A SYSTEMATIC AND RAPID APPROACH TO DESIGN SPACE EXPLORATION FOR TENSOR ACCELERATORS

QIJING JENNY HUANG, NVIDIA
jennyhuang@nvidia.com

Outline

1. Accelerator design space exploration (DSE)

2. Taxonomy of DSE Tools

3. An overview of our approach

   - An optimization-driven mapper: CoSA

   - A search space transformation: VAESA

   - A differentiable formulation: DOSA

4. Challenges and opportunities

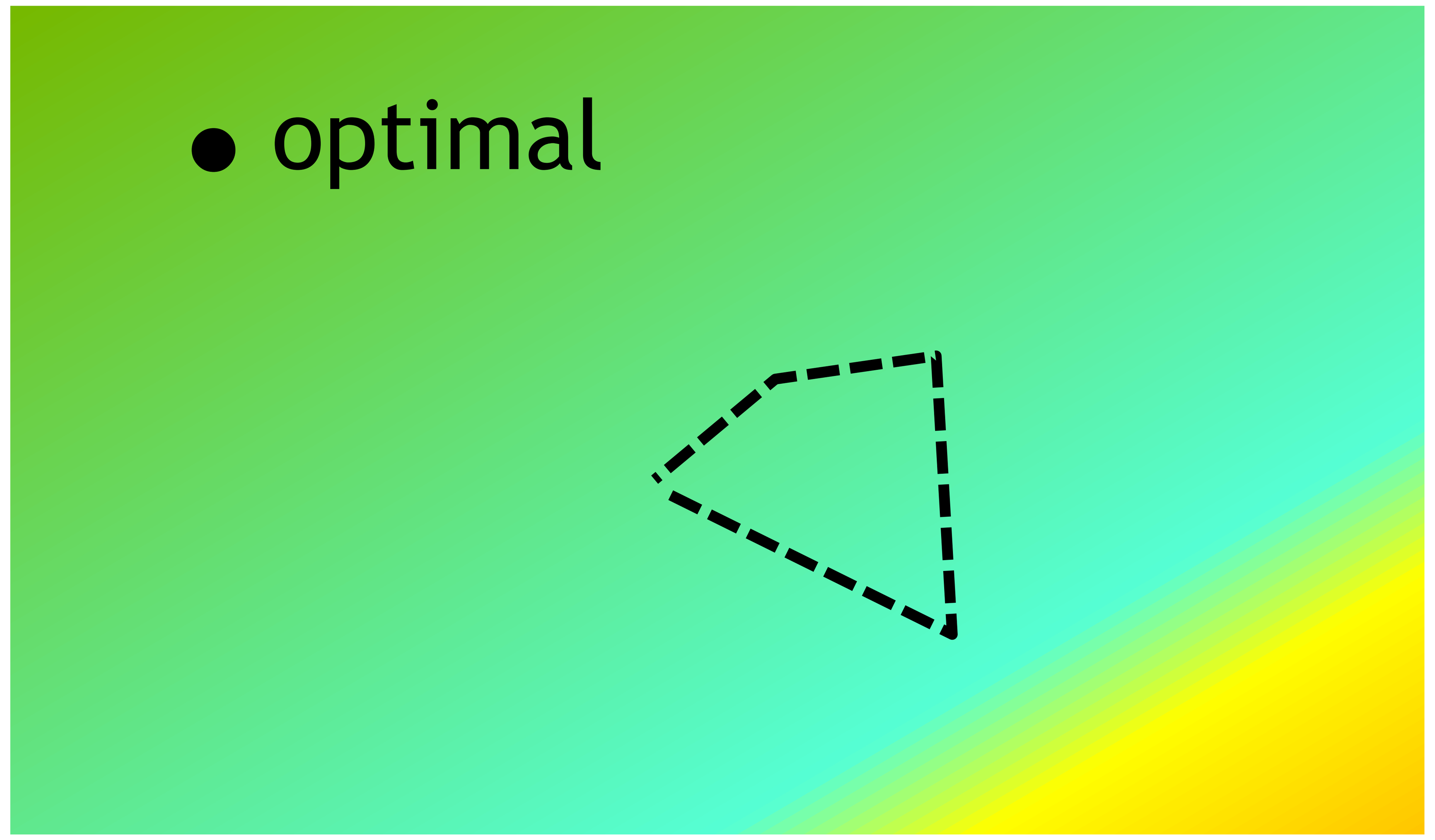
Performance feedback

“Design is not just what it looks like and feels like. Design is how it works.”

