

# Deep Neural Network and Accelerator Co-Design: Present and Future

Cong (Callie) Hao

Assistant Professor  
Georgia Institute of Technology  
School of Electrical and Computer Engineering



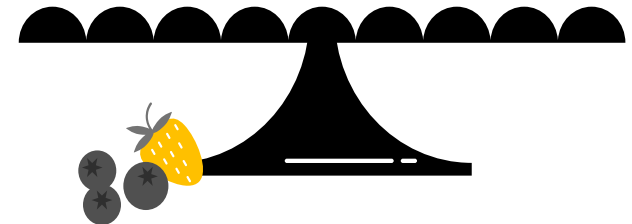
Sharclab @ Georgia Tech <https://sharclab.ece.gatech.edu/>

# What is DNN and Accelerator Co-design?

## Deep Neural Network (DNN) Design



## Accelerator Design

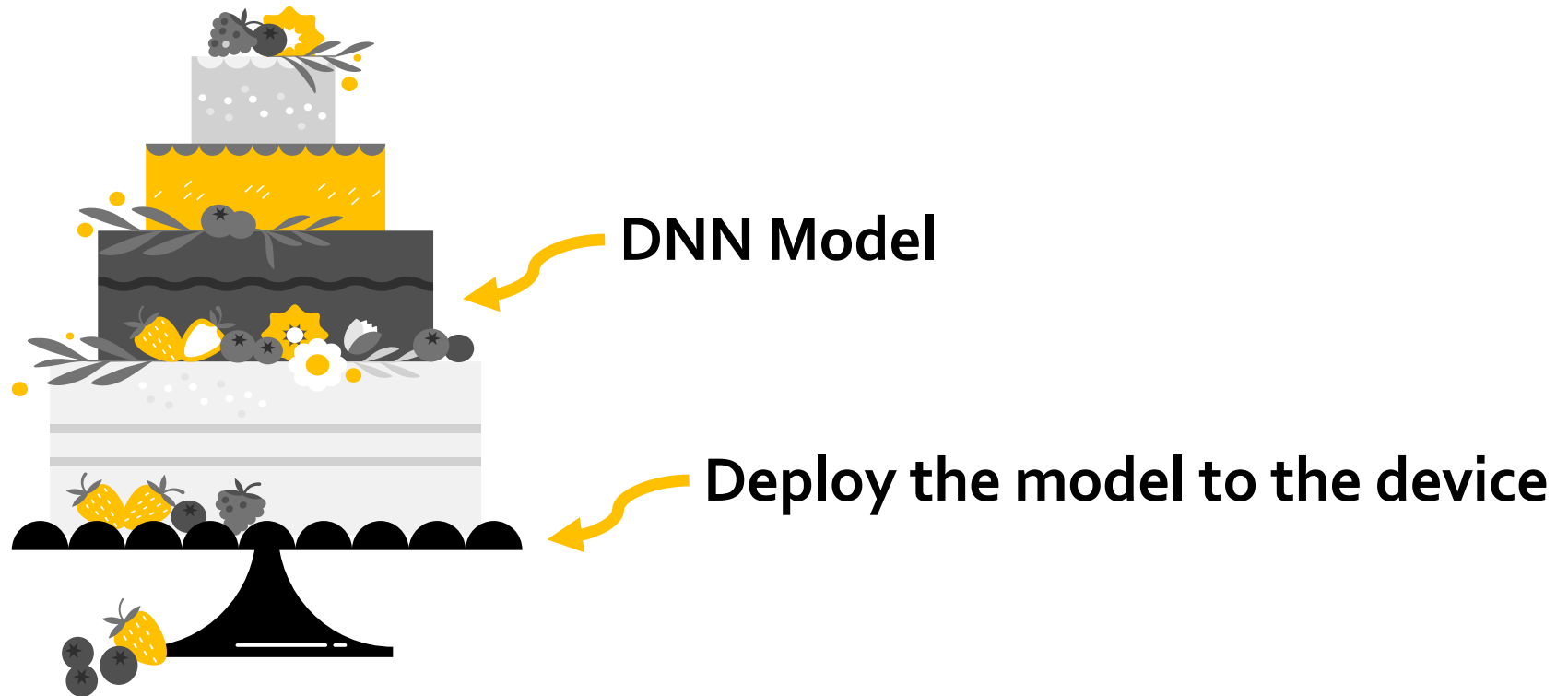


# What is DNN and Accelerator Co-design?

Deep Neural Network (DNN) Design



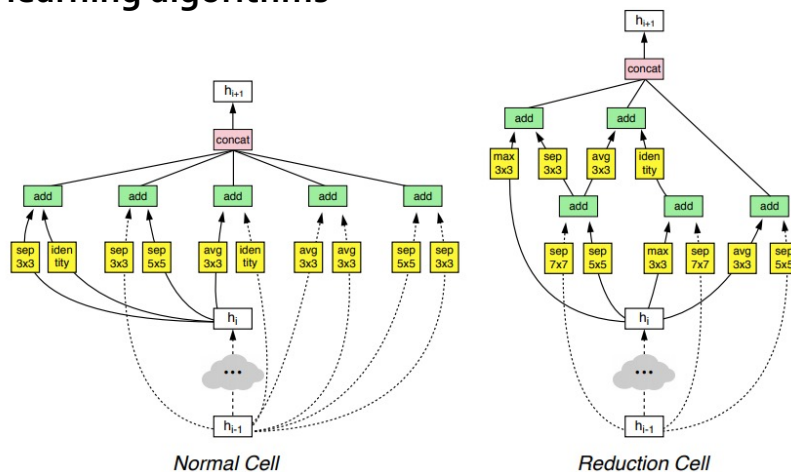
Accelerator Design



# What is DNN and Accelerator Co-design?

## Deep Neural Network (DNN) Design

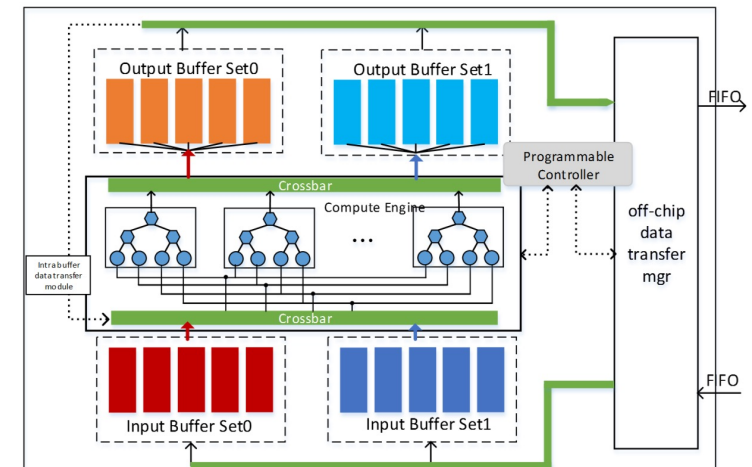
- An automatic neural architecture search (NAS) methodology – a.k.a. AutoML
- Boosts the quality and accuracy of machine learning algorithms



Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." CVPR 2018

## Accelerator Design

- On GPU/TPU/NPU: optimized (tuned) neural network implementations
- On **FPGA**: **customized** DNN accelerators



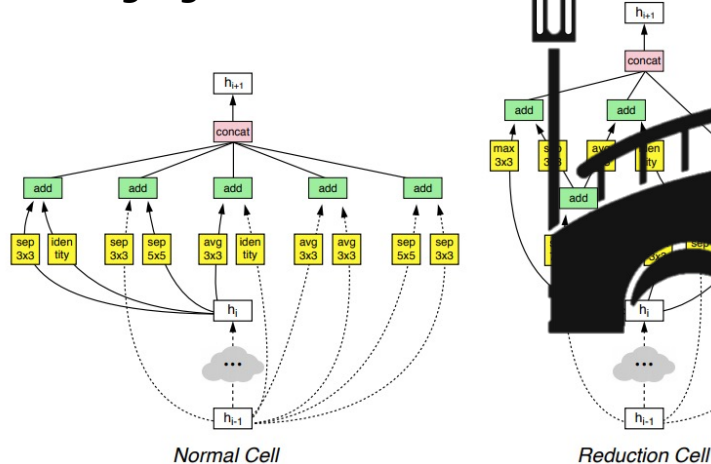
Chen Zhang, et al. "Optimizing fpga-based accelerator design for deep convolutional neural networks", FPGA 2015



# What is DNN and Accelerator Co-design?

## Deep Neural Network (DNN) Design

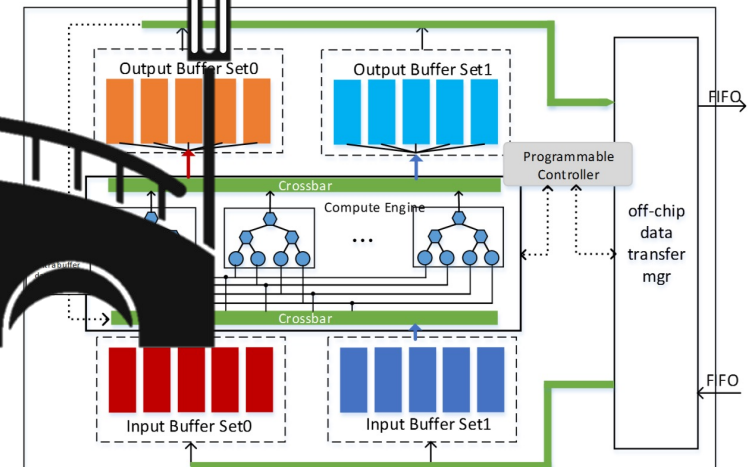
- An automatic neural architecture search (NAS) methodology – a.k.a. AutoML
- Boosts the quality and accuracy of machine learning algorithms



Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." CVPR 2018

## Accelerator Design

- On GPU/TPU/NPU: optimized (tuned) neural network implementations
- On **FPGA**: **customized** DNN accelerators



Chen Zhang, et al. "Optimizing fpga-based accelerator design for deep convolutional neural networks", FPGA 2015

# Three Levels of Co-Design in Cake Factory...

Level 0

Bake a cake...



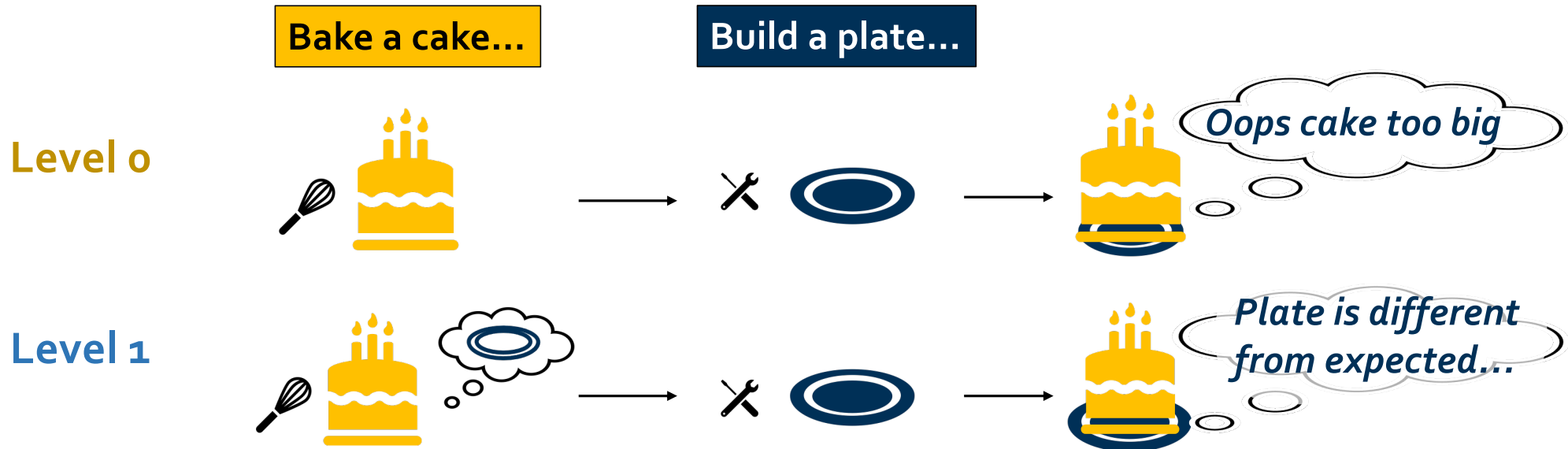
Build a plate...



