Terminal Brain Damage:
Exposing the Graceless Degradation in Deep Neural Networks under Hardware Fault Attacks

Sanghyun Hong[1], Pietro Frigo[2], Yiğitcan Kaya[1], Cristiano Guiffrida[2], Tudor Dumitraș[1]

[1]*University of Maryland, College Park*, [2]*Vrije Universiteit Amsterdam*

1990: Optimal Brain Damage – Graceful Degradations

: we can remove 60% of model parameters, without the accuracy drop

DNN's Resilience – False Sense of Security

- **Techniques** that rely on the *graceful degradation*
  - **Parameter pruning**[1]: to reduce the inference cost
  - **Parameter quantization**[2]: to compress the network size
  - **Blend noises to parameters**[3]: to improve the robustness

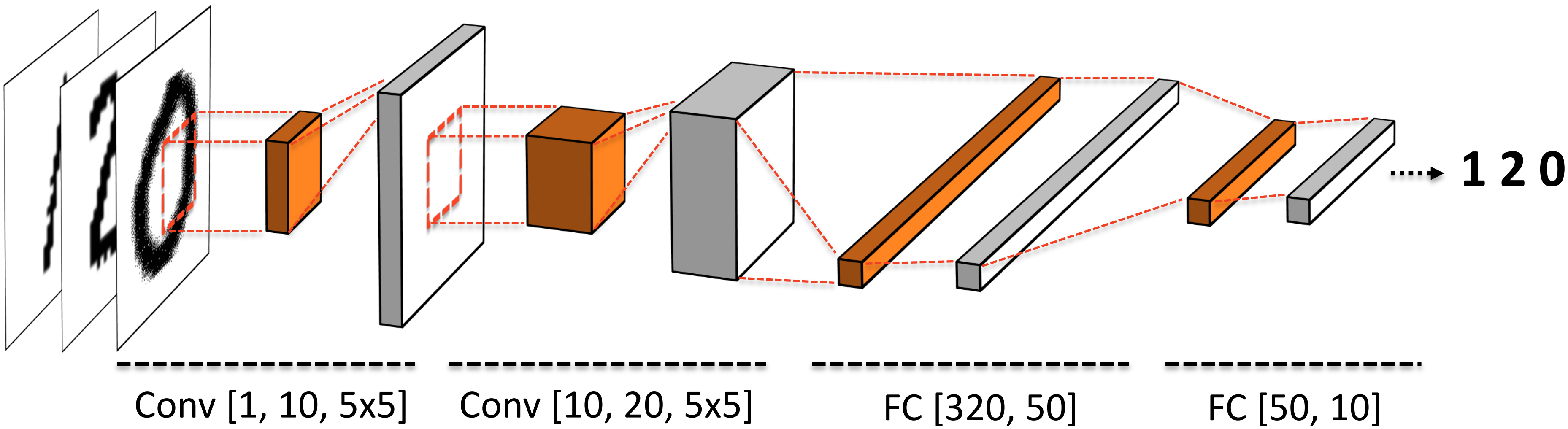
- **Prior work** showed it is *difficult to cause the accuracy drop*
  - **Indiscriminate poisoning**[4]: blend a lot of poisons ≈ *11% drop*
  - **Storage media errors**[5]: a lot of random bit errors ≈ *5% drop*
  - **Hardware fault attacks**[6,7]: a lot of random faults ≈ *7% drops*

They focus on the best-case or the average-case perturbations

What is the **WORST-CASE perturbation** (a bit-flip) that inflicts a **SIGNIFICANT** accuracy drop exceeding 10%?