Steve Jobs

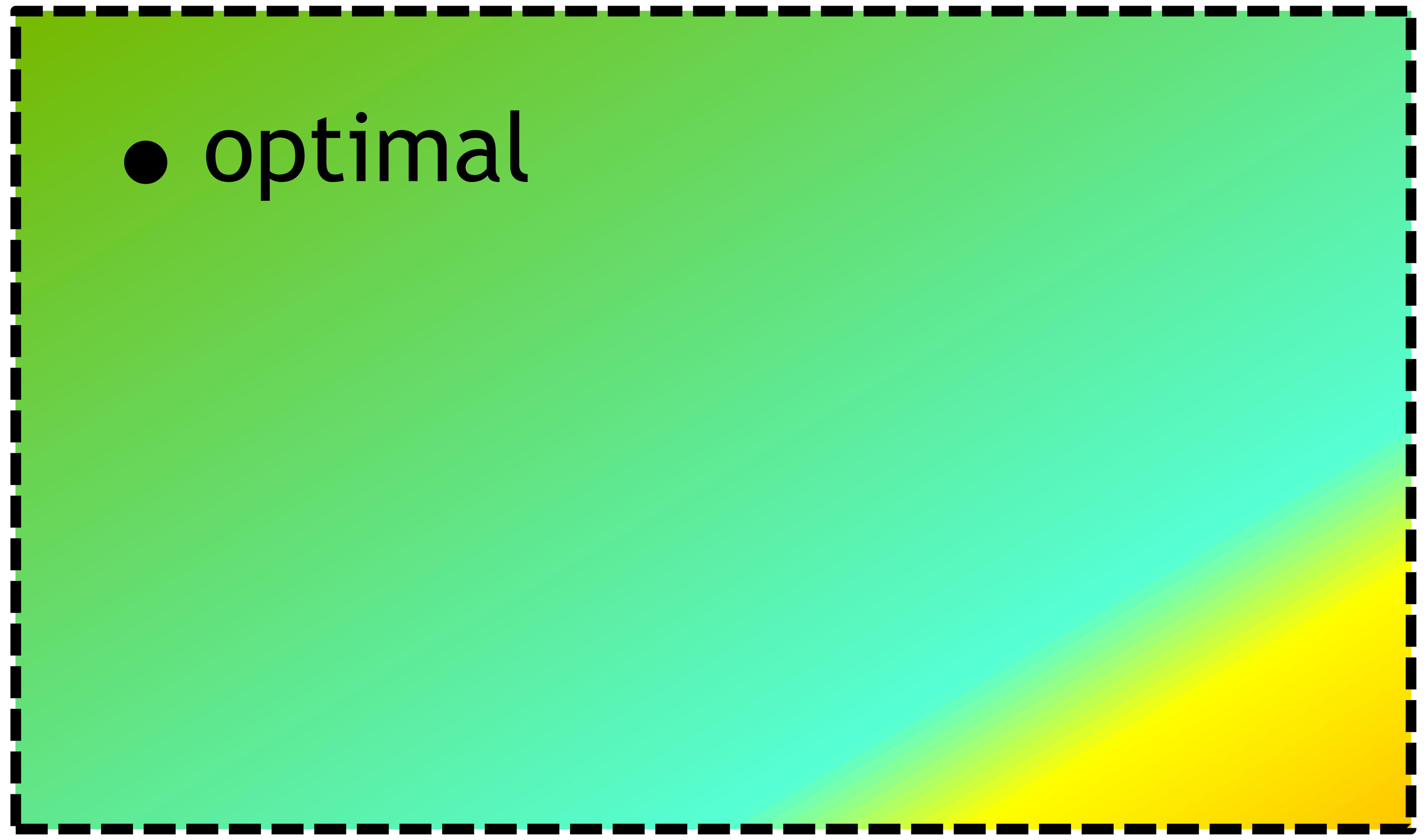
DSE-DRIVEN ARCHITECTURE DESIGN

Heuristics



Design Space

DSE



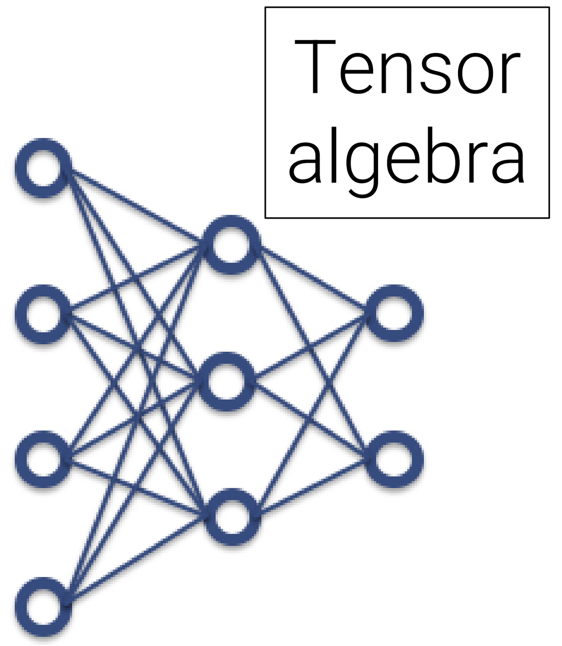
Design Space

- Architectural design should be done using DSE with:
  - Clearly defined objectives
  - A vast design space

ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and the objectives



Workload
specs

TARGET WORKLOADS

Tensor Algebra

- **Tensor algebra** is a category of computation that can be expressed by symbols and operations of tensors
  - Example workloads:
    - Matrix-Matrix Mult, Conv, BLAS, …

