Towards Efficient Methods for Training Robust Deep Neural Networks

Sanjit Bhat (MIT PRIMES), Mentor: Dimitris Tsipras (MIT)
PRIMES CS Conference, October 13, 2018

Acknowledgements

Thank you to:

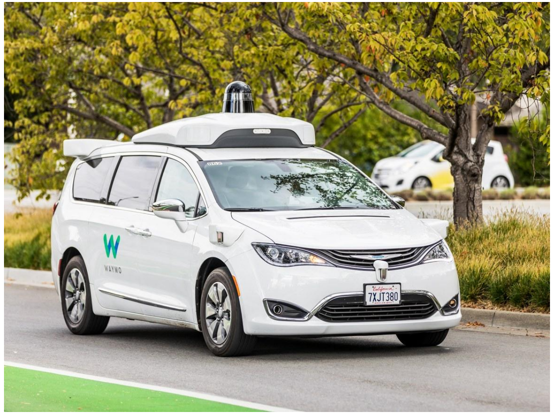
- My parents, for their emotional support
- Dimitris Tsipras, for the discussions, advice, and motivation
- Prof. Srini Devadas, for the PRIMES CS track
- Dr. Slava Gerovitch, for the PRIMES program. Having the opportunity to pursue research in high school has been truly amazing :-)

Introduction

Deep Learning (DL) can surpass humans

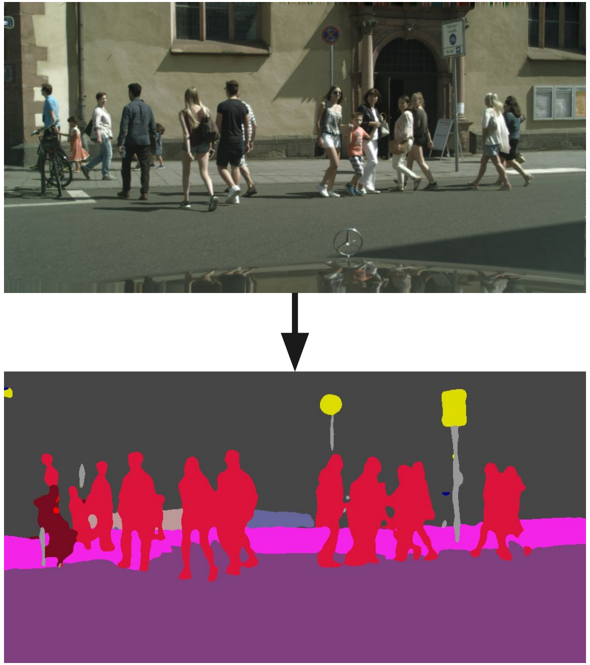
Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

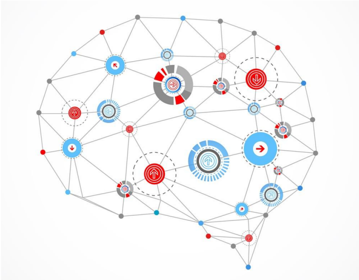
DL in security-critical applications



Is DL ready for this?

