Efficient Hardware and Neural Architecture Co-Search with Hot Start

—— A New Road for NN to HW (ROAD4NN2HW)

Weiwen Jiang, Ph.D.

Postdoc Research Associate

Department of Computer Science and Engineering

University of Notre Dame

wjiang2@nd.edu | https://wjiang.nd.edu

A series of ROAD4NN2HW works are conducted at
Univ. of Notre Dame in **Prof. Yiyu Shi's** group and Univ. of Pittsburgh in **Prof. Jingtong Hu's** group

UNIVERSITY OF NOTRE DAME

University of Pittsburgh

Embedded Computing Hardware Has Been in Every Corner

Agriculture

Military

Power System

Manufacture

Education

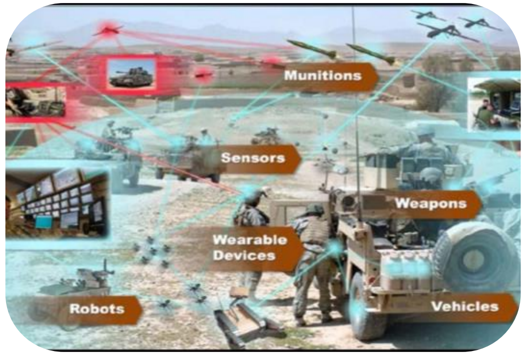
Medical Operation

Finance

……

Today, AI is Going to Every Embedded Computing Hardware

Agriculture

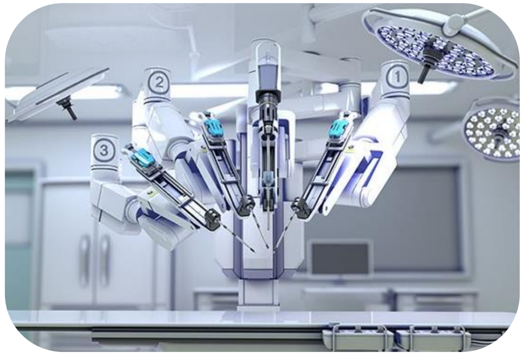


Military

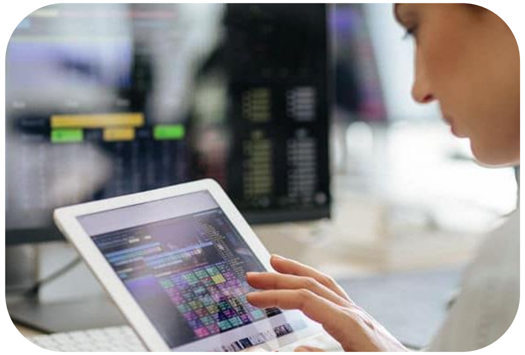
Power System

Manufacture

Education



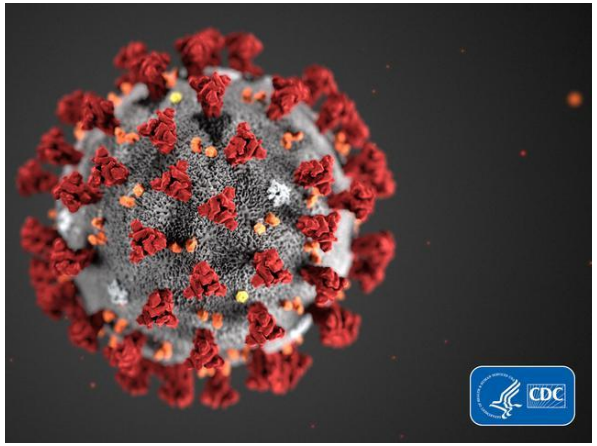
Medical Operation

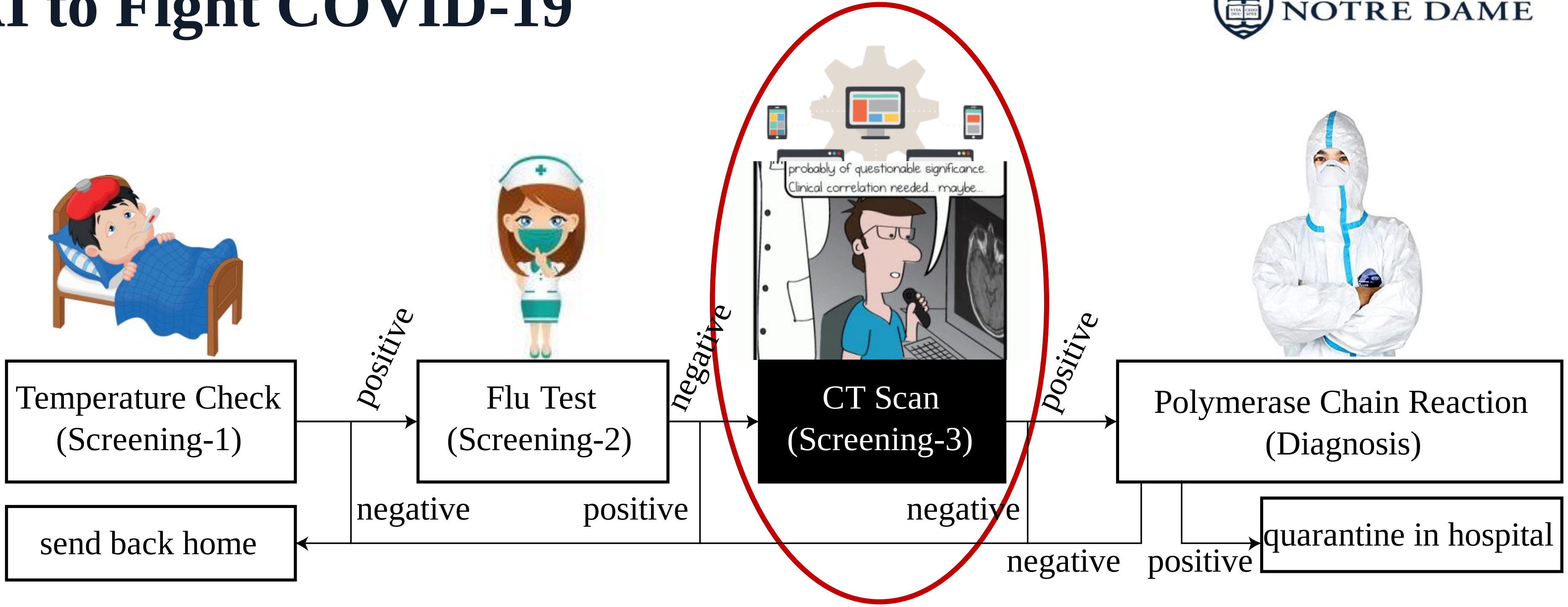


Finance

……

Example: Equip AI to Fight COVID-19





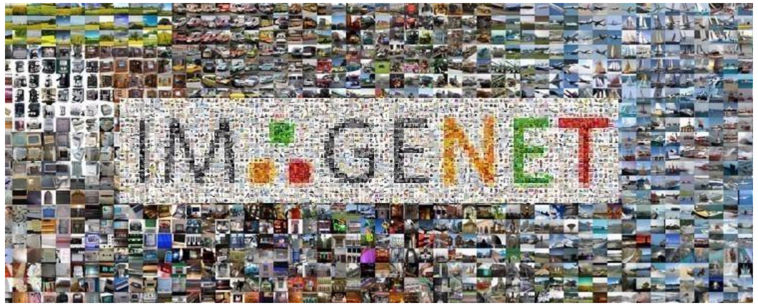
Challenge	Response
Shortage of rRT-PCR test kits	**Accurate screening**
Burden on radiologists in reading CT scan results	**AI judgement to reduce burden**
Days of deployment is intolerant	**Plug-and-play in clinics within Hours**

[ref] How a country serious about coronavirus does testing and quarantine. https://www.youtube.com/watch?v=e3gCbkeARbY. [Online; accessed 03/17/2020]

Today's Solution.

Matching Datasets/Applications and Neural Networks

Datasets / Applications

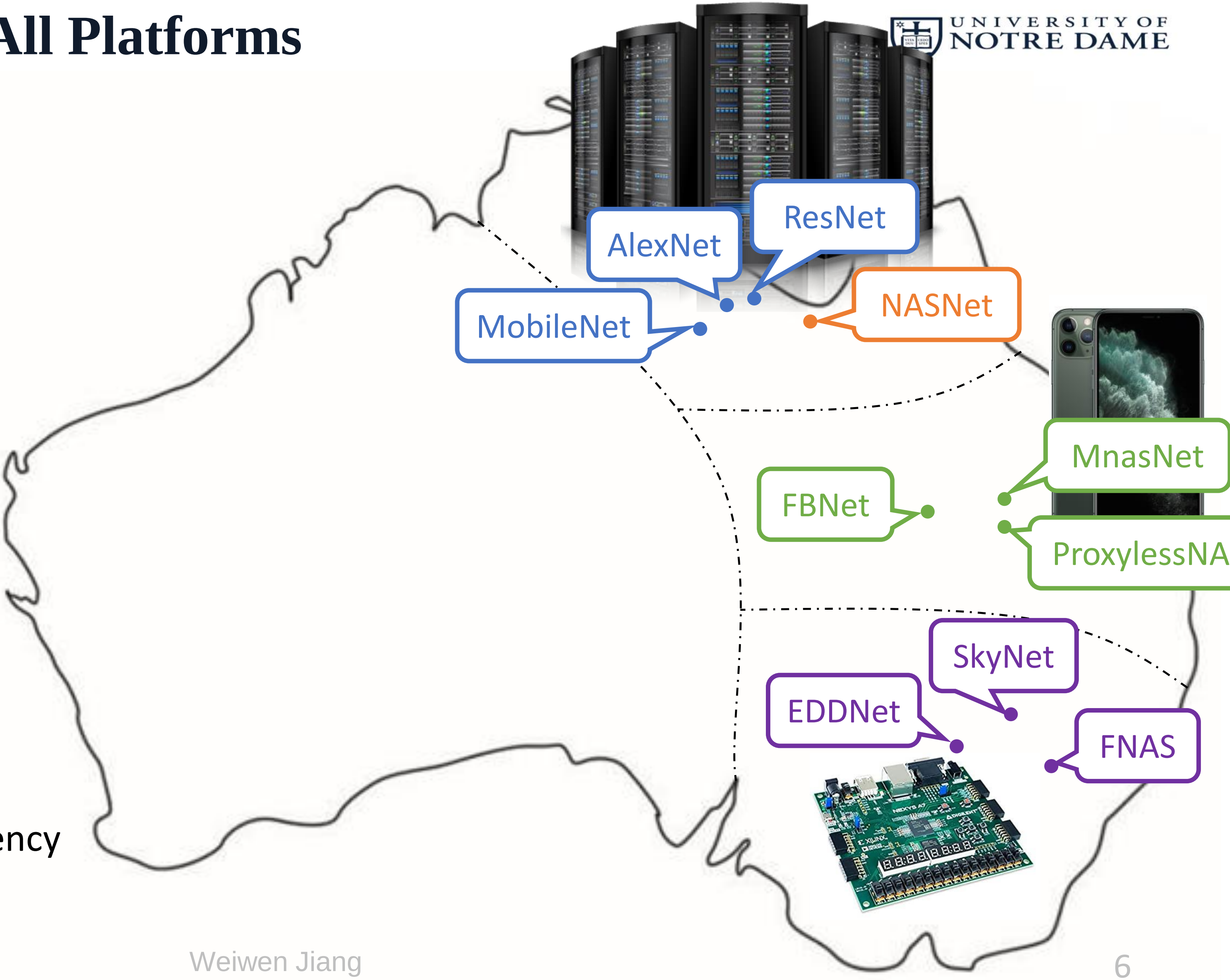


COCO
Common Objects in Context

CITYSCAPES
DATASET

One Network cannot Fit All Platforms

- **Cloud / Server**
  - Resource Unlimited
  - Maximizing Accuracy
  - AlexNet, VGGNet, ResNet, …
- **Mobile Phones**
  - Fixed Hardware
  - Accuracy v.s. Latency
  - MnasNet, ProxylessNAS, …
- **FPGA Accelerators**
  - Hardware Design Flexibility
  - Accuracy, Timing, Energy Efficiency
  - FNAS, SkyNet, EDDNet, …



