Beyond the Bridge: Contention-Based Covert and Side Channel Attacks on Multi-GPU Interconnect

Yicheng Zhang[1], Ravan Nazaraliyev[1], Sankha Baran Dutta[2], Nael Abu-Ghazaleh[1], Andres Marquez[2], Kevin Barker[2]

yzhan846@ucr.edu

[1]*University of California, Riverside*

[2]*Pacific Northwest National Laboratory*

UC RIVERSIDE

Pacific Northwest NATIONAL LABORATORY

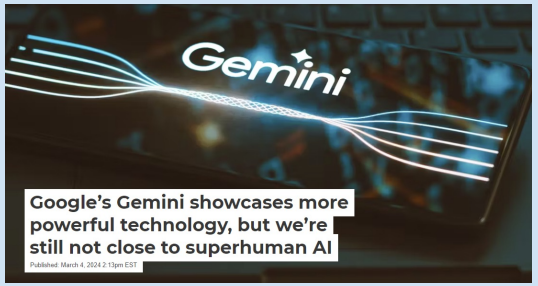
Multi-GPU Systems

- Multi-GPU Systems: Widely used across various fields.

Large Language Model

ChatGPT

Google's Gemini



Compute Graphics

Computer-generated imagery (CGI)

Computer Graphic Arts

Autonomous Vehicle

Tesla

Cybertruck Fails Are a Daily Delight to the Haters

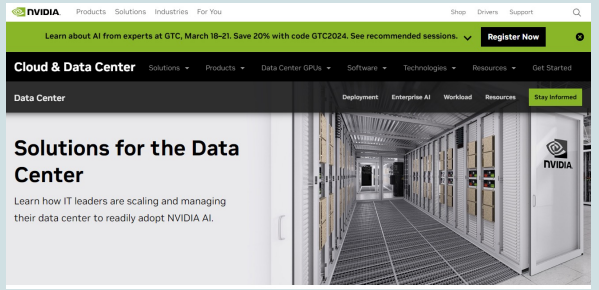
Wild crashes, malfunctions and ill-advised off-roading have made Elon Musk's steel-paneled Tesla a symbol of vehicular folly

BY MILES KLEE MARCH 7, 2024

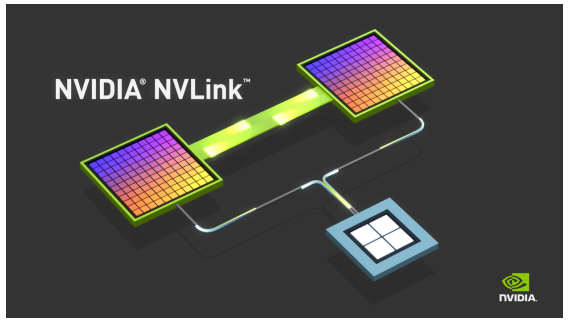
Data Centers

Google Cloud Platform

NVIDIA



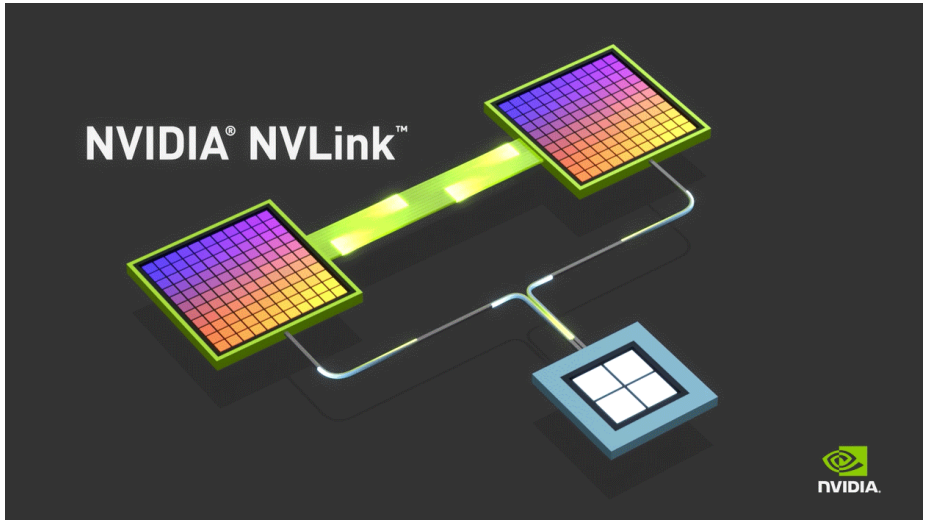
Outline



- <u>Background: Multi-GPU interconnect.</u>
- Threat model and leakage vectors.
- Cross-GPU covert channel attacks.
- Cross-GPU side channel attacks.
- Mitigation.

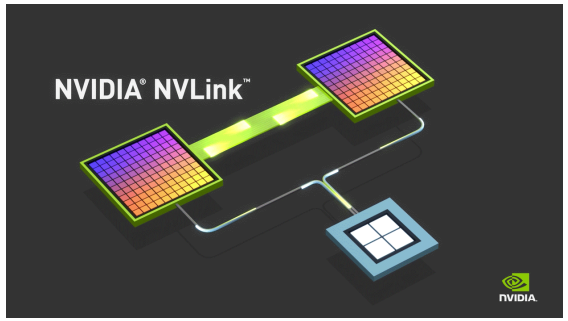
Background: Multi-GPU interconnect

- **NVLink**: High-speed, high-bandwidth interconnect by NVIDIA.
- **Direct Links**: Supports CPU-to-GPU and GPU-to-GPU connections.
- **Bidirectional**: Each link has two sublinks, one for each direction.
- **PCIe**: A serial expansion bus standard for connecting a computer to one or more peripheral devices.



https://www.nvidia.com/en-us/design-visualization/nvlink-bridges/

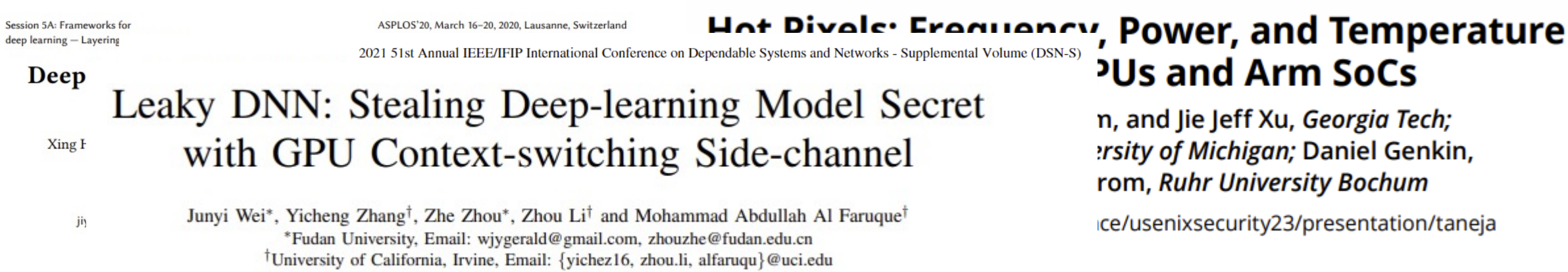
Outline



- Background: Multi-GPU interconnect.
- Threat model and leakage vectors.
- Cross-GPU covert channel attacks.
- Cross-GPU side channel attacks.
- Mitigation.

Known Side-channel Attacks on GPU

- Previous GPU attacks focused on a single GPU.
  - This required the **co-location** of the victim and the spy on the same GPU.
- "Spy in the GPU-box" [ISCA'23] demonstrated a prime and probe attack on remote GPU's l2 cache.
  - But they did not explore the interconnects between GPUs.


