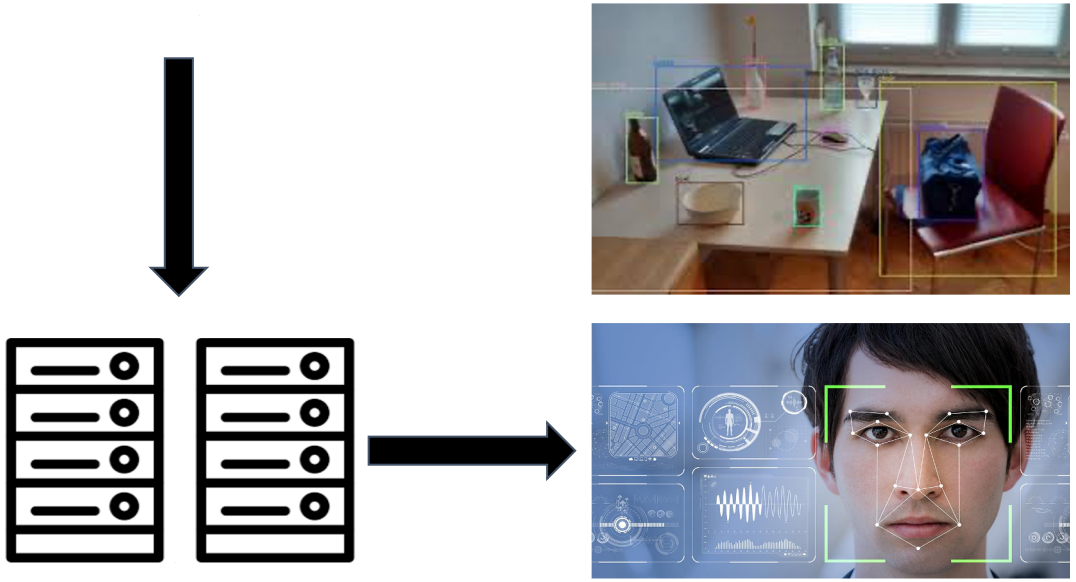
Hermes Attack: Steal DNN Models with Lossless Inference Accuracy

Yuankun Zhu[#], Yueqiang Cheng[*], Husheng Zhou[$], Yantao Lu[+]
[#]University of Texas at Dallas, [*]Baidu Security, [$]Vmware, [+]Syracuse University

Motivations

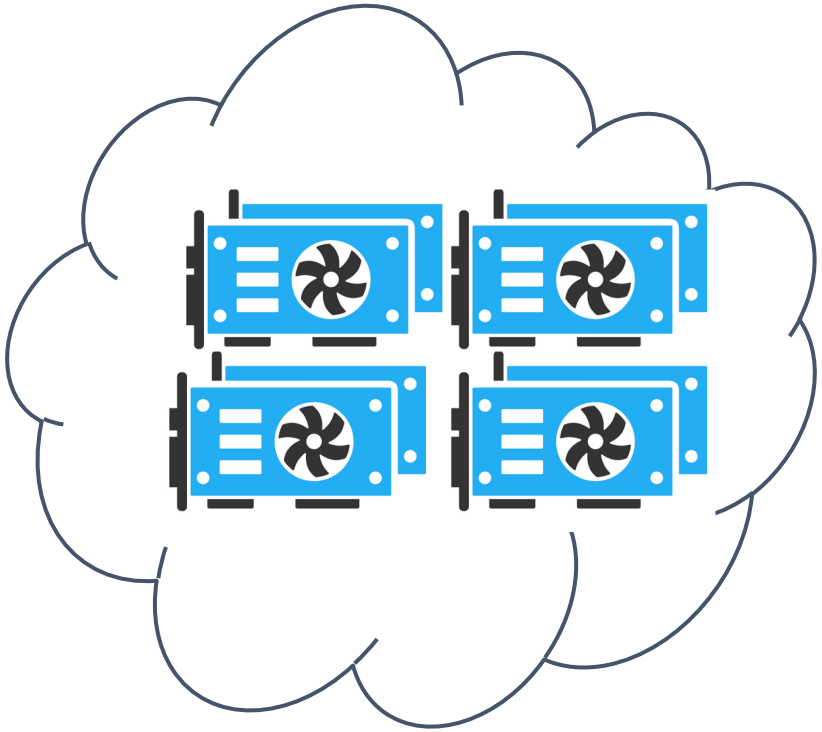
Autonomous Driving

Privatization Deployments



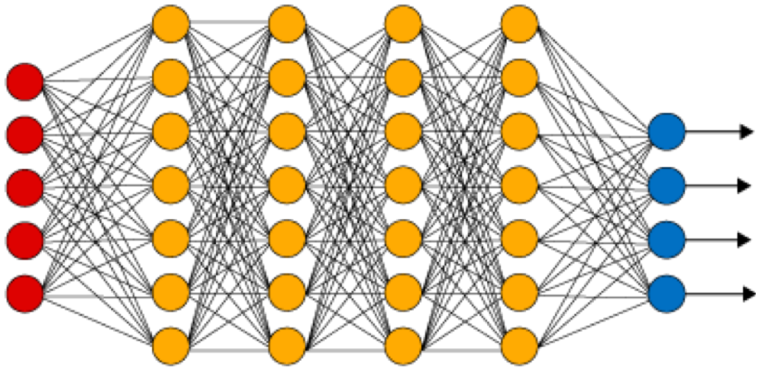
Motivations

$4.5/hr per TPU

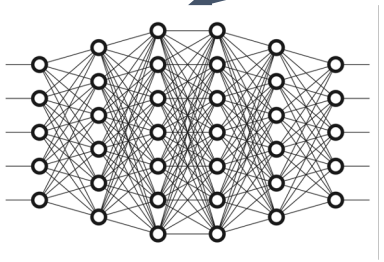


≈$400,000

DNN Model



Motivations



Architecture [23,24,55,18,38]

- Kernel size = (3,3)
- Pool size = (3,3)
- Strides = (1,1)

...

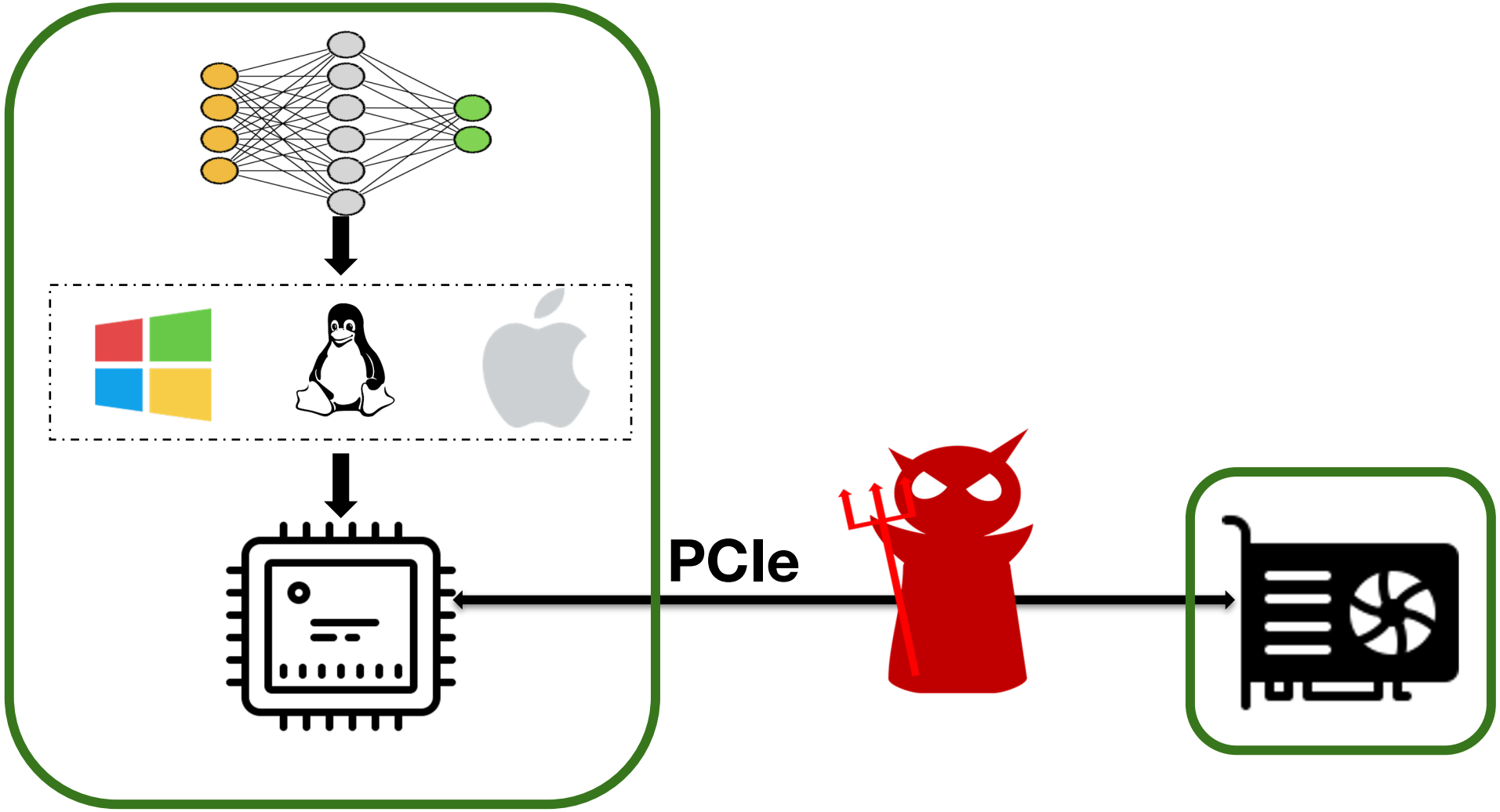
Hyper-parameters [58,55,51,38]

