UC San Diego

HAtNet: Hardware Attestation of Neural Networks

Nojan Sheybani[1], Huili Chen[1], Xinqiao Zhang[1], Siam Hussain[1], and Farinaz Koushanfar[1]
{nsheyban, huc044, x5zhang, siamumar, farinaz}@ucsd.edu
[1]University of California San Diego

Abstract

❖ Presenting HAtNet, an **on-device DNN attestation** method that certifies the legitimacy of underlying hardware for running a given DL model

❖ Leveraging **Algorithm/Software/Hardware co-design** approach to develop HAtNet. HAtNet binds the parameter distribution of the trained model with a legitimate hardware platform

❖ Enabling **usage control** and **intellectual property (IP) protection** of DL platforms

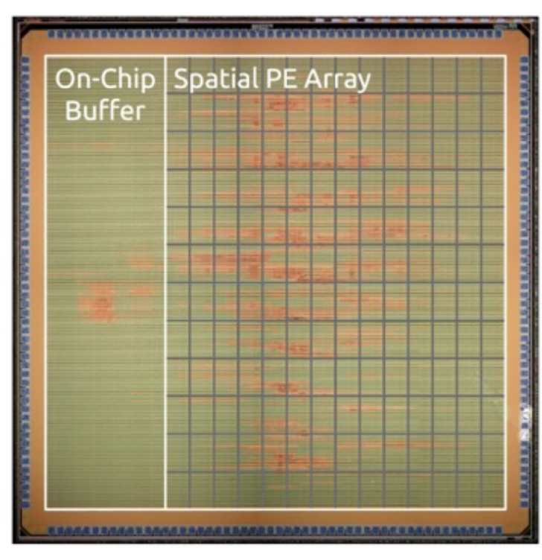
❖ Corroborating HatNet's **effectiveness, reliability**, and **efficiency** on various DNN benchmarks

Motivation

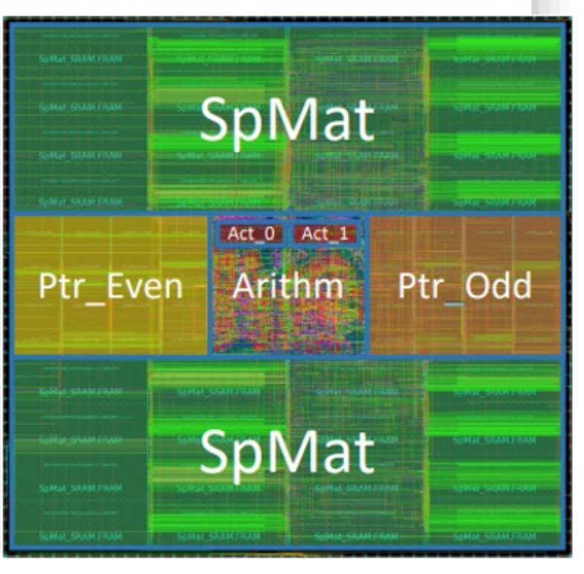
❖ Developing high-performance, large-scale DL models (e.g., Transformer, BERT, GPT-3) is both **time-** and **resource-consuming**

❖ **Functional DL model** shall be considered as **IP** of the designer and needs to be **protected** to preserve the commercial advantage of the DL model owner

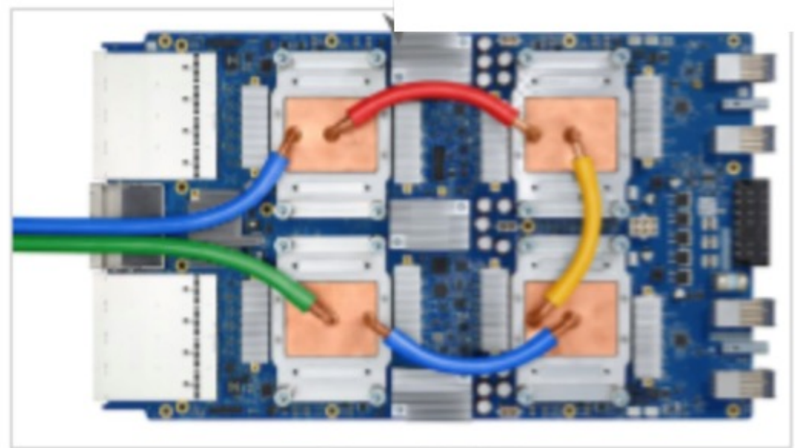
[Eyeriss ISCA'17]



