AI Psychiatry: Forensic Investigation of Deep Learning Networks in Memory Images

David Oygenblik, Carter Yagemann, Joseph Zhang, Arianna Mastali, Jeman Park, Brendan Saltaformaggio

Georgia Tech

davido@gatech.edu
davidoygenblik.github.io

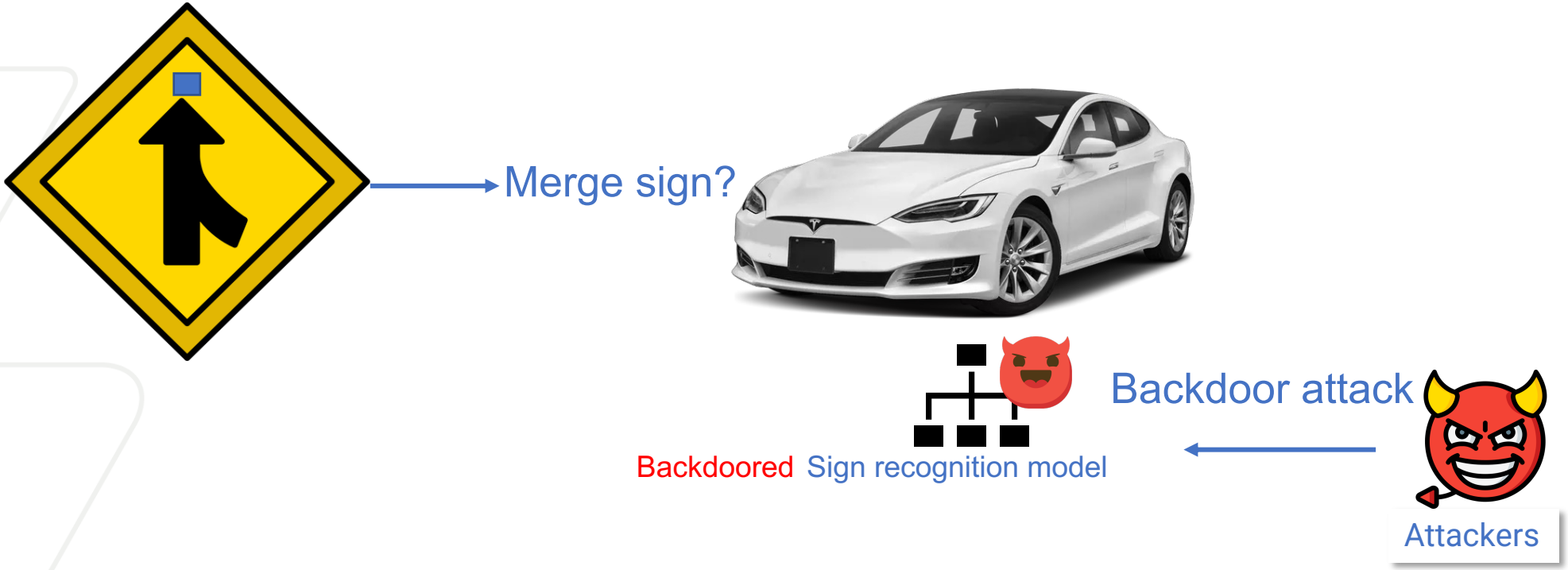
ARTIFACT EVALUATED
usenix ASSOCIATION
AVAILABLE

ARTIFACT EVALUATED
usenix ASSOCIATION
FUNCTIONAL

Georgia Tech Cyber Forensics Innovation Lab

ML Model Investigation

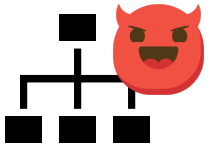
ML Model Investigation



ML Model Investigation

ge sign?

Detective Pika to the rescue!…



Backdoored Sign recognition model

Backdoor attack

Attackers

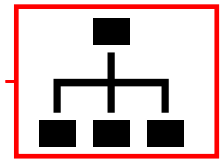
ML Model Investigation

Prone to:

Adversarial examples

Data poisoning attacks

Backdoors

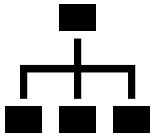


Sign recognition model

Pika's primary suspect!

ML Model Investigation

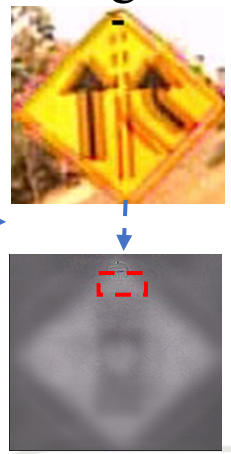
"Lets go DL model vetting tools!"



Sign recognition model

DL model vetting tools

Successful investigation!



Unsolved Challenges…

To Pika's dismay, it isn't that simple…

Sign recognition model

"Lets go DL Model Vetting Tools!"

