zero, the determinant of  $A$  is non-zero.
6. ==Full Rank==: A matrix with a non-zero determinant is invertible and thus has full rank.
```



Golden era for Machine Learning (ML)

You

Make a proof of the following property: Upper triangular matrices with all elements in their diagonal being non-zero and every second element in the diagonal is equal to 1, are full rank.

The proof should be in markdown format such that it is parsed correctly here. Make the proof brief to fit in 15 lines

Copilot

Sure, here's a brief proof in Markdown format:

```
==Proof==

Consider an upper triangular matrix  $\begin{pmatrix} A \end{pmatrix}$  of size  $\begin{pmatrix} n \times m \end{pmatrix}$  with the following properties:
1. All diagonal elements are non-zero.
2. Every second element in the diagonal is equal to 1.

We need to show that  $\begin{pmatrix} A \end{pmatrix}$  is full rank, i.e.,  $\begin{pmatrix} \text{rank}(A) = n \end{pmatrix}$ .

1. ==Upper Triangular Matrix==: By definition, all elements below the main diagonal are zero.
2. ==Non-zero Diagonal Elements==: Since all diagonal elements are non-zero,  $\begin{pmatrix} A \end{pmatrix}$  has no zero rows or columns.
3. ==Determinant==: The determinant of an upper triangular matrix is the product of its diagonal elements.
4. ==Diagonal Elements==: Given that every second diagonal element is 1, the product of the diagonal elements is non-zero.
5. ==Non-zero Determinant==: Since the product of the diagonal elements is non-zero, the determinant of  $\begin{pmatrix} A \end{pmatrix}$  is non-zero.
6. ==Full Rank==: A matrix with a non-zero determinant is invertible and thus has full rank.
```

GR

Can you explain the biggest issues of Machine Learning to my grandma in terms that relate with her view of the world?

GR

Sure, I'll try my best to explain the biggest issues in Machine Learning in a way that relates to your grandma's view of the world.

One of the biggest issues in Machine Learning is that it requires a lot of data to work well. Imagine if your grandma wanted to bake a cake from scratch, but didn't have all the ingredients or the right recipe. Without all the necessary information, the cake might not turn out the way she wants it to. Similarly, Machine Learning algorithms need a lot of data to learn and make accurate predictions. If the data is not diverse, accurate or enough the algorithm may not work well or even return a bias result.

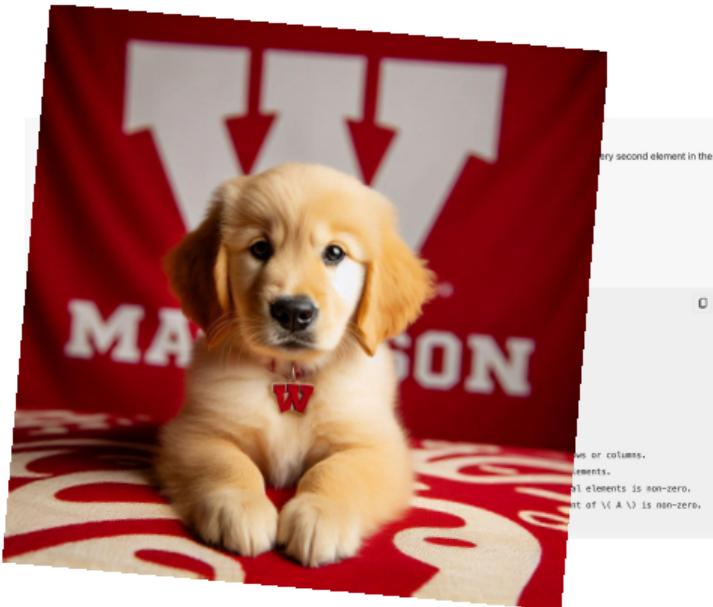
Another big issue is that Machine Learning algorithms can be very complex, and it can be difficult to understand how they make decisions. Imagine if your grandma was following a recipe for a cake, but the instructions were written in a foreign language that she doesn't understand. She wouldn't know what to do next, and the cake might not turn out the way she wants it to. Similarly, it can be challenging to understand how Machine Learning algorithms make decisions, which can make it difficult to trust the predictions they make.

A final issue is that Machine Learning algorithms can be very sensitive to the parameters that are used to train them. Imagine if your grandma was baking a cake and she used a little too much or too little of one ingredient. The cake might not turn out the way she wants it to. Similarly, Machine Learning algorithms can be sensitive to the parameters that are used to train them, and small changes in these parameters can lead to large changes in the predictions made by the algorithm.

All these challenges are important to be considered and addressed by Machine Learning experts to make the most accurate and reliable predictions for the best solutions for different fields.



Golden era for Machine Learning (ML)



GR Can you explain the biggest issues of Machine Learning to my grandma in terms that relate with her view of the world?

GR Sure, I'll try my best to explain the biggest issues in Machine Learning in a way that relates to your grandma's view of the world.

One of the biggest issues in Machine Learning is that it requires a lot of data to work well. Imagine if your grandma wanted to bake a cake from scratch, but didn't have all the ingredients or the right recipe. Without all the necessary information, the cake might not turn out the way she wants it to. Similarly, Machine Learning algorithms need a lot of data to learn and make accurate predictions. If the data is not diverse, accurate or enough the algorithm may not work well or even return a bias result.

Another big issue is that Machine Learning algorithms can be very complex, and it can be difficult to understand how they make decisions. Imagine if your grandma was following a recipe for a cake, but the instructions were written in a foreign language that she doesn't understand. She wouldn't know what to do next, and the cake might not turn out the way she wants it to. Similarly, it can be challenging to understand how Machine Learning algorithms make decisions, which can make it difficult to trust the predictions they make.

A final issue is that Machine Learning algorithms can be very sensitive to the parameters that are used to train them. Imagine if your grandma was baking a cake and she used a little too much or too little of one ingredient. The cake might not turn out the way she wants it to. Similarly, Machine Learning algorithms can be sensitive to the parameters that are used to train them, and small changes in these parameters can lead to large changes in the predictions made by the algorithm.

All these challenges are important to be considered and addressed by Machine Learning experts to make the most accurate and reliable predictions for the best solutions for different fields.



Golden era for Machine Learning (ML)

6 I want to create an overlapping effect in a beamer presentation in Latex. I have used the following code for adding the images, now I want you to include two new images that are overlaid over the previous in onslide effects 3 and 4, where the new images will also be rotated by 10 degrees.

```
\begin{tikzpicture}
  \onslide<1-3>{
    \node[anchor=west,inner sep=0] (image) at {-6.5,0} {\raisebox{4mm}{\includegraphics[width=0.44\ linewidth]{figures/introduction/chaptpt_minimum_keras_code_classification.jpg}}};
    \node[anchor=west,inner sep=0] (image) at {0,0} {\includegraphics[width=0.4\ linewidth]{figures/introduction/chaptpt_biggest_challenges_grandma.jpg}};
  }
  \onslide<1-2>{
    \draw[white, thick, fill=white, rectangle] (-6.5, -4) rectangle (-0.3, 2.35);
    \draw[white, thick, fill=white, rectangle] (0, -4) rectangle (5.8, 2.7); % only covering the answer in the second image.
  }
  \onslide<1-2>{
    \draw[white, thick, fill=white, rectangle] (0, -4) rectangle (5.8, 3.3);
  }
\end{tikzpicture}
```

I'll help you modify the code to add two new overlaid and rotated images in onslide effects 3 and 4.

Beamer presentation with overlapping images
Click to open code

Key changes made to the code:
1. Extended the visibility of original images to slide 4 (`\onslide<1-4>`)

Overlapping Images in Beamer Presentation ▾

← Beamer presentation with overlapping images

```
\begin{tikzpicture}
  % Original images (slides 1-4)
  \onslide<1-4>{
    \node[anchor=west,inner sep=0] (image1) at {-6.5,0} {\raisebox{4mm}{\includegraphics[width=0.44\ linewidth]{figures/introduction/chaptpt_minimum_keras_code_classification.jpg}}};
    \node[anchor=west,inner sep=0] (image2) at {0,0} {\includegraphics[width=0.4\ linewidth]{figures/introduction/chaptpt_biggest_challenges_grandma.jpg}};
  }

  % White rectangles for initial hiding (slide 1)
  \onslide<1>{
    \draw[white, thick, fill=white, rectangle] (-6.5, -4) rectangle (-0.3, 2.35);
  }

  % White rectangle for second image (slide 1-2)
  \onslide<1-2>{
    \draw[white, thick, fill=white, rectangle] (0, -4) rectangle (5.8, 3.3);
  }

  % FIRST overlaid rotated image (slide 1)
  \onslide<1>{
    \node[anchor=west,inner sep=0,rotate=10] at {-6.5,0} {\raisebox{4mm}{\includegraphics[width=0.44\ linewidth]{figures/introduction/chaptpt_minimal_keras_code_classification.jpg}}};
  }

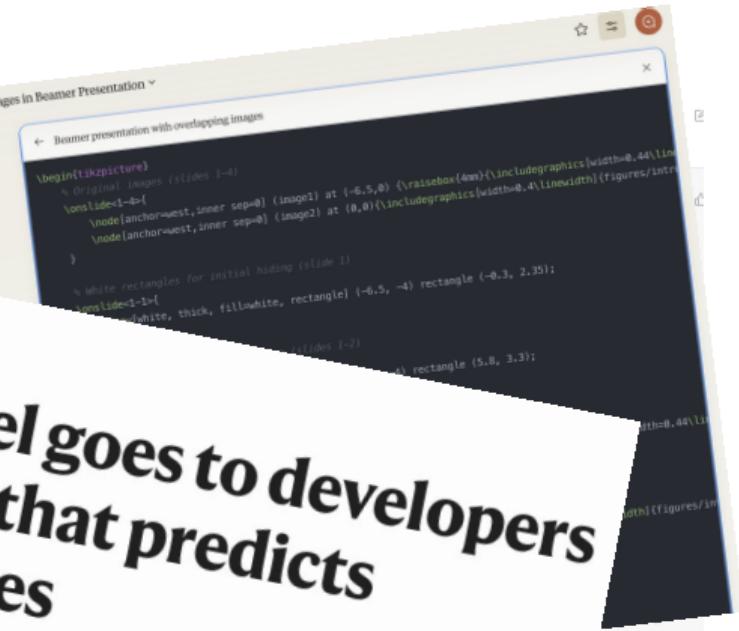
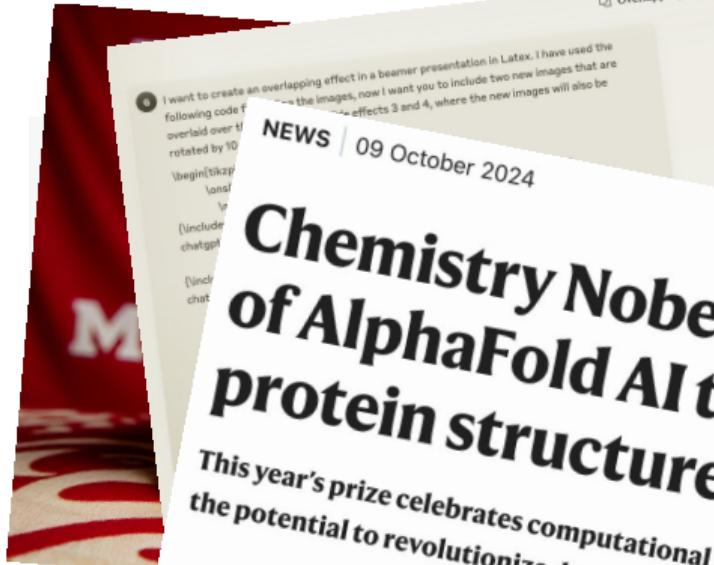
  % Second overlaid rotated image (slide 4)
  \onslide<4>{
    \node[anchor=west,inner sep=0,rotate=10] at {0,0} {\includegraphics[width=0.4\ linewidth]{figures/introduction/chaptpt_biggest_challenges_grandma.jpg}};
  }
\end{tikzpicture}
```

Machine Learning algorithms can not turn out the way she is expected to. Machine Learning algorithms can be sensitive to the parameters that are used to train them, and small changes in these parameters can lead to large changes in the predictions made by the algorithm.

All these challenges are important to be considered and addressed by Machine Learning experts to make the most accurate and reliable predictions for the best solutions for different fields.



Golden era for Machine Learning (ML)



Golden era for Machine Learning (ML)



Sign in | Enter search | This week's magazine

News Features Newsletters Podcasts Video Comment Culture Games | This week's magazine

Health Space Physics Technology Environment Mind Humans Life Mathematics Chemistry Earth Society

Mathematics

DeepMind and OpenAI claim gold in International Mathematical Olympiad

Two AI models have achieved gold medal standard for the first time in a prestigious competition for young mathematicians – and their developers claim these AIs could soon crack tough scientific problems

By Alex Wilkins

22 July 2025



“...numerical tools that have transformed biology and have drug discovery.



Quiz time

Are the two sentences below the same?

- ▶ *Sentence 1* : Ky. Company Wins Grant to Study Peptides (AP) AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Sentence 2* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.



Quiz time

Are the two sentences below the same?

- ▶ *Sentence 1* : Ky. Company Wins Grant to Study Peptides (AP) AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Sentence 2* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Difference* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.



Quiz time

Are the two sentences below the same?

- ▶ *Sentence 1* : Ky. Company Wins Grant to Study Peptides (AP) AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Sentence 2* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Difference* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.

Harder to achieve when there is no reference text.



Quiz time

Are the two sentences below the same?

- ▶ *Sentence 1* : Ky. Company Wins Grant to Study Peptides (AP) AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Sentence 2* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
- ▶ *Difference* : Ky. Company Wins Grant to StuJdy Peptides (AP) AFP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.

Harder to achieve when there is no reference text.



Is there a scenario that concerns ME?



MATT BURGESS SECURITY AUG 6, 2025 9:00 AM

Hackers Hijacked Google's Gemini AI With a Poisoned Calendar Invite to Take Over a Smart Home

For likely the first time ever, security researchers have shown how AI can be hacked to create real-world havoc, allowing them to turn off lights, open smart shutters, and more.



Motivation: Why Do We Care?