Session 5A: Frameworks for deep learning – Layering

ASPLOS'20, March 16–20, 2020, Lausanne, Switzerland

2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)

Deep

Leaky DNN: Stealing Deep-learning Model Secret with GPU Context-switching Side-channel

Junyi Wei*, Yicheng Zhang†, Zhe Zhou*, Zhou Li† and Mohammad Abdullah Al Faruque†

*Fudan University, Email: wjygerald@gmail.com, zhouzhe@fudan.edu.cn

†University of California, Irvine, Email: {yichez16, zhou.li, alfaruqu}@uci.edu

Hot Pixels: Frequency, Power, and Temperature

PUs and Arm SoCs

n, and Jie Jeff Xu, *Georgia Tech;*

rsity of Michigan; Daniel Genkin,

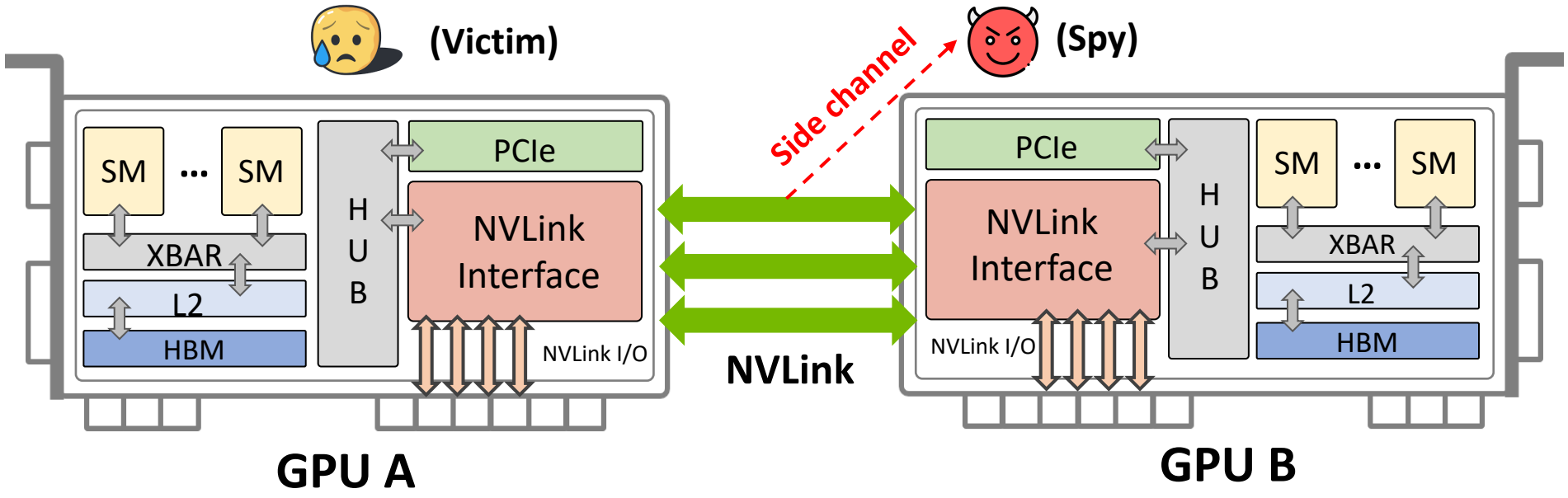
rom, *Ruhr University Bochum*

ce/usenixsecurity23/presentation/taneja

[ISCA'23] Dutta, Sankha Baran, et al. "Spy in the GPU-box: Covert and side channel attacks on multi-GPU systems." Proceedings of the 50th Annual International Symposium on Computer Architecture. 2023.

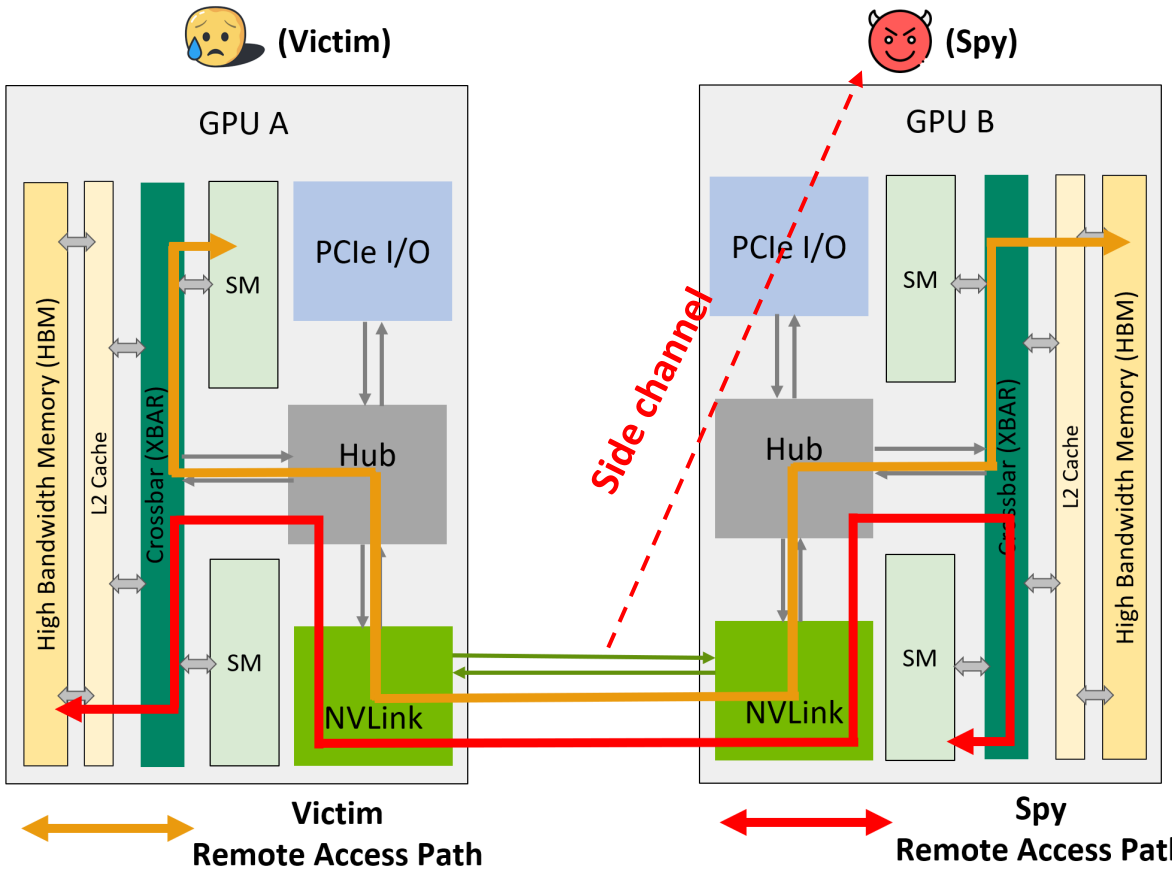
Threat model

- No need for co-location.