- Algebraic expression:

$$C_{ij} = \sum_k A_{ik} B_{kj}$$

- Implementation:

```
for i in [0, I):
 for j in [0,J):
  for k in [0,K):
   C[i][j] +=
        A[i][k] * B[k][j]
```

Matrix-Matrix Mult

TARGET WORKLOADS

Tensor Algebra

- **Tensor Algebra** is a category of computation that can be expressed by symbols and operations of tensors
  - Example workloads:
    - Matrix-Matrix Mult, Conv, BLAS, …

- Algebraic expression:

$$C_{ij} = \sum_k A_{ik} B_{kj}$$

- Implementation:

```
for i in [0, I):
 for j in [0,J):
  for k in [0,K):
   C[i][j] +=
           A[i][k] * B[k][j]
```

Matrix-Matrix Mult

Properties

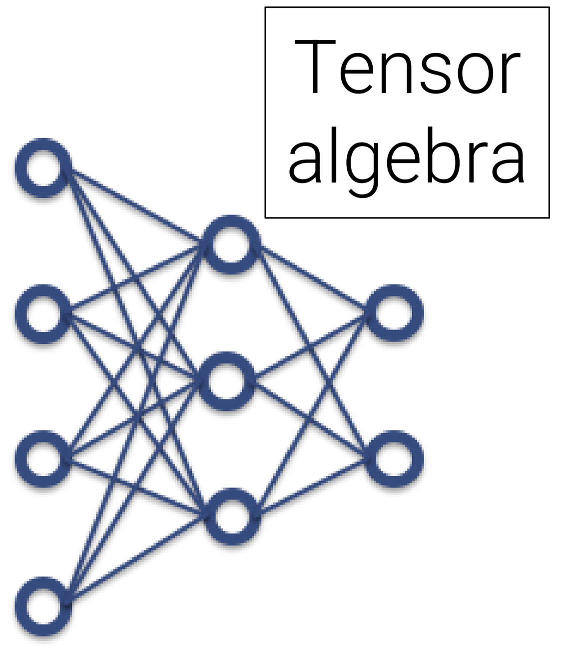
1. Known iteration space bounds
2. Regular memory access patterns
3. Repeated control flow

These properties give rise to many optimizations in accelerator DSE

ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and the objectives

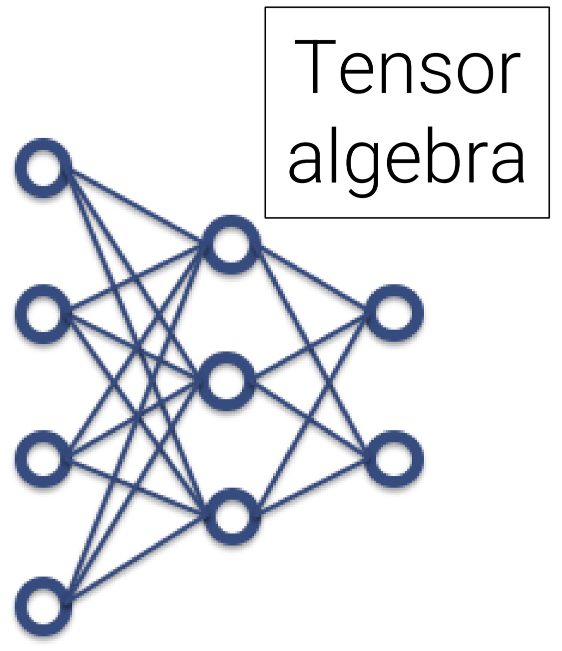


Workload specs

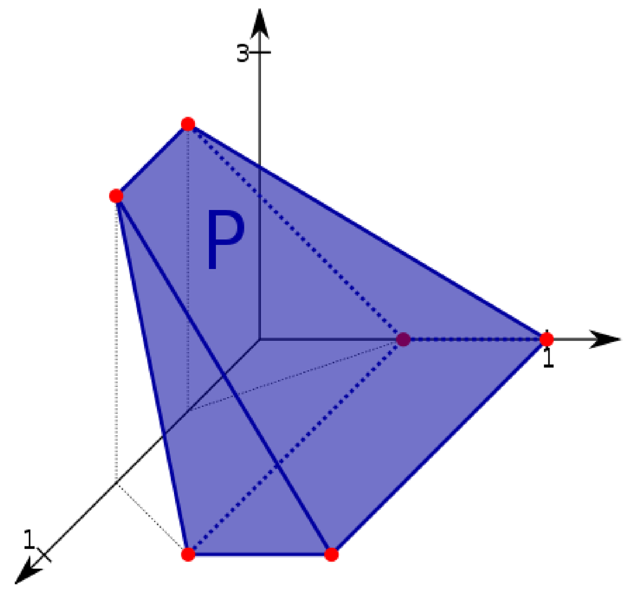
ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and objectives