Real-World Implications:

► Autonomous Vehicles

- Stop sign → Speed limit sign
- Traffic light misclassification

► Medical Imaging Systems

- Benign → Malignant tumor
- Patient risk assessment

► Security Systems

- Office security manipulation
- Spam filter manipulation

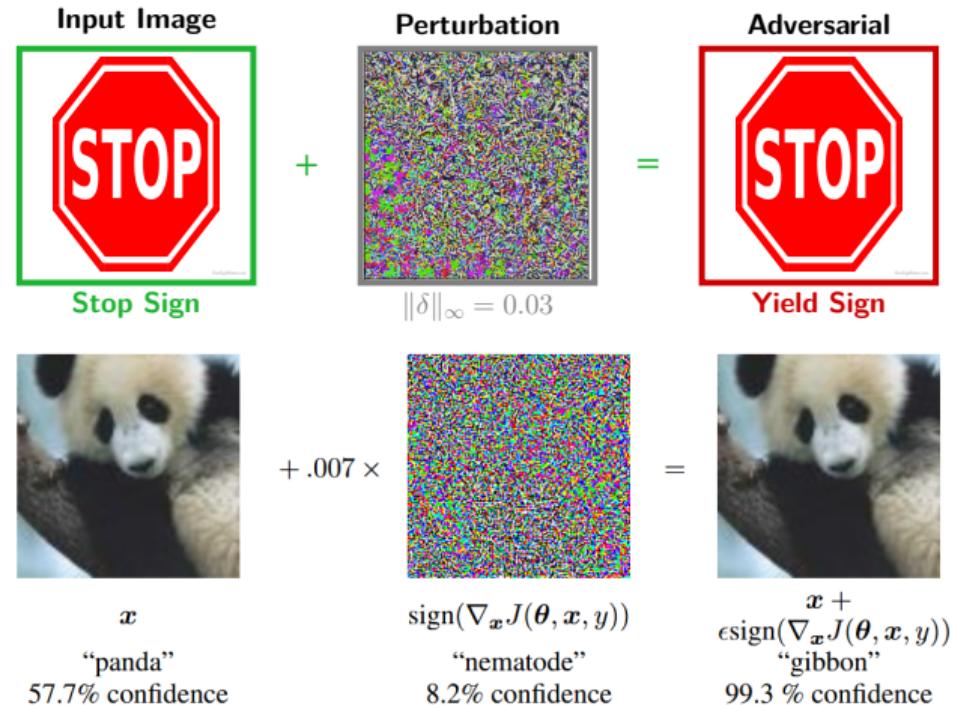


Figure from: Goodfellow et al., "Explaining and Harnessing Adversarial Examples", 2014



Outline

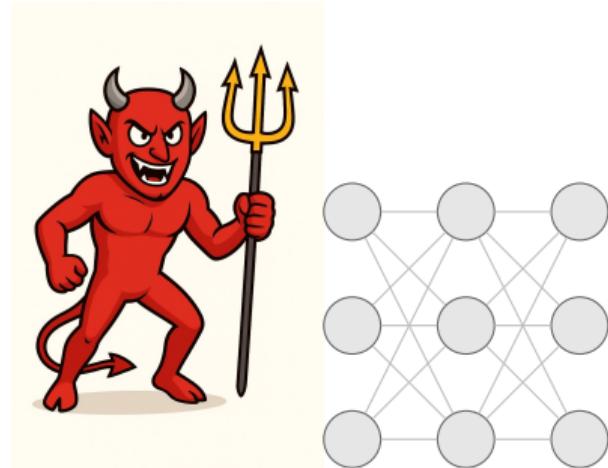
Adversarial robustness: Where it all started

Defense against adversarial attacks

Adversarial robustness in text

Adversarial certificates

Challenges and future work



Robustness on images



Figure:

Szegedy, et al. 'Intriguing properties of neural networks.' In *ICLR*, 2014.

¹Image from the NeurIPS 2018 tutorial, "Adversarial Robustness: Theory and Practice", by Zico Kolter and Aleksander Madry.
<https://adversarial-ml-tutorial.org/>



Robustness on images

“pig”



“airliner”



Figure:

Szegedy, et al. ‘Intriguing properties of neural networks.’ In *ICLR*, 2014.

¹Image from the NeurIPS 2018 tutorial, “Adversarial Robustness: Theory and Practice”, by Zico Kolter and Aleksander Madry.
<https://adversarial-ml-tutorial.org/>



Robustness on images

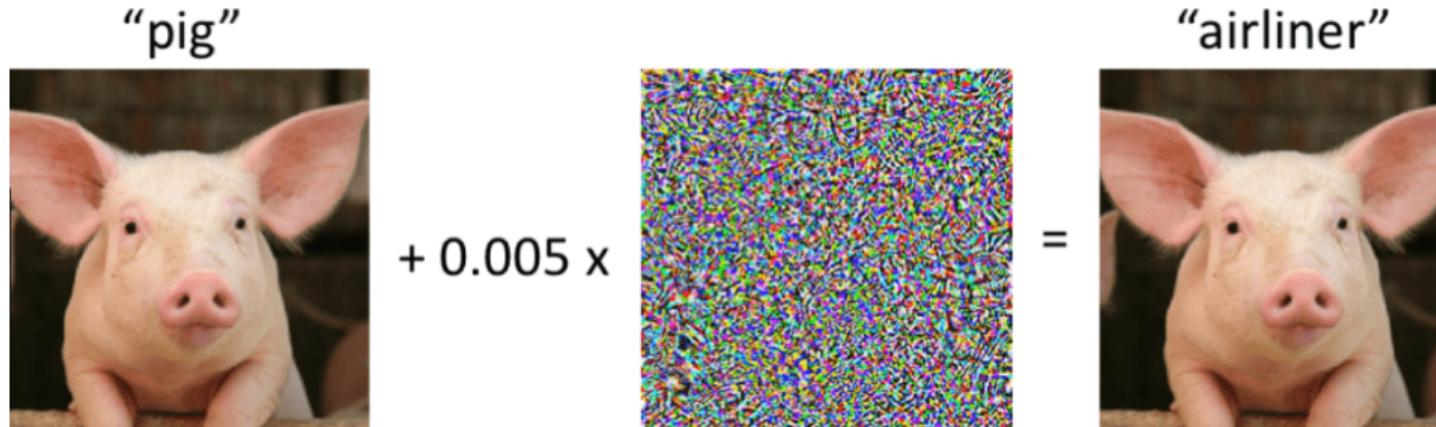
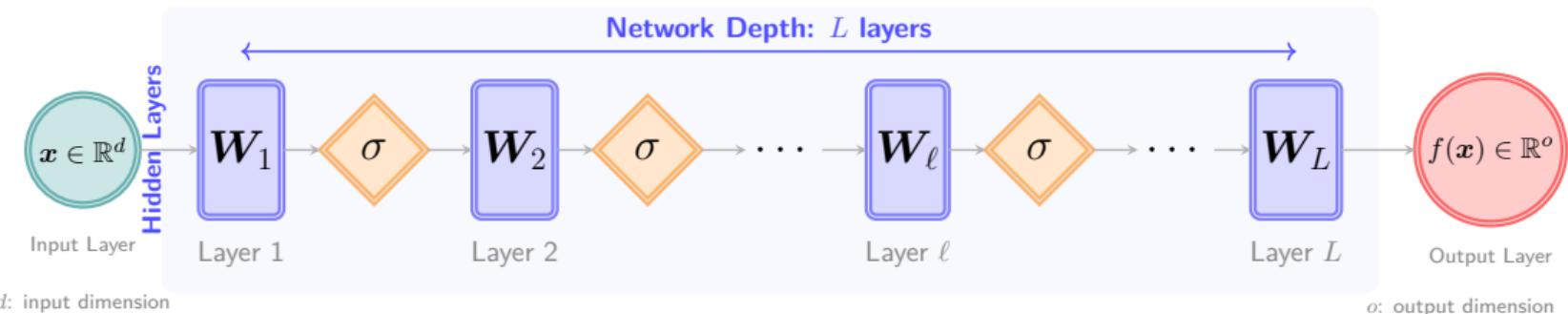


Figure: An adversarial perturbation can affect dramatically a neural network.¹

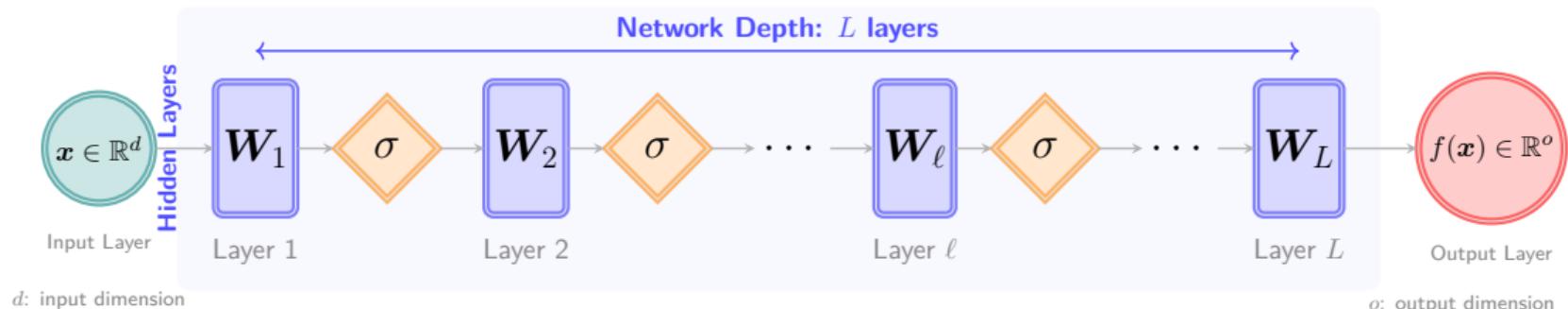
Szegedy, et al. 'Intriguing properties of neural networks.' In *ICLR*, 2014.

¹Image from the NeurIPS 2018 tutorial, "Adversarial Robustness: Theory and Practice", by Zico Kolter and Aleksander Madry.
<https://adversarial-ml-tutorial.org/>

Multi-layer neural architecture



Multi-layer neural architecture



$$f^{(0)}(\mathbf{x}) = \mathbf{x},$$

$$f^{(l)}(\mathbf{x}) = \sigma \left(\begin{bmatrix} \text{activation} \\ \downarrow \end{bmatrix} \begin{bmatrix} \text{weight} \\ \downarrow \\ \mathbf{W}_l \end{bmatrix} \begin{bmatrix} \text{input features} \\ \downarrow \\ f^{(l-1)}(\mathbf{x}) \end{bmatrix} \right), \quad (\text{MLP})$$

$$f_{\mathbf{w}}(\mathbf{x}) = f^{(L)}(\mathbf{x}) = \mathbf{W}_L f^{(L-1)}(\mathbf{x}), \quad \theta := [W_1, W_2, \dots, W_L].$$

Parameters: $\mathbf{W}_1 \in \mathbb{R}^{m \times d}$, $\mathbf{W}_L \in \mathbb{R}^{o \times m}$, $\{\mathbf{W}_l \in \mathbb{R}^{m \times m}\}_{l=2}^{L-1}$ (weights).



Question

Can a small perturbation to *the input signal* lead to a large change in the model's output?



Adversarial Examples

Definition

Let $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ be a model with parameters θ and let (x, y) , with $x \in \mathbb{R}^p$ and y be the input and corresponding label. The adversarial example is constructed by

$$\max_{\delta} \{\mathcal{L}(f_\theta(x + \delta), y) \quad \text{subject to} \quad \|\delta\|_\infty \leq \epsilon\} \quad (1)$$

where \mathcal{L} is the loss function and ϵ is the perturbation budget.

- Note that the goal is to find δ such that $f_\theta(x + \delta) \neq y$ while keeping $\|\delta\|_\infty$ small.



Adversarial Examples

Definition

Let $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ be a model with parameters θ and let (x, y) , with $x \in \mathbb{R}^p$ and y be the input and corresponding label. The adversarial example is constructed by

$$\max_{\delta} \{\mathcal{L}(f_\theta(x + \delta), y) \quad \text{subject to} \quad \|\delta\|_\infty \leq \epsilon\} \quad (1)$$

where \mathcal{L} is the loss function and ϵ is the perturbation budget.

- Note that the goal is to find δ such that $f_\theta(x + \delta) \neq y$ while keeping $\|\delta\|_\infty$ small.
- Different norms can be used: ℓ_∞ (max perturbation per pixel), ℓ_2 (Euclidean distance), ℓ_0 (number of pixels changed), etc.



Adversarial Attacks in Images

- ▶ **Gradient-based attacks:** L-BFGS (Szegedy et al., ICLR 2014), FGSM (Goodfellow et al., ICLR 2015), DeepFool (Moosavi-Dezfooli et al., CVPR 2016), C&W (Carlini & Wagner, S&P 2017), PGD (Madry et al., ICLR 2018), MI-FGSM (Dong et al., CVPR 2018), GAAS (Tramèr et al., ArXiv 2017), SparseFool (Modas et al., CVPR 2019), SmoothFool (Dabouei et al., WACV 2020), ...
- ▶ **Transferable / Black-box attacks:** Universal Adversarial Perturbations (Moosavi-Dezfooli et al., CVPR 2017), NES-based attacks (Ilyas et al., ICML 2018), Square Attack (Andriushchenko et al., ECCV 2020), AdvCam (Duan et al., CVPR 2020), ...
- ▶ **Ensemble attacks:** AutoAttack (Croce & Hein, ICML 2020), Adaptive AutoAttack (Liu et al., CVPR 2022), ...
- ▶ **Universal / Image-agnostic attacks:** UAP (Moosavi-Dezfooli et al., CVPR 2017), UAN (Hayes & Danezis, SPW 2018), ...
- ▶ **Adversarial training & defenses using attacks:** PGD Adversarial Training (Madry et al., ICLR 2018), Ensemble Adversarial Training (Tramèr et al., ICLR 2018), Free-AT (Shafahi et al., NeurIPS 2019), Fast-AT (Wong et al., NeurIPS 2020), ...
- ▶ **Surveys:** Costa et al. (arXiv 2023), Akhtar & Mian (ArXiv 2018), Xu et al. (ArXiv 2020), Chakraborty et al. (ArXiv 2018), ...



Gradient-based Adversarial Attacks

Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015)

$$\begin{aligned}\delta_{\text{FGSM}} &= \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y)), \quad \|\delta_{\text{FGSM}}\|_\infty \leq \epsilon, \\ x_{\text{adv}} &= x + \delta_{\text{FGSM}}.\end{aligned}$$

- Keeps the *direction* of the gradient but forces every pixel to move by the maximal allowable amount ϵ .



Gradient-based Adversarial Attacks

Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015)

$$\begin{aligned}\delta_{\text{FGSM}} &= \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y)), \quad \|\delta_{\text{FGSM}}\|_\infty \leq \epsilon, \\ x_{\text{adv}} &= x + \delta_{\text{FGSM}}.\end{aligned}$$

- Keeps the *direction* of the gradient but forces every pixel to move by the maximal allowable amount ϵ .

Question

What are the limitations of FGSM?



Gradient-based Adversarial Attacks

Projected Gradient Descent (PGD) (Madry et al., 2018)

Let d be the input dimension and T the number of maximum iterations. Let $\delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)^d$. Then:

$$\begin{aligned}\delta_{t+1} &= \Pi_{\|\cdot\|_\infty \leq \epsilon} (\delta_t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f(x + \delta_t), y))) , \\ x_{\text{adv}} &= x + \delta_T \quad \text{after } T \text{ iterations} .\end{aligned}$$

- Multi-step iterative attack with random initialization and projection to maintain ℓ_∞ constraint.
- Step size $\alpha < \epsilon$ allows for more precise adversarial perturbations.



Gradient-based Adversarial Attacks

Projected Gradient Descent (PGD) (Madry et al., 2018)

Let d be the input dimension and T the number of maximum iterations. Let $\delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)^d$. Then:

$$\begin{aligned}\delta_{t+1} &= \Pi_{\|\cdot\|_\infty \leq \epsilon} (\delta_t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f(x + \delta_t), y))) , \\ x_{\text{adv}} &= x + \delta_T \quad \text{after } T \text{ iterations} .\end{aligned}$$

- Multi-step iterative attack with random initialization and projection to maintain ℓ_∞ constraint.
- Step size $\alpha < \epsilon$ allows for more precise adversarial perturbations.



Gradient-based Adversarial Attacks

Projected Gradient Descent (PGD) (Madry et al., 2018)

Let d be the input dimension and T the number of maximum iterations. Let $\delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)^d$. Then:

$$\begin{aligned}\delta_{t+1} &= \Pi_{\|\cdot\|_\infty \leq \epsilon} (\delta_t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f(x + \delta_t), y))) , \\ x_{\text{adv}} &= x + \delta_T \quad \text{after } T \text{ iterations} .\end{aligned}$$

- Multi-step iterative attack with random initialization and projection to maintain ℓ_∞ constraint.
- Step size $\alpha < \epsilon$ allows for more precise adversarial perturbations.

Question

What are the limitations of PGD?



Question

Most of the developed attacks apply to *any* neural network architecture, i.e., they are architecture-agnostic. Is it true though that the attack success is architecture-agnostic?