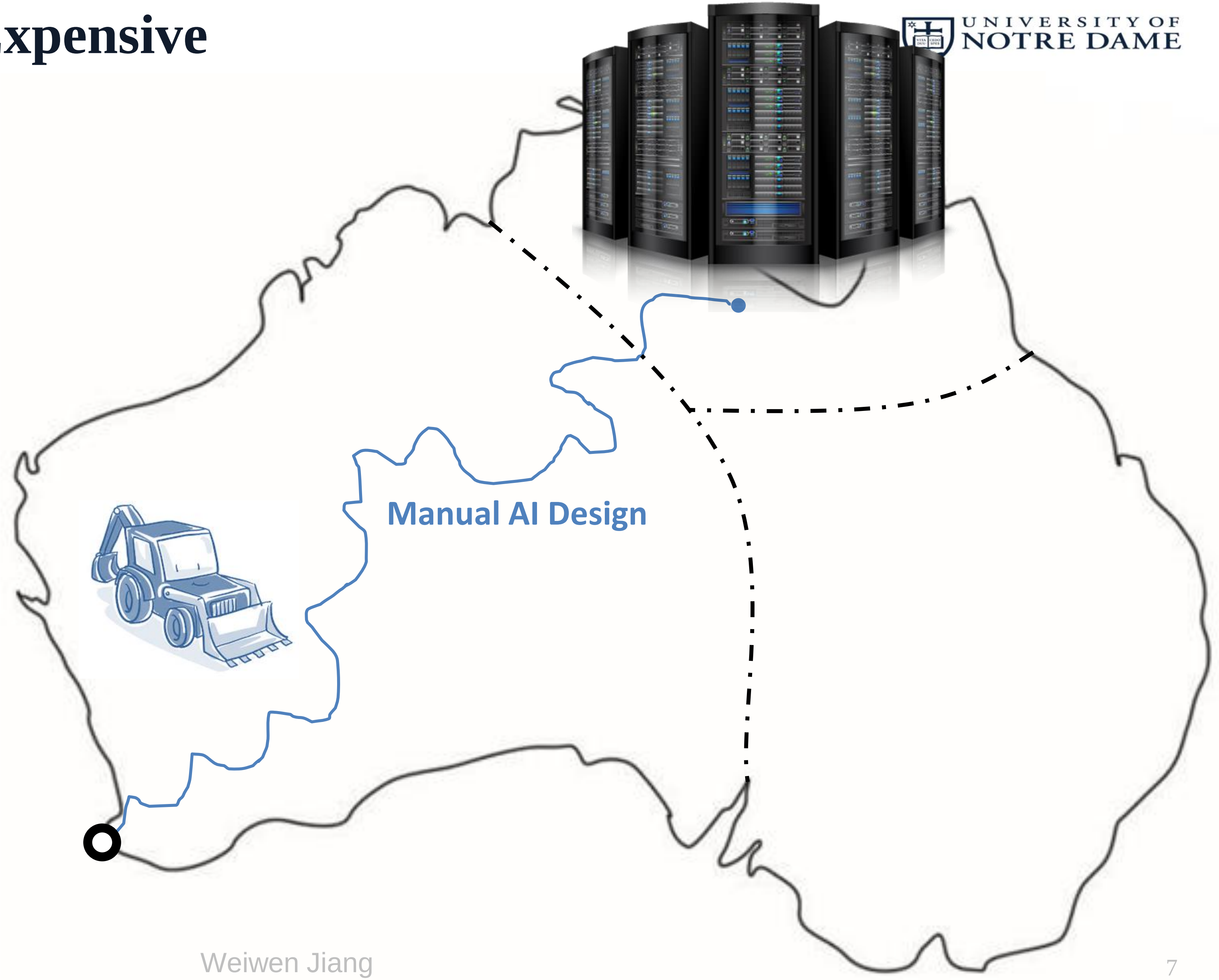
Manual Design is TOO Expensive

1 year for only 1 application

Name	Time
AlexNet	2012
ZFNet	2013
VGGNet	2014
RestNet	2015
GoogleNet	2016

Problem

- **Domain knowledge and excessive labor**
- **It takes too long to devise new architectures**



Automatic Neural Architecture Search (NAS)

Name	Time
NAS	Nov. 2016
NASNet	Jul. 2017

Sample architecture NN
with probability p

Controller (RNN)

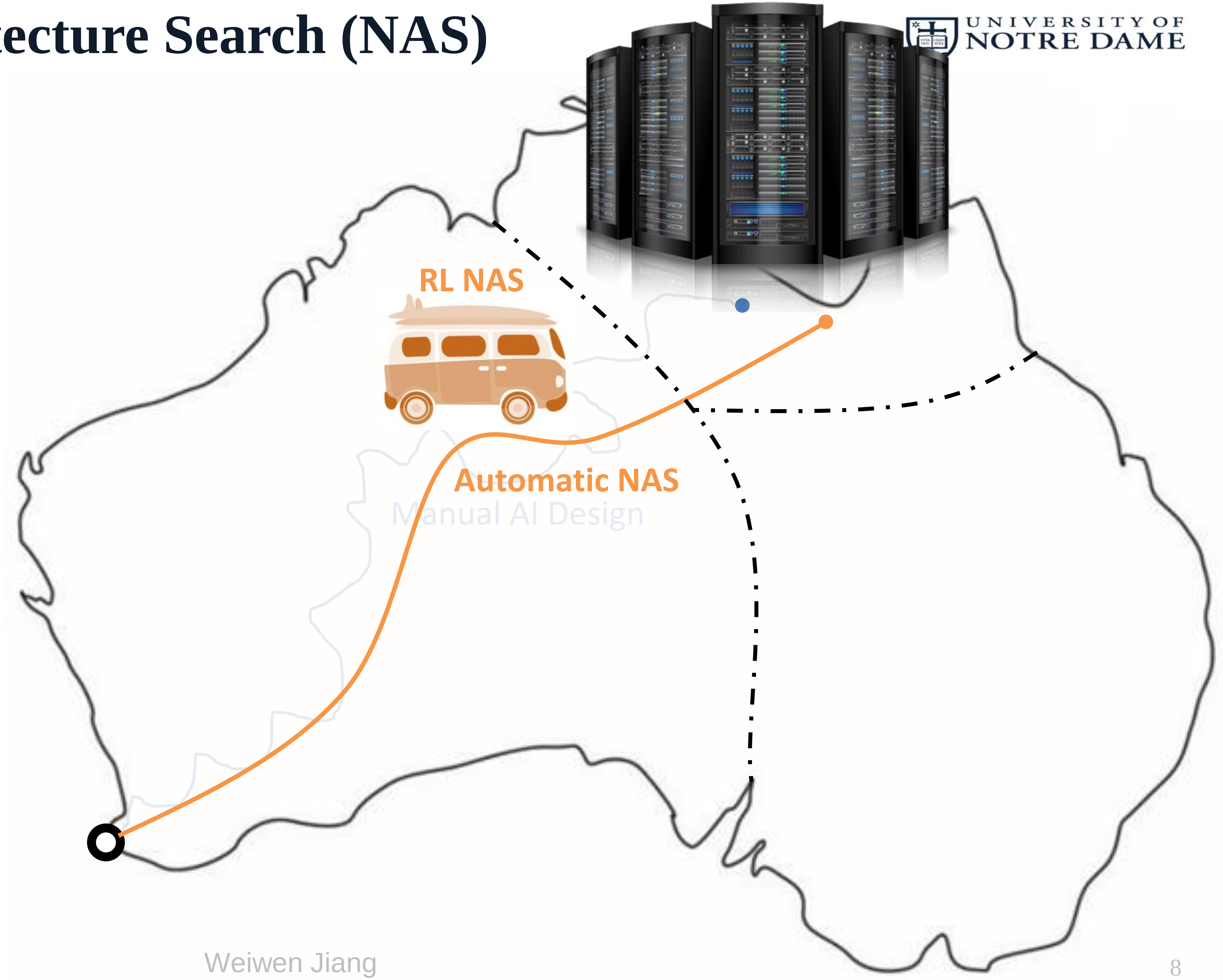
Train from Scratch
To Obtain Accuracy A

Compute gradient of p and
scale it by A to update the
controller

Reinforcement Learning Based NAS

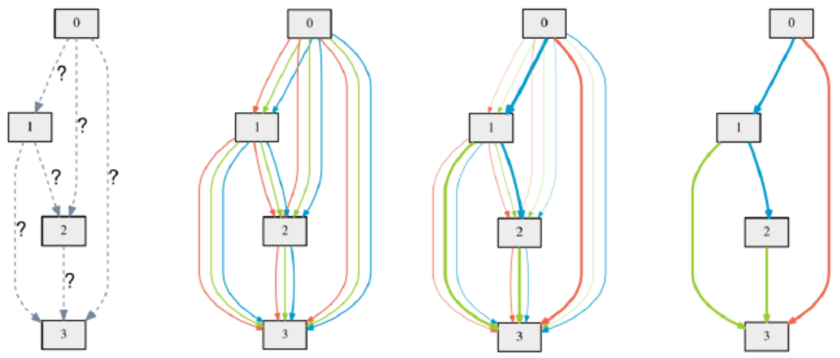
Problem

- **Low Efficiency, hundreds or even thousands of GPU hours**
- **Mono-Objective: Accuracy**