[EIE ISCA'17]



[TPU ISCA'17]



Methodology

❖ HAtNet consists of two stages:

1 **Off-line marking phase**: Hardware provider generates a **unique, device-specific FP** and finetunes the model with the **FP-regularized loss**:

$$\mathcal{L} = \mathcal{L}_0 + \gamma\, MSE(f_j - Xw),\ \ f_j = \sum_{i=1}^{v} b_{ij}\, u_j.$$

FP is stored in secure memory of the target hardware

2 **Online attestation phase: Extracts FP** from the unknown/queried device when the trigger is activated:

$$f'_j = Xw',\ b_j{}' = f'^{T}_j * U$$

❖ HAtNet deploys a **hybrid trigger mechanism**:

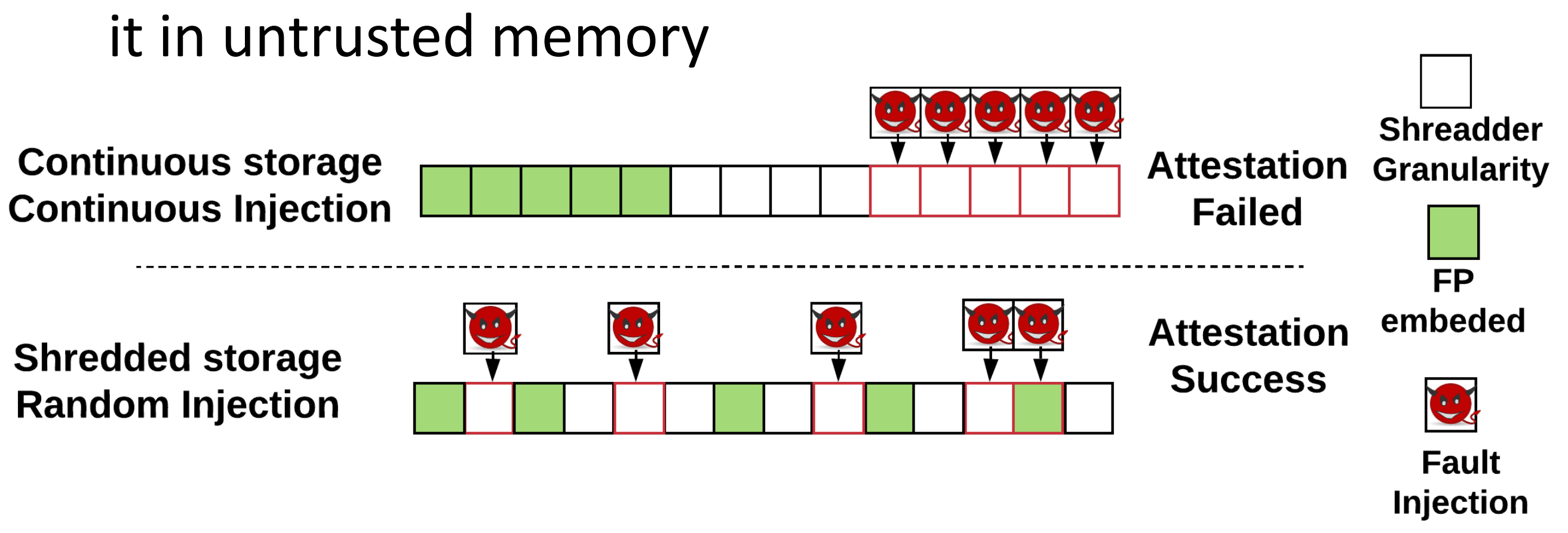
○ **Static trigger**: OS detects DNN program's start request

○ **Dynamic trigger:** Two sources: (1) memory change signal from OS monitor, and (2) fixed-frequency timer