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

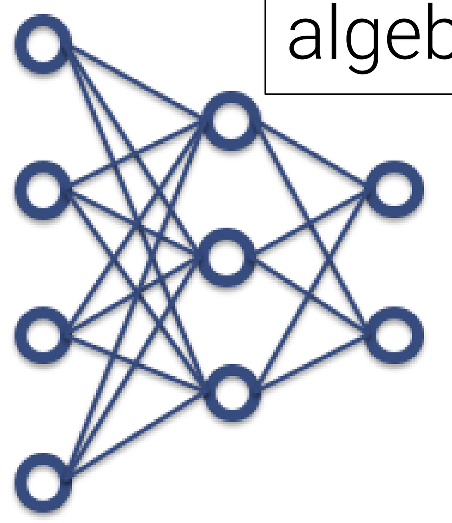
Arch design space

ACCELERATOR DESIGN SPACE EXPLORATION

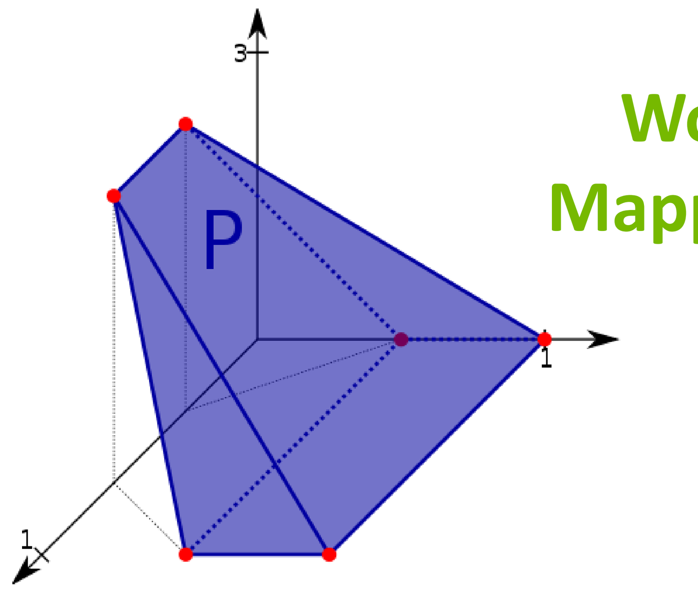
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