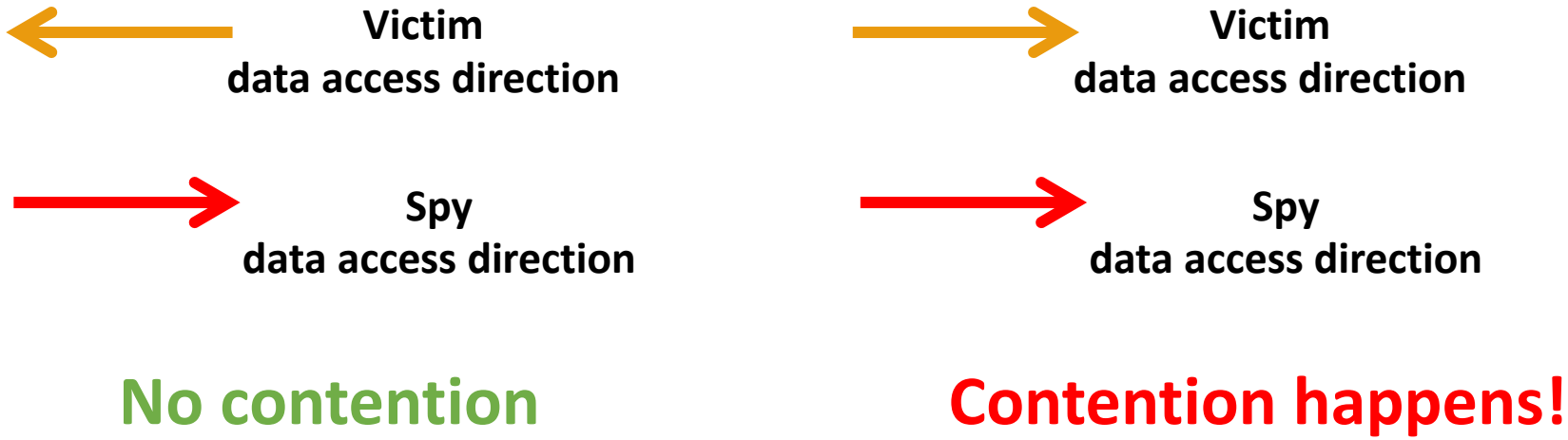
Leakage Vectors: Contention-based

- Contention on a shared NVLink can lead to an increase in data transfer.



Leakage Vectors: Contention-based

- Contention measurement on NVLink.
  - Contention direction influence.



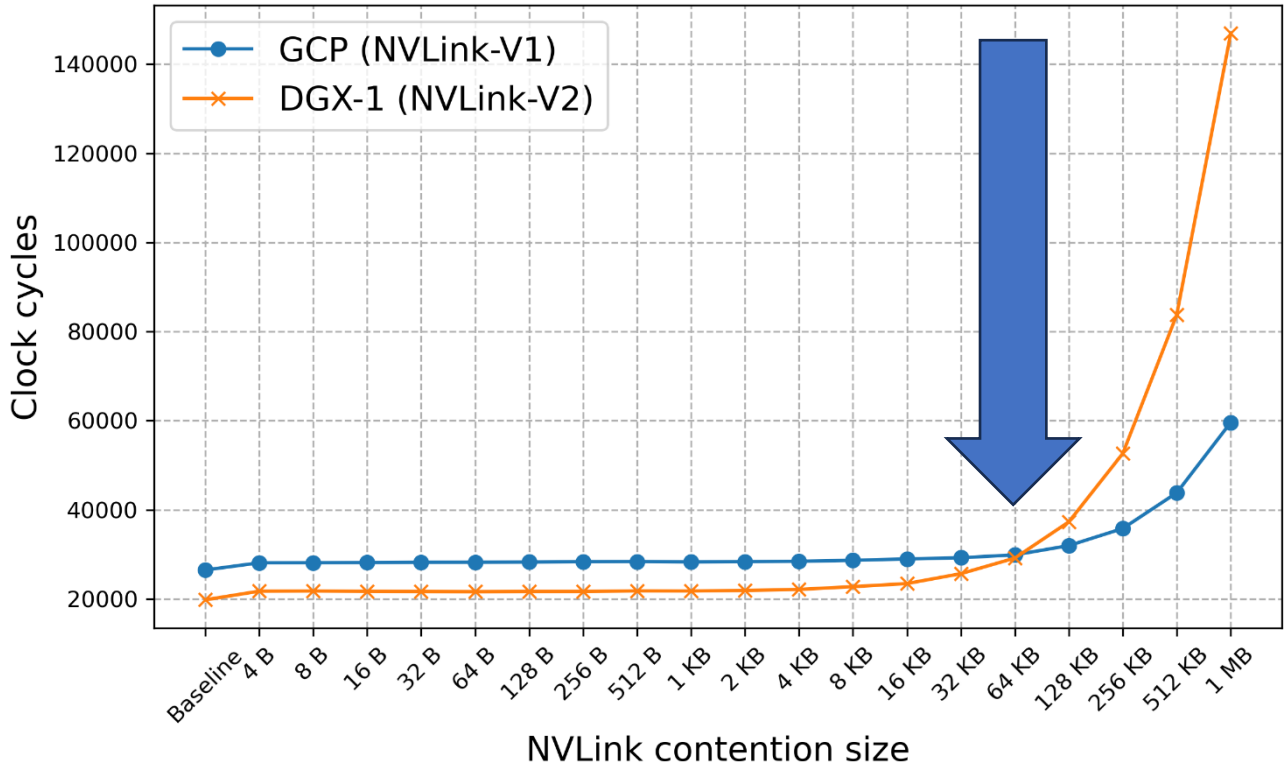
Leakage Vectors: Contention-based

- Contention measurement on NVLink.
  - Contention direction influence.
  - Contention size influence.

Victim
data access direction

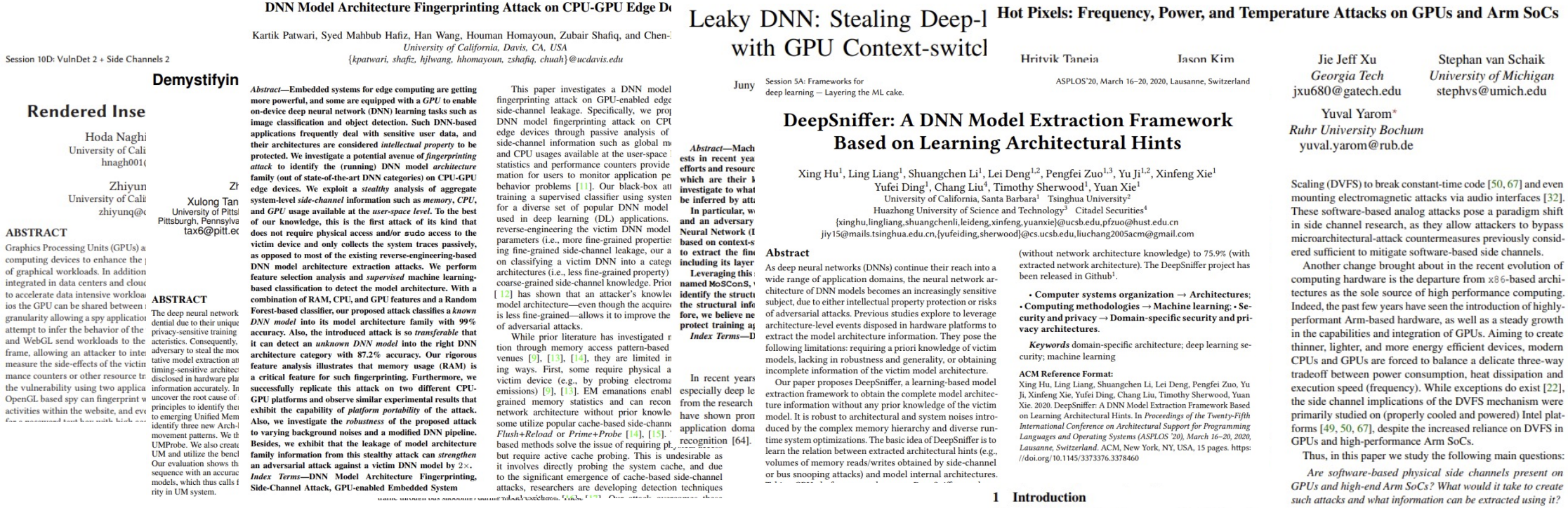
Spy
data access direction

Contention happens!