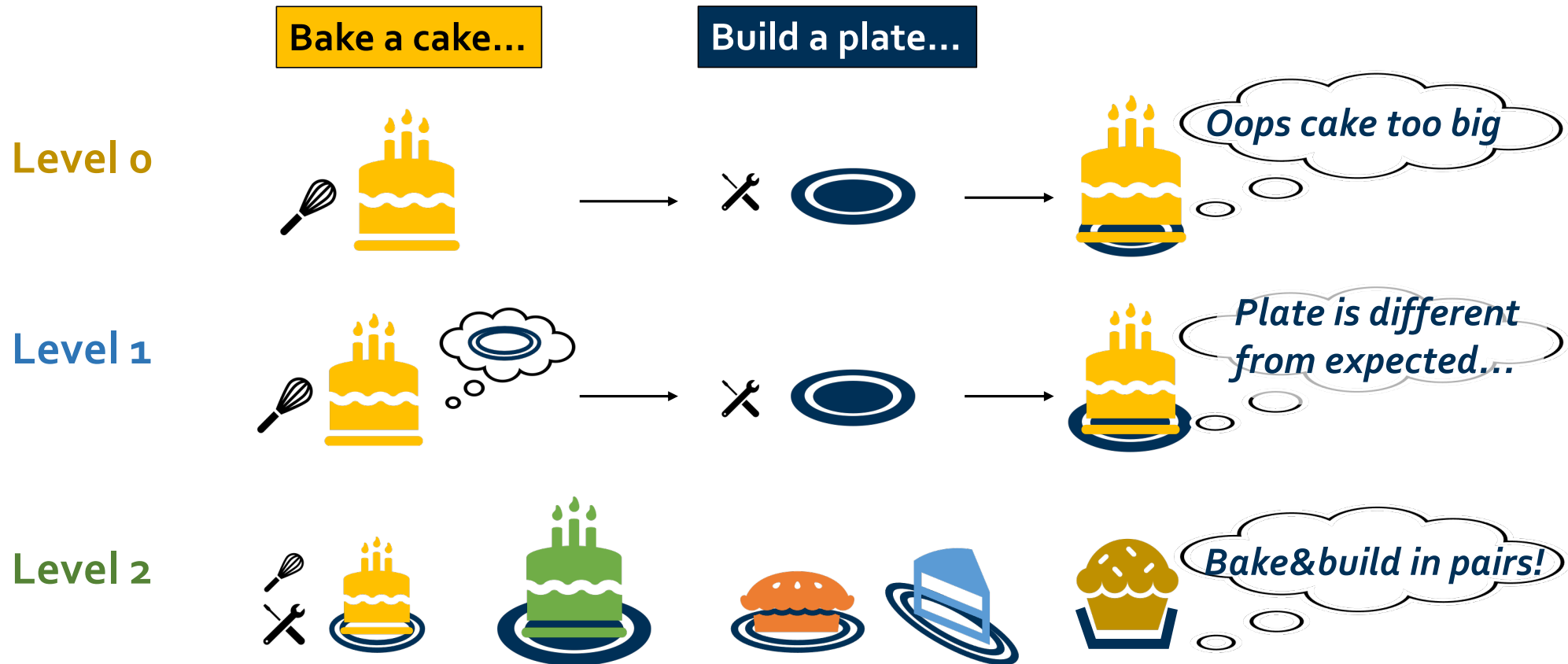
*Oops cake too big*



# Three Levels of Co-Design in Cake Factory...

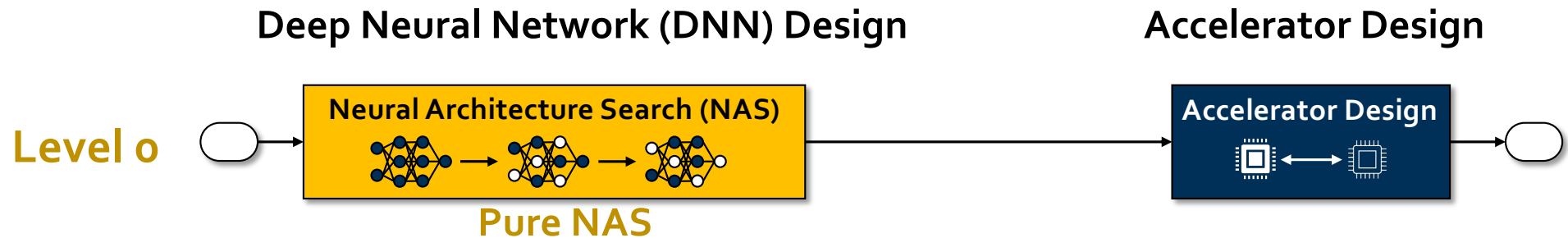


# Three Levels of Co-Design in Cake Factory...

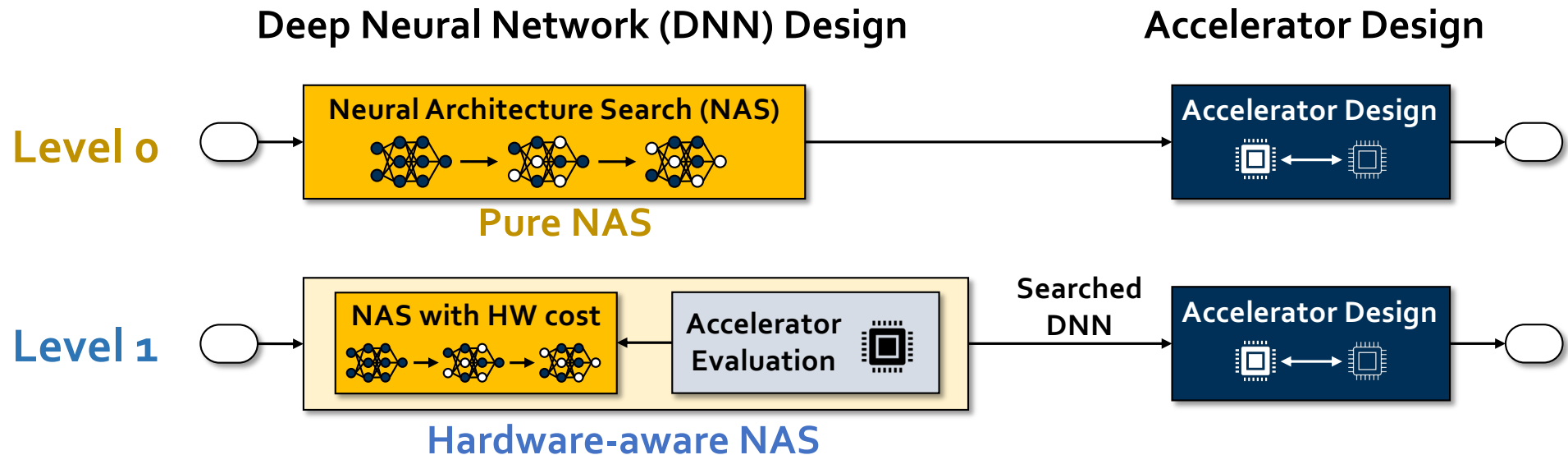




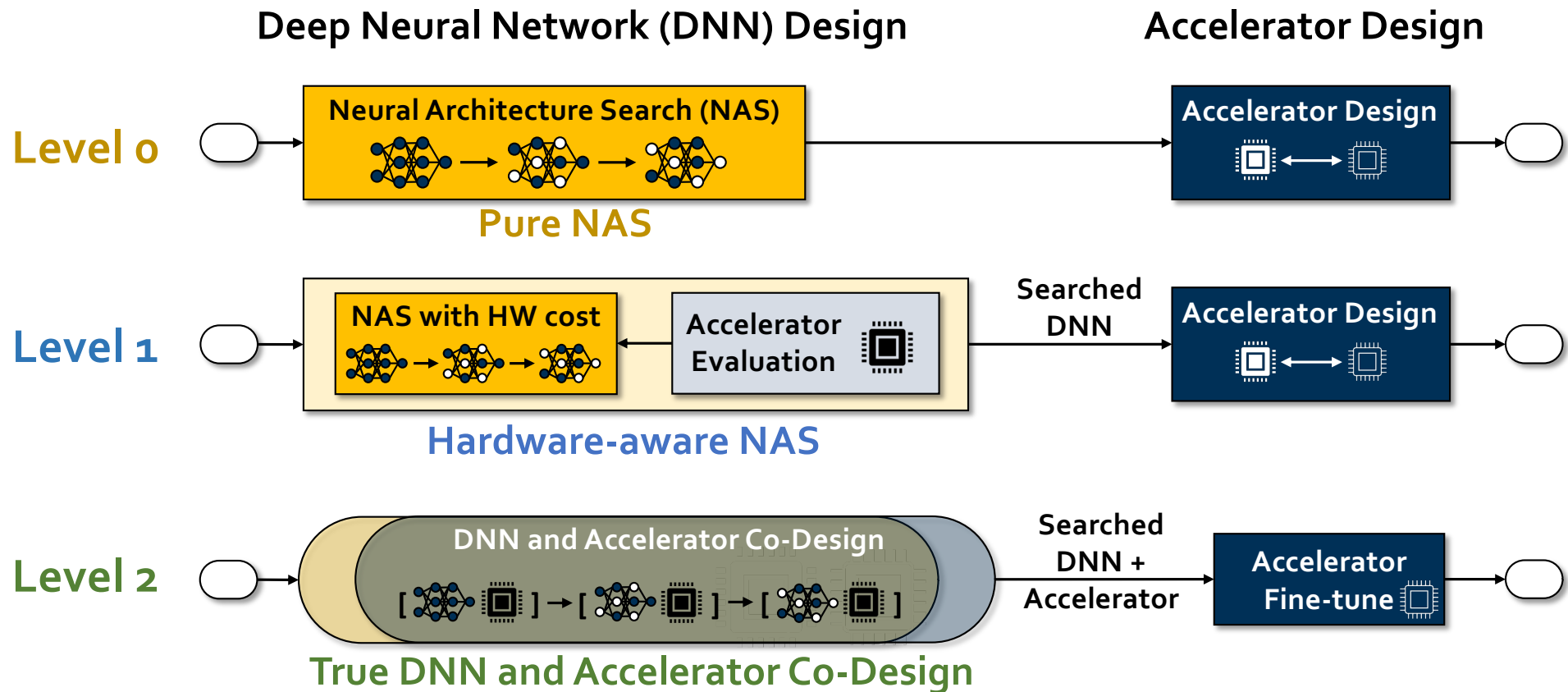
# Three Levels of Co-Design for DNN/Accelerator



# Three Levels of Co-Design for DNN/Accelerator



# Three Levels of Co-Design for DNN/Accelerator

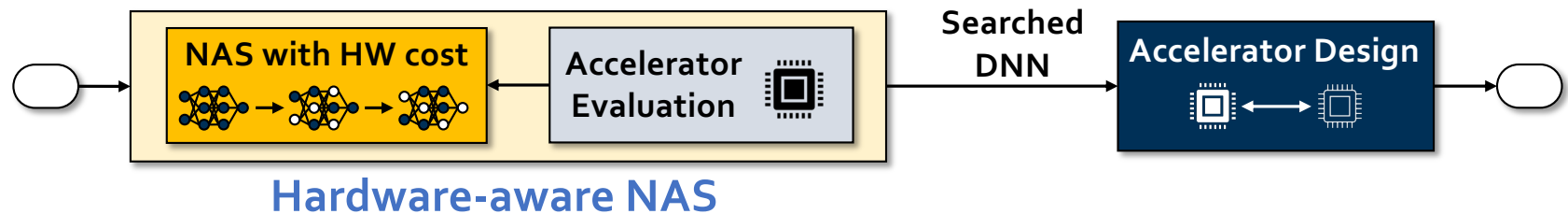


# Three Levels of Co-Design for DNN/Accelerator

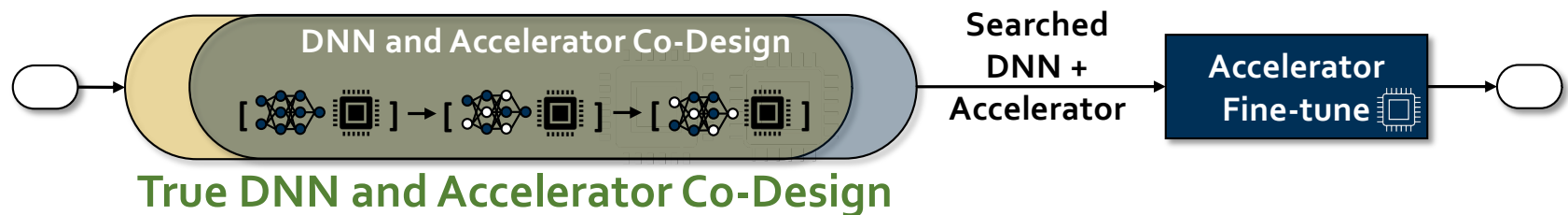
Deep Neural Network (DNN) Design

Accelerator Design

Level 1



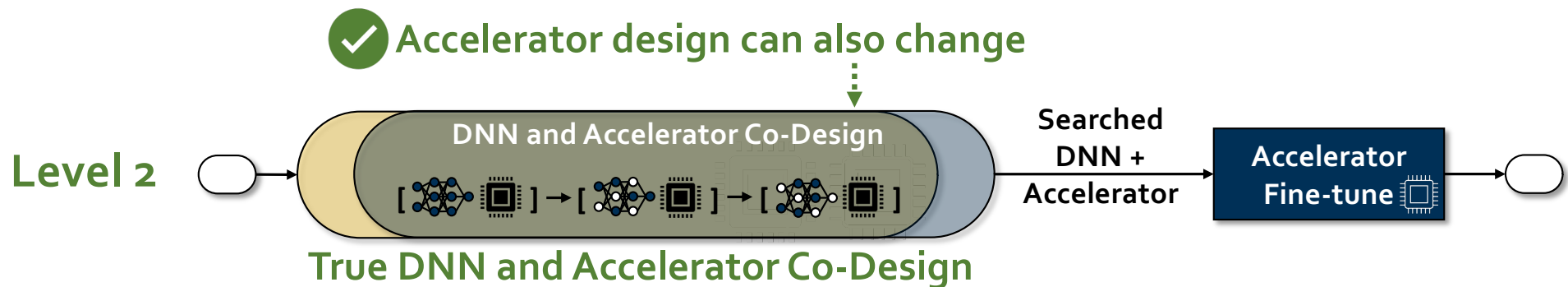
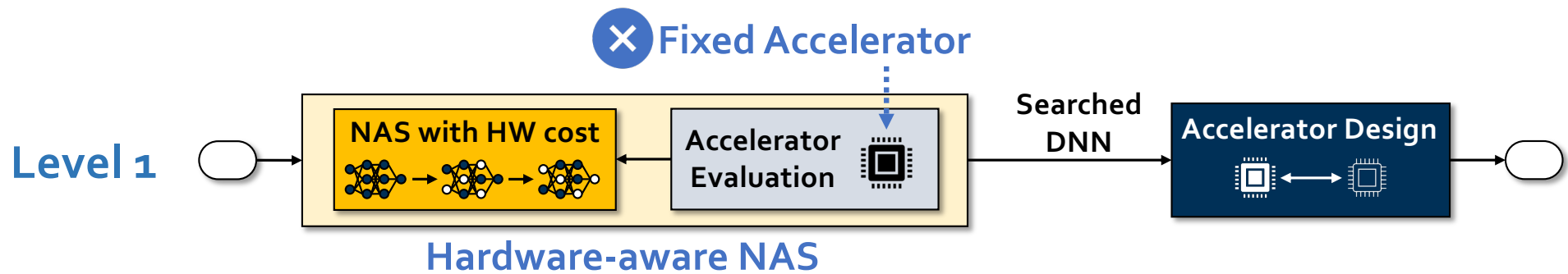
Level 2



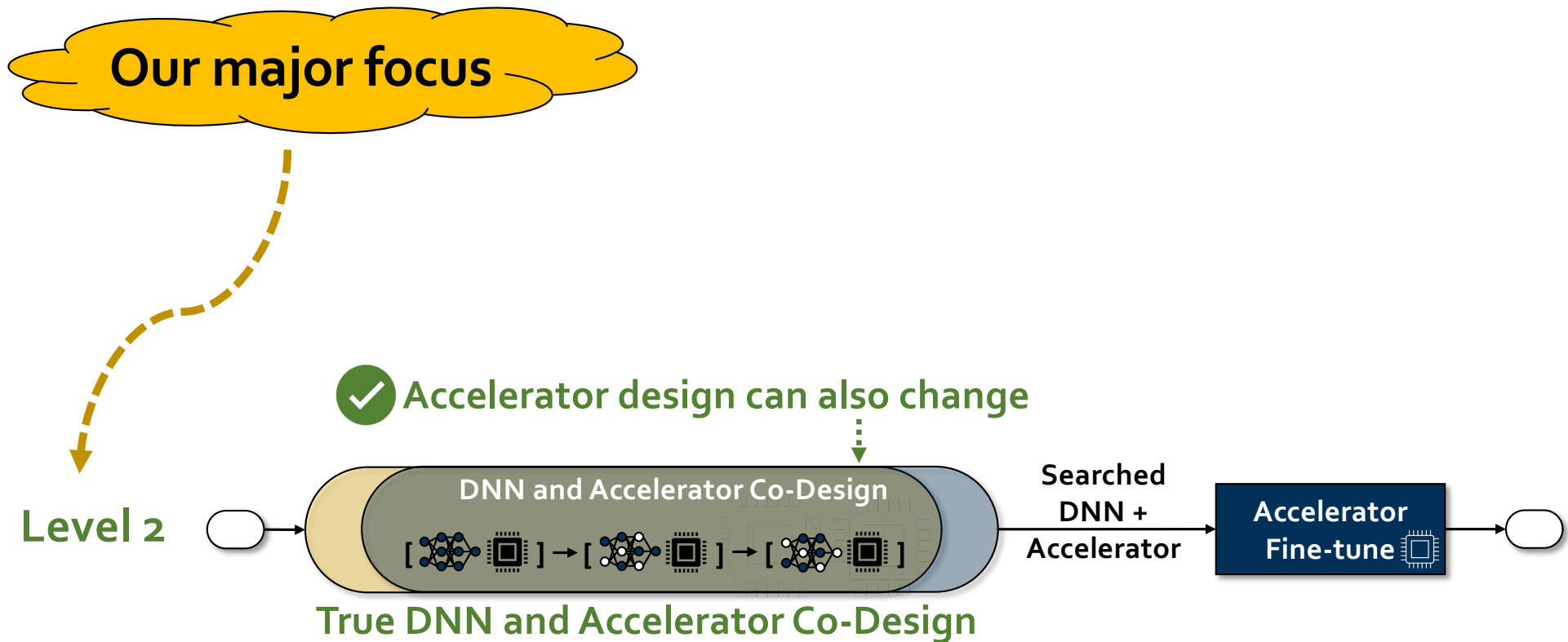
# Three Levels of Co-Design for DNN/Accelerator

Deep Neural Network (DNN) Design

Accelerator Design



# Level 2 Co-Design for DNN/Accelerator



# Level 2 Co-Design for DNN/Accelerator

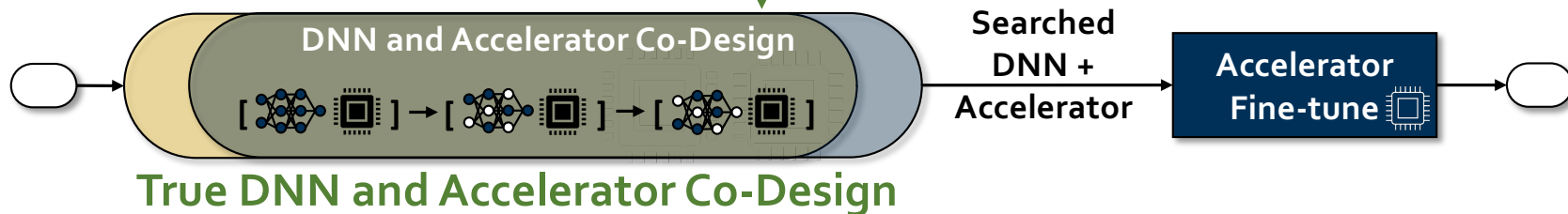
Our major focus

Software: Neural Architecture Search (NAS)

Hardware: Implementation Search

Level 2

✓ Accelerator design can also change



# Level 2 Co-Design for DNN/Accelerator

Our major focus

Software: Neural Architecture Search (NAS)

Hardware: Implementation Search

+

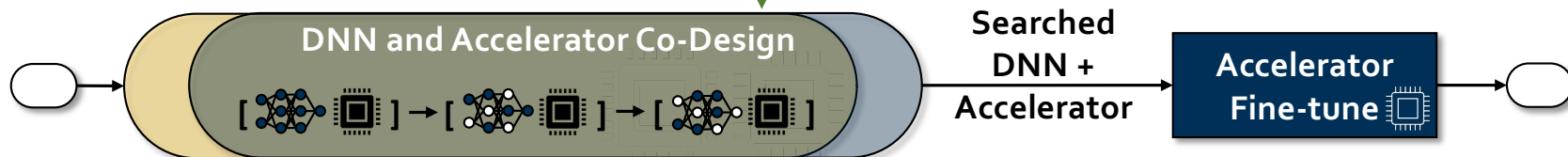
||

NAIS



Level 2

✓ Accelerator design can also change



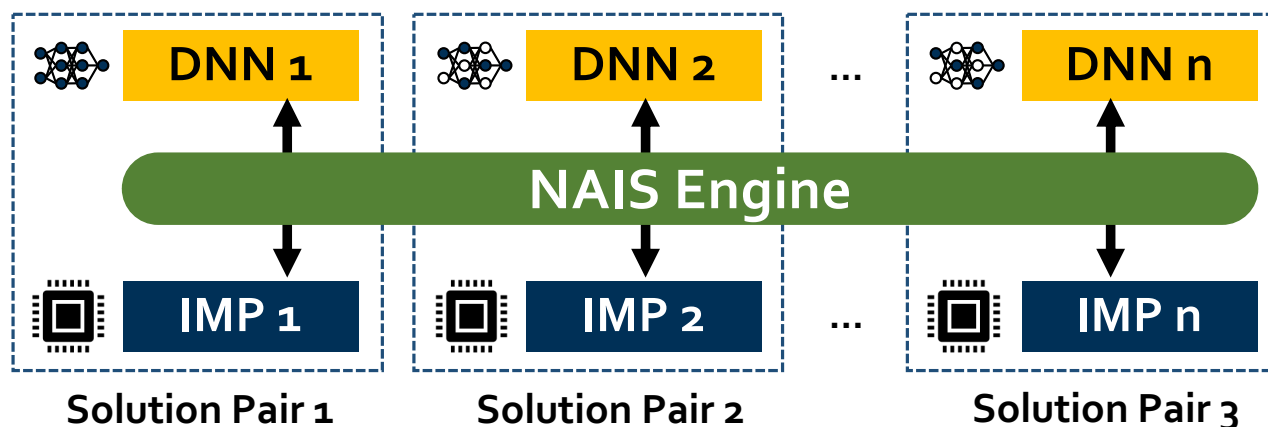
True DNN and Accelerator Co-Design





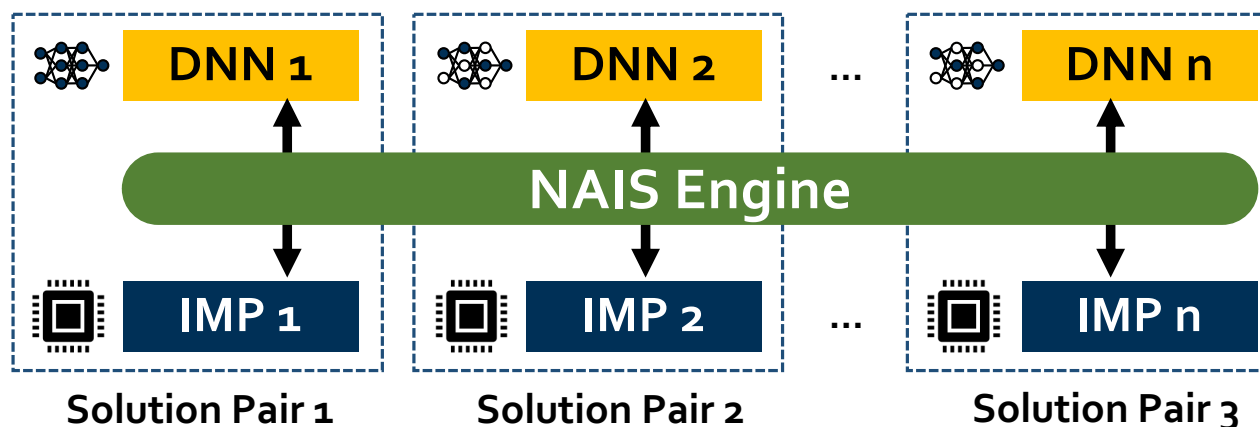
# NAIS: Simultaneous NAS + Implementation

