

Hardware/Software Co-Design of Deep Learning Accelerators

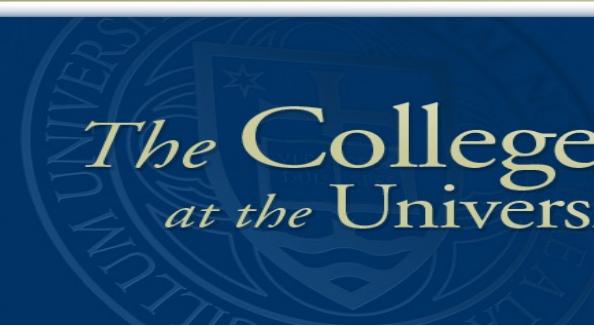
Yiyu Shi, Ph.D.

Professor, Dept. of Computer Science and Engineering,

Site Director, NSF I/UCRC on Alternative and Sustainable Intelligent Computing,

University of Notre Dame

yshi4@nd.edu



*The College of Engineering
at the University of Notre Dame*



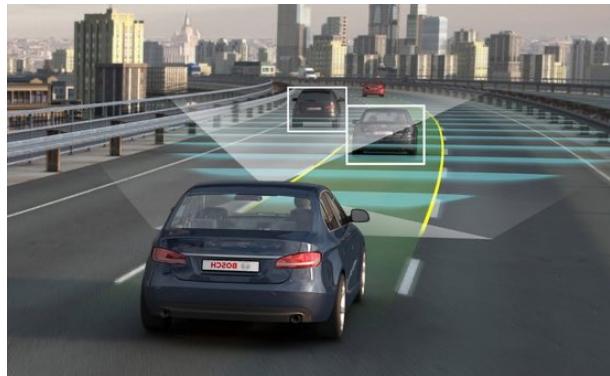
The Prevalence of AI



Disease diagnose



Game play



Autonomous driving



Real-time translation

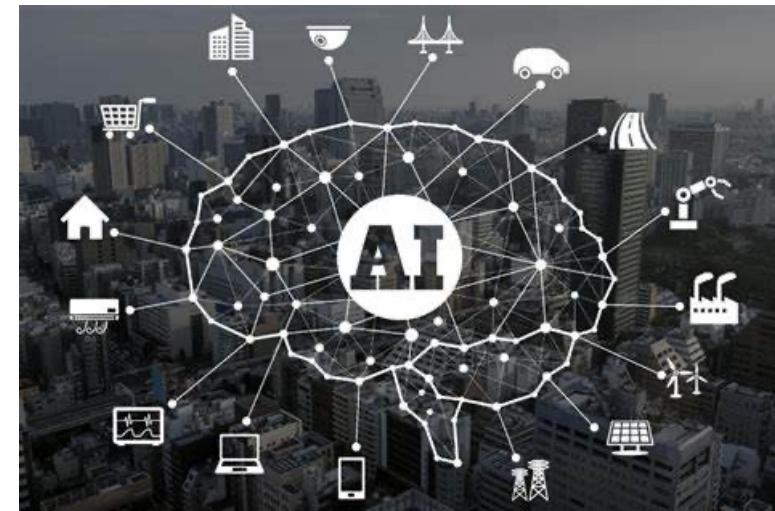
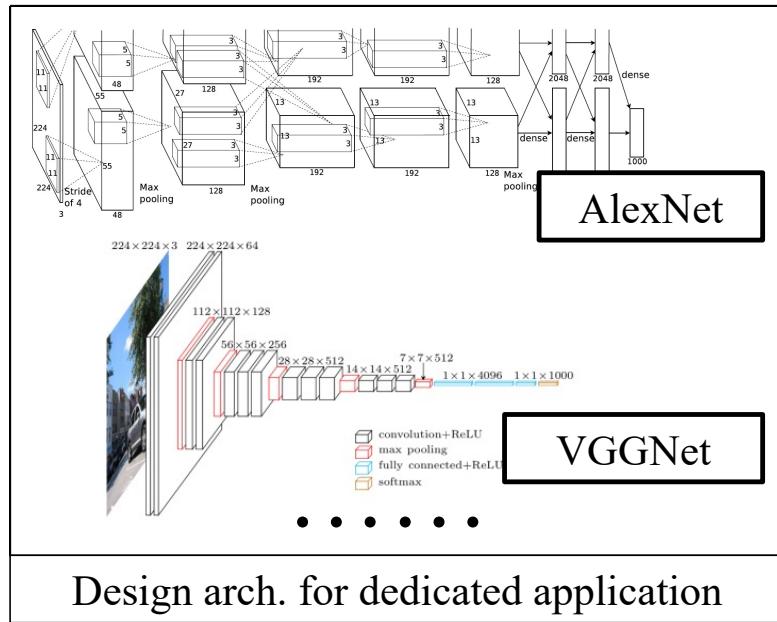
1



*The College of Engineering
at the University of Notre Dame*



Human Invented Neural Architectures

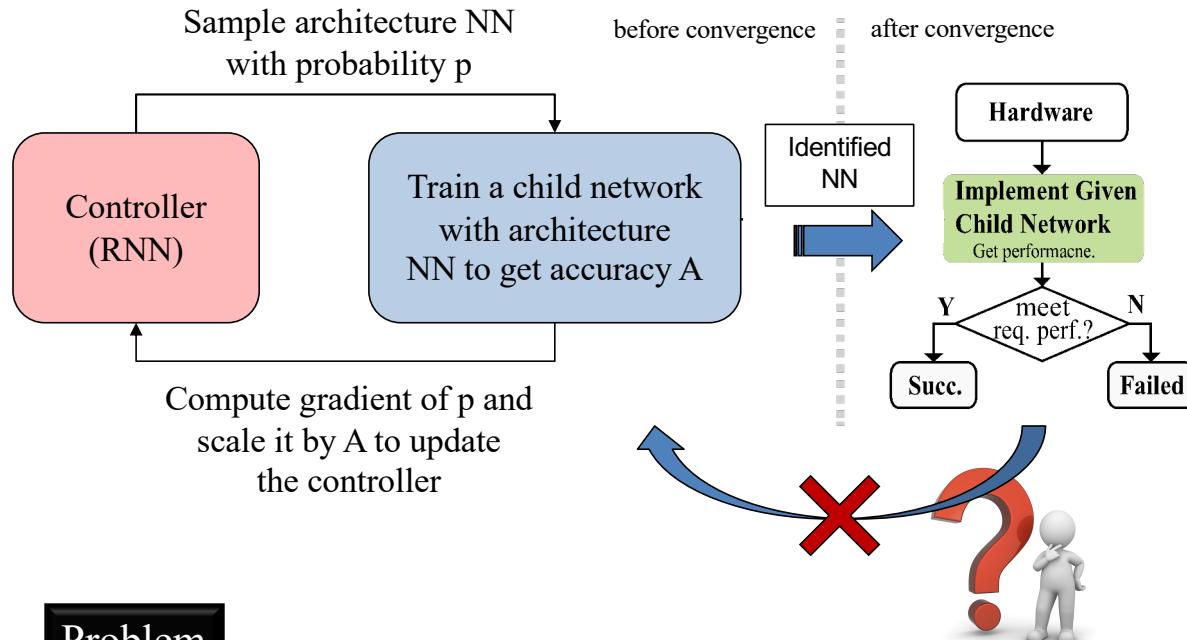


Era of AI Democratization

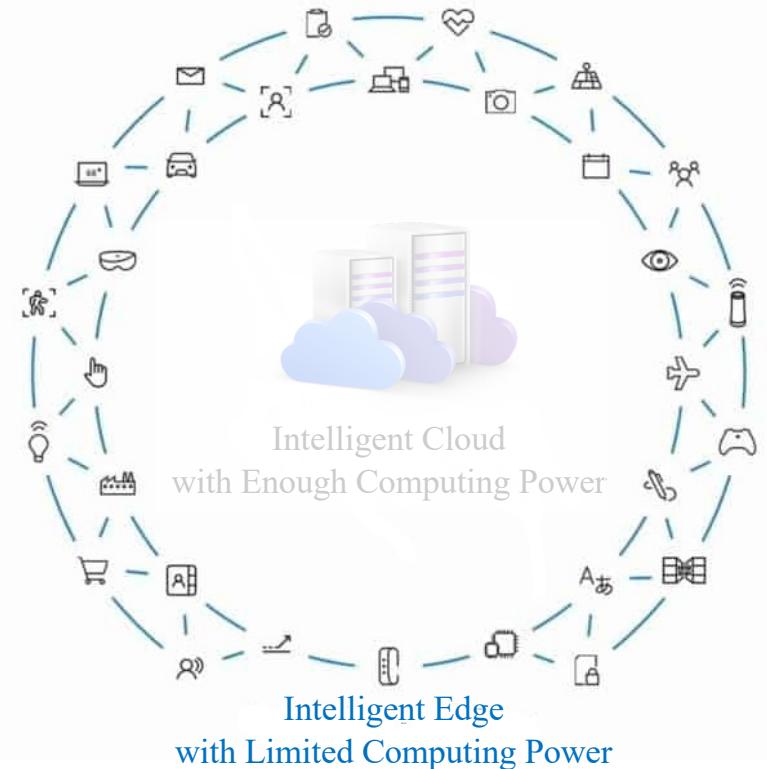
Problem

- Domain knowledge and excessive labor
- It is impossible to manually design specific arch. for each dedicated application in the era of AI democratization

Neural Architecture Search (NAS)

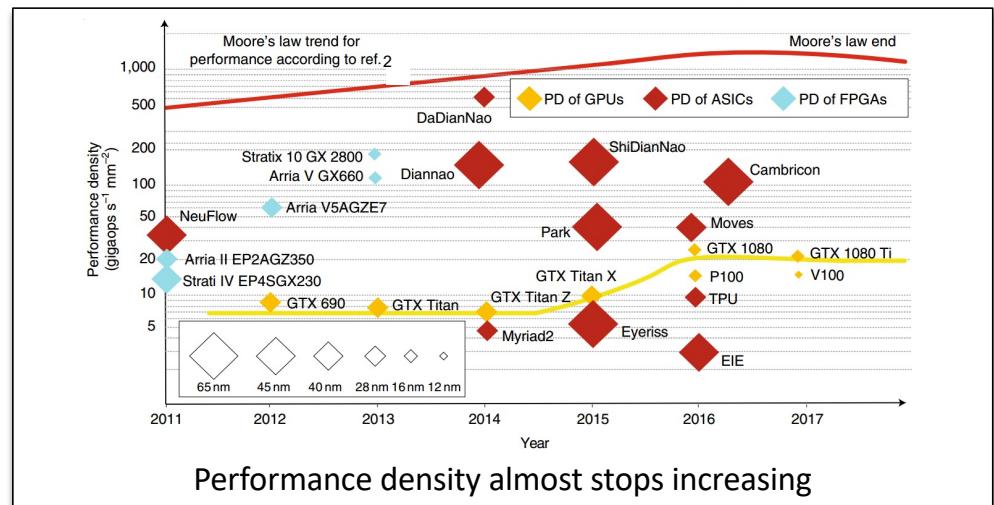
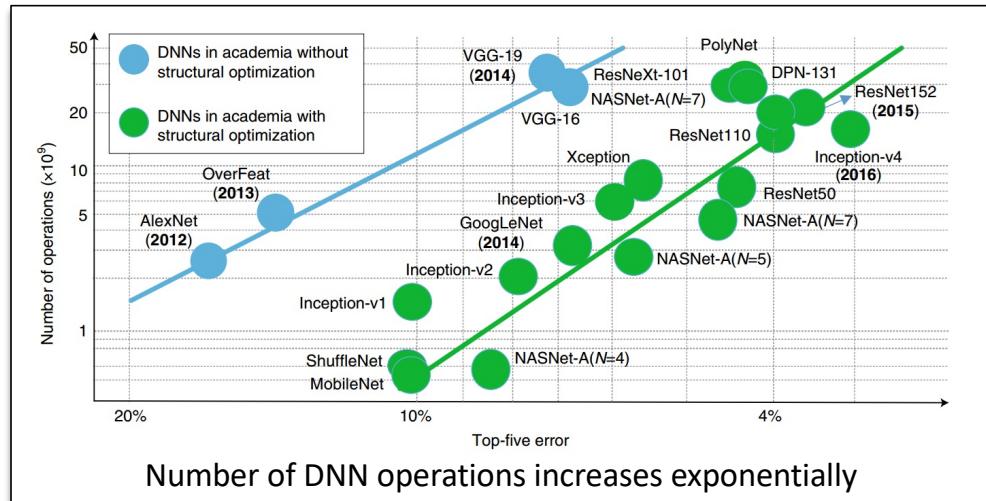


- No constraint on hardware resource consumed



REF: Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." *ICLR 2017*

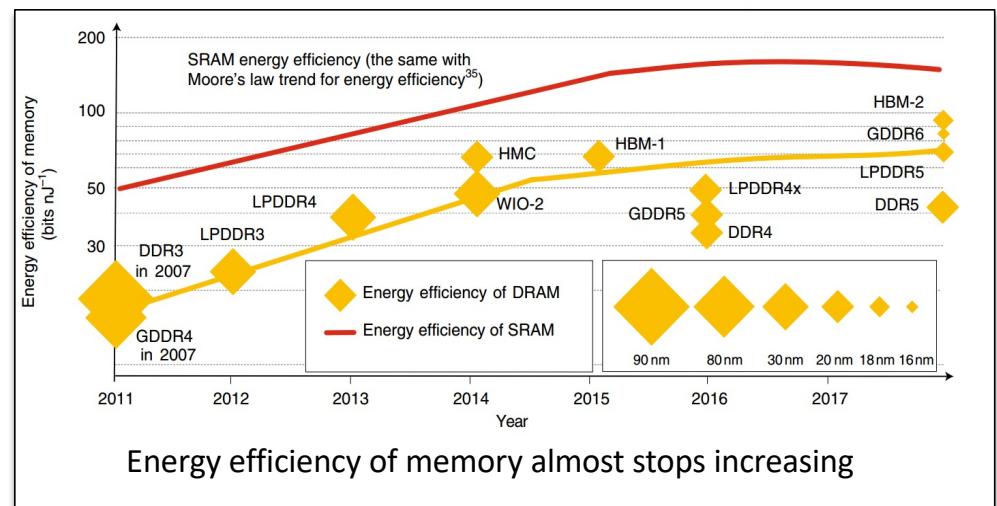
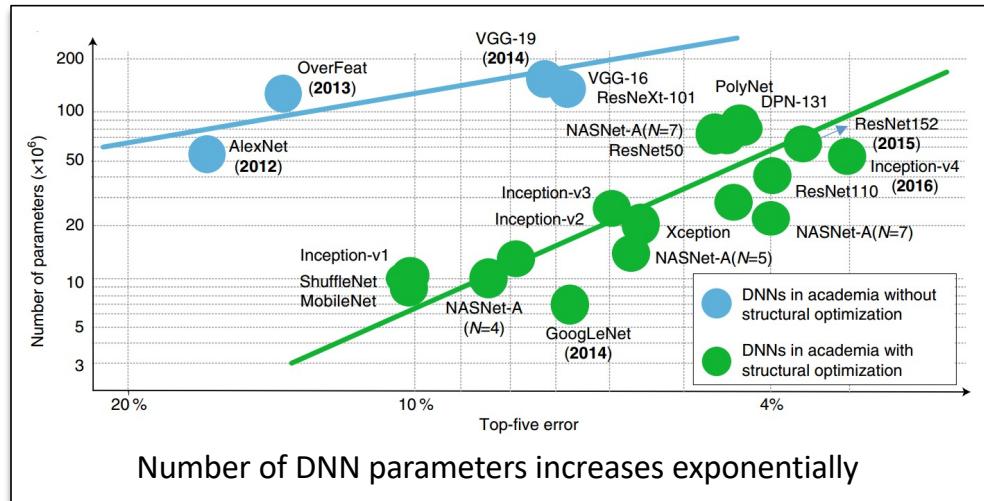
Gap Between Neural Networks and Hardware Accelerators



Xiaowei Xu, Yukun Ding, Sharon Hu, Michael Niemier, Jason Cong, Yu Hu and Yiyu Shi, "Scaling of Deep Neural Networks for Edge Inference: A Race between Data Scientists and Hardware Architects", Nature Electronics 1, pp. 216-222, 2018.