The good (width), the bad (depth), and the ugly (initialization)

Definition (Perturbation Stability)

The perturbation stability of a neural network $f_{\theta}(x)$ is defined as follows:

$$\mathcal{P}(f, \epsilon) = \mathbb{E}_{\mathbf{x}, \hat{\mathbf{x}}, \theta} \left\| \nabla_{\mathbf{x}} f_{\theta}(\mathbf{x})^\top (\mathbf{x} - \hat{\mathbf{x}}) \right\|_2, \quad \forall \mathbf{x} \sim \mathcal{D}_X, \quad \hat{\mathbf{x}} \sim \text{Unif}(\mathbb{B}(\epsilon, \mathbf{x})).$$

where θ is the neural network parameter, \mathcal{D}_X is the input data distribution, and ϵ is the perturbation radius. $\text{Unif}(\mathbb{B}(\epsilon, \mathbf{x}))$ means the uniform distribution inside the sphere with the center \mathbf{x} and radius ϵ .



The good (width), the bad (depth), and the ugly (initialization)

Definition (Perturbation Stability)

The perturbation stability of a neural network $f_{\theta}(x)$ is defined as follows:

$$\mathcal{P}(f, \epsilon) = \mathbb{E}_{\mathbf{x}, \hat{\mathbf{x}}, \theta} \left\| \nabla_{\mathbf{x}} f_{\theta}(\mathbf{x})^\top (\mathbf{x} - \hat{\mathbf{x}}) \right\|_2, \quad \forall \mathbf{x} \sim \mathcal{D}_X, \quad \hat{\mathbf{x}} \sim \text{Unif}(\mathbb{B}(\epsilon, \mathbf{x})).$$

where θ is the neural network parameter, \mathcal{D}_X is the input data distribution, and ϵ is the perturbation radius. $\text{Unif}(\mathbb{B}(\epsilon, \mathbf{x}))$ means the uniform distribution inside the sphere with the center \mathbf{x} and radius ϵ .

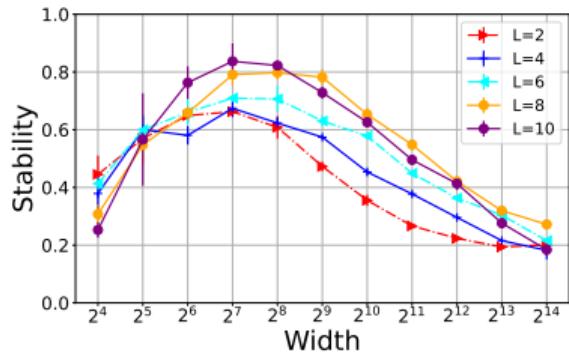
Table: The *perturbation stability* of a deep ReLU-net with depth L , width m under common initializations.

Initialization name	Bound for $\mathcal{P}(f, \epsilon)/\epsilon^*$	Trend w.r.t width	Trend w.r.t depth
LeCun (LeCun et al., 2012)	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}} \right) \left(\frac{\sqrt{2}}{2} \right)^{L-2}$	$\nearrow \searrow$	\swarrow
He (He et al., 2015)	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	$\nearrow \searrow$	\nearrow
NTK (Allen-Zhu et al., 2019)	$\sqrt{L^3 m} e^{-m/L^3} + 1$	$\nearrow \searrow$	\nearrow

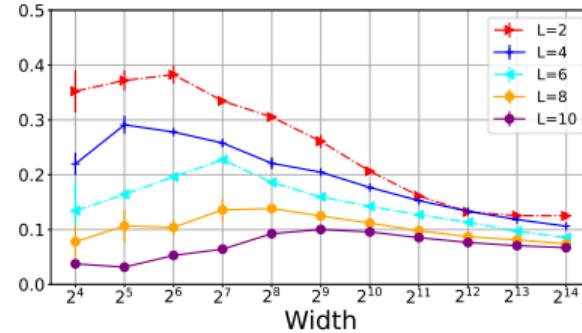
* The larger perturbation stability means worse average robustness.



Experimental validation on MLPs



(a) He initialization

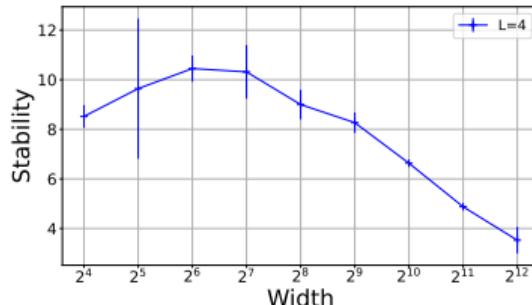


(b) LeCun initialization

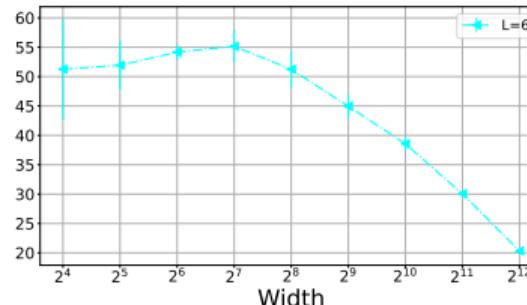
Figure: The *perturbation stability* with respect to depth of $L = 2$ to 10 in MLPs.



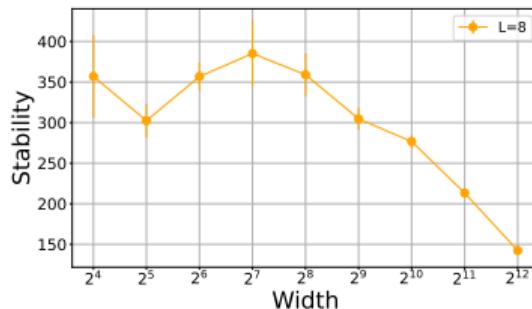
Lazy-training experiment on CNN



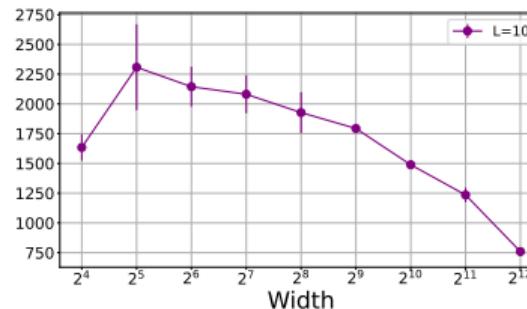
(a) $L = 4$



(b) $L = 6$



(c) $L = 8$



(d) $L = 10$

Figure: Relationship between the *perturbation stability* and width of CNN under He initialization for different depths of $L = 4, 6, 8$ and 10 .



Can we obtain guarantees for other architectures or defense mechanisms?

Interesting open questions

- ▶ *What is the trend (on robustness) with respect to **any** initialization and **any** hyper-parameter?*
- ▶ *Can we prove the effectiveness on robustness of **modern** defense techniques?*



Outline

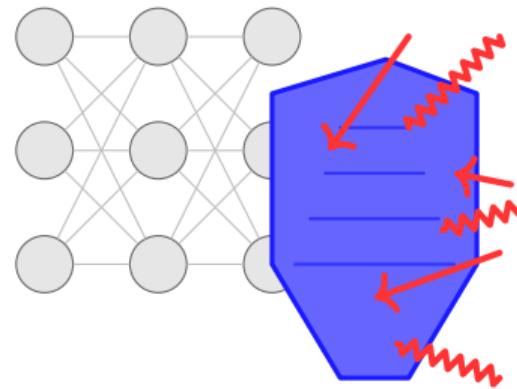
Adversarial robustness: Where it all started

Defense against adversarial attacks

Adversarial robustness in text

Adversarial certificates

Challenges and future work



Adversarial Training

Adversarial Training Objective

The objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in S} \mathcal{L}(f_{\theta}(x + \delta), y) \right],$$

where $S = \{\delta : \|\delta\|_{\infty} \leq \epsilon\}$ is the perturbation set.

- Train the model to be robust against worst-case adversarial perturbations within a specified threat model.



Adversarial Training

Adversarial Training Objective

The objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in S} \mathcal{L}(f_{\theta}(x + \delta), y) \right],$$

where $S = \{\delta : \|\delta\|_{\infty} \leq \epsilon\}$ is the perturbation set.

- Train the model to be robust against worst-case adversarial perturbations within a specified threat model.
 - ▶ **Inner Maximization (The Attacker):** $\max_{\delta \in S} \mathcal{L}(\theta, x + \delta, y)$
 - ▶ For the current parameters θ , find the perturbation δ in the allowed set S that **maximizes the loss**.
 - ▶ This can be approximated using attacks like **PGD** or solved exactly for simple models.



Adversarial Training

Adversarial Training Objective

The objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in S} \mathcal{L}(f_{\theta}(x + \delta), y) \right],$$

where $S = \{\delta : \|\delta\|_{\infty} \leq \epsilon\}$ is the perturbation set.

- Train the model to be robust against worst-case adversarial perturbations within a specified threat model.
 - ▶ **Inner Maximization (The Attacker):** $\max_{\delta \in S} \mathcal{L}(\theta, x + \delta, y)$
 - ▶ For the current parameters θ , find the perturbation δ in the allowed set S that **maximizes the loss**.
 - ▶ This can be approximated using attacks like **PGD** or solved exactly for simple models.
 - ▶ **Outer Minimization (The Defender):** $\min_{\theta} \mathbb{E}[\dots]$
 - ▶ Update the model parameters θ to **minimize the loss** on these worst-case adversarial examples found by the attacker.
 - ▶ Standard gradient descent on the adversarial loss.



Adversarial Training

Adversarial Training Objective

The objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in S} \mathcal{L}(f_{\theta}(x + \delta), y) \right],$$

where $S = \{\delta : \|\delta\|_{\infty} \leq \epsilon\}$ is the perturbation set.

- Train the model to be robust against worst-case adversarial perturbations within a specified threat model.
 - ▶ **Inner Maximization (The Attacker):** $\max_{\delta \in S} \mathcal{L}(\theta, x + \delta, y)$
 - ▶ For the current parameters θ , find the perturbation δ in the allowed set S that **maximizes the loss**.
 - ▶ This can be approximated using attacks like **PGD** or solved exactly for simple models.
 - ▶ **Outer Minimization (The Defender):** $\min_{\theta} \mathbb{E}[\dots]$
 - ▶ Update the model parameters θ to **minimize the loss** on these worst-case adversarial examples found by the attacker.
 - ▶ Standard gradient descent on the adversarial loss.
 - ▶ **On the literature:**
 - ▶ Originally proposed by Szegedy et al. (2014), formalized by Madry et al. (2018).
 - ▶ The work of Robey et al. (2023) demonstrates how this formulation can be tricky to implement.



Adversarial Training

Adversarial Training Objective

The objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in S} \mathcal{L}(f_{\theta}(x + \delta), y) \right],$$

where $S = \{\delta : \|\delta\|_{\infty} \leq \epsilon\}$ is the perturbation set.

- Train the model to be robust against worst-case adversarial perturbations within a specified threat model.
 - ▶ **Inner Maximization (The Attacker):** $\max_{\delta \in S} \mathcal{L}(\theta, x + \delta, y)$
 - ▶ For the current parameters θ , find the perturbation δ in the allowed set S that maximizes the loss.
 - ▶ This can be approximated using attacks like PGD or solved exactly for simple models.
 - ▶ **Outer Minimization (The Defender):** $\min_{\theta} \mathbb{E}[\dots]$
 - ▶ Update the model parameters θ to minimize the loss on these worst-case adversarial examples found by the attacker.
 - ▶ Standard gradient descent on the adversarial loss.
 - ▶ **On the literature:**
 - ▶ Originally proposed by Szegedy et al. (2014), formalized by Madry et al. (2018).
 - ▶ The work of Robey et al. (2023) demonstrates how this formulation can be tricky to implement.

Question

Can the minimization and maximization switch places?



Improved Adversarial Training

TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization (Zhang et al., 2019)

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\mathcal{L}_{CE}(\theta, x, y) + \beta \cdot \max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}_{KL}(f_\theta(x) \| f_\theta(x + \delta)) \right]. \quad (2)$$

- ▶ **Natural accuracy term:** $\mathcal{L}_{CE}(\theta, x, y)$ ensures good performance on clean examples.
- ▶ **Robustness term:** The attacker maximizes KL-divergence between output distributions of clean and adversarial examples, encouraging **smooth decision boundaries**.
- ▶ **Trade-off parameter:** β balances natural accuracy vs. adversarial robustness.



Robustness of the worst-performing class

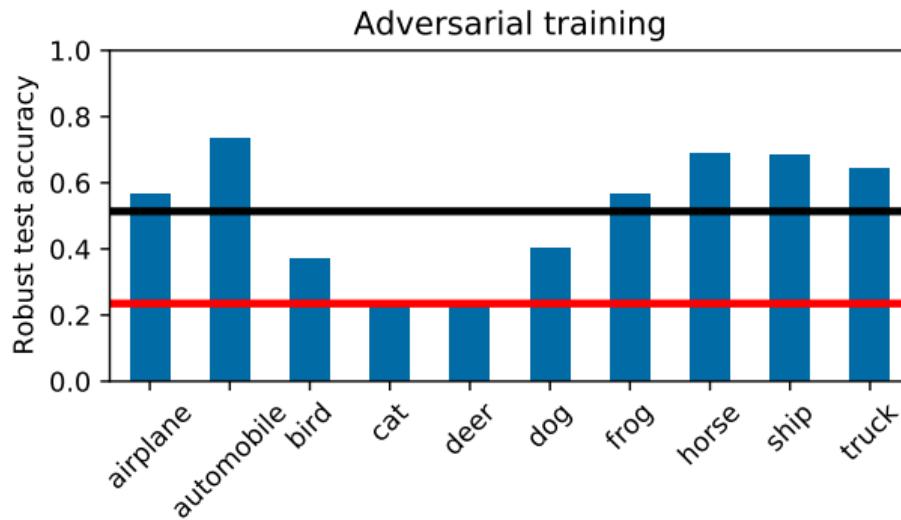


Figure: The worst class accuracy is much lower than the average class accuracy.



Robustness of the worst-performing class

Adversarial Training for worst-performing class

Let $f_{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}$ be a model with parameters θ and let $\{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathbb{R}^p$ and y_i be the corresponding labels. The adversarial training optimization for optimizing the worst-performing class is given by

$$\min_{\theta} \max_{y \in [k]} \left\{ \mathbb{E}_{x \sim p_{\mathcal{D}}(\cdot | y)} \left[\max_{\delta \in \mathcal{S}} \mathcal{L}(f_{\theta}(x + \delta), y) \right] \right\}, \quad (3)$$

where \mathcal{L} is the loss function.

T Pethick, G Chrysos, V Cevher. 'Revisiting adversarial training for the worst-performing class.' *Transactions on Machine Learning Research (TMLR)*, 2023.