- Simultaneous NAS and Implementation search



# NAIS: Simultaneous NAS + Implementation

- Simultaneous NAS and Implementation search



## "One-click AI"

- Automated AI algorithm development and deployment



## "Good AI"

- Bridge the gap between SW/HW for higher quality solutions



# Key Methodologies for Co-Design

Software:

Neural Architecture  
Search Space

Hardware:

Implementation Search  
Space



# Key Methodologies for Co-Design

Software:

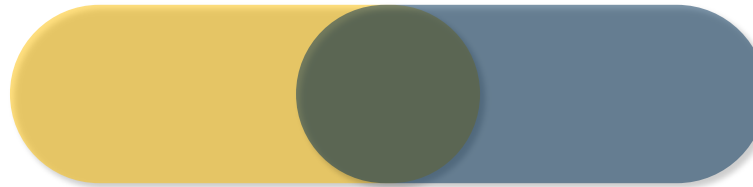
Neural Architecture  
Search Space

Hardware:

Implementation Search  
Space

1

Method 1: find something  
in the middle and connect  
to both SW and HW



FPGA/DNN Co-Design  
[DAC'19, SysML'20]

# Key Methodologies for Co-Design

## Software:

Neural Architecture  
Search Space

## Hardware:

Implementation Search  
Space

**1** Method 1: find something  
in the middle and connect  
to both SW and HW



FPGA/DNN Co-Design  
[DAC'19, SysML'20]

**2** Method 2: merge the  
two spaces – formulate  
both in one equation



A True NAIS work  
EDD [ICCAD'19, DAC'20]

# Key Methodologies for Co-Design

Software:

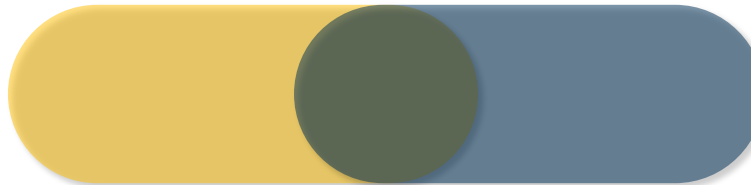
Neural Architecture  
Search Space

Hardware:

Implementation Search  
Space

1

Method 1: find something  
in the middle and connect  
to both SW and HW



FPGA/DNN Co-Design  
[DAC'19, SysML'20]

Hao, Cong, Xiaofan Zhang, Yuhong Li, Sitao Huang, Jinjun Xiong, Kyle Rupnow, Wen-mei Hwu, and Deming Chen. "FPGA/DNN co-design: An efficient design methodology for IoT intelligence on the edge." *ACM/IEEE DAC*, 2019. (seems to be most cited in DAC 2019)



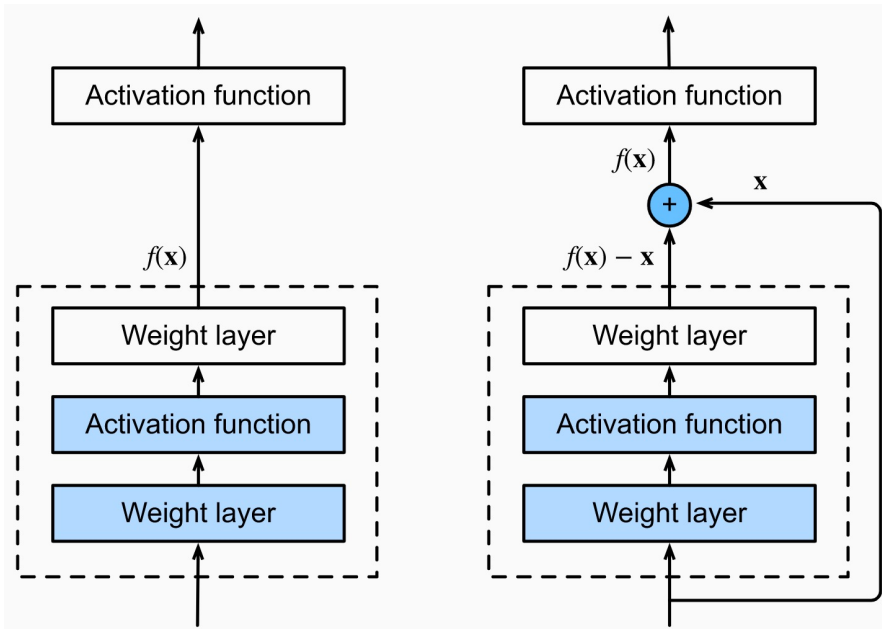
# FPGA/DNN Co-Design

- DNNs

- Accelerators (FPGA)



- **DNNs** are usually built by repeated or similar **basic blocks**



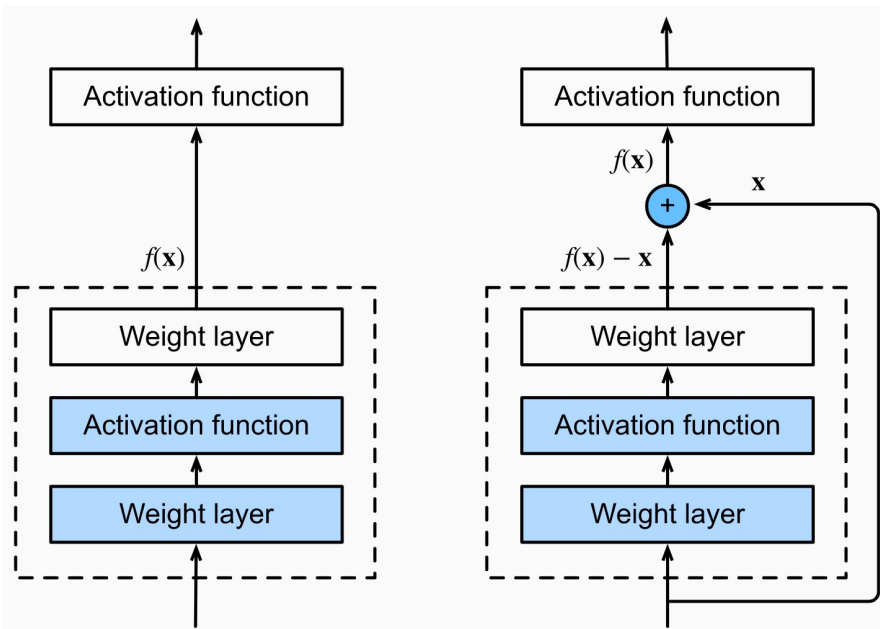
[https://d2l.ai/chapter\\_convolutional-modern/resnet.html](https://d2l.ai/chapter_convolutional-modern/resnet.html)

- **Accelerators (FPGA)**



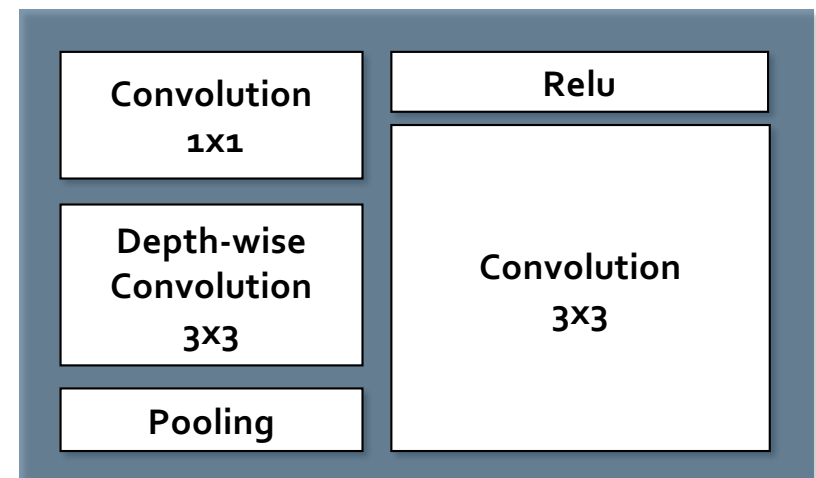


- DNNs are usually built by repeated or similar basic blocks



[https://d2l.ai/chapter\\_convolutional-modern/resnet.html](https://d2l.ai/chapter_convolutional-modern/resnet.html)

- Accelerators (FPGA) are usually built by Processing Elements (PE)

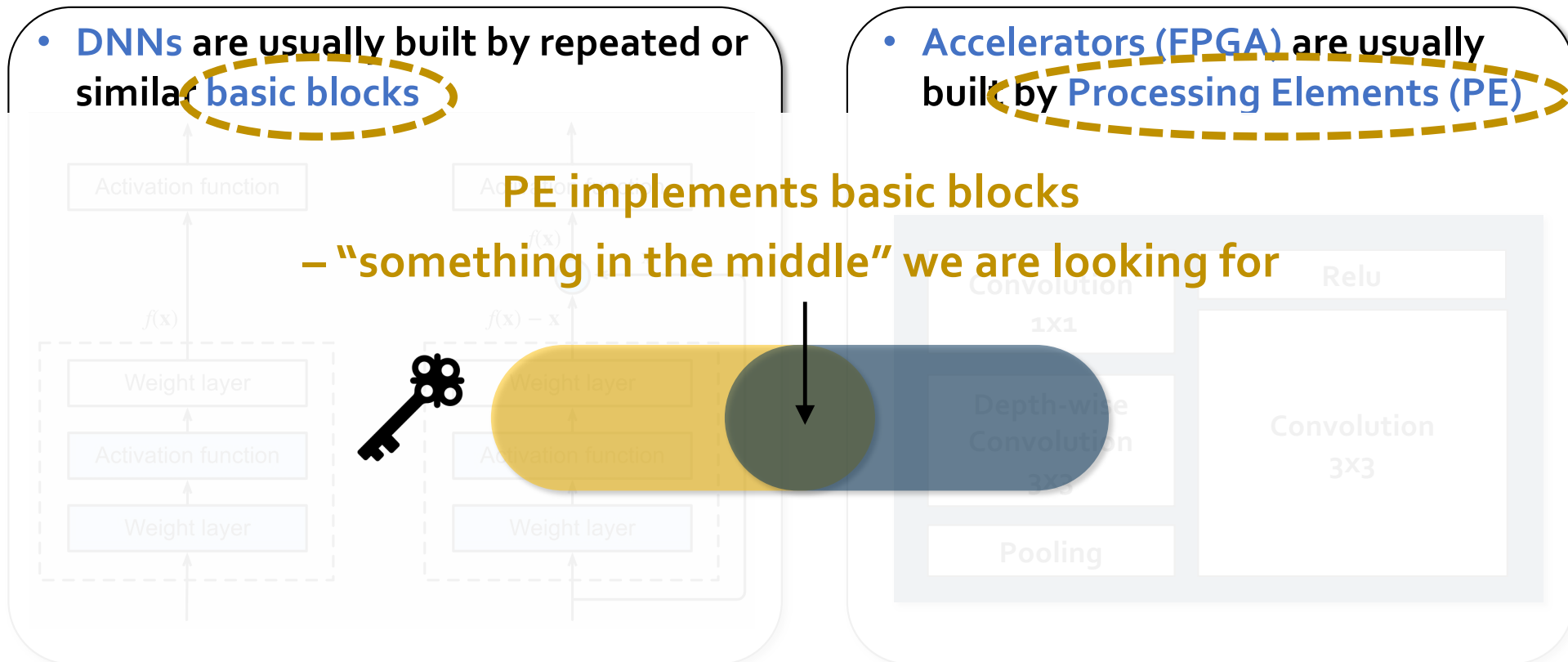


- DNNs are usually built by repeated or similar **basic blocks**

- Accelerators (FPGA) are usually built by **Processing Elements (PE)**

PE implements basic blocks

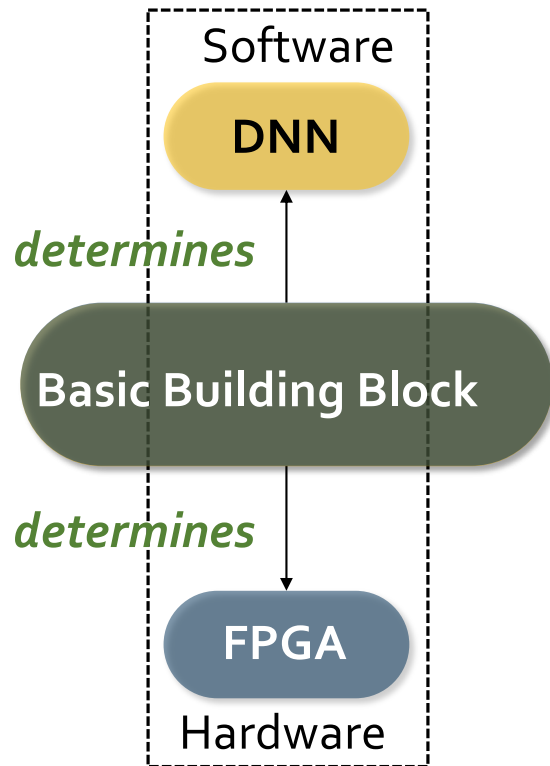
– “something in the middle” we are looking for



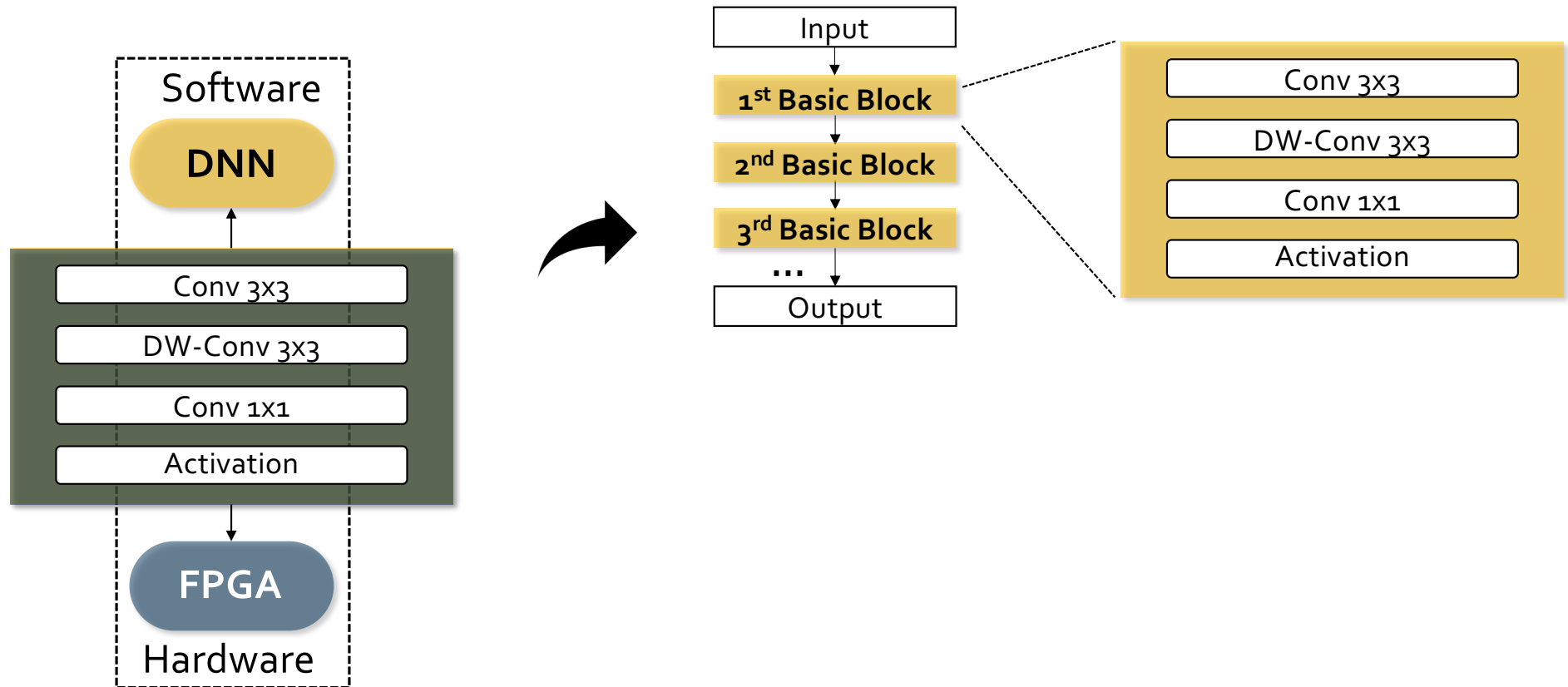
[https://d2l.ai/chapter\\_convolutional-modern/resnet.html](https://d2l.ai/chapter_convolutional-modern/resnet.html)