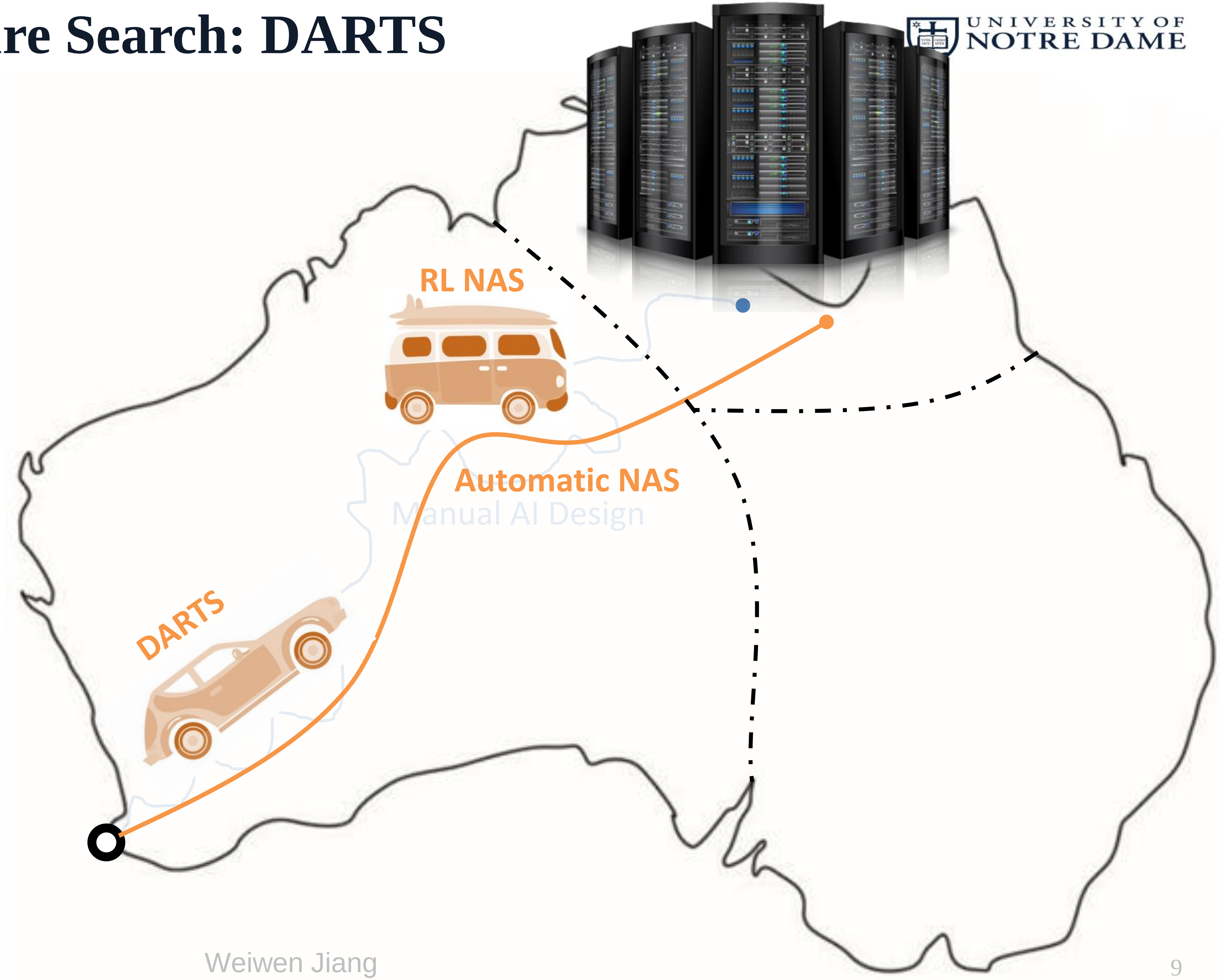


Differentiable Architecture Search: DARTS

Name	Time
DARTS	Jun. 2018



DARTS

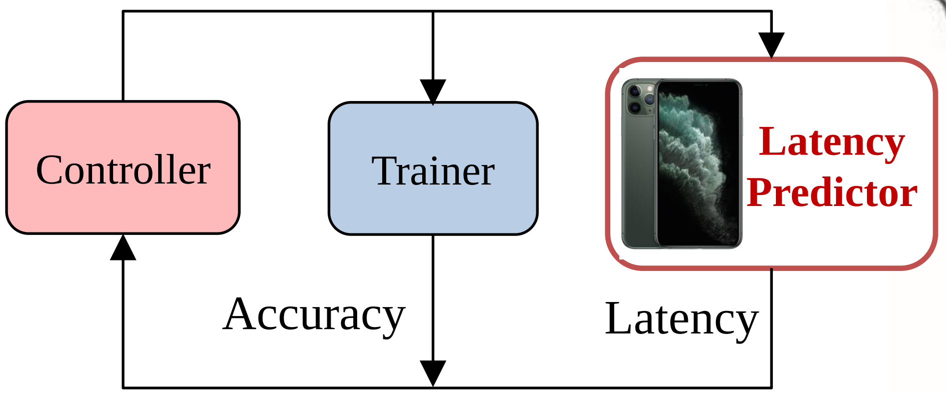


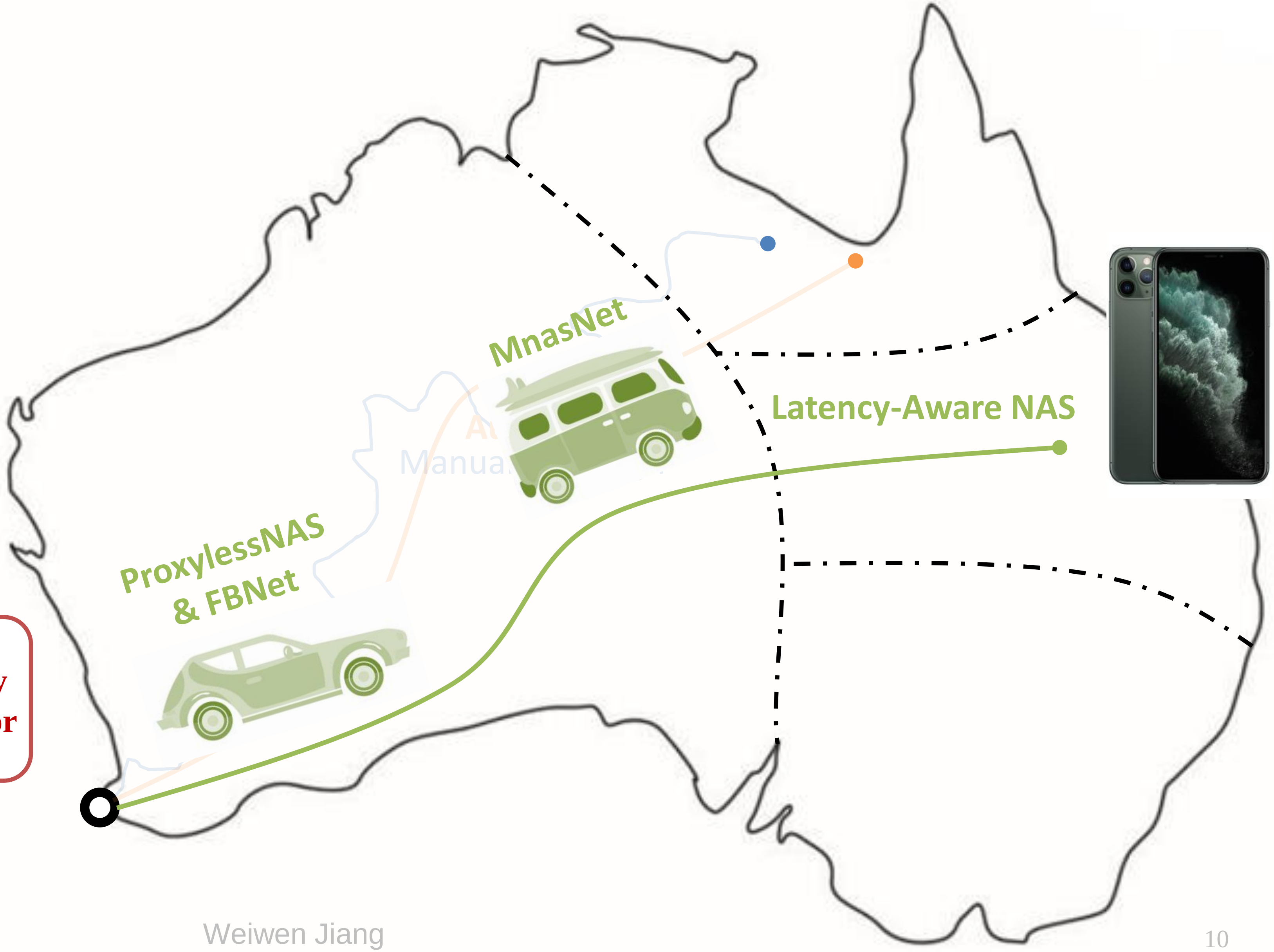
Latency-Aware NAS for Mobile Phones

UNIVERSITY OF NOTRE DAME

Name	Time
MnasNet	Jul. 2018
ProxylessNAS	Dec. 2018
FBNet	Dec. 2018

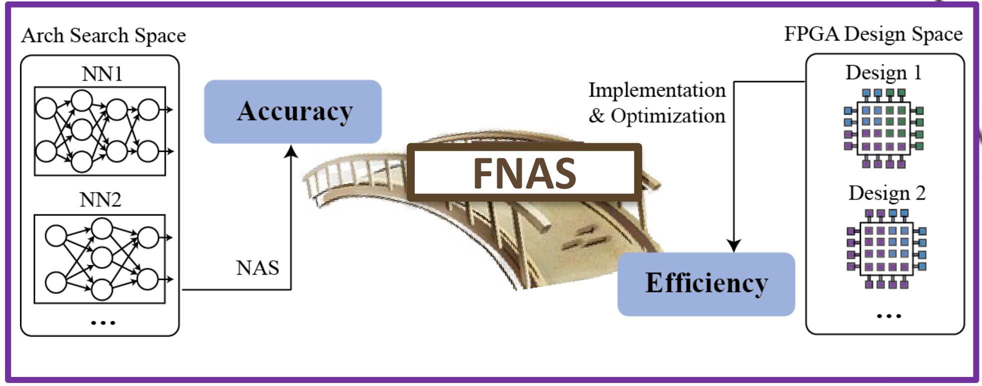
Latency-Aware NAS



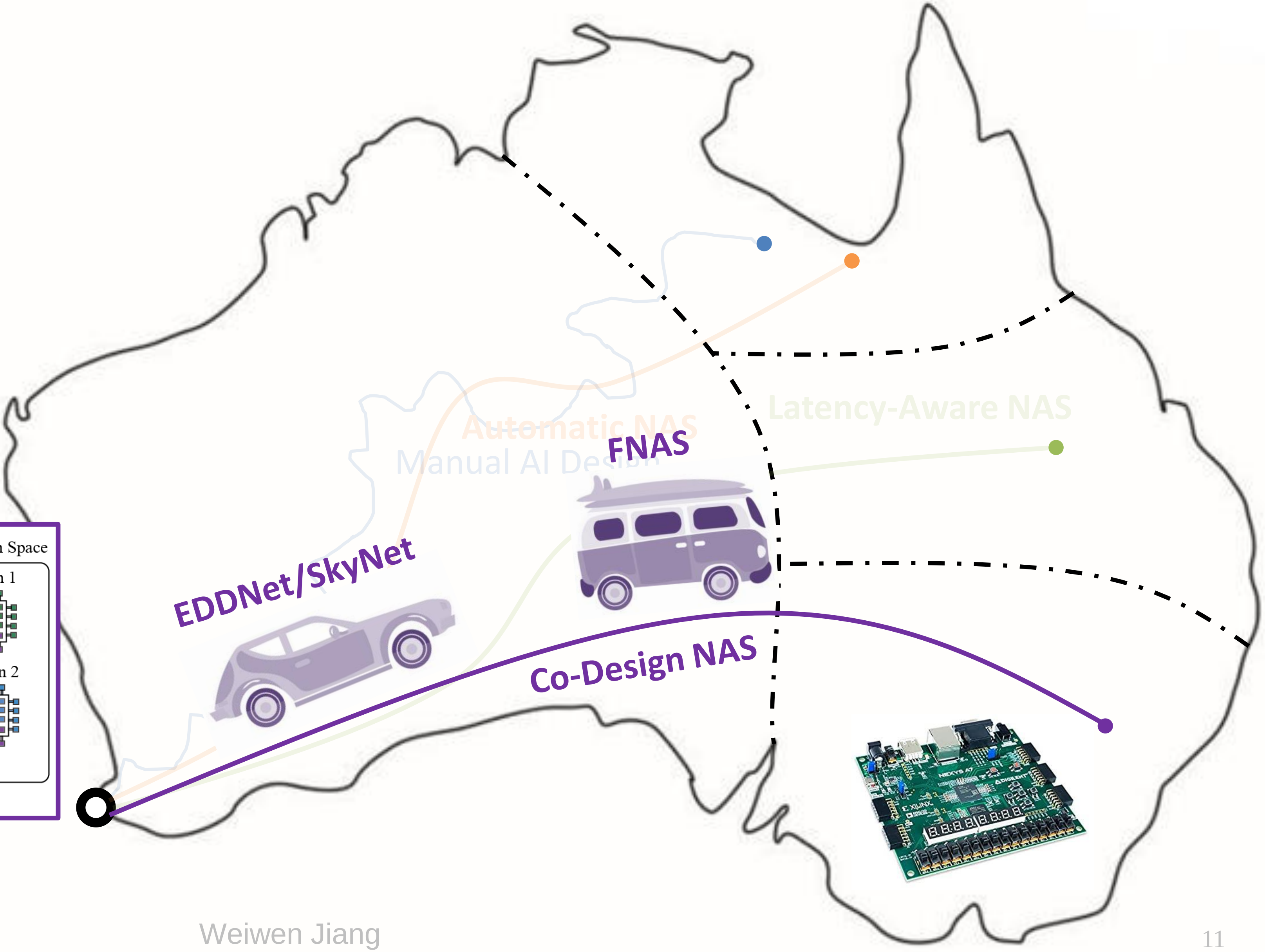


Network-FPGA Co-Design Framework using NAS

Name	Time
FNAS (ours)	Jan. 2019
DNN/FPGA	Apr. 2019
SkyNet	Sep. 2019
EDDNet	May. 2020

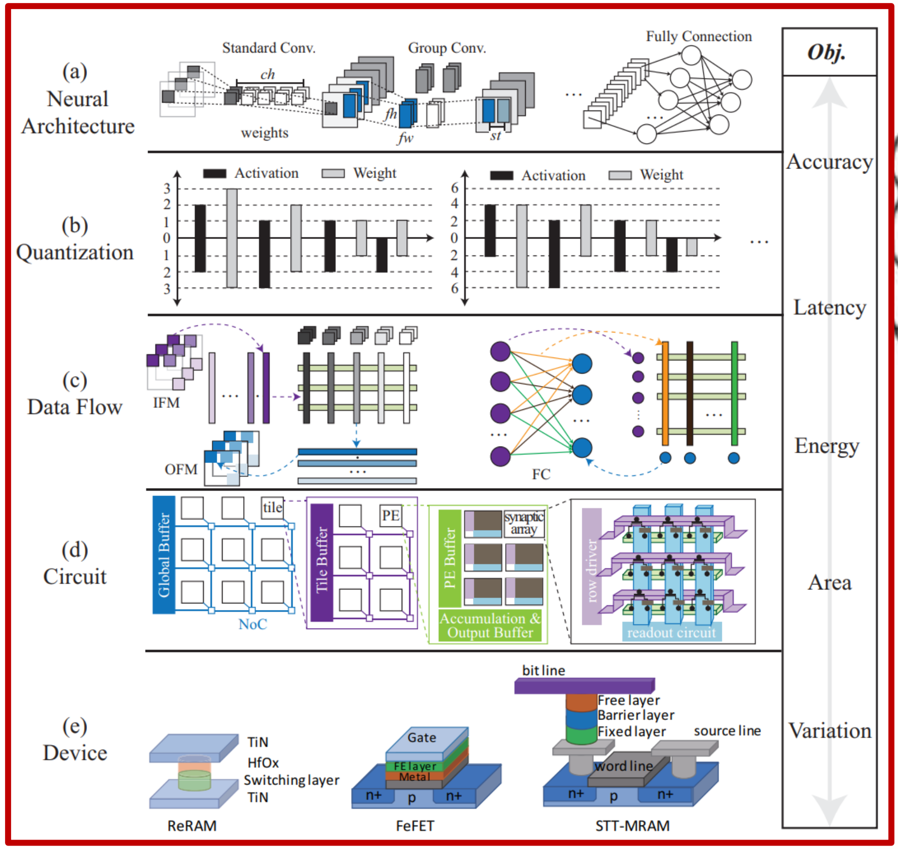


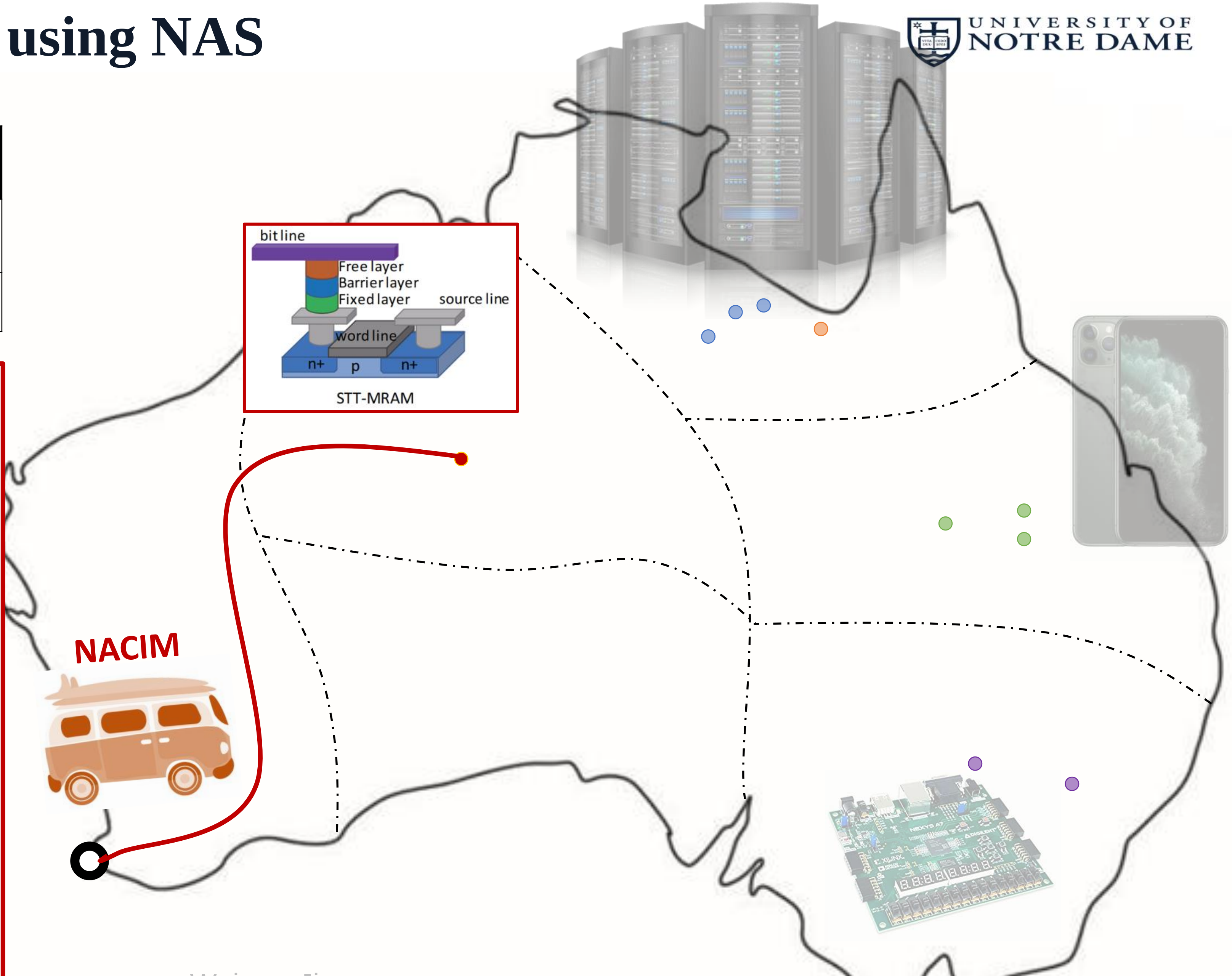
DAC’19 (BEST PAPER NOMINATION)