① Vetting tools assume model access!

Can we get stored copy of the model?

No, model encrypted or inaccessible!

② Cars employ online learning….

Online learning = unique weights

How to get unique car weights?

③ Pika has the model weights, so what?

Vetting tool needs instrumentable model…

How is Pika going to reuse them?

Has Detective Pika Failed?

To Pika's dismay, it isn't that simple…

Sign recognition Model

① Vetting tools assume model access!

Can we get stored copy of the model?

No, model encrypted or inaccessible!

② Cars employ online learning….

Online learning = unique weights

How to get unique car weights?

③ Pika has the model weights, so what?

Vetting tool needs instrumentable model…

How is Pika going to reuse them?

AIP Investigation

– Pika

Model recovery and model rehosting!

GPU memory

CPU memory

AiP

Recover Model Structure

① AiPs model identification and layer recovery!

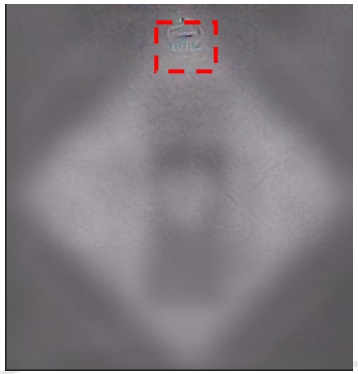
Recover Model Weights

② AiPs feedback driven tensor recovery!

Enable Model Reuse

③ AiPs tensor mapping and model rehosting!

AiP outputs a vet-able Model



How to get the encrypted model?

How to access the weights?

How does Pika reuse the model?

AiP: Model Identification & Layer Attribution

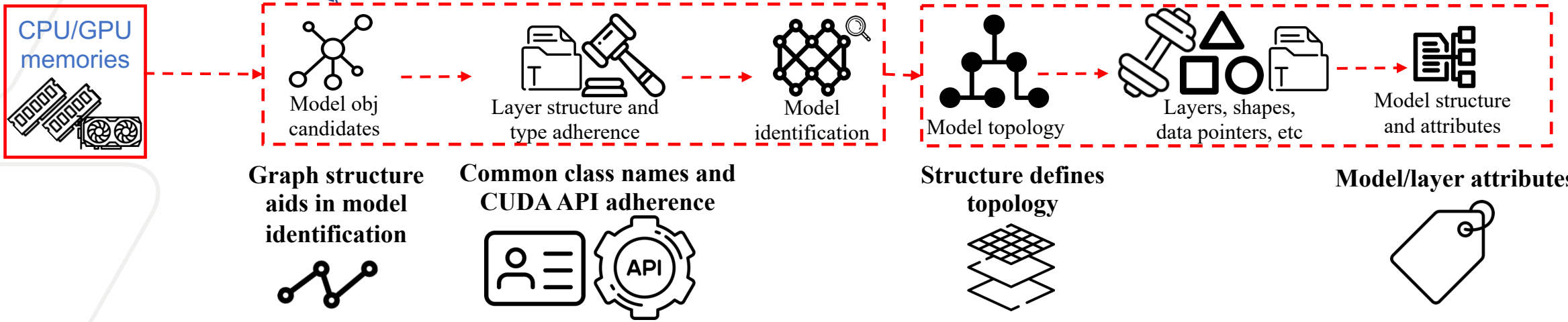
Pika wow!