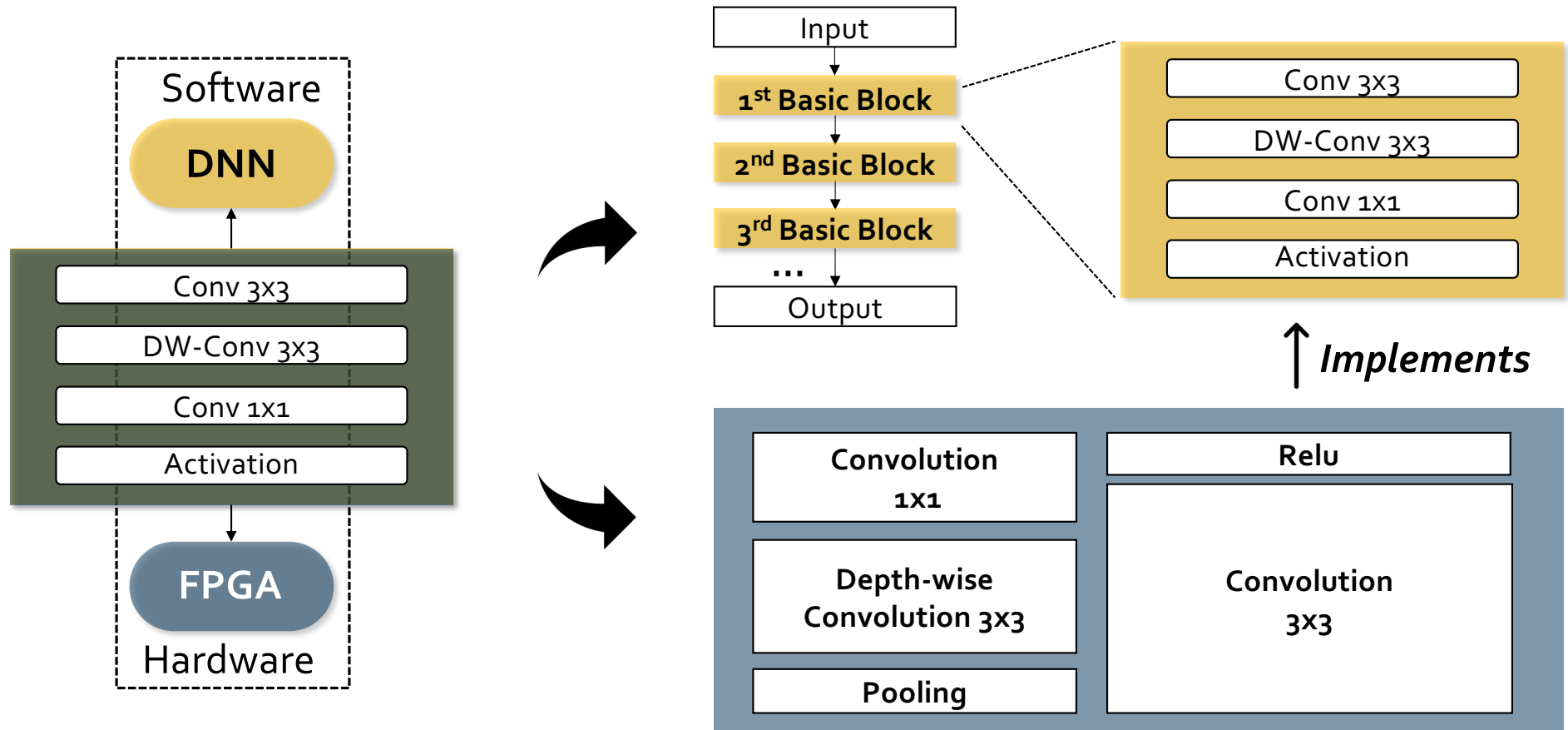
# FPGA/DNN Co-Design

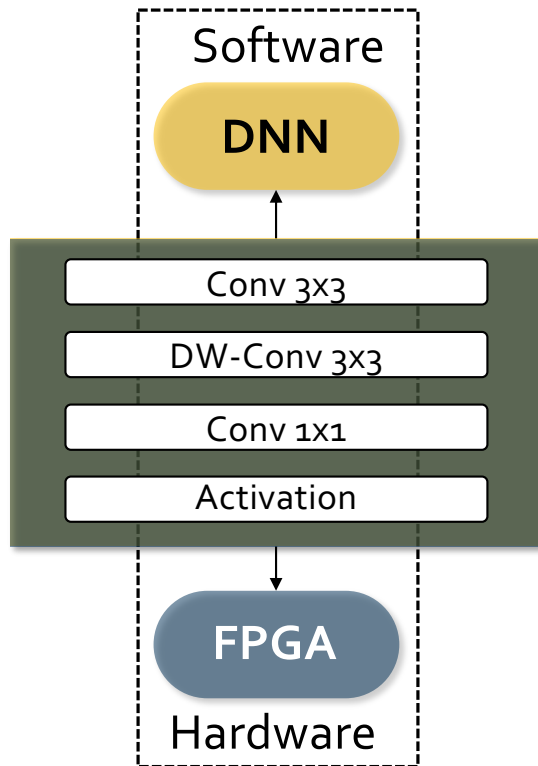


# FPGA/DNN Co-Design



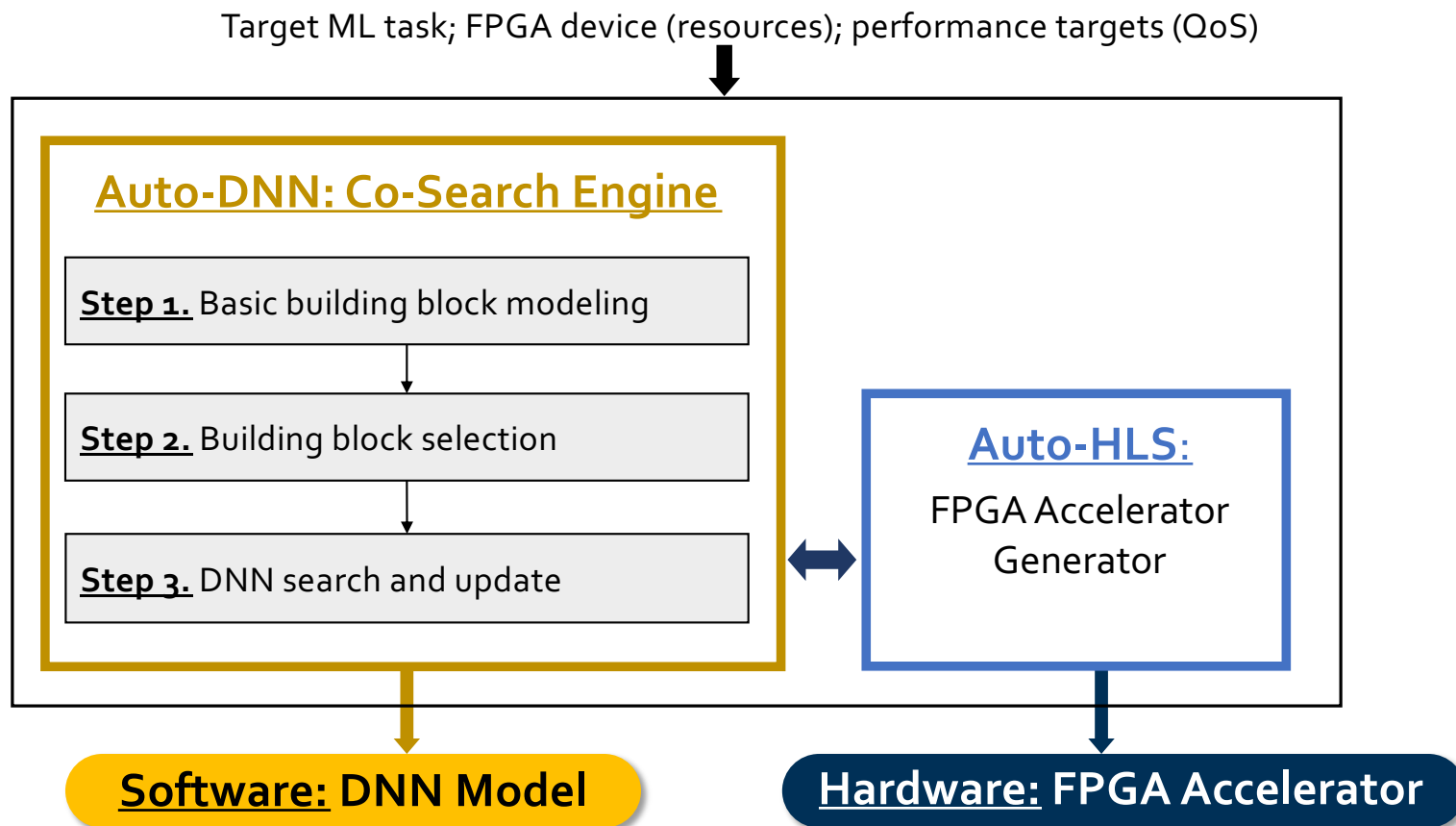
# FPGA/DNN Co-Design



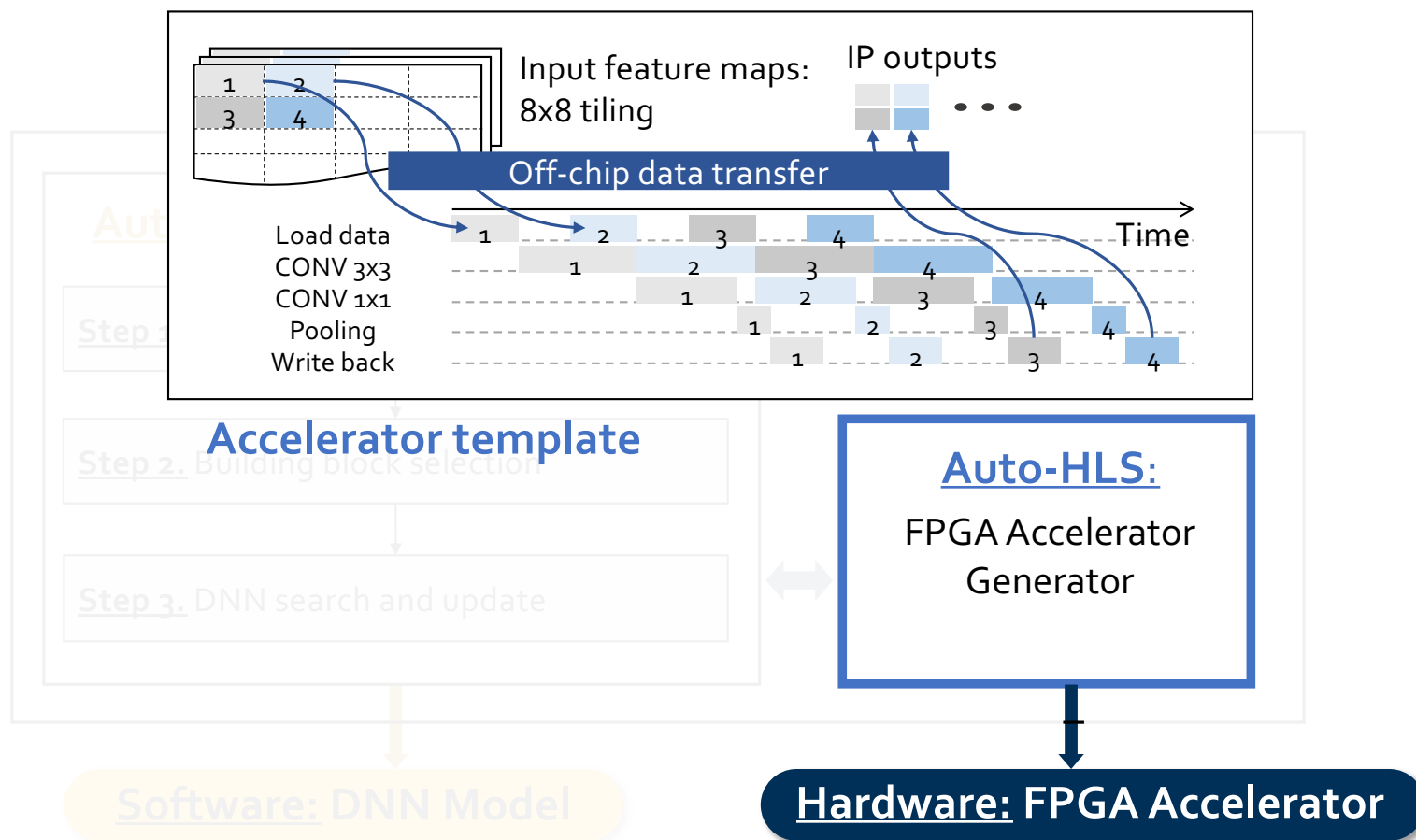


How to choose these basic building blocks from a large design space?