Leakage Vectors: Leaky Counter-based

- Prior work exploits GPU performance counters as side channel leakages.
- **<u>NVLink-related Performance Counters.</u>**



Session 10D: VulnDet 2 + Side Channels 2

Demystifyin

Rendered Inse

Hoda Naghi
University of Cali
hnagh001(

Zhiyun
University of Cali
zhiyunq@(

ABSTRACT

Graphics Processing Units (GPUs) a
computing devices to enhance the
of graphical workloads. In addition
integrated in data centers and cloud
to accelerate data intensive workloa
ios the GPU can be shared between
granularity allowing a spy applicatio
attempt to infer the behavior of the
and WebGL send workloads to the
frame, allowing an attacker to inte
measure the side-effects of the victi
mance counters or other resource tr
the vulnerability using two applica
OpenGL based spy can fingerprint w
activities within the website, and eve

Zh
Xulong Tan
University of Pitts
Pittsburgh, Pennsylva
tax6@pitt.ec

ABSTRACT

The deep neural network
dential due to their unique
privacy-sensitive training
acteristics. Consequently,
adversary to steal the mo
tative model extraction at
timing-sensitive architec
disclosed in hardware pla
information accurately. In
uncover the root cause of
principles to identify the
to emerging Unified Mem
identify three new Arch-
movement patterns. We th
UMProbe. We also create
UM and utilize the benc
Our evaluation shows th
sequence with an accurac
models, which thus calls
rity in UM system.

DNN Model Architecture Fingerprinting Attack on CPU-GPU Edge D

Kartik Patwari, Syed Mahbub Hafiz, Han Wang, Houman Homayoun, Zubair Shafiq, and Chen-
University of California, Davis, CA, USA
{*kpatwari, shafiz, hjlwang, hhomayoun, zshafiq, chuah*}*@ucdavis.edu*

Abstract*—Embedded systems for edge computing are getting more powerful, and some are equipped with a *GPU* to enable on-device deep neural network (DNN) learning tasks such as image classification and object detection. Such DNN-based applications frequently deal with sensitive user data, and their architectures are considered *intellectual property* to be protected. We investigate a potential avenue of *fingerprinting attack* to identify the (running) DNN model *architecture* family (out of state-of-the-art DNN categories) on CPU-GPU edge devices. We exploit a *stealthy* analysis of aggregate system-level *side-channel* information such as *memory*, *CPU*, and *GPU* usage available at the *user-space level*. To the best of our knowledge, this is the first attack of its kind that does not require physical access and/or `sudo` access to the victim device and only collects the system traces passively, as opposed to most of the existing reverse-engineering-based DNN model architecture extraction attacks. We perform feature selection analysis and *supervised* machine learning-based classification to detect the model architecture. With a combination of RAM, CPU, and GPU features and a Random Forest-based classifier, our proposed attack classifies a *known DNN model* into its model architecture family with 99% accuracy. Also, the introduced attack is so *transferable* that it can detect an *unknown DNN model* into the right DNN architecture category with 87.2% accuracy. Our rigorous feature analysis illustrates that memory usage (RAM) is a critical feature for such fingerprinting. Furthermore, we successfully replicate this attack on two different CPU-GPU platforms and observe similar experimental results that exhibit the capability of *platform portability* of the attack. Also, we investigate the *robustness* of the proposed attack to varying background noises and a modified DNN pipeline. Besides, we exhibit that the leakage of model architecture family information from this stealthy attack can *strengthen* an adversarial attack against a victim DNN model by 2×.

***Index Terms*—DNN Model Architecture Fingerprinting, Side-Channel Attack, GPU-enabled Embedded System**

This paper investigates a DNN model fingerprinting attack on GPU-enabled edge side-channel leakage. Specifically, we pro DNN model fingerprinting attack on CPU edge devices through passive analysis of side-channel information such as global m and CPU usages available at the user-space statistics and performance counters provide mation for users to monitor application pe behavior problems [11]. Our black-box att training a supervised classifier using systen for a diverse set of popular DNN model used in deep learning (DL) applications. parameters (i.e., more fine-grained properties ing fine-grained side-channel leakage, our a on classifying a victim DNN into a categ architectures (i.e., less fine-grained property) coarse-grained side-channel knowledge. Prior [12] has shown that an attacker's knowle model architecture—even though the acquire is less fine-grained—allows it to improve the of adversarial attacks.

While prior literature has investigated tion through memory access pattern-based venues [9], [13], [14], they are limited in ing ways. First, some require physical a victim device (e.g., by probing electroma emissions) [9], [13]. EM emanations enabl grained memory statistics and can recon network architecture without prior knowle some utilize popular cache-based side-channe *Flush+Reload* or *Prime+Probe* [14], [15]. based methods solve the issue of requiring ph but require active cache probing. This is undesirable as it involves directly probing the system cache, and due to the significant emergence of cache-based side-channel attacks, researchers are developing detection techniques

Leaky DNN: Stealing Deep-l
with GPU Context-switcl

Juny

Session 5A: Frameworks for
deep learning — Layering the ML cake.

***Abstract*—Mach**
ests in recent yea
efforts and resourc
which are their
investigate to what
be inferred by att
In particular,
and an adversary
Neural Network (
based on context-s
to extract the fin
including its layer
Leveraging this
named MoSConS,
identify the structu
the structural info
fore, we believe ne
protect training a
***Index Terms*—D**

In recent years
especially deep le
from the research
have shown prom
application doma
recognition [64].

Hot Pixels: Frequency, Power, and Temperature Attacks on GPUs and Arm SoCs

Hritvik Taneja Jason Kim

ASPLOS'20, March 16–20, 2020, Lausanne, Switzerland

Jie Jeff Xu
Georgia Tech
jxu680@gatech.edu

Stephan van Schaik
University of Michigan
stephvs@umich.edu

Yuval Yarom*
Ruhr University Bochum
yuval.yarom@rub.de

Scaling (DVFS) to break constant-time code [50, 67] and even mounting electromagnetic attacks via audio interfaces [32]. These software-based analog attacks pose a paradigm shift in side channel research, as they allow attackers to bypass microarchitectural-attack countermeasures previously considered sufficient to mitigate software-based side channels.

Another change brought about in the recent evolution of computing hardware is the departure from `x86`-based architectures as the sole source of high performance computing. Indeed, the past few years have seen the introduction of highly-performant Arm-based hardware, as well as a steady growth in the capabilities and integration of GPUs. Aiming to create thinner, lighter, and more energy efficient devices, modern CPUs and GPUs are forced to balance a delicate three-way tradeoff between power consumption, heat dissipation and execution speed (frequency). While exceptions do exist [22], the side channel implications of the DVFS mechanism were primarily studied on (properly cooled and powered) Intel platforms [49, 50, 67], despite the increased reliance on DVFS in GPUs and high-performance Arm SoCs.

Thus, in this paper we study the following main questions:

Are software-based physical side channels present on GPUs and high-end Arm SoCs? What would it take to create such attacks and what information can be extracted using it?