1101010101
1010101110
1101010111
1001011001

Parameters [58,43]

- **All** existing attacks can ***only*** reconstruct ***partial*** model
- **Hermes Attack** is the ***first*** work that can ***fully*** steal DNN model with ***lossless*** interface accuracy

Threat Model



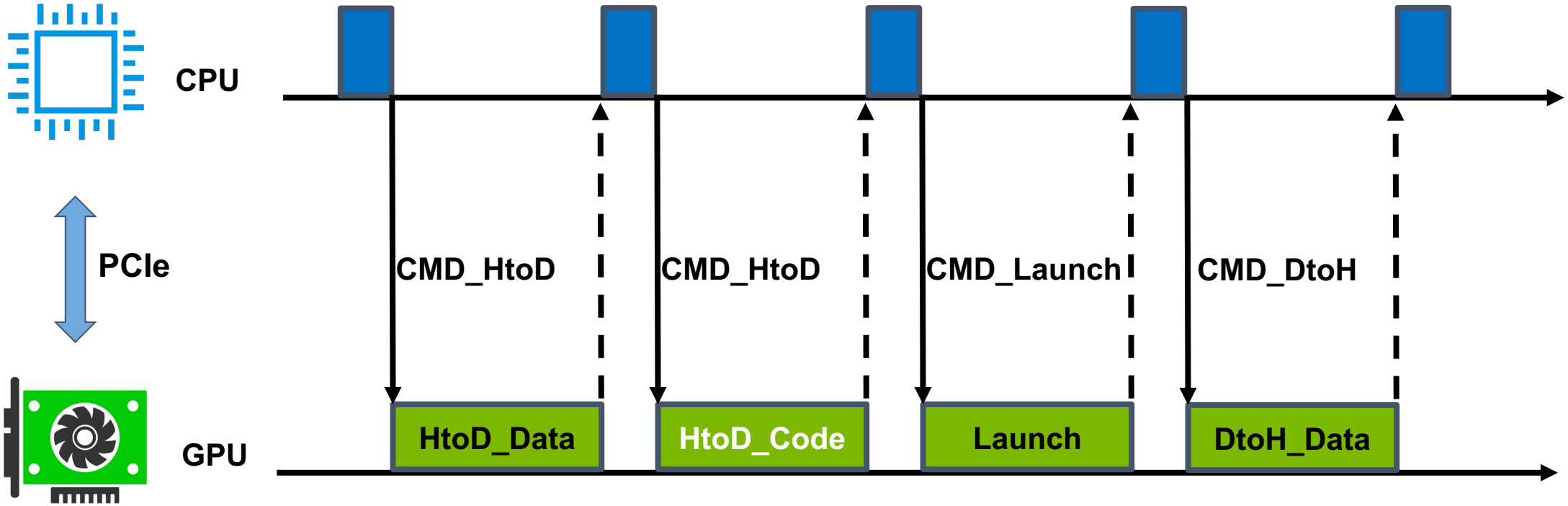
Challenges and Solutions

C1: Closed-source Code and Undocumented Data Structures → **S1:** Reverse Engineering

C2: Numerous Noises and Chaotic Orders in PCIe packets → **S2:** Sanitization and Order Correction

C3: Semantic Loss in PCIe Traffic → **S3:** Semantic Reconstruction

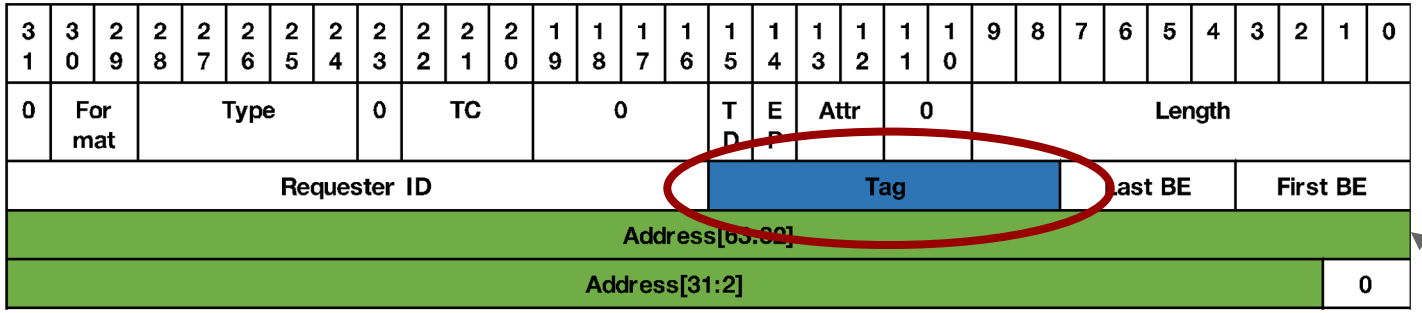
How GPU Works



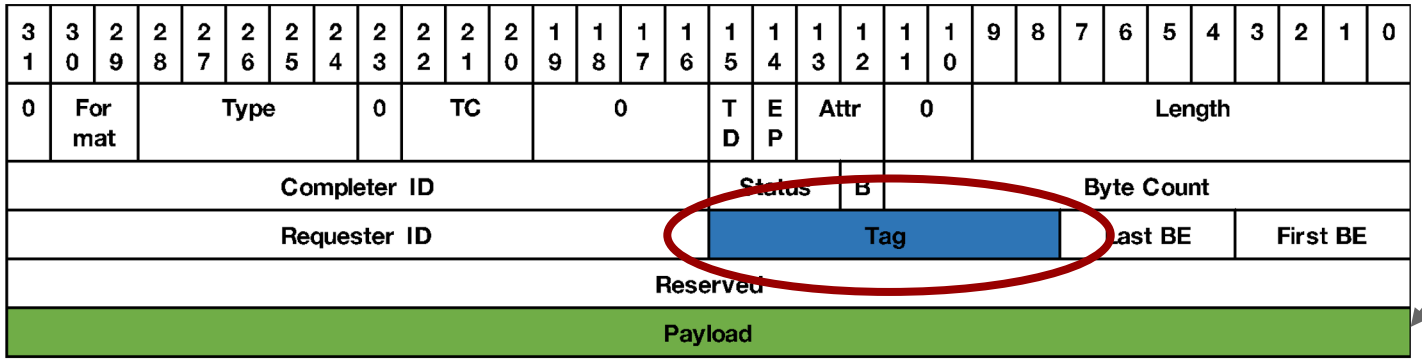
O1: **Command driven** interaction working mode