# DNN/FPGA Co-Design Flow

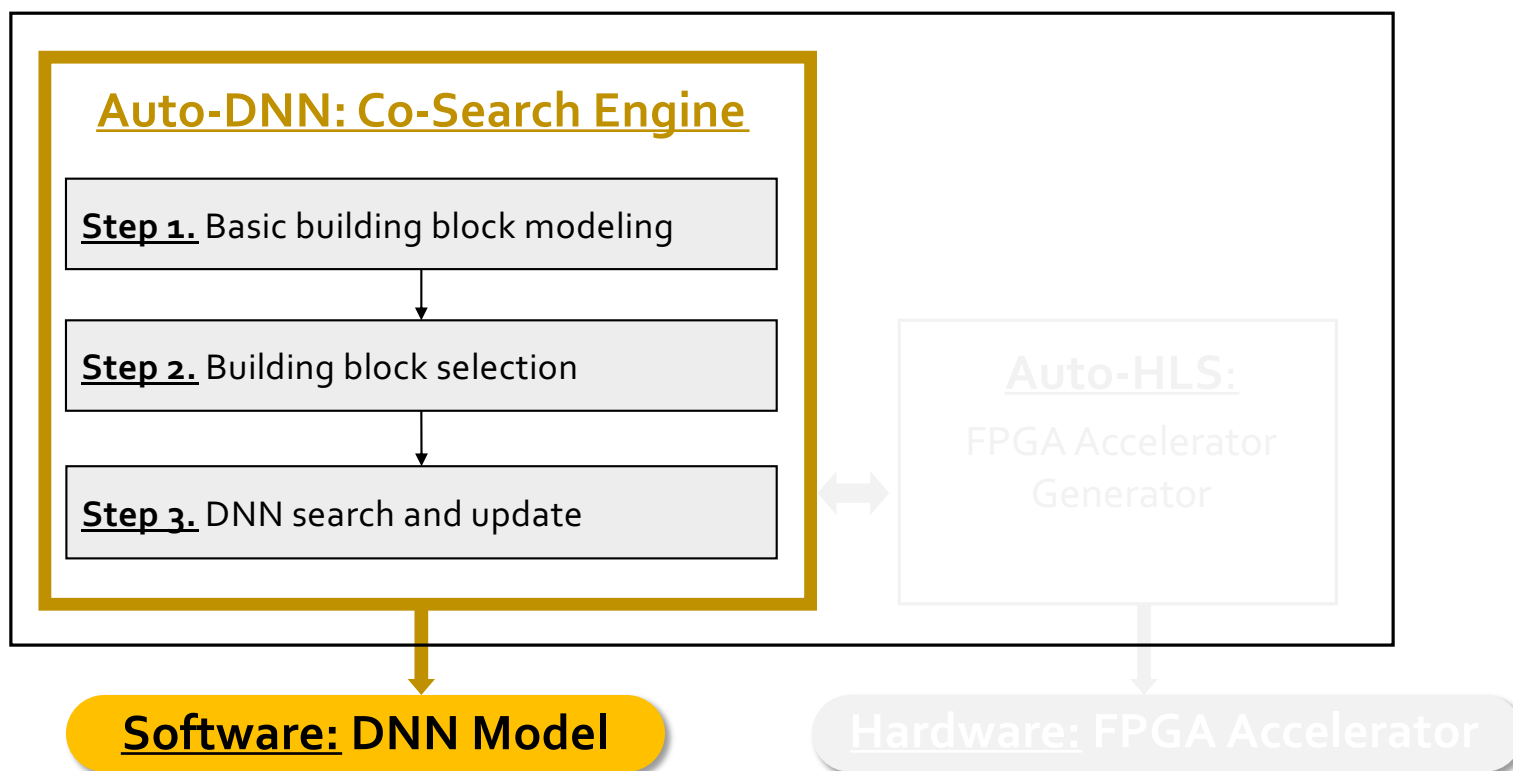


# DNN/FPGA Co-Design Flow

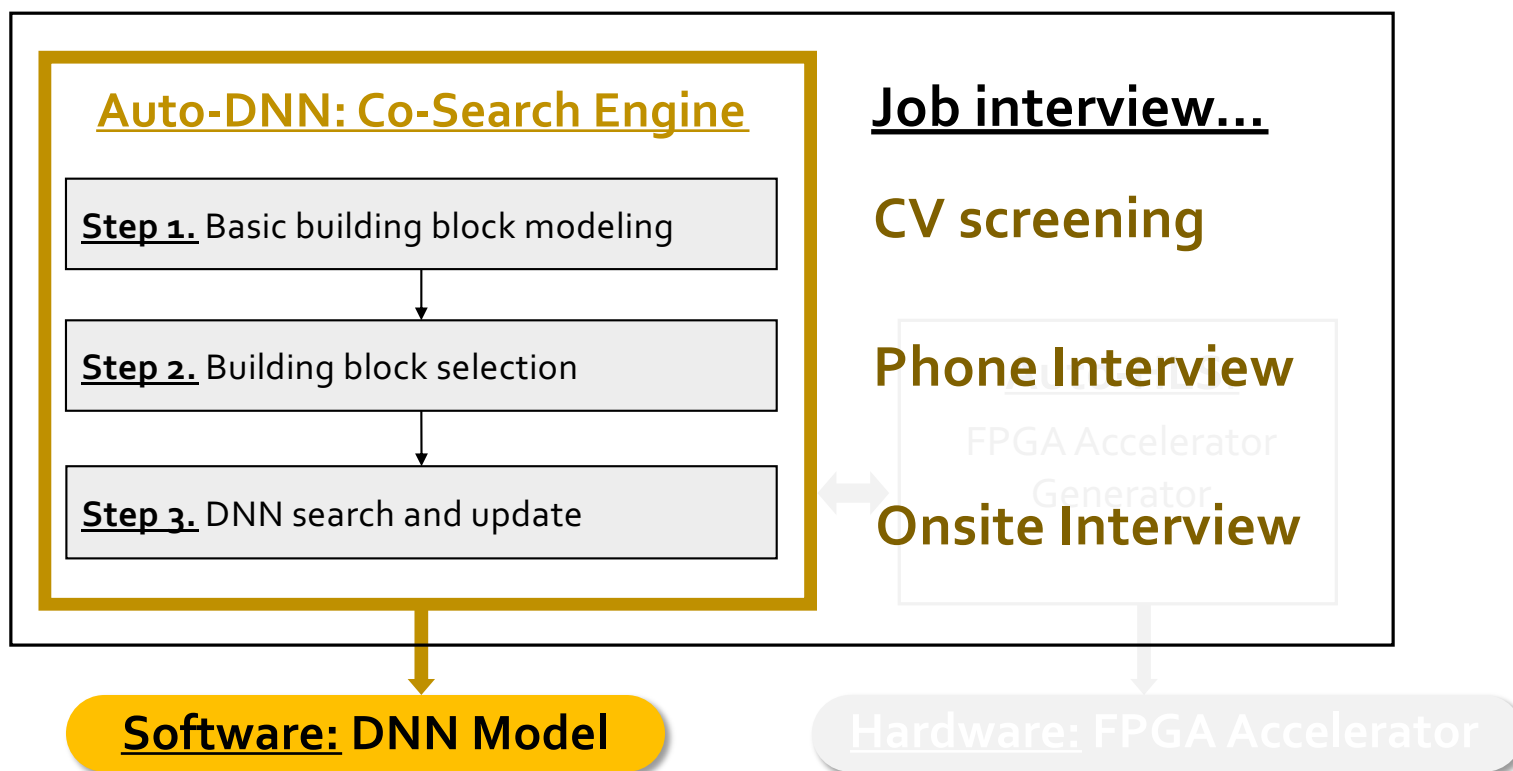




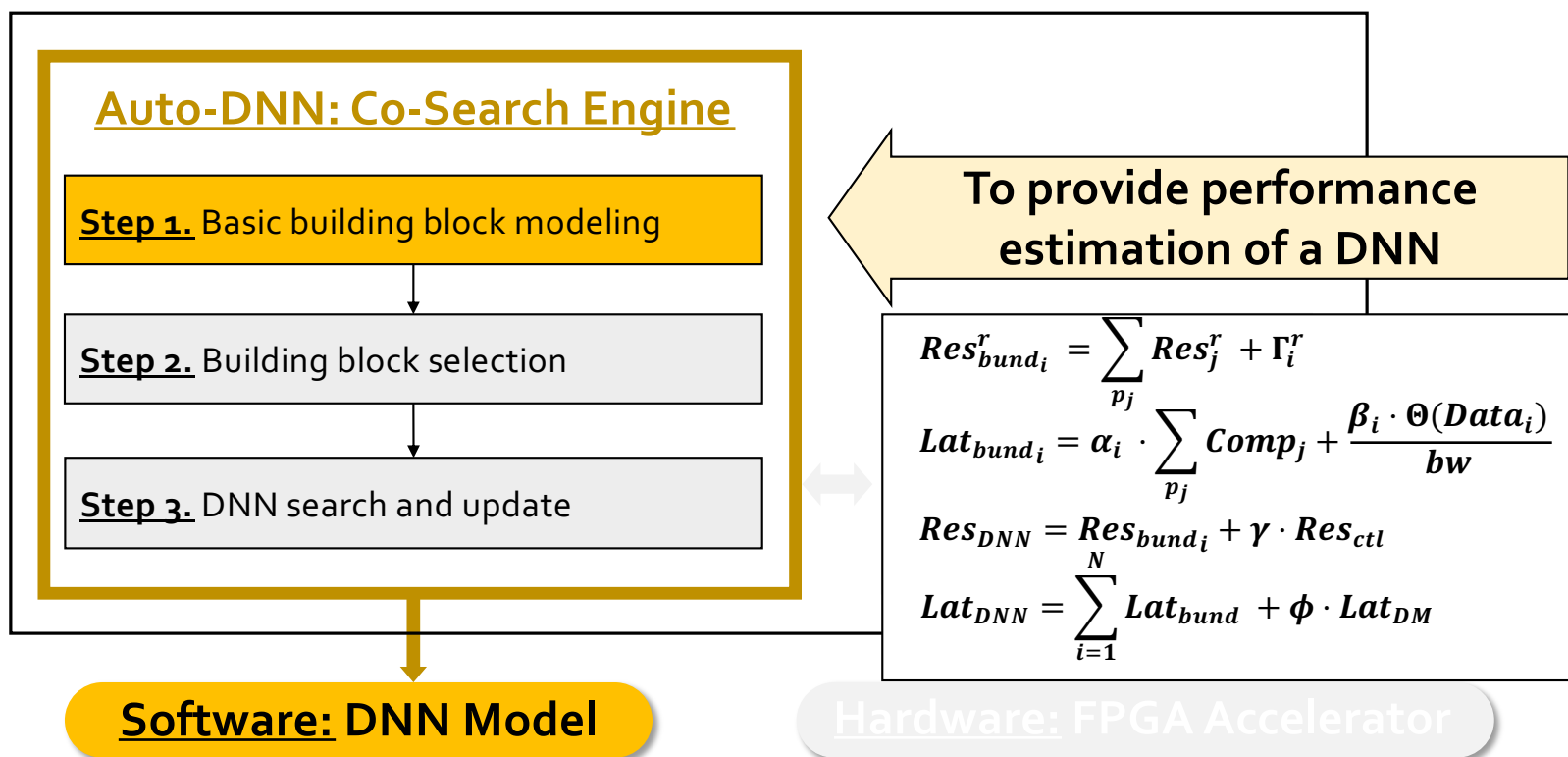
# DNN/FPGA Co-Design Flow



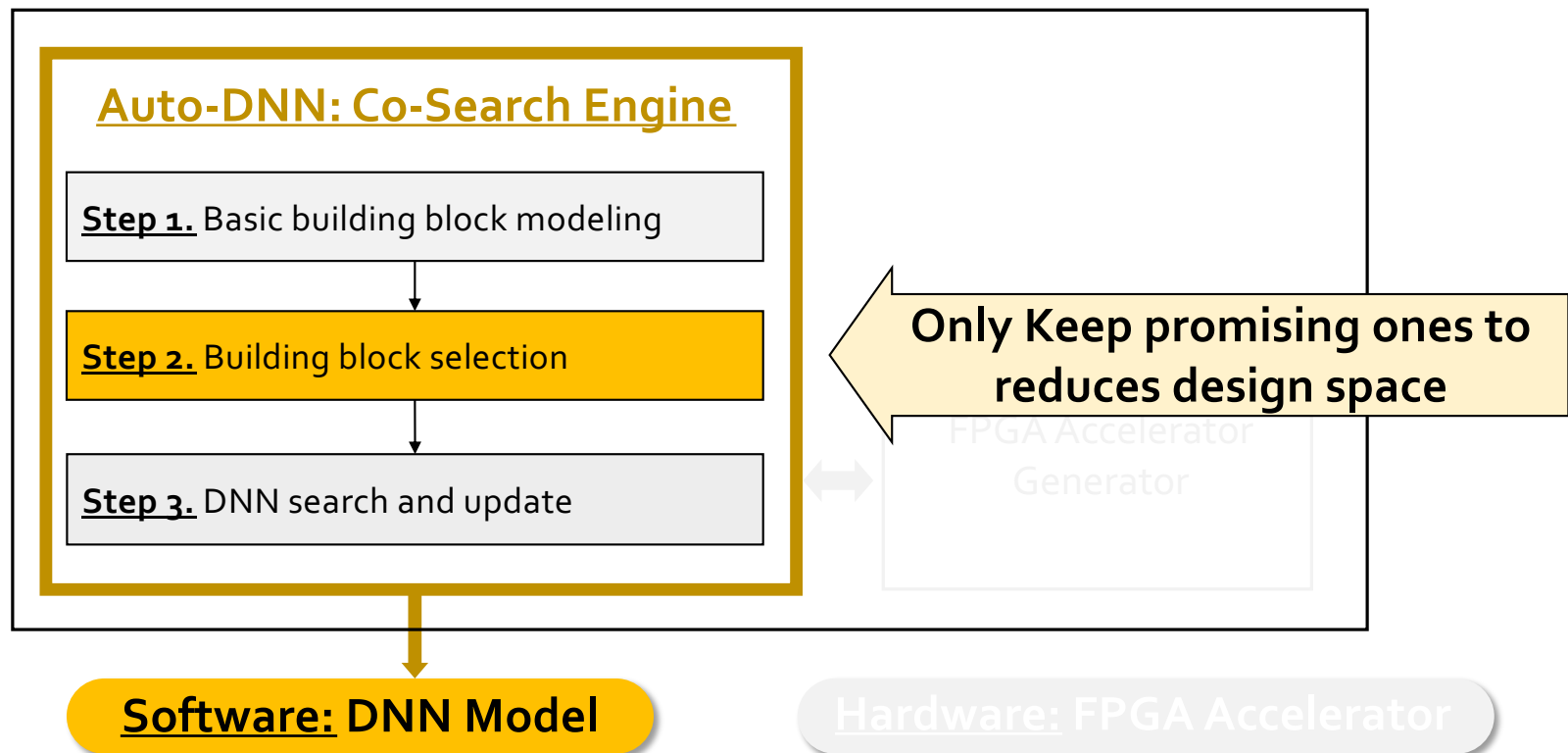
# DNN/FPGA Co-Design Flow



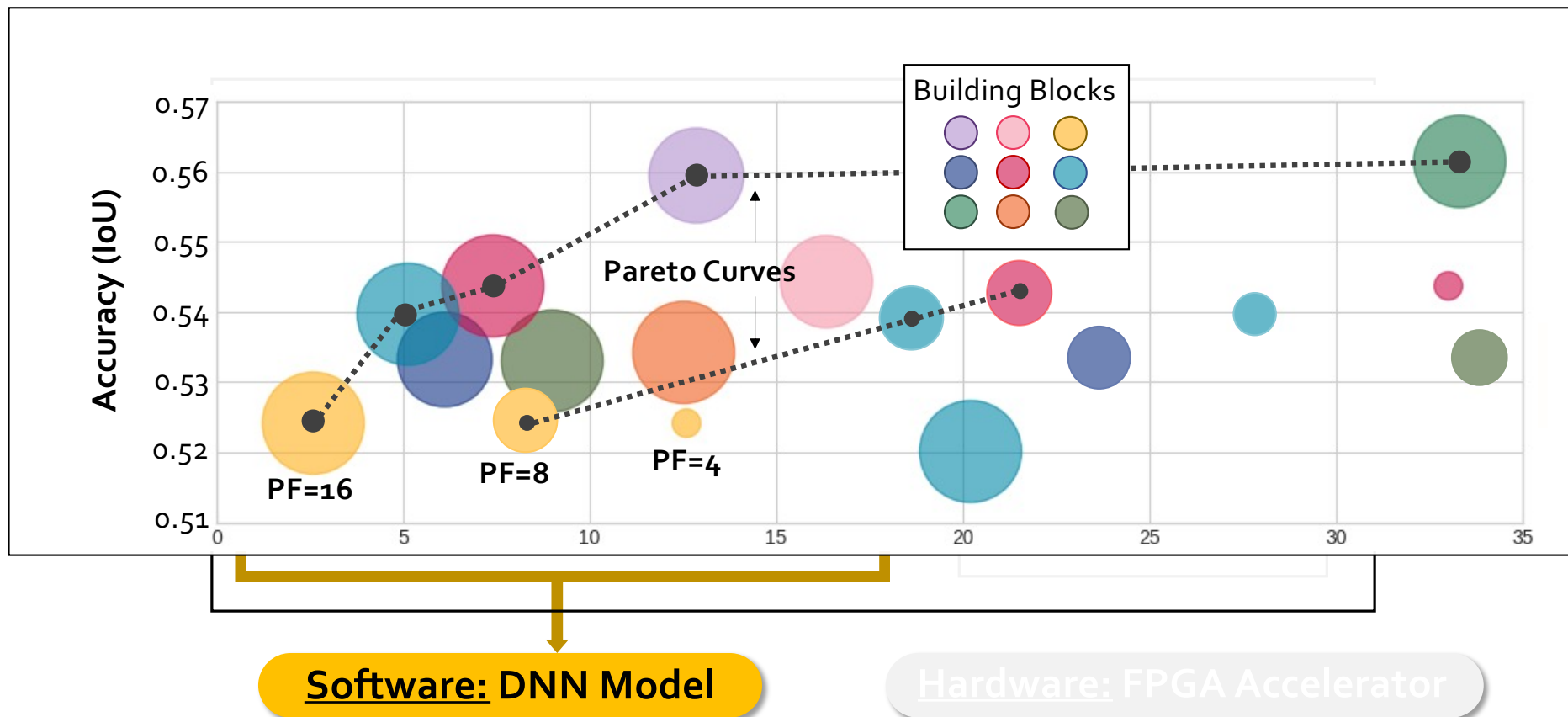
# DNN/FPGA Co-Design Flow



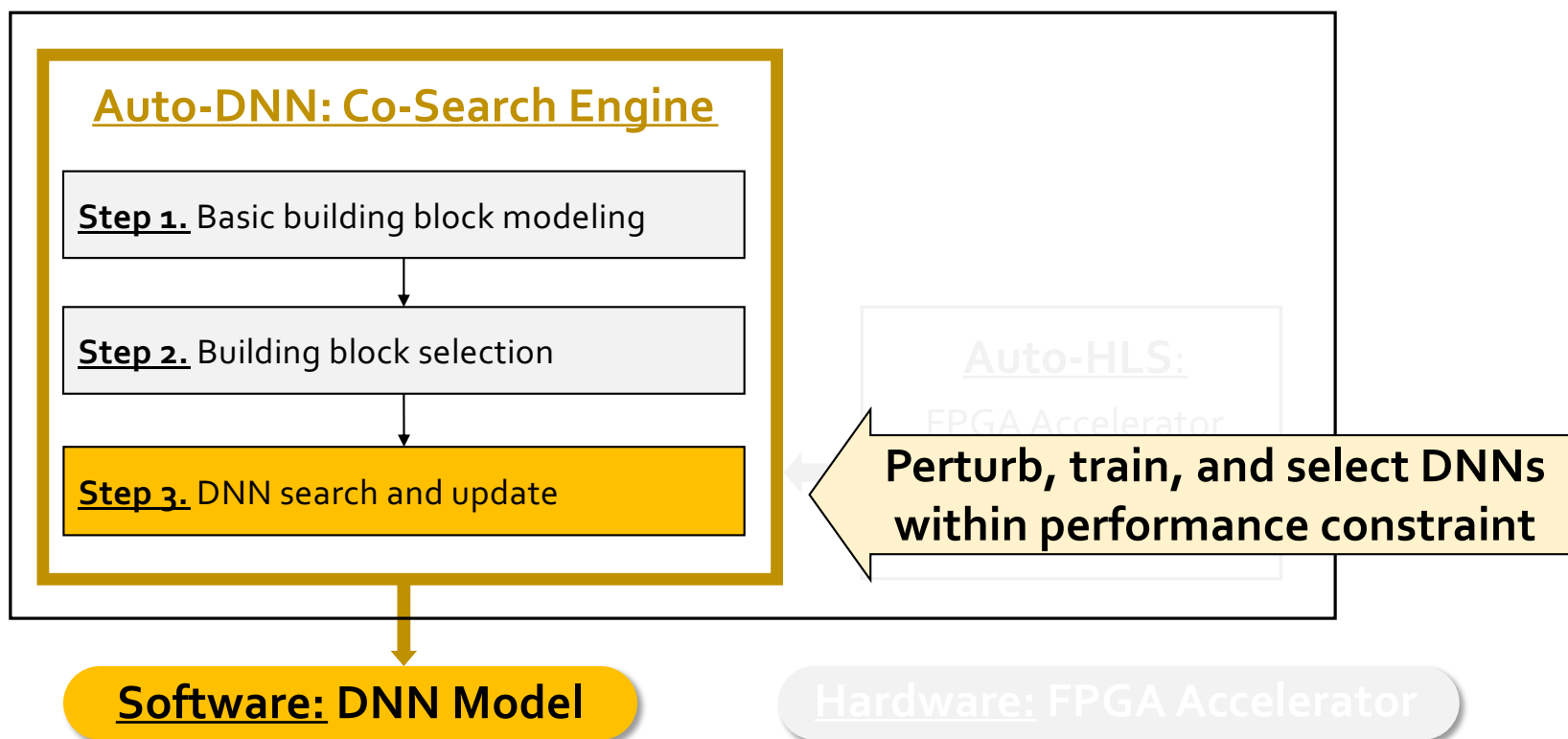
# DNN/FPGA Co-Design Flow



# DNN/FPGA Co-Design Flow



# DNN/FPGA Co-Design Flow



# NAIS Victory – DAC System Design Contest



- Design Automation Conference System Design Contest (DAC-SDC)
  - Object detection on FPGA/GPU
- Our Achievements
  - 2018: **Third place** @ FPGA (3 out of 51)

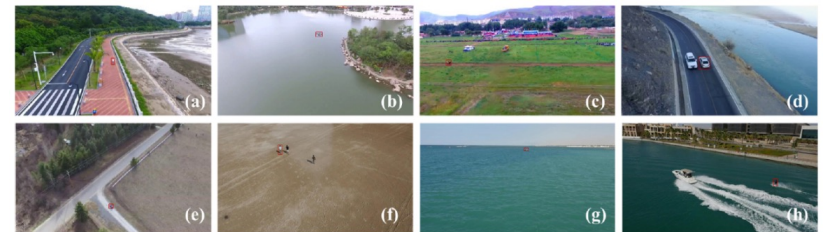


# NAIS Victory – DAC System Design Contest

- Design Automation Conference System Design Contest (DAC-SDC)
  - Object detection on FPGA/GPU
- Our Achievements
  - 2018: **Third place** @ FPGA (3 out of 51)



Independently designed DNN and FPGA accelerator – a lot of iterations!



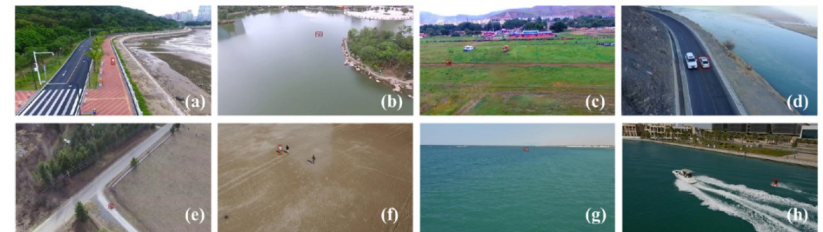


# NAIS Victory – DAC System Design Contest

- Design Automation Conference System Design Contest (DAC-SDC)
  - Object detection on FPGA/GPU
- Our Achievements
  - 2018: **Third place** @ FPGA (3 out of 51)
  - 2019: **Double championship** @ FPGA and GPU (1 out of 58, 1 out of 56)



Independently designed DNN and FPGA accelerator – a lot of iterations!



# NAIS Victory – DAC System Design Contest

- Design Automation Conference System Design Contest (DAC-SDC)
  - Object detection on FPGA/GPU
- Our Achievements
  - **2018: Third place** @ FPGA (3 out of 51)
  - **2019: Double championship** @ FPGA and GPU (1 out of 58, 1 out of 56)



Independently designed DNN and FPGA accelerator – a lot of iterations!



NAIS co-design leads to victory!