DeepSniffer: A DNN Model Extraction Framework Based on Learning Architectural Hints

Xing Hu[1], Ling Liang[1], Shuangchen Li[1], Lei Deng[1,2], Pengfei Zuo[1,3], Yu Ji[1,2], Xinfeng Xie[1]
Yufei Ding[1], Chang Liu[4], Timothy Sherwood[1], Yuan Xie[1]
University of California, Santa Barbara[1] Tsinghua University[2]
Huazhong University of Science and Technology[3] Citadel Securities[4]
{xinghu,lingliang,shuangchenli,leideng,xinfeng,yuanxie}@ucsb.edu,pfzuo@hust.edu.cn
jiy15@mails.tsinghua.edu.cn,{yufeiding,sherwood}@cs.ucsb.edu,liuchang2005acm@gmail.com

Abstract

As deep neural networks (DNNs) continue their reach into a wide range of application domains, the neural network architecture of DNN models becomes an increasingly sensitive subject, due to either intellectual property protection or risks of adversarial attacks. Previous studies explore to leverage architecture-level events disposed in hardware platforms to extract the model architecture information. They pose the following limitations: requiring a priori knowledge of victim models, lacking in robustness and generality, or obtaining incomplete information of the victim model architecture.

Our paper proposes DeepSniffer, a learning-based model extraction framework to obtain the complete model architecture information without any prior knowledge of the victim model. It is robust to architectural and system noises introduced by the complex memory hierarchy and diverse runtime system optimizations. The basic idea of DeepSniffer is to learn the relation between extracted architectural hints (e.g., volumes of memory reads/writes obtained by side-channel or bus snooping attacks) and model internal architectures. (without network architecture knowledge) to 75.9% (with extracted network architecture). The DeepSniffer project has been released in Github[1].

• Computer systems organization → Architectures; • Computing methodologies → Machine learning; • Security and privacy → Domain-specific security and privacy architectures.

Keywords domain-specific architecture; deep learning security; machine learning

ACM Reference Format:
Xing Hu, Ling Liang, Shuangchen Li, Lei Deng, Pengfei Zuo, Yu Ji, Xinfeng Xie, Yufei Ding, Chang Liu, Timothy Sherwood, Yuan Xie. 2020. DeepSniffer: A DNN Model Extraction Framework Based on Learning Architectural Hints. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20), March 16–20, 2020, Lausanne, Switzerland.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3373376.3378460

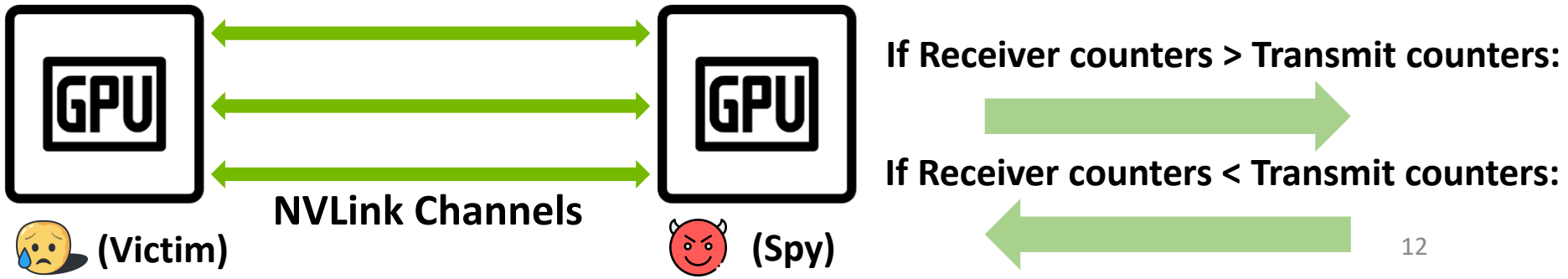
1 Introduction

Leakage Vectors: Leaky Counter-based

- NVLink counters.

Category	Counter Name
Throughput	nvlink_receive/transmit_throughput
User	nvlink_user_data_received/transmitted, nvlink_user_write_data_transmitted, nvlink_user_response_data_received
Total	nvlink_total_data_received/transmitted, nvlink_total_response_data_received, nvlink_total_write_data_transmitted
Atomic operation	nvlink_total/user_nratom_data_transmitted, nvlink_total/user_ratom_data_transmitted

- Observation 1: The NVLink receive/transmit attributes reveal NVLink data transaction direction.

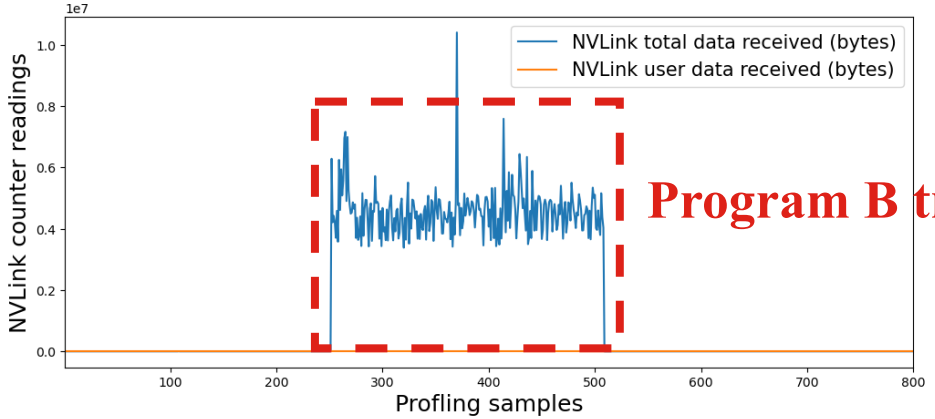


Leakage Vectors: Leaky Counter-based

- User vs Total.

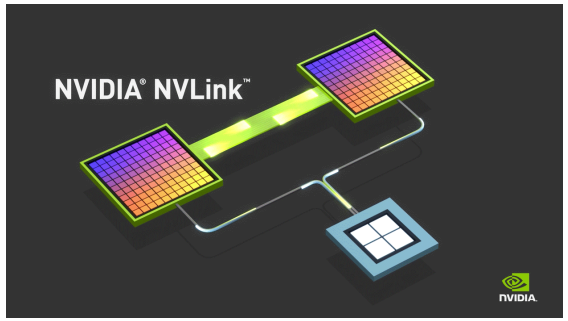
Category	Counter Name
Throughput	nvlink_receive/transmit_throughput
User	nvlink_user_data_received/transmitted, nvlink_user_write_data_transmitted, nvlink_user_response_data_received
Total	nvlink_total_data_received/transmitted, nvlink_total_response_data_received, nvlink_total_write_data_transmitted
Atomic operation	nvlink_total/user_nratom_data_transmitted, nvlink_total/user_ratom_data_transmitted

- Observation 2: When NVLink is shared, NVLink total counters reveal all NVLink data transaction patterns.



Program B transfers 1 MB data via NVLink.