O2: All data and code are passed through **PCIe bus**

PCIe Packet Relationship



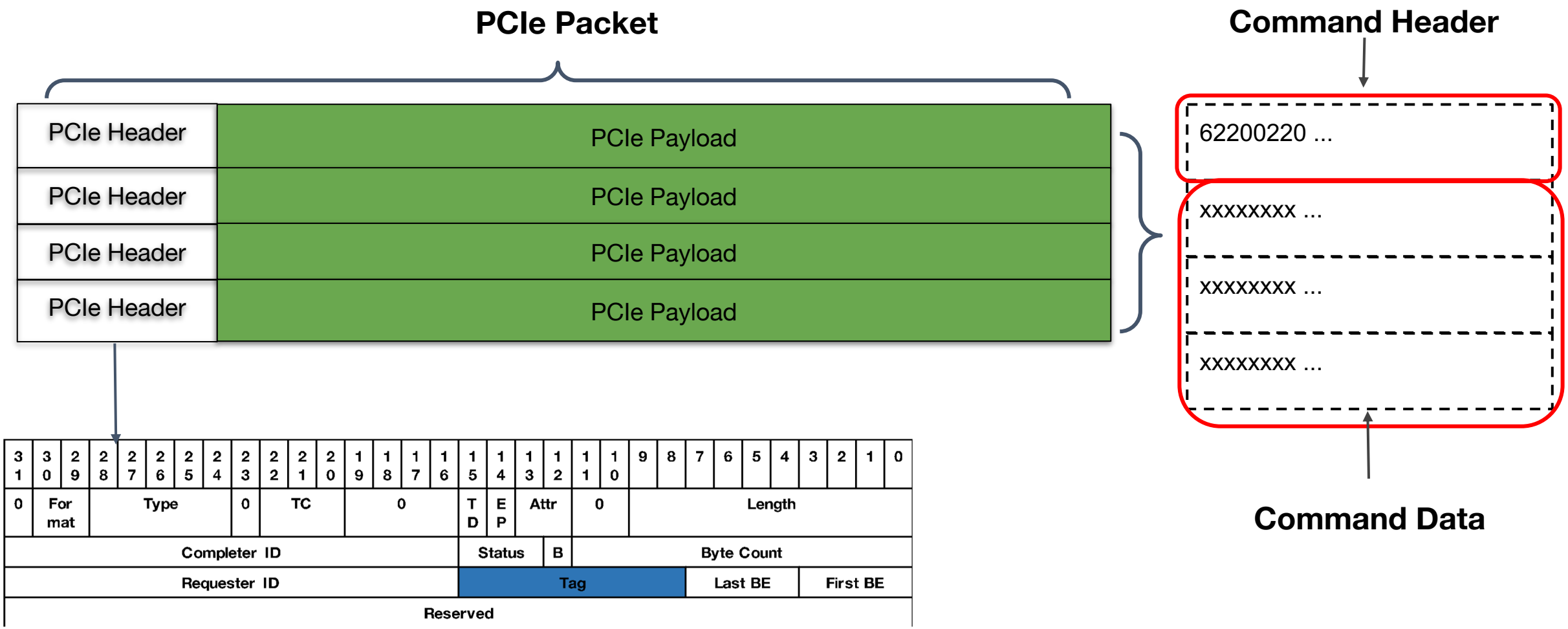
Example of Read Request TLP



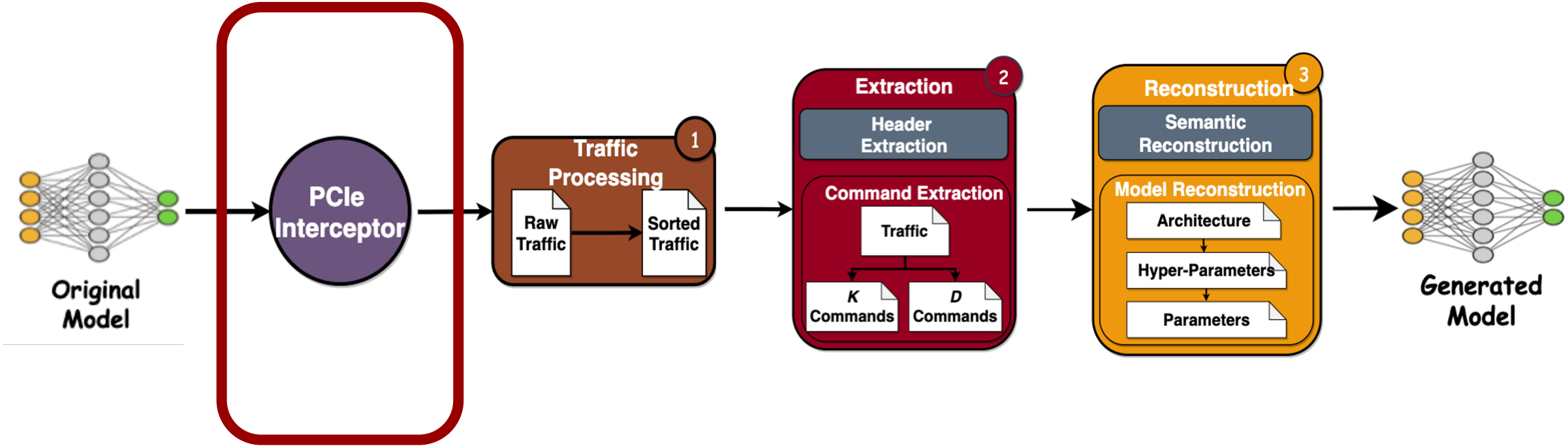
Example of Completion TLP

Address and Payload are connected through the same tag value