# NAIS Victory – DAC System Design Contest

- Media coverage and open-source code



<https://www.ibm.com/blogs/research/2019/06/winning-ai-algorithms-drones/>



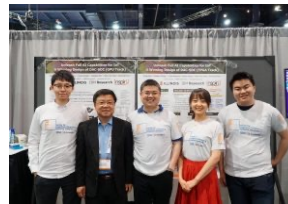
2018: <https://github.com/onioncc/iSmartDNN>



2019: <https://github.com/TomGoo8/SkyNet>



2020: [https://github.com/jgoeders/dac\\_sdc\\_2020\\_designs](https://github.com/jgoeders/dac_sdc_2020_designs)



# Key Methodologies for Co-Design

Software:

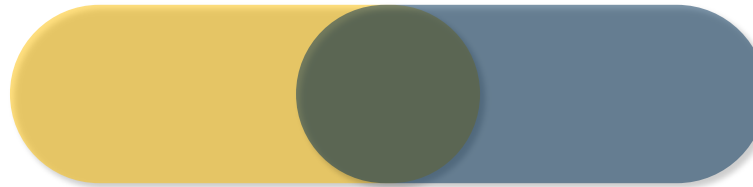
Neural Architecture  
Search Space

Hardware:

Implementation Search  
Space

1

Method 1: find something  
in the middle and connect  
to both SW and HW



FPGA/DNN Co-Design  
[DAC'19, SysML'20]



# Key Methodologies for Co-Design

## Software:

Neural Architecture  
Search Space

## Hardware:

Implementation Search  
Space

Li, Yuhong, Cong Hao, Xiaofan Zhang, Xinheng Liu, Yao Chen, Jinjun Xiong, Wen-mei Hwu, and Deming Chen. "**EDD: Efficient differentiable DNN architecture and implementation co-search for embedded AI solutions.**" DAC 2020

**2** Method 2: merge the  
two spaces – **formulate  
both in one equation**

A True NAIS work  
EDD [ICCAD'19, DAC'20]



# Key Methodologies for Co-Design

Software:

Neural Architecture  
Search Space

$\{A\}$

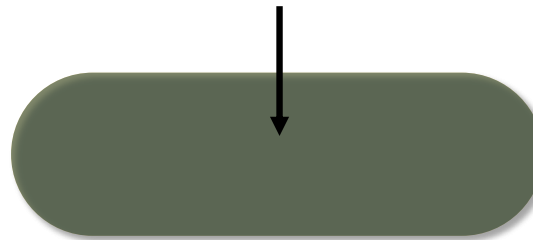
Hardware:

Implementation Search  
Space

$\{I\}$

- Put  $\{A, I\}$  into one formulation, preferably **differentiable**
- Solve  $\{A, I\}$  using continuous optimization, e.g., **Gradient Descent**

**2** Method 2: merge the  
two spaces – **formulate  
both in one equation**



A True NAIS work  
EDD [ICCAD'19, DAC'20]

# NAIS Formulation

## Existing Formulation

*Hardware-ware NAS*

$$\min: \mathcal{L} = \underbrace{Acc_{loss}(A)}_{A \text{ is differentiable with respect to } \mathcal{L}} \cdot \underbrace{Perf_{loss}(I_0)}_{\text{Implementation } I_0 \text{ is fixed (not in the search space)}}$$

$A$  is **differentiable**  
with respect to  $\mathcal{L}$

Implementation  $I_0$  is **fixed**  
(**not** in the search space)

## NAIS Formulation

*True co-design*

$$\min: \mathcal{L} = \underbrace{Acc_{loss}(A, I)}_{A \text{ is differentiable with respect to } \mathcal{L}} \cdot \underbrace{Perf_{loss}(I)}_{\text{Implementation } I \text{ is also variable}} + \beta \cdot \underbrace{C^{RES(I) - RES_{ub}}}_{\text{Consider resource constraints}}$$

$A$  is **differentiable**  
with respect to  $\mathcal{L}$

Implementation  $I$   
is also **variable**

Consider resource  
constraints

# NAIS Formulation

## Existing Formulation

Hardware-ware NAS

$$\min: \mathcal{L} = \underbrace{Acc_{loss}(A)}_{A \text{ is differentiable with respect to } \mathcal{L}} \cdot \underbrace{Perf_{loss}(I_0)}_{\text{Implementation } I_0 \text{ is fixed (not in the search space)}}$$

$A$  is **differentiable**  
with respect to  $\mathcal{L}$

Implementation  $I_0$  is **fixed**  
(**not** in the search space)

## NAIS Formulation

True co-design

$$\min: \mathcal{L} = \underbrace{Acc_{loss}(A, I)}_{A \text{ is differentiable with respect to } \mathcal{L}} \cdot \underbrace{Perf_{loss}(I)}_{\text{Implementation } I \text{ is also variable}} + \beta \cdot \underbrace{C^{RES(I) - RES_{ub}}}_{\text{Consider resource constraints}}$$

$A$  is **differentiable**  
with respect to  $\mathcal{L}$

Implementation  $I$   
is also **variable**

Consider resource  
constraints

## Challenge

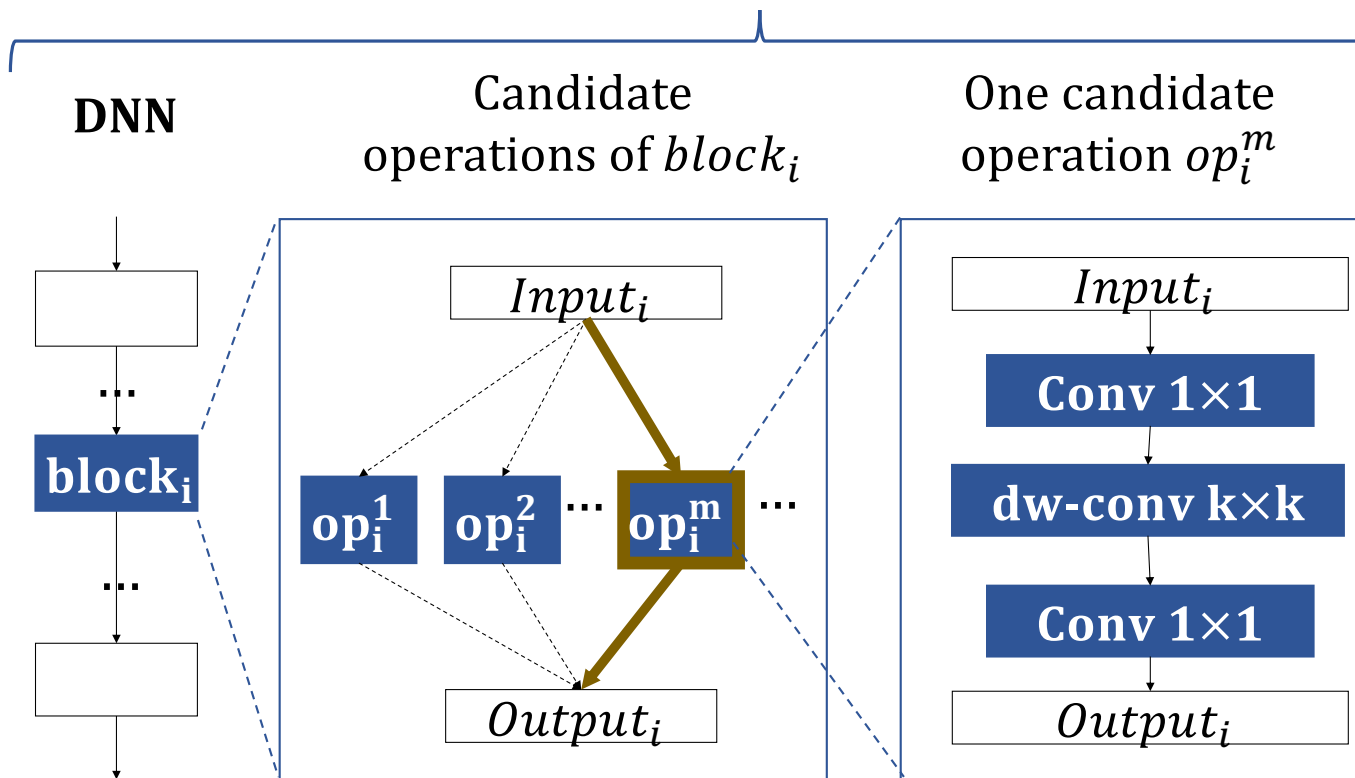
How to formulate  $I$  as **differentiable** with respect to  $\mathcal{L}$ ?





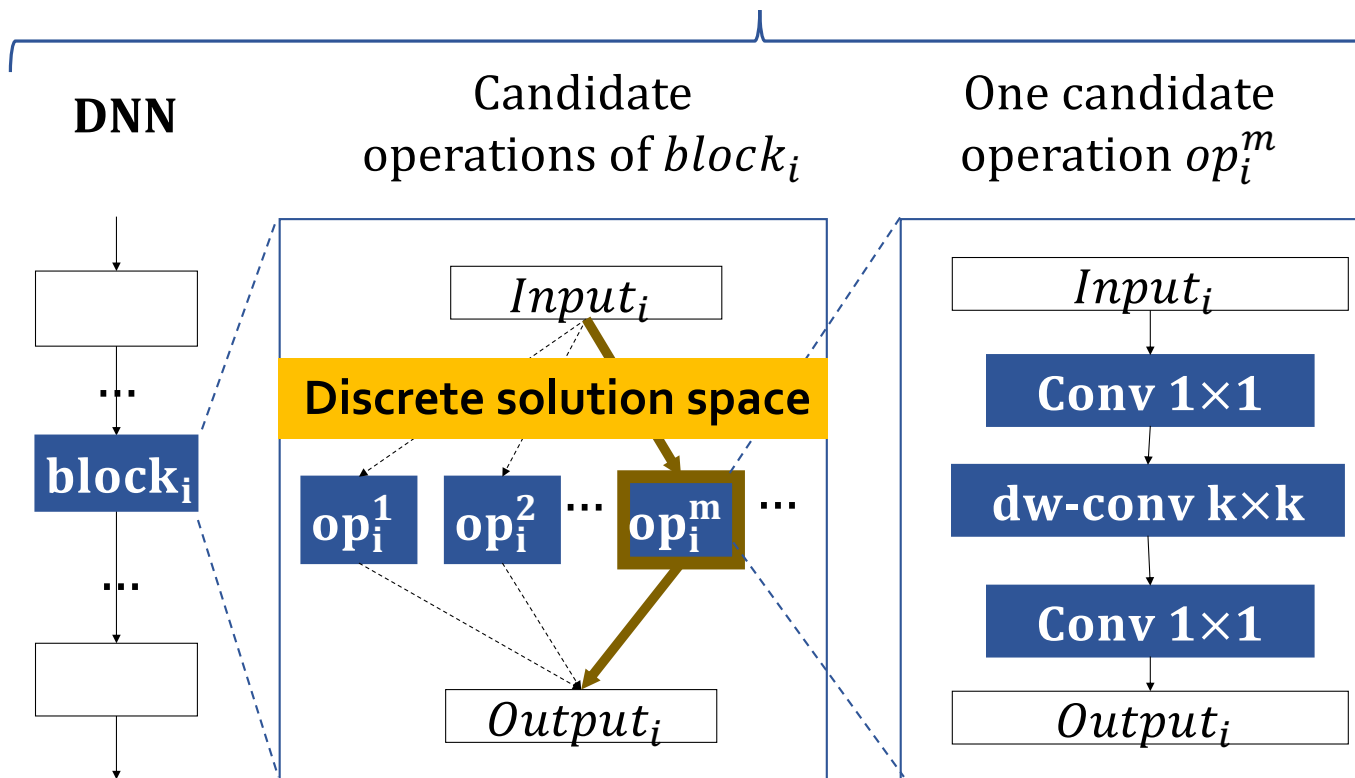
# Differentiable DNN Architecture Search

## Neural Architecture Search (NAS)



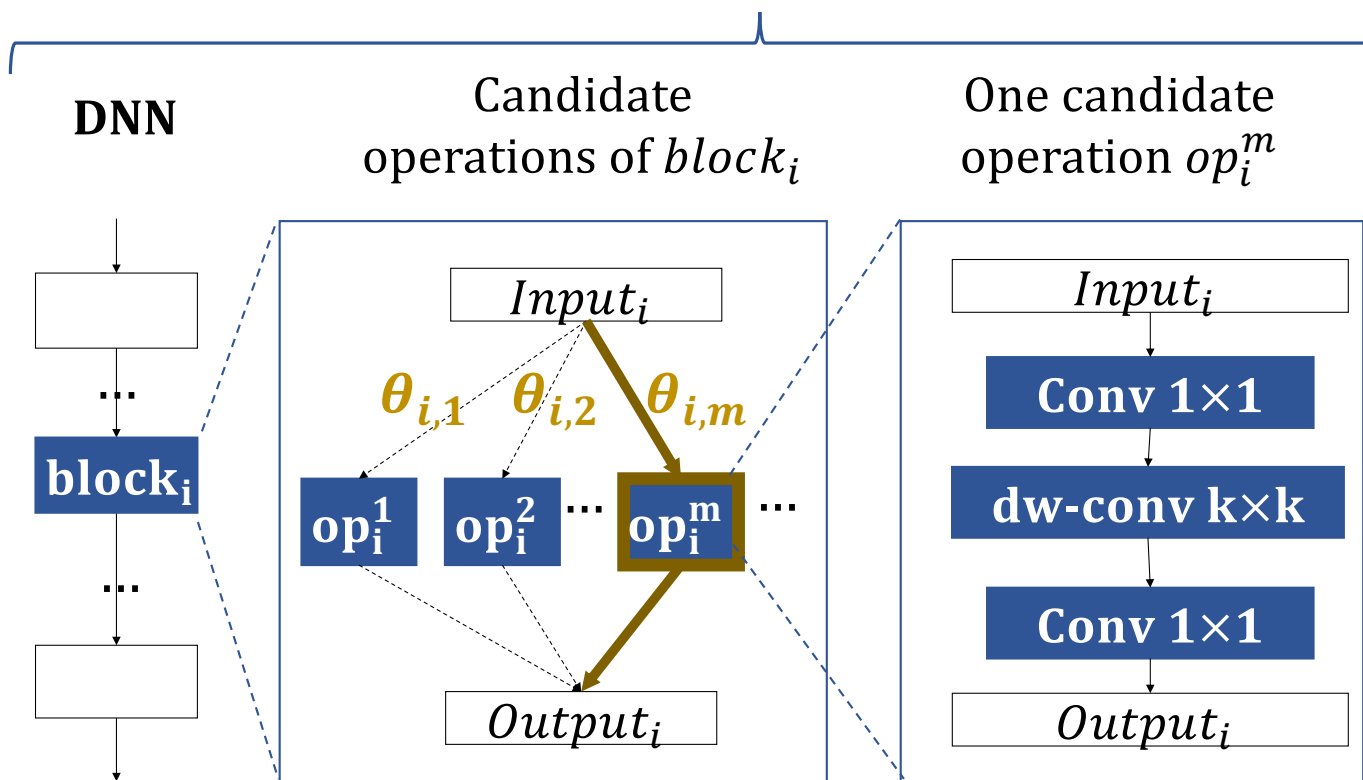
# Differentiable DNN Architecture Search

## Neural Architecture Search (NAS)



# Differentiable DNN Architecture Search

## Neural Architecture Search (NAS)



From discrete to continuous  
for differentiable:

**Gumbel-Softmax**

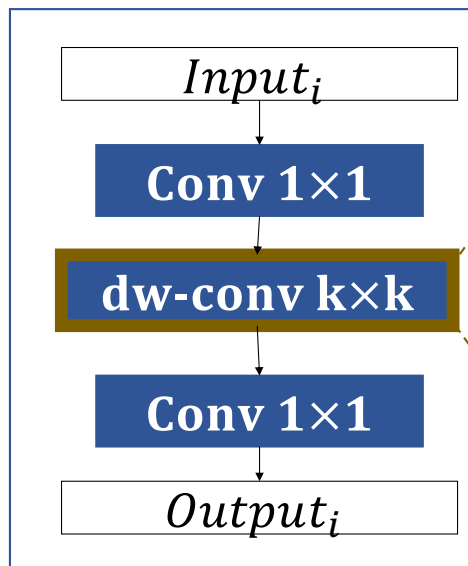
- Sampling parameter  $\theta_{i,m}$
- Operations sampled following *Gumbel-Softmax* distribution
- $\theta_{i,m}$  is differentiable with respect to  $\mathcal{L}$