Model identification and layer attribute collection

AiP

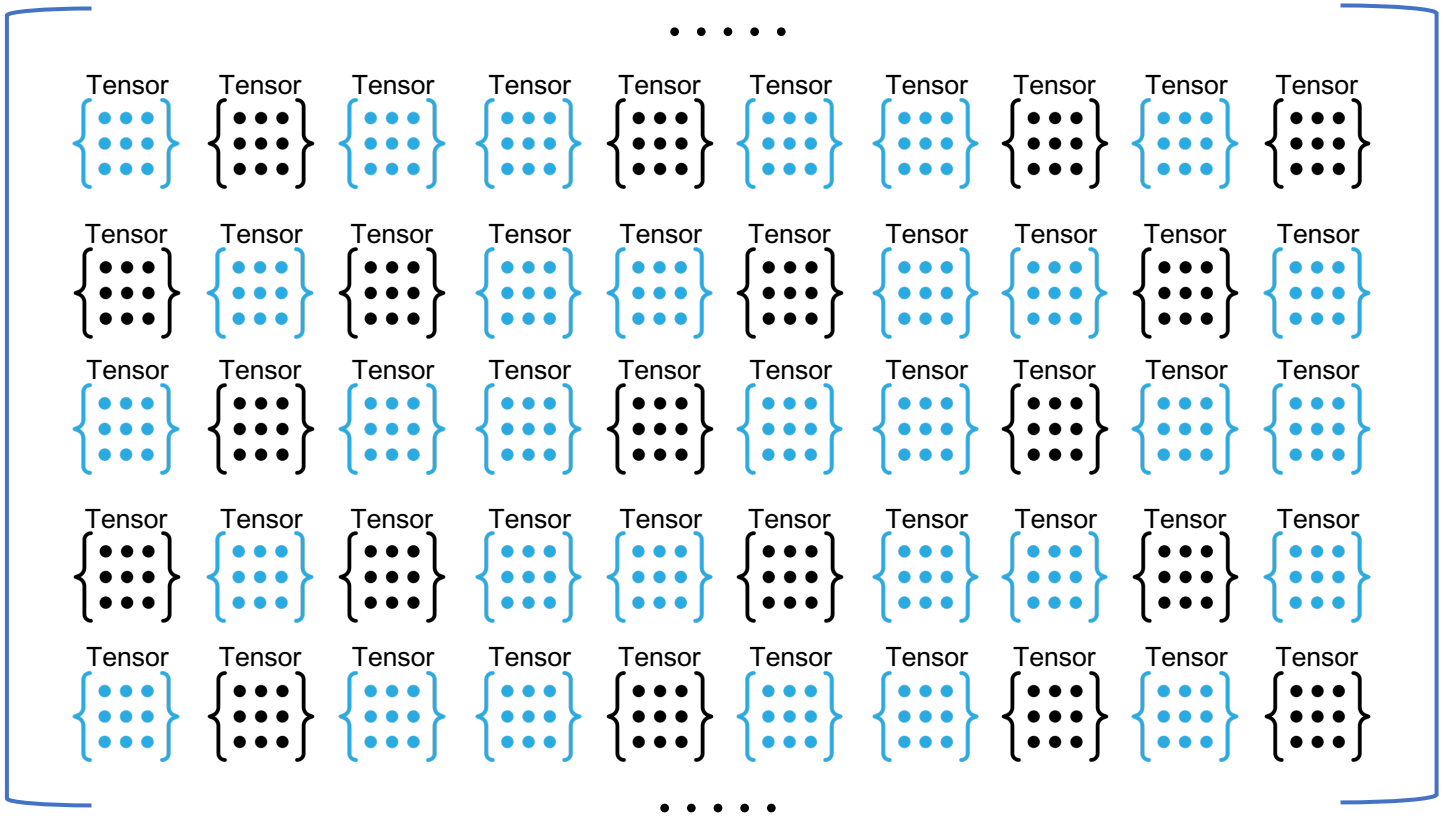
- A Grateful Pika

ML model = directed graph! Search for directed graph structures

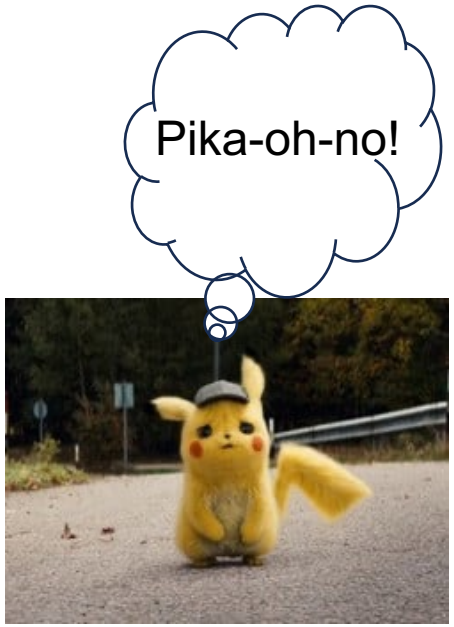


Too many suspects…

Can be **thousands** of tensors in memory!



DL process heap



- A worried pika

A problem!