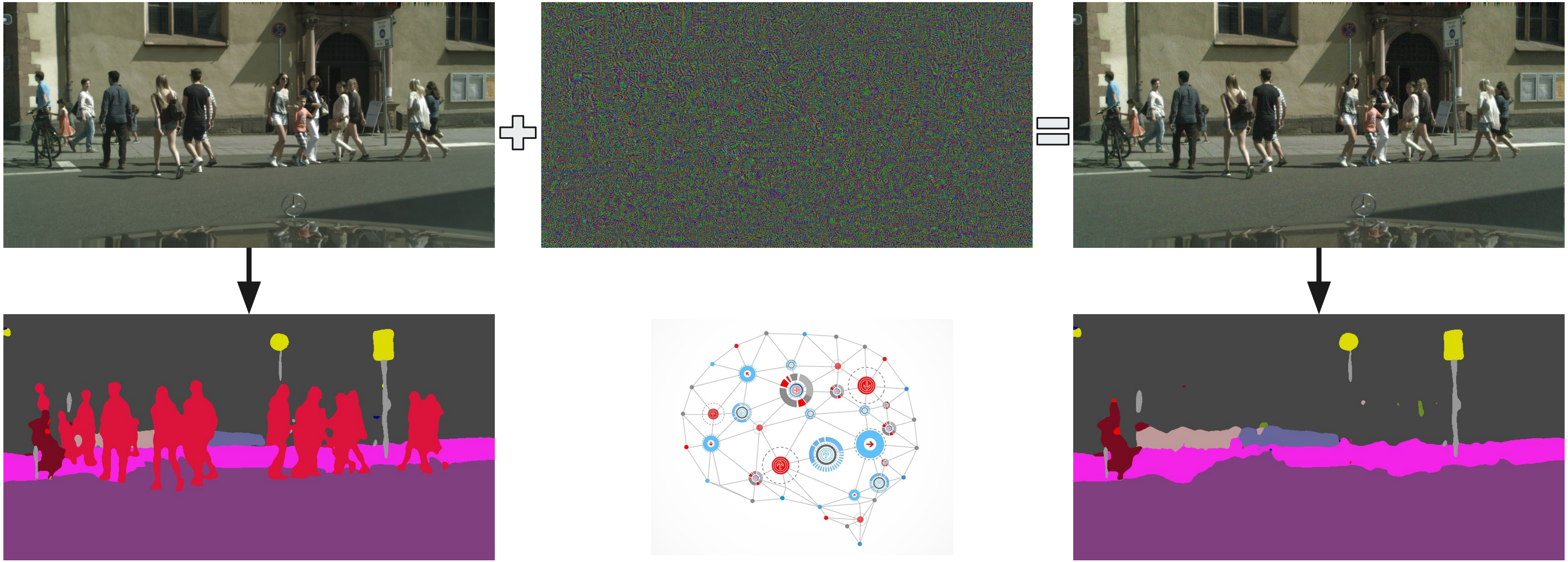
Deep Neural Network (DNN) - Natural Setting





V. Fischer, M. Kumar, J. Metzen, T. Brox
“Adversarial Examples for Semantic Image Segmentation”

DNN - Adversarial Setting



Why do we need robust DNNs?

Reliability

- Some natural phenomena (e.g., rain) can trick classifiers
- Train more reliable natural classifiers

Intelligence

- Goal of ML: Make intelligent systems
- Humans wouldn’t get fooled, but these systems do

Background

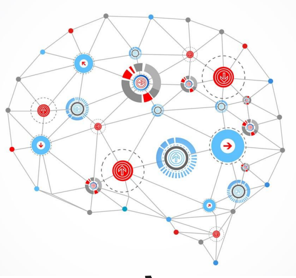
How do we train robust DNNs?

Adversarial Training - A robust training method

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}} [\mathcal{L}(x, y; \theta)]$$

Natural Training Set

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



Model Parameters

Adversarial Training - A robust training method

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

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

Why is Adversarial Training difficult?