# Differentiable Implementation Search

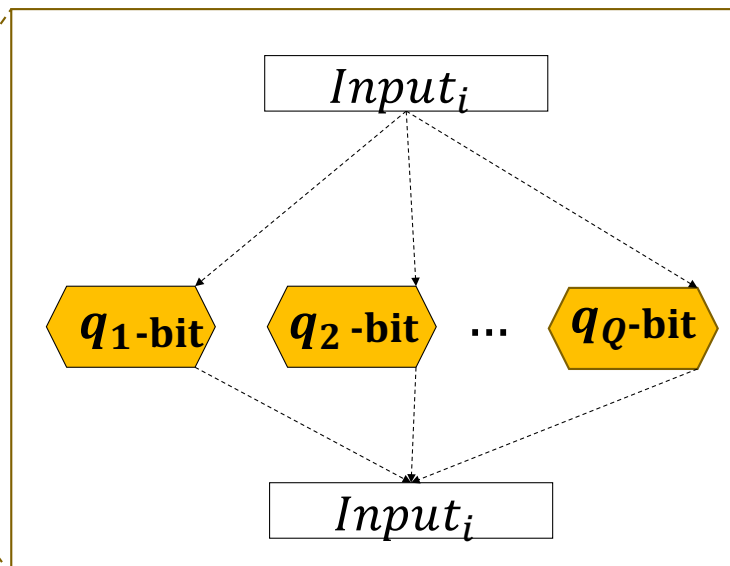
## NAS

One candidate operation  $op_i^m$



## Implementation Search

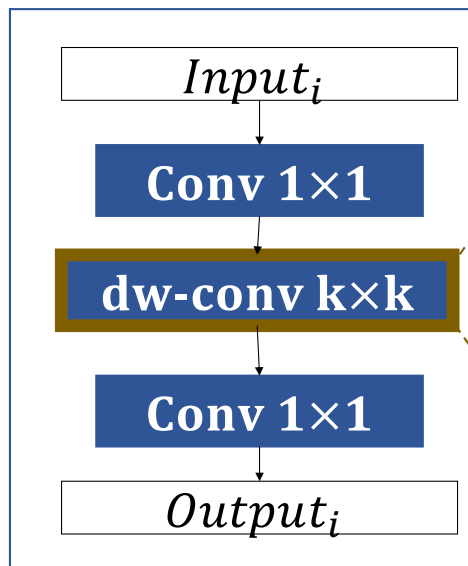
Quantization of  $op_i^m$



# Differentiable Implementation Search

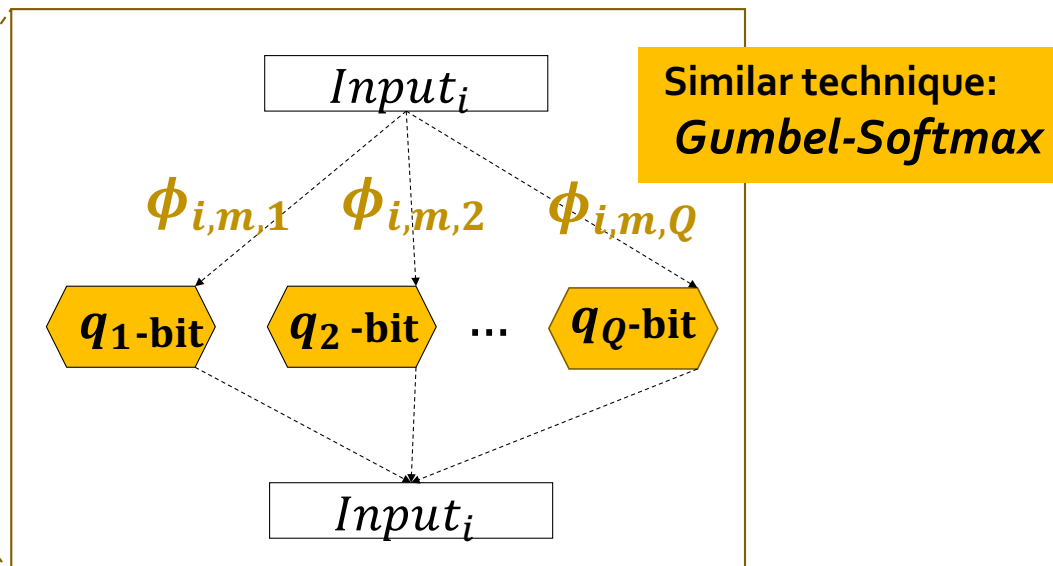
## NAS

One candidate operation  $op_i^m$



## Implementation Search

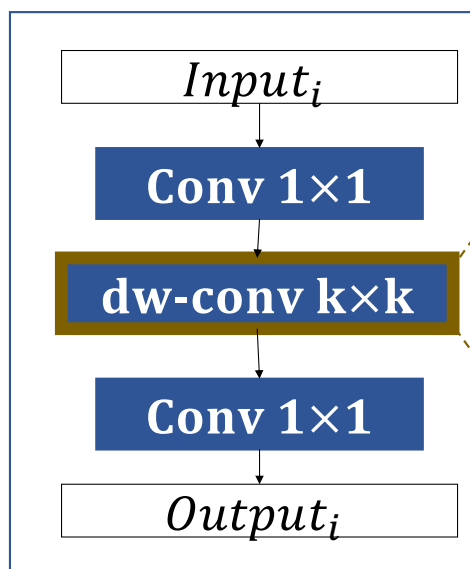
Quantization of  $op_i^m$



# Differentiable Implementation Search

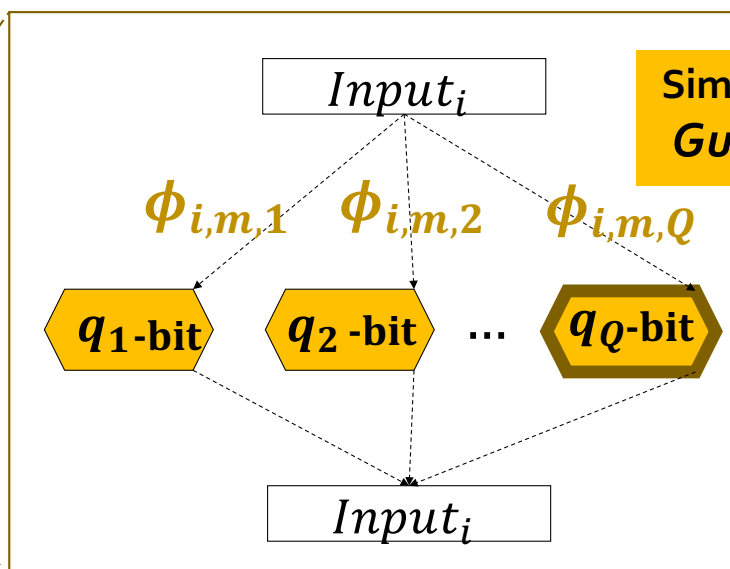
## NAS

One candidate operation  $op_i^m$



## Implementation Search

Quantization of  $op_i^m$



Similar technique:  
**Gumbel-Softmax**

$I_i^m$  : other implementation variables

- $Perf^q(op_i^m) = f(I_i^m)$
- $Res^q(op_i^m) = g(I_i^m)$



# Now Since Everything is Differentiable...

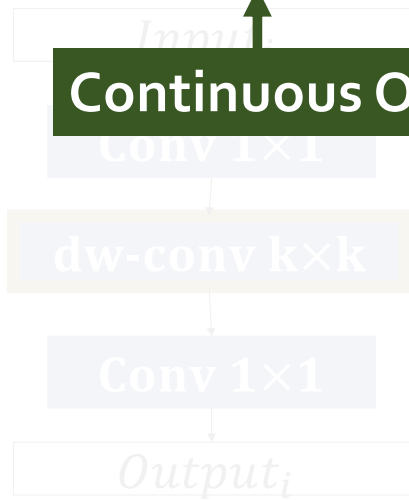
## NAS



*Differentiable*

$$\min: \mathcal{L} = \text{Acc}_{\text{loss}}(\mathbf{A}, \mathbf{I}) \cdot \text{Perf}_{\text{loss}}(\mathbf{I}) + \beta \cdot \mathcal{C}^{\text{RES}(\mathbf{I}) - \text{RES}_{ub}}$$

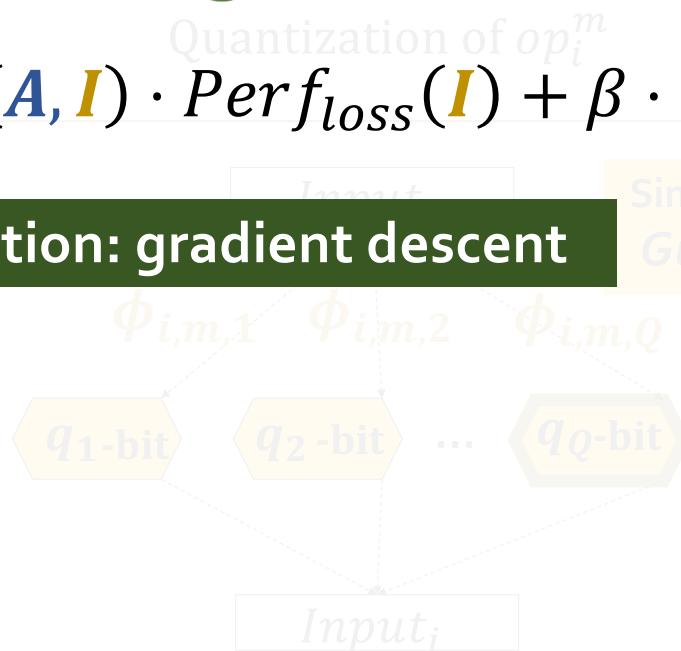
Continuous Optimization: gradient descent



## Implementation Search



*Differentiable*



Similar technique:  
*Gumbel-Softmax*

$I_i^m$  : other implementation variables

- $\text{Perf}^q(\text{op}_i^m) = f(I_i^m)$
- $\text{Res}^q(\text{op}_i^m) = g(I_i^m)$



# Comparisons with hardware-aware NAS

	Test Error (%)		GPU Latency	FPGA Latency
	Top-1	Top-5	Titan RTX	ZCU102 [22]
<b>Baseline Models</b>				
GoogleNet	30.22	10.47	27.75 ms	13.25 ms
MobileNet-V2	28.1	9.7	17.87 ms	10.85 ms
ShuffleNet-V2	30.6	11.7	21.91 ms	NA
ResNet18	30.2	10.9	9.71 ms	10.15ms
<b>Hardware-aware NAS Models</b>				
MNasNet-A1	24.8	7.5	17.94 ms	8.78 ms
FBNet-C	24.9	7.6	22.54 ms	12.21 ms
Proxyless-cpu	24.7	7.6	21.34 ms	10.81 ms
Proxyless-Mobile	25.4	7.8	21.23 ms	10.78 ms
Proxyless-gpu	24.9	7.5	15.72 ms	10.79 ms
<b>EDD-Net-1</b>	25.3	7.7	<b>11.17 ms</b>	11.15 ms
<b>EDD-Net-2</b>	25.4	7.9	13.00 ms	<b>7.96 ms</b>

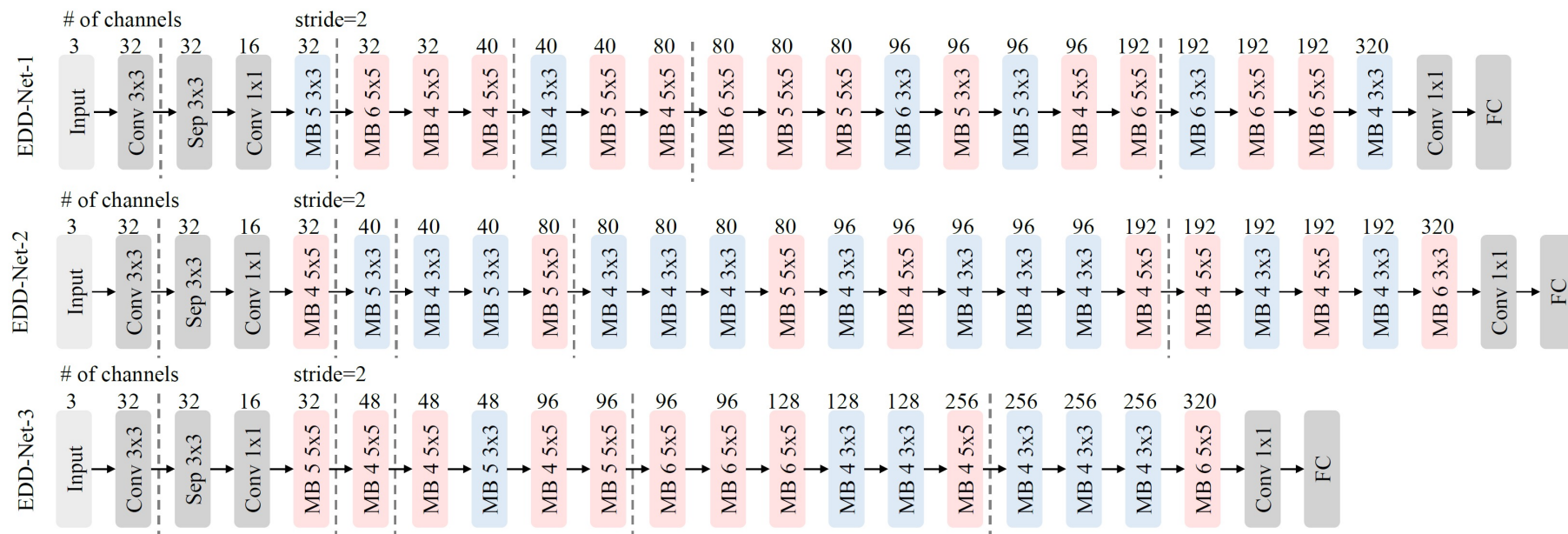
GPU-oriented DNN

FPGA-oriented DNN





# Comparisons with hardware-aware NAS



EDD-Net-1: targets GPU

EDD-Net-2: targets recursive FPGA accelerator

EDD-Net-3: targets pipelined FPGA accelerator

# Follow-up Works using Differentiable Approach

Dna: **Differentiable** network-**accelerator** co-search

[Y Zhang](#), [Y Fu](#), [W Jiang](#), [C Li](#), [H You](#), [M Li](#)... - arXiv preprint arXiv ..., 2020 - arxiv.org

ConCoDE: Hard-constrained **Differentiable** Co-Exploration Method for Neural Architectures and Hardware **Accelerators**

D Hong, K Choi, HY Lee, [J Yu](#), Y Kim, [N Park](#), J Lee - 2021 - openreview.net

Dance: Differentiable accelerator/network co-exploration

K Choi, D Hong, [H Yoon](#), [J Yu](#), [Y Kim](#)... - 2021 58th ACM/IEEE ..., 2021 - ieeeexplore.ieee.org

DIAN: Differentiable accelerator-network co-search towards maximal dnn efficiency

[Y Zhang](#), [Y Fu](#), [W Jiang](#), [C Li](#), [H You](#)... - 2021 IEEE/ACM ..., 2021 - ieeeexplore.ieee.org

Triple-Search: **Differentiable** Joint-Search of Networks, Precision, and **Accelerators**

Y Fu, Y Zhang, [H You](#), Y Lin - 2020 - openreview.net



# What's NAIS's Next?

Software: Neural Architecture Search (NAS)



Hardware: Implementation Search



NAIS



# What's NAIS's Next?

Software: Neural Architecture Search (NAS)



Hardware: Implementation Search



NAIS



# What's NAIS's Next?

Software: Neural Architecture Search (NAS)



Hardware: Implementation Search



NAIS



Multi-modal Multi-task Models

# What's NAIS's Next?

Software: Neural Architecture Search (NAS)



Hardware: Implementation Search



NAIS



Multi-modal Multi-task Models



Heterogeneous Platform  
Mapping-aware NAIS



# Multi-modal Multi-Task Models (MMMT)

- **Multi-modal**: process and relate information from multiple **modalities**
  - Text, visual, vocal, motion, etc.



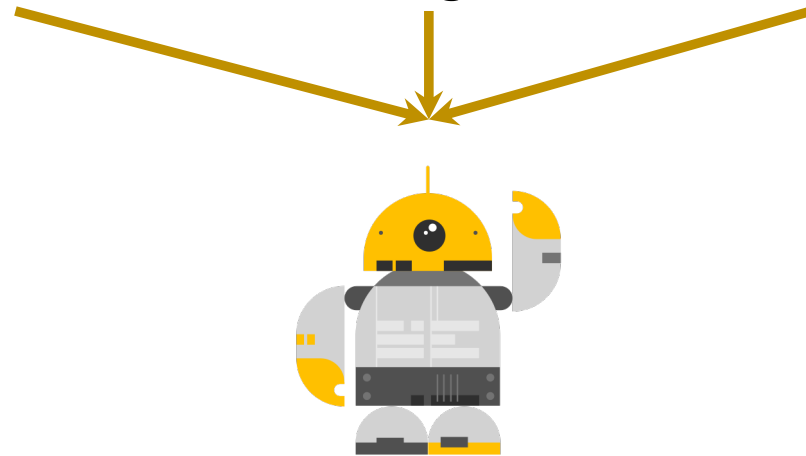
Sound



Image

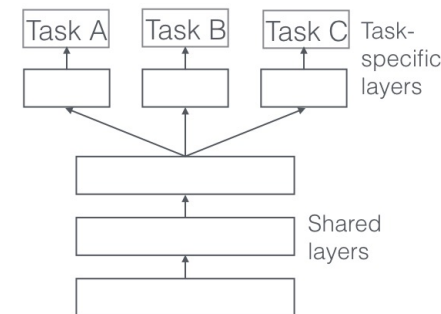
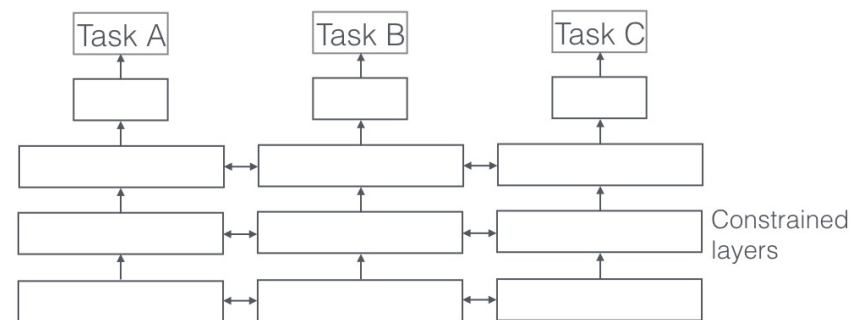


Motion



# Multi-modal Multi-Task Models (MMMT)

- **Multi-modal**: process and relate information from multiple modalities
  - Text, visual, vocal, motion, etc.
- **Multi-task**: to learn multiple related tasks jointly
  - Knowledge transfer
  - Improve the generalization performance
  - Mitigate training (labeled) data sparsity

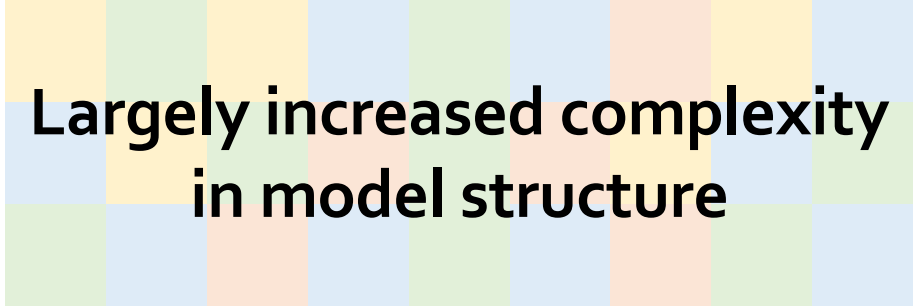


Ruder, Sebastian. "An overview of multi-task learning in deep neural networks." *arXiv preprint arXiv:1706.05098* (2017).