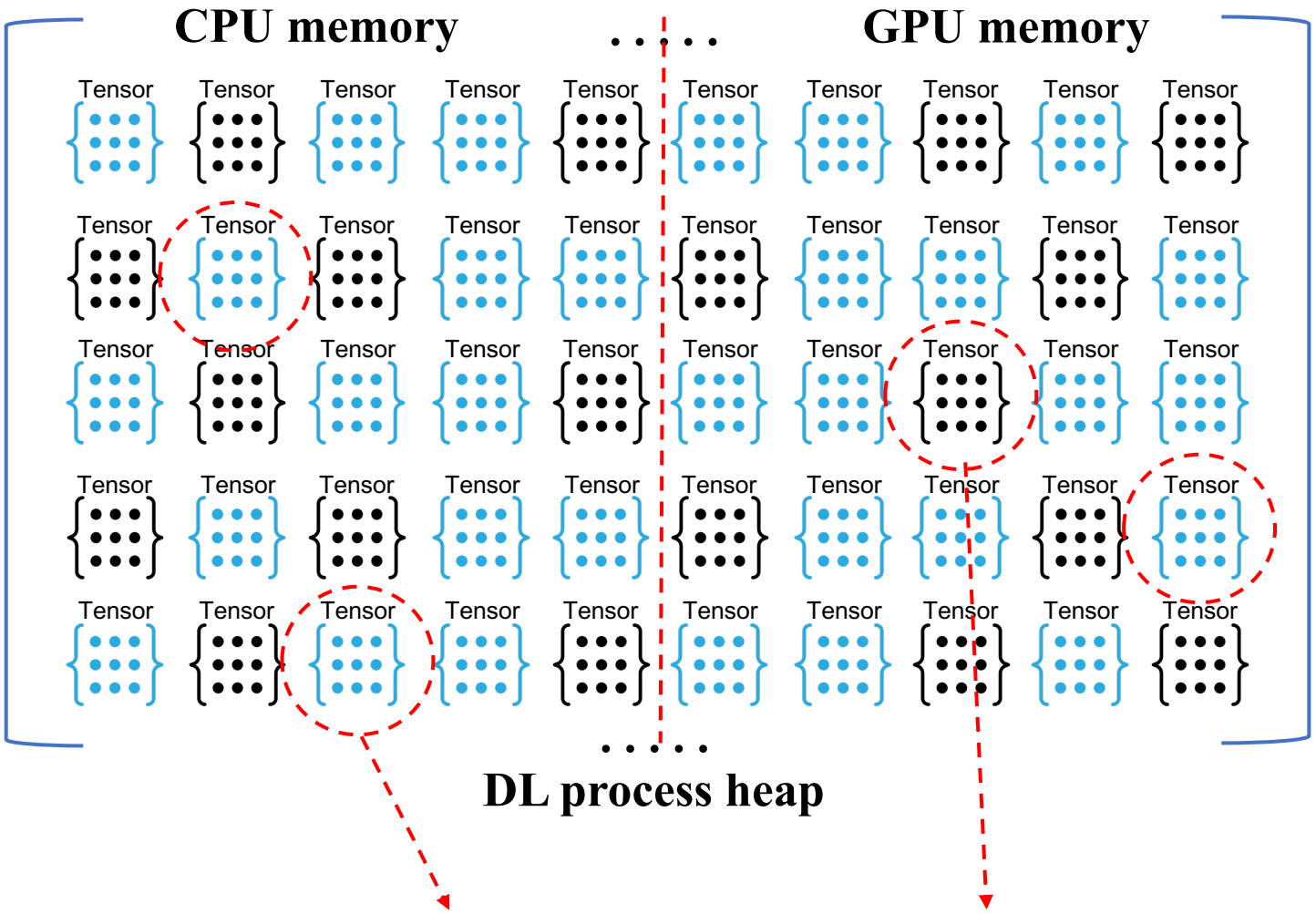
Too many suspects…

Can be **thousands** of tensors in memory!

Tensors with matching attributes can exist!

Optimizers and layer activations exist….

Tensors can be distributed across either the CPU or GPU?