Gap Between Neural Networks and Hardware Accelerators



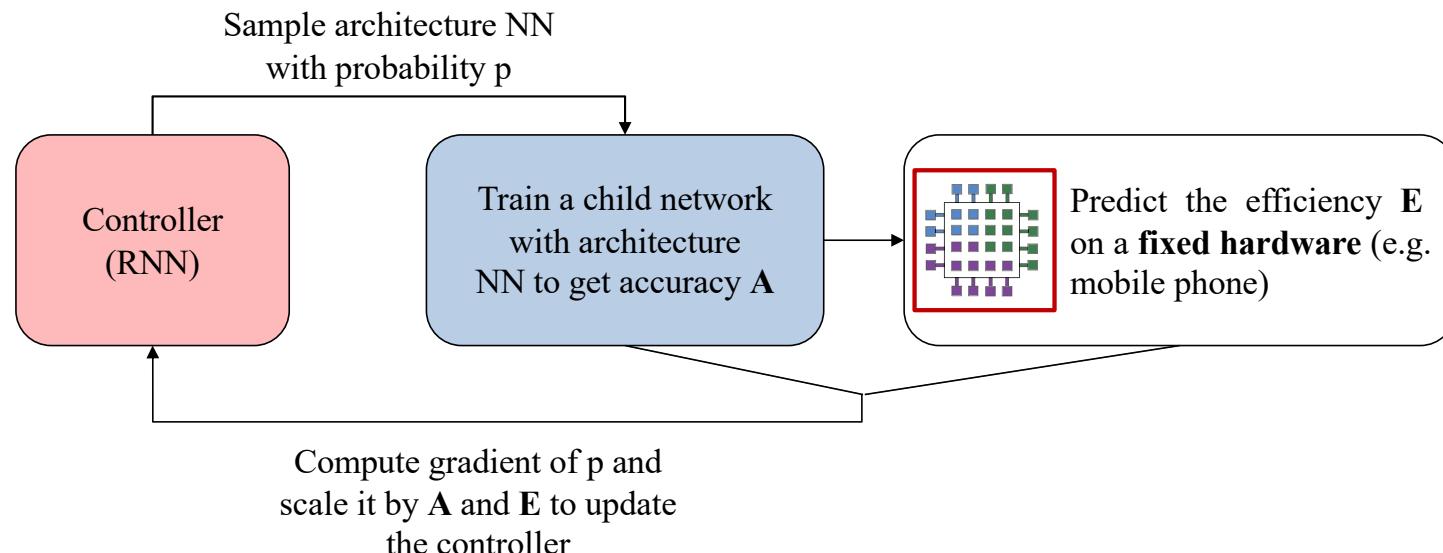
Xiaowei Xu, Yukun Ding, Sharon Hu, Michael Niemier, Jason Cong, Yu Hu and Yiyu Shi, "Scaling of Deep Neural Networks for Edge Inference: A Race between Data Scientists and Hardware Architects", Nature Electronics 1, pp. 216-222, 2018.

5

The College of Engineering
at the University of Notre Dame

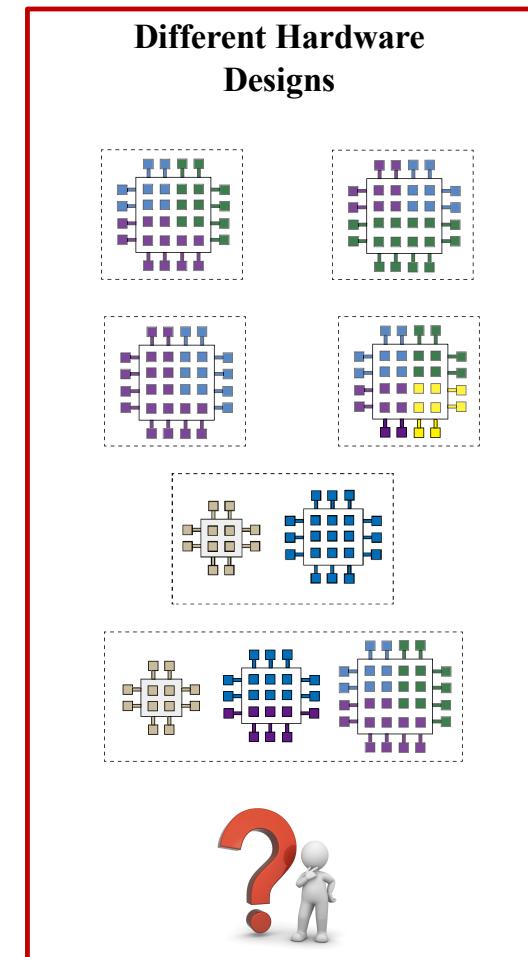


Hardware-Aware NAS



Problem

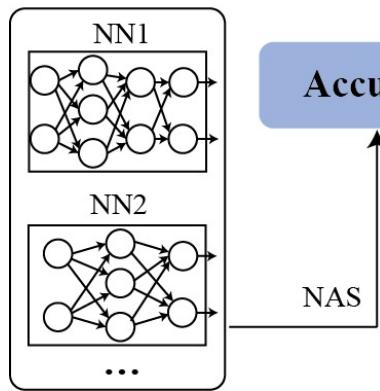
- **It works for particular fixed hardware, but not suitable for programmable hardware**



REF: Mingxing Tan, et al. "MnasNet: Platform-Aware Neural Architecture Search for Mobile." arXiv 2018

A Missing Link between Two Design Spaces

Arch Search Space

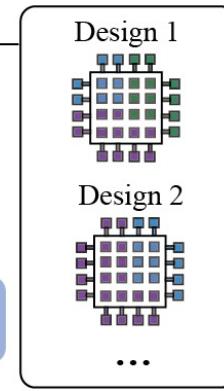


Accuracy

NAS

Neural Architecture Search

FPGA Design Space



Implementation & Optimization

Efficiency

Neural Architecture Implementation on Hardware

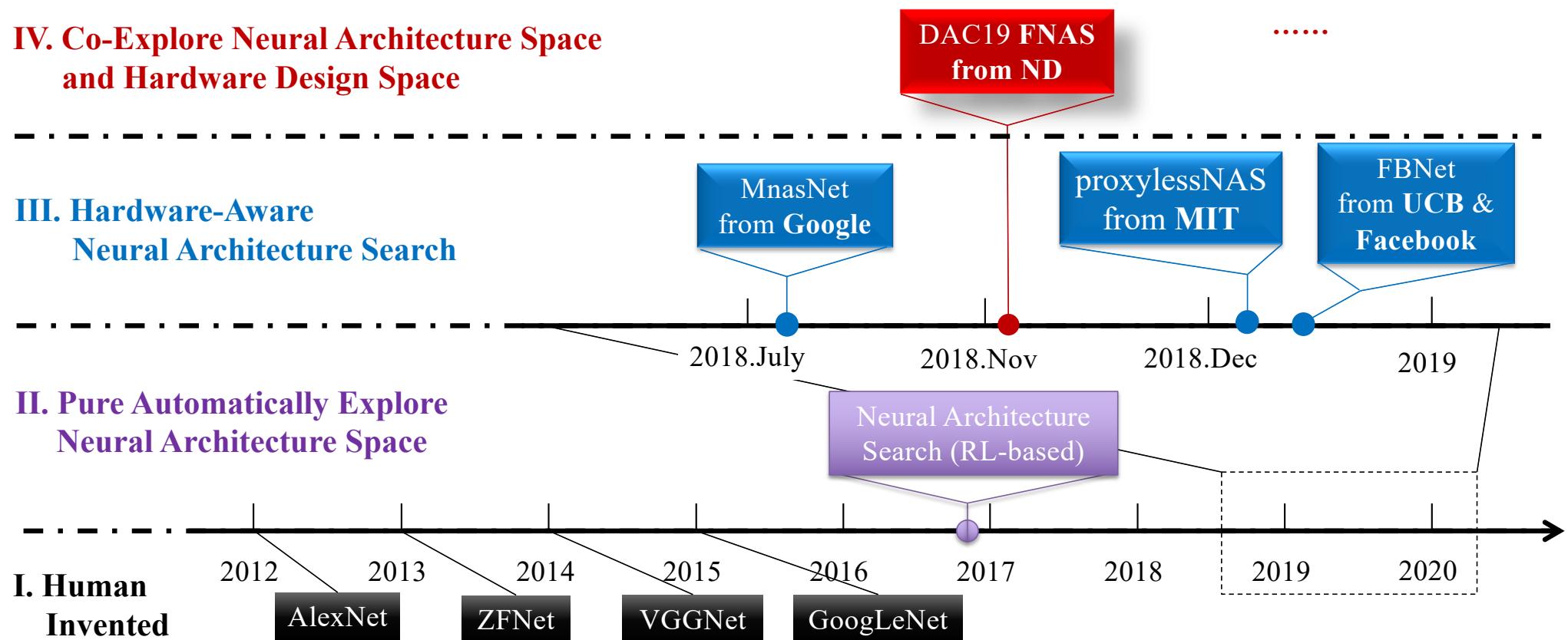


The College of Engineering
at the University of Notre Dame



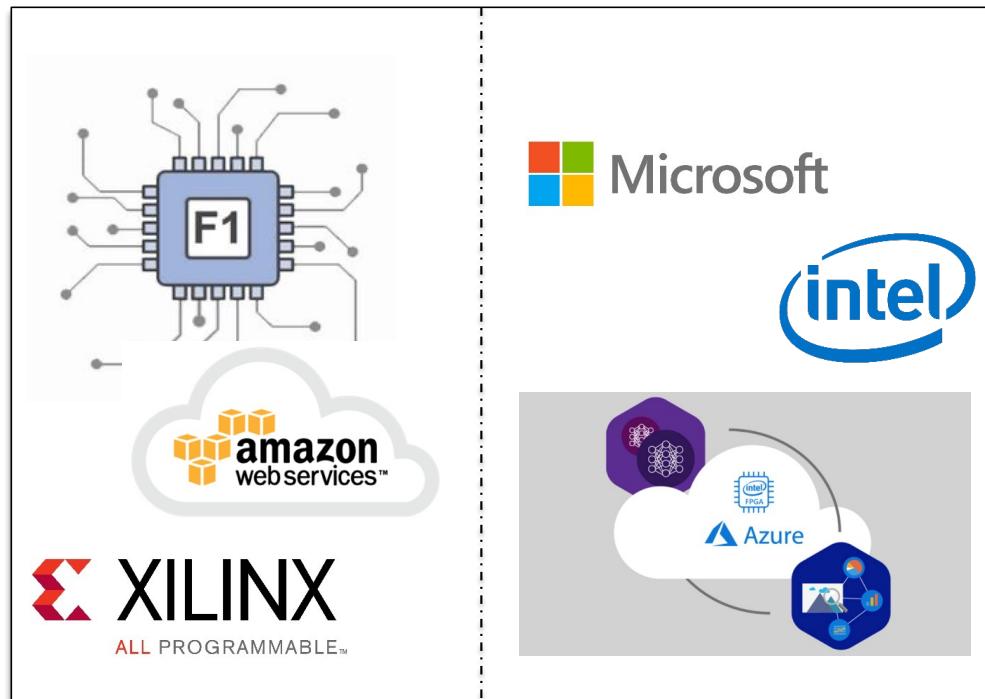
Evolution of Exploring Deep Neural Architectures