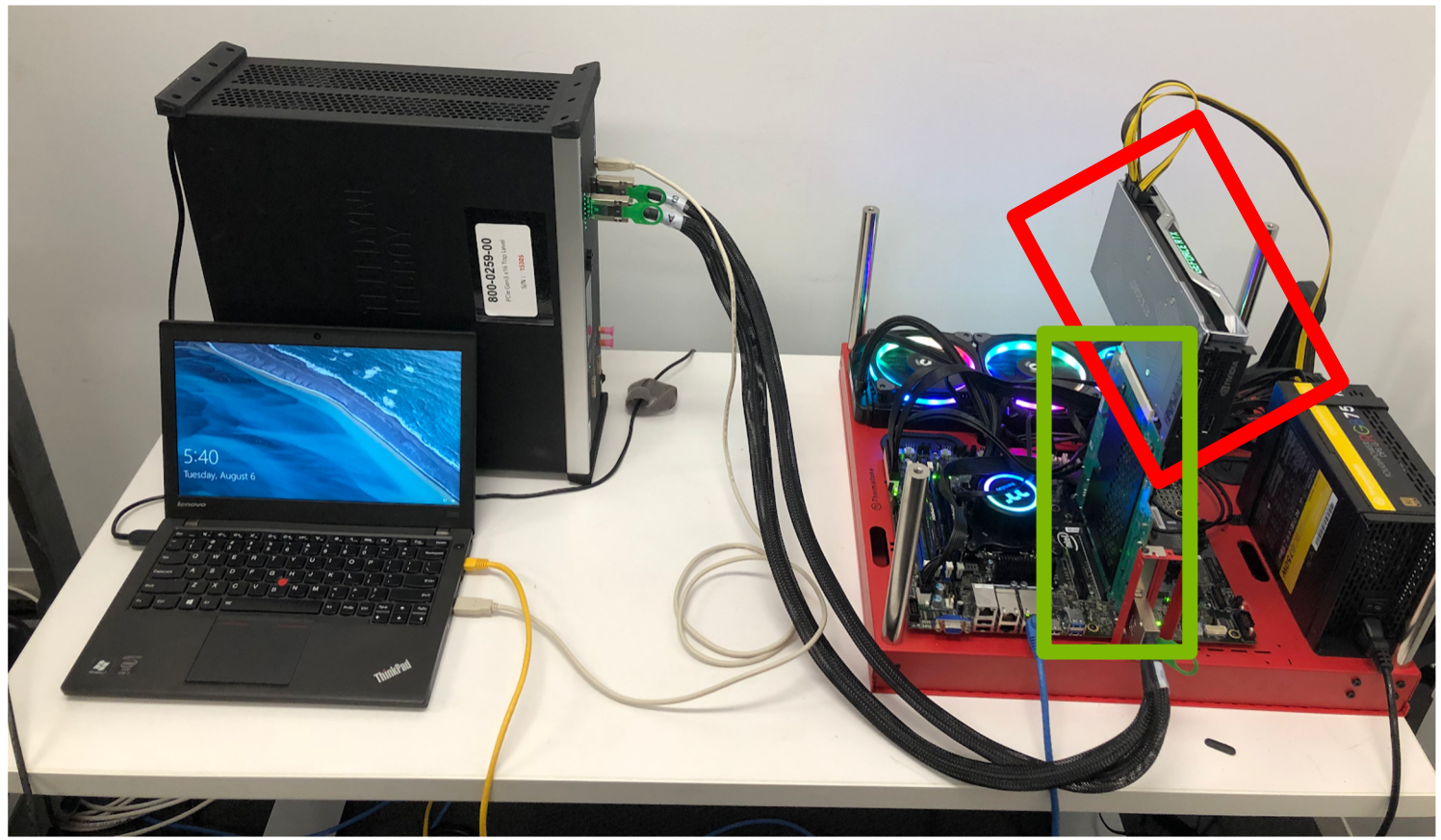
Terminologies



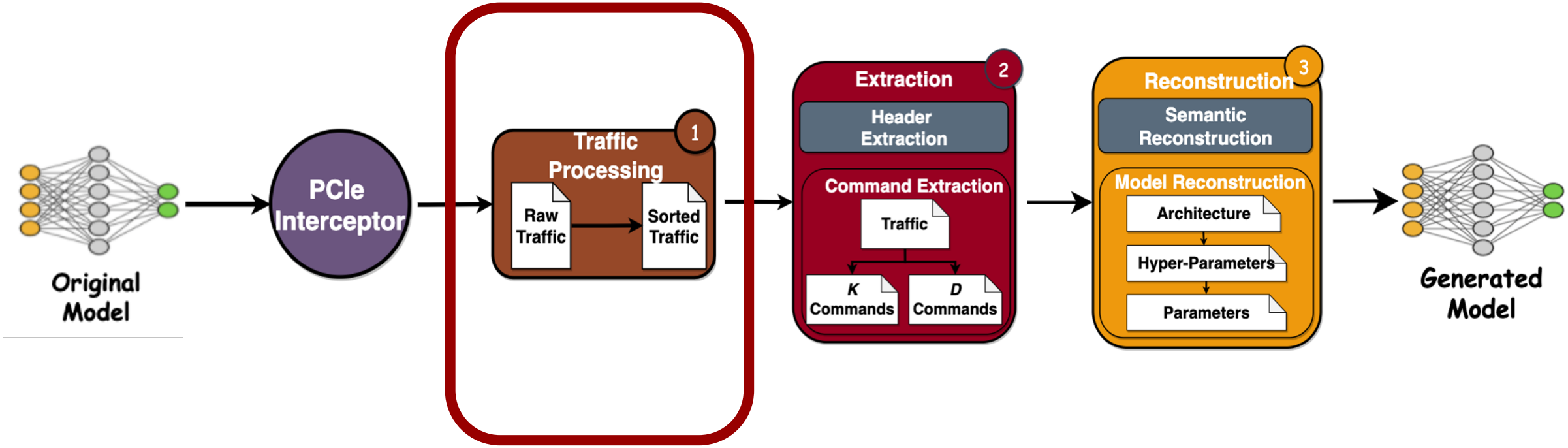
Attack Overview



PCIe Inteceptor



Attack Overview



Traffic Processing

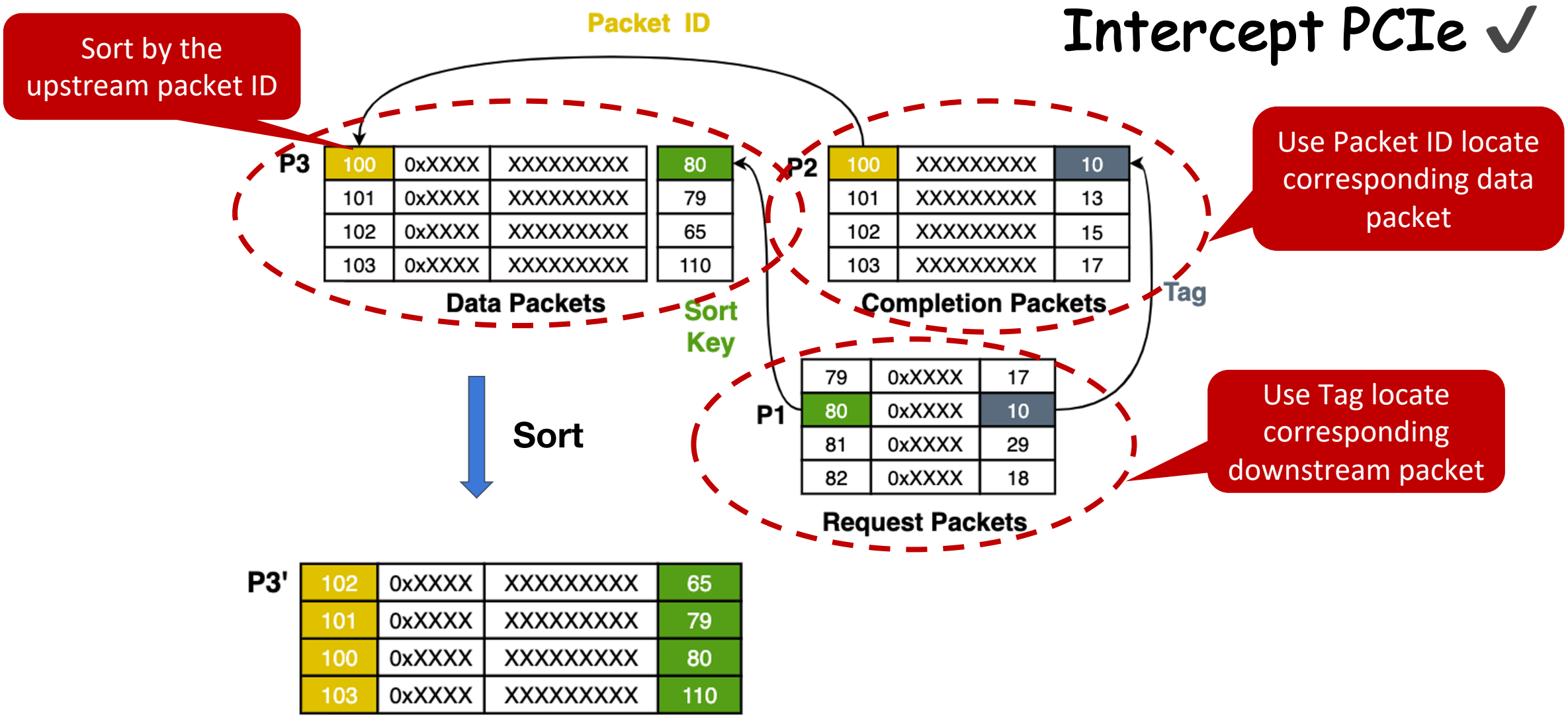
- PCIe Bus Snooping Device: PCIe protocol Analyzer