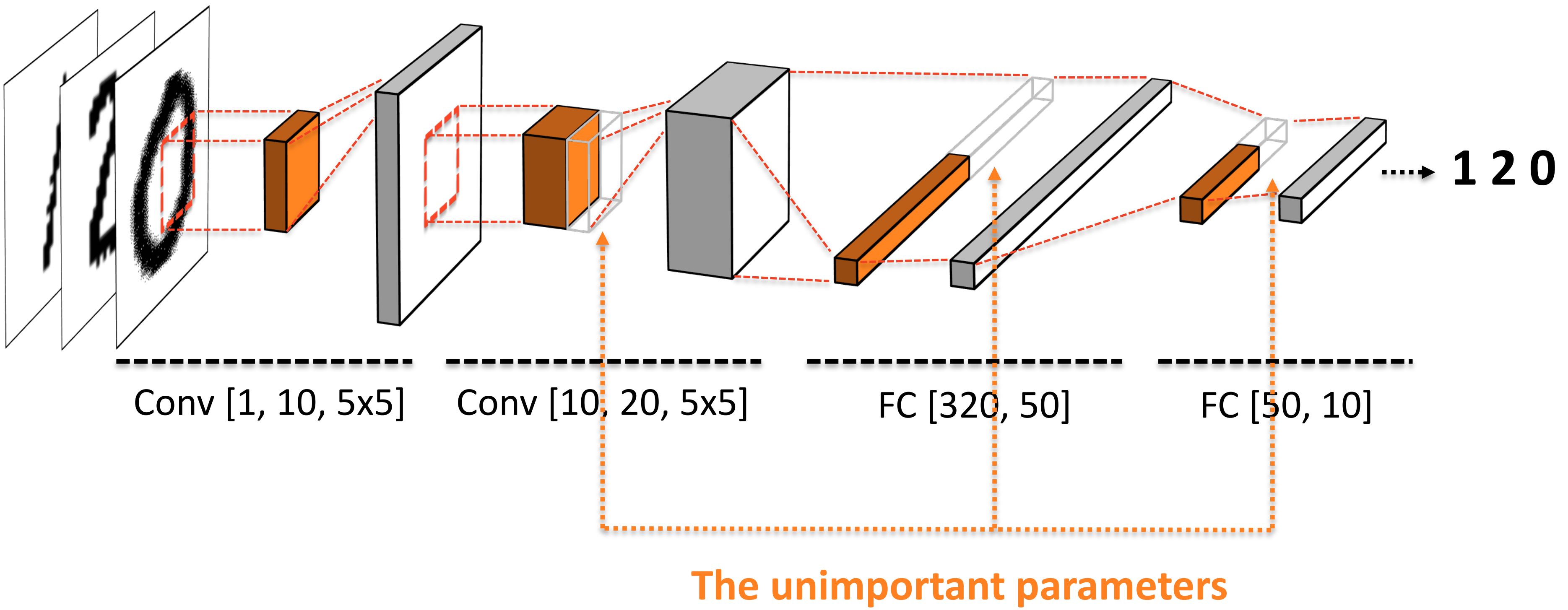
Illustration: How DNN Computes

- Accuracy: 98.53%



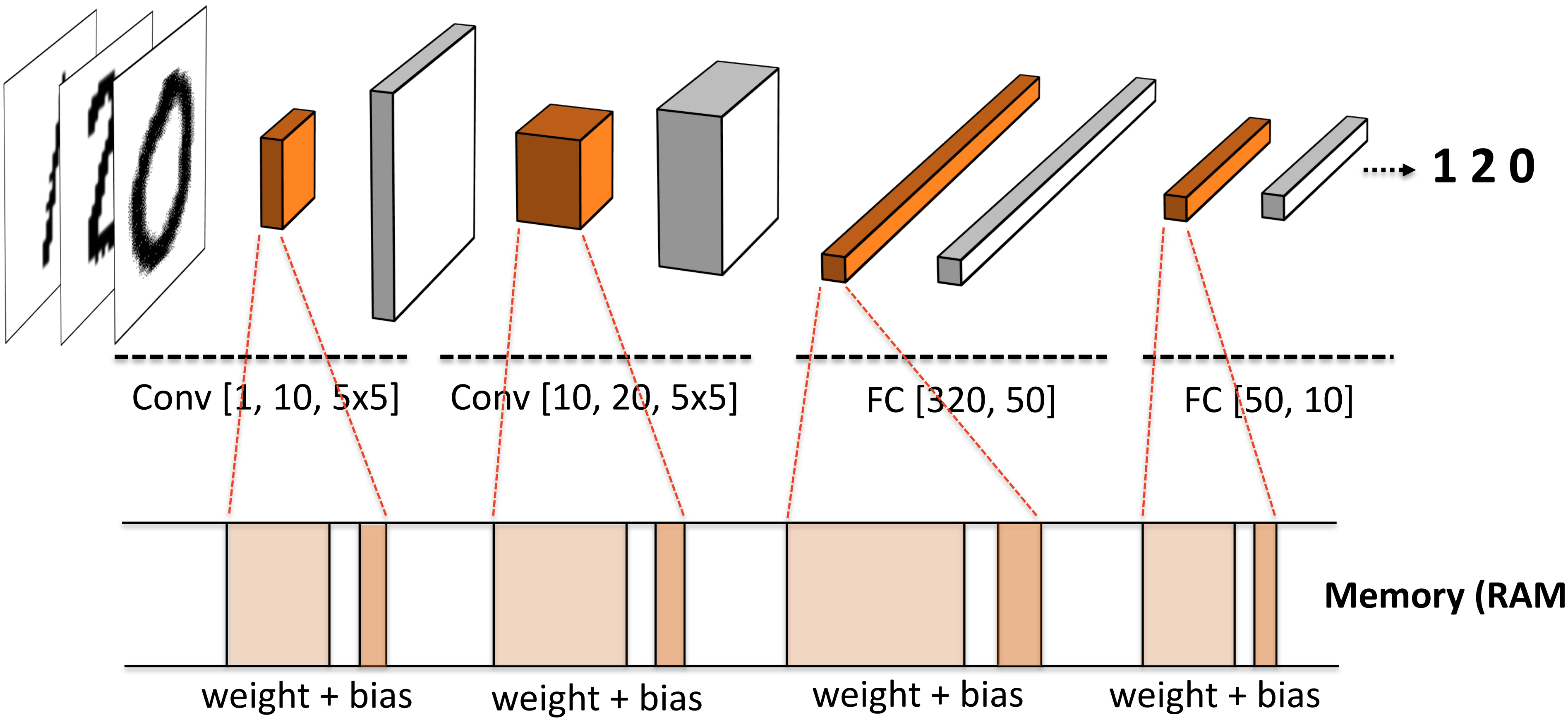
Prior Work: Optimal Brain Damage

- Accuracy: **98.53% (0% drop)**



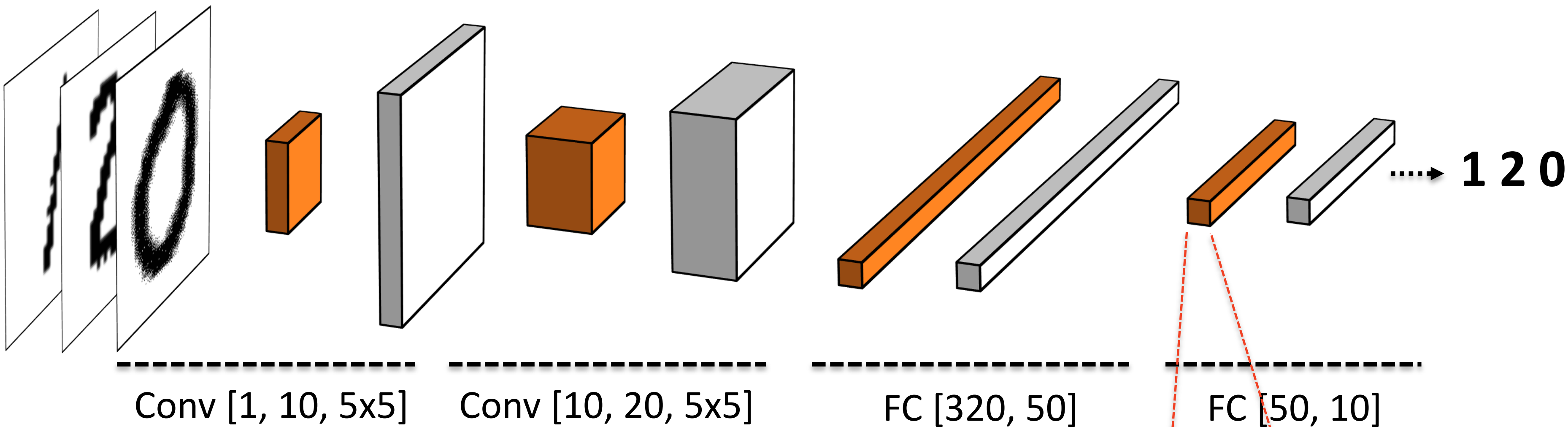
Prior Work: Hardware Fault Attacks

- Accuracy: 98.53%



Prior Work: Hardware Fault Attacks

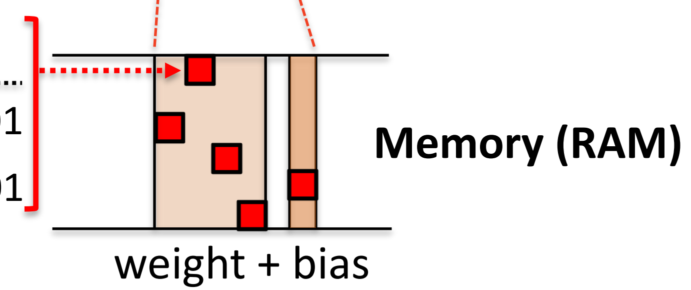
- Accuracy: **93.53% (5% drop)**