IV. Co-Explore Neural Architecture Space and Hardware Design Space

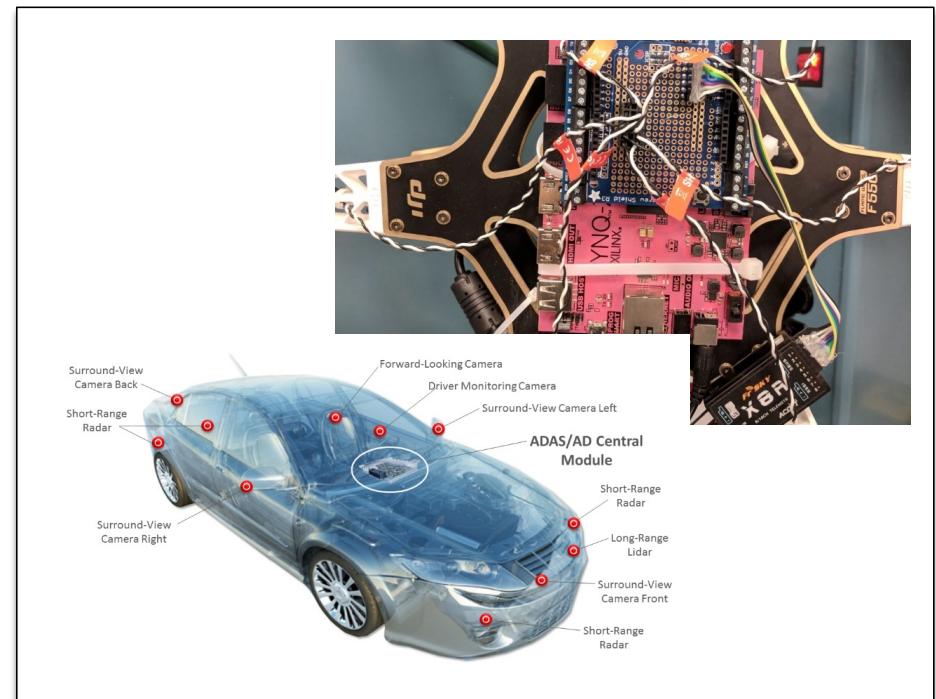


FPGAs in DNN Applications

FPGA in Cloud Computing



FPGA in Edge Computing

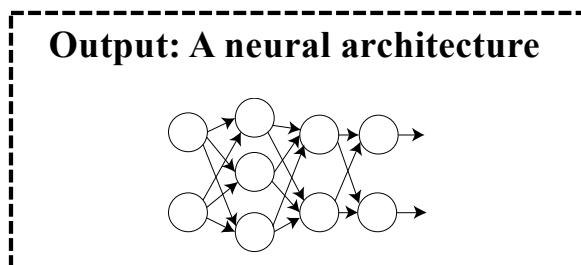
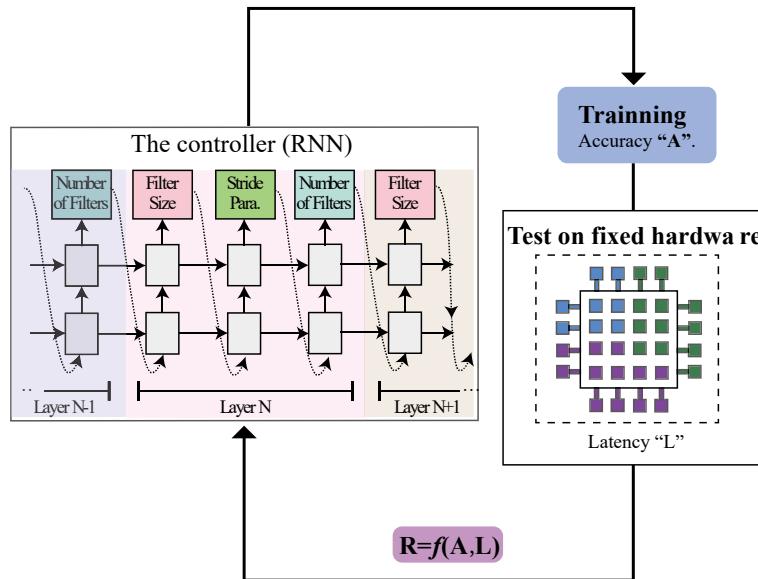


*The College of Engineering
at the University of Notre Dame*

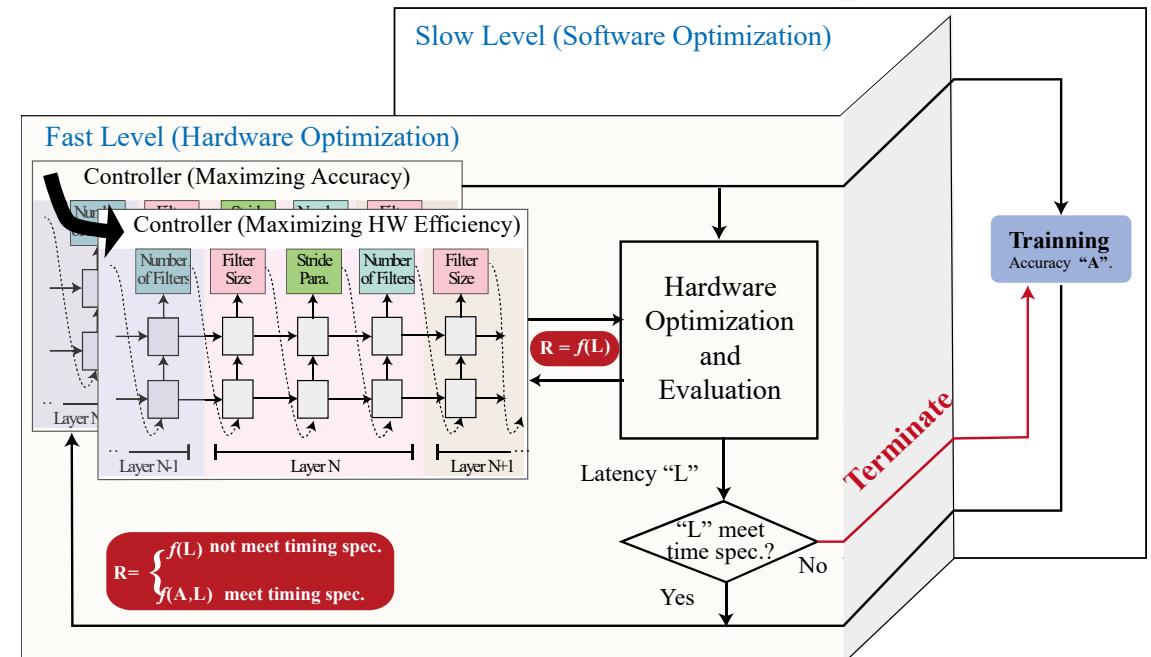
[ref] Where Do FPGAs Stand in Auto IC Race? https://www.eetimes.com/document.asp?doc_id=1333419#

[ref] PYNO in UAV <http://brennancain.com/pynocopter-an-open-source-fpga-overlay-for-uavs/>

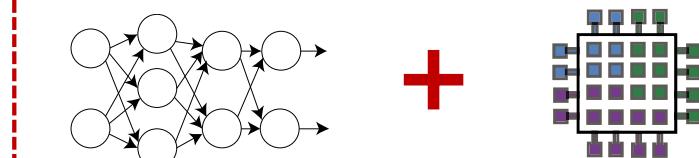
HW-Aware NAS vs. FPGA/Neural Architecture Co-Design (FNAS)



HW-Aware NAS

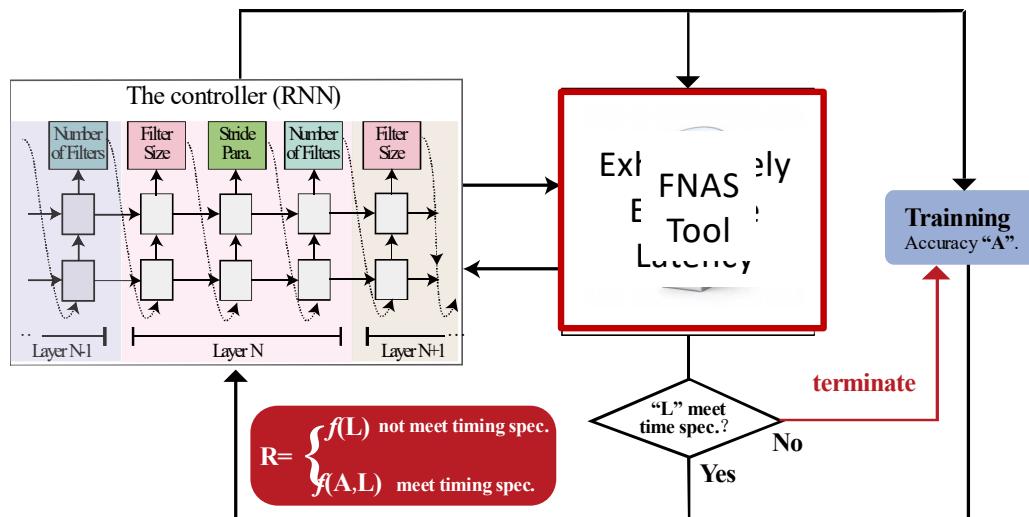


Output: A pair of neural architecture and hardware design



FNAS

Solutions & Challenges



Our Solution: FNAS tools to respond to challenges

FNAS-Design **C1**
“Design on Program Logic”

FNAS-GG **C2**
“Tile-based Task Graph Generator”

FNAS-Sched **C2**
“Scheduler on Processing System”

FNAS-Analyzer **C3**
Estimate Performance “L”

Naïve Solution: HW-Aware + Exhaustively Evaluate Lat.

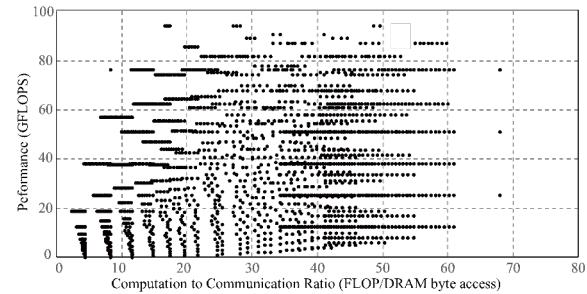


Fig1. Possible designs for Layer 5 of AlexNet on ZCU102

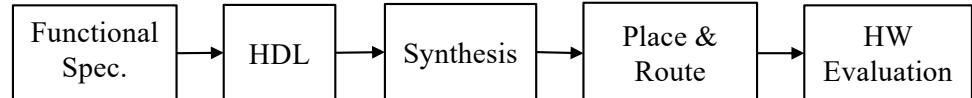


Fig2. Procedure of performance evaluation

Challenges:

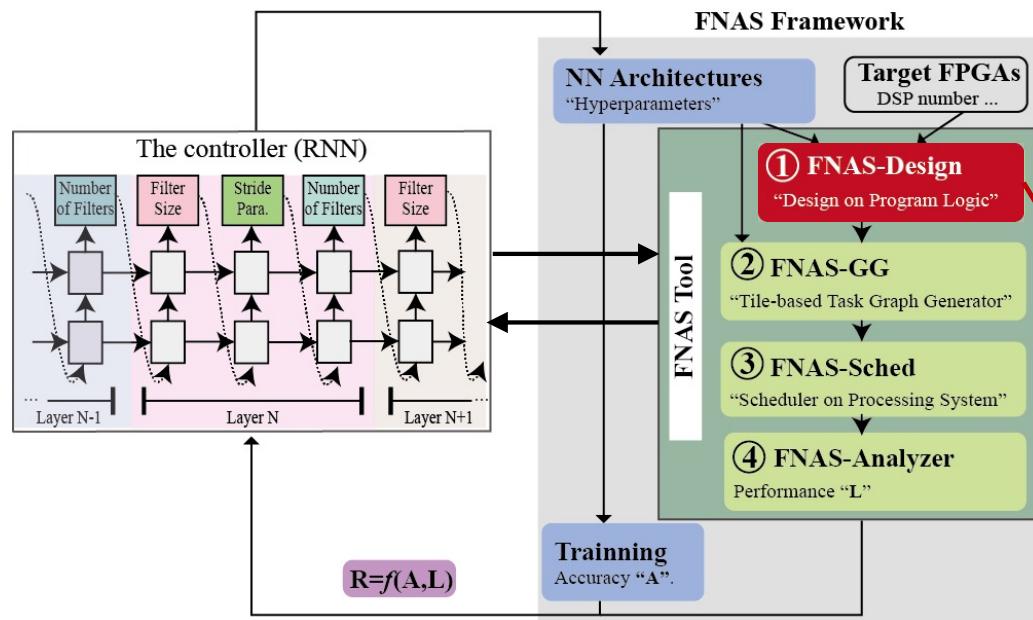
C1: Huge design space!

C2: Multi-FPGA design!

C3: Time-consuming evaluation!

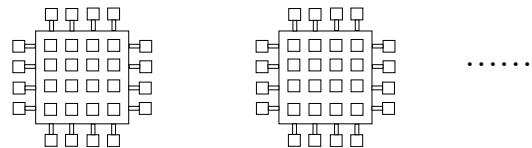
Infeasible

FNAS: Design Optimization (on-chip design)



Given:

1. FPGAs with attributes including LUTs, DSPs, BRAM, etc.

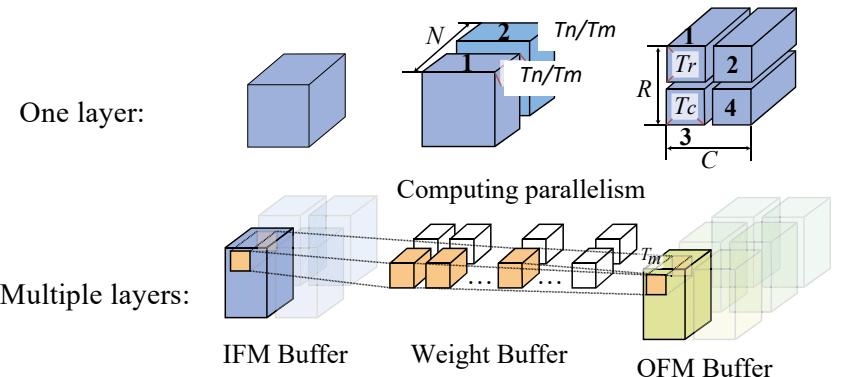


2. A neural architecture with determined hyperparameters

On-chip accelerator design:

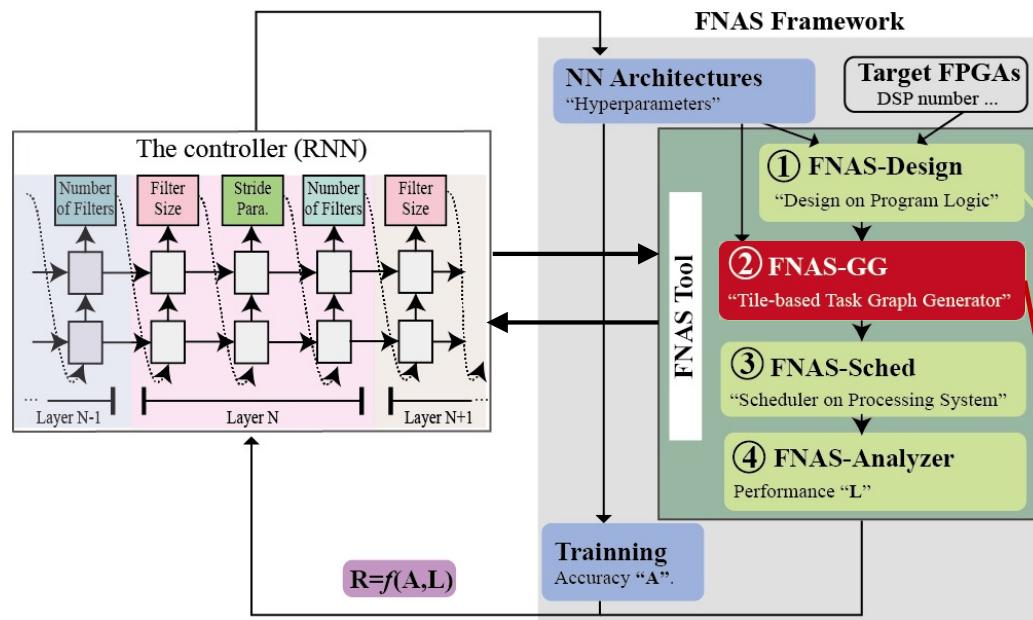
Determine:

1. On-chip buffer allocation; 2. Accelerator size for computing
(note: both are determined by tiling parameters, T_m , T_n , T_r , T_c)



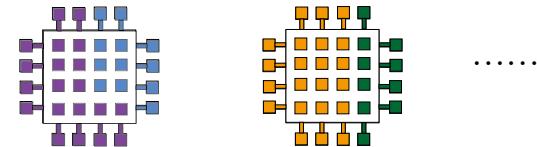
REF: Chen Zhang et al. 2015. Optimizing fpga-based accelerator design for deep convolutional neural networks. In Proc. of FPGA.