Packet	Dir	Type	Length	Address	Payload
168	R→	MRd64/CplD (SC)	16	00000008:1B521080	00000000 0DD06A81 00000000 0DD16A81 00000000 0DD26A81 00000000 0DD36A81 ...
169	R→	MRd64/CplD (SC)	16	00000008:1B521040	00000000 0D5AAE82 00000000 0D5BAE82 00000000 0DD8B281 00000000 0DD9B281 ...
170	R→	MRd64/CplD (SC)	16	00000008:1B5210C0	00000000 0D94977E 00000000 0D95977E 00000000 0D96977E 00000000 0D97977E ...
171	R→	MRd64/CplD (SC)	32	00000008:1B521100	00000000 0D549581 00000000 0D559581 00000000 0D569581 00000000 0D579581 ...
172	R→	MRd64/CplD (SC)	16	00000008:1B521180	00000000 0D10D082 00000000 0D11D082 00000000 0D12D082 00000000 0D13D082 ...
173	R→	MRd64/CplD (SC)	16	00000008:1B521200	00000000 0D1C157E 00000000 0D1D157E 00000000 0D1E157E 00000000 0D1F157E ...
174	R→	MRd64/CplD (SC)	16	00000008:1B5211C0	00000000 0D908E81 00000000 0D918E81 00000000 0D928E81 00000000 0D938E81 ...
175	R→	MRd64/CplD (SC)	16	00000008:1B52124	00000000 0D507681 00000000 0D517681 00000000 0D527681 00000000 0D537681 ...
176	R→	MRd64/CplD (SC)	32	00000008:1B521280	00000000 0D28D17B 00000000 0D29D17B 00000000 0D2AD17B 00000000 0D2BD17B ...
177	R→	MRd64/CplD (SC)	32	00000008:1B521300	00000000 0DF46B81 00000000 0DF56B81 00000000 0DF66B81 00000000 0DF76B81 ...
178	R→	MRd64/CplD (SC)	32	00000008:1B521400	00000000 0DEC637E 00000000 0DED637E 00000000 0DEE637E 00000000 0DEF637E ...
179	R→	MRd64/CplD (SC)	32	00000008:1B521380	00000000 0DF8AE82 00000000 0DF9AE82 00000000 0DFAAE82 00000000 0DFBAE82 ...
180	R→	MRd64/CplD (SC)	32	00000008:1B521480	00000000 0D2C8781 00000000 0D2D8781 00000000 0D2E8781 00000000 0D2F8781

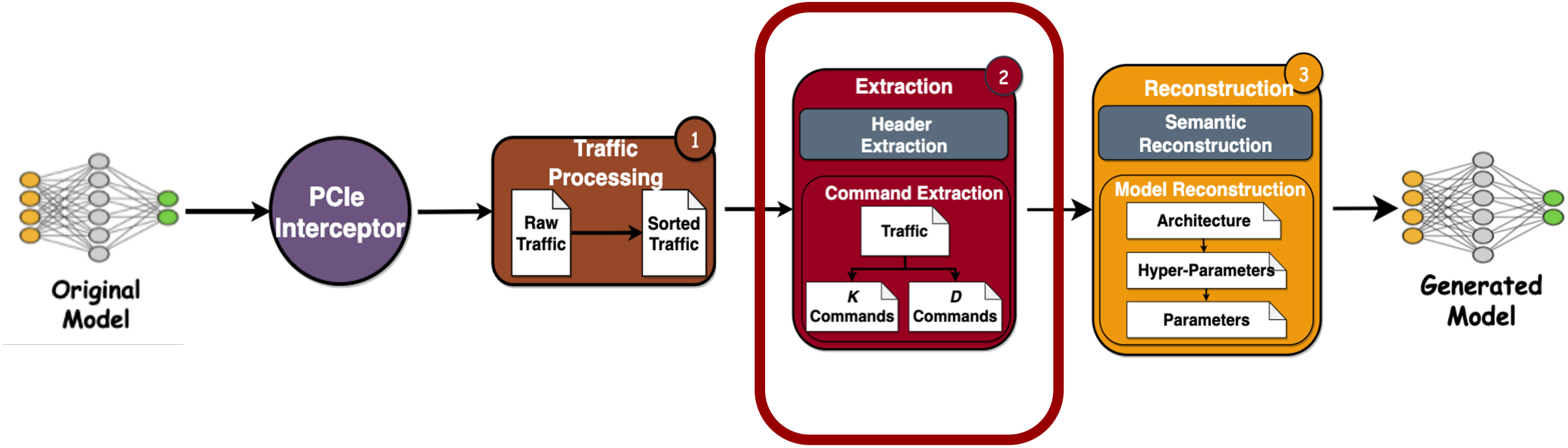
Out of order!

Sort Payload

Intercept PCIe ✓



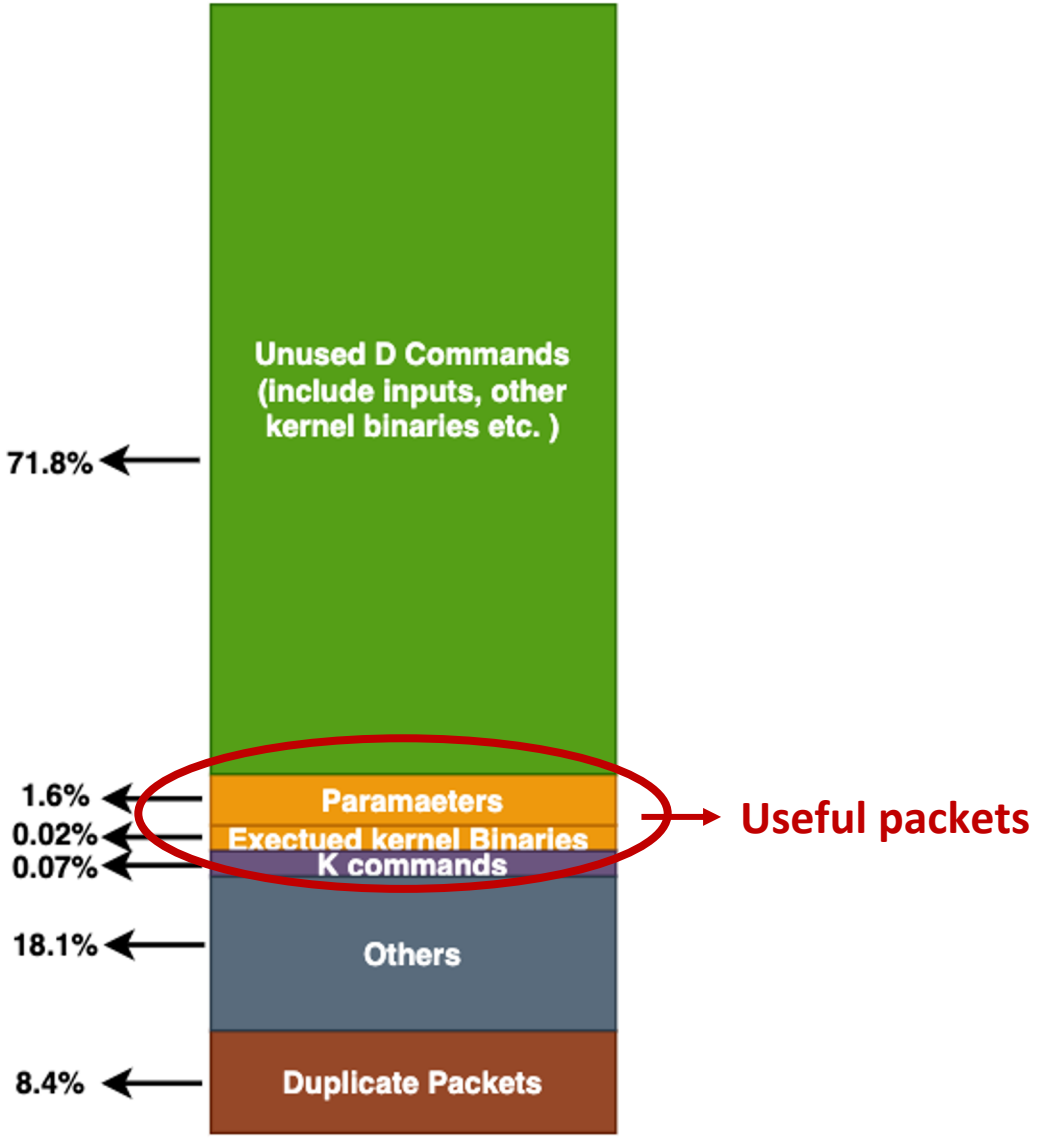
Attack Overview



Numerous Noises

- In a Traffic sample:
  - Number of PCIe packets: 1,077,757
  - Number of useful packets: ~20,000
  - 98.4% of the packets are noise

Focus on specific GPU Commands!