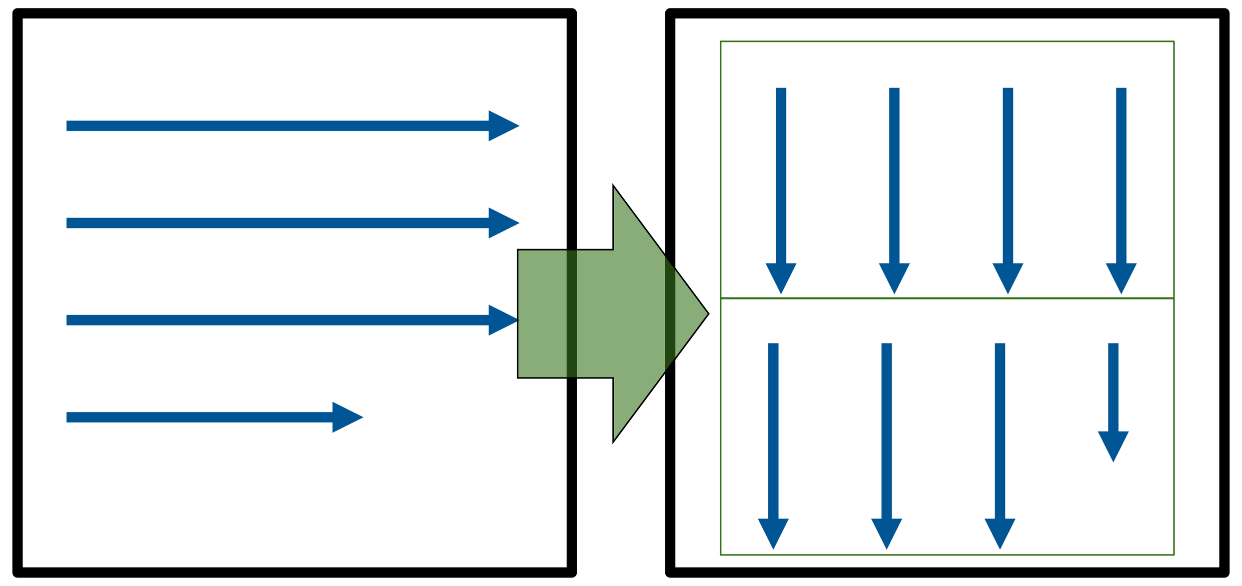
Arch design space

Workload specs
Mapping constraints

Arch design space

STEP #2: Optimize the mapping of the workload on the HW design

Mapper



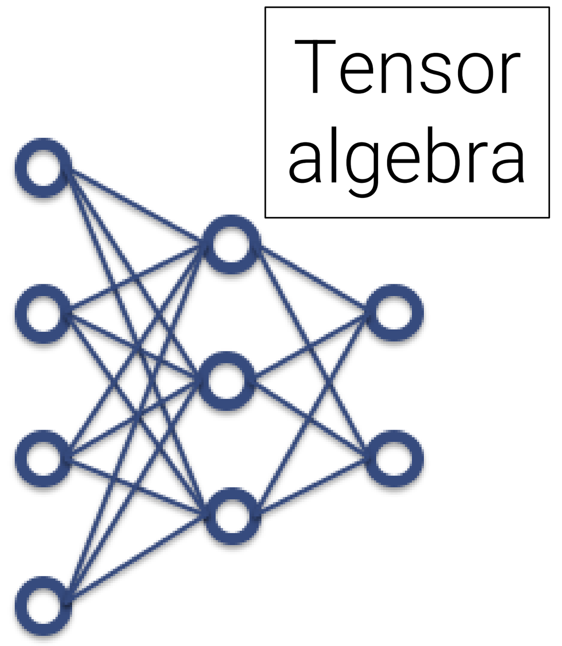
- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

New arch configs

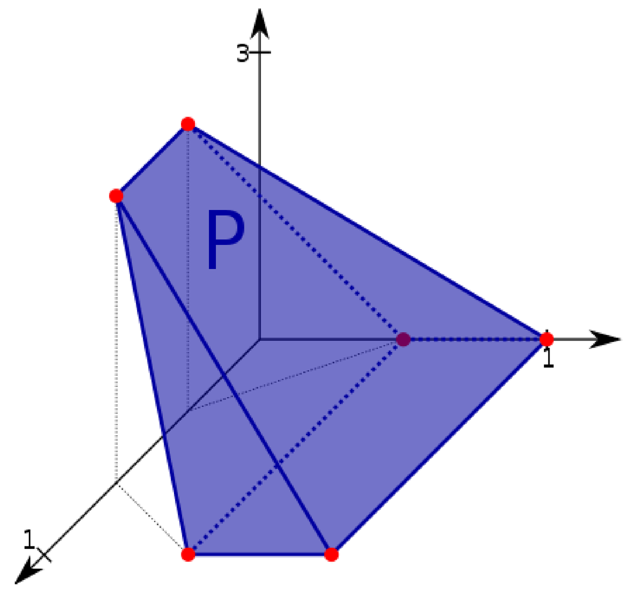
ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and objectives



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

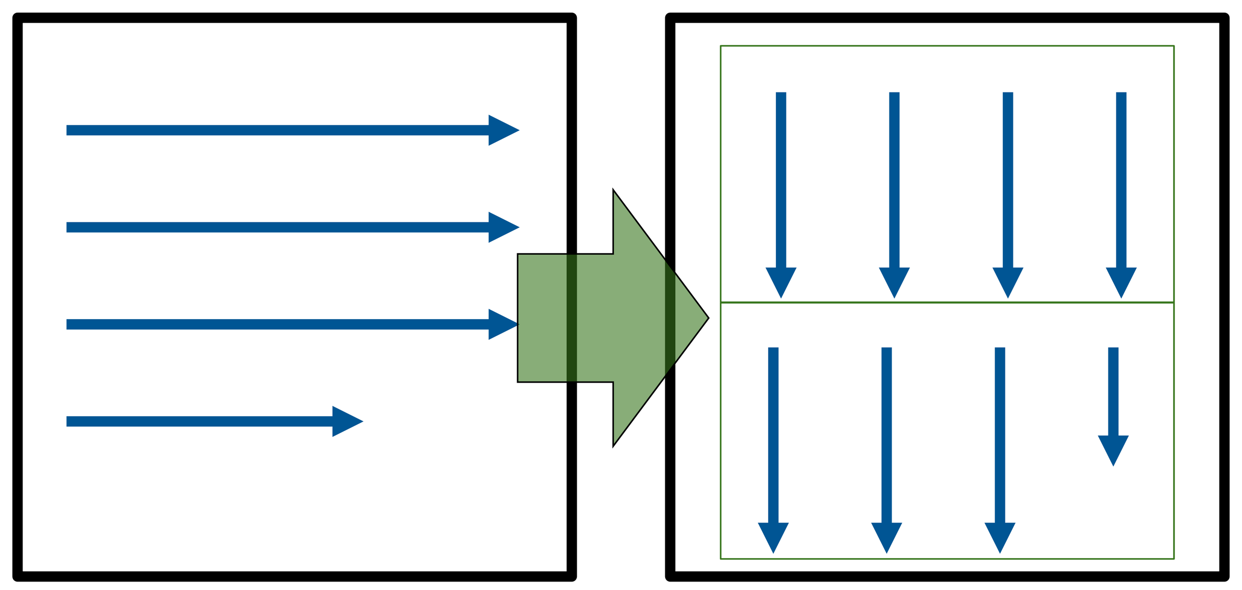
Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

Arch design space

STEP #2: Optimize the mapping of the workload on the HW design

Mapper



- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

Workload specs
Arch configs
Objectives
Mapping

STEP #3: Evaluate the performance

Evaluation Tool

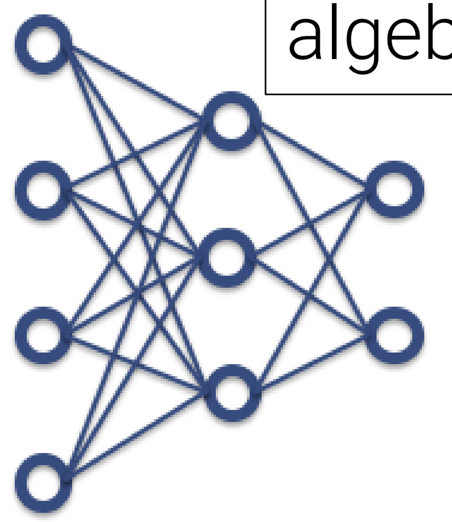
- Latency
- Energy
- Area
- ...