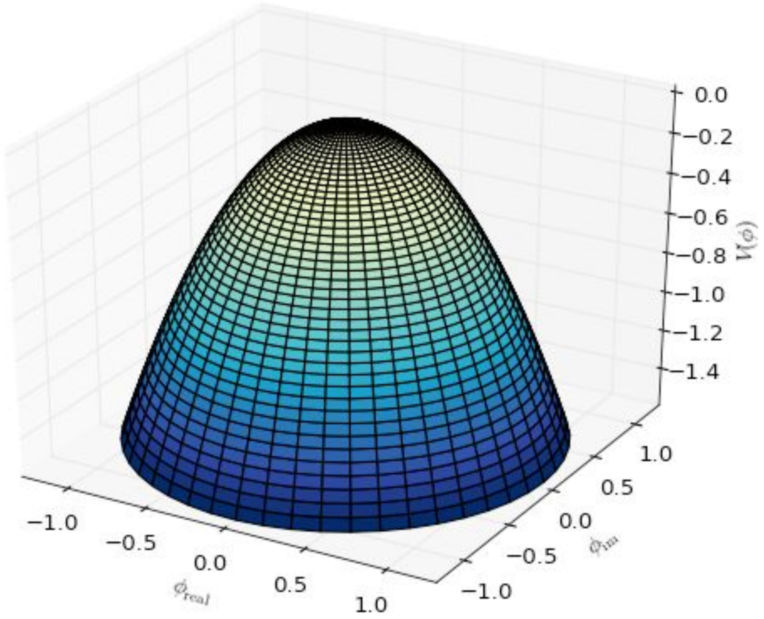
- Takes a long time to compute good adversarial examples
- Training waits for these examples
- Process repeats several times before DNN finally becomes robust

➢ **Time-Intensive Process**

Research focus: How can we make Adversarial Training more efficient?

Technique 1: A closer look at Adversarial Training

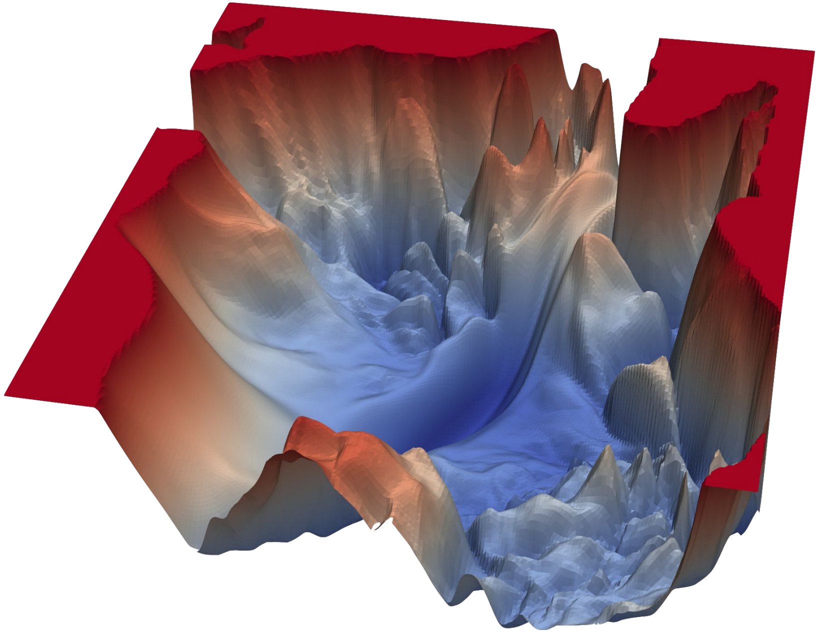
Concave loss landscapes are easily maximizable



- Goal of adversary: Get to maximum loss
- Hypothetical loss landscape

DNNs have tricky, non-concave loss landscapes

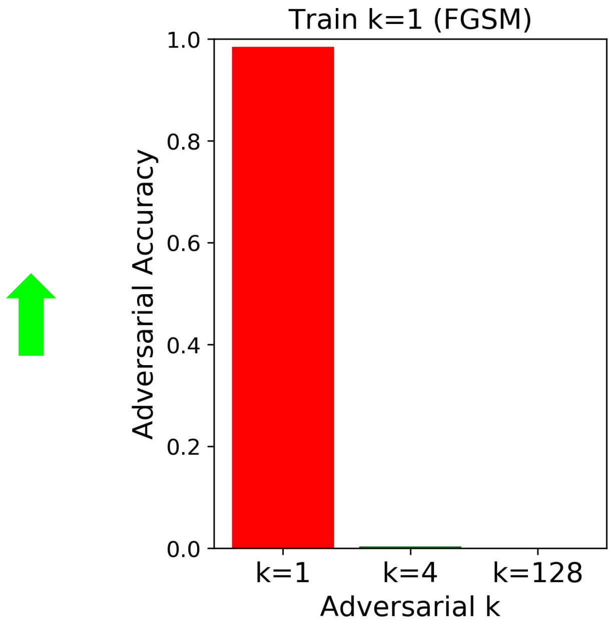
- Actual loss landscape
- Hard to find maxima, so need multiple steps
- Each step re-calculates trajectory, which is **Time Intensive**