Header Extraction

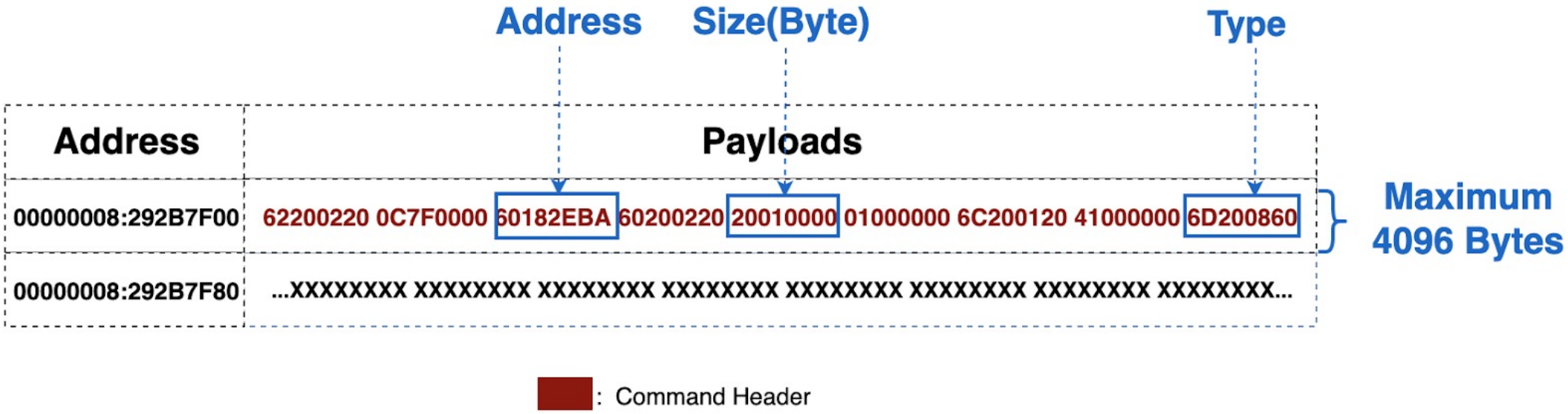
- GPU is controlled by CPU using commands. Command types includes:
  - GPU Initialization Command
  - Synchronization Command
  - Data Movement Command
  - Kernel Launch Command

- We only need **Data Movement Commands**, noted as **D commands**, and **Kernel Launch Commands**, noted as ***K* commands**.

- Identify headers of ***D* cmds** and ***K* cmds** through offline training.

Header Extraction

Example of a GPU command header



Command Extraction

	Address	Payloads
Gap	00000008:292B7F00	...XXXXXXXX XXXXXXXX XXXXXXXX 62200220 0C7F0000 60182EBA 60200220 20000000
	00000008:188B6700	...XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX...
	00000008:292B7F80	01000000 6C200120 41000000 6D200860 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX ...

: Noise : Command Header

Command Header Noise: when a command header has been split into two packets, the noise may appear between these two packets.

	Address	Payloads
Gap	00000008:292B7F00	62200220 0C7F0000 60182EBA 60200220 20000000 01000000 6C200120 41000000 6D200860
	00000008:188B5760	...XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX...
	00000008:292B7F80	...XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX...

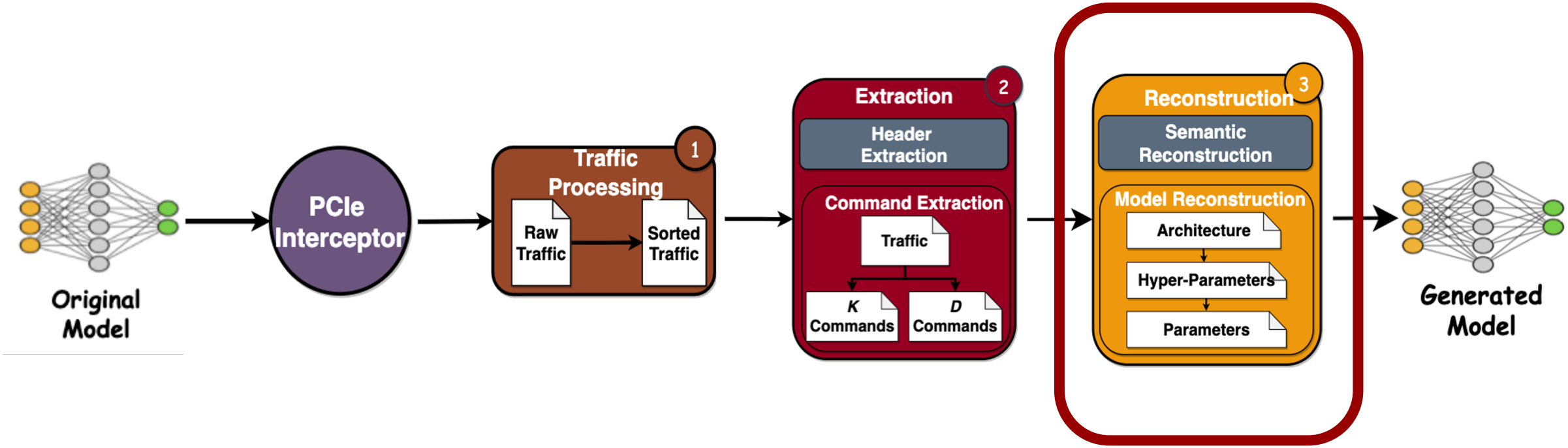
: Noise : Command Header

Command Data Noise: noise packet appeared among command data packets.

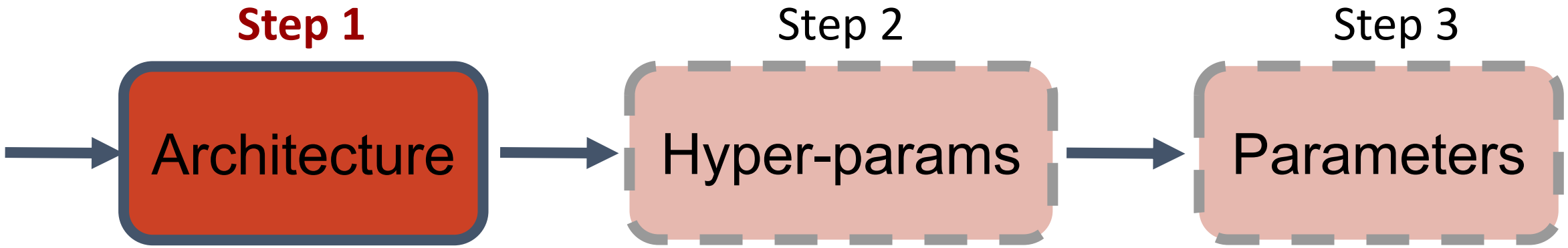
Check address consistency !

Command Extraction ✓

Attack Overview



Model Reconstruction



Semantic Reconstruction Challenges

- **Layer type information is lost in PCIe traffic**
- Every layer is completed through several kernels.
- Every Kernel is launched through a ***K* command.**
- Every *K* Command include an address pointed to its kernel binaries.

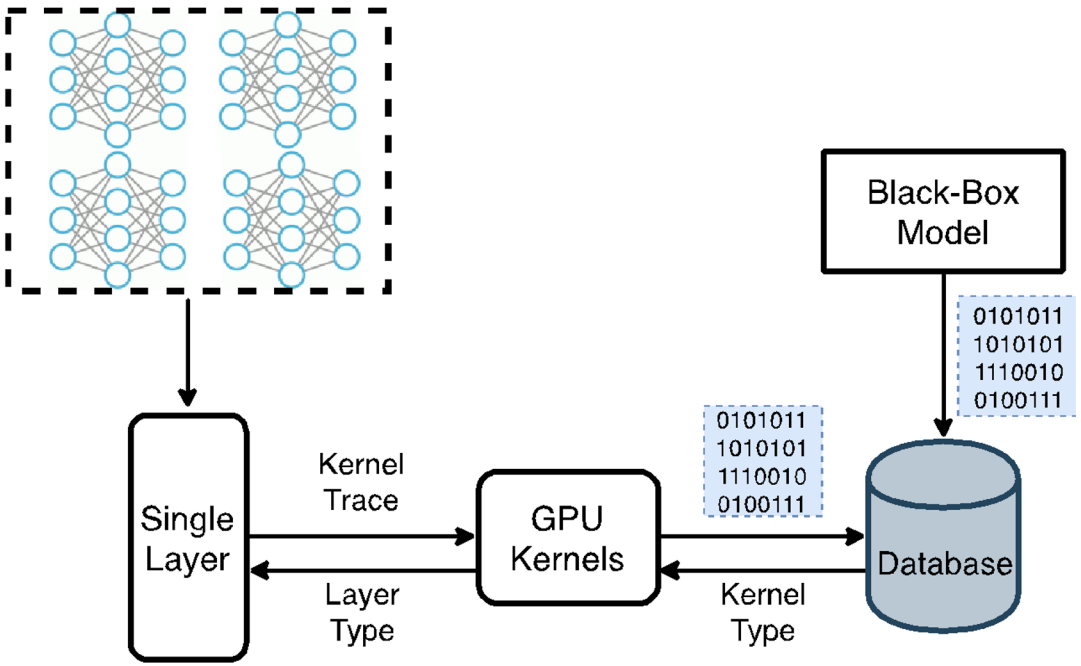
Every *K* Command include an address pointed to its kernel binaries.

```
62200220 0E000000 405ECF01 60200220 00010000 01000000
6C200120 41000000 6D204060     ...

62200220 0B000000 00004300 60200220 20010000 01000000
6C200120 41000000 6D204860     ...     0B000000 405ECF01
```