Sign | Exponent | Mantissa

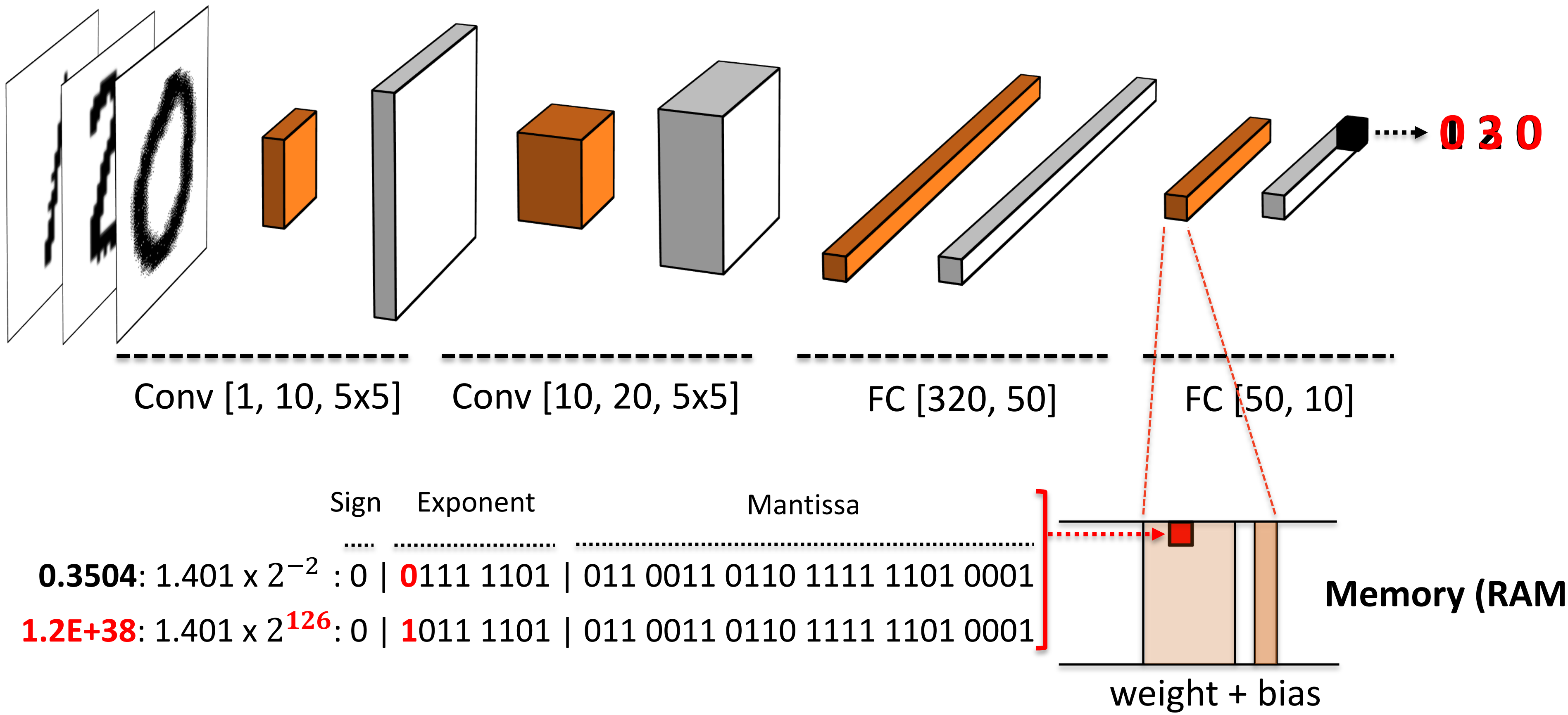
0.3504: 1.401 x 2^{-2} : 0 | 0111 1101 | 011 0011 0110 1111 1101 0001

0.0219: 1.401 x 2^{-6} : 0 | 0011 1001 | 011 0011 0110 1111 1101 0001



Can We Find a Worst-case Bit-flip?

- Accuracy: **57.52% (41.01% drop)**



Research Questions

- **RQ-1**: How vulnerable are DNNs to a single bit-flip?
- **RQ-2**: What properties influence this vulnerability?
- **RQ-3**: Can an attacker exploit this vulnerability?
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?

Research Questions

- **RQ-1**: How vulnerable are DNNs to a single bit-flip?
- **RQ-2**: What properties influence this vulnerability?
- **RQ-3**: Can an attacker exploit this vulnerability?
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?

RQ-1: How Vulnerable are DNNs to a Bit-flip?