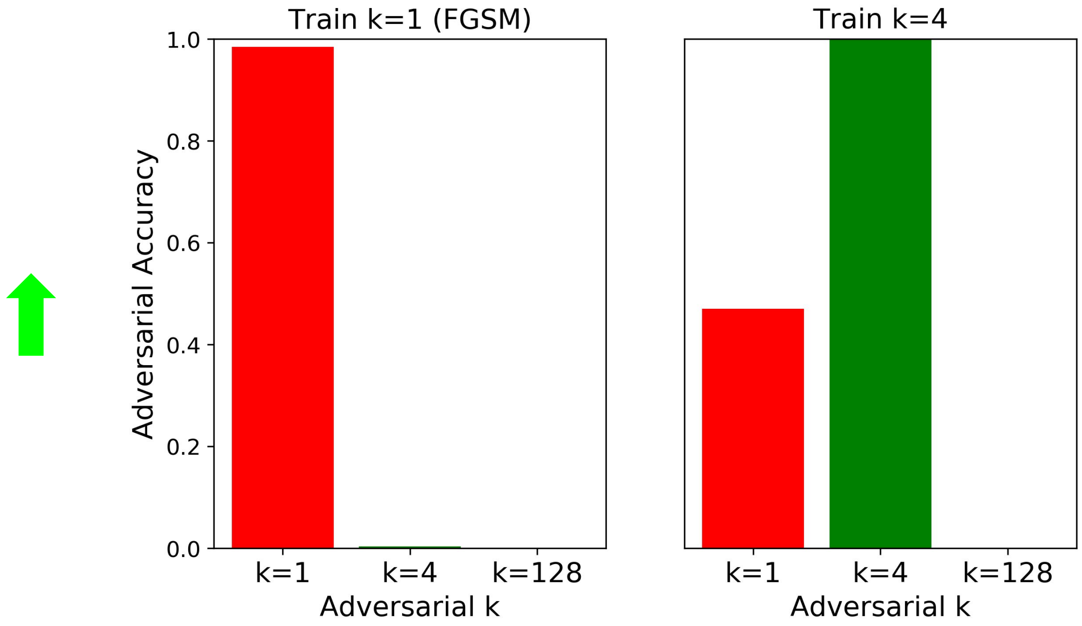
H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein
“Visualizing the Loss Landscape of Neural Nets”

How strong does the adversary need to be?