Offline Database Generation

- **Determine kernel types of kernel binaries**
  - Extract corresponding kernel binaries from PCIe traffic.
  - Obtain kernel traces from profiling tools.
- **Determine kernels of layers**
  - Repeat inference procedure on different models.
  - Obtain kernel trace of each layer.
- **Put <kernel binaries, layer types> pairs into database.**

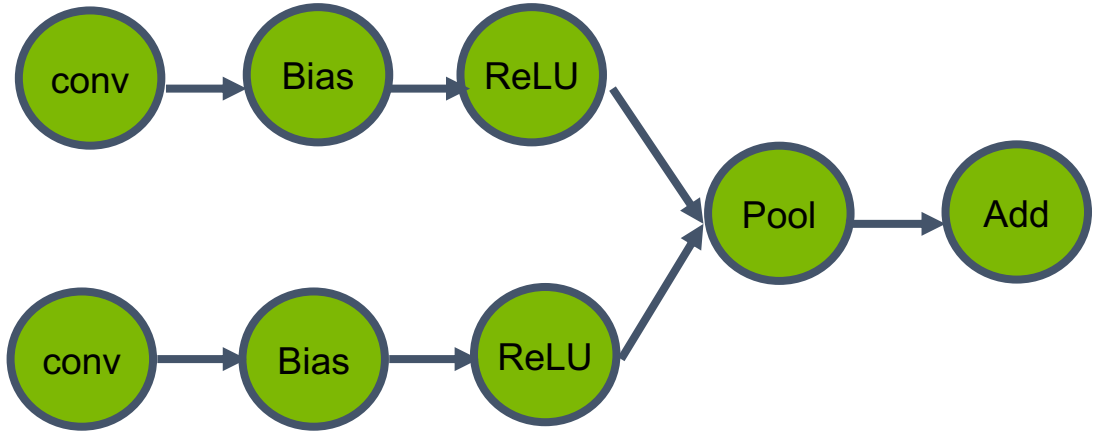


Extract Architecture

- Build Data Flow Graph
  - Treat kernel as vertice
  - Treat Input address and output address as flow-from and flow-to
- Substitute kernel vertice as DNN layers

Extract Architecture

- Build Data Flow Graph
  - Treat kernel as vertice
  - Treat Input address and output address as flow-from and flow-to
- Substitute kernel vertice as DNN layers



Model Reconstruction

Extract Hyper-Parameters

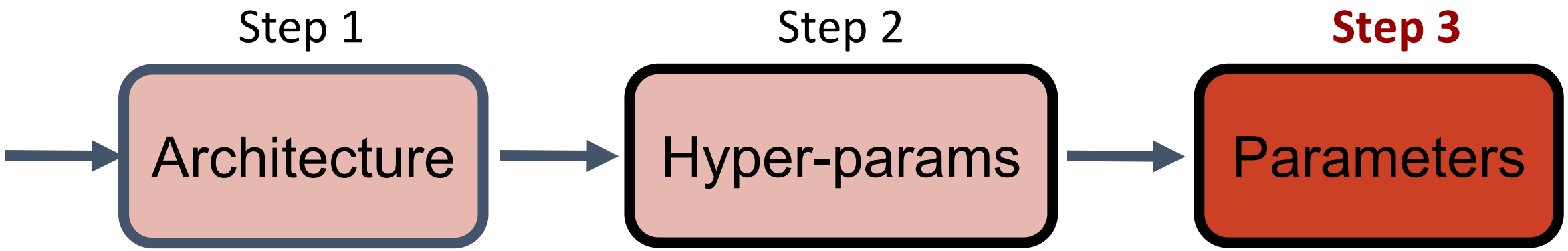
- **Directly obtained from PCIe**
  - Fetch from command data, use identified offset.
  - eg. Kernel size, Strides, Filters, etc.

Address	Payloads
00000008:C6ED0800	62200220 0B000000 00512D01 ... 6C200120 41000000 6D202061
00000008:C6ED0880	...XXXXXXXX XXXXXXXX 03000000 03000000 XXXXXXXX...

kernel size

- **Infer**
  - Check if related kernel exist
  - eg. Activation function, Use_bias

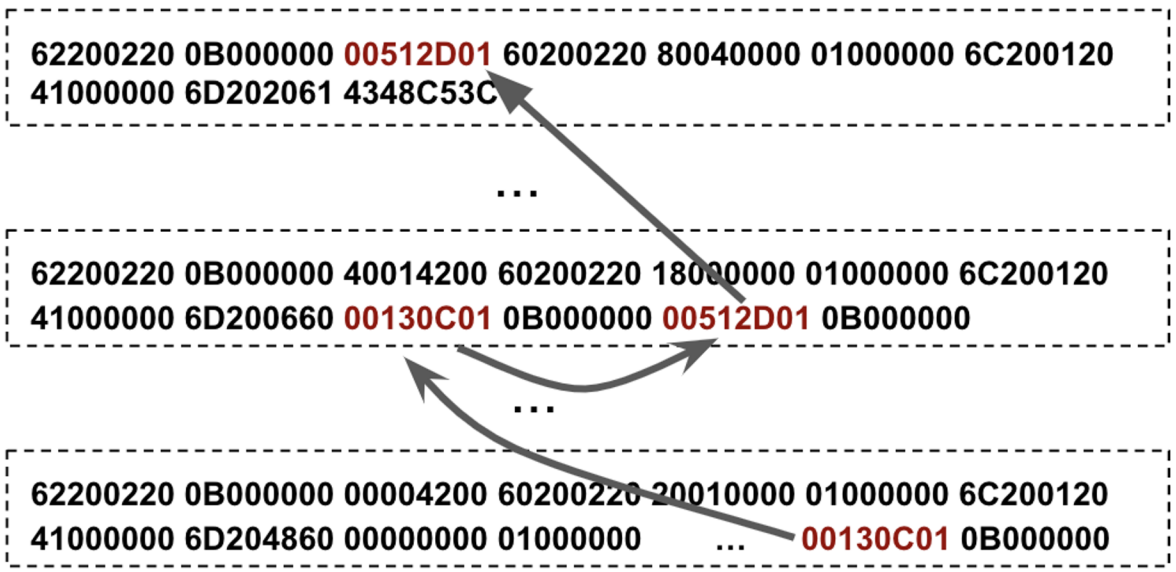
Model Reconstruction



Extract Parameters

- **Observation:**
  - Weights and bias are not called by kernels directly, they are first called by

    CudaMemcpyDeviceToDevice
  - This API is also use **kernel launch command** to complete.