Hardware Optimization

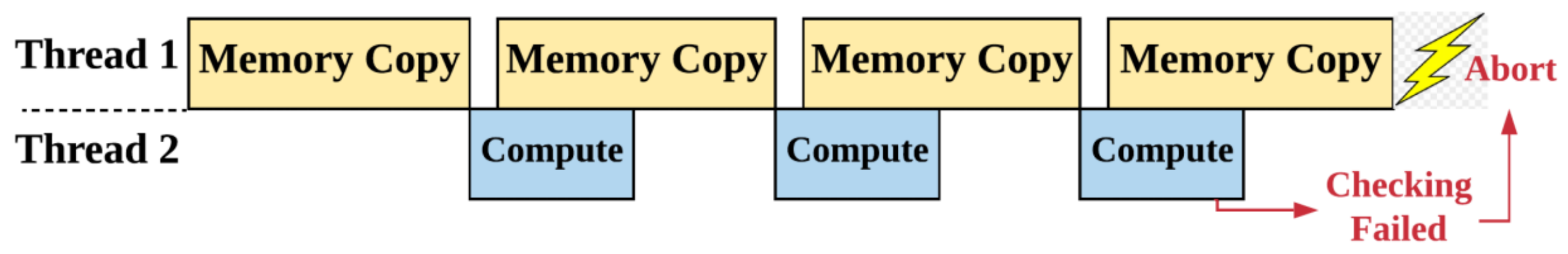
❖ HAtNet incorporates multiple HW optimization techniques for security and overhead consideration

○ **Shredder storage: shuffle weight** data before storing it in untrusted memory



○ **Data pipeline:** Hide the data communication latency

○ **Early stopping:** Skip unnecessary computation



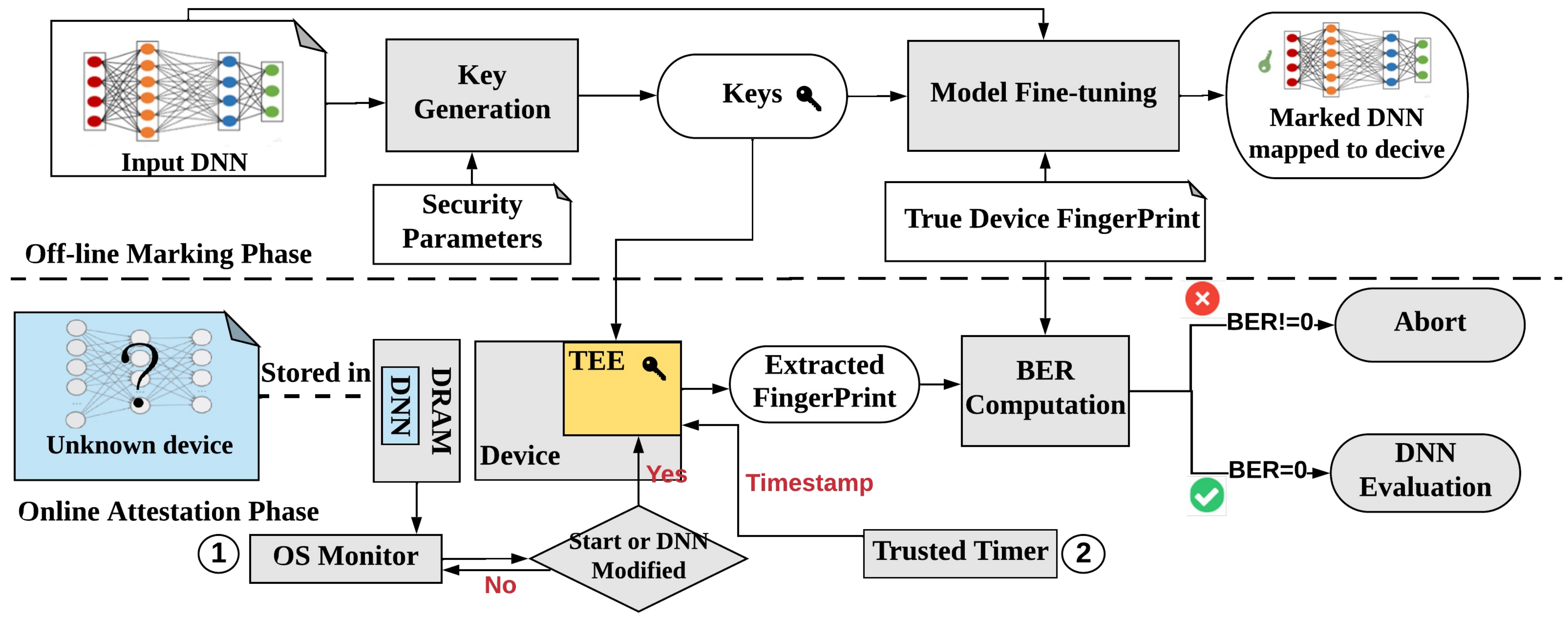
HAtNet's Global Flow

❖ **Threat Model**

○ The company is the IP holder. He / She sells the pre-trained DNNs together with the legitimate DL device.