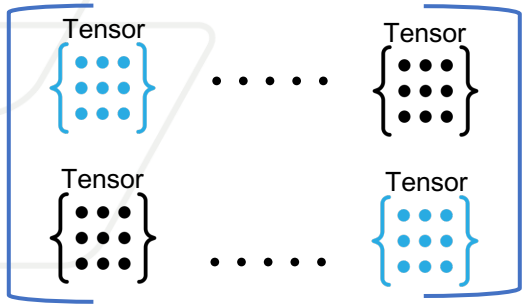


Shape: (3, 32, 32, 64)
Element count: 196,608

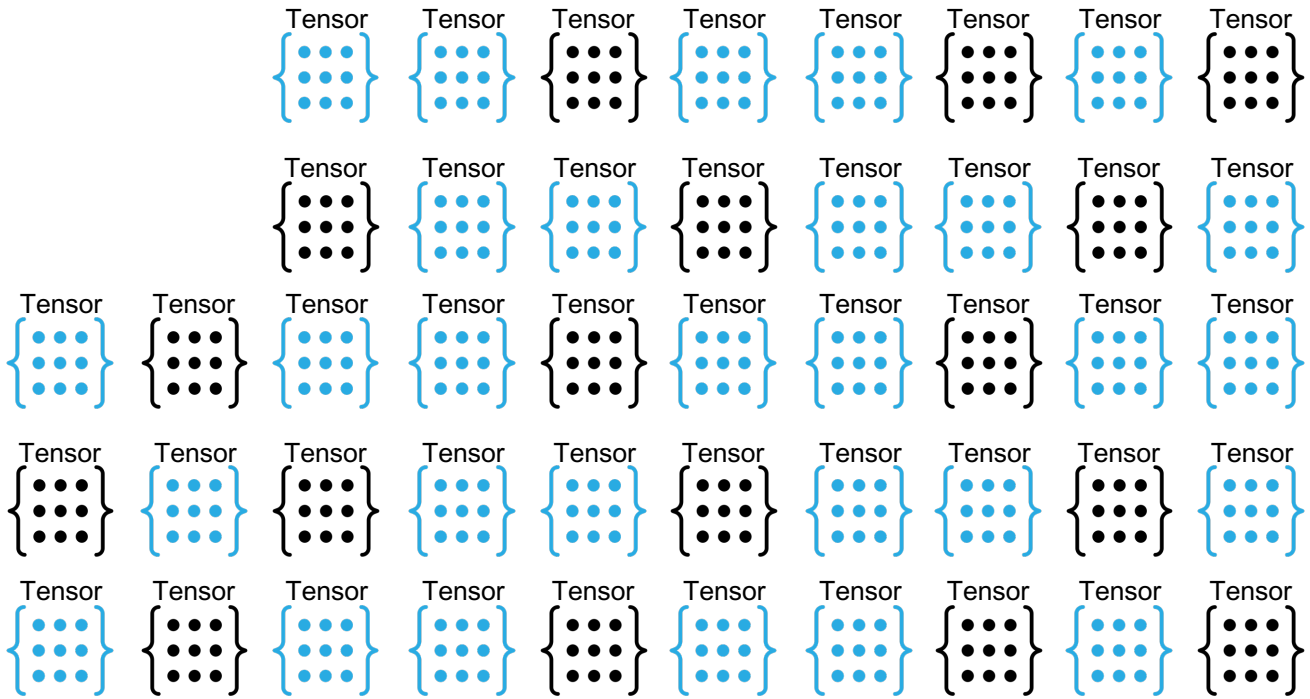
Shape: (3, 32, 32, 64)
Element count: 196,608

- A worried pika

AiP: Feedback Driven Tensor Recovery

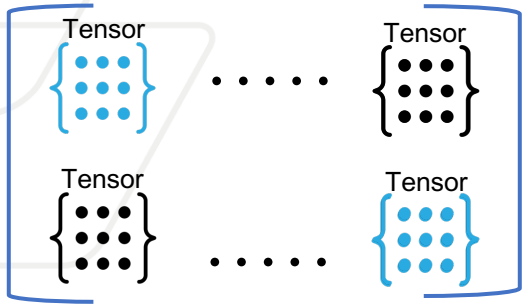


DL process heap



- An Inquisitive Pika

AiP: Feedback Driven Tensor Recovery



DL process heap

Feedback driven tensor recovery!

AiP

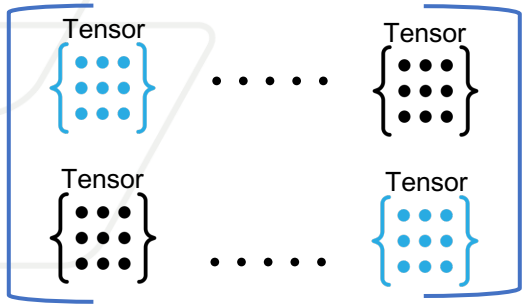
Pika-hmm…

- An inquisitive Pika

Key insights:

- **One model node = Two tensors (i.e. conv = activation + bias)**

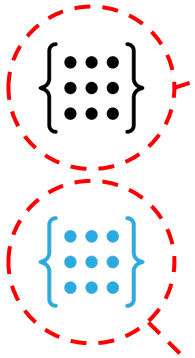
AiP: Feedback Driven Tensor Recovery



DL process heap

Key insights:

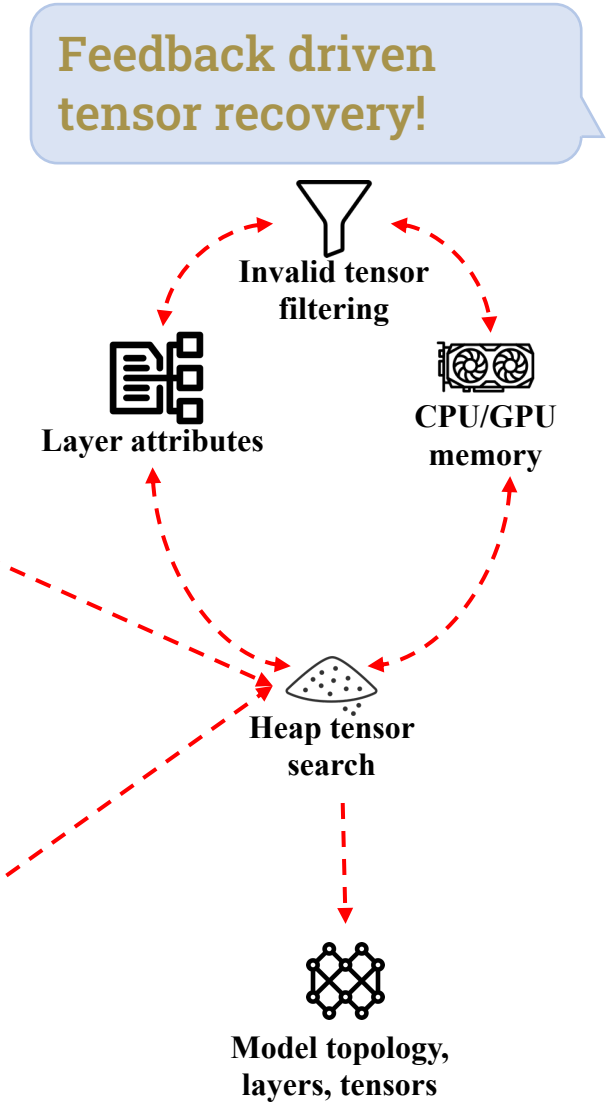
- **One model node = Two tensors (i.e. conv = activation + bias)**
- **"Identical" tensors implies some are not correct**
- **There was an invalid data pointer!**
- **Model recovered!**



Tensor attributes:

Shape: (3, 32, 32, 64)
Element count: 196,608
Ref Count: 1
Data ptr: 0x7fff4def
Device_str: 'gpu'

Shape: (3, 32, 32, 64)
Element Count: 196,608
Ref count: 1
Data ptr: 0x7def4dfe
Device_str: 'gpu'