ACCELERATOR DESIGN SPACE EXPLORATION

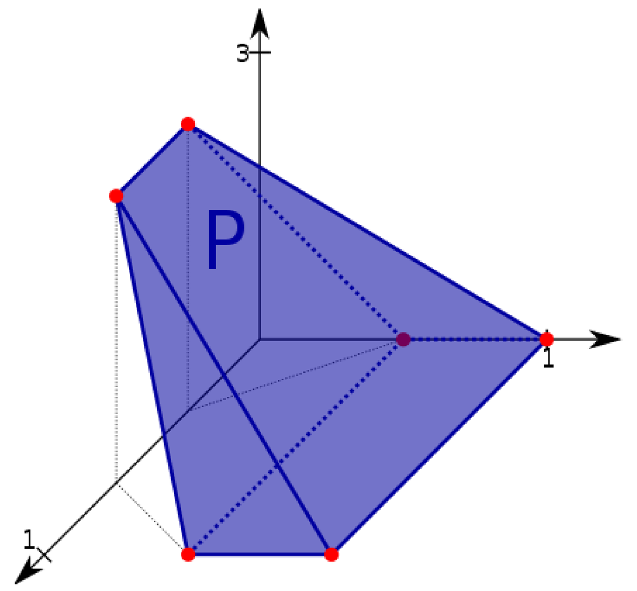
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

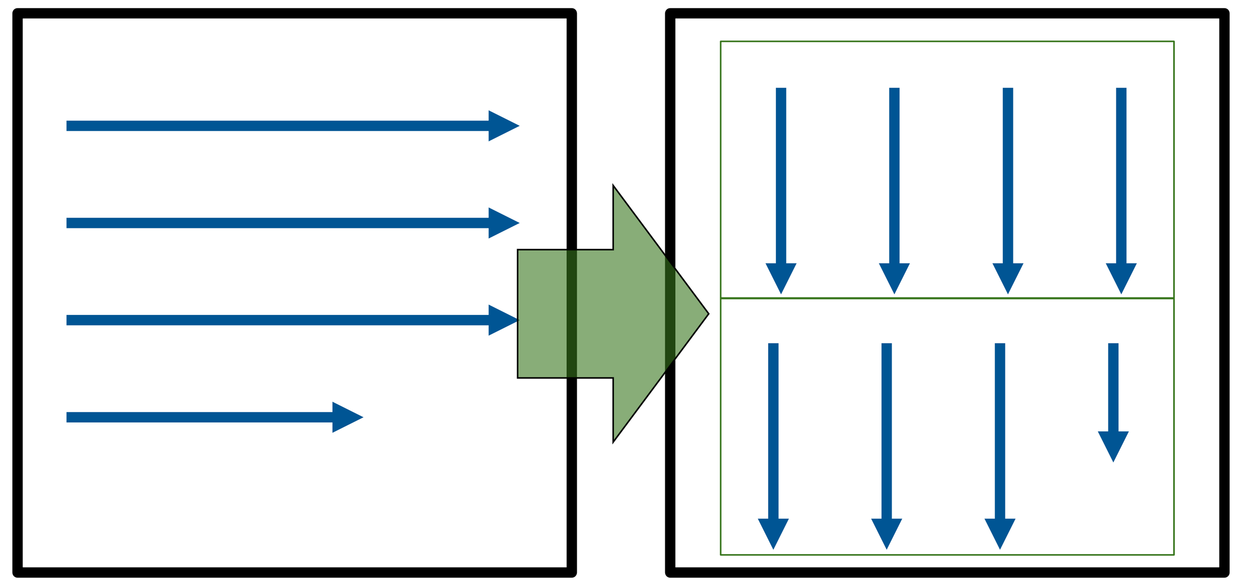
Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