FNAS: Graph Generator

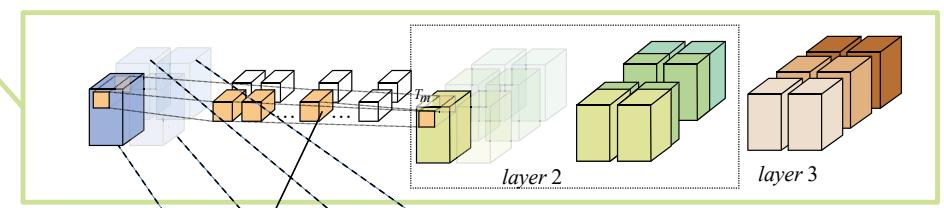


Given:

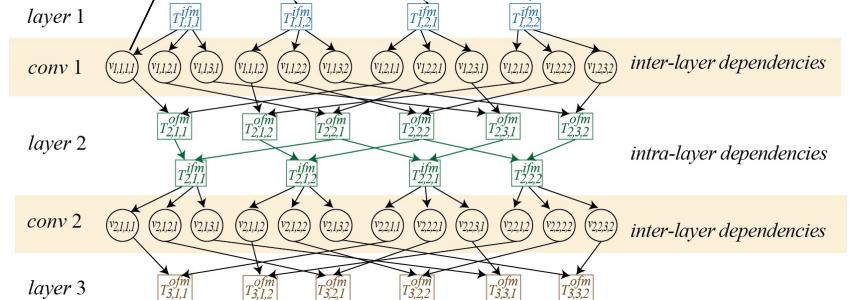
1. FPGAs with attributes including LUTs, DSPs, BRAM, etc.



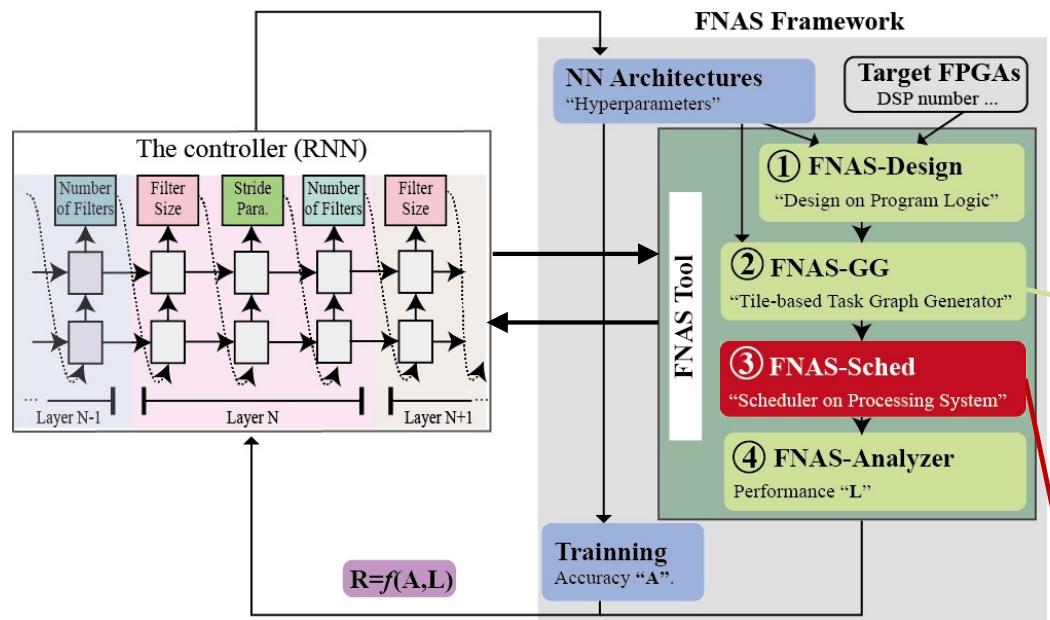
2. A neural architecture with determined hyperparameters



High-level graph abstraction

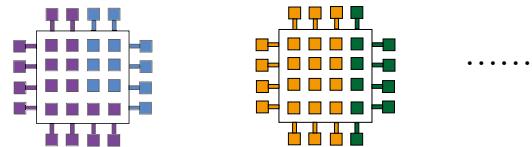


FNAS: Schedule (off-chip design)

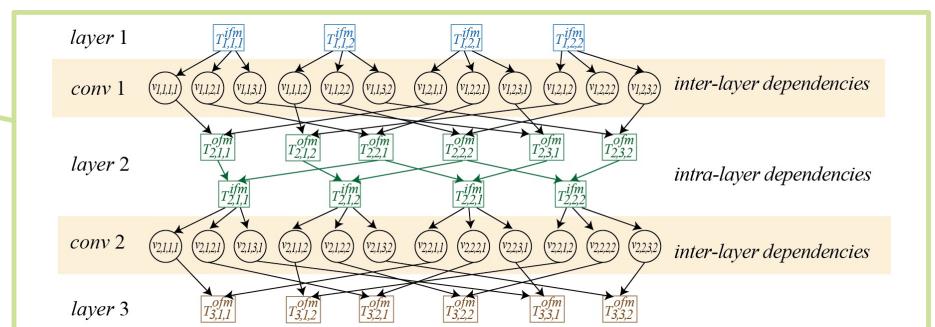


Given:

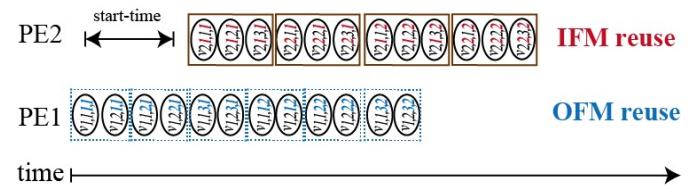
1. FPGAs with attributes including LUTs, DSPs, BRAM, etc.



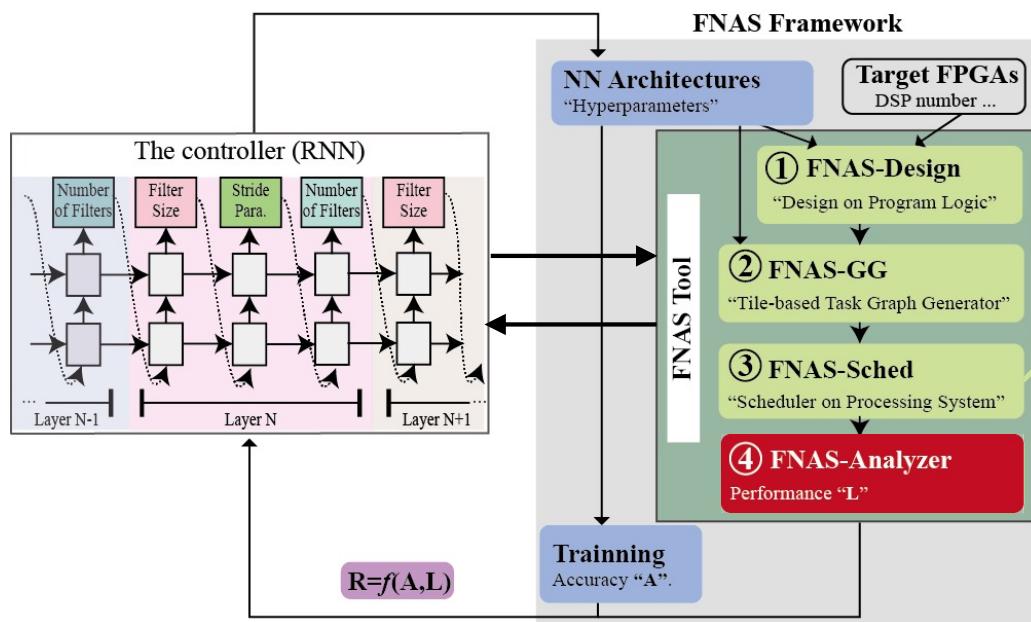
2. A neural architecture with determined hyperparameters



Schedule of tasks in graph on multiple FPGAs

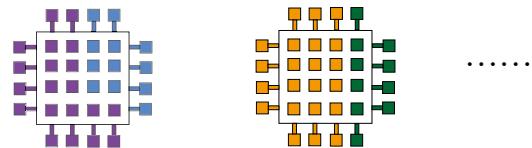


FNAS: Analyzer

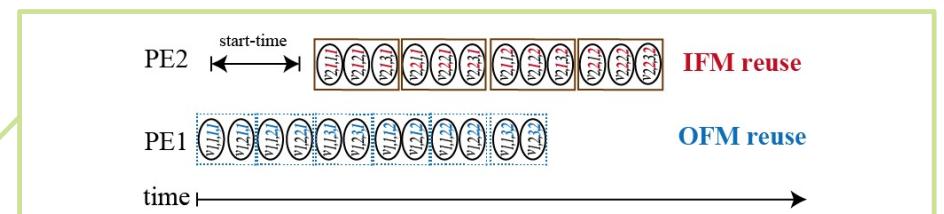


Given:

1. FPGAs with attributes including LUTs, DSPs, BRAM, etc.



2. A neural architecture with determined hyperparameters



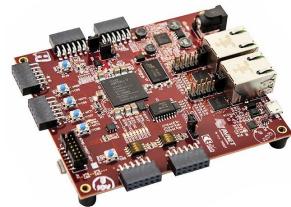
$$\text{Latency} = \text{pipeline start time} + \text{processing time}$$

Output:

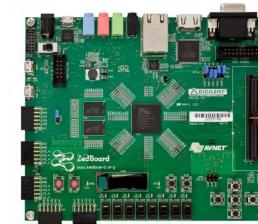
1. A tailored FPGA Design
2. The system latency

Experimental Setting

FPGAs

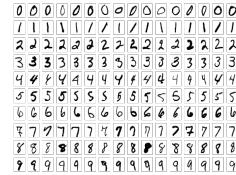


Xilinx 7A50T



Xilinx 7Z020

Datasets



MNIST



CIFAR-10



ImageNet

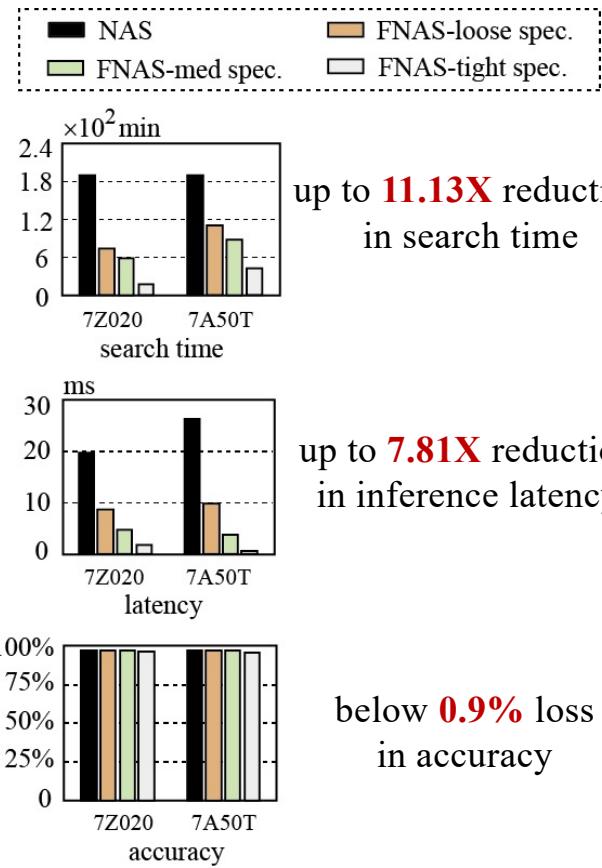
	Layer Num.	up to 5	up to 10	up to 15
NAS Search Space	Filter Size	[5, 7, 14]	[1, 3, 5, 7]	[1, 3, 5, 7]
	Filter Num.	[9, 18, 36]	[24, 36, 48, 64]	[16, 32, 64, 128]

HW Search Space Channel Tiling Para. (Tm,Tn); Row Tiling Para. (Tr); Col Tiling Para. (Tc); Schedule

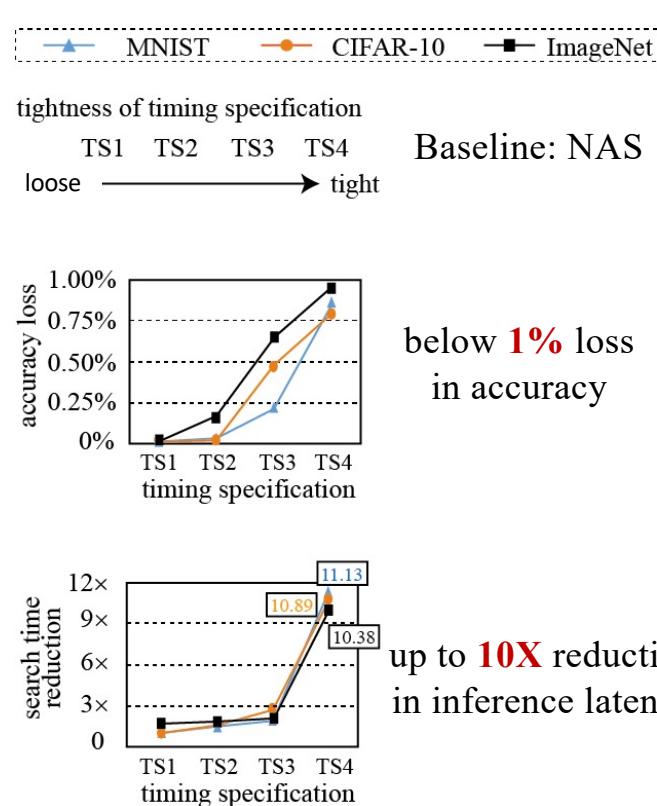
Timing Spec. (ms) [2, 5, 10, 20] [1.5, 2, 2.5, 10] [2.5, 5, 7.5, 10]

Experimental Results

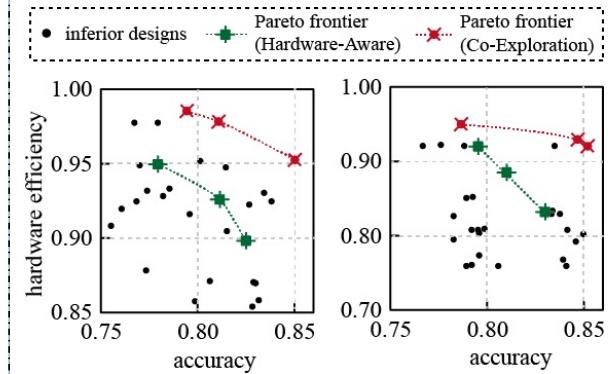
Different Hardware (MNIST)