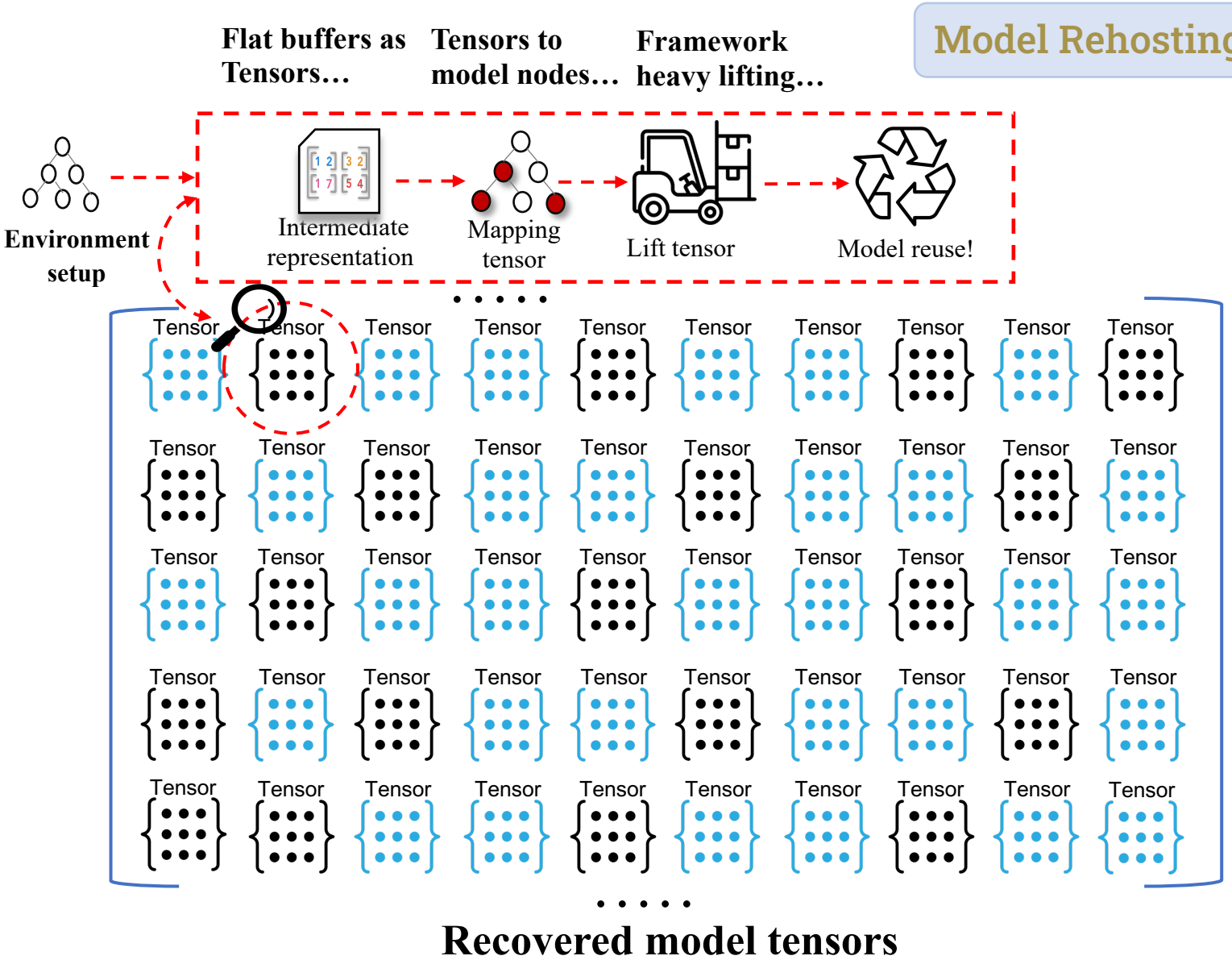


AiP

Pika-hmm…

- An inquisitive Pika

AiP: Model Rehosting and Reuse



Model Rehosting!

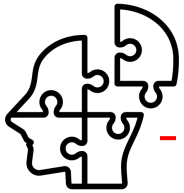
AiP

Pika-hmm…

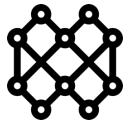
- An inquisitive Pika

Solving the Case!

Pika can finally vet the model!

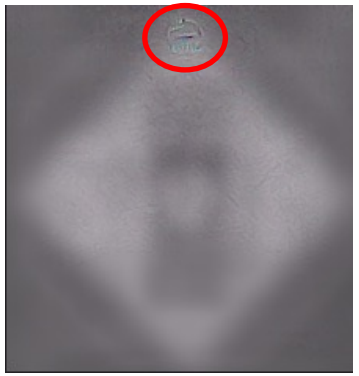


AiP



Recovered model

ML vetting tool



Pika found a model backdoor!!

Thanks AiP!

- A thankful Pika

No worries detective, it was all you.

AiP

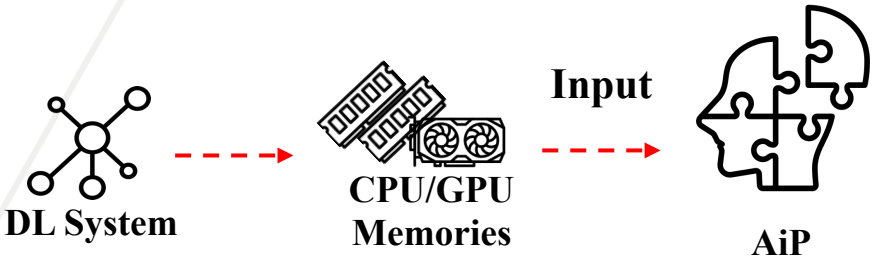
Experiment Setup

Models: 30 total models deployed. 5 Models deployed across 3 versions of PyTorch and TensorFlow (2 frameworks).

Datasets: Models trained on **LISA, CIFAR10, IMDB** datasets.

AiP Input: CPU/GPU memories for each of the 30 model deployments.

Shown here: 10 models from most recent frameworks, more in the paper



AiP Recovery

Model	Framework	Weights Recovered		# GPU Ptrs
		#	% Acc	
Resnet152v1	TensorFlow	94M	100.0	940
SSD-MobileNetV1	TensorFlow	21M	100.0	145
MobileNetV2	TensorFlow	6M	100.0	268
VGG16	TensorFlow	16M	100.0	34
BD-LSTM	TensorFlow	3M	100.0	14
Resnet152v1	PyTorch	60M	100.0	777
MobileNetV1	PyTorch	3M	100.0	137
MobileNetV2	PyTorch	6M	100.0	268
VGG16	PyTorch	16M	100.0	34
BD-LSTM	PyTorch	5M	100.0	11

Model Recovery

Models with upwards of **94 M** weights are recovered with **100%** accuracy.

100% accuracy guaranteed at inference by graph guided recovery.

AiP Recovery

Model	Framework	Weights Recovered		# GPU Ptrs
		#	% Acc	
Resnet152v1	TensorFlow	94M	100.0	940
SSD-MobileNetV1	TensorFlow	21M	100.0	145
MobileNetV2	TensorFlow	6M	100.0	268
VGG16	TensorFlow	16M	100.0	34
BD-LSTM	TensorFlow	3M	100.0	14
Resnet152v1	PyTorch	60M	100.0	777
MobileNetV1	PyTorch	3M	100.0	137
MobileNetV2	PyTorch	6M	100.0	268
VGG16	PyTorch	16M	100.0	34
BD-LSTM	PyTorch	5M	100.0	11

Model Recovery

Models with upwards of **94 M** weights are recovered with **100%** accuracy.

100% accuracy guaranteed at inference by graph guided recovery.

GPU tensors (as many as **940** and as low as **11**) recovered successfully.