Robustness of the worst-performing class

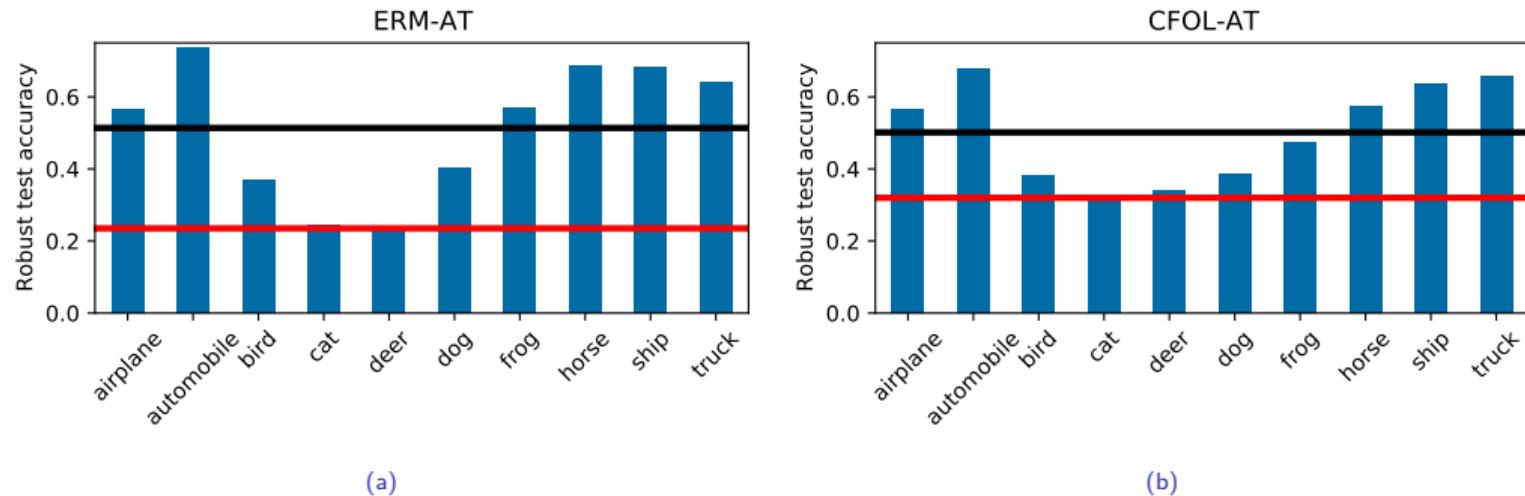


Figure: Robust test accuracy of (a) Empirical Risk Minimization and (b) the proposed method.

T Pethick, G Chrysos, V Cevher. 'Revisiting adversarial training for the worst-performing class.' *Transactions on Machine Learning Research (TMLR)*, 2023. Code: [↻](#)
<https://github.com/LIONS-EPFL/class-focused-online-learning-code>



Question

Can we search for a more robust architecture?



Searching for a more robust architecture

1. Utilize cell-based architecture search (Liu et al., 2019).

Y Wu, F Liu, CJ Simon-Gabriel, **G Chrysos**, V Cevher, 'Robust NAS benchmark under adversarial training: benchmark, theory, and beyond'. In *ICLR*, 2024.



Searching for a more robust architecture

1. Utilize cell-based architecture search (Liu et al., 2019).
2. Train the architectures with adversarial training.

Y Wu, F Liu, CJ Simon-Gabriel, **G Chrysos**, V Cevher, 'Robust NAS benchmark under adversarial training: benchmark, theory, and beyond'. In *ICLR*, 2024.



Searching for a more robust architecture

1. Utilize cell-based architecture search (Liu et al., 2019).
2. Train the architectures with adversarial training.
3. Run for over 100,000 GPU hours (!)

Y Wu, F Liu, CJ Simon-Gabriel, **G Chrysos**, V Cevher, 'Robust NAS benchmark under adversarial training: benchmark, theory, and beyond'. In *ICLR*, 2024.



Searching for a more robust architecture

1. Utilize cell-based architecture search (Liu et al., 2019).
2. Train the architectures with adversarial training.
3. Run for over 100,000 GPU hours (!)
4. Insight 1: Significant variance in the performance across models.

Y Wu, F Liu, CJ Simon-Gabriel, **G Chrysos**, V Cevher, 'Robust NAS benchmark under adversarial training: benchmark, theory, and beyond'. In *ICLR*, 2024.



Searching for a more robust architecture

1. Utilize cell-based architecture search (Liu et al., 2019).
2. Train the architectures with adversarial training.
3. Run for over 100,000 GPU hours (!)
4. Insight 1: Significant variance in the performance across models.
5. Insight 2: Correlation between high-performing architectures and highly robust architectures.

Y Wu, F Liu, CJ Simon-Gabriel, **G Chrysos**, V Cevher, 'Robust NAS benchmark under adversarial training: benchmark, theory, and beyond'. In *ICLR*, 2024.



Outline

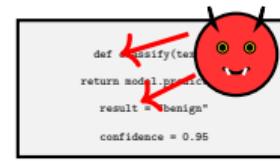
Adversarial robustness: Where it all started

Defense against adversarial attacks

Adversarial robustness in text

Adversarial certificates

Challenges and future work



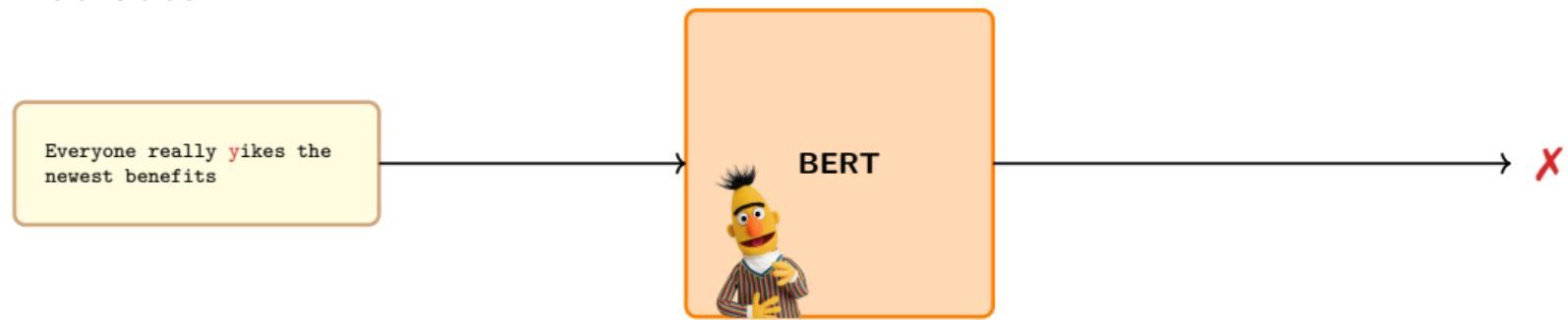
Robustness in Natural Language Processing (NLP)

- ▶ **Adversarial Attacks (AA)**
 - ▶ **Character-level:** HotFlip (Ebrahimi et al., ACL 2018), DeepWordBug (Gao et al., SPW/ArXiv 2018), TextBugger (Li et al., NDSS 2019), ...
 - ▶ **Token / word-level:** GenAdv (Alzantot et al., EMNLP 2018), PWWS (Ren et al., ACL 2019), BERT-ATTACK (Li et al., EMNLP 2020), TextFooler (Jin et al., AAAI 2020), BAE (Garg et al., EMNLP 2020), PSO (Zang et al., ACL 2020), ...
 - ▶ **Sentence / paraphrase:** AdvSQuAD (Jia & Liang, EMNLP 2017), SCPN (Iyyer et al., NAACL 2018), Universal Adversarial Triggers (Wallace et al., EMNLP 2019), AdversarialQA (Bartolo et al., 2020), ...
- ▶ **Transferability of AA:** Yuan et al. (EMNLP 2021), topic CVaR (Oren et al., EMNLP 2019), ...
- ▶ **Adversarial training & frameworks:** Miyato et al. (ICLR 2017), FreeLB (Zhu et al., ICLR 2020), TextAttack framework (Morris et al., EMNLP 2020), ...
- ▶ **Certifiable robustness:** Certified Robustness to Adversarial Word Substitutions (Jia et al., EMNLP 2019), SAFER (Ye et al., 2020), ...
- ▶ **Benchmarks and datasets:** AdvSQuAD (Jia & Liang, EMNLP 2017), WinoBias (Zhao et al., NAACL 2018), ANLI (Nie et al., ACL 2020), AdversarialQA (Bartolo et al., 2020), ...
- ▶ **Additional resources:** EMNLP 21 Tutorial on Robust NLP (<https://robustnlp-tutorialgithub.io/>), Mozes et al, 2021 (EMNLP), <https://github.com/thunlp/TAADpapers>, ...



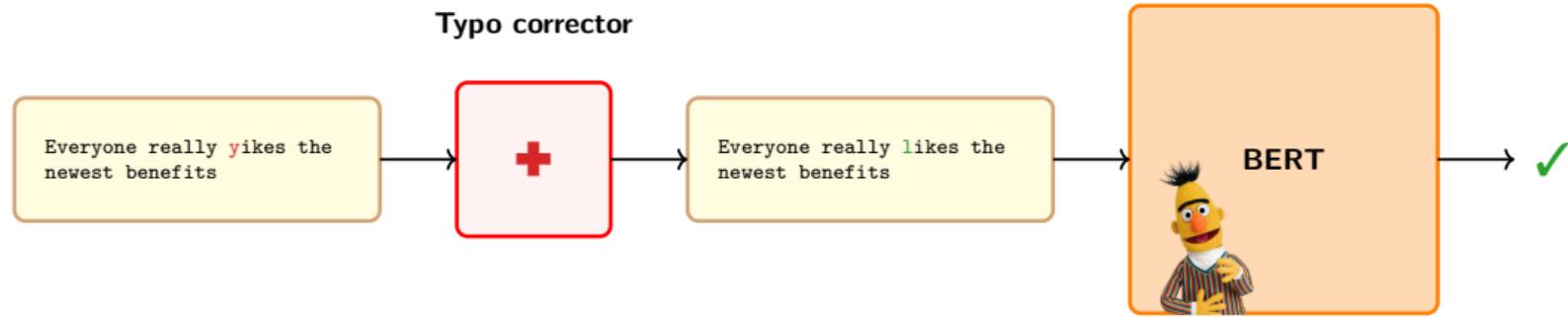
Character-level attacks can be corrected [Jones, Jia, Raghunathan, Liang, ACL (2020)]

- Is this true?



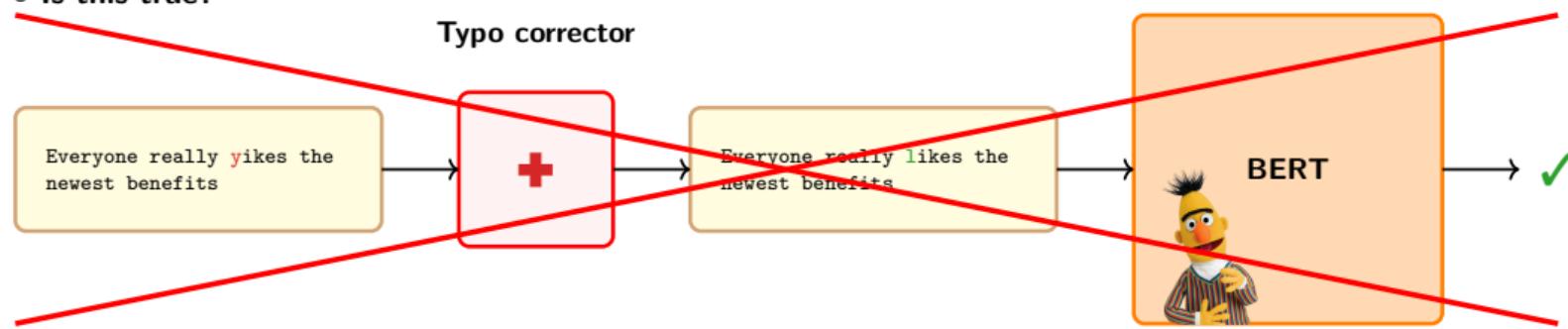
Character-level attacks can be corrected [Jones, Jia, Raghunathan, Liang, ACL (2020)]

- Is this true?



Character-level attacks can be corrected [Jones, Jia, Raghunathan, Liang, ACL (2020)]

- Is this true?

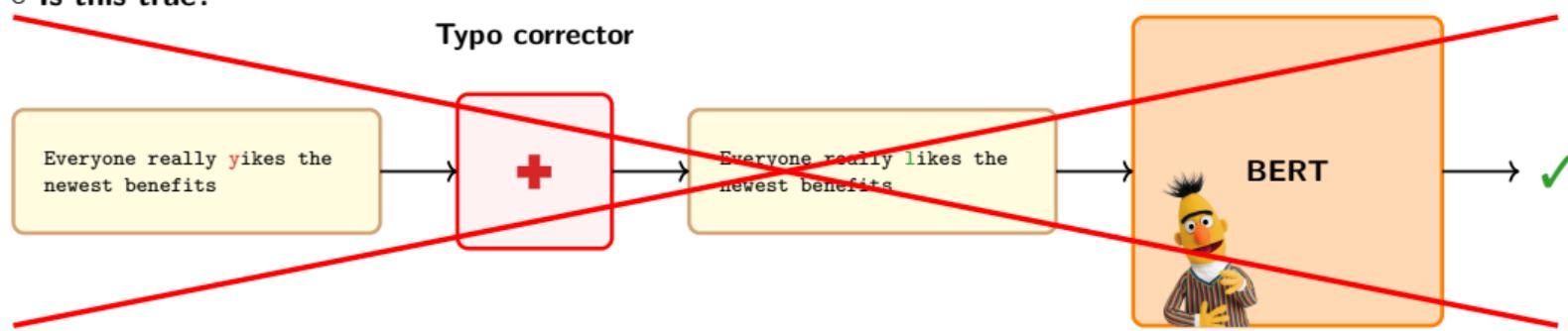


- ▷ Typo correctors might fail!



Character-level attacks can be corrected [Jones, Jia, Raghunathan, Liang, ACL (2020)]

- Is this true?

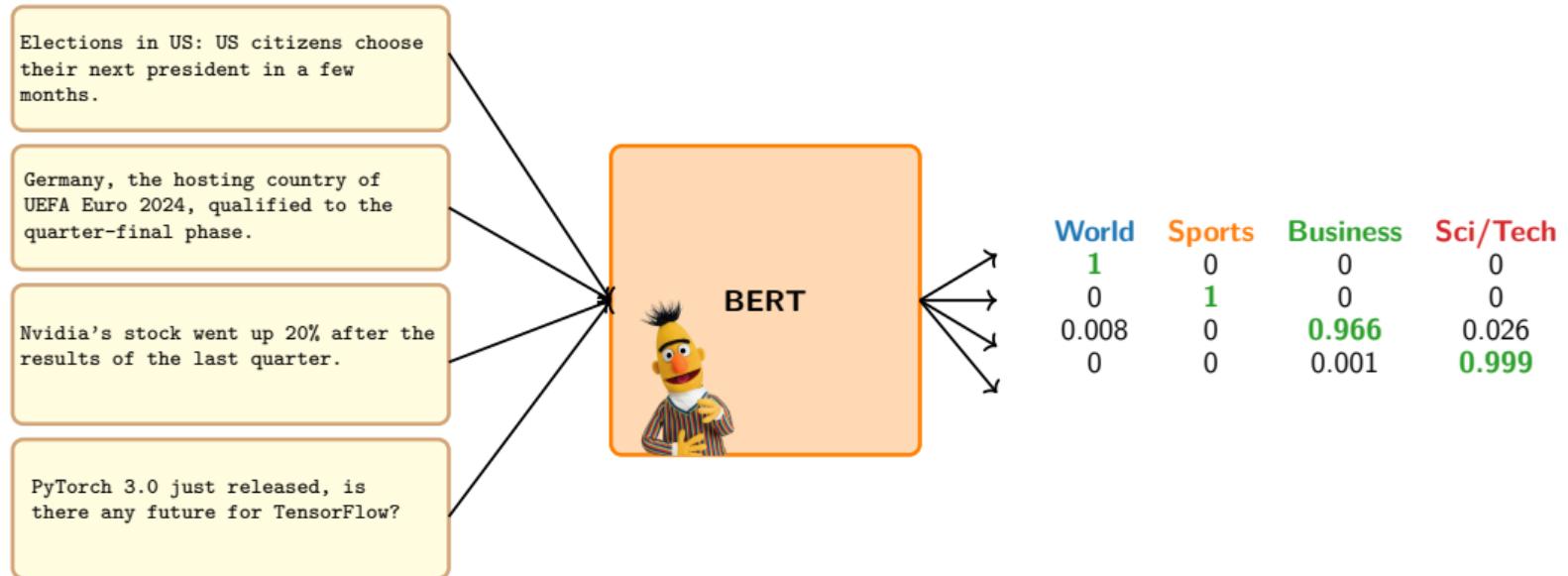


- ▷ Typo correctors might fail!
- ▷ Charmer: How can we obtain efficient and effective character-level attacks then?