- Metric
  - Relative Accuracy Drop [RAD] = $\frac{(acc_{clean} - acc_{corrupted})}{acc_{clean}}$
- Methodology
  - Flip (0→1 and 1→0) each bit in all parameters of a model
  - Measure the RAD over the entire validation set, each time
  - **Achilles bit**: when the bit flips, the flip inflicts **RAD > 10%**
- Vulnerability
  - **Max RAD**: the maximum RAD that an Achilles bit can inflict
  - **Ratio**: the percentage of vulnerable parameters in a model

RQ-1: Vulnerability Analysis in MNIST

Network	Acc.	# Params	Max RAD	Ratio
B(ase)	95.71	21,840	98 %	50%
B-Wide	98.46	85,670	99 %	50%
B-PReLU	98.13	21,843	99 %	99%
B-Dropout	96.86	21,840	99 %	49%
B-DP-Norm	97.97	21,962	99 %	51%
L(eNet)5	98.81	61,706	99 %	47%
L5-Dropout	98.72	61,706	99 %	45%
L5-D-Norm	99.05	62,598	98 %	49%

- Maximum RAD ≈ 98% in all models
- > 45% of params are vulnerable in all the MNIST models

RQ-1: How Vulnerable Are Larger Models?

- Metric
  - Relative Accuracy Drop [**RAD**] = $\frac{(acc_{clean} - acc_{corrupted})}{acc_{clean}}$
- Methodology
  - Flip (0→1 and 1→0) each bit in all parameters of a model
  - Measure the RAD over the entire validation set, each time [e.g. VGG16-ImageNet: examine 138M parameters ≈ **942 days**]

RQ-1: How Vulnerable Are Larger Models?

- Metric
  - Relative Accuracy Drop [**RAD**] = $\frac{(acc_{clean} - acc_{corrupted})}{acc_{clean}}$
- Methodology
  - Flip (0→1 and 1→0) each bit in all parameters of a model
  - Measure the RAD over the entire validation set, each time
- Speed-up heuristics
  - Sampled validation set (SV): use **10% of the** validation set
  - Inspect only specific bits (SB): **the exponents** or **their MSBs**
  - Sampled parameters (SP): uniformly **sample 20k parameters**

RQ-1: Vulnerability Analysis in Large Models

Dataset	Network	Acc.	# Params	SV	SB	SP	Max RAD	Ratio
CIFAR-10	B(ase)	83.74	776K	✓	$\checkmark_{exp}$	X		
	B-Slim	82.19	197K	✓	$\checkmark_{exp}$	X		
	B-Dropout	81.18	776K	✓	$\checkmark_{exp}$	X		
	B-D-Norm	80.17	778K	✓	$\checkmark_{exp}$	X		
	AlexNet	83.96	2.5M	✓	$\checkmark_{exp}$	X		
	VGG16	91.34	14.7M	✓	$\checkmark_{exp}$	X		
ImageNet	AlexNet	79.07	61.1M	✓	$\checkmark_{31st}$	✓ (20K)		
	VGG16	90.38	138.4M	✓	$\checkmark_{31st}$	✓ (20K)		
	ResNet50	92.86	25.6M	✓	$\checkmark_{31st}$	✓ (20K)		
	DenseNet161	93.56	28.9M	✓	$\checkmark_{31st}$	✓ (20K)		
	InceptionV3	88.65	27.2M	✓	$\checkmark_{31st}$	✓ (20K)		

RQ-1: Vulnerability Analysis in Large Models

Dataset	Network	Acc.	# Params	SV	SB	SP	Max RAD	Ratio
CIFAR-10	B(ase)	83.74	776K	✓	$\checkmark_{exp}$	✗	94 %	46.8%
	B-Slim	82.19	197K	✓	$\checkmark_{exp}$	✗	93 %	46.7%
	B-Dropout	81.18	776K	✓	$\checkmark_{exp}$	✗	94 %	40.5%
	B-D-Norm	80.17	778K	✓	$\checkmark_{exp}$	✗	97 %	45.9%
	AlexNet	83.96	2.5M	✓	$\checkmark_{exp}$	✗	96 %	47.3%
	VGG16	91.34	14.7M	✓	$\checkmark_{exp}$	✗	99 %	46.2%
ImageNet	AlexNet	79.07	61.1M	✓	$\checkmark_{31st}$	✓ (20K)	100 %	47.3%
	VGG16	90.38	138.4M	✓	$\checkmark_{31st}$	✓ (20K)	99 %	42.1%
	ResNet50	92.86	25.6M	✓	$\checkmark_{31st}$	✓ (20K)	100 %	47.8%
	DenseNet161	93.56	28.9M	✓	$\checkmark_{31st}$	✓ (20K)	100 %	49.0%
	InceptionV3	88.65	27.2M	✓	$\checkmark_{31st}$	✓ (20K)	100 %	40.8%

Research Questions

- **RQ-1**: How vulnerable are DNNs to a single bit-flip?
- **RQ-2**: What properties influence this vulnerability?
- **RQ-3**: Can an attacker exploit this vulnerability?
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?