AiP Recovery

Model	Framework	Weights Recovered		# GPU Ptrs
		#	% Acc	
Resnet152v1	TensorFlow	94M	100.0	940
SSD-MobileNetV1	TensorFlow	21M	100.0	145
MobileNetV2	TensorFlow	6M	100.0	268
VGG16	TensorFlow	16M	100.0	34
BD-LSTM	TensorFlow	3M	100.0	14
Resnet152v1	PyTorch	60M	100.0	777
MobileNetV1	PyTorch	3M	100.0	137
MobileNetV2	PyTorch	6M	100.0	268
VGG16	PyTorch	16M	100.0	34
BD-LSTM	PyTorch	5M	100.0	11

Model Recovery

Models with upwards of **94 M** weights are recovered with **100%** accuracy.

100% accuracy guaranteed at inference by graph guided recovery.

GPU tensors (as many as **940** and as low as **11**) recovered successfully.

Models from multiple application domains (containing different operations types) are recovered.

AiP Recovery

Model	Framework	Weights Recovered		# GPU Ptrs
		#	% Acc	
Resnet152v1	TensorFlow	94M	100.0	940
SSD-MobileNetV1	TensorFlow	21M	100.0	145
MobileNetV2	TensorFlow	6M	100.0	268
VGG16	TensorFlow	16M	100.0	34
BD-LSTM	TensorFlow	3M	100.0	14
Resnet152v1	PyTorch	60M	100.0	777
MobileNetV1	PyTorch	3M	100.0	137
MobileNetV2	PyTorch	6M	100.0	268
VGG16	PyTorch	16M	100.0	34
BD-LSTM	PyTorch	5M	100.0	11

Model Rehosting

Rehosting: Tested Model = Deployed Model

AiP Rehosting

Model	Framework	# Layers Rehosted	% Accuracy	
			Deployed	Rehosted
Resnet152v1	TensorFlow	3	97.3	97.3
SSD-MobileNetV1	TensorFlow	4	97.9	97.9
MobileNetV2	TensorFlow	4	82.6	82.6
VGG16	TensorFlow	2	72.1	72.1
BD-LSTM	TensorFlow	3	84.2	84.2
Resnet152v1	PyTorch	3	97.2	97.2
MobileNetV1	PyTorch	4	98.5	98.5
MobileNetV2	PyTorch	4	64.1	64.1
VGG16	PyTorch	2	66.5	66.5
LSTM	PyTorch	3	79.5	79.5

Model Rehosting

Rehosting: Tested Model = Deployed Model

The accuracy for the deployed and AiP rehosted models are the same indicating successful rehosting!

AiP Rehosting

Model	Framework	# Layers Rehosted	% Accuracy	
			Deployed	Rehosted
Resnet152v1	TensorFlow	3	97.3	97.3
SSD-MobileNetV1	TensorFlow	4	97.9	97.9
MobileNetV2	TensorFlow	4	82.6	82.6
VGG16	TensorFlow	2	72.1	72.1
BD-LSTM	TensorFlow	3	84.2	84.2
Resnet152v1	PyTorch	3	97.2	97.2
MobileNetV1	PyTorch	4	98.5	98.5
MobileNetV2	PyTorch	4	64.1	64.1
VGG16	PyTorch	2	66.5	66.5
LSTM	PyTorch	3	79.5	79.5

Model Rehosting

Rehosting: Tested Model = Deployed Model

The accuracy for the deployed and AiP rehosted models are the same indicating successful rehosting!

Number of layer types rehosted for each model are the same (even across frameworks!).

AiP Rehosting

Model	Framework	# Layers Rehosted	% Accuracy	
			Deployed	Rehosted
Resnet152v1	TensorFlow	3	97.3	97.3
SSD-MobileNetV1	TensorFlow	4	97.9	97.9
MobileNetV2	TensorFlow	4	82.6	82.6
VGG16	TensorFlow	2	72.1	72.1
BD-LSTM	TensorFlow	3	84.2	84.2
Resnet152v1	PyTorch	3	97.2	97.2
MobileNetV1	PyTorch	4	98.5	98.5
MobileNetV2	PyTorch	4	64.1	64.1
VGG16	PyTorch	2	66.5	66.5
LSTM	PyTorch	3	79.5	79.5

Much More in the Paper!

- Forensics during online-learning
- Investigation of backdoored models
- Comparison to black-box approaches

Many thanks!

Georgia Tech Research Institute

AI Psychiatry: Forensic Investigation of Deep Learning Networks in Memory Images

Oygenblik, D., Yagemann, C., Zhang, J., Mastali, A., Park, J., Saltaformaggio, B.

USENIX, 2024

https://github.com/CyFI-Lab-Public/AIP.git

Thank you!

Questions? ☺

Georgia Tech Cyber Forensics Innovation Lab

David Oygenblik
davido@gatech.edu
davidoygenblik.github.io

Georgia Tech