Natural Language Processing (NLP) tasks

- Classification:

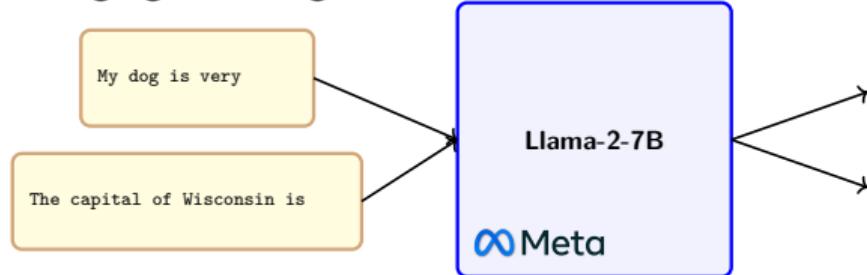


Devlin, et al 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.' In NAACL, 2019.



Natural Language Processing (NLP) tasks

- **Language Modeling:**



anxious	ag	sk	friendly	protect	...
0.092	0.077	0.054	0.045	0.044	...
Madison	the	and	Alabama	New	...
0.489	0.459	0.028	0.001	0.001	...



Discrete Sequential Inputs? Just text!

- **Defining text:**

- ▷ Γ

- Alphabet, e.g., keyboard characters or tokens.



Discrete Sequential Inputs? Just text!

- **Defining text:**

- ▷ Γ

- Alphabet, e.g., keyboard characters or tokens.

- $\mathcal{S}(\Gamma) = \{c_1 c_2 \cdots c_m : c_i \in \Gamma, \forall m \in \mathbb{N} \setminus 0\}$

- Sentences.



Discrete Sequential Inputs? Just text!

- **Defining text:**

- ▷ Γ

Alphabet, e.g., keyboard characters or tokens.

- ▷ $S(\Gamma) = \{c_1 c_2 \cdots c_m : c_i \in \Gamma, \forall m \in \mathbb{N} \setminus 0\}$

Sentences.

- ▷ $S \in \{0, 1\}^{m \times |\Gamma|} : \|s_i\|_1 = 1, \forall i = 1, \dots, m$

One-hot representation.



Discrete Sequential Inputs? Just text!

- **Defining text:**

- ▷ Γ Alphabet, e.g., keyboard characters or tokens.
- ▷ $\mathcal{S}(\Gamma) = \{c_1 c_2 \cdots c_m : c_i \in \Gamma, \forall m \in \mathbb{N} \setminus 0\}$ Sentences.
- ▷ $S \in \{0, 1\}^{m \times |\Gamma|} : \|s_i\|_1 = 1, \forall i = 1, \dots, m$ One-hot representation.

- **Learning tasks:**

- ▷ Classification: $h_{\theta} : \mathcal{S}(\Gamma) \rightarrow \mathbb{R}^o$ so that $\hat{y} = \arg \max_{j \in \{1, \dots, o\}} f_{\theta}(S)_j$
- ▷ Language Modeling: $f_{\theta} : \mathcal{S}(\Gamma) \rightarrow [0, 1]^{|\Gamma|}$ so that $\|f_{\theta}(S)\|_1 = 1$



Adversarial Robustness in NLP

Adversarial Robustness in NLP

Given a classification model f_{θ} , a sentence-label pair (S, y) and a subset $\mathcal{R}(S) \subset \mathcal{S}(\Gamma)$ where we would like to be robust, we would like to have:

$$f_{\theta}(S')_y - \max_{\hat{y} \neq y} f_{\theta}(S')_{\hat{y}} > 0, \quad \forall S' \in \mathcal{R}(S). \quad \bullet$$



Adversarial Robustness in NLP

Adversarial Robustness in NLP

Given a classification model f_θ , a sentence-label pair (S, y) and a subset $\mathcal{R}(S) \subset \mathcal{S}(\Gamma)$ where we would like to be robust, we would like to have:

$$f_\theta(S')_y - \max_{\hat{y} \neq y} f_\theta(S')_{\hat{y}} > 0, \quad \forall S' \in \mathcal{R}(S). \quad \bullet$$

- Two critical problems:

- ▶ **Adversarial Attacks:** How can we find counter examples for \bullet ?
- ▶ **Verification:** How can we efficiently prove that \bullet holds?



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S'') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Hallo}) =$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Hallo}) = 1$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Helo}) =$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Helo}) = 1$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Helloo}) =$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S'') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Helloo}) = 1$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Haloo}) =$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$d_{\text{lev}}(\text{Hello}, \text{Haloo}) = 2$$



Levenshtein distance

Levenshtein distance [Levenshtein (1966)]

For any two sentences $S, S' \in \mathcal{S}(\Gamma)$, the Levenshtein distance is defined as:

$$d_{\text{lev}}(S, S') := \begin{cases} |S| , & \text{if } |S'| = 0 \\ |S'| , & \text{if } |S| = 0 \\ d_{\text{lev}}(S_{2:}, S'_{2:}) , & \text{if } S_1 = S'_1 \\ 1 + \min \left\{ \begin{array}{l} d_{\text{lev}}(S_{2:}, S'_{2:}) \\ d_{\text{lev}}(S_{2:}, S') \\ d_{\text{lev}}(S, S'_{2:}) \end{array} \right\} , & \text{otherwise .} \end{cases}$$

Examples

Levenshtein distance d_{lev} from $S = \text{"Hello"}$ to several modifications.

$$\begin{aligned} d_{\text{lev}}(\text{Hello}, \text{Helo}) &= 1 & d_{\text{lev}}(\text{Hello}, \text{Hallo}) &= 1 \\ d_{\text{lev}}(\text{Hello}, \text{Helloo}) &= 1 & d_{\text{lev}}(\text{Hello}, \text{Haloo}) &= 2 . \end{aligned}$$



Adversarial Attacks in NLP: Two dominant approaches

- Two "Schools":
- **Character-level attacks:** "Small Levenshtein distance with S "

$$\mathcal{R}(S) = \{S' \in \mathcal{S}(\Gamma) : d_{\text{lev}}(S, S') \leq k\}$$

Gao, et al 'Black-box generation of adversarial text sequences to evade deep learning classifiers.' In *IEEE SPW*, 2018.



Adversarial Attacks in NLP: Two dominant approaches

- Two "Schools":
- **Character-level attacks:** "Small Levenshtein distance with S "

$$\mathcal{R}(S) = \{S' \in \mathcal{S}(\Gamma) : d_{\text{lev}}(S, S') \leq k\}$$

Gao, et al 'Black-box generation of adversarial text sequences to evade deep learning classifiers.' In *IEEE SPW*, 2018.

- **Token-level attacks:** "Words can be changed by synonyms"

$$\mathcal{R}("My\ dog\ is\ good") = \left\{ \begin{array}{l} "My\ puppy\ is\ good" \\ "My\ dog\ is\ nice" \\ "My\ puppy\ is\ nice" \\ \vdots \end{array} \right\}$$

▷ Synonym = high cosine similarity between embeddings.

Jin, et al 'Is BERT really robust? a strong baseline for natural language attack on text classification and entailment.' In *AAAI*, 2020.



Adversarial Attacks in NLP: Problems

- Adversarial attacks should be *efficient* and *imperceptible*
- Character-level:
 - ▷ Large search space and non-differentiable tokenizer ✗
 - ▷ Easily avoided with typo-correctors ✗

Pruthi, et al 'Combating adversarial misspellings with robust word recognition.' In **ACL**, 2019.

Jones, et al 'Robust encodings: A framework for combating adversarial typos.' In **ACL**, 2020.

- Token-level:
 - ▷ Efficient with gradient based approaches ✓
 - ▷ Often change the meaning ✗

Morris, et al 'Reevaluating adversarial examples in natural language.' In **EMNLP**, 2020.



Character-level attacks: Can we do better?

- First sample from the MNLI-m dataset. Token-level (●) and character-level (●) attacks.

Original	Everyone really likes the newest benefits
TextGrad ●	Everyone really hates the newest benefits
TextFooler ●	Nobody really likes the newest benefits
DeepWordBug ●	Everyone really yikes the newest benefits



Character-level attacks: Can we do better?

- First sample from the MNLI-m dataset. Token-level (●) and character-level (●) attacks.

Original	Everyone really likes the newest benefits
TextGrad ●	Everyone really hates the newest benefits
TextFooler ●	Nobody really likes the newest benefits
DeepWordBug ●	Everyone really yikes the newest benefits

- Are typo-correctors really the solution?
- Can we be more efficient?

Abad Rocamora, et al 'Revisiting Character-level Adversarial Attacks for Language Models.' In ICML, 2024.



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original

They don't care about us



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original

They don't care about us

Expand

_T_h_e_y_ _d_o_n_’_t_ _c_a_r_e_ _a_b_o_u_t_ _u_s_



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original

They don't care about us

Expand

_T_h_e_y_ _d_o_n_’_t_ _c_a_r_e_ _a_b_o_u_t_ _u_s_

Replace

_T_h_e_y_ _d_o_n_’_t_ _c_aWr_e_ _a_b_o_u_t_ _u_s_



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original

They don't care about us

Expand

_T_h_e_y_ _d_o_n_’_t_ _c_a_r_e_ _a_b_o_u_t_ _u_s_

Replace

_T_h_e_y_ _d_o_n_’_t_ _c_aWr_e_ _a_b_o_u_t_ _u_s_

Contract

They don't caWre about us



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original

They don't care about us

Expand

_T_h_e_y_ _d_o_n_’_t_ _c_a_r_e_ _a_b_o_u_t_ _u_s_

Replace

_T_h_e_y_ _d_o_n_’_t_ _c_aWr_e_ _a_b_o_u_t_ _u_s_

Contract

They don't caWre about us

- At $k = 1$, we have $|\mathcal{R}(S)| \leq (2 \cdot |S| + 1) \cdot |\Gamma|$



Characterizing single-character perturbations

- Any single-character modification can be represented as:

Original	They don't care about us
Expand	_T_h_e_y_ _d_o_n_’_t_ _c_a_r_e_ _a_b_o_u_t_ _u_s_
Replace	_T_h_e_y_ _d_o_n_’_t_ _c_aWr_e_ _a_b_o_u_t_ _u_s_
Contract	They don't caWre about us

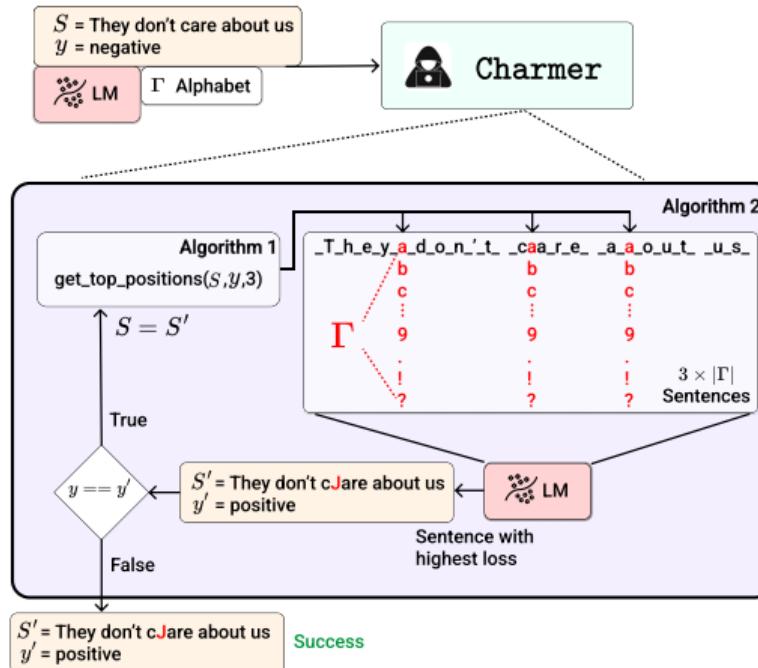
- At $k = 1$, we have $|\mathcal{R}(S)| \leq (2 \cdot |S| + 1) \cdot |\Gamma|$

- Key idea in Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]:**

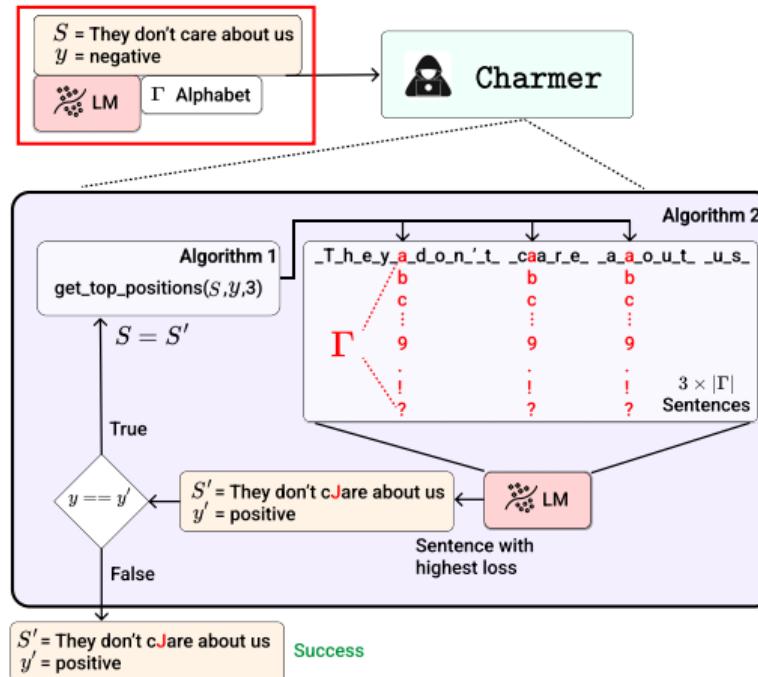
- ▷ Greedy approach: One character at a time.
- ▷ Evaluate the loss replacing a “test character” in the $2|S| + 1$ positions.
- ▷ Get top- n positions and try $n \cdot |\Gamma|$ sentences.