Arch design space

STEP #2: Optimize the mapping of the workload on the HW design

Mapper



- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

Performance feedback

STEP #3: Evaluate the performance

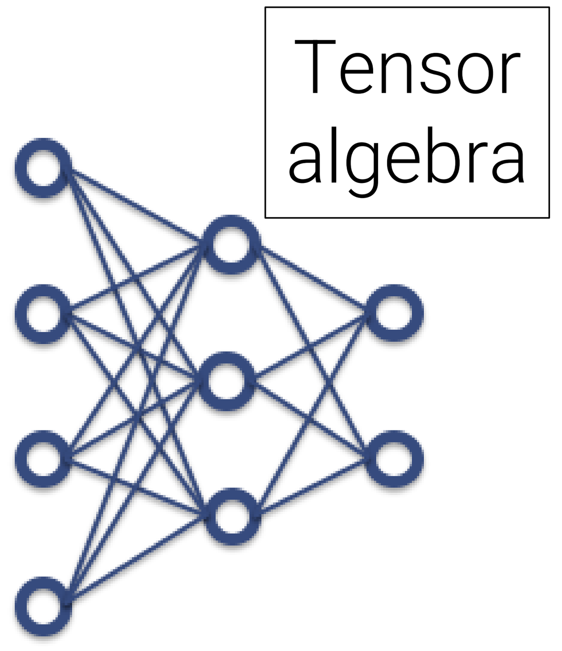
Evaluation Tool

- Latency
- Energy
- Area
- ...

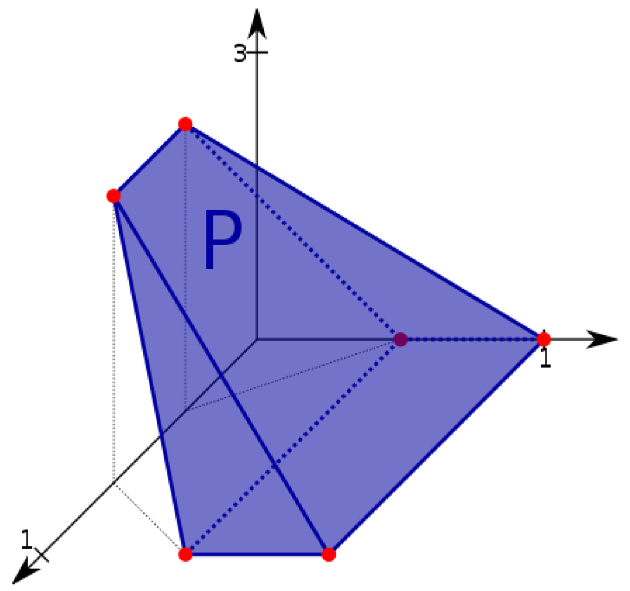
ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and objectives



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

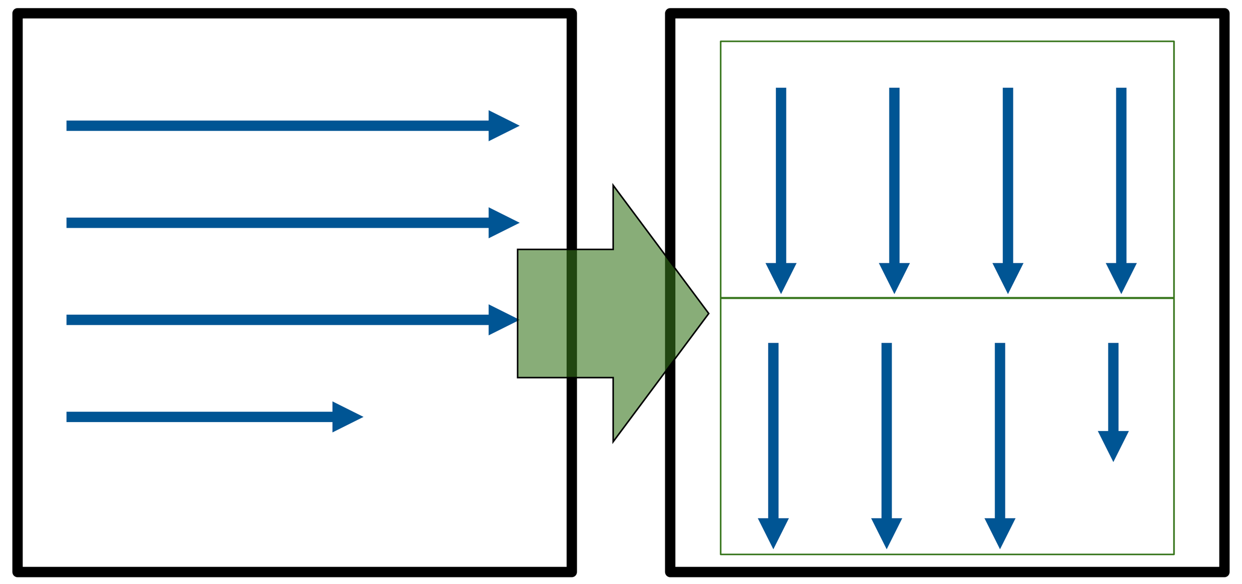
Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

Arch design space

STEP #2: Optimize the mapping of the workload on the HW design

Mapper



- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

STEP #3: Evaluate the performance

Evaluation Tool

- Latency
- Energy
- Area
- ...