Model Reconstruction ✓

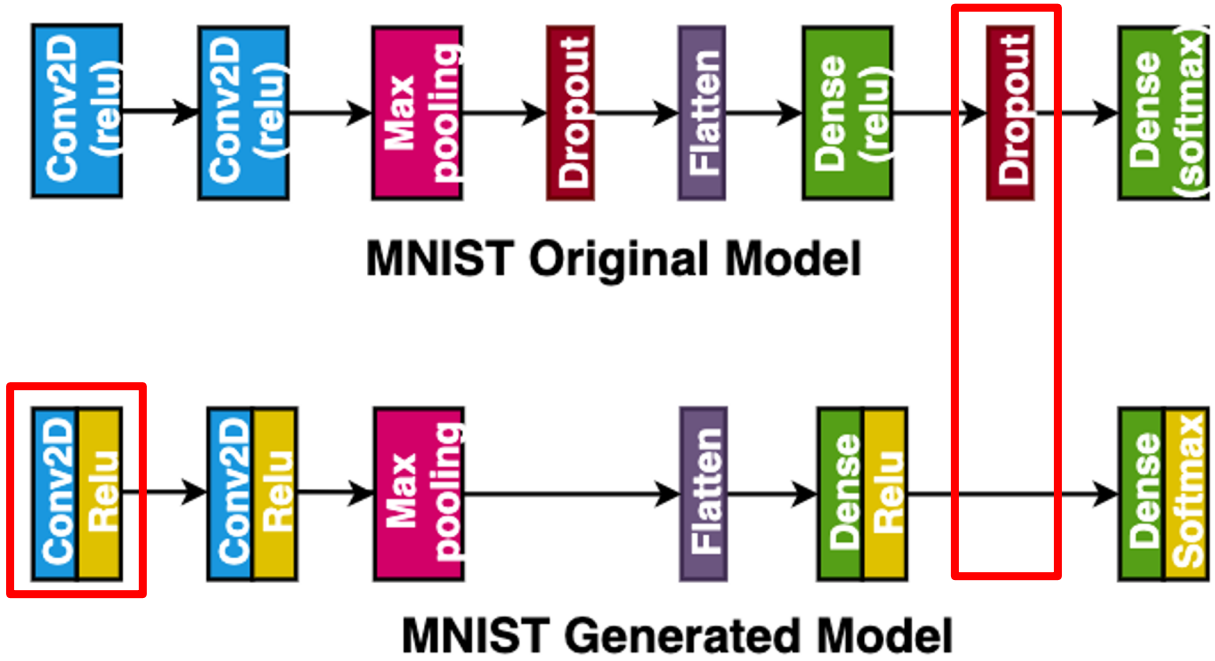
Details in our paper https://arxiv.org/abs/2006.12784

Evaluation

- Operating System: Ubuntu 16.04
- Tested GPU platform:
  - NVIDIA Geforce GT 730
  - NVIDIA Geforce GTX 1080 Ti
  - NVIDIA Geforce RTX 2080 Ti
- Tested DNN models:
  - MNIST
  - VGG-16
  - ResNet-20
- GPU Programing Interface: CUDA 10.1
- GPU Driver: NVIDIA 418.43
- DNN platform: keras + tensorflow

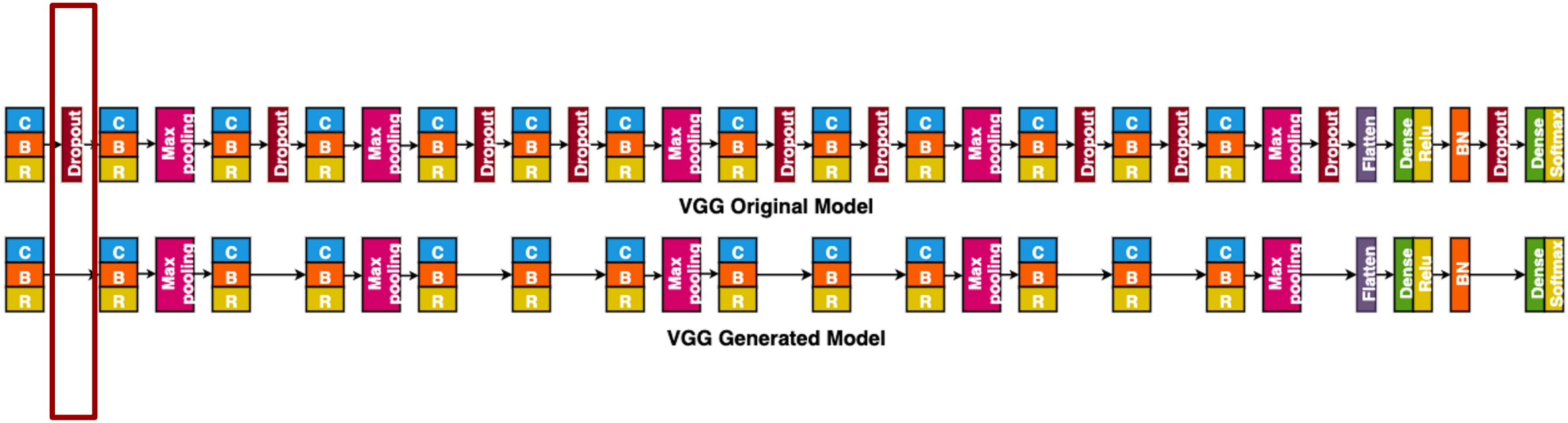
Evaluation

MNIST Architecture Comparison



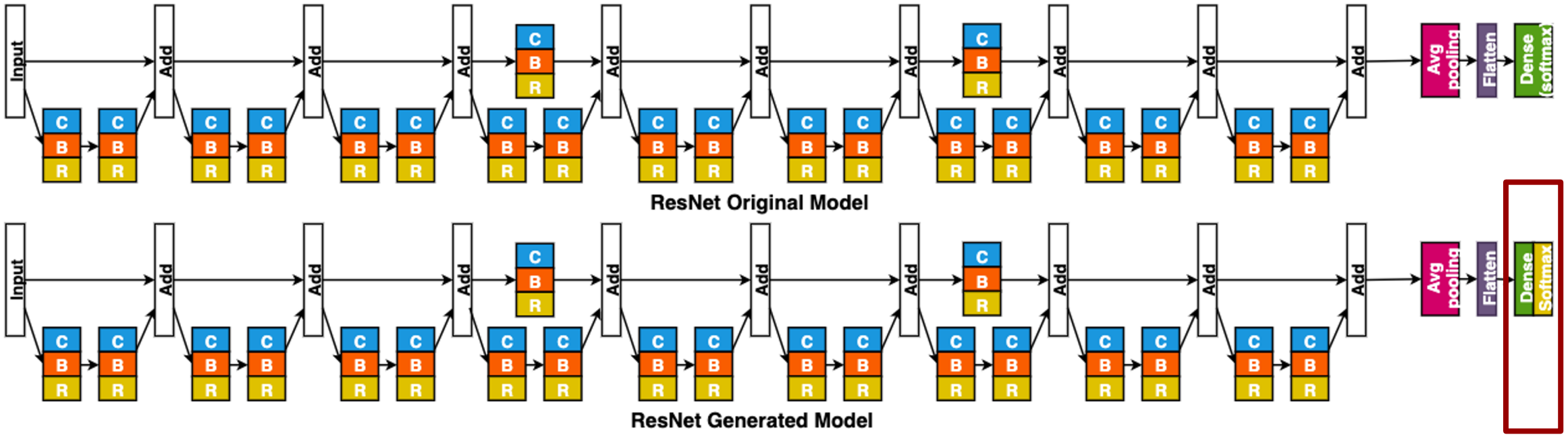
Evaluation

VGG Architecture Comparison



Evaluation

ResNet Architecture Comparison



Evaluation

Identity Evaluation

Metrics	Model	Original	Reconstructed		
N/A	N/A	N/A	GT 730	1080 Ti	2080 Ti
Accuracy	MNIST	98.25%		%	98.25%
	VGG	93.59%		%	93.59%
	ResNet	91.45%	91.45%	91.45%	91.45%
Inference Times(s)	MNIST	2	2	2	2
	VGG	65			61
	ResNet	20			21

Same Accuray

Same Efficiency

Evaluation

Performance Evaluation

	MNIST			VGG			ResNet		
Platform	GT 730	1080 Ti	2080 Ti	GT 730	1080 Ti	2080 Ti	GT 730	1080 Ti	2080 Ti
# of D Commands	25,680	28,590	24,342	27,287	27,677	24,931	28,433	28,518	25,577
# of K Commands	216	139	181	903	628	793	1011	886	988
# of Completion Packets	1,077,756	2,244,115	2,959,613	4,284,946	2,615,895	3,354,411	975,257	2,052,657	2,717,451
Generation Time (min)	5	8	11	17	11	12	6	9	10


Q&A