RQ-2: Properties that Influence the Vulnerability

- (Network-level) DNN-properties
- (Parameter-level) Bitwise representation

RQ-2: Impact of the Common Techniques

- (Network-level) DNN-properties
  - The **dropout** and **batch-norm do not affect** the vulnerability

Dataset	Network	Base acc.	# Params	SV	SB	SP	Max RAD	Ratio
MINIST	L(eNet)5	98.81	61,706	✗	✗	✗	**99 %**	**47%**
	L5-Dropout	98.72	61,706	✗	✗	✗	**99 %**	**45%**
	L5-D-Norm	99.05	62,598	✗	✗	✗	**98 %**	**49%**
CIFAR-10	B(ase)	83.74	776 K	✓	✓	✗	**94 %**	**47%**
	B-Dropout	81.18	776 K	✓	✓	✗	**94 %**	**41%**
	B-D-Norm	80.17	778 K	✓	✓	✗	**97 %**	**46%**

RQ-2: Impact of the Other DNN Properties

- (Network-level) DNN-properties
  - The **dropout** and **batch-norm cannot reduce** the vulnerability
  - The **vulnerability increases proportionally** with the width
  - The **activation with negative values doubles** the vulnerability
  - The **vulnerability is consistent** across 19 DNNs' architectures
    - [8 MNIST, 5 CIFAR-10, and 5 ImageNet architectures]

RQ-2: Impact of the Parameter Sign

- (Parameter-level) Bitwise representation
  - Flip **the MSB of the exponents** mostly lead to [RAD > 10%]
  - The **only (0→1) flip direction** leads to [RAD > 10%]
  - The **positive parameters** are likely to be vulnerable to bit-flips than the negative parameters

Research Questions

- **RQ-1**: How vulnerable are DNNs to a single bit-flip?
- **RQ-2**: What properties influence this vulnerability?
- **RQ-3**: Can an attacker exploit this vulnerability?
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?

RQ-3: Threat Model – Attacker's Capability

- Capability
  - **Surgical:** can cause a bit-flip at an intended location
  - **Blind:** cannot control the location of a bit-flip

RQ-3: Threat Model – Attacker's Knowledge

- Capability
  - **Surgical:** can cause a bit-flip at an intended location
  - **Blind:** cannot control the location of a bit-flip

- Knowledge:
  - **White-box:** knows the victim model internals
  - **Black-box:** has no knowledge of the victim model

RQ-3: Threat Model – Single Bit Adversary

Surgical

Strongest attacker

$P(hit\ an\ Achilles\ bit) \approx$ **100%**

Black-box

White-box

Weakest attacker

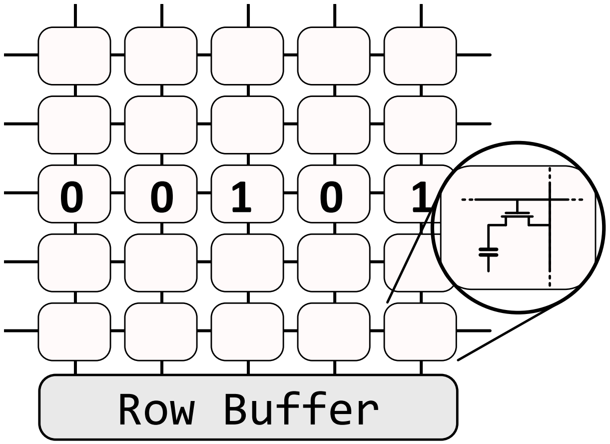
$P(hit\ an\ Achilles\ bit) \approx \varepsilon$

[VGG16: 42.1% / 32-bits ≈ 1.32%]

Blind

RQ-3: Practical Weapon – Rowhammer

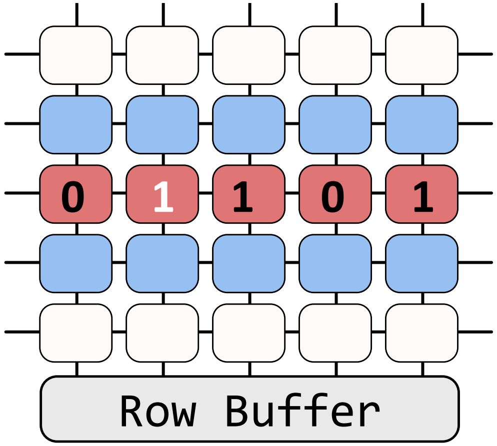
- Rowhammer attacks
  - **Single-bit corruption primitives** at DRAM-level
  - **Software-induced** hardware fault attacks
    [The attacker only requires a user-level access to memory]