Network-CIM Co-Design using NAS

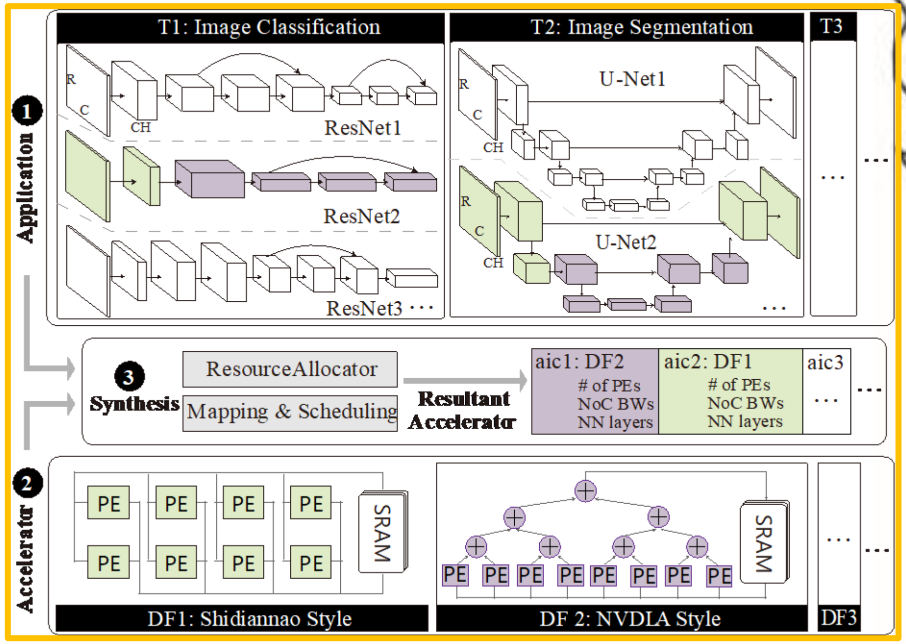
Name	Time
NACIM (ours)	Oct. 2019
Accepted by IEEE Trans. on Computers	

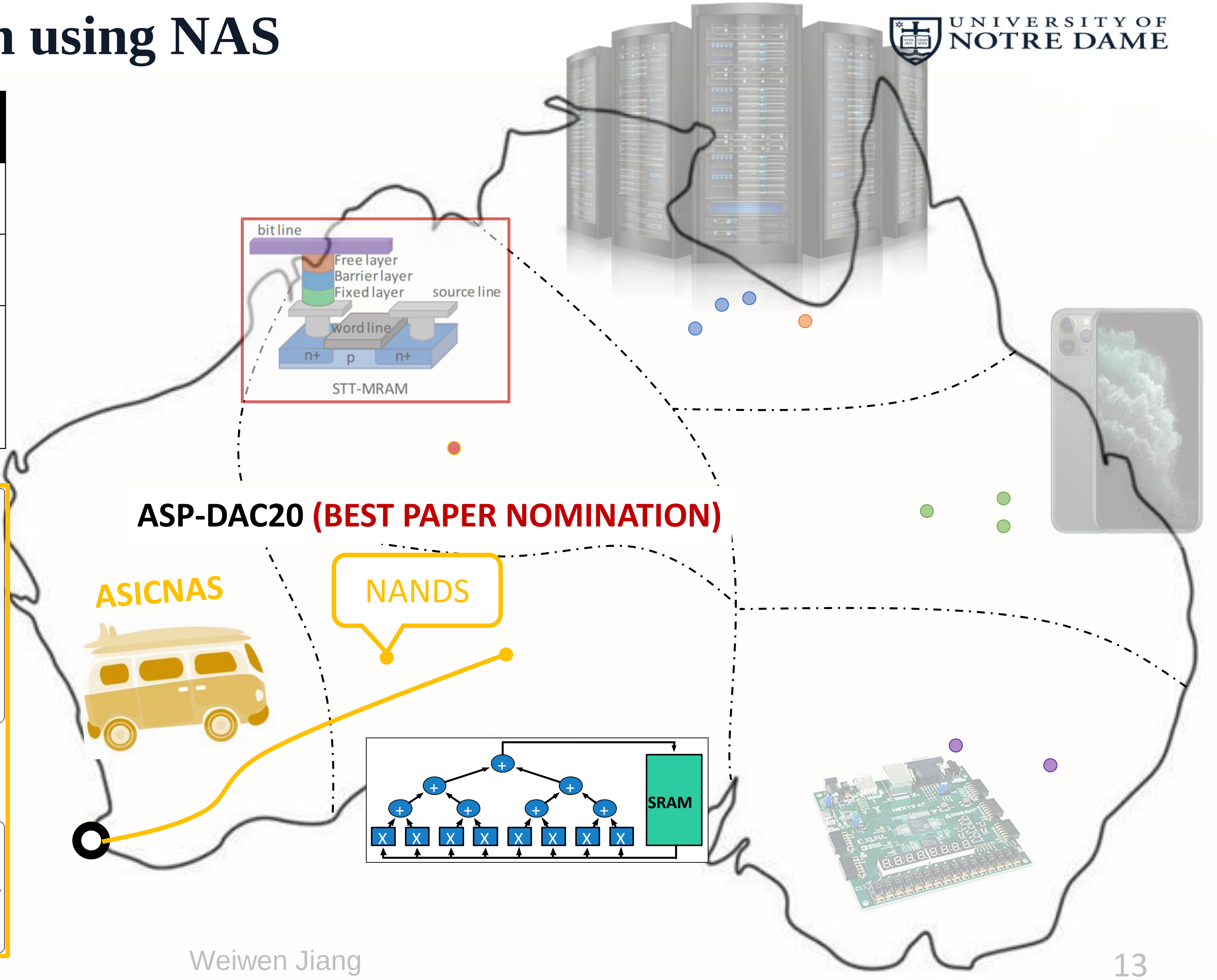




Network-ASIC Co-Design using NAS

Name	Time
NANDS (ours)	Jan. 2020
ASICNAS (ours)	Feb. 2020
DAC'20, 69.3, WEDNESDAY July 22 *Pedal to the M(eta)L: Accelerating Deep Learning to the Next Level*	

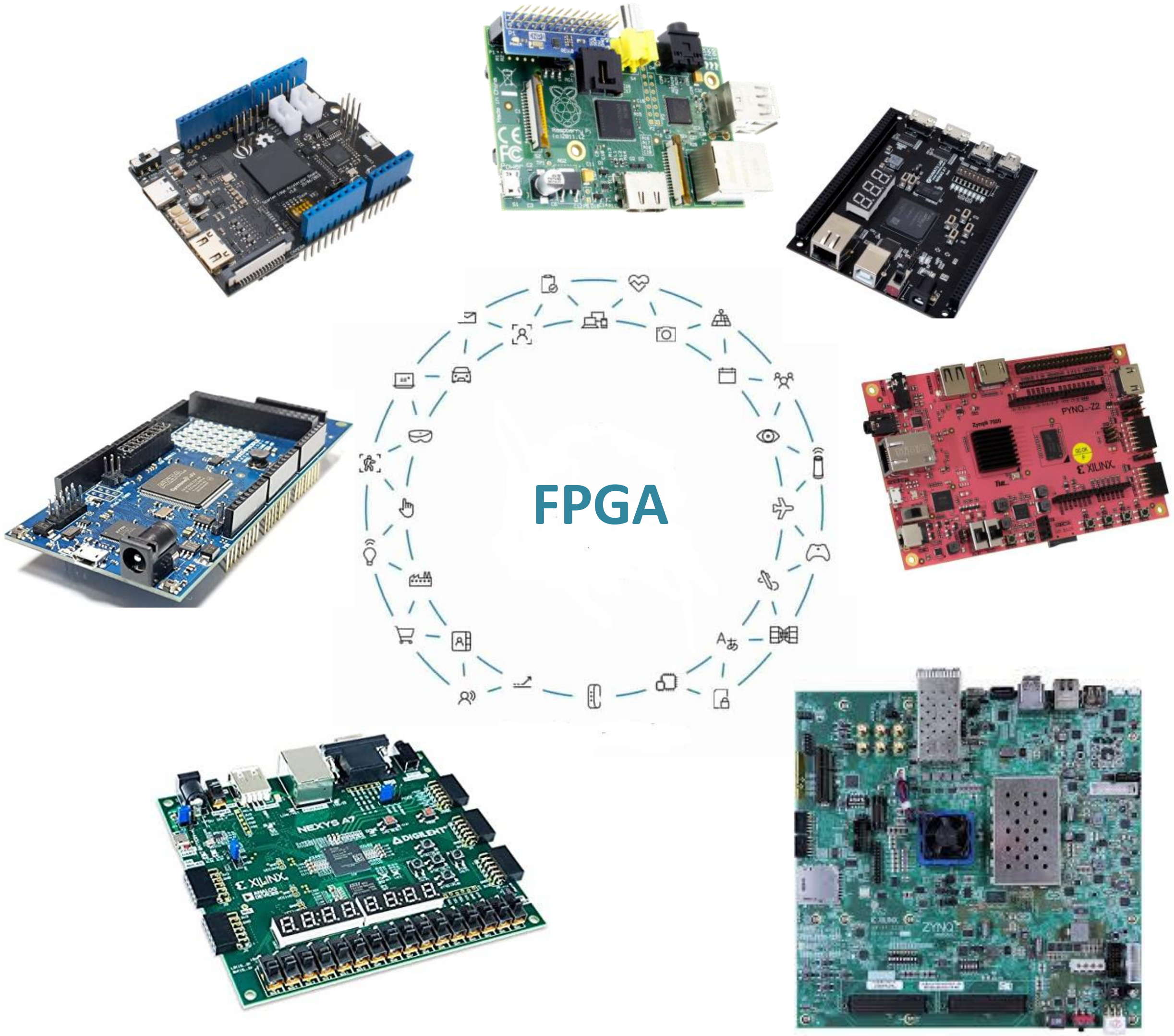




So far, everything looks good.

What’s the problem?

Intelligence is Widely Needed in Hardware Devices <u>NOT</u> Platforms



Needs of NN for Each Device, Not For Each Platform

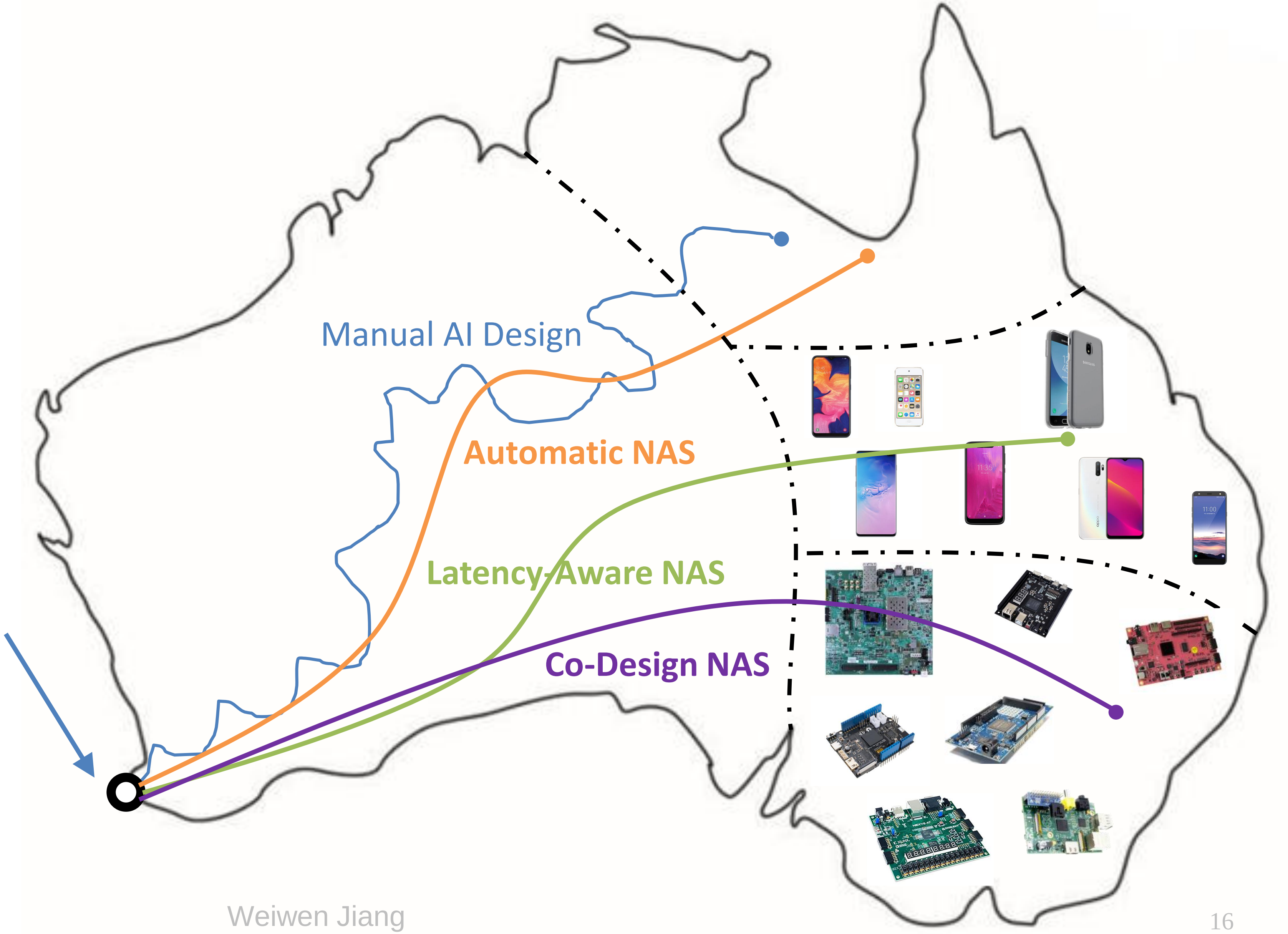
UNIVERSITY OF NOTRE DAME

Tens to Hundreds of GPU Hour for each device is inefficient

WHY INEFFICIENT?

Search from Scratch!

- Cold Start
- Lengthy Training Time



Rethinking: Why Always Conduct NAS from Cold?