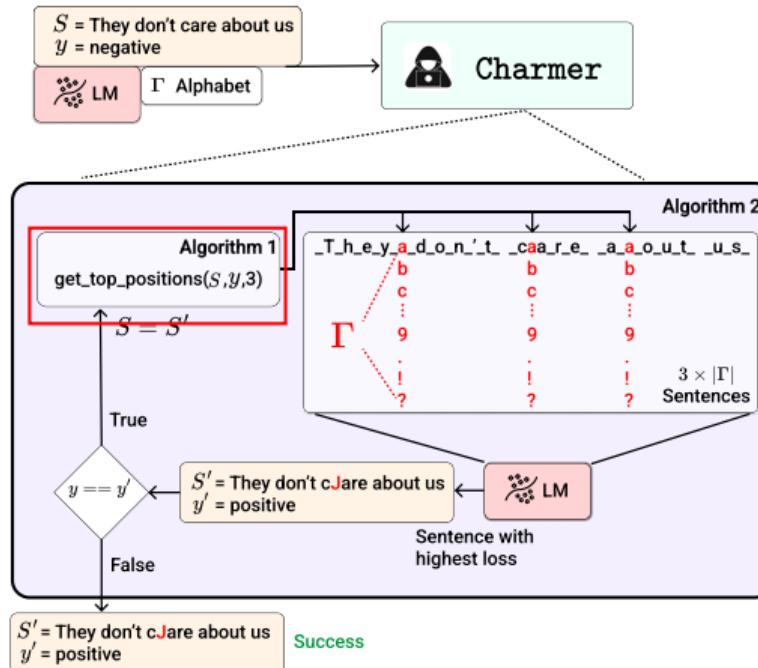
Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]



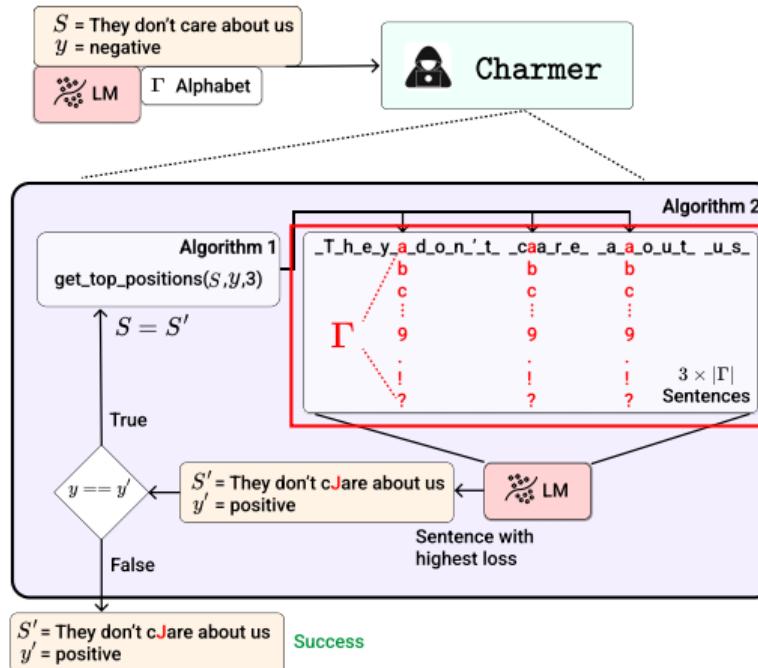
Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]



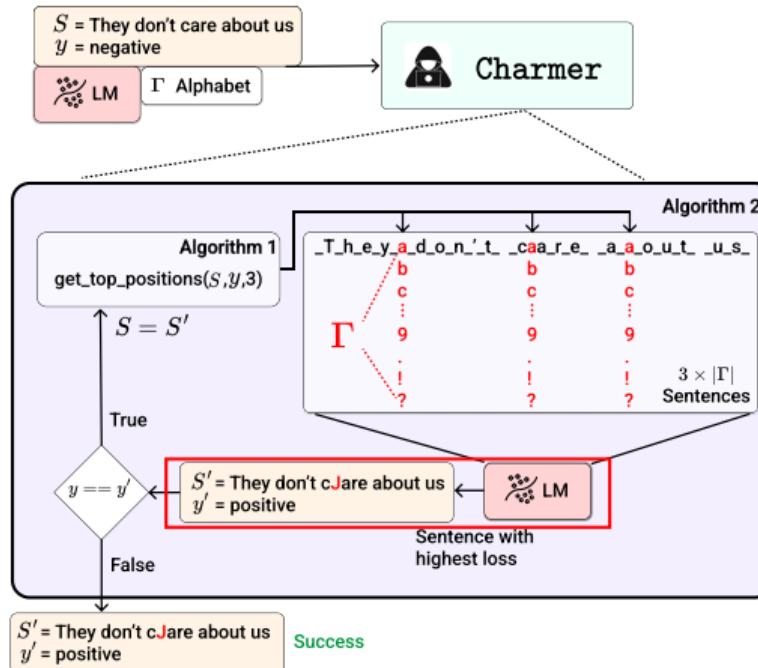
Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]



Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]



Charmer [Abad Rocamora, Wu, Liu, Chrysos, Cevher, ICML (2024)]



TextAttack Benchmark: Higher ASR and Similarity

Table: Attack Success Rate (ASR) and similarity comparison. Token-level (●) and character-level (●) attacks.

Method	BERT				RoBERTa			
	ASR (%) ↑	$d_{lev}(S, S') \downarrow$	$Sim(S, S') \uparrow$	Time (s) ↓	ASR (%) ↑	$d_{lev}(S, S') \downarrow$	$Sim(S, S') \uparrow$	Time (s) ↓
AG-News	GBDA ●	42.09	$17.76 \pm (9.33)$	$0.93 \pm (0.05)$	$13.86 \pm (3.14)$	-	-	-
	BAE-R ●	17.09	$15.07 \pm (10.59)$	$0.97 \pm (0.02)$	$1.61 \pm (1.36)$	18.27	$15.29 \pm (10.34)$	$0.97 \pm (0.02)$
	BERT-attack ●	29.90	$20.66 \pm (16.91)$	$0.93 \pm (0.05)$	$5.58 \pm (12.92)$	27.55	$16.96 \pm (12.95)$	$0.94 \pm (0.04)$
	DeepWordBug ●	60.51	$11.75 \pm (8.00)$	$0.78 \pm (0.18)$	$0.81 \pm (0.52)$	56.81	$11.81 \pm (7.69)$	$0.79 \pm (0.16)$
	TextBugger ●	50.85	$19.79 \pm (17.93)$	$0.90 \pm (0.06)$	$1.53 \pm (1.13)$	51.21	$21.42 \pm (19.28)$	$0.90 \pm (0.06)$
	TextFooler ●	78.98	$53.18 \pm (39.30)$	$0.84 \pm (0.11)$	$3.75 \pm (2.76)$	84.48	$52.45 \pm (36.97)$	$0.84 \pm (0.11)$
	TextGrad ●	85.85	$55.38 \pm (30.33)$	$0.77 \pm (0.11)$	$7.98 \pm (9.24)$	78.75	$31.94 \pm (15.57)$	$0.86 \pm (0.07)$
	CWBA ●	86.72	$15.71 \pm (7.17)$	$0.65 \pm (0.19)$	$174.15 \pm (130.91)$	81.39	$13.73 \pm (11.24)$	$0.86 \pm (0.11)$
	Pruthi et al. (2019) ●	90.02	$6.25 \pm (4.69)$	$0.86 \pm (0.14)$	$49.47 \pm (48.26)$	88.91	$6.55 \pm (5.13)$	$0.86 \pm (0.14)$
	Charmer-Fast ●	<u>95.86</u>	<u>$4.85 \pm (3.96)$</u>	$0.92 \pm (0.08)$	$3.12 \pm (3.88)$	<u>91.87</u>	<u>$4.87 \pm (4.07)$</u>	$0.91 \pm (0.09)$
	Charmer ●	98.51	$3.68 \pm (3.08)$	$0.95 \pm (0.06)$	$8.74 \pm (11.10)$	96.88	$3.73 \pm (3.07)$	$0.95 \pm (0.05)$
SST-2	GBDA ●	83.37	$12.20 \pm (6.94)$	$0.85 \pm (0.11)$	$9.32 \pm (1.78)$	-	-	-
	BAE-R ●	66.38	$10.10 \pm (7.00)$	$0.83 \pm (0.18)$	$1.24 \pm (0.86)$	63.16	$10.22 \pm (6.33)$	$0.85 \pm (0.16)$
	BERT-attack ●	69.57	$12.19 \pm (9.55)$	$0.87 \pm (0.09)$	$239.80 \pm (1763.30)$	64.21	$11.26 \pm (7.18)$	$0.86 \pm (0.10)$
	DeepWordBug ●	81.39	$3.74 \pm (2.95)$	$0.80 \pm (0.17)$	$0.22 \pm (0.12)$	84.27	$4.61 \pm (3.47)$	$0.75 \pm (0.20)$
	TextBugger ●	68.49	$5.97 \pm (5.87)$	$0.91 \pm (0.06)$	$1.75 \pm (0.91)$	61.10	$6.85 \pm (6.54)$	$0.90 \pm (0.05)$
	TextFooler ●	<u>95.16</u>	$17.17 \pm (12.51)$	$0.82 \pm (0.15)$	$0.90 \pm (0.57)$	95.00	$17.76 \pm (12.45)$	$0.82 \pm (0.15)$
	TextGrad ●	94.04	$21.61 \pm (11.30)$	$0.75 \pm (0.13)$	$19.94 \pm (22.32)$	95.49	$17.07 \pm (9.57)$	$0.81 \pm (0.10)$
	CWBA ●	72.92	$8.55 \pm (3.78)$	$0.53 \pm (0.26)$	$33.81 \pm (33.86)$	49.84	$8.88 \pm (3.94)$	$0.65 \pm (0.17)$
	Pruthi et al. (2019) ●	90.94	$2.22 \pm (1.35)$	$0.85 \pm (0.14)$	$4.86 \pm (4.02)$	92.93	$2.52 \pm (1.57)$	$0.84 \pm (0.14)$
	Charmer-Fast ●	100.00	<u>$1.74 \pm (1.02)$</u>	$0.89 \pm (0.13)$	<u>$0.34 \pm (0.31)$</u>	<u>99.39</u>	<u>$2.29 \pm (1.53)$</u>	$0.84 \pm (0.15)$
	Charmer ●	100.00	$1.47 \pm (0.74)$	$0.90 \pm (0.11)$	$1.27 \pm (0.84)$	99.51	$1.76 \pm (1.12)$	$0.89 \pm (0.12)$



TextAttack Benchmark: Higher ASR and Similarity

Table: Attack Success Rate (ASR) and similarity comparison. Token-level (●) and character-level (●) attacks.

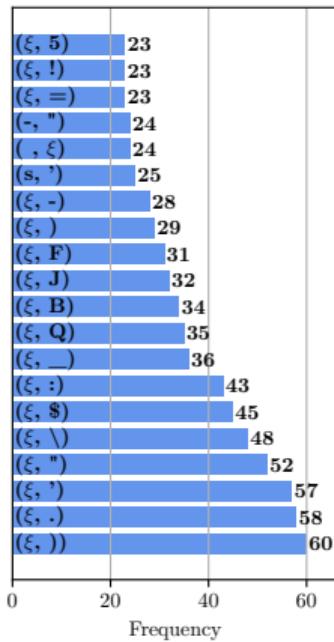
Method	BERT				RoBERTa			
	ASR (%) ↑	$d_{lev}(S, S') \downarrow$	$\text{Sim}(S, S') \uparrow$	Time (s) ↓	ASR (%) ↑	$d_{lev}(S, S') \downarrow$	$\text{Sim}(S, S') \uparrow$	Time (s) ↓
AG-News	GBDA ●	42.09	$17.76 \pm (9.33)$	$0.93 \pm (0.05)$	$13.86 \pm (3.14)$	-	-	-
	BAE-R ●	17.09	$15.07 \pm (10.59)$	$0.97 \pm (0.02)$	$1.61 \pm (1.36)$	18.27	$15.29 \pm (10.34)$	$0.97 \pm (0.02)$
	BERT-attack ●	29.90	$20.66 \pm (16.91)$	$0.93 \pm (0.05)$	$5.58 \pm (12.92)$	27.55	$16.96 \pm (12.95)$	$0.94 \pm (0.04)$
	DeepWordBug ●	60.51	$11.75 \pm (8.00)$	$0.78 \pm (0.18)$	$0.81 \pm (0.52)$	56.81	$11.81 \pm (7.69)$	$0.79 \pm (0.16)$
	TextBugger ●	50.85	$19.79 \pm (17.93)$	$0.90 \pm (0.06)$	$1.53 \pm (1.13)$	51.21	$21.42 \pm (19.28)$	$0.90 \pm (0.06)$
	TextFooler ●	78.98	$53.18 \pm (39.30)$	$0.84 \pm (0.11)$	$3.75 \pm (2.76)$	84.48	$52.45 \pm (36.97)$	$0.84 \pm (0.11)$
	TextGrad ●	85.85	$55.38 \pm (30.33)$	$0.77 \pm (0.11)$	$7.98 \pm (9.24)$	78.75	$31.94 \pm (15.57)$	$0.86 \pm (0.07)$
	CWBA ●	86.72	$15.71 \pm (7.17)$	$0.65 \pm (0.19)$	$174.15 \pm (130.91)$	81.39	$13.73 \pm (11.24)$	$0.86 \pm (0.11)$
	Pruthi et al. (2019) ●	90.02	$6.25 \pm (4.69)$	$0.86 \pm (0.14)$	$49.47 \pm (48.26)$	88.91	$6.55 \pm (5.13)$	$0.86 \pm (0.14)$
	Charmer-Fast ●	95.86	$4.85 \pm (3.96)$	$0.92 \pm (0.08)$	$3.12 \pm (3.88)$	91.87	$4.87 \pm (4.07)$	$0.91 \pm (0.09)$
	Charmer ●	98.51	$3.68 \pm (3.08)$	$0.95 \pm (0.06)$	$8.74 \pm (11.10)$	96.88	$3.73 \pm (3.07)$	$0.95 \pm (0.05)$
SST-2	GBDA ●	83.37	$12.20 \pm (6.94)$	$0.85 \pm (0.11)$	$9.32 \pm (1.78)$	-	-	-
	BAE-R ●	66.38	$10.10 \pm (7.00)$	$0.83 \pm (0.18)$	$1.24 \pm (0.86)$	63.16	$10.22 \pm (6.33)$	$0.85 \pm (0.16)$
	BERT-attack ●	69.57	$12.19 \pm (9.55)$	$0.87 \pm (0.09)$	$239.80 \pm (1763.30)$	64.21	$11.26 \pm (7.18)$	$0.86 \pm (0.10)$
	DeepWordBug ●	81.39	$3.74 \pm (2.95)$	$0.80 \pm (0.17)$	$0.22 \pm (0.12)$	84.27	$4.61 \pm (3.47)$	$0.75 \pm (0.20)$
	TextBugger ●	68.49	$5.97 \pm (5.87)$	$0.91 \pm (0.06)$	$1.75 \pm (0.91)$	61.10	$6.85 \pm (6.54)$	$0.90 \pm (0.05)$
	TextFooler ●	<u>95.16</u>	$17.17 \pm (12.51)$	$0.82 \pm (0.15)$	$0.90 \pm (0.57)$	95.00	$17.76 \pm (12.45)$	$0.82 \pm (0.15)$
	TextGrad ●	94.04	$21.61 \pm (11.30)$	$0.75 \pm (0.13)$	$19.94 \pm (22.32)$	95.49	$17.07 \pm (9.57)$	$0.81 \pm (0.10)$
	CWBA ●	72.92	$8.55 \pm (3.78)$	$0.53 \pm (0.26)$	$33.81 \pm (33.86)$	49.84	$8.88 \pm (3.94)$	$0.65 \pm (0.17)$
	Pruthi et al. (2019) ●	90.94	$2.22 \pm (1.35)$	$0.85 \pm (0.14)$	$4.86 \pm (4.02)$	92.93	$2.52 \pm (1.57)$	$0.84 \pm (0.14)$
	Charmer-Fast ●	100.00	<u>$1.74 \pm (1.02)$</u>	$0.89 \pm (0.13)$	<u>$0.34 \pm (0.31)$</u>	99.39	<u>$2.29 \pm (1.53)$</u>	$0.84 \pm (0.15)$
	Charmer ●	100.00	$1.47 \pm (0.74)$	<u>$0.90 \pm (0.11)$</u>	$1.27 \pm (0.84)$	99.51	$1.76 \pm (1.12)$	<u>$0.89 \pm (0.12)$</u>



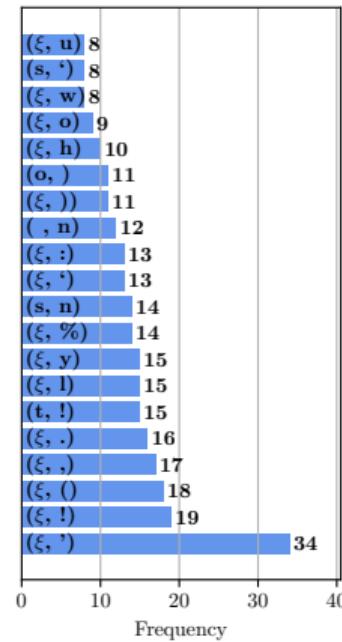
What are the most common changes?

Figure: Most common replacements: The pair of characters (c_1, c_2) indicates that c_1 is replaced by c_2 in the sentence.

AG-News



SST-2



Code: <https://github.com/LIONS-EPFL/charmer>



What about typo-correctors?

- The PJC constraints:

- ▷ Repeat: Not perturb the same word twice.
- ▷ First: Not perturb the first character of a word.