Outline



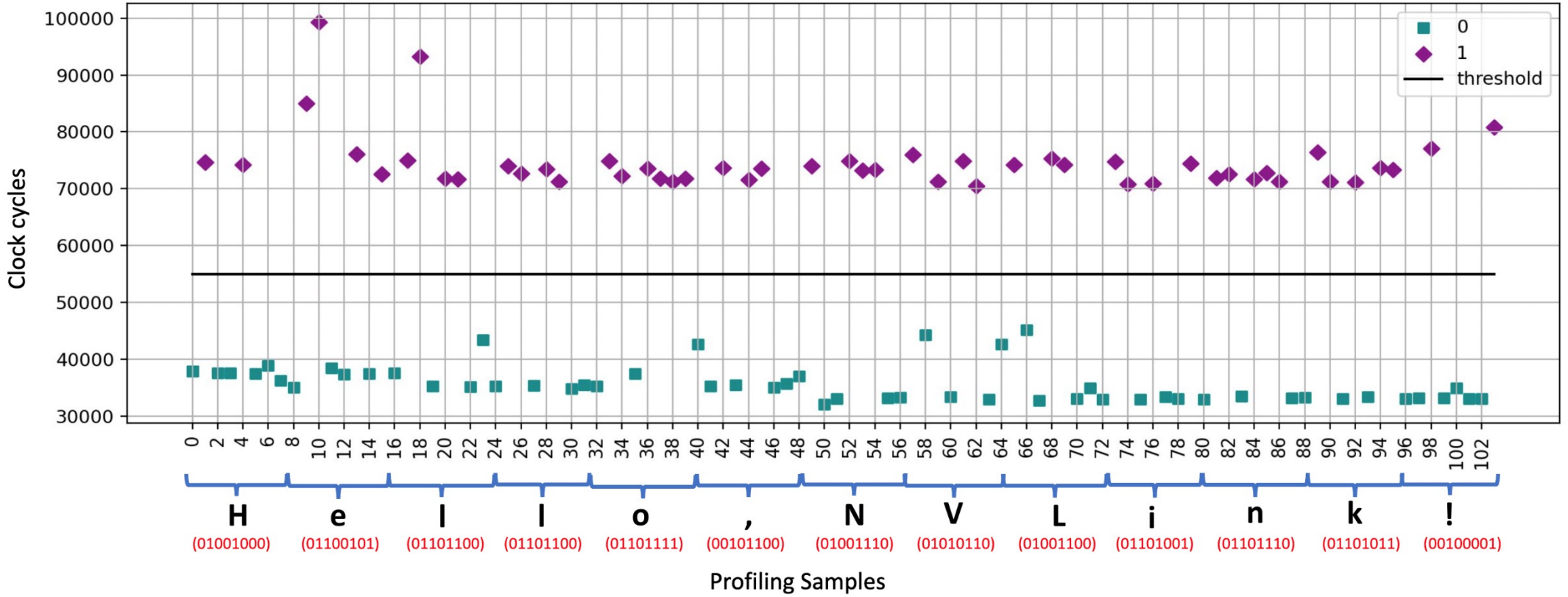
- Background: Multi-GPU interconnect.
- Threat model and leakage vectors.
- <u>Cross-GPU covert channel attacks.</u>
- Cross-GPU side channel attacks.
- Mitigation.

Cross-GPU Covert Channel Attacks

- Sender and receiver share NVLink interconnect.
  - To signal bit ‘1’:
    - Sender transfers data via NVLink to force congestion.
  - To signal bit ‘0’:
    - Sender idles for a pre-defined duration.

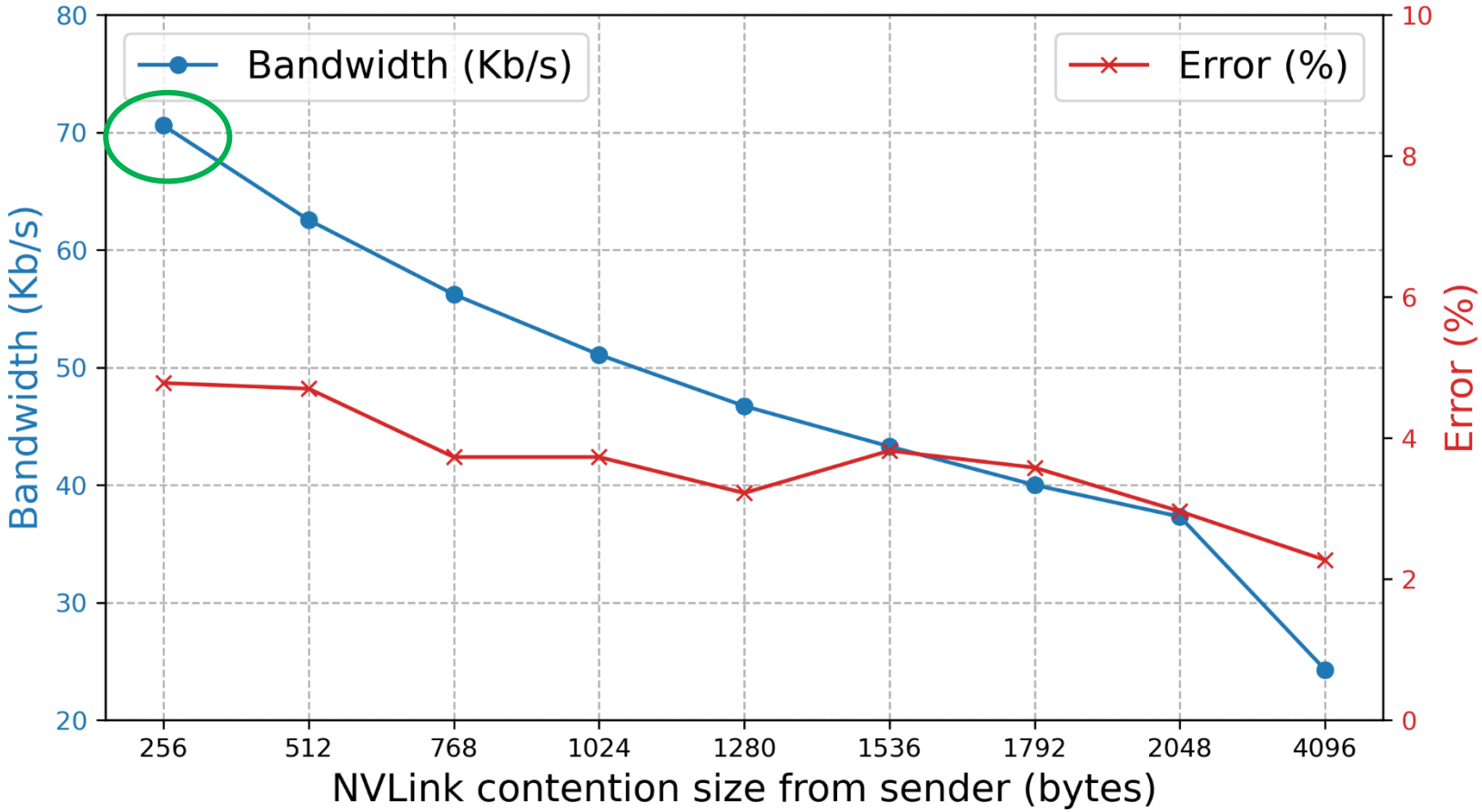
Cross-GPU Covert Channel Attacks

- Covert message (“Hello,NVLink!”).

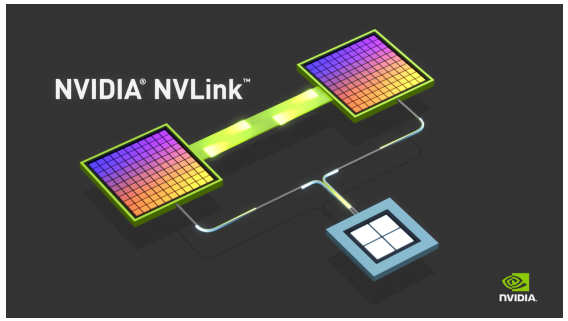


Cross-GPU Covert Channel Attacks

- Bandwidth and error rate.



Outline



- Background: Multi-GPU interconnect.
- Threat model and leakage vectors.
- Cross-GPU covert channel attacks.
- Cross-GPU side channel attacks.
- Mitigation.