HotNAS: < 3 GPU Hours (ImageNet); < 20 GPU Minutes (CIFAR-10)

Accepted by CODES+ISSS'20

UNIVERSITY OF NOTRE DAME

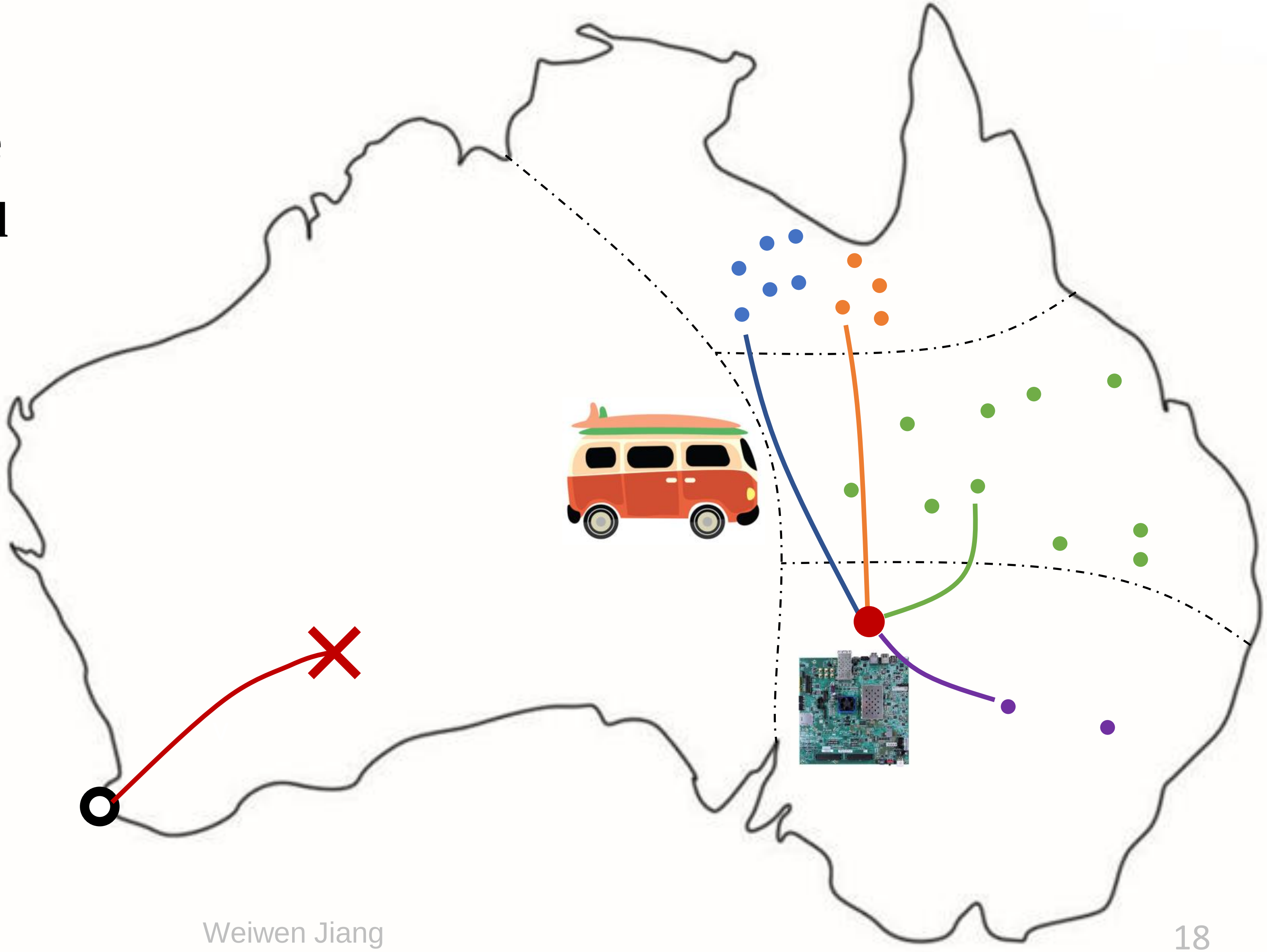
EDGECORTIX

University of Pittsburgh

HotNAS: Search from Hot

◆ **Pave a new ROAD from the existing trained NNs (Model Zoo) to hardware**

+ Significantly reduce **search time**
+ With little or no **accuracy** loss
+ Guarantee to meet the given **hardware constraints**



HotNAS: Problem Definition

Given:

- Pre-trained **model zoo**
- Hardware design templates
- Design specifications

Search:

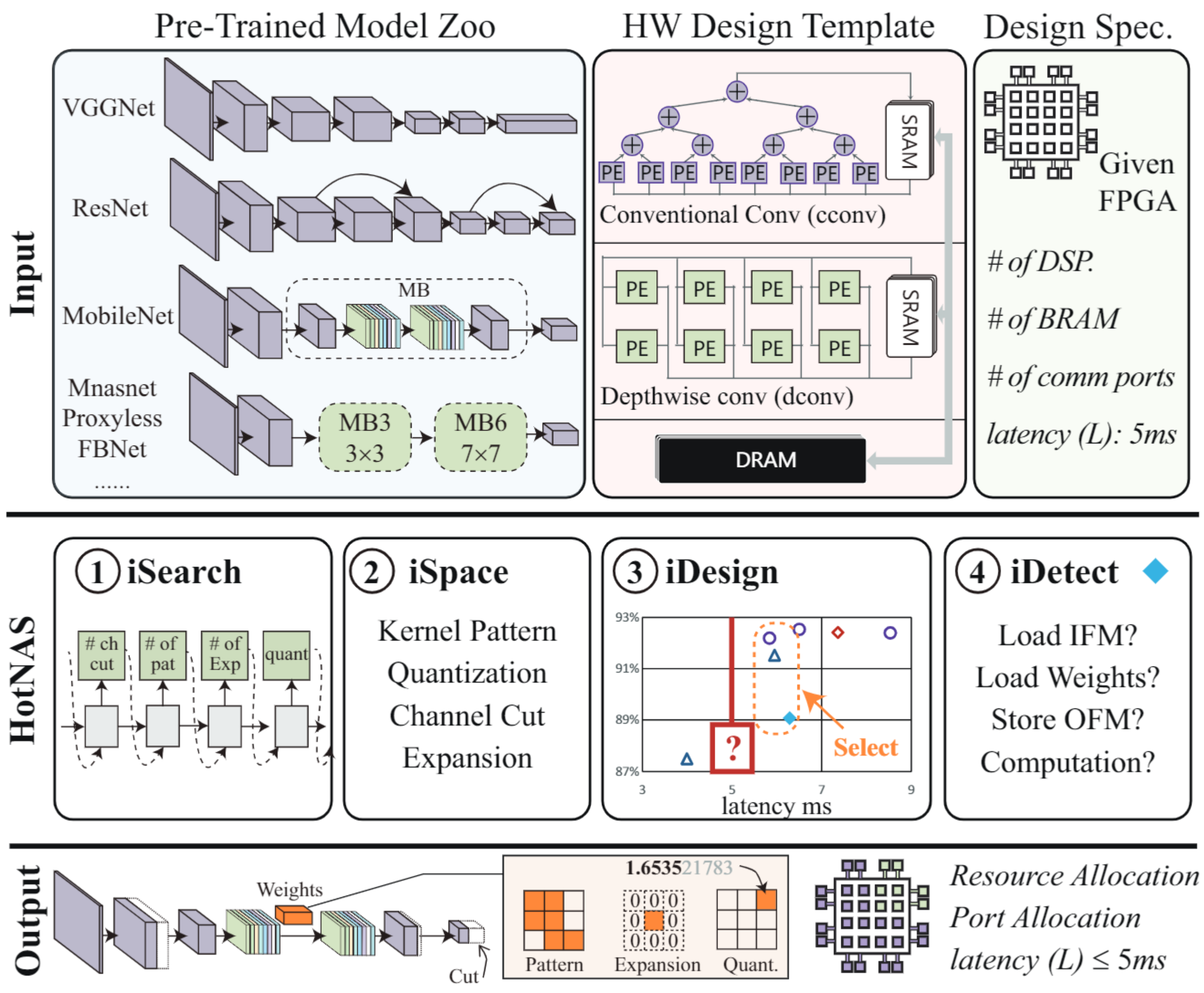
- Network **architecture** hyperparameters

  (i.e., # of channel, kernel size, connections, etc.)
- **Hardware design** hyperparameters

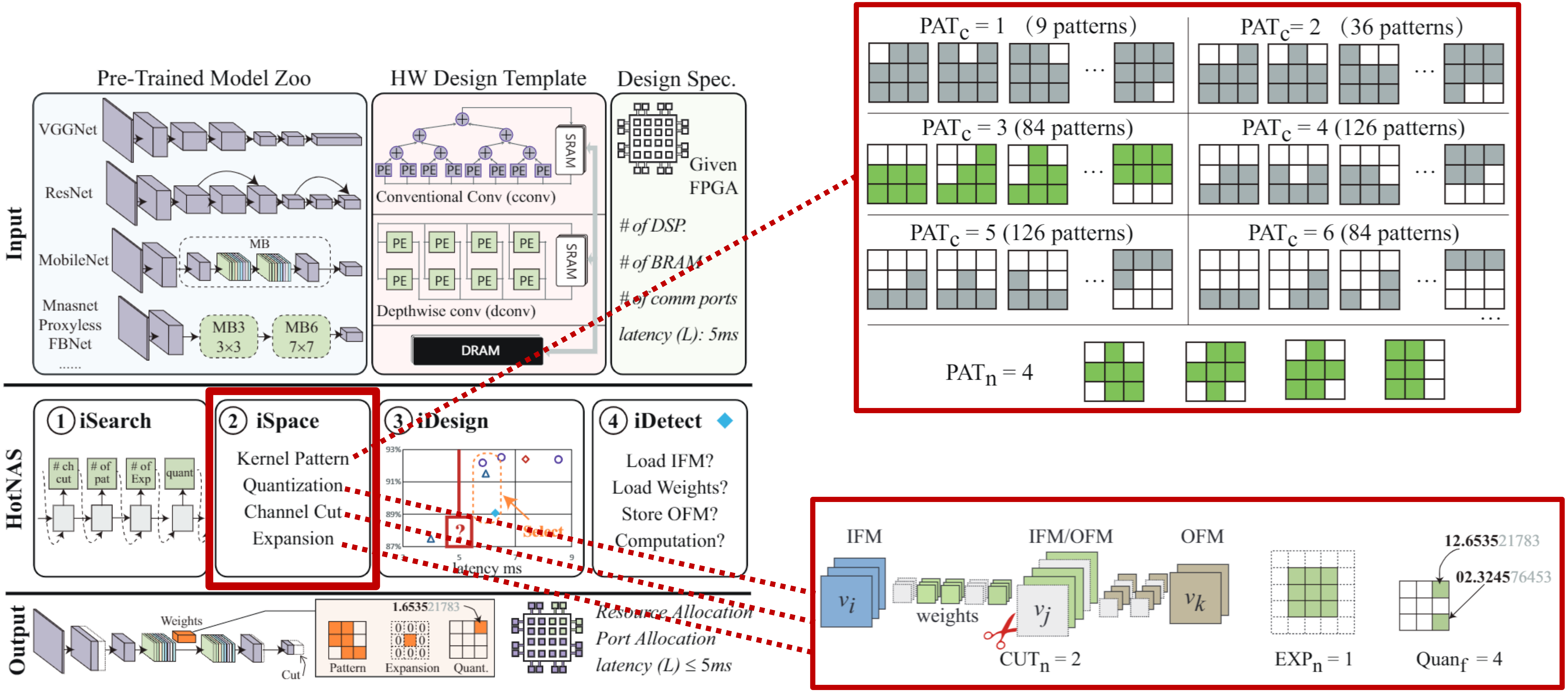
  (i.e., titling parameters, bandwidth, etc.)
- **Model compression** *(i.e., quantization, pruning)*

Objective:

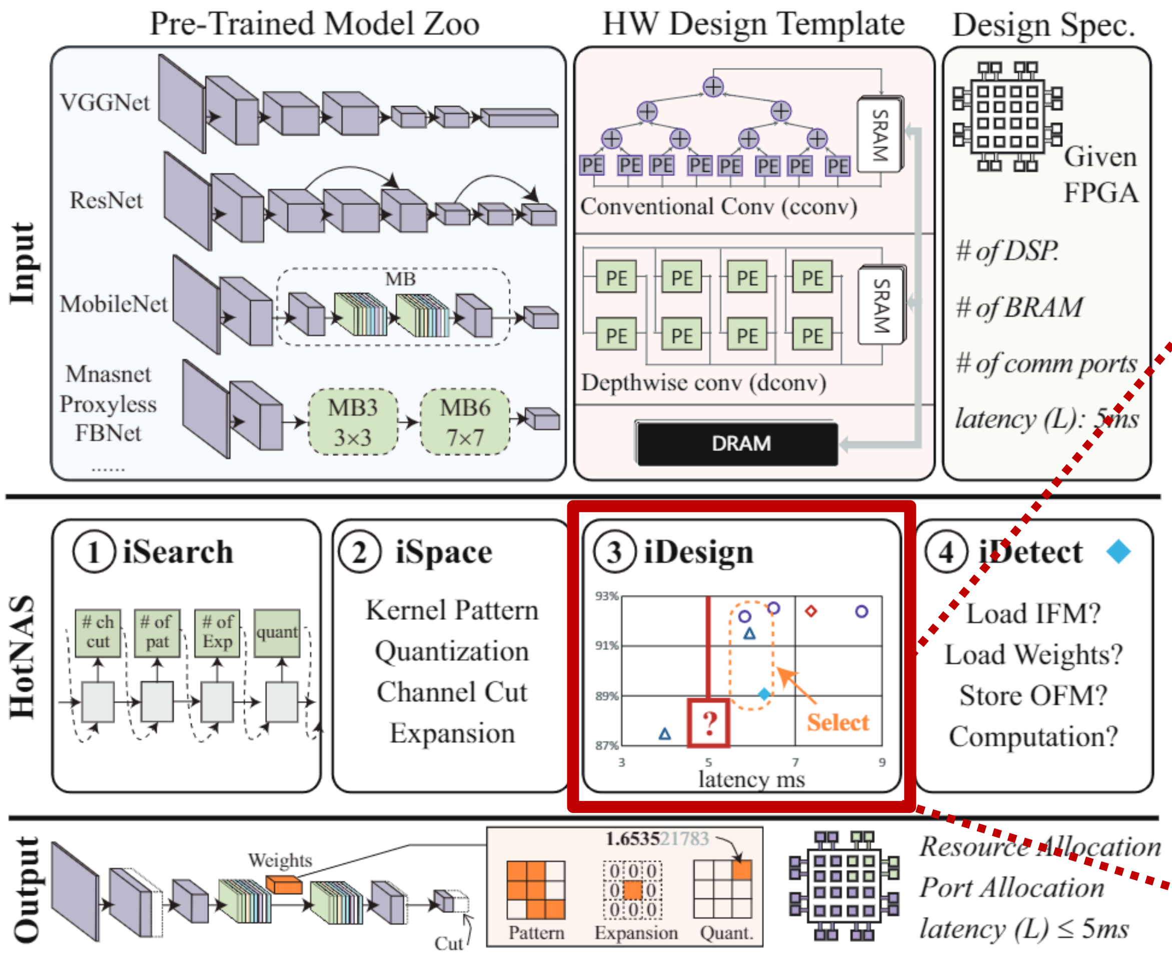
- Maximizing accuracy
- **Guarantee** latency to meet requirements