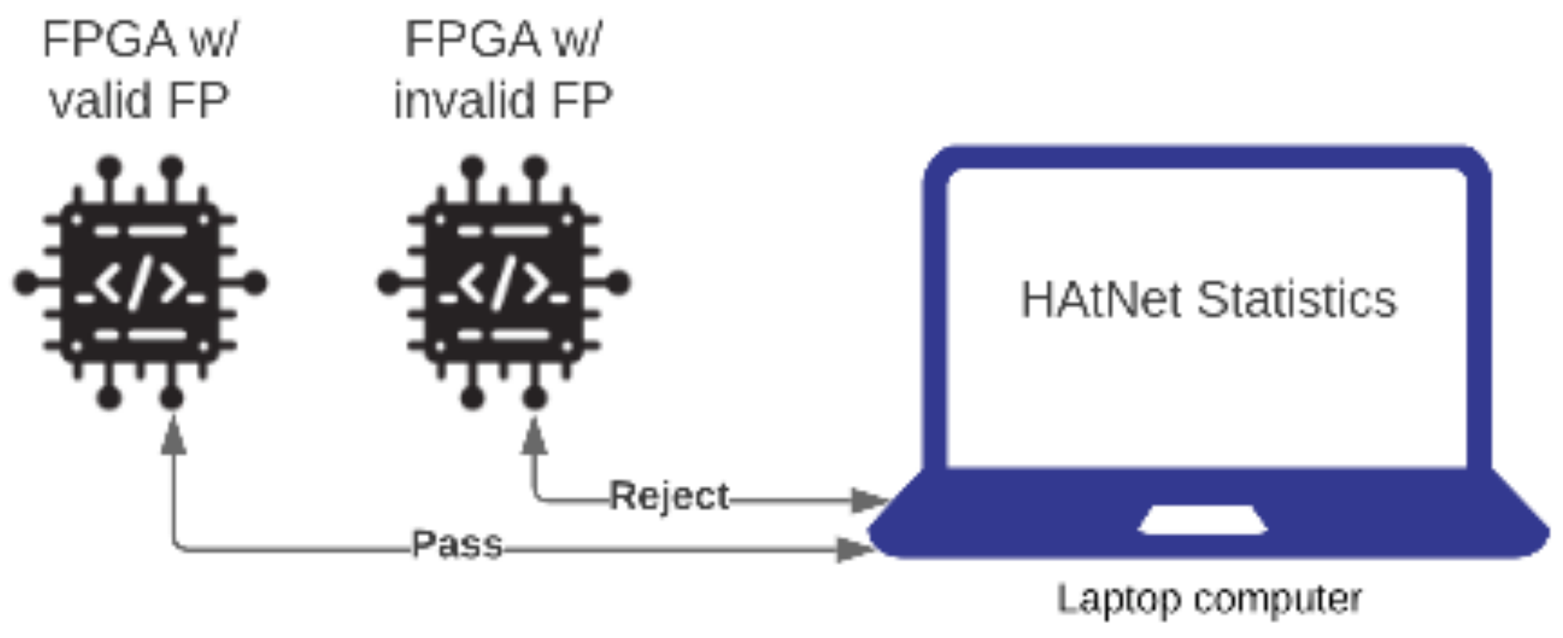
○ The attacker could be a malicious user who wants to run the DNN on an unauthorized hardware platform

❖ HAtNet generates device-specific fingerprint (FP) and **binds device's FP** to the DNN by embedding the FP in the **weight distribution** of the DL model