Different Datasets (7Z020)



Compare to HW-Aware NAS (CIFAR-10 + 7Z020)



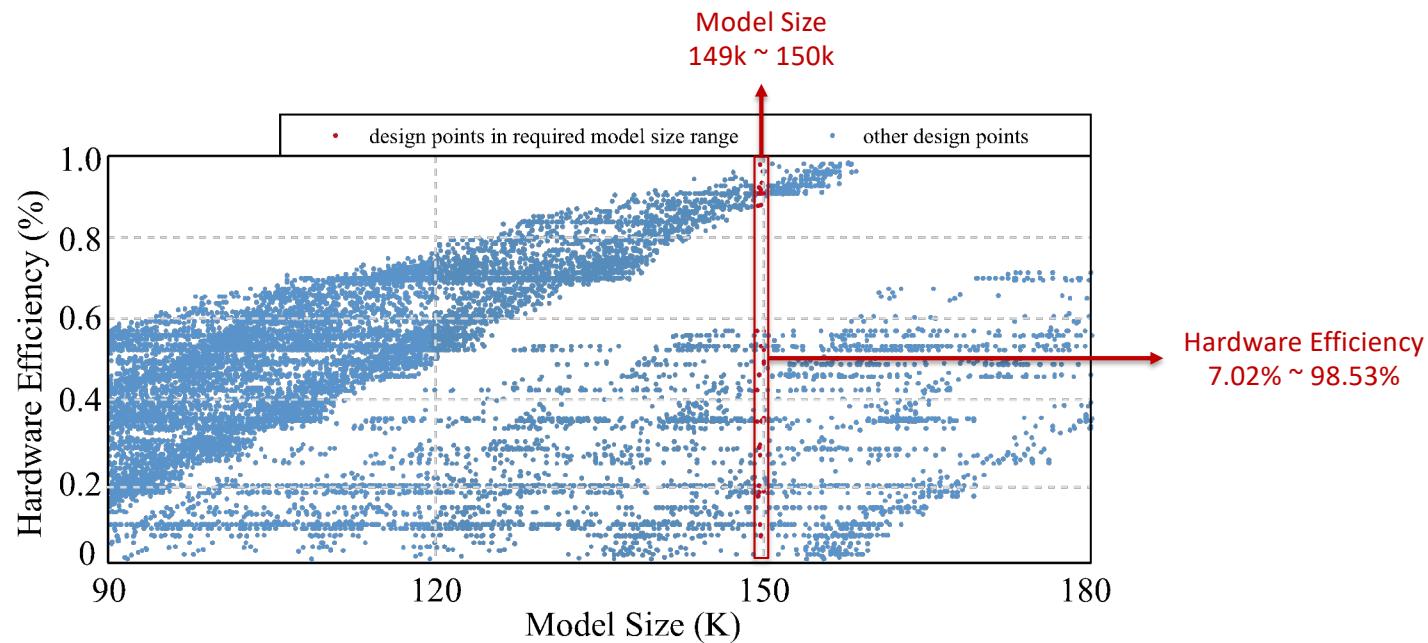
FNAS can significantly
push forward
the Pareto frontiers between
accuracy and efficiency
tradeoff

Experimental Results: Superior to Existing Approaches

Optimizing Hardware Efficiency								
Comparison the proposed Co-Exploration with Hardware-Aware NAS and Heuristic Sequential Optimization								
Dataset	Models	Depth	Parameters	Accuracy (Top1)	Accuracy (Top5)	Pipeline Eff.	FPS	Energy Eff. GOPS/W
CIFAR-10	Hardware-Aware NAS	13	0.53M	84.53%	-	73.27%	16.2	0.84
	Sequential Optimization	13	0.53M	84.53%	-	92.20%	29.7	1.36
	Co-Exploration (OptHW)	10	0.29M	80.18%	-	99.69%	35.5	2.55
	Co-Exploration (OptSW)	14	0.61M	85.19%	-	92.15%	35.5	1.91
ImageNet	Hardware-Aware NAS	15	0.44M	68.40%	89.84%	81.07%	6.8	0.34
	Sequential Optimization	15	0.44M	68.40%	89.84%	86.75%	10.4	0.46
	Co-Exploration (OptHW)	17	0.54M	68.00%	89.60%	96.15%	12.1	1.01
	Co-Exploration (OptSW)	15	0.48M	70.24%	90.53%	93.89%	10.5	0.74

Optimizing Network Accuracy

Experimental Results: Importance of Co-Exploration

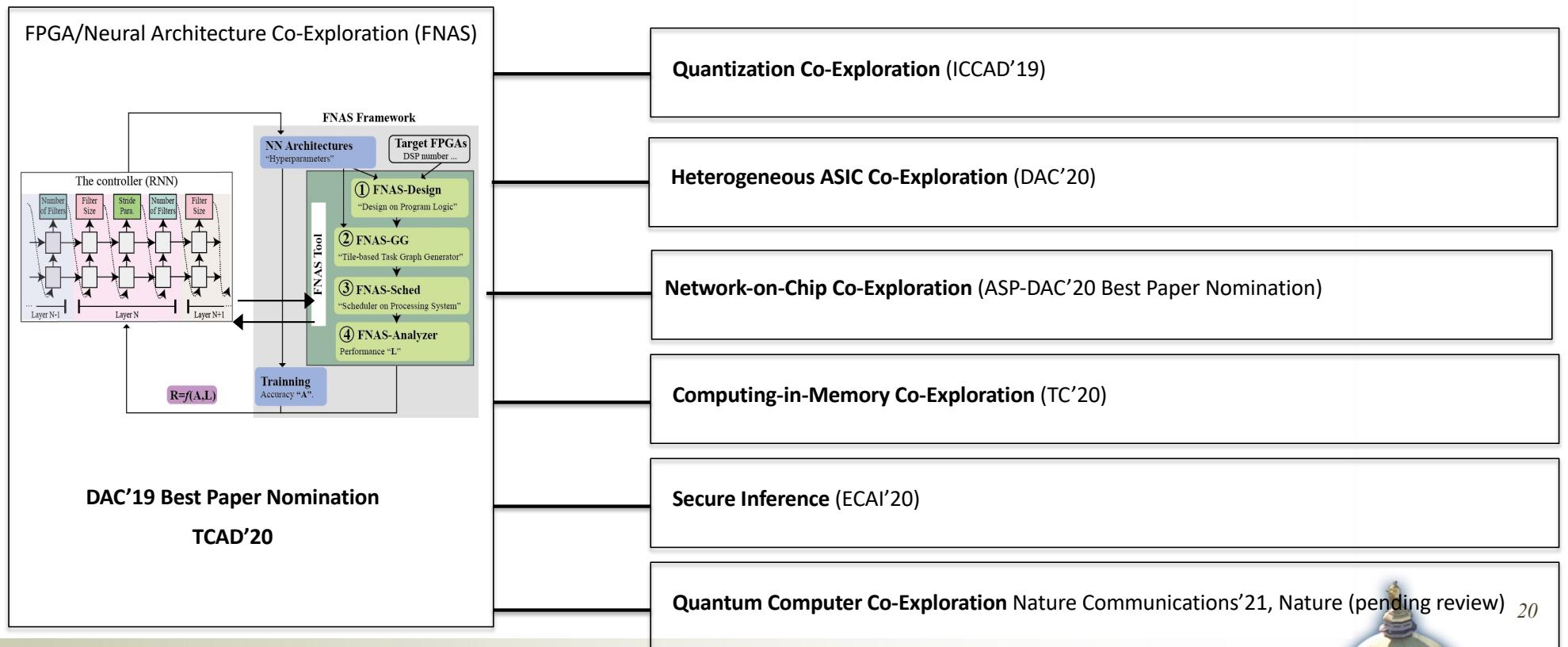


In the design space:

Models with similar model sizes may have distinct hardware efficiency

=> **Cannot restrict model size** to guarantee **hardware efficiency**

Co-Exploration of Neural Architectures

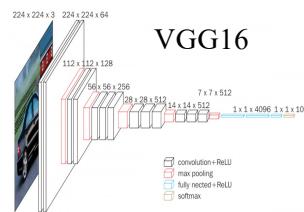


*The College of Engineering
at the University of Notre Dame*

QuanNAS: Architecture-Hardware-Quantization Co-Exploration

Motivation

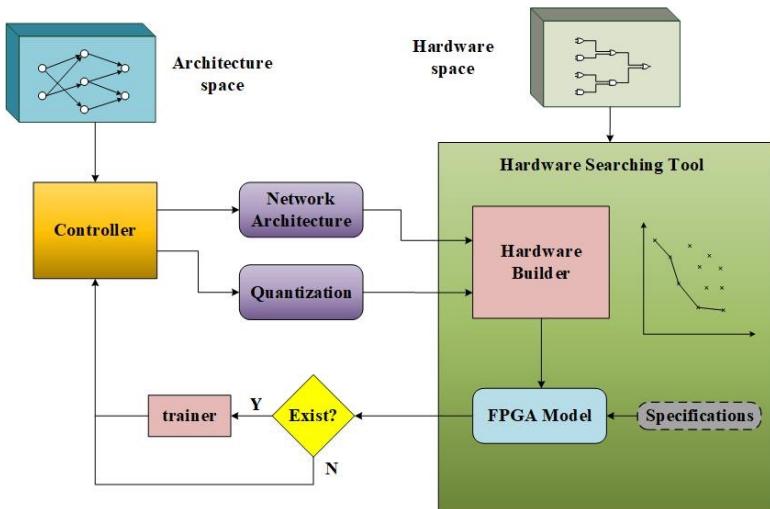
Layer	Activation
CONV3-64	96Mb
CONV3-128	51.2Mb
CONV3-256	25.6Mb
CONV3-512	12.8Mb



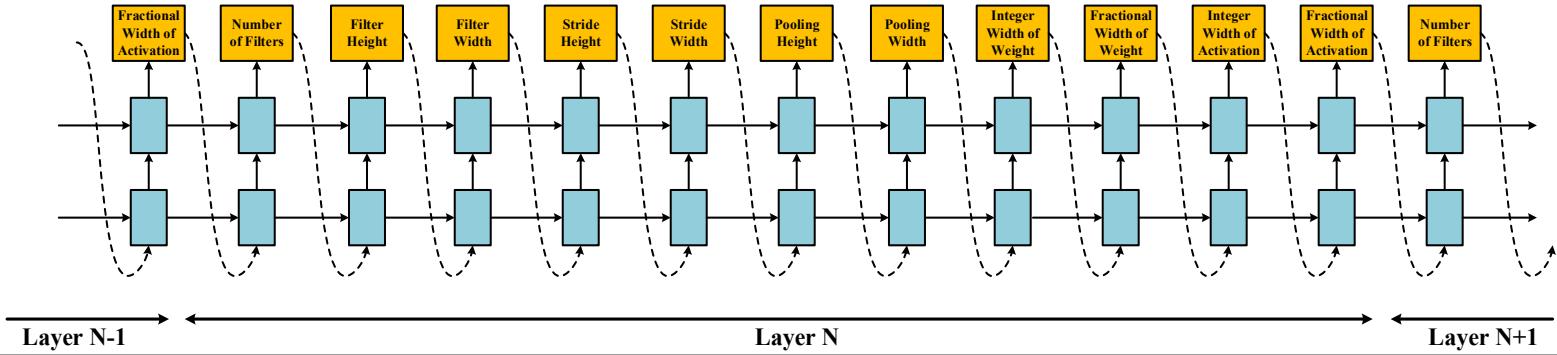
FPGA	Memory	DSP
ZCU102	32.1Mb	2520
ZC706	19.1Mb	900
GT900	47Mb	1518/3036
GX320	17Mb	985/1970



Co-Exploration Framework



Controller



QuanNAS Results

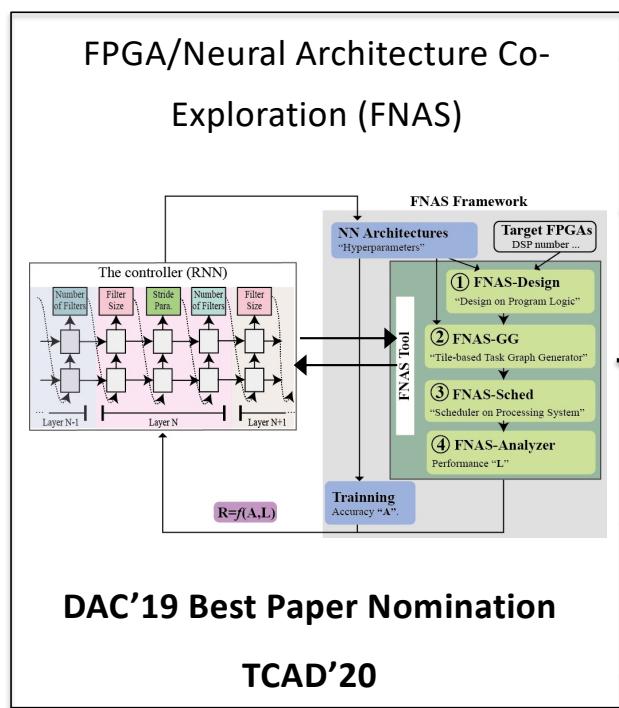
Table 3: Implementation information of the sampled designs. For network A and B, the designs are found by quantization search to certain architectures in Table 2. For D, E and F, the quantization and implementation on hardware are designed together with their architectures. The quantization details are shown in Figure 4.