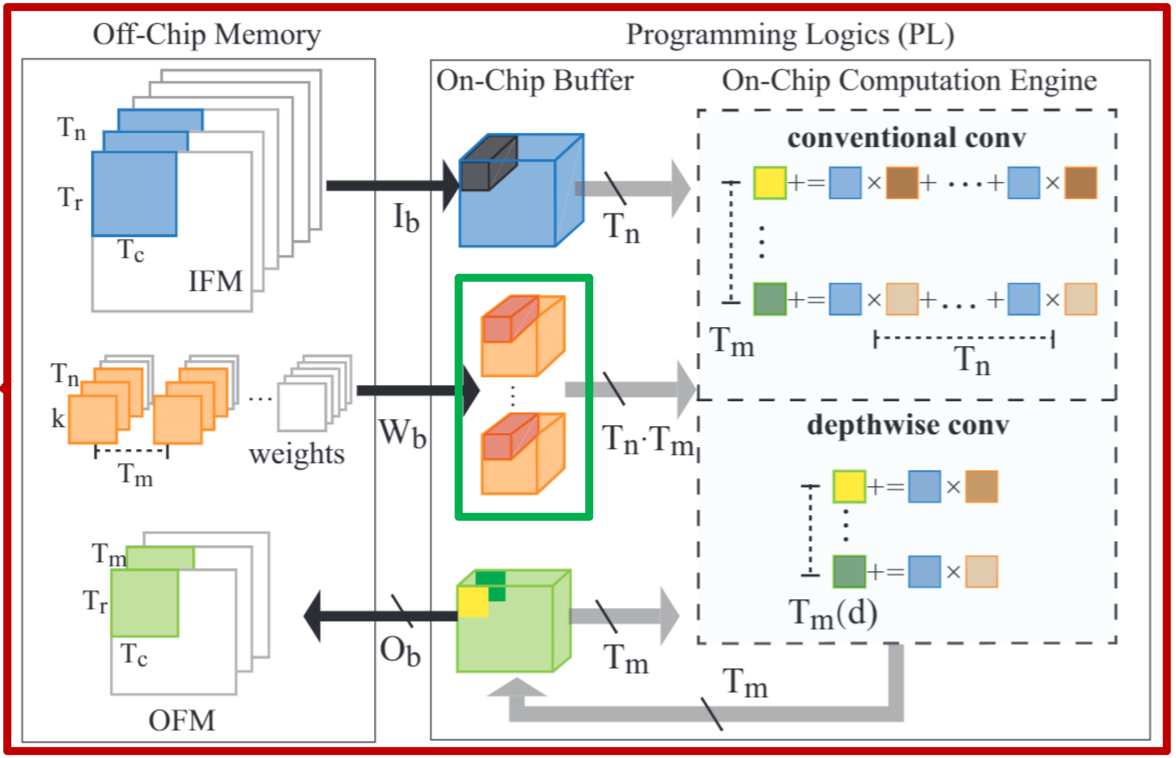


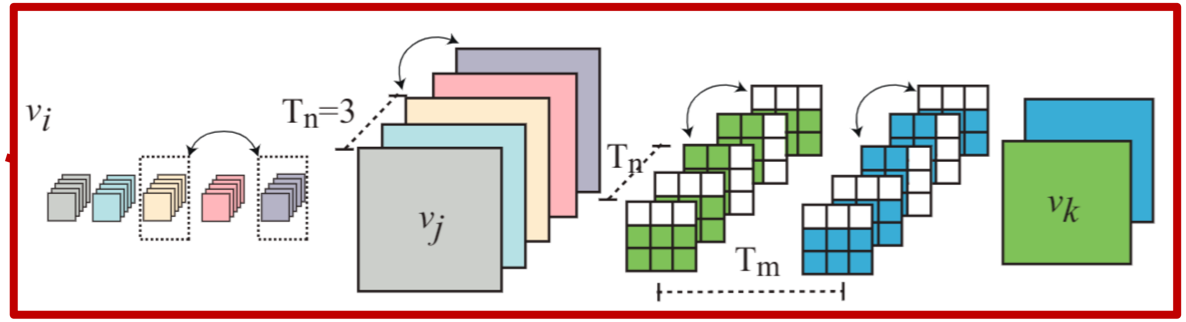
HotNAS: iSpace



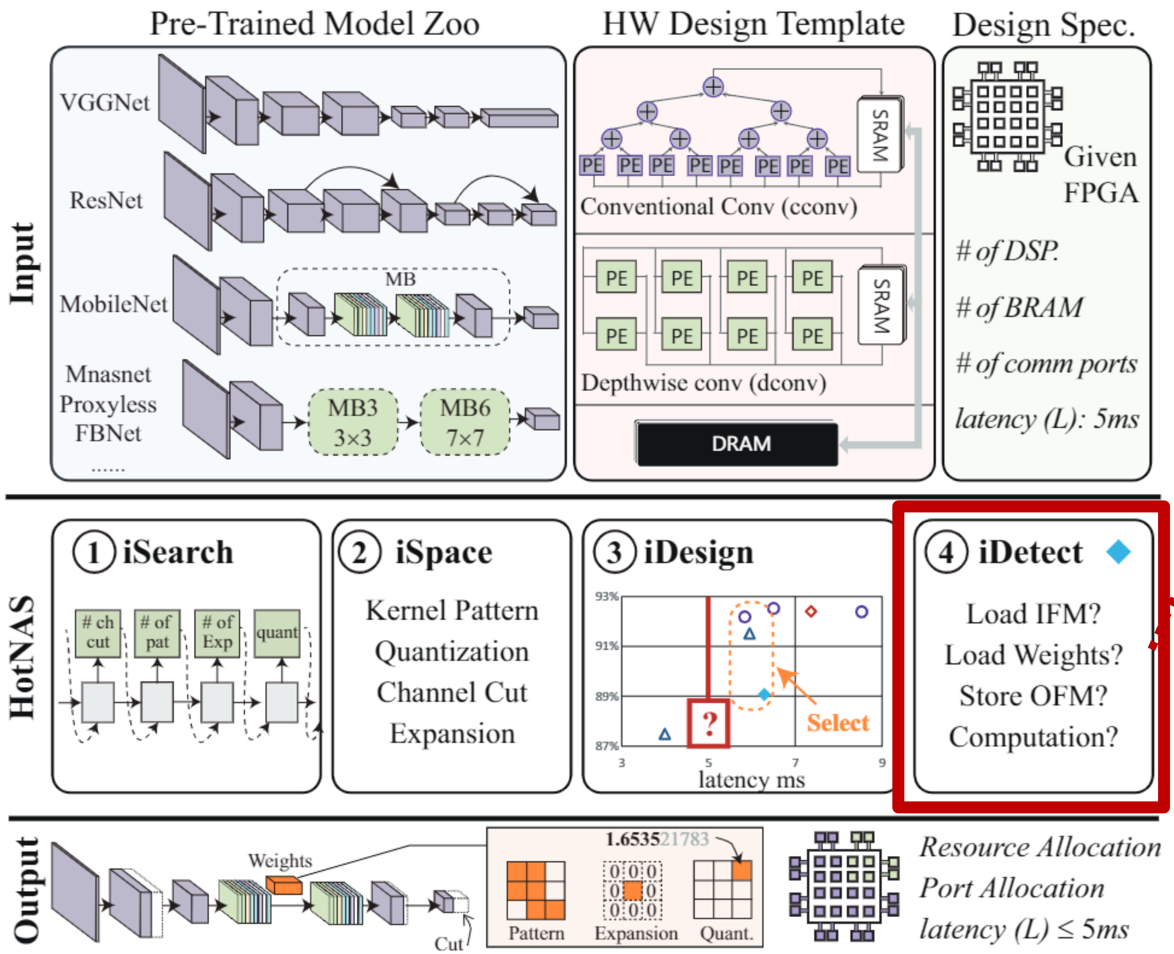
HotNAS: iDesign







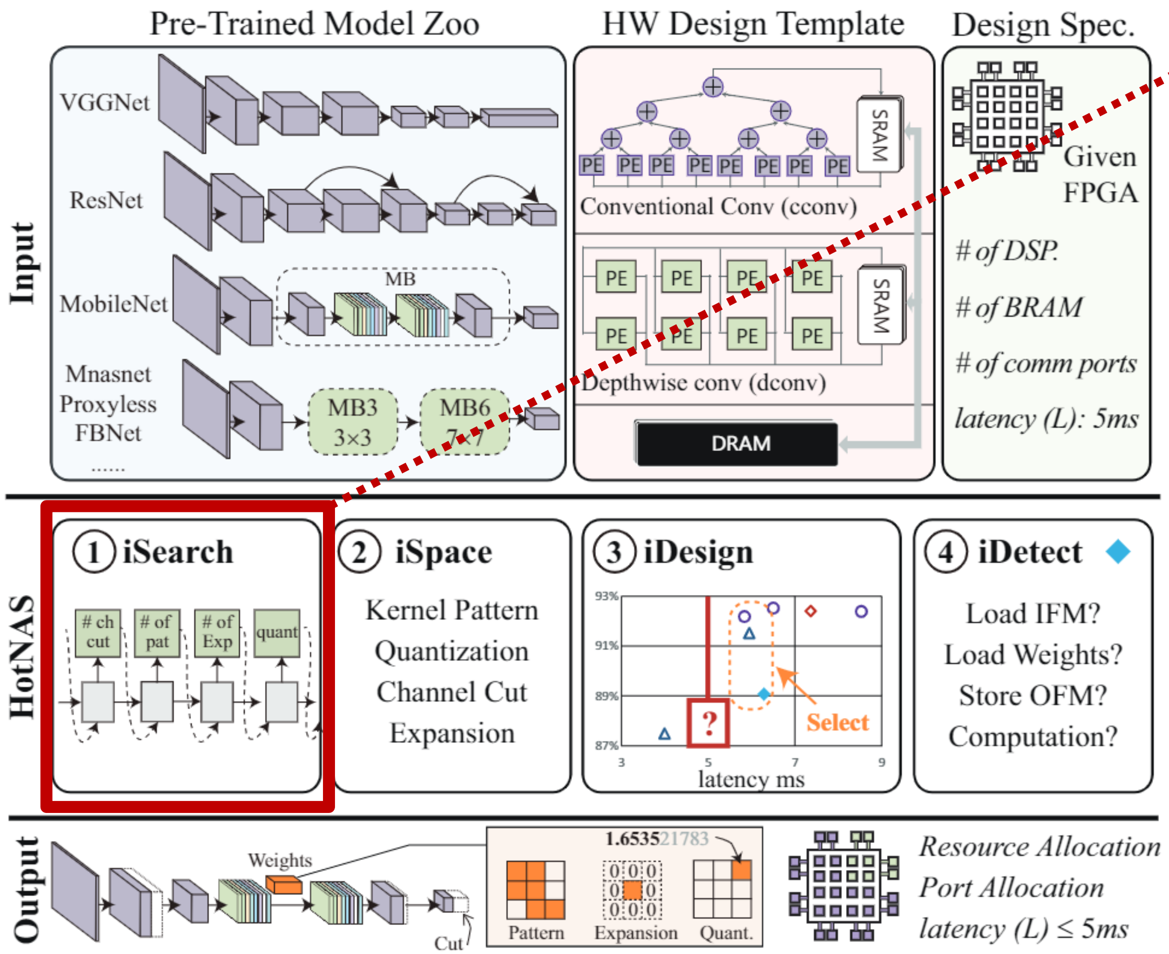
HotNAS: iDetect

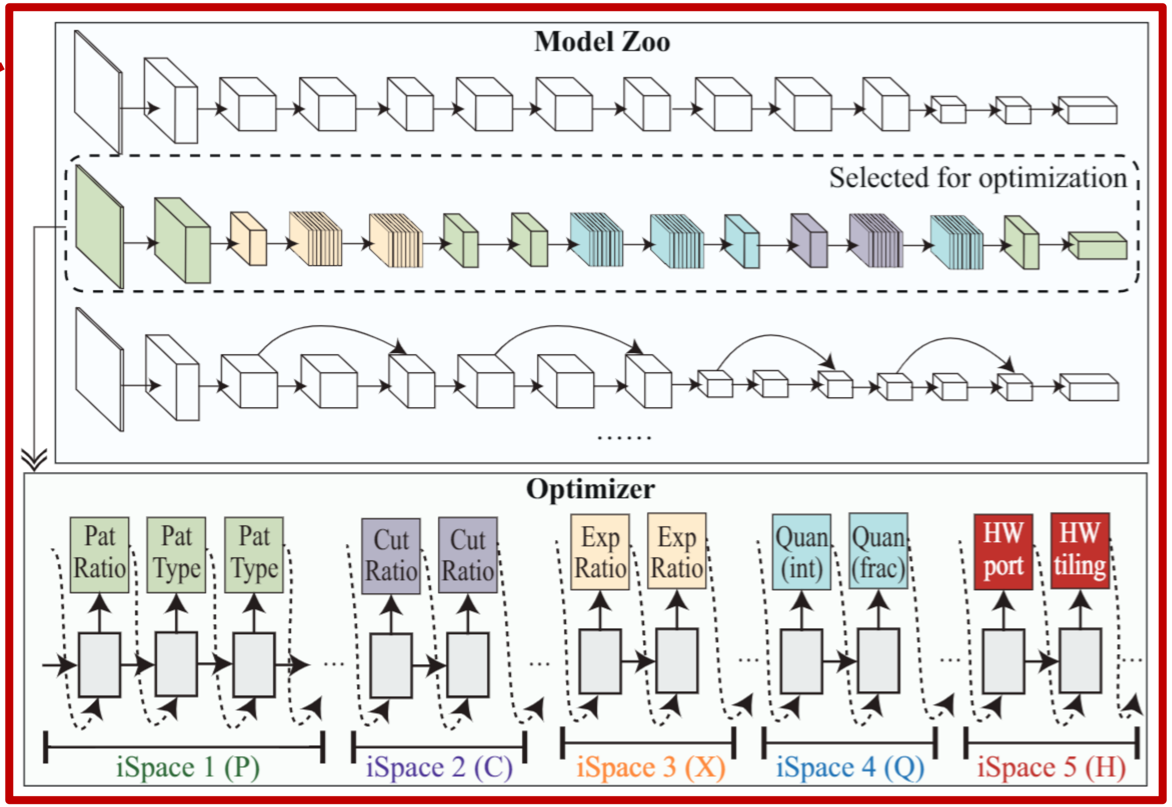


Property 2: Given a layer and design parameters, we can detect the performance bottlenecks by considering Lat_1 and Lat_2 as follows:

- **O**: if Lat_2 is dominated by tO_{mem}, the performance bottleneck is on transmitting OFM data, otherwise,
- **I**: if Lat_1 is dominated by tI_{mem}, the performance bottleneck is on transmitting IFM data,
- **W**: if Lat_1 is dominated by tW_{mem}, the performance bottleneck is on transmitting weights,
- **C**: if Lat_1 is dominated by $tComp$, we have fully utilized the involved computation resource.

HotNAS: iSearch





Results of HotNAS

HotNAS for ImageNet

On ImageNet, comparison of the state-of-the-art neural architectures with timing constraints of *5ms*

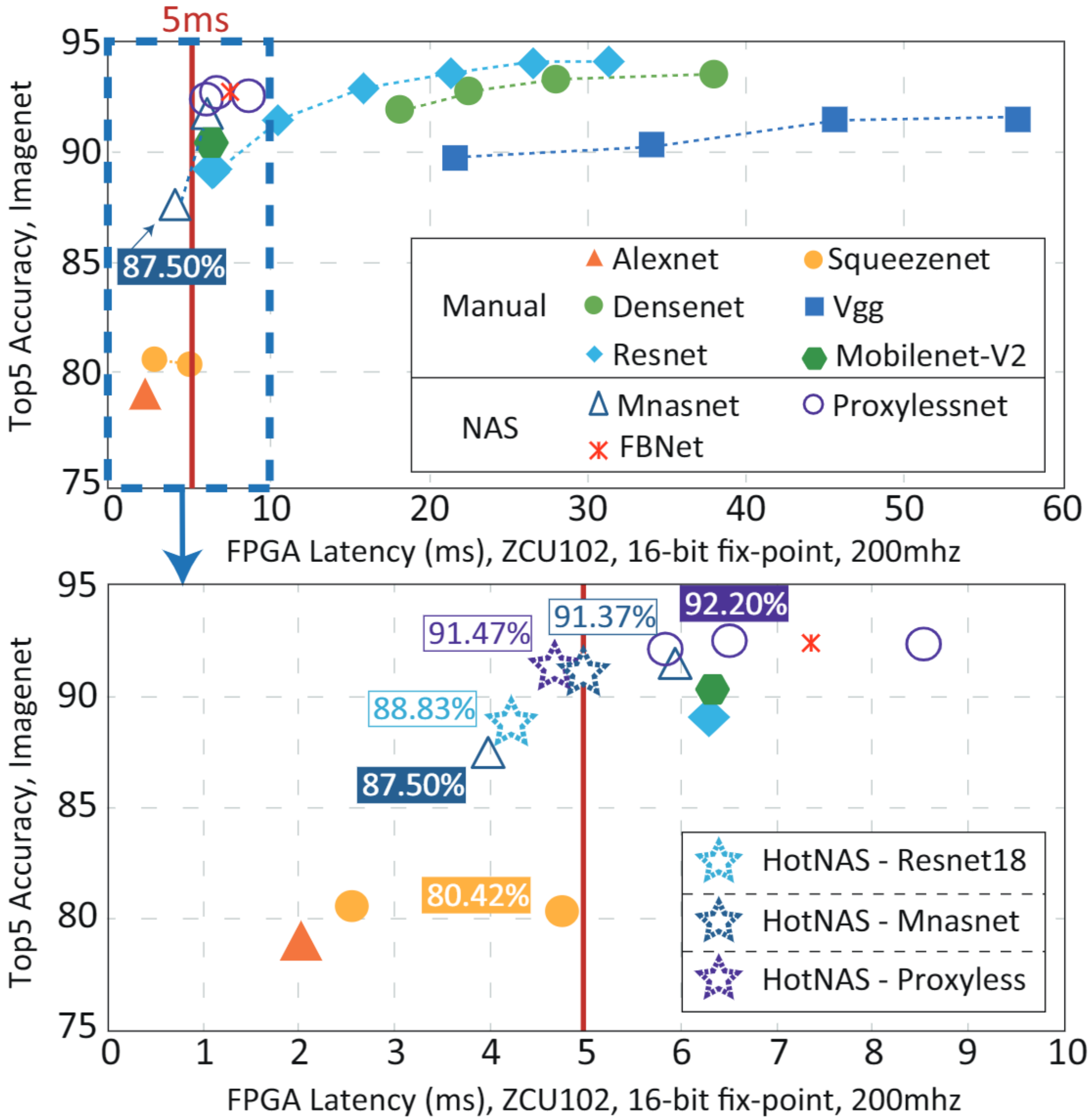
Model	Type	Latency	Sat.	Param. (#)	Param. (S)	Top-1	Top-5	Top-1 Imp.	Top-5 Imp.	GPU Time
AlexNet	manually	2.02	✓	61.1M	122.20MB	56.52%	79.07%	-	-	-
*MnasNet 0.5 **	*auto*	*3.99*	✓	*2.22M*	*4.44MB*	*67.60%*	*87.50%*	-	-	*40,000H*
SqueezeNet 1.0	manually	4.76	✓	1.25M	2.50MB	58.09%	80.42%	-	-	-
ProxylessNAS	auto	5.83	×	4.08M	8.16MB	74.59%	92.20%	-	-	200H
MnasNet	auto	5.94	×	4.38M	8.77MB	73.46%	91.51%	-	-	40,000H
Resnet	manually	6.27	×	11.69M	23.38MB	69.76%	89.08%	-	-	-
HotNAS-Resnet(4ms)	**auto**	**4.00**	✓	**10.99M**	**17.49MB**	**68.27%**	**88.21%**	**0.67%**	**0.71%**	**2H22M**
HotNAS-Resnet	**auto**	**4.22**	✓	**11.19M**	**17.90MB**	**69.14%**	**88.83%**	**1.54%**	**1.33%**	**2H01M**
HotNAS-ProxylessNAS	**auto**	**4.86**	✓	**4.38M**	**8.31MB**	**73.39%**	**91.47%**	**5.79%**	**3.97%**	**2H37M**
HotNAS-Mnasnet	**auto**	**4.99**	✓	**4.07M**	**6.56MB**	**73.24%**	**91.37%**	**5.64%**	**3.87%**	**1H50M**

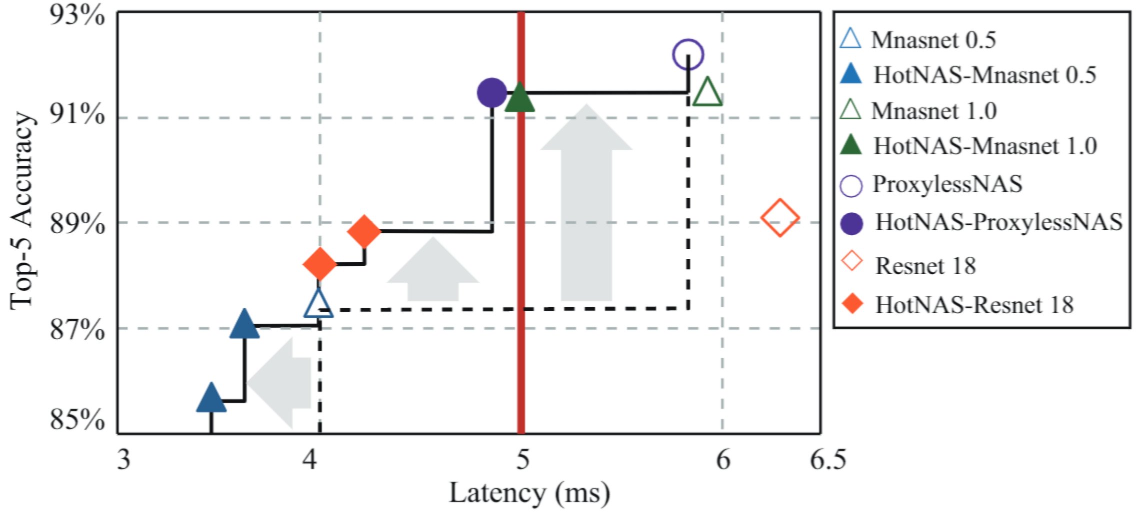
"*": baseline; "*auto & manually*": the model identified by NAS or human experts; "× & ✓": violate or meet timing constraints.

- ✓ Can guarantee accommodate the model to hardware to **satisfy the timing requirement**
- ✓ Can reduce the GPU time of co-search from 200 hours to **less than 3 hours, even using reinforcement learning**
- ✓ Can **improve the Top-1 accuracy by 5.79%** compared with the existing one that can satisfy hardware constraint

HotNAS for ImageNet: Push Forward Pareto Frontier

UNIVERSITY OF NOTRE DAME

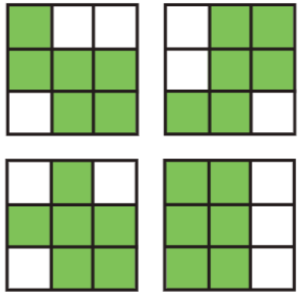




✓ Significantly push forward the Pareto frontier between the **latency and accuracy tradeoff**

✓ HotNAS works for all existing models in the model zoo to reduce the latency while keeping accuracy

HotNAS for ImageNet: Results Visualization on ResNet-18

Layers/HW	iDetect	iSpace	Exploration Results	Red. (ms)
layer1[0].conv1 layer1[0].conv2 layer1[1].conv1 layer1[1].conv2 layer2[0].conv2 layer2[1].conv1 layer2[1].conv2	C	Pattern	PATr=3, PATn=4 	0.57
layer4[0].conv1 layer4[1].conv1	I	Channel	$512 \to 480$ $512 \to 496$	0.15
layer4[0].conv1 layer4[1].conv1 layer4[0].conv2 layer4[1].conv2	- - W	Quant.	$[1, 15] \to [1, 7]$	1.01
I_b W_b	-	HW	$18 \to 20$ $6 \to 5$	0.32
Total				**2.05**

- ✓ Different technique for different layers, which is determined by iSpace.
- ✓ Hardware design exploration can further improve performance.

On CIFAR-10: HotNAS Detail Results

Model	Accuracy			Latency (ms)		
	baseline	**HotNAS**	**comp.**	baseline	**HotNAS**	**impr.**
ResNet	93.33%	**93.36%**	**+0.03%**	3.44	**1.93**	**43.90%**
DenseNet	94.14%	**94.19%**	**+0.05%**	4.01	**2.87**	**28.55%**
MobileNet	94.17%	**94.27%**	**+0.10%**	2.14	**1.79**	**16.74%**
BiTNet	97.07%	**97.13%**	**+0.06%**	6.88	**3.56**	**48.26%**

- ✓ Improve accuracy
- ✓ reduce latency

Model	1-epoch-search			fast-search		
	Accuracy	Latency	GPU Time	Accuracy	Latency	GPU Time
ResNet	93.36%	1.93	7M21S	92.74%	1.84	**3M26S**
DenseNet	94.08%	2.79	55M26S	**94.19%**	2.87	**12M04S**
MobileNet	94.27%	1.79	20M15S	94.21%	1.79	**4M26S**
BiTNet	97.13%	3.56	2H20M	97.04%	3.84	**18M44S**