Experimental Results

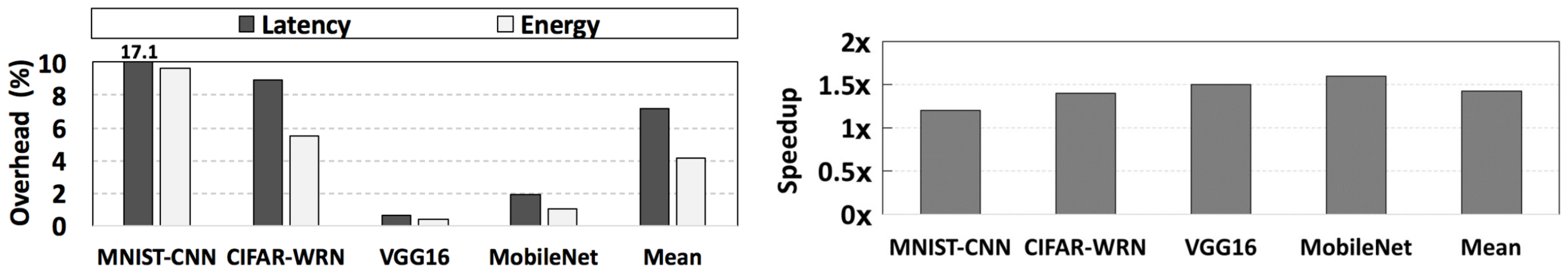
❖ We evaluate HAtNet on various DNN benchmarks to corroborate its properties:



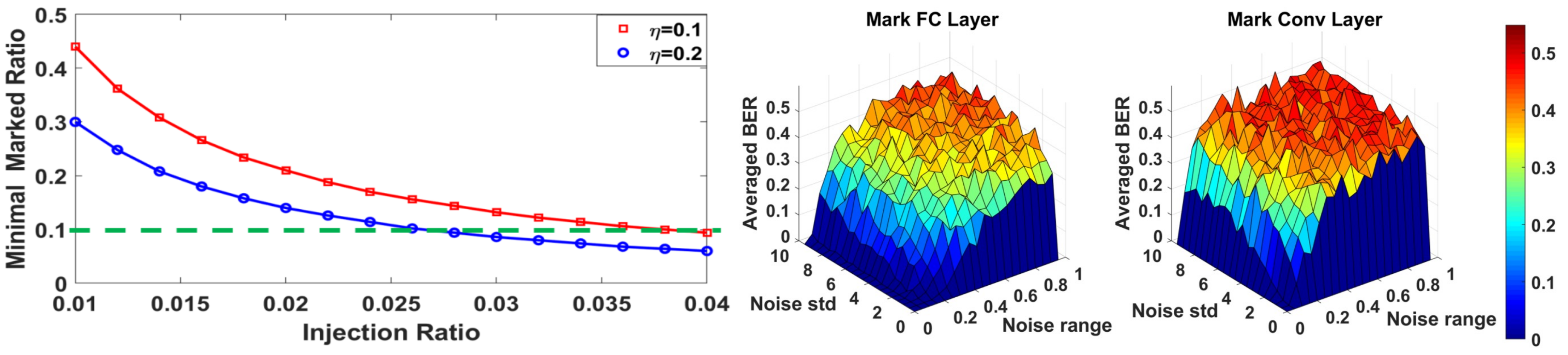
○ **Fidelity:** Test accuracy of the marked DNN is comparable as the corresponding baseline

Benchmark	Dataset	Model Size (MB)	Multiply-Add Operations (Mops)	Marked Layer Size (MB)	Baseline Accuracy (%)	Marked Accuracy (%)
MNIST-CNN	MNIST [34]	1.3	24	0.13 (10.1%)	99.52	99.66
CIFAR-WRN	CIFAR10 [35]	2.4	198	0.29 (12.3%)	91.85	92.03
VGG16	ImageNet [36]	276.7	25180	28.3 (10.2%)	91.2	92.23
MobileNet	ImageNet [36]	8.4	569	1.05 (12.6%)	85.83	85.75

○ **Efficiency:** (a) Low runtime and energy overhead of online attestation, (b) data pipeline speedup



○ **Security and Reliability analysis**:



Conclusion

❖ Devising HAtNet, an effective, lightweight, reliable and secure on-device attestation framework that authenticates the legitimacy of the hardware to run the protected DL model

❖Leveraging Algorithm/Software/Hardware **co-design** principle to achieve **hardware-bounded IP protection** and **device usage control** of DL hardware.