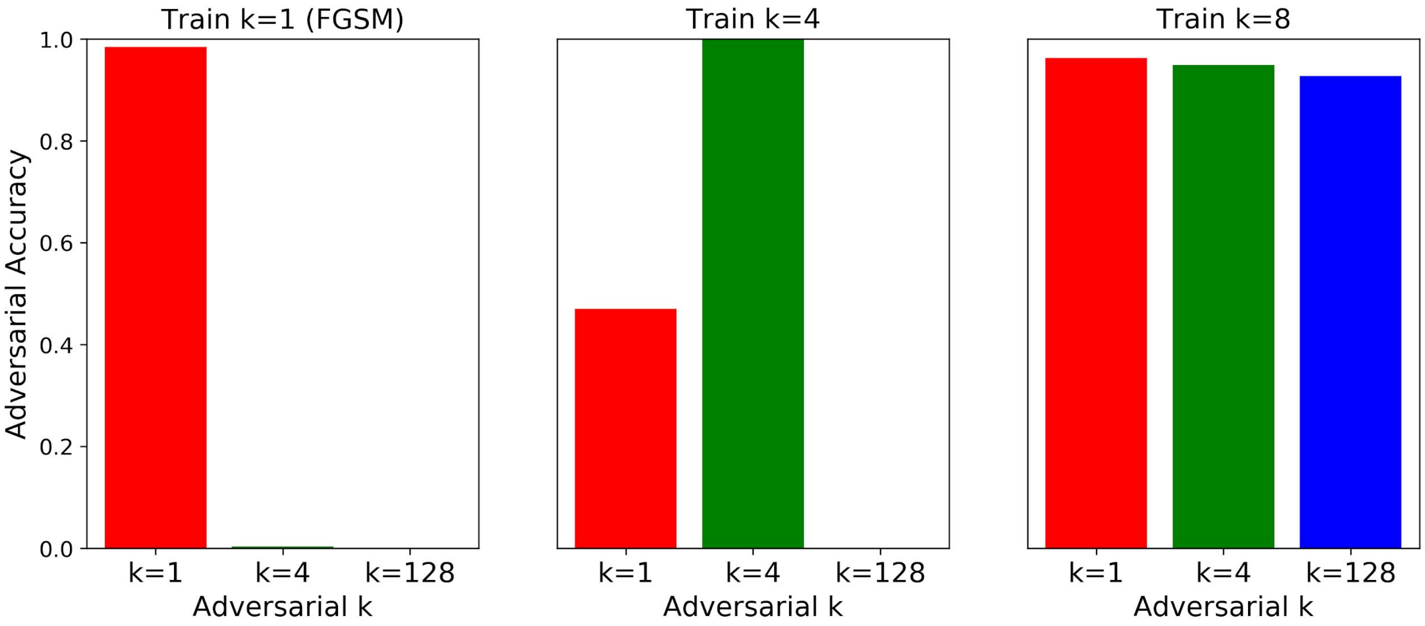
How strong does the adversary need to be?



How strong does the adversary need to be?

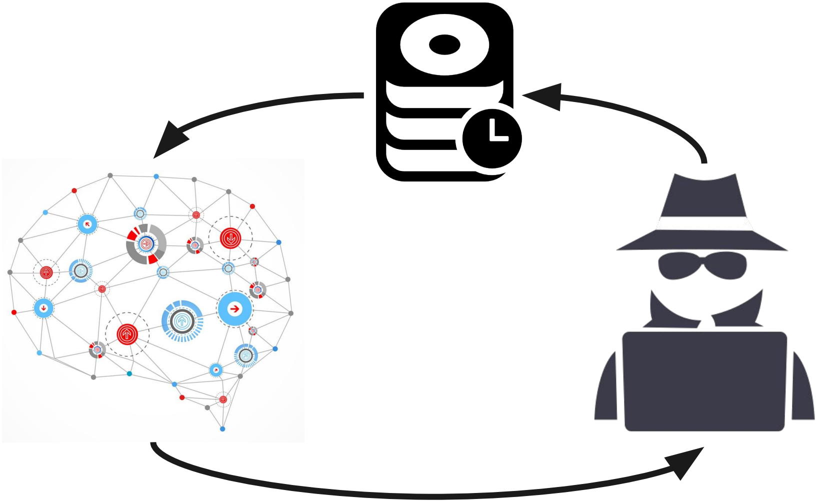


How strong does the adversary need to be?