- ▷ Last: Not perturb the last character of a word.
- ▷ Length: Not perturb words with less than 4 chars.
- ▷ LowEng: Only replace by lowercase characters in the English alphabet.

Defense	Acc. (%)	PJC?	ASR (%)	$d_{\text{lev}}(S, S')$	$\text{Sim}(S, S')$
None	92.43	✗	100.00	$1.47 \pm (0.74)$	$0.90 \pm (0.11)$
		✓	96.65	$1.86 \pm (1.14)$	$0.87 \pm (0.14)$
Pruthi et al. (2019)	88.53	✗	100.00	$1.28 \pm (0.51)$	$0.90 \pm (0.11)$
		✓	70.34	$2.08 \pm (1.49)$	$0.85 \pm (0.14)$
Jones et al. (2020)	83.94	✗	100.00	$1.43 \pm (0.71)$	$0.88 \pm (0.11)$
		✓	0.96	$1.14 \pm (0.38)$	$0.92 \pm (0.06)$



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive film- maker.		

- Errors:
 - ▷ charming ⇒ ch█rming ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive film- maker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td em- bark a major career as a commercial ye't inven- tive filmmaker.	

- Errors:

▷ charming ⇒ ch█rming ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td em- bark a major career as a commercial ye't inven- tive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad em- braces a major career as a commercial n't inven- tive filmmaker.

- Errors:

▷ charming ⇒ ch█rming ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td em- bark a major career as a commercial ye't inven- tive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad em- braces a major career as a commercial n't inven- tive filmmaker.

- Errors:
 - ▷ charming ⇒ ch█rming ⇒ cold running



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td em- bark a major career as a commercial ye't inven- tive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad em- braces a major career as a commercial n't inven- tive filmmaker.

- Errors:
 - charming ⇒ ch█rming ⇒ cold running
 - desperate ⇒ des█perate ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td em- bark a major career as a commercial ye't inven- tive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad em- braces a major career as a commercial n't inven- tive filmmaker.

- Errors:
 - charming ⇒ ch█rming ⇒ cold running
 - desperate ⇒ des█perate ⇒ does pleasure



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td embark a major career as a commercial ye't inventive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad embraces a major career as a commercial n't inventive filmmaker.

- Errors:
 - charming ⇒ ch█rming ⇒ cold running
 - desperate ⇒ des█perate ⇒ does pleasure
 - to ⇒ td ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td embark a major career as a commercial ye't inventive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad embraces a major career as a commercial n't inventive filmmaker.

- Errors:
 - charming ⇒ ch█rming ⇒ cold running
 - desperate ⇒ des█perate ⇒ does pleasure
 - to ⇒ td ⇒ bad



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td embark a major career as a commercial ye't inventive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad embraces a major career as a commercial n't inventive filmmaker.

- Errors:

- ▷ charming ⇒ ch█rming ⇒ cold running
- ▷ desperate ⇒ des█perate ⇒ does pleasure
- ▷ to ⇒ td ⇒ bad
- ▷ yet ⇒ ye't ⇒



Typo-correctors might play against you

- SST-2 attacked and corrected sentences for BERT.

Original	Attacked	(Pruthi et al, 2019) Recognition
it 's a charming and often affecting journey . unflinchingly bleak and desperate allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.	it 's a ch█rming and often affecting journey . unflinchingly bleak and des█perate allows us to hope that nolan is poised td embark a major career as a commercial ye't inventive filmmaker.	it 's a cold running and often affecting journey . unflinching bleak and does pleasure allows us to hope that nolan is polished bad embraces a major career as a commercial n't inventive filmmaker.

- Errors:

- ▷ charming ⇒ ch█rming ⇒ cold running
- ▷ desperate ⇒ des█perate ⇒ does pleasure
- ▷ to ⇒ td ⇒ bad
- ▷ yet ⇒ ye't ⇒ n't



Conclusions and future work

- ▶ Language models are prone to character-level attacks. **Few** characters are sufficient to poison them.



Conclusions and future work

- ▶ Language models are prone to character-level attacks. **Few** characters are sufficient to poison them.
- ▶ Typo-correctors do not solve the problem.



Conclusions and future work

- ▶ Language models are prone to character-level attacks. **Few** characters are sufficient to poison them.
- ▶ Typo-correctors do not solve the problem.
- ▶ How can we defend against character-level attacks?



Conclusions and future work

- ▶ Language models are prone to character-level attacks. **Few** characters are sufficient to poison them.
- ▶ Typo-correctors do not solve the problem.
- ▶ How can we defend against character-level attacks?
- ▶ Are character-level attacks a more potent attack in jailbreaking?



Conclusions and future work

- ▶ Language models are prone to character-level attacks. **Few** characters are sufficient to poison them.
- ▶ Typo-correctors do not solve the problem.
- ▶ How can we defend against character-level attacks?
- ▶ Are character-level attacks a more potent attack in jailbreaking?

Code:  <https://github.com/LIONS-EPFL/charmer>



Outline

Adversarial robustness: Where it all started

Defense against adversarial attacks

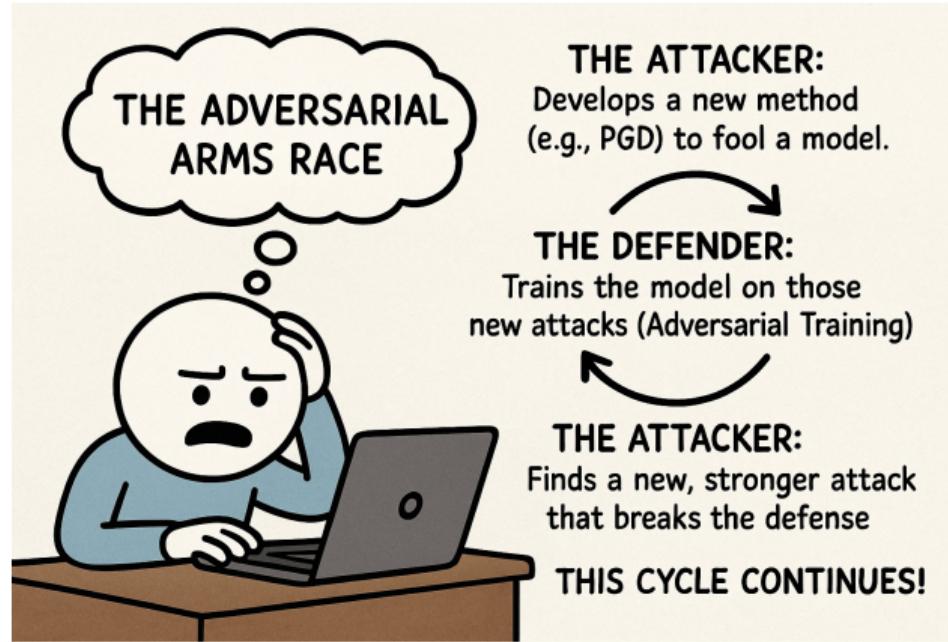
Adversarial robustness in text

Adversarial certificates

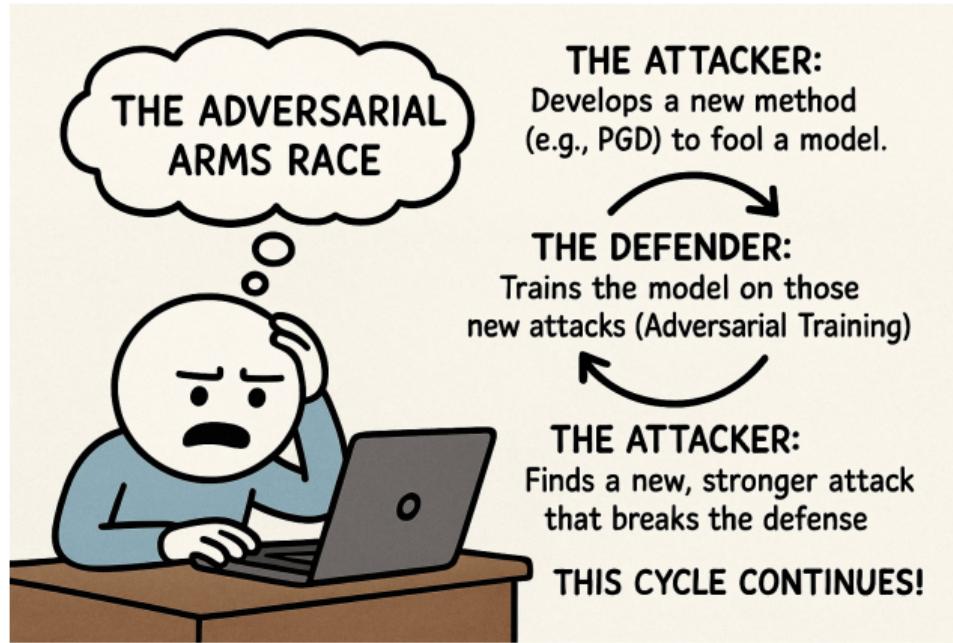
Challenges and future work



The Adversarial Attack-Defense race



The Adversarial Attack-Defense race



The Core Problem

Empirical defenses give a *false sense of security*. They only show robustness against attacks we already know. How can we be sure there isn't another, smarter attack just around the corner?



The Verification Problem: Can We Trust Our Models?

The Fundamental Question

Given a neural network f_θ and an input x , can we **guarantee** that $\arg \max f_\theta(x') = \arg \max f_\theta(x)$ for all x' in some neighborhood $\mathcal{B}(x, \epsilon)$?



The Verification Problem: Can We Trust Our Models?

The Fundamental Question

Given a neural network f_θ and an input x , can we **guarantee** that $\arg \max f_\theta(x') = \arg \max f_\theta(x)$ for all x' in some neighborhood $\mathcal{B}(x, \epsilon)$?

Why This Matters

- ▶ **Safety-critical systems:** Autonomous vehicles, medical diagnosis, aerospace
- ▶ **Security applications:** Malware detection, authentication systems
- ▶ **High-stakes decisions:** Legal, financial, hiring algorithms



Empirical vs. Certified Robustness

Empirical Robustness

- ▶ Test against known attacks (PGD, C&W, etc.)
- ▶ No guarantees against unknown attacks
- ▶ Fast and scalable
- ▶ Like testing software with some test cases

Certified Robustness

- ▶ Mathematical proof of robustness
- ▶ Guarantees against *all possible* attacks
- ▶ Computationally expensive
- ▶ Like formal verification in software

An analogy with bridges:



Empirical vs. Certified Robustness

Empirical Robustness

- ▶ Test against known attacks (PGD, C&W, etc.)
- ▶ No guarantees against unknown attacks
- ▶ Fast and scalable
- ▶ Like testing software with some test cases

Certified Robustness

- ▶ Mathematical proof of robustness
- ▶ Guarantees against *all possible* attacks
- ▶ Computationally expensive
- ▶ Like formal verification in software

An analogy with bridges:

- ▶ **Testing:** Checking if a bridge holds a 10-ton truck and a 20-ton truck.
- ▶ **Proving:** Using physics and engineering principles to prove the bridge can withstand *any* load up to 50 tons.



Empirical vs. Certified Robustness

Empirical Robustness

- ▶ Test against known attacks (PGD, C&W, etc.)
- ▶ No guarantees against unknown attacks
- ▶ Fast and scalable
- ▶ Like testing software with some test cases

Certified Robustness

- ▶ Mathematical proof of robustness
- ▶ Guarantees against *all possible* attacks
- ▶ Computationally expensive
- ▶ Like formal verification in software

An analogy with bridges:

- ▶ **Testing:** Checking if a bridge holds a 10-ton truck and a 20-ton truck.
- ▶ **Proving:** Using physics and engineering principles to prove the bridge can withstand *any* load up to 50 tons.

Key Insight

Empirical robustness \neq True robustness. We need certificates!



Certified Adversarial Robustness

The Certified Guarantee

For a given input x , a certification method provides a **guaranteed radius** R such that no adversarial attack within an ℓ_p -ball of that radius can change the classification.

$$\|\delta\|_p \leq R \implies \arg \max f(x + \delta) = \arg \max f(x) .$$



Complete Verification

Definition (Verification)

Given a classification model $f_{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}^k$, a pair (x, y_0) input region $\mathcal{X} = \{x : \|x - x_0\|_{\infty} \leq \epsilon\}$, such that :

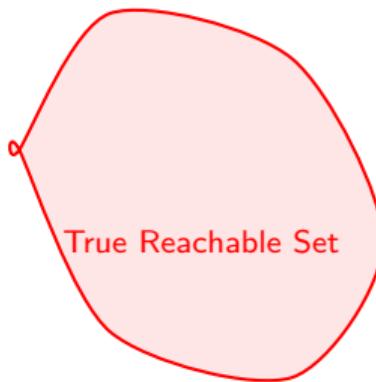
$$\text{Verify: } \max_{j \neq y_0} f_{\theta}(x)_j < f_{\theta}(x)_{y_0}, \quad \forall x \in \mathcal{X}.$$



Convex Relaxations: Making Verification Scalable

Key Idea (Wong and Kolter, 2018)

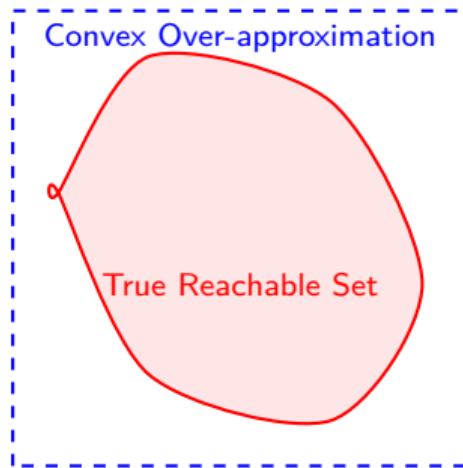
Replace the complex, non-convex verification problem with a **convex relaxation** that provides sound over-approximations.



Convex Relaxations: Making Verification Scalable

Key Idea (Wong and Kolter, 2018)

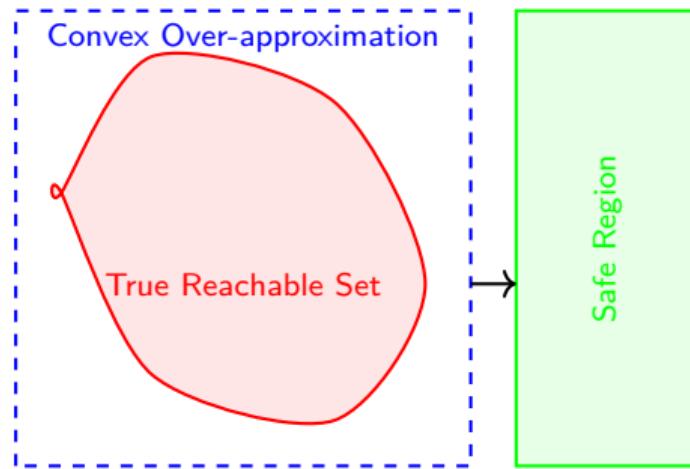
Replace the complex, non-convex verification problem with a **convex relaxation** that provides sound over-approximations.



Convex Relaxations: Making Verification Scalable

Key Idea (Wong and Kolter, 2018)

Replace the complex, non-convex verification problem with a **convex relaxation** that provides sound over-approximations.