Design	rL	rT	Acc w/o quantization	Acc w/ quantization	#LUTs	Throughput (frames/s)	parameter size (kbits)
A ₁ -d ₁	100,000	500	87.76%	80.23%	99,871	556	1,867
A ₁ -d ₂	100,000	1000	87.76%	25.79%	99,848	1157	1,189
B ₁ -d ₁	100,000	500	89.71%	87.64%	96,904	512	3,463
B ₁ -d ₂	100,000	1000	89.71%	64.35%	98,752	1020	2,784
B ₁ -d ₃	300,000	2000	89.71%	50.93%	285,441	2083	2,835
D	30,000	1000	83.65%	82.98%	29,904	1293	457
E	100,000	1000	86.99%	82.76%	94,496	1042	1,923
F	300,000	2000	87.03%	84.92%	299,860	2089	1,217

Co exploration is more robust to quantization error

Hardware performance is maintained

Co-Exploration of Neural Architectures



Quantization Co-Exploration (ICCAD'19)

Heterogeneous ASIC Co-Exploration (DAC'20)

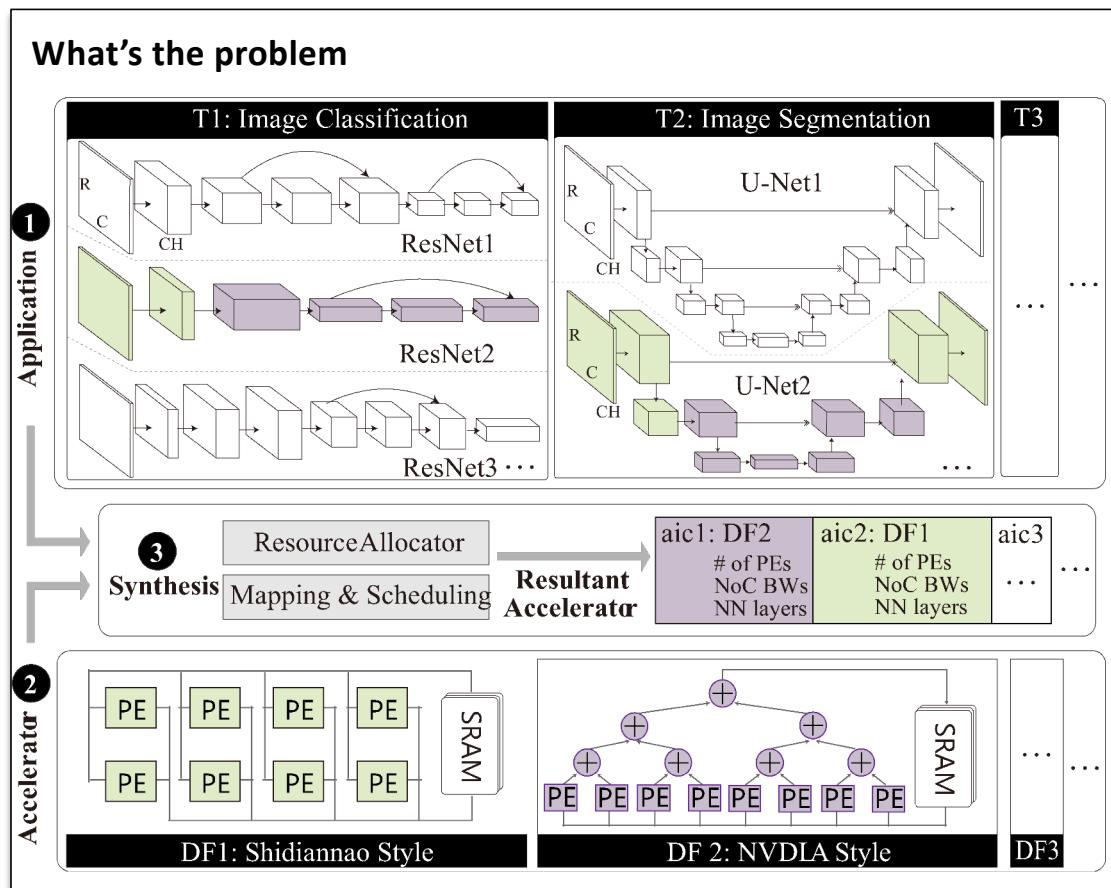
Network-on-Chip Co-Exploration (ASP-DAC'20 Best Paper Nomination)

Computing-in-Memory Co-Exploration (TC'20)

Secure Inference (ECAI'20)



NASAIC: NAS and Heterogeneous ASIC Accelerator Co-Exploration



Challenge1: ASIC has huge design space

Solution1: Create Template Pool to fix topology

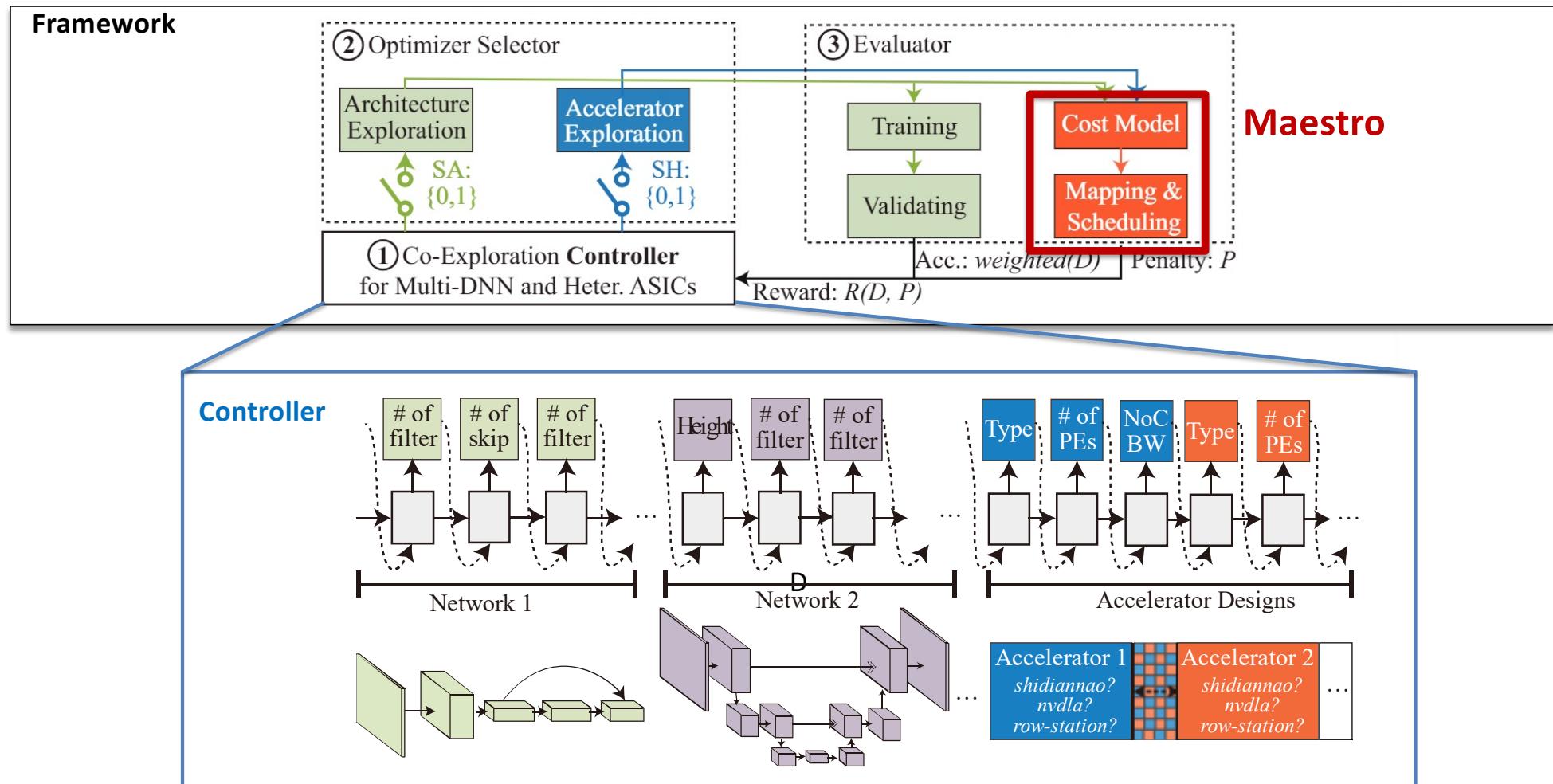
Challenge2: Multiple tasks in application

Solution2: Simultaneously search architectures

Challenge3: Performance Model

Solution3: Maestro

NASAIC: Exploration Flow



NASAIC: Results

Workloads

- one classification task on CIFAR-10 dataset
- one segmentation task on Nuclei dataset

Design Specifications

Latency: 8e5 cycles; **Energy:** 2e9 nJ; **Area:** 4e9 μm^2

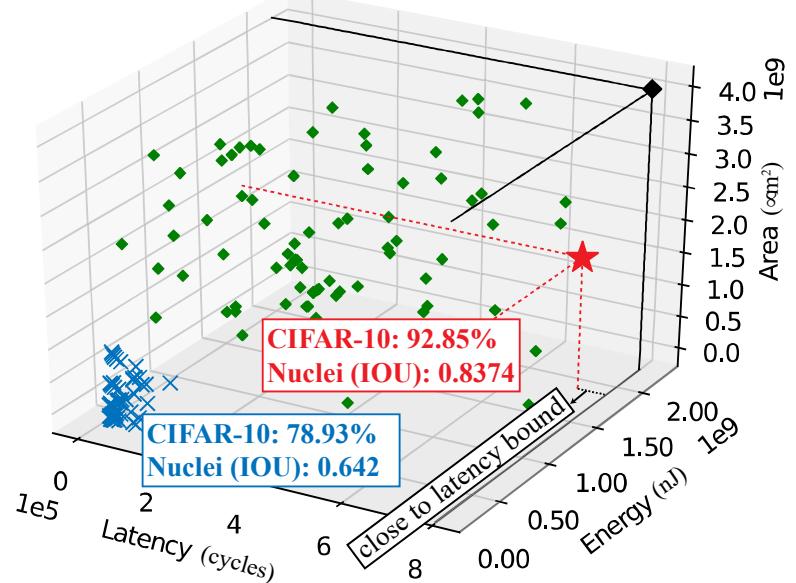
Result 2:

Table I: Comparison between successive NAS and ASIC design (NAS→ASIC), ASIC design followed by hardware-aware NAS (ASIC→HW-NAS), and NASAIC.

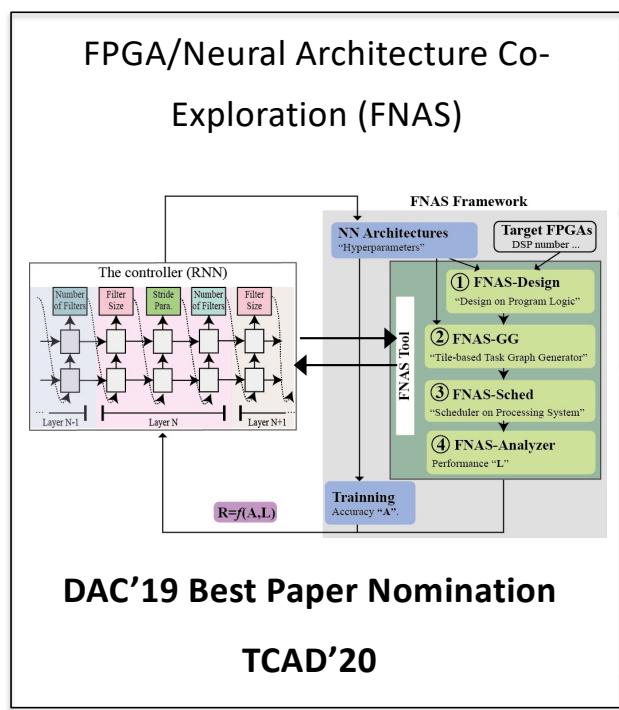
Work.	Approach	Hardware	Dataset	Accuracy	L /cycles	E /nJ	A / μm^2
W1	NAS→ASIC	$\langle dla, 2112, 48 \rangle$ $\langle shi, 1984, 16 \rangle$	CIFAR-10 Nuclei	94.17% 83.94%	9.45e5 ×	3.56e9 ×	4.71e9 ×
	ASIC→HW-NAS	$\langle dla, 1088, 24 \rangle$ $\langle shi, 2368, 40 \rangle$	CIFAR-10 Nuclei	91.98% 83.72%	5.8e5 ✓	1.94e9 ✓	3.82e9 ✓
	NASAIC	$\langle dla, 576, 56 \rangle$ $\langle shi, 1792, 8 \rangle$	CIFAR-10 Nuclei	92.85% 83.74%	7.77e5 ✓	1.43e9 ✓	2.03e9 ✓

Result 1:

◆ Design Specifications ♦ Explored Solutions by NASAIC
 ✕ Lower bounds by the smallest architectures ★ Best Solutions



Co-Exploration of Neural Architectures



Quantization Co-Exploration (ICCAD'19)

Heterogeneous ASIC Co-Exploration (DAC'20)

Network-on-Chip Co-Exploration (ASP-DAC'20 Best Paper Nomination)

Computing-in-Memory Co-Exploration (TC'20)

Secure Inference (ECAI'20)

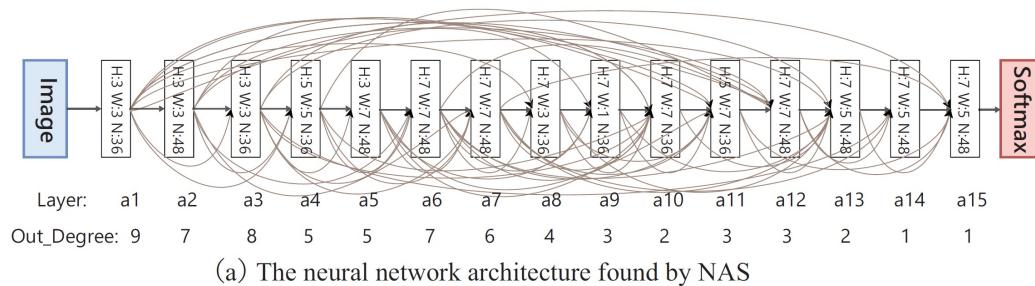
27

The College of Engineering
at the University of Notre Dame

27

NANDS: Co-Explore NoC Design and Neural Architectures

Motivational Example

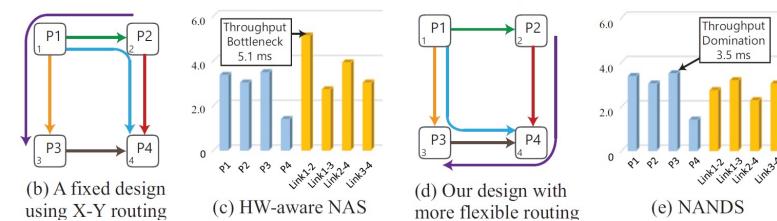


Operation Time	Platforms	Single Processing Element	4 Processing Elements	
			Bus Interconnection	2-D Mesh NoC
Computation (ms)		12.4	3.4	3.4
Data transmission (ms)	—		14.7	6.2

(b) The timing performance of network implementations on different platforms

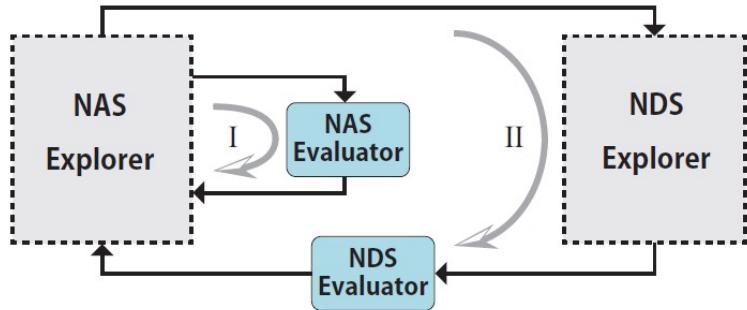
Observations:

- Timing Performance can be improved on platforms with more processing elements
- Communication becomes the performance bottleneck
- Fixed design leads lower performance



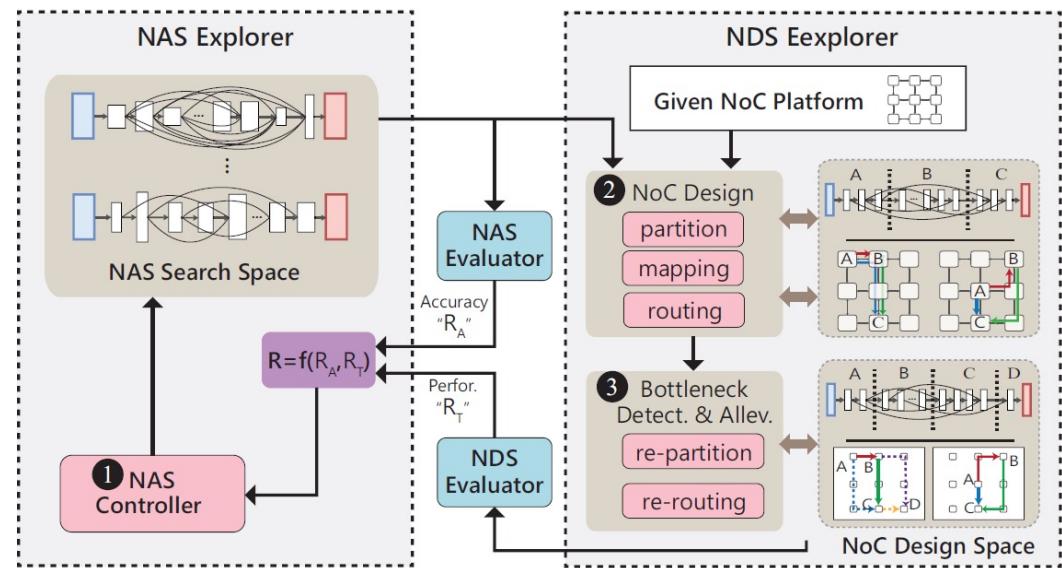
31.37% Improvement

NANDS: Framework



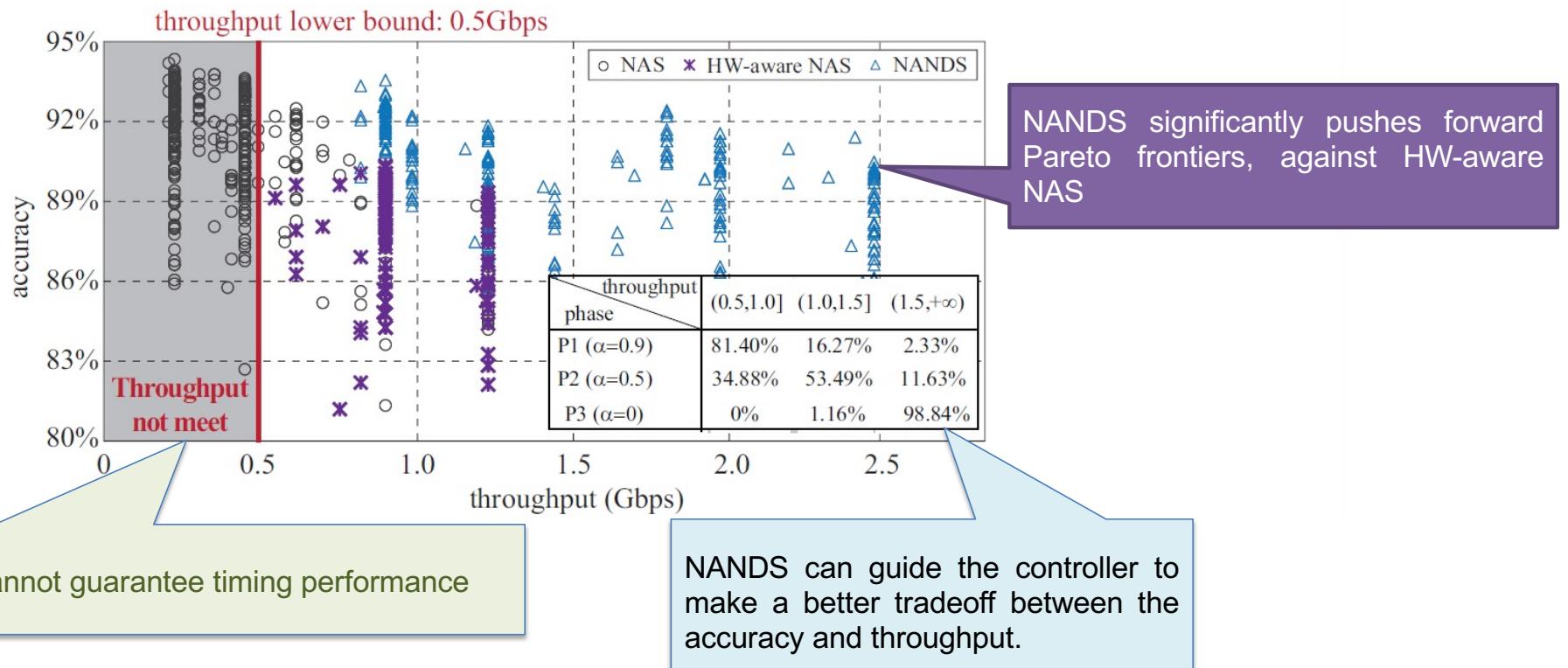
Two exploration loops in NANDS:

- **Loop I: Neural Architecture Search.**
- **Loop II: Automatic Hardware Design**

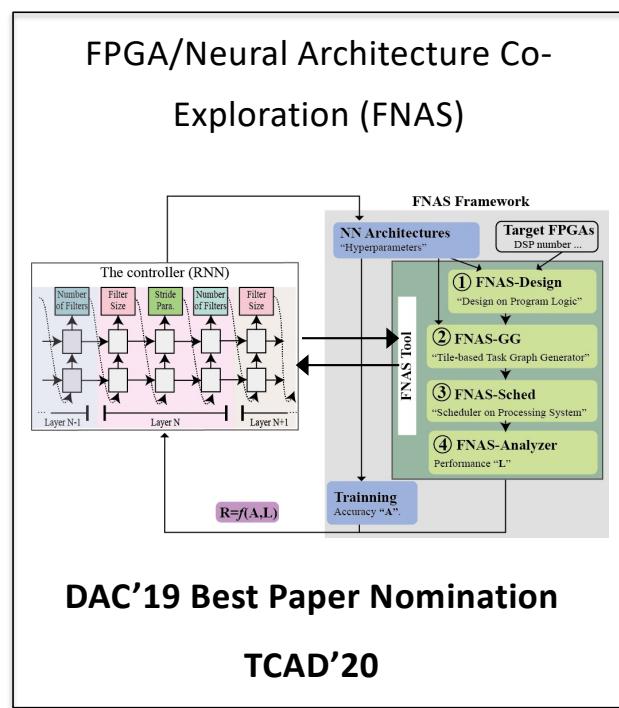


- ① NAS Controller: predict hyperparameters
- ② NoC Design: generate hardware design (e.g., partition, mapping and routing)
- ③ Bottleneck Detection and Alleviation: maximize throughput of NoC.

NANDS: Results



Co-Exploration of Neural Architectures



Quantization Co-Exploration (ICCAD'19)

Heterogeneous ASIC Co-Exploration (DAC'20)

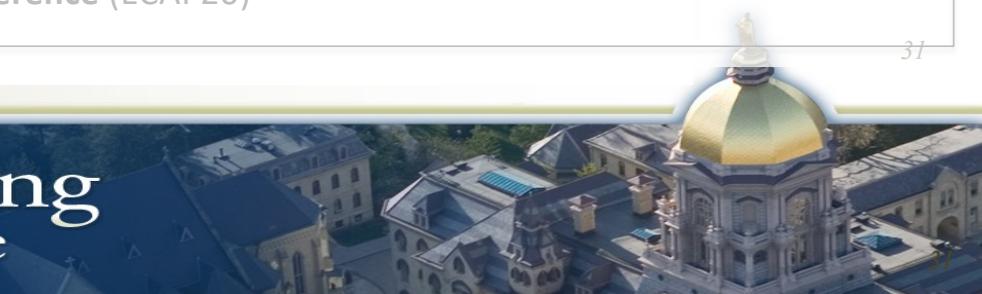
Network-on-Chip Co-Exploration (ASP-DAC'20 Best Paper Nomination)

Computing-in-Memory Co-Exploration (TC'20.)

Secure Inference (ECAI'20)

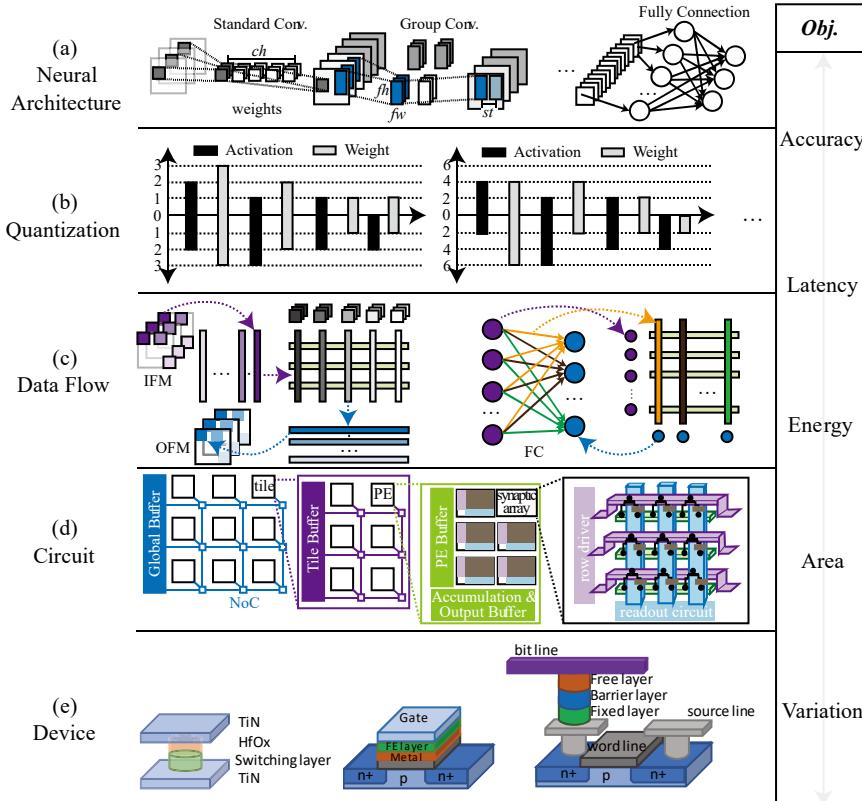
31

*The College of Engineering
at the University of Notre Dame*



NACIM: Device-Circuit-Architecture Co-Exploration

Cross-Layer Optimization

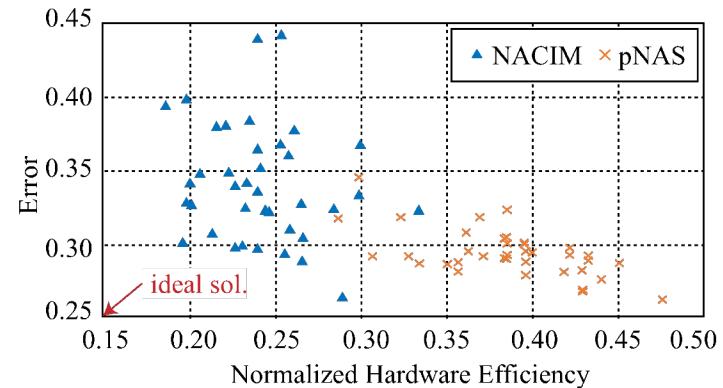


Results

Noise and Hardware Aware

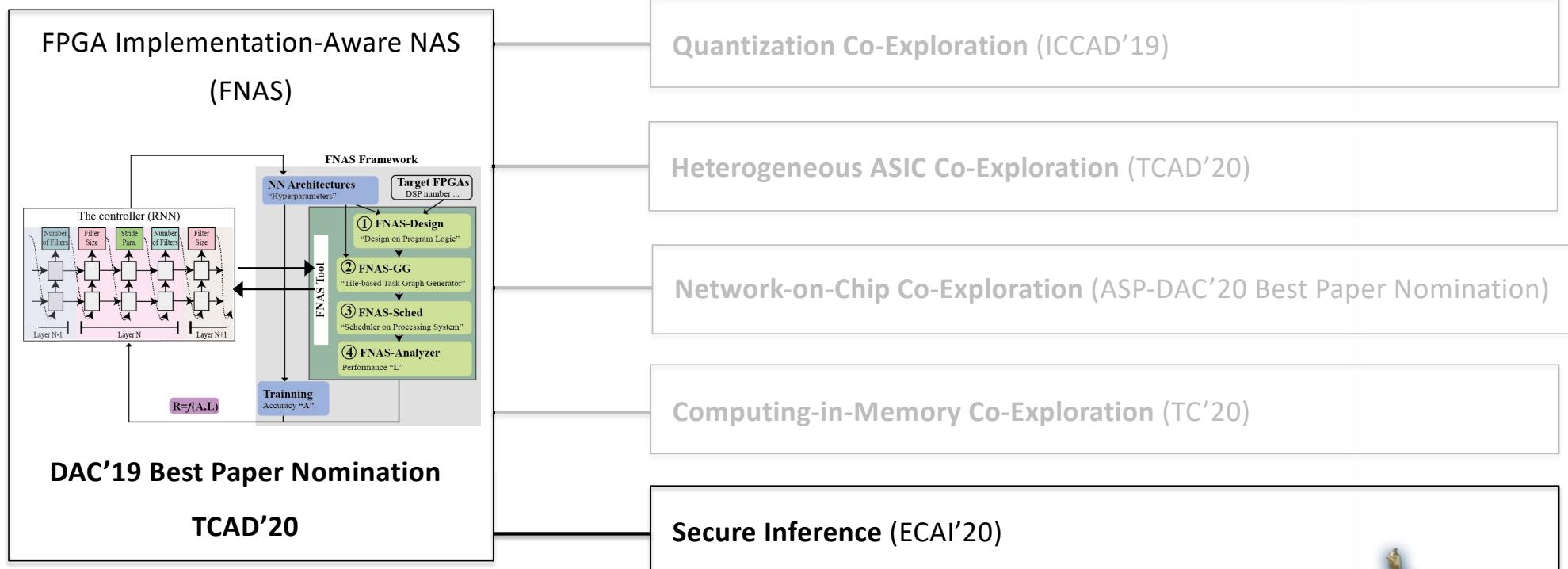
Approach	Accuracy	Acc w/ variation	Area (μm^2)	EDP ($pJ * ns$)	Speed (TOPs)	E.-E. (TOPs/W)
QuantNAS	84.92%	8.48%	$3.24 * 10^6$	$8.08 * 10^{12}$	0.285	5.14
pNAS	73.88%	70.76%	$2.07 * 10^6$	$4.18 * 10^{12}$	0.110	7.14
NACIM _{hw}	73.58%	70.12%	$1.78 * 10^6$	$2.21 * 10^{12}$	0.204	12.3
NACIM _{sw}	73.88%	73.45%	$1.97 * 10^6$	$3.76 * 10^{12}$	0.234	16.3