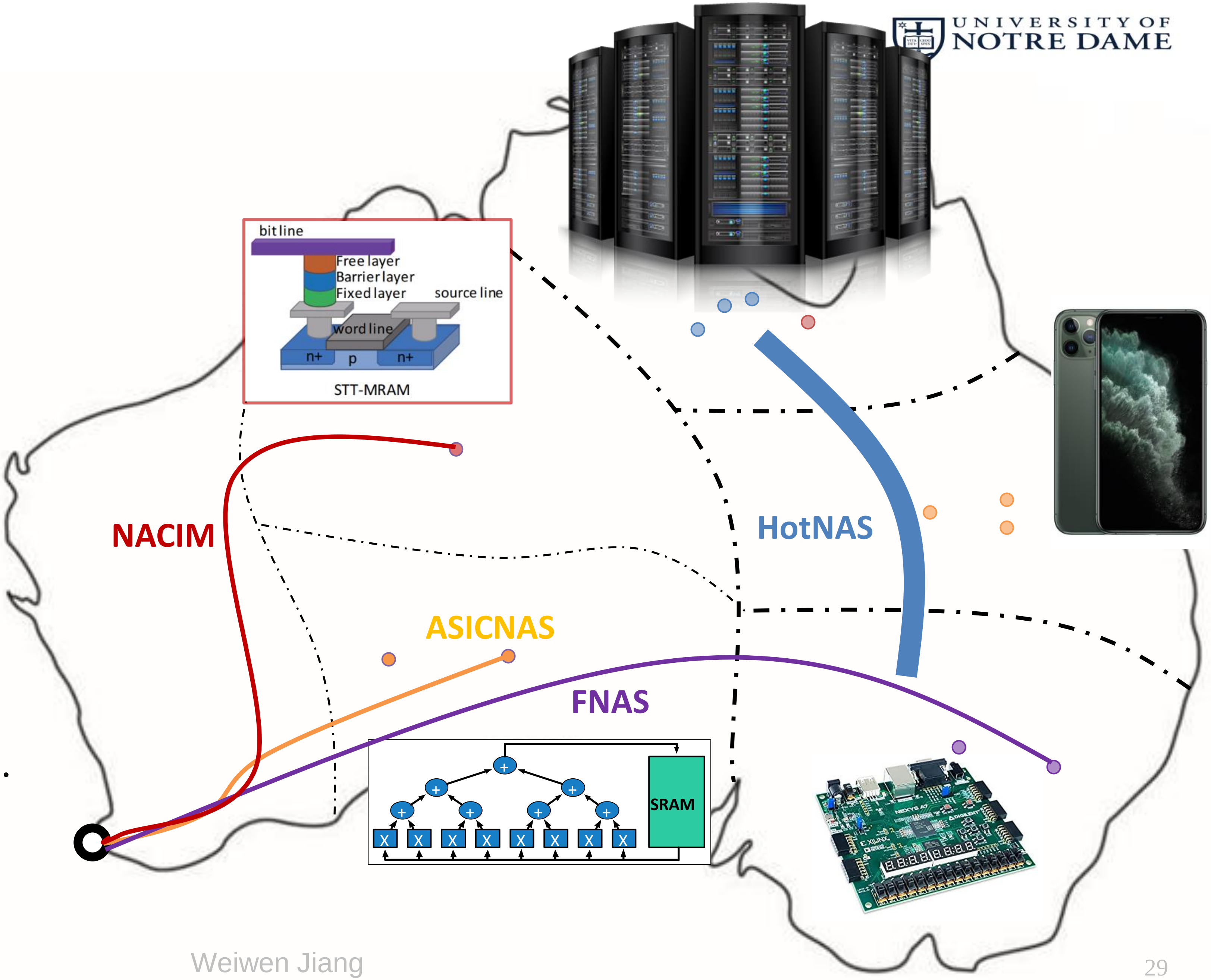
- ✓ Complete the search process in **20 minutes**
- ✓ Little or even no accuracy loss

Conclusion

- **FNAS**, **ASICNAS**, **NACIM**
  Pave ROADs for NN to different platforms.

- **HotNAS** paves a new ROAD for pre-trained NN to devices.

- Other directions?
  - ✓ Metrics: Privacy, Robustness, etc.
  - ✓ Applications: Medical, NLP, etc.
  - ✓ Models: RNN, GNN, …



Reference (1)

UNIVERSITY OF NOTRE DAME

Our related works:

[1] W. Jiang, L. Yang, S. Dasgupta, J. Hu and Y. Shi, “Standing on the Shoulders of Giants: Hardware and Neural Architecture Co-Search with Hot Start”, Accepted by **CODES+ISSS 2020**

[2] W. Jiang, Q. Lou, Z. Yan, L. Yang, J. Hu, X. S. Hu and Y. Shi, “Device-Circuit-Architecture Co-Exploration for Computing-in-Memory Neural Accelerators”, IEEE Transactions on Computers (**TC**), Accepted, 2020.

[3] W. Jiang, L. Yang, E. H.-M. Sha, Q. Zhuge, S. Gu, S. Dasgupta, Y. Shi and J. Hu, “Hardware/Software Co-Exploration of Neural Architectures”, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (**TCAD**), Accepted, 2020.

[4] L. Yang, Z. Yan, M. Li, H. Kwon, L. Lai, T. Krishana, V. Chandra, W. Jiang, and Y. Shi, “Co-Exploration of Neural Architectures and Heterogeneous ASIC Accelerator Designs Targeting Multiple Tasks”, Accepted by **DAC 2020**.

[5] B. Song, W. Jiang, Q. Lu, Y. Shi and T. Sato, “NASS: Optimizing Secure Inference via Neural Architecture Search”, Accepted by ECAI 2020.

[6] X. Yan, W. Jiang, Y. Shi and C. Zhuo, "MS-NAS: Multi-Scale Neural Architecture Search for Medical Image Segmentation," Accepted by **MICCAI 2020**

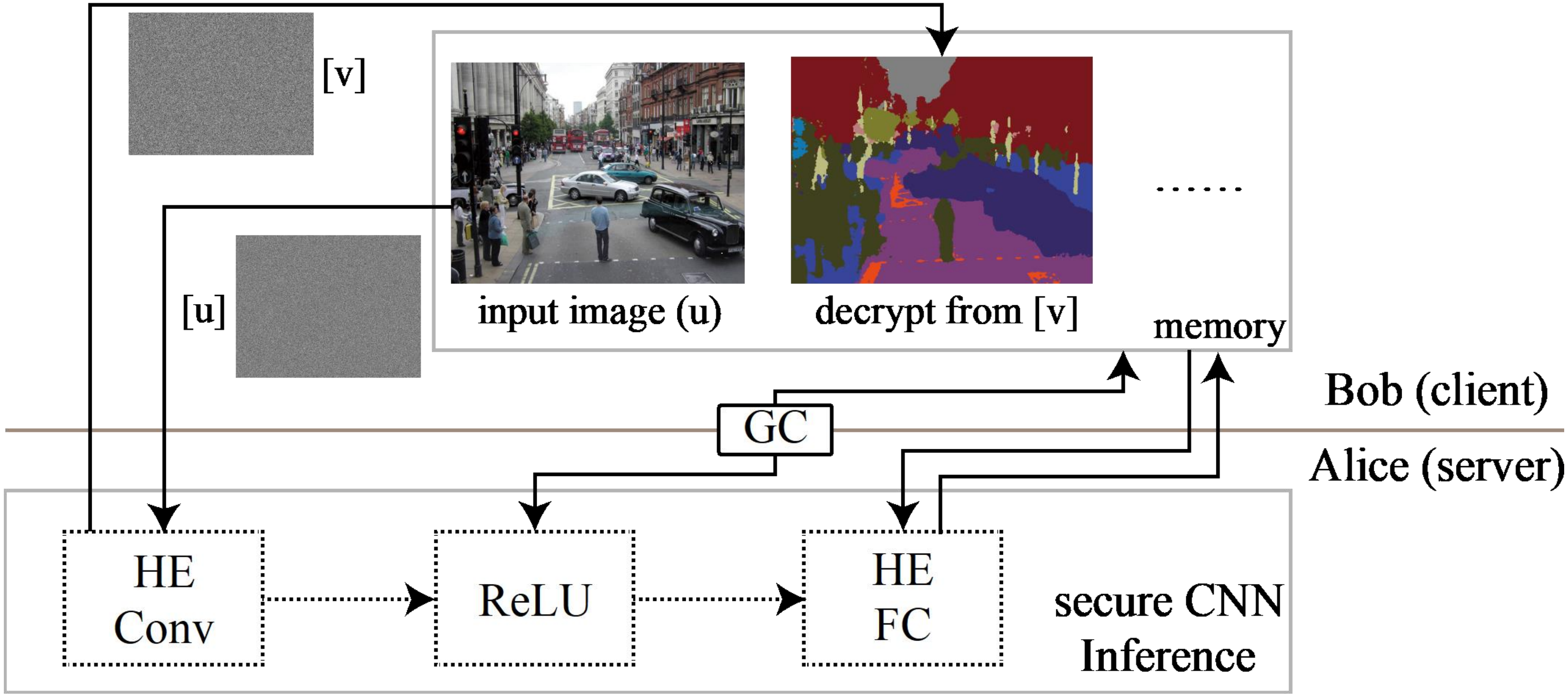
[7] L. Yang, W. Jiang, W. Liu, E. H.-M. Sha, Y. Shi and J. Hu, “Co-Exploring Neural Architecture and Network-on-Chip Design for Real-Time Artificial Intelligence”, Proc. Asia and South Pacific Design Automation Conference (**ASP-DAC**), Beijing, Jan. 2020 **(BEST PAPER NOMINATION)**

[8] W. Jiang, E. H.-M. Sha, X. Zhang, L. Yang, Q. Zhuge, Y. Shi and J. Hu, “Achieving Super-Linear Speedup across Multi-FPGA for Real-Time DNN Inference”, **CODES+ISSS’19** **(BEST PAPER NOMINATION)**

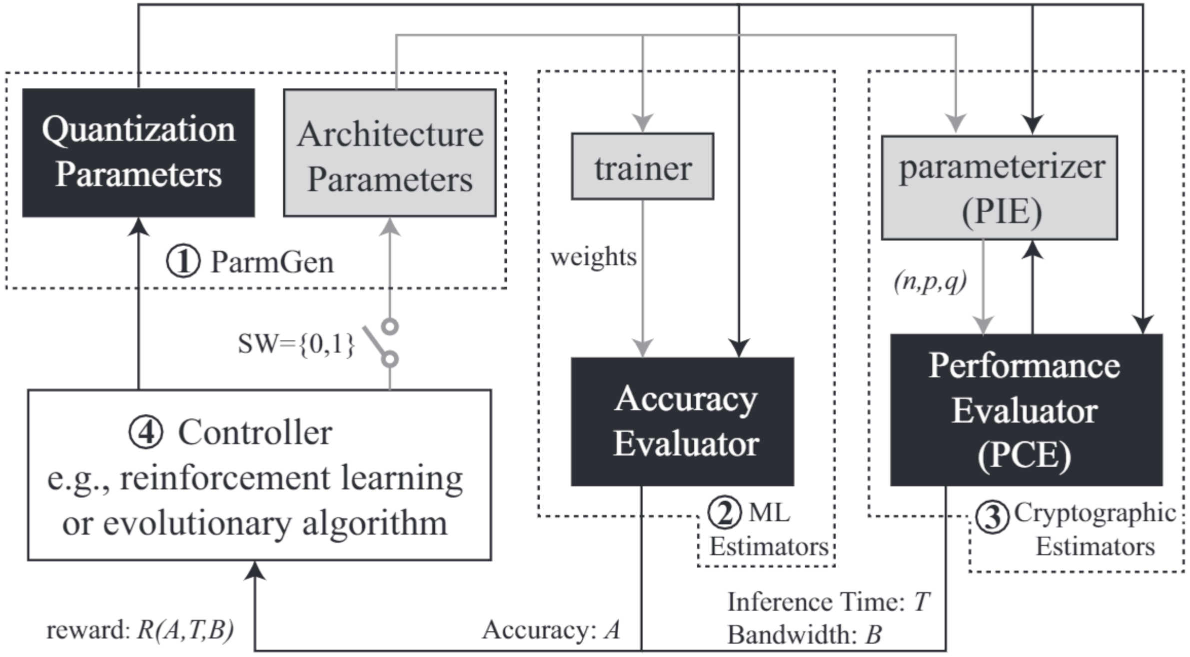
[9] W. Jiang, X. Zhang, E. H.-M. Sha, L. Yang, Q. Zhuge, Y. Shi, and J. Hu, “Accuracy vs. Efficiency: Achieving Both through FPGA-Implementation Aware Neural Architecture Search”, **DAC’19** **(BEST PAPER NOMINATION)**

NASS: Identifying Secure Inference Architecture via NAS

Privacy and Security Problems: homomorphic encryption & garbled circuits



NASS: Framework and Results



Gazelle			Best Searched by NASS		
Layer	Dimension	Quant.	Layer	Dimension	Quant.
CR	$(64 \times 3 \times 3)$	23	CR	$(24 \times 5 \times 3)$	$(8, 8)$
CR	$(64 \times 3 \times 3)$	23	CR	$(48 \times 3 \times 5)$	$(6, 7)$
PL	(2×2)	23	PL	(2×2)	$(8, 8)$
CR	$(64 \times 3 \times 3)$	23	CR	$(48 \times 5 \times 7)$	$(7, 6)$
CR	$(64 \times 3 \times 3)$	23	CR	$(36 \times 3 \times 3)$	$(6, 5)$
PL	(2×2)	23	PL	(2×2)	$(8, 8)$
CR	$(64 \times 3 \times 3)$	23	CR	$(24 \times 7 \times 1)$	$(4, 6)$
CR	$(64 \times 3 \times 3)$	23			
FC	(1024×10)	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		

- Determination of hyper-parameters and quantization
- Performance Modeling

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

Reference (2)

Related works:

[1] Zoph, B. and Le, Q.V., 2016. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

[2] Zoph, B., Vasudevan, V., Shlens, J. and Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. CVPR'18

[3] Liu, H., Simonyan, K. and Yang, Y., 2018. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055.

[4] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A. and Le, Q.V., 2019. Mnasnet: Platform-aware neural architecture search for mobile. CVPR'19.

[5] Cai, H., Zhu, L. and Han, S., 2018. Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332.

[6] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y. and Keutzer, K., 2019. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In CVPR'19.

[7] Zhang, X., Lu, H., Hao, C., Li, J., Cheng, B., Li, Y., Rupnow, K., Xiong, J., Huang, T., Shi, H. and Hwu, W.M., 2019. Skynet: a hardware-efficient method for object detection and tracking on embedded systems. arXiv preprint arXiv:1909.09709.

[8] Hao, C., Zhang, X., Li, Y., Huang, S., Xiong, J., Rupnow, K., Hwu, W.M. and Chen, D., 2019, June. FPGA/DNN Co-Design: An Efficient Design Methodology for 1oT Intelligence on the Edge. In DAC'19

[9] Li, Y., Hao, C., Zhang, X., Liu, X., Chen, Y., Xiong, J., Hwu, W.M. and Chen, D., 2020. EDD: Efficient Differentiable DNN Architecture and Implementation Co-search for Embedded AI Solutions. arXiv preprint arXiv:2005.02563.

Thank You!

HotNAS paper will be put at soon:

http://wjiang.nd.edu