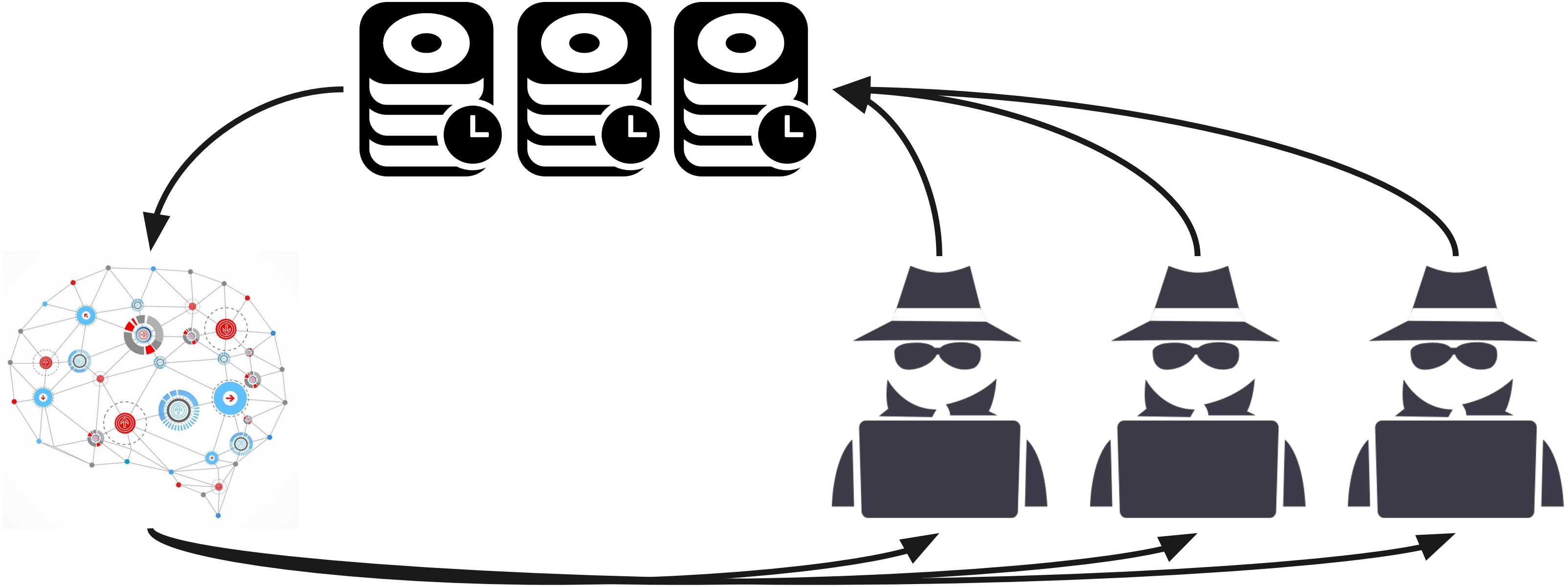


Technique 2: Asynchronous parallelization

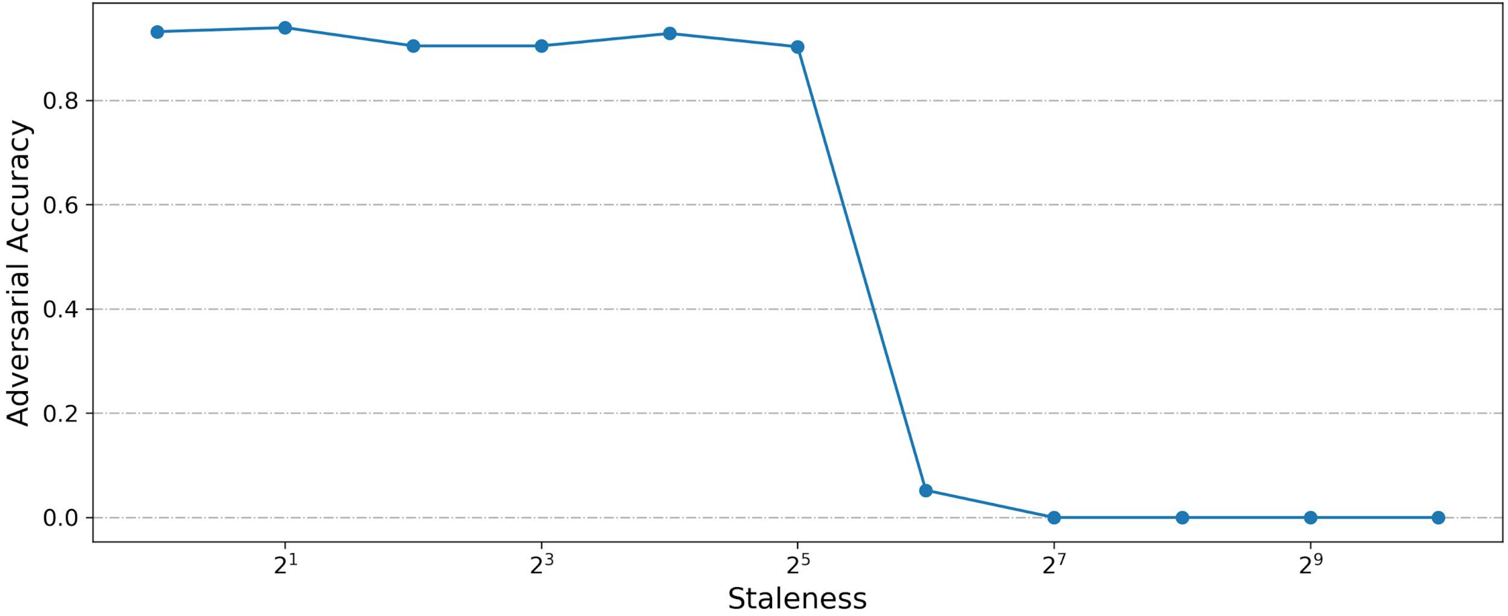
Re-Visiting Adversarial Training



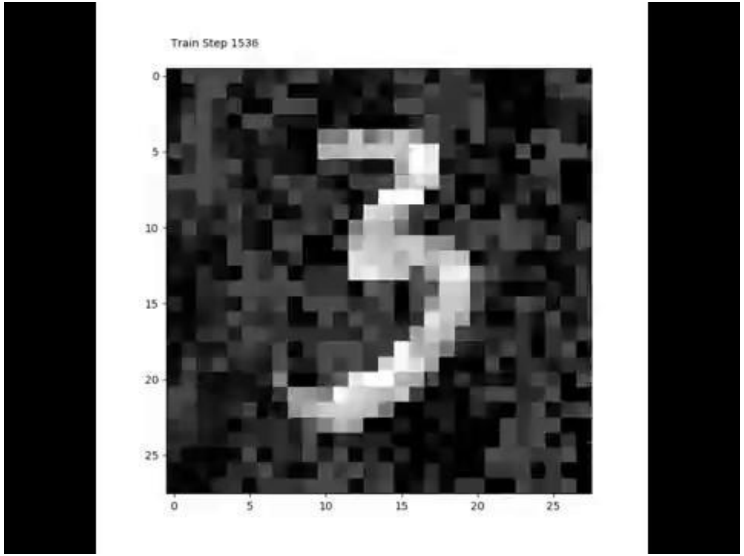