W/O consideration of device variation leading results useless



Performance Model: Modified NeuroSim

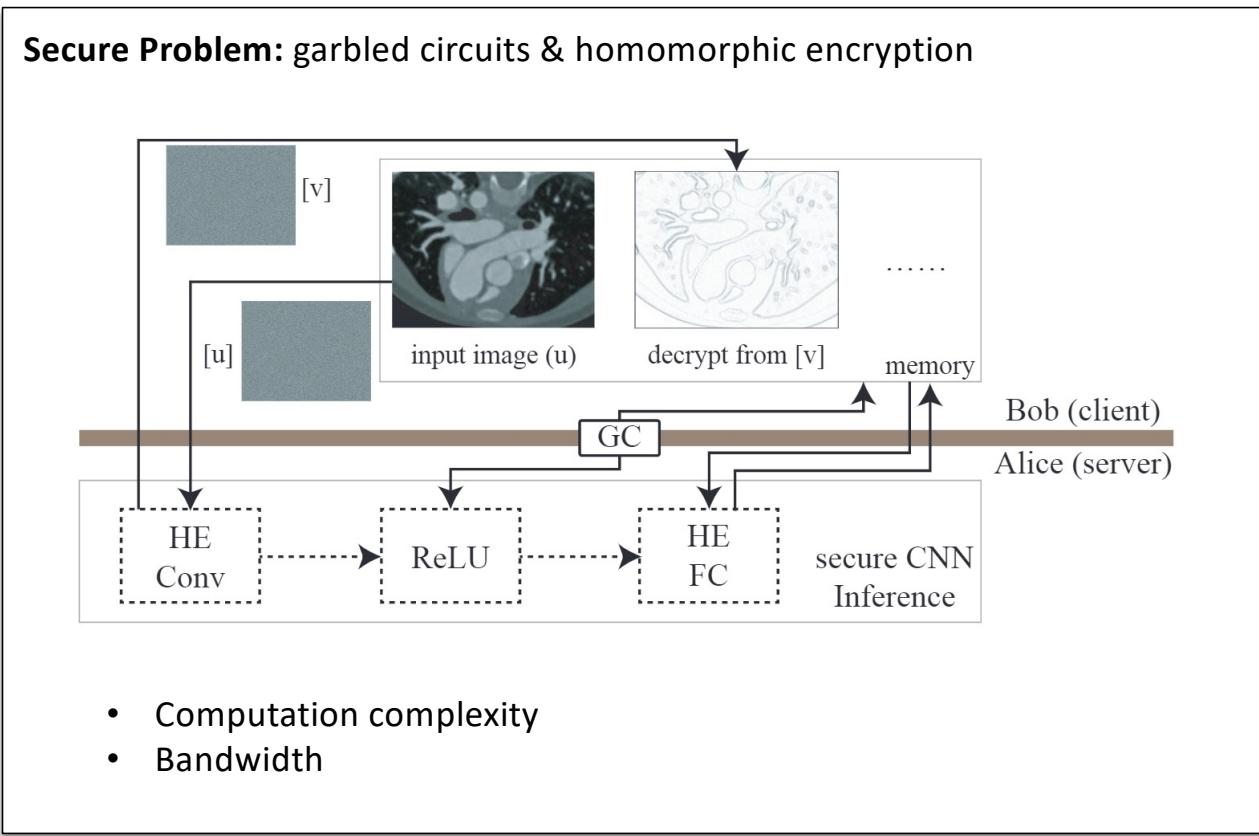
Co-Exploration of Neural Architectures



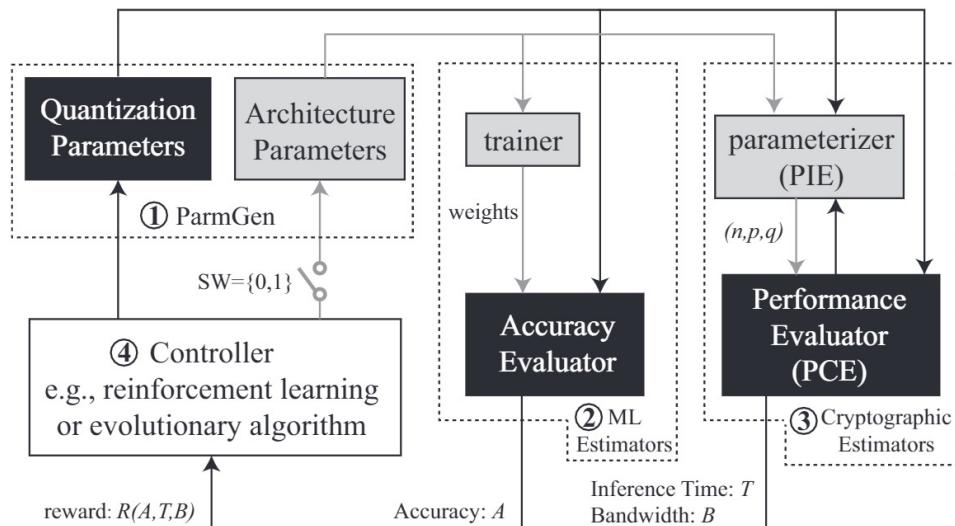
33



NASS: Identifying Secure Inference Architecture via NAS



NASS: Framework and Results

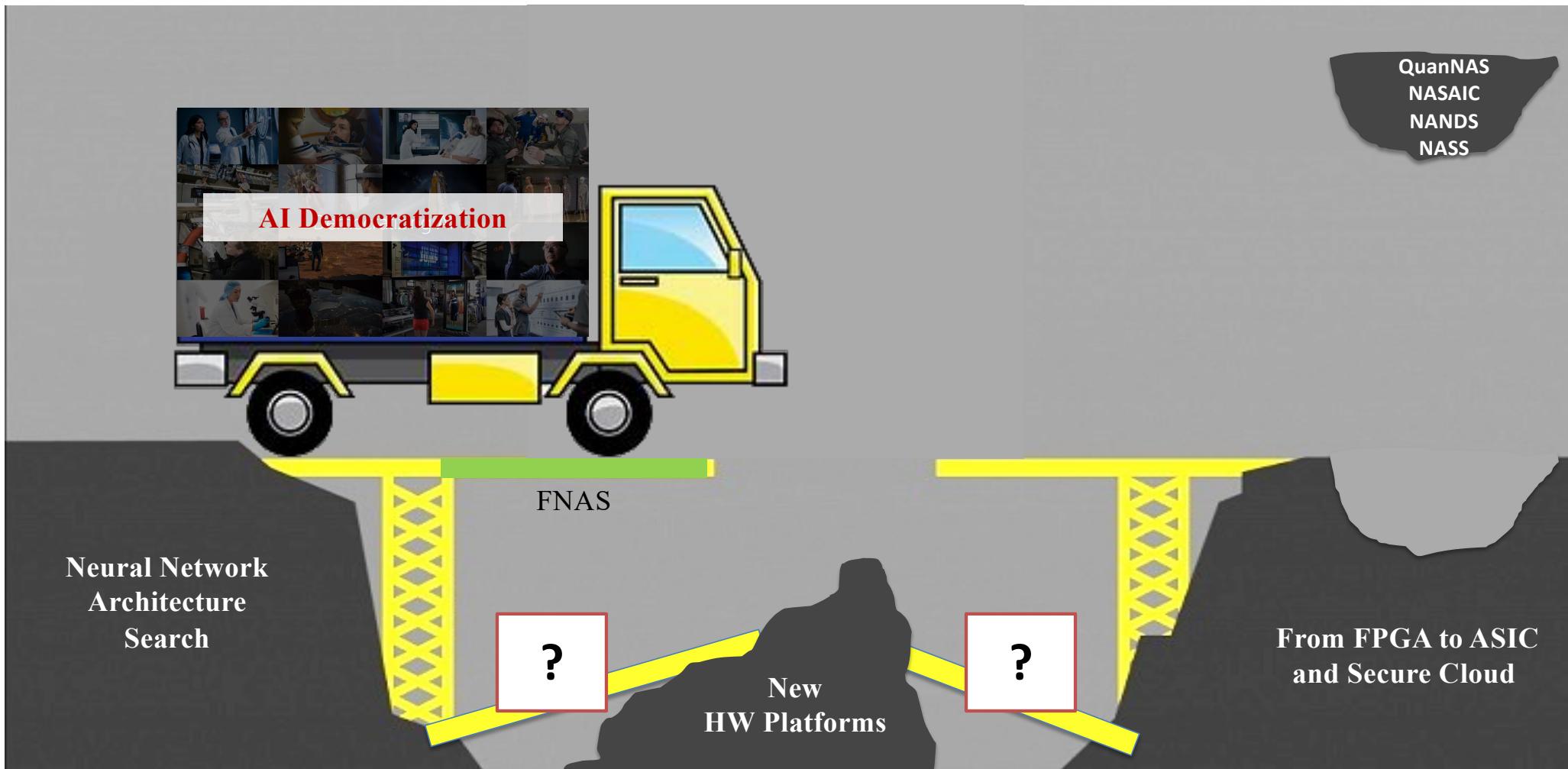


- Determination of hyper-parameters and quantization
- Performance Modeling

Gazelle			Best Searched by NASS		
Layer	Dimension	Quant.	Layer	Dimension	Quant.
CR	(64 × 3 × 3)	23	CR	(24 × 5 × 3)	(8, 8)
CR	(64 × 3 × 3)	23	CR	(48 × 3 × 5)	(6, 7)
PL	(2 × 2)	23	PL	(2 × 2)	(8, 8)
CR	(64 × 3 × 3)	23	CR	(48 × 5 × 7)	(7, 6)
CR	(64 × 3 × 3)	23	CR	(36 × 3 × 3)	(6, 5)
PL	(2 × 2)	23	PL	(2 × 2)	(8, 8)
CR	(64 × 3 × 3)	23	CR	(24 × 7 × 1)	(4, 6)
CR	(64 × 3 × 3)	23	FC	(1024 × 10)	(16, 16)
Accuracy: 81.6%			Accuracy: 84.6%		
Bandwidth: 1.815 GBytes			Bandwidth: 977 MB		
PAHE Time: 3.22 s GC Time: 13.2 s Total Time: 16.4 s			PAHE Time: 1.62 s GC Time: 6.38 s Total Time: 8.0 s		

- Improve accuracy by 3%
- Decrease 2X bandwidth requirement
- Decrease 2X computation time in server side

Conclusion and Future Work



Selected works from our group on this topic

- [1] Weiwen Jiang, Xinyi Zhang, Edwin H.-M. Sha, Qingfeng Zhuge, Lei Yang, Yiyu Shi and Jingtong Hu, "Accuracy vs. Efficiency: Achieving Both through FPGA-Implementation Aware Neural Architecture Search," in Proc. of **DAC 2019**. (**Best Paper Nomination**)
- [2] Weiwen Jiang, Edwin Sha, Xinyi Zhang, Lei Yang, Qingfeng Zhuge, Yiyu Shi and Jingtong Hu, "Achieving Super-Linear Speedup across Multi-FPGA for Real-Time DNN Inference," **CODES+ISSS 2019** and **ACM TECS** (**Best Paper Nomination**)
- [3] Lei Yang, Weiwen Jiang, Weichen Liu, Edwin Sha, Yiyu Shi and Jingtong Hu, "Co-Exploring Neural Architecture and Network-on-Chip Design for Real-Time Artificial Intelligence," **ASP-DAC 2020** (**Best Paper Nomination**)
- [5] Qing Lu, Weiwen Jiang, Xiaowei Xu, Yiyu Shi and Jingtong Hu, "On Neural Architecture Search for Resource-Constrained Hardware Platforms," in Proc. of **ICCAD 2019** (Invited Paper)
- [6] Weiwen Jiang, Lei Yang, Edwin Hsing-Mean Sha, Qingfeng Zhuge, Shouzhen Gu, Sakyasingha Dasgupta, Yiyu Shi, Jingtong Hu, "Hardware/Software Co-Exploration of Neural Architectures, **IEEE Trans. Of Computer Aided Design of Integrated Circuits and Systems**, 2020
- [7] Weiwen Jiang, Qiuwen Lou, Zheyu Yan, Lei Yang, Jingtong Hu, Xiaobo Sharon Hu, Yiyu Shi, "Device-Circuit-Architecture Co-Exploration for Computing-in-Memory Neural Accelerators". **IEEE Trans. on Computers**, 2020
- [8] Song Bian, Weiwen Jiang, Qing Lu, Yiyu Shi, and Takashi Sato, "NASS: Optimizing Secure Inference via Neural Architecture Search", in Proc. of **ECAI, 2020**.
- [9] , "Co-Exploration of Neural Architectures and Heterogeneous ASIC Accelerator Designs Targeting Multiple Tasks", in Proc. of **DAC 2020**

Selected works from our group on this topic

- [10] Weiwen Jiang, Lei Yang, Sakyasingha Dasgupta, Jingtong Hu and Yiyu Shi, "Standing on the Shoulders of Giants: Hardware and Neural Architecture Co-Search with Hot Start," in Proc. of International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2020
- [11] Xiaowei Xu, Yukun Ding, Sharon Hu, Michael Niemier, Jason Cong, Yu Hu and Yiyu Shi, "Scaling of Deep Neural Networks for Edge Inference: A Race between Data Scientists and Hardware Architects", Nature Electronics 1, pp. 216-222, 2018.
- [12] Weiwen Jiang, Bike Xie, Chun-Chen Liu and Yiyu Shi, "Integrating Memristors and CMOS for Better AI," Nature Electronics, September 2019
- [13] Yukun Ding, Weiwen Jiang, Qiuwen Lou, Jinglan Liu, Jinjun Xiong, Xiaobo Sharon Hu, Xiaowei Xu, and Yiyu Shi, "Hardware design and the competency awareness of a neural network," Nature Electronics, 3, pp. pages514–523, 2020.
- [14] Weiwen Jiang, Jinjun Xiong and Yiyu Shi, "A Co-Design Framework of Neural Networks and Quantum Circuits Towards Quantum Advantage," Nature Communications, 2021

Thank You!

*The College of Engineering
at the University of Notre Dame*