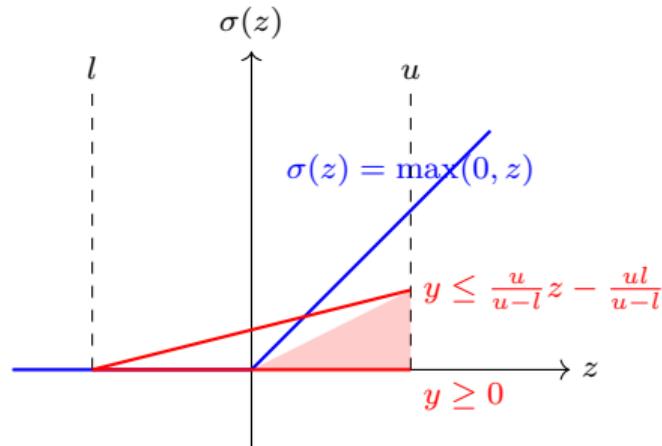
- ▶ If convex relaxation \subseteq safe region \Rightarrow certified robust
- ▶ If convex relaxation $\not\subseteq$ safe region \Rightarrow inconclusive
- ▶ Key challenge: Making relaxations as **tight** as possible



Linear Bound Propagation: Layer-by-Layer Analysis

The Linear Relaxation Approach

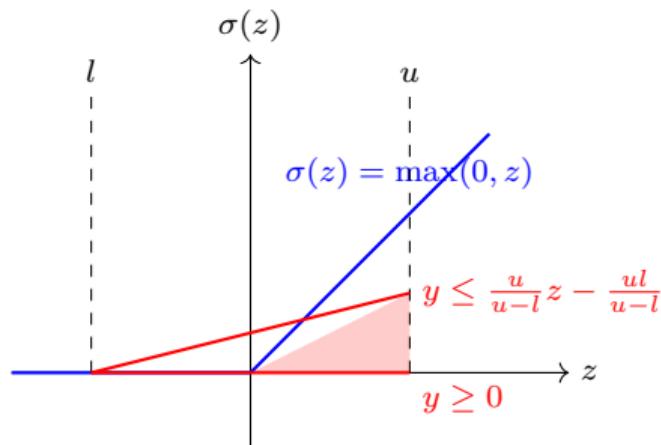
For ReLU activation $\sigma(z) = \max(0, z)$ with bounds $z \in [l, u]$:



Linear Bound Propagation: Layer-by-Layer Analysis

The Linear Relaxation Approach

For ReLU activation $\sigma(z) = \max(0, z)$ with bounds $z \in [l, u]$:



Linear constraints:

$$y \geq 0 \quad (4)$$

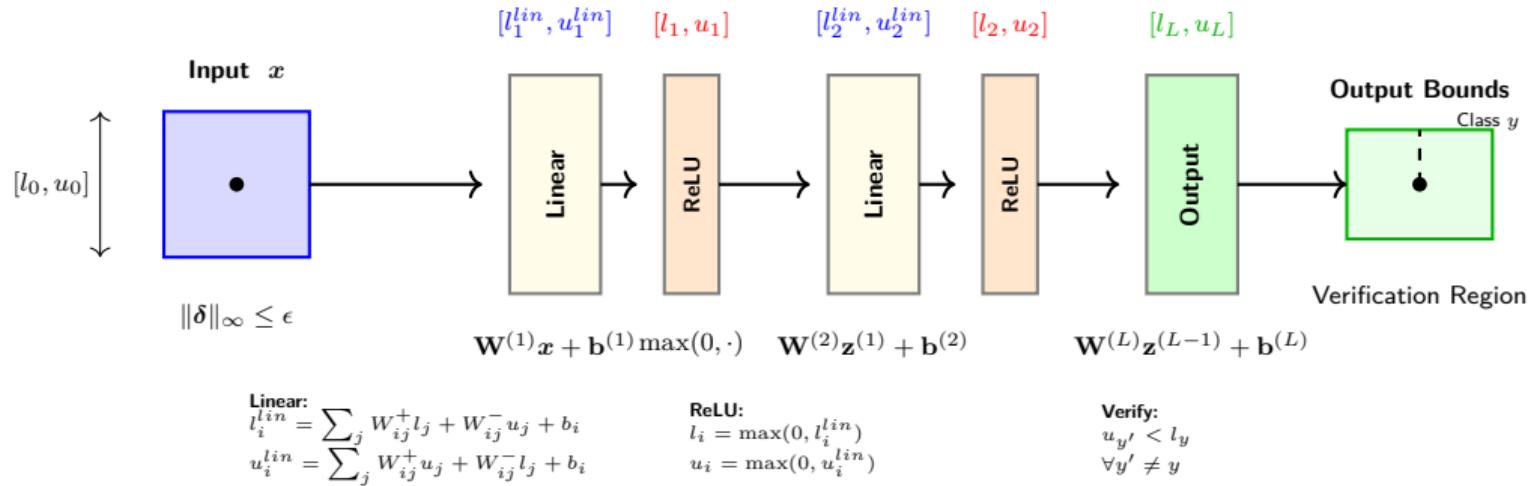
$$y \leq \frac{u}{u-l}(z - l) \quad (5)$$

Propagate through network:

- ▶ Start with input bounds
- ▶ Each layer: linear transformation + ReLU relaxation
- ▶ Final bounds on output logits



Interval Bound Propagation (IBP) for Verification

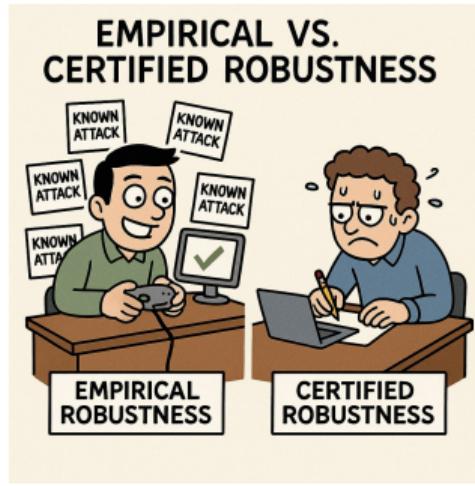


Interval Bound Propagation Algorithm

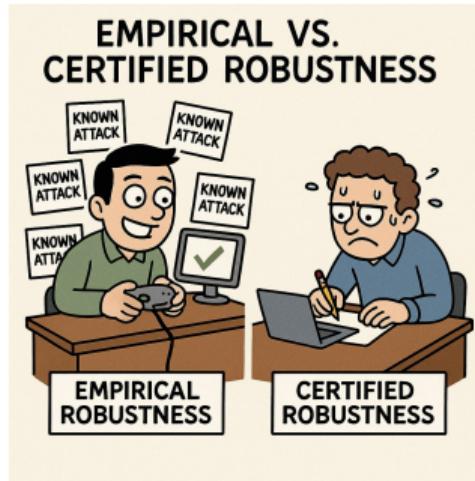
1. **Initialize:** Input bounds $[l_0, u_0]$ from $\| \delta \|_\infty \leq \epsilon$
2. **Forward Pass:** For each layer $i = 1, \dots, L$:
 - ▶ Compute linear bounds: $[l_i^{lin}, u_i^{lin}]$ using interval arithmetic
 - ▶ Apply activation: $[l_i, u_i] = \text{ReLU}([l_i^{lin}, u_i^{lin}])$
3. **Verify:** Check if $\max_{y' \neq y} u_{y'} < l_y$ at output layer.



Why don't we use verification instead of defense/attack methods then?



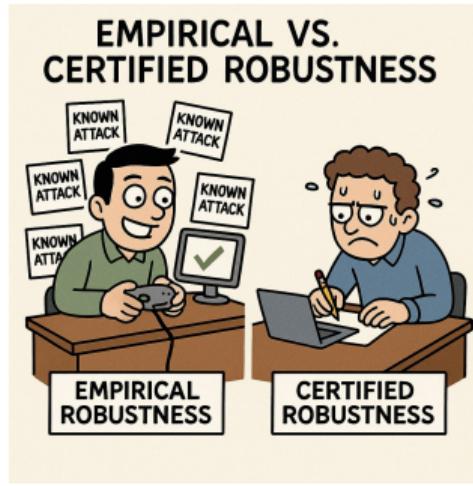
Why don't we use verification instead of defense/attack methods then?



A fundamental challenge: high certified robustness often results in lower standard accuracy on clean, unperturbed data.



Why don't we use verification instead of defense/attack methods then?



A fundamental challenge: high certified robustness often results in lower standard accuracy on clean, unperturbed data.

Interesting open questions

Is this trade-off fundamental? Or can we design new methods that are both highly accurate and provably robust?



Adversarial Robustness in NLP

Adversarial Robustness in NLP

Given a classification model f_{θ} , a sentence-label pair (S, y) and a subset $\mathcal{R}(S) \subset \mathcal{S}(\Gamma)$ where we would like to be robust, we would like to have:

$$f_{\theta}(S')_y - \max_{\hat{y} \neq y} f_{\theta}(S')_{\hat{y}} > 0, \quad \forall S' \in \mathcal{R}(S). \quad \bullet$$

- Solving $\min_{S' \in \mathcal{R}(S)} f_{\theta}(S')_y - \max_{\hat{y} \neq y} h_{\theta}(S')_{\hat{y}}$ is a hard problem Lei et al. (2019).
- **What if we just lowerbound** $f_{\theta}(S')_y - \max_{\hat{y} \neq y} f_{\theta}(S')_{\hat{y}}$?



Lipschitz verification

- Let us define $g_{y,\hat{y}}(S) = f_\theta(S)_y - f_\theta(S)_{\hat{y}}$.
- We want our function to have *Lipschitzness*:

$$|g_{y,\hat{y}}(S) - g_{y,\hat{y}}(S')| \leq G_{y,\hat{y}} \cdot d_{\text{lev}}(S, S') \quad \forall S' \in \mathcal{S}(\Gamma),$$

- With $G_{y,\hat{y}}$ we can compute the *certified radius* $k_y^*(S) := \min_{\hat{y} \neq y} \left\lfloor \frac{g_{y,\hat{y}}(S)}{G_{y,\hat{y}}} \right\rfloor$.



Verification of Character-level CNNs [Abad Rocamora, Chrysos, Cevher, ICLR (2025)]

- Let our l -layer classifier be:

$$f_{\theta}(S) = \left(\sum_{i=1}^{m+l \cdot (q-1)} f_i^{(l)}(S) \right) \theta, \text{ where } f^{(j)}(S) = \begin{cases} \sigma(C^{(j)}(f^{(j-1)}(S))) & \forall j = 1, \dots, l \\ SE & j = 0 \end{cases},$$

where σ is ReLU, E is the embedding matrix and $C^{(j)}$ are convolutional layers with parameters $\mathcal{K}^{(j)}$.

- Can be verified for character replacements with Interval Bound Propagation (Huang et al., 2019).



Outline

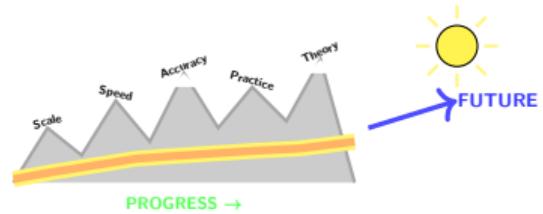
Adversarial robustness: Where it all started

Defense against adversarial attacks

Adversarial robustness in text

Adversarial certificates

Challenges and future work



Challenges in Robustness in the foundation model era

1. Can we create a unified verification framework that scales to modern architectures?



Challenges in Robustness in the foundation model era

1. Can we create a unified verification framework that scales to modern architectures?
2. Can we certify the robustness in large models, e.g., GPT-5, Gemini-2.5?



Challenges in Robustness in the foundation model era

1. Can we create a unified verification framework that scales to modern architectures?
2. Can we certify the robustness in large models, e.g., GPT-5, Gemini-2.5?
3. What is the impact of the model architecture on jailbreaking?



Challenges in Robustness in the foundation model era

1. Can we create a unified verification framework that scales to modern architectures?
2. Can we certify the robustness in large models, e.g., GPT-5, Gemini-2.5?
3. What is the impact of the model architecture on jailbreaking?
4. What is the role of post-training, e.g., reinforcement learning, in robustness?



Challenges in Certification in the foundation model era

- ▶ How can we certify even larger models (like LLMs) and more complex datasets efficiently?
- ▶ Can we certify properties of generative models, reinforcement learning agents, or object detectors?
- ▶ Can we develop certified defenses that don't suffer from a major drop in standard accuracy?



Collaborators on topics presented today (alphabetical order):

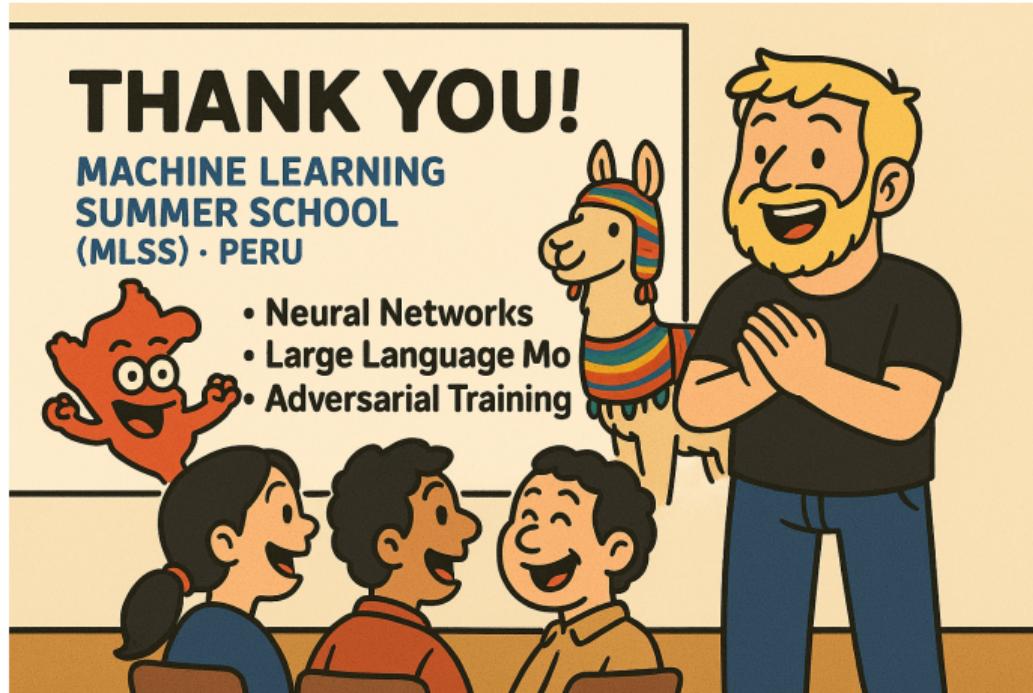
Vokan Cevher, Fabian Latorre, Fanghui Liu, Thomas Pethick, Elias Abad Rocamora, CJ Simon-Gabriel, Yongtao Wu, Zhenyu Zhu.

Special thanks to the UW Madison team (alphabetical order):

Muhammad Ashiq, Andrea Tseng, Yiheng Zhang



Thank you for your attention!



Source: GPT-5.



References |

- Elias Abad Rocamora, Grigoris Chrysos, and Volkan Cevher. Certified robustness under bounded levenshtein distance. In *International Conference on Learning Representations (ICLR)*, 2025.
- Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning (ICML)*, pages 242–252, 2019.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*, 2019.
- Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.



References II

- Qi Lei, Lingfei Wu, Pin-Yu Chen, Alex Dimakis, Inderjit S Dhillon, and Michael J Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Proceedings of Machine Learning and Systems*, 1:146–165, 2019.
- Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.
- Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2019.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.
- Danish Pruthi, Bhwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- Alexander Robey, Fabian Latorre, George J Pappas, Hamed Hassani, and Volkan Cevher. Adversarial training should be cast as a non-zero-sum game. In *International Conference on Learning Representations (ICLR)*, 2023.
- Elias Abad Rocamora, Yongtao Wu, Fanghui Liu, Grigoris G Chrysos, and Volkan Cevher. Revisiting character-level adversarial attacks for language models. In *International Conference on Machine Learning (ICML)*, 2024.



References III

- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.
- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning (ICML)*, pages 5286–5295, 2018.
- Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, pages 7472–7482, 2019.
- Zhenyu Zhu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 36094–36107, 2022.