Double-sided Rowhammer attack

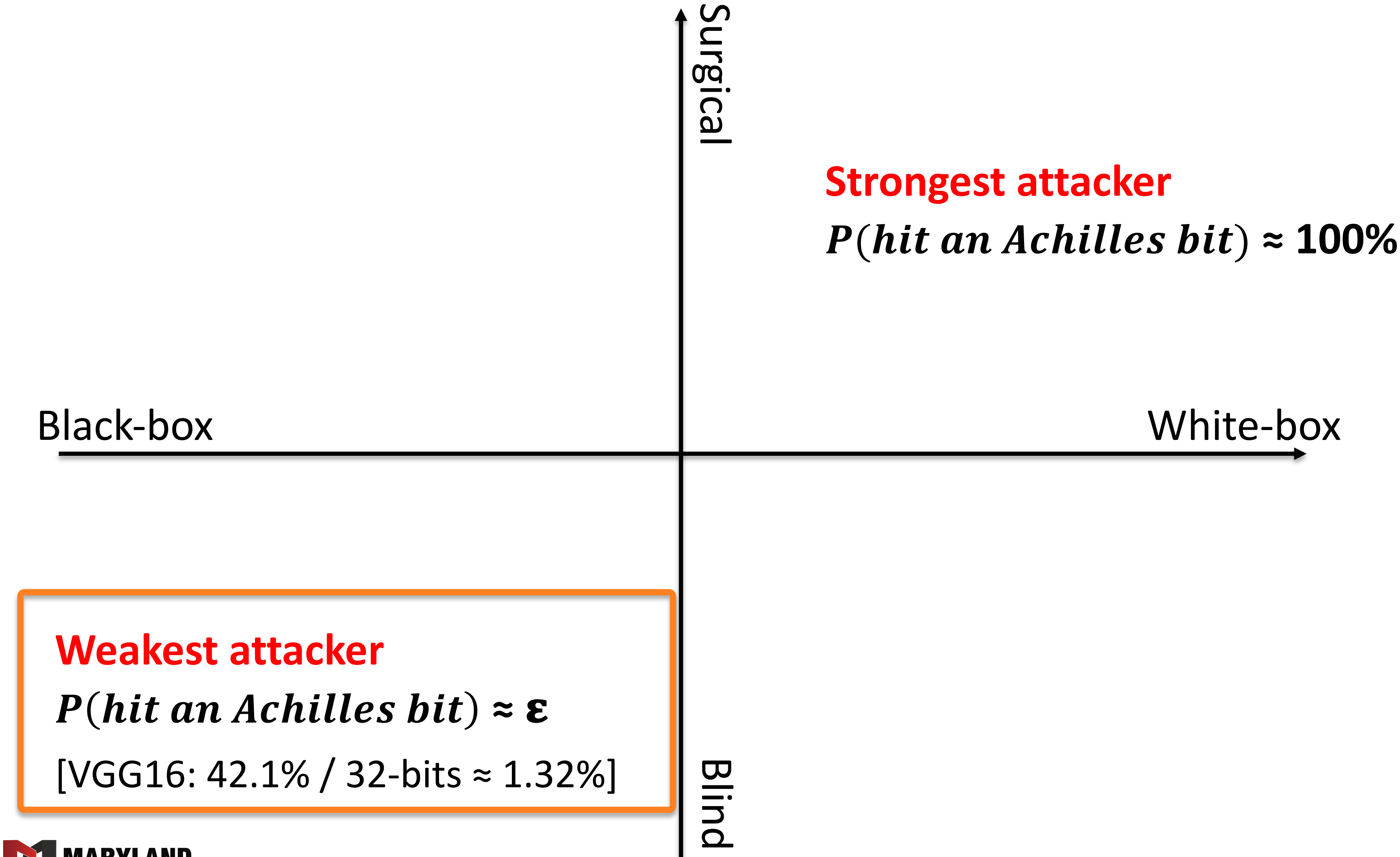
RQ-3: Practical Weapon – Rowhammer

- Rowhammer attacks
  - **Single-bit corruption primitives** at DRAM-level
  - **Software-induced** hardware fault attacks
    [The attacker only requires a user-level access to memory]



Double-sided Rowhammer attack

RQ-3: Threat Model (Re-visited)