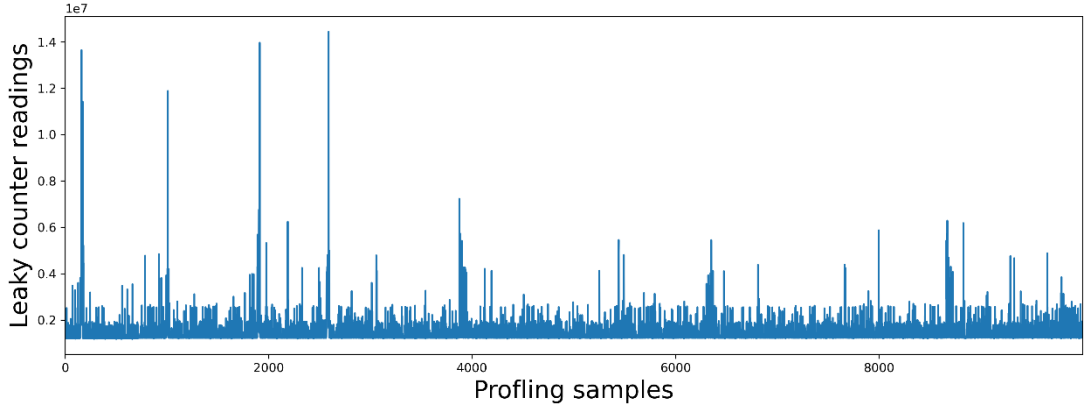
Cross-GPU Side Channel Attacks

- Attack 1: Application Fingerprinting.
- **Victim:** conducts her application across multi-GPU systems.
  - 8 HPC applications + 10 DNN models.
- **Spy:** operates in the background, persistently tracking NVLink leakage vectors.

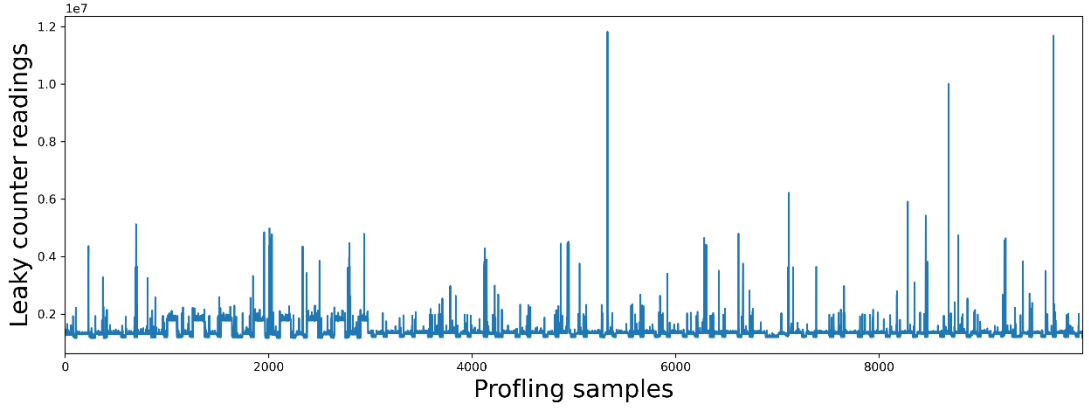
NVLink Leakage Trace

- *"nvlink_total_data_received"*.
  - Total data bytes received through NVLinks.

"rf" from openMM benchmarks



"ResNet-50"



Cross-GPU Side Channel Attacks

- Attack 1: Application Fingerprinting.
- Evaluation among 18 applications.
  - Features engineering.
  - Classification.

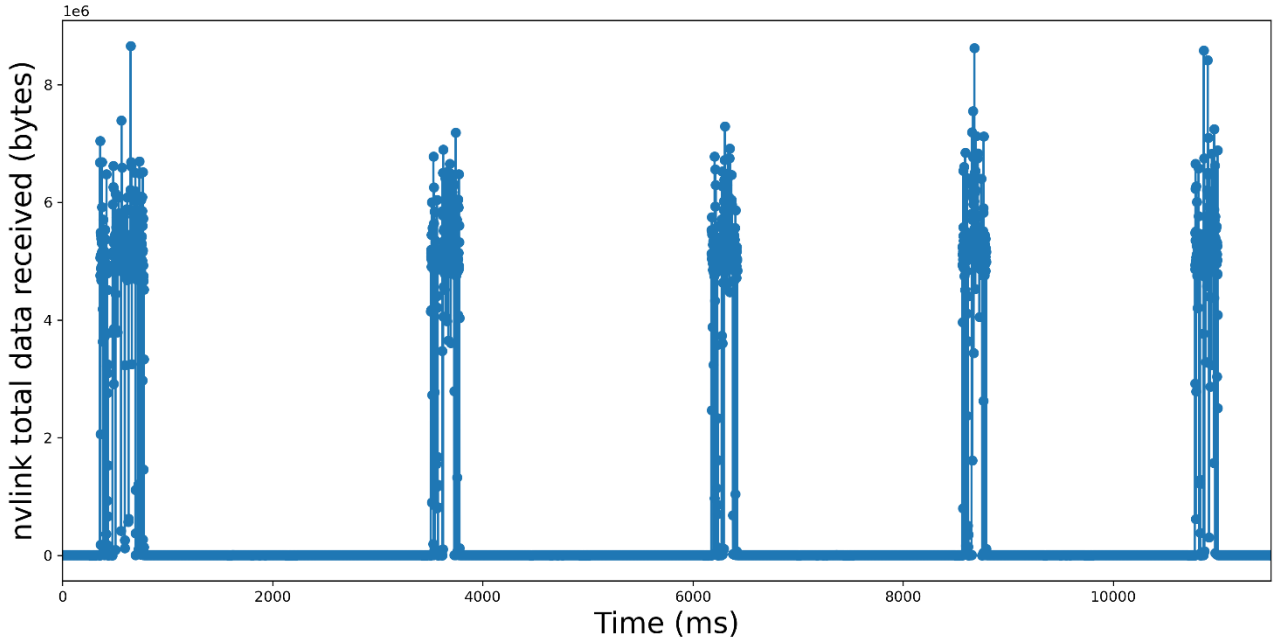
	DGX			GCP		
	F1	**Prec**	**Rec**	**F1**	**Prec**	**Rec**
KNN	25.96	31.45	26.11	55.77	55.97	58.89
XGBoost	90.87	91.45	91.11	97.78	98.06	97.78
LightGBM	**92.22**	93.12	92.22	**96.10**	96.93	96.11

Cross-GPU Side Channel Attacks

- Attack 2: Fingerprinting 3D graphics character rendering.
- **Victim:** renders her 3D graphics character across multi-GPU systems.
  - 50 fully rigged 3D characters from the Blender Studio open movies.
- **Spy:** operates in the background, persistently tracking NVLink leakage vectors.

NVLink Leakage Trace

- *"nvlink_total_data_received"*.
  - Total data bytes received through NVLinks.