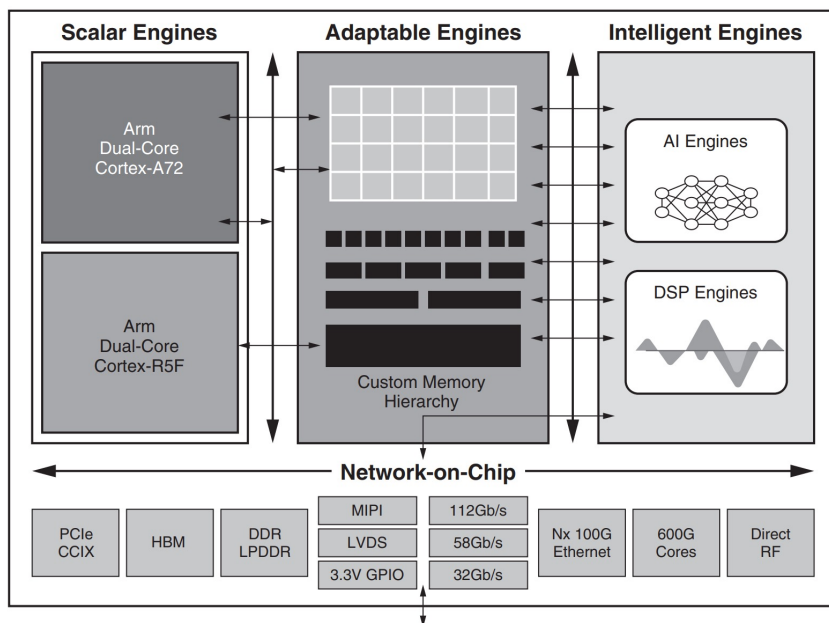
# Multi-modal Multi-Task Models (MMMT)

- **Multi-modal:** process and relate information from multiple **modalities**
  - Text, visual, vocal, motion, etc.
- **Multi-task:** to learn multiple related tasks jointly
  - Knowledge transfer
  - Improve the generalization performance
  - Mitigate training (labeled) data sparsity

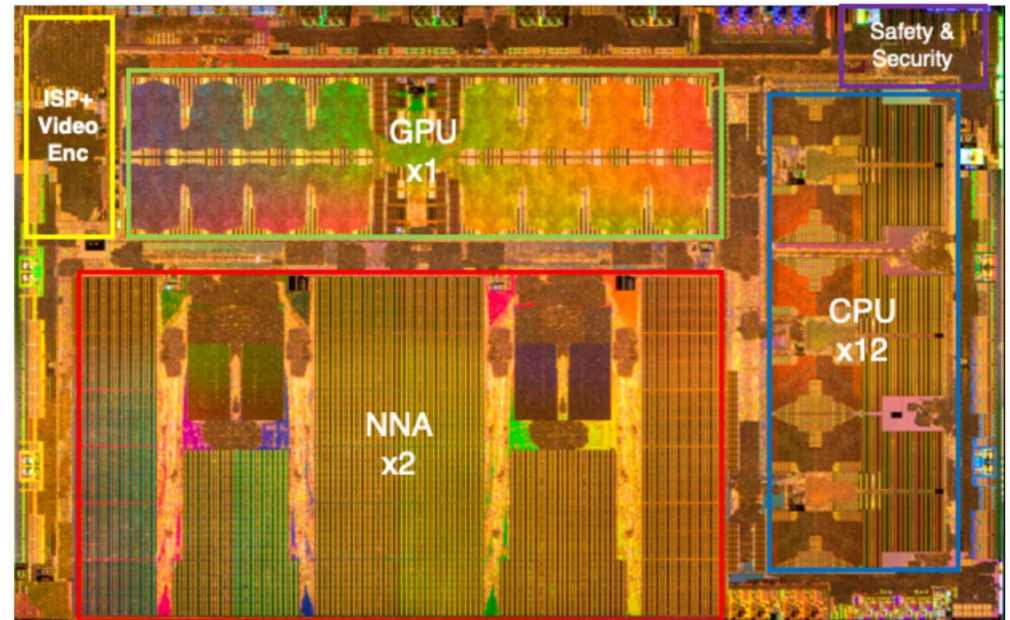


**Largely increased complexity  
in model structure**

# Heterogeneous Platforms



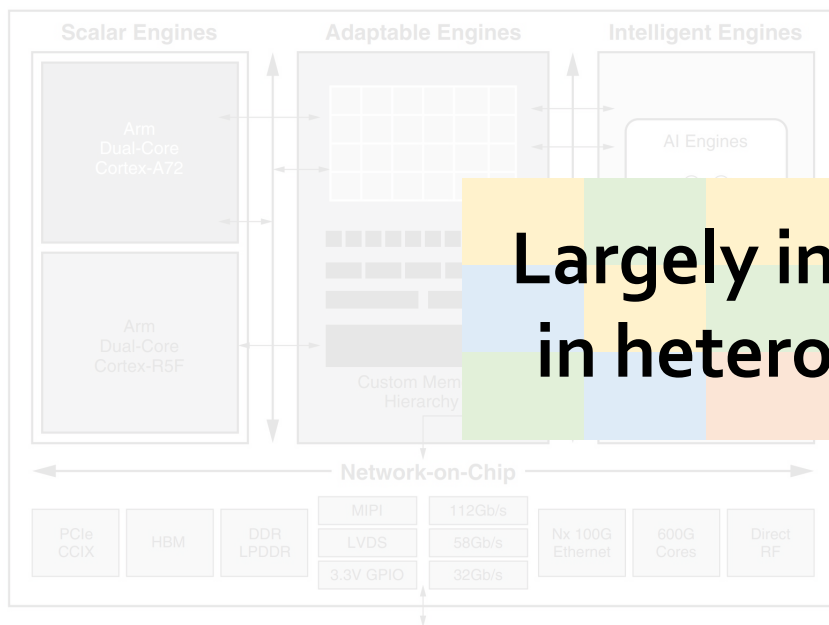
[https://www.xilinx.com/support/documentation/white\\_papers/wp505-versal-acap.pdf](https://www.xilinx.com/support/documentation/white_papers/wp505-versal-acap.pdf)



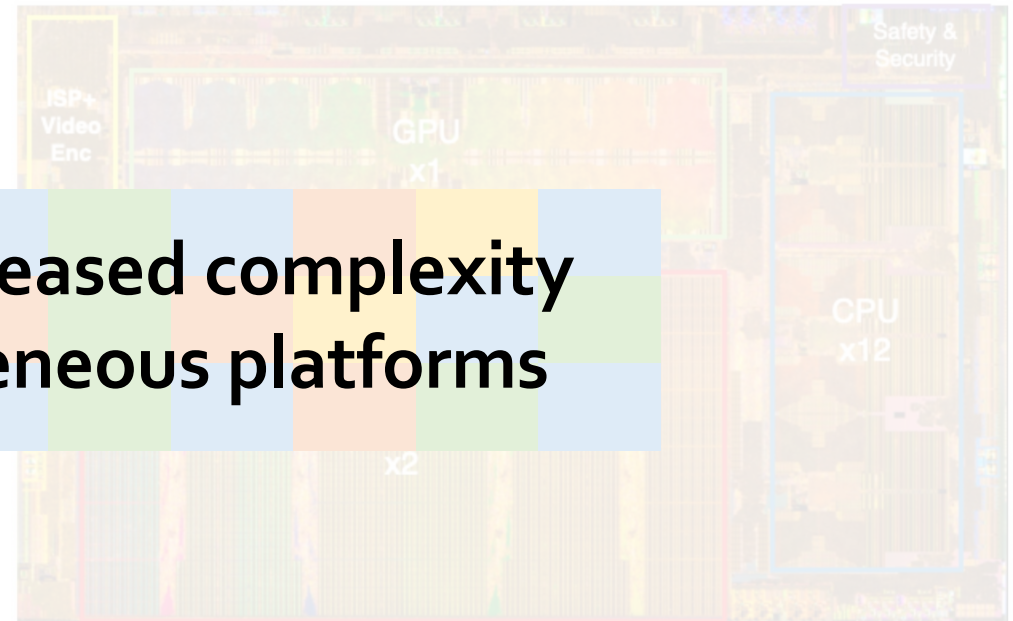
Talpes, Emil, et al. "Compute solution for tesla's full self-driving computer." *IEEE Micro* 40, no. 2 (2020): 25-35.



# Heterogeneous Platforms



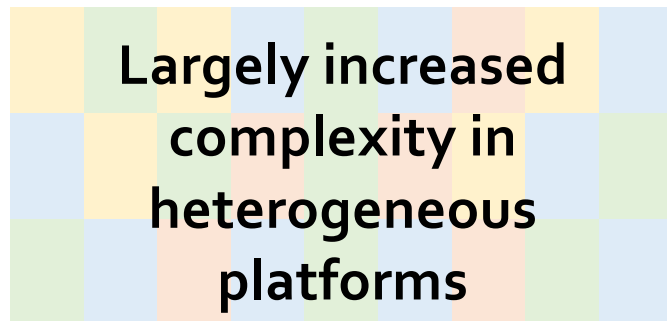
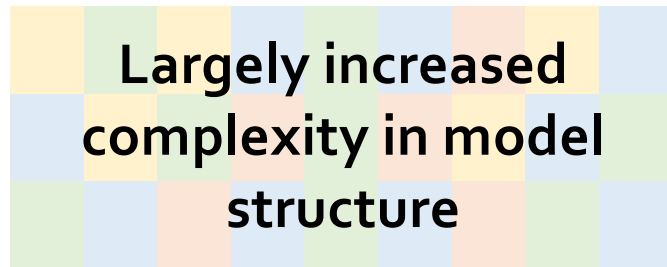
[https://www.xilinx.com/support/documentation/white\\_papers/wp505-versal-acap.pdf](https://www.xilinx.com/support/documentation/white_papers/wp505-versal-acap.pdf)



Talpes, Emil, et al. "Compute solution for tesla's full self-driving computer." *IEEE Micro* 40, no. 2 (2020): 25-35.



# When MMMT Meets Heterogeneity



**Mapping** starts to matter...

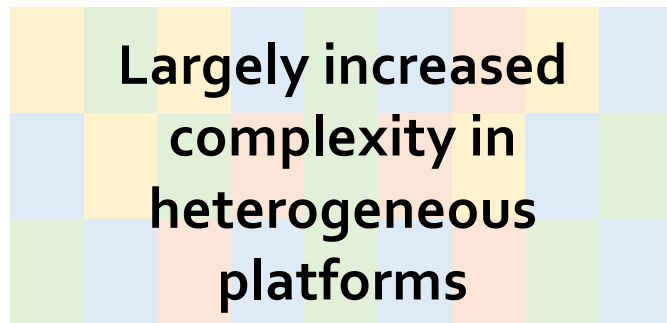
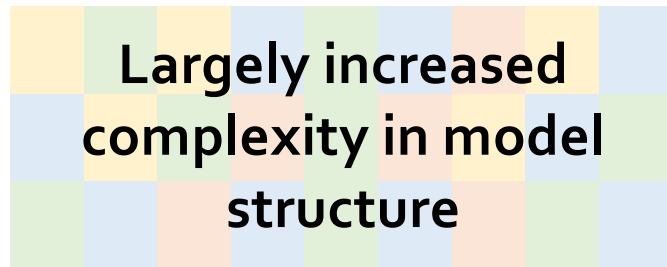


**Scheduling** starts to matter...



**Optimization** on each device  
also matters...

# When MMMT Meets Heterogeneity



**Mapping** starts to matter...



**Scheduling** starts to matter...



**Optimization** on each device  
also matters...

**NAIS**



Mapping Formulation

Scheduling Formulation

...

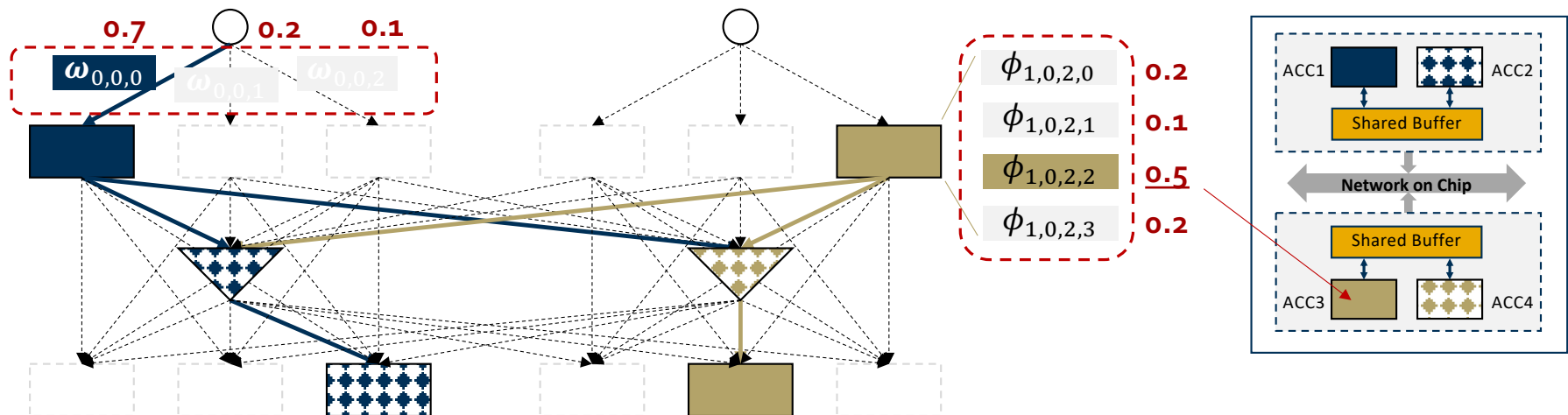


# An Example of NAIS + Mapping Formulation

NAIS

+

Mapping Formulation



Hao, Cong, and Deming Chen. "Software/Hardware Co-design for Multi-modal Multi-task Learning in Autonomous Systems." In 2021 IEEE 3rd AICAS, 2021.



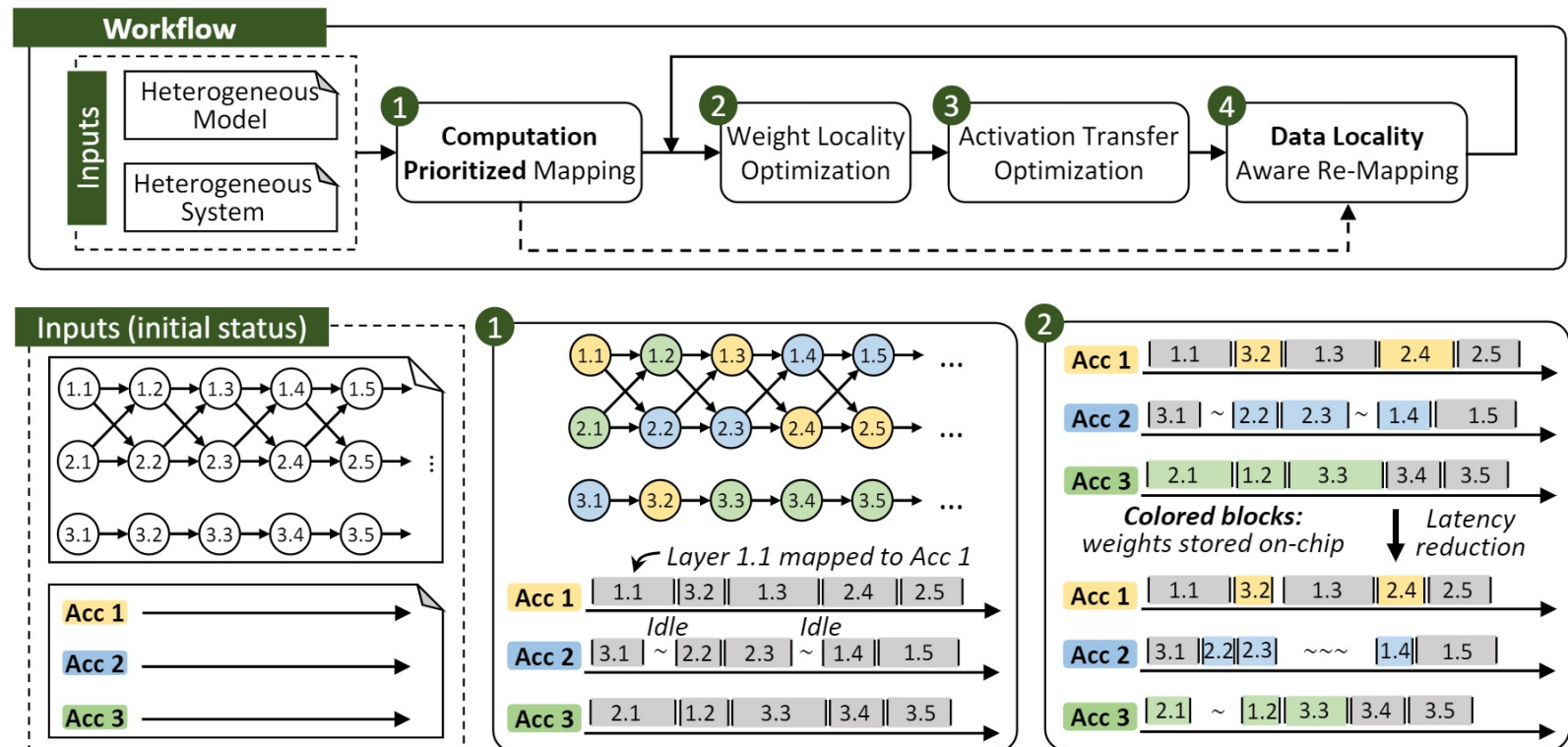
# NAIS + Scheduling + Mapping

NAIS

+

Mapping

Scheduling



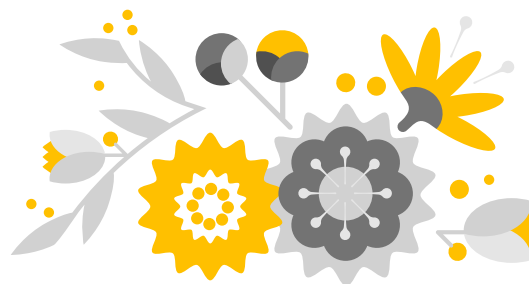
Xinyi, Zhang, Cong Hao, et al., "**H2H: Heterogeneous Model to Heterogeneous System Mapping with Computation and Communication Awareness**" To appear at DAC'22



# NAIS + MMMT + Heterogeneity

NAIS

+



Mapping

Hao, Cong, and Deming Chen. "Software/Hardware Co-design for Multi-modal Multi-task Learning in Autonomous Systems." *IEEE 3rd AICAS*, 2021.

Scheduling

Xinyi, Zhang, Cong Hao, et al., "H2H: Heterogeneous Model to Heterogeneous System Mapping with Computation and Communication Awareness" To appear at DAC'22

Implementation Optimization

Li, Yuhong, Cong Hao, et al. "EDD: Efficient differentiable dnn architecture and implementation co-search for embedded ai solutions." *ACM/IEEE DAC*, 2020





# Summary & Thanks!

1. **Basic**: DNN and Accelerator Co-design – three levels
2. **NAIS**: **simultaneous** neural architecture and implementation co-search
3. **Future**: when multi-modal multi-task (**MMMT**) models meet **heterogeneous** platforms

NAIS



Contact:

[callie.hao@ece.gatech.edu](mailto:callie.hao@ece.gatech.edu)

Sharc-lab @ Georgia Tech

(<https://sharclab.ece.gatech.edu/>)