RQ-3: If Our Adversary Can Flip Multiple-Bits

Surgical

Strongest attacker

$P(\boldsymbol{hit\ an\ Achilles\ bit}) \approx$ **100%**

Black-box

White-box

~~(Weakest)~~ **Stronger attacker**

$\sum P(\boldsymbol{hit\ an\ Achilles\ bit}) \gg$ **1.3%**

Blind

MARYLAND CYBERSECURITY CENTER

RQ-3: The Weakest Attacker with Rowhammer

- Evaluation
  - **MLaaS scenario**: a VM runs under the Rowhammer pressure
    - A Python process that constantly queries the **VGG16 ImageNet** model
    - Make bit-flips to the process memory: both on the **code** and **data** [**Consequences**: **RAD > 10%**, **process crash**, or RAD <= 10%]
  - Method: **Hammertime**[1] DB
    - Explore Rowhammer effects systematically in **12 different DRAM chips** [**Vulnerability of DRAM**: based on the number of bits subjected to flip]
  - Experiments
    - 25 experiments for each of 12 different DRAM chips
    - 300 cumulative bit-flip attempts for each experiment

[1]Tatar et al., *Defeating Software Mitigations against Rowhammer: a Surgical Precision Hammer,* RAID'18

RQ-3: The Weakest Attacker with Rowhammer

- Blind attack results
  - The attacker can inflict the **Terminal Brain Damage (RAD > 10%)** to the victim model, effectively
    - On average, **62% (15.6/25)** of the experiments were successful
    - With the **most** vulnerable DRAM chip, **96% (24/25)** successes
    - With the **least** vulnerable DRAM chip, **4% (1/25)** successes
  - It is **Challenging to Detect** the blind attacker
    - **Only 6 crashes** observed over the entire 7.5k bit-flip attempts

Blind Rowhammer attack is practical against DNN models

Research Questions

- **RQ-1**: How vulnerable are DNNs to single bit-flips?
- **RQ-2**: What properties influence this vulnerability?
- **RQ-3**: Can an attacker exploit this vulnerability?
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?

RQ-4: Rowhammer Defenses

- Hardware-supported defenses to fault attack
  - ECC: Error correcting code in memory[1]
  - Detection based on hardware performance counters[2]

- System-level defenses to fault attack
  - CATT: Memory isolation of the kernel and user space[3]
  - ZebRAM: Software-based isolation of every DRAM row[4]

They require infrastructure-wide changes, or they are not effective against other hardware faults

[1]Kim et al., *Flipping Bits in Memory without Accessing Them: An Experimental Study of DRAM …,* ACM SIGARCH'14
[2]Aweke et al., *Anvil: Software-based Protection against Next-generation Rowhammer attacks,* ACM SIGPLAN'16
[3]Brasser et al., *Can't Touch This: Software-only Mitigation against Rowhammer Attacks …,* USENIX'17
[4]Konoth et al., *Zebram: Comprehensive and Compatible Software Protection against Rowhammer Attacks,* OSDI'18

RQ-4: Can We Mitigate this Vulnerability?

- Investigate DNN-level defenses:
  - **Restrict activation magnitudes**: Tanh or ReLU6
  - **Use low-precision numbers**: quantization or binarization

RQ-4: Pros and Cons of Our Defenses

- Pros
  - Both the directions **reduce the # of vulnerable parameters**
- Cons
  - Require to **re-train a whole model** from scratch

RQ-4: Pros and Cons of Our Defenses

- Pros
  - Both directions **reduce the # of vulnerable parameters**
  - Substitute activation functions **without re-training**

- Cons
  - Require to **re-train a whole model** from scratch
  - **Expect the accuracy drop** of a model without re-training

Summary of Our Results

- **RQ-1**: How vulnerable are DNNs to single bit-flips?
  All DNNs have a bit whose flip causes RAD up to 100%
  40-50% of all parameters in a model are vulnerable
- **RQ-2**: What properties influence this vulnerability?
  The vulnerability is consistent across multiple DNNs
- **RQ-3**: Can an attacker exploit this vulnerability?
  Blind Rowhammer attacker can exploit this practically
- **RQ-4**: Can we utilize DNN-level mechanisms for mitigation?
  We reduce the vulnerable parameters in a model; but ours degrade the performance or require the re-training

Conclusions and Implications

- DNNs are not resilient to worst-case parameter perturbations
  - Re-examine techniques relying on **graceful degradations** with security lens
- The vulnerability of DNNs to μ-arch. attacks is under-studied
  - Explore and evaluate **new attacks**, particularly thought hard
  - These attacks may be inflicted with **weak attackers**, e.g. blind Rowhammer
- For AI systems, system-level defenses are not sufficient
  - Consider additional **model-level defenses** that account for DNN properties

Thank you!

Sanghyun Hong

shhong@cs.umd.edu

http://hardwarefail.ml