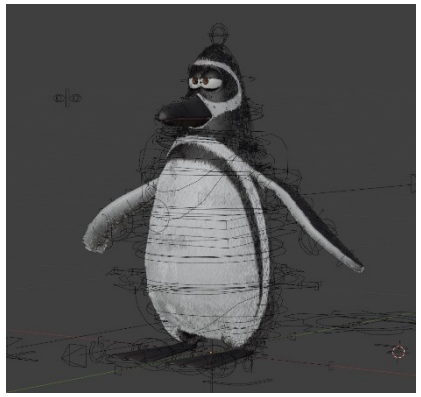


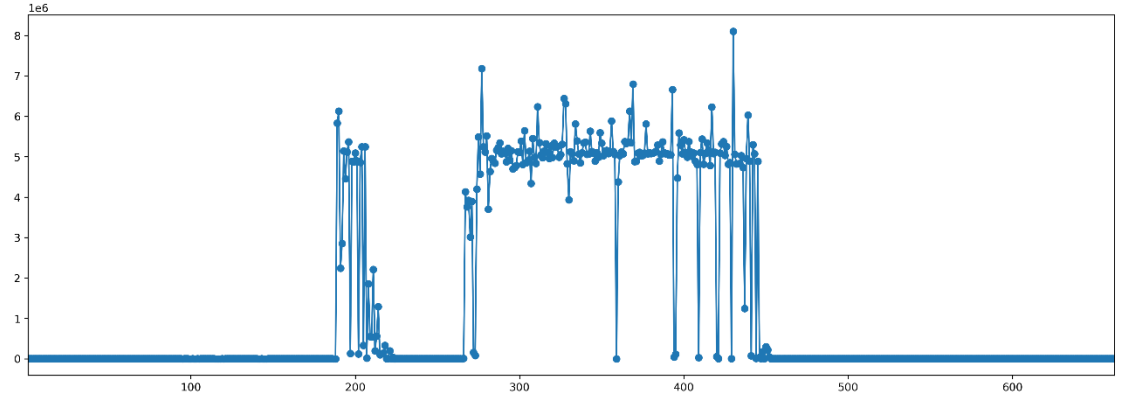
"NVLink leakage traces of 5 consecutive frames"

NVLink Leakage Trace

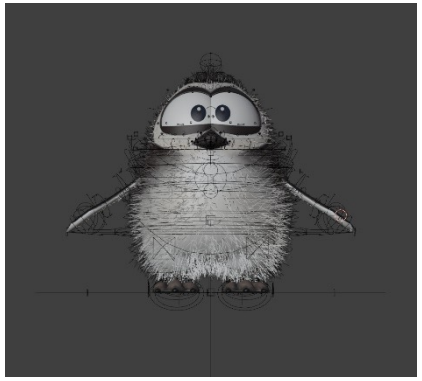
- *"nvlink_total_data_received"*.
  - Total data bytes received through NVLinks.

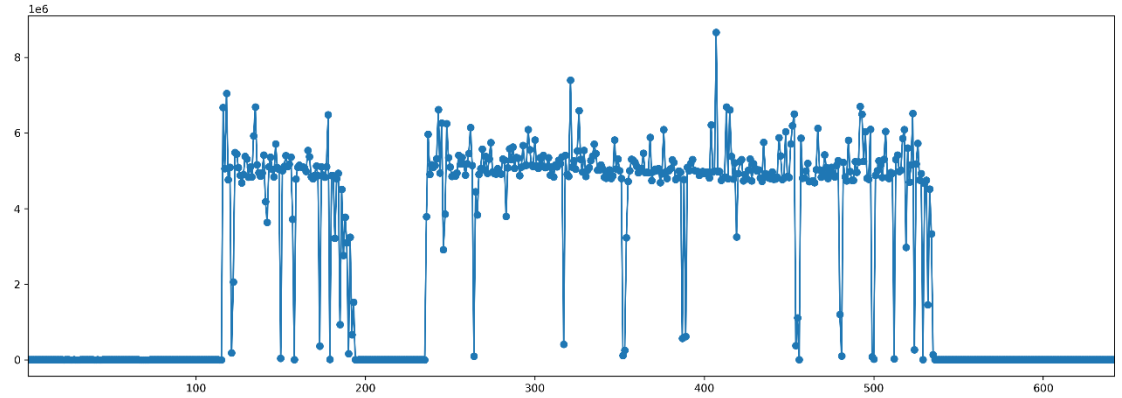
Character 1: "Pinguino"





Character 2: "Oti"



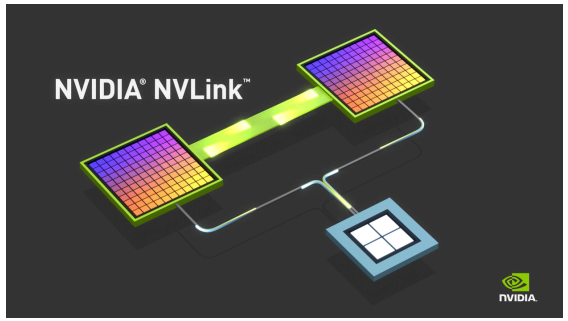


Cross-GPU Side Channel Attacks

- Attack 2: Fingerprinting 3D graphics character rendering.
- Evaluation among 50 characters.
  - Features engineering.
  - Classification.

	F1	Prec	Rec
KNN	59.74	62.71	62.50
XGBoost	90.11	93.10	90.50
LightGBM	**91.56**	94.11	92.00

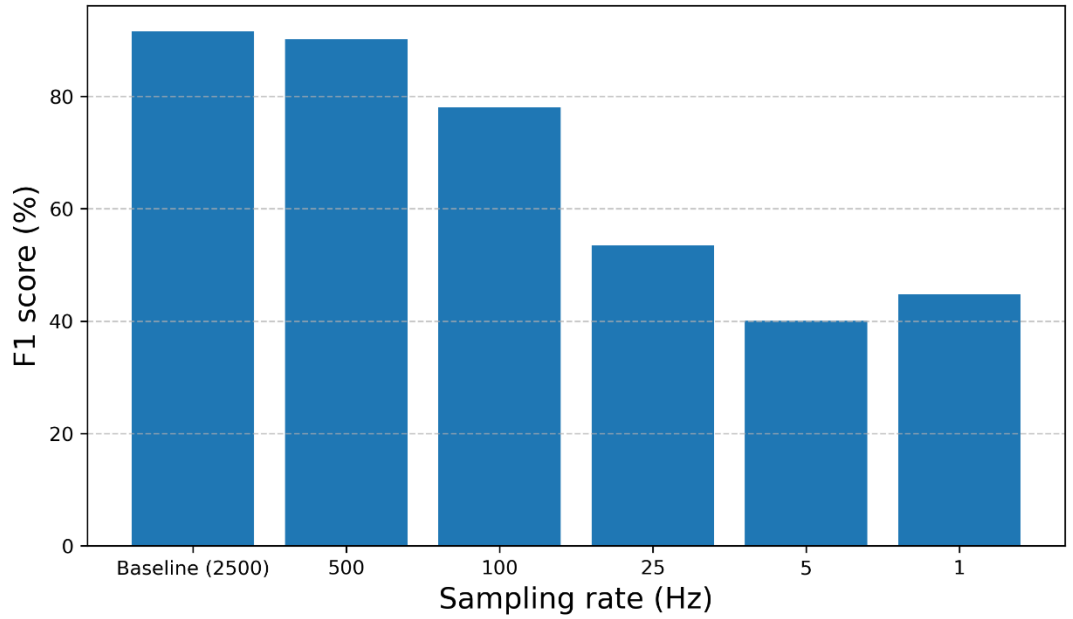
Outline



- Background: Multi-GPU interconnect.
- Threat model and leakage vectors.
- Cross-GPU covert channel attacks.
- Cross-GPU side channel attacks.
- <u>Mitigation.</u>

Mitigation

- Restricting access to high-resolution clock instructions.
- Detecting abnormal NVLink monitoring and/or contention.
- Managing access to leaky counters.



Conclusion

- Covert and Side-channels on multi-GPU interconnect.
  - Through contention and leaky counters (**First**).
- Cross-GPU covert channel attack.
- Two end-to-end cross-GPU side channel attacks.
- Mitigation based on limiting the precision or rate is not effective.
- Future work:
  - Finer-grained side channel attack; better profiling systems for interconnect.

Thank you!

Any questions?

Yicheng Zhang

yzhan846@ucr.edu

https://yichez.site