Re-Visiting Adversarial Training



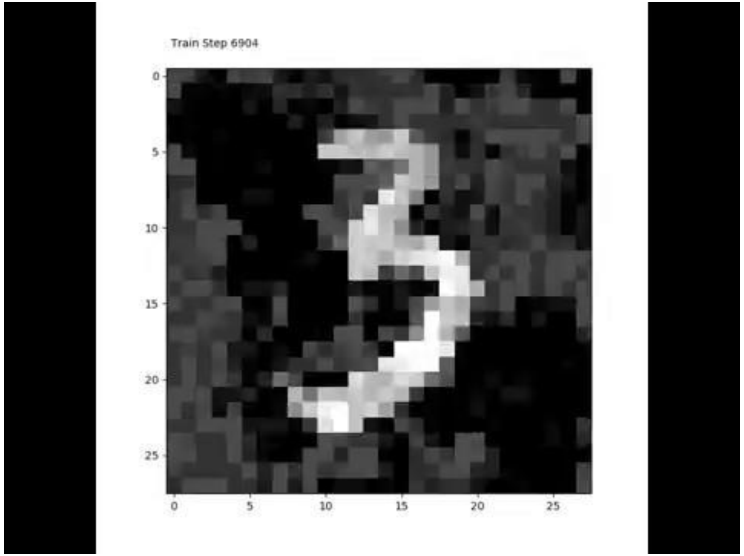
High staleness training doesn't work



Staleness can be pathological

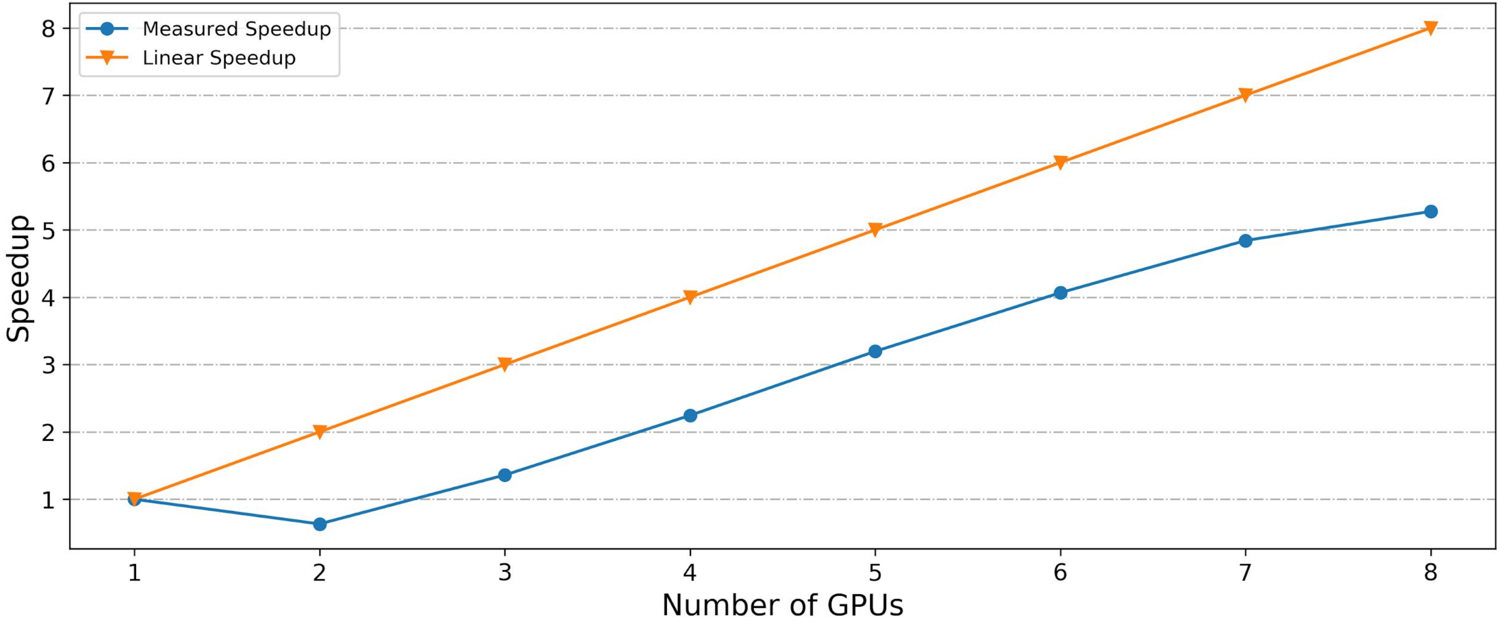


32 Staleness (Good)



64 Staleness (Bad)

Almost-linear speedup



4 hrs to 9 mins

Combining both techniques, we achieve a 26x reduction in robust MNIST training time