Performance feedback

STEP #4: Update the search algorithm and select the next design points

Search Strategy

Workload specs

Arch design space

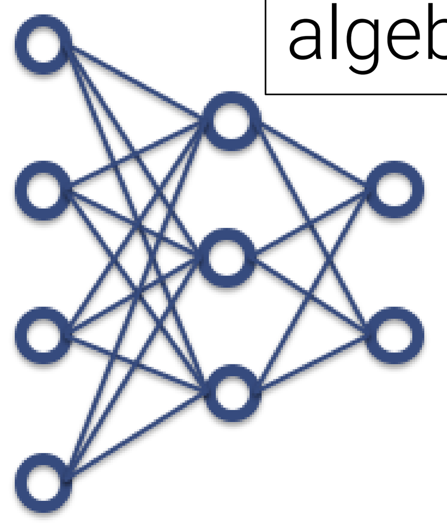
- Random Search
- Black-box Optimization
- Gradient-based Optimization
- ...

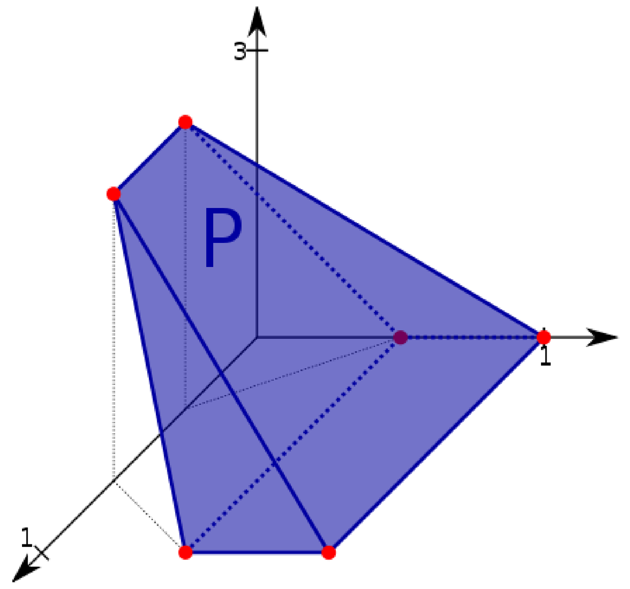
ACCELERATOR DESIGN SPACE EXPLORATION

Four key steps

Step #1: Define the design space and objectives

Tensor algebra





Workload specs

Mapping constraints

Metrics
Latency
Energy
Area
EDP
…

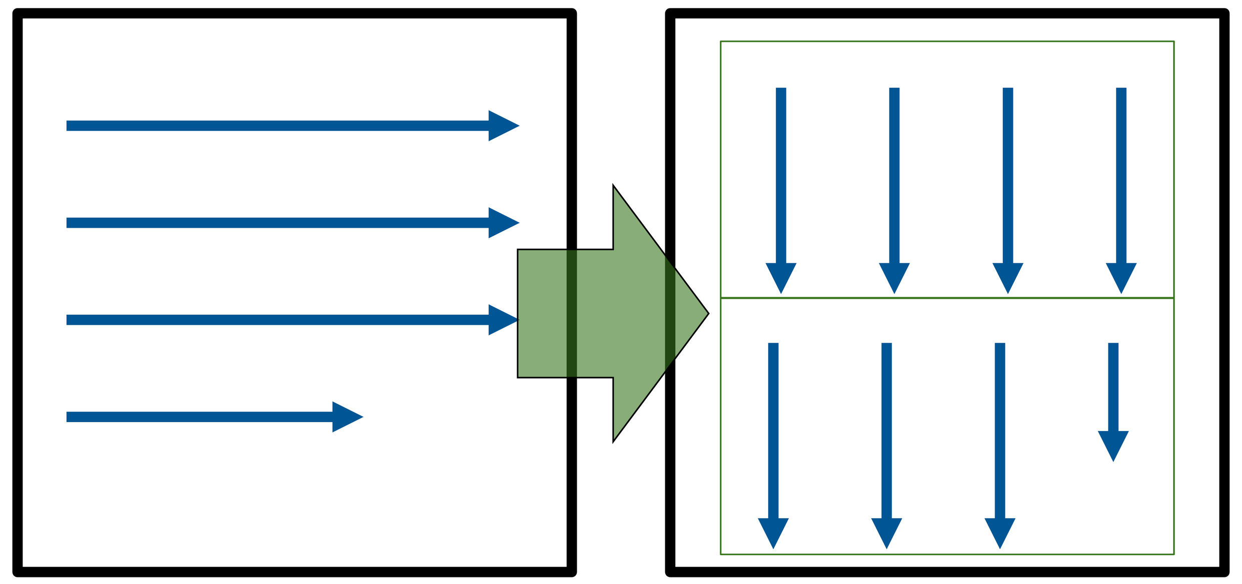
Objectives

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

Arch design space

STEP #2: Optimize the mapping of the workload on the HW design

Mapper



- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

STEP #3: Evaluate the performance

Evaluation Tool

- Latency
- Energy
- Area
- …

New arch configs

STEP #4: Update the search algorithm and select the next design points

Search Strategy

- Random Search
- Black-box Optimization
- Gradient-based Optimization
- …

NVIDIA

KEY CHALLENGES IN DSE

Large design space and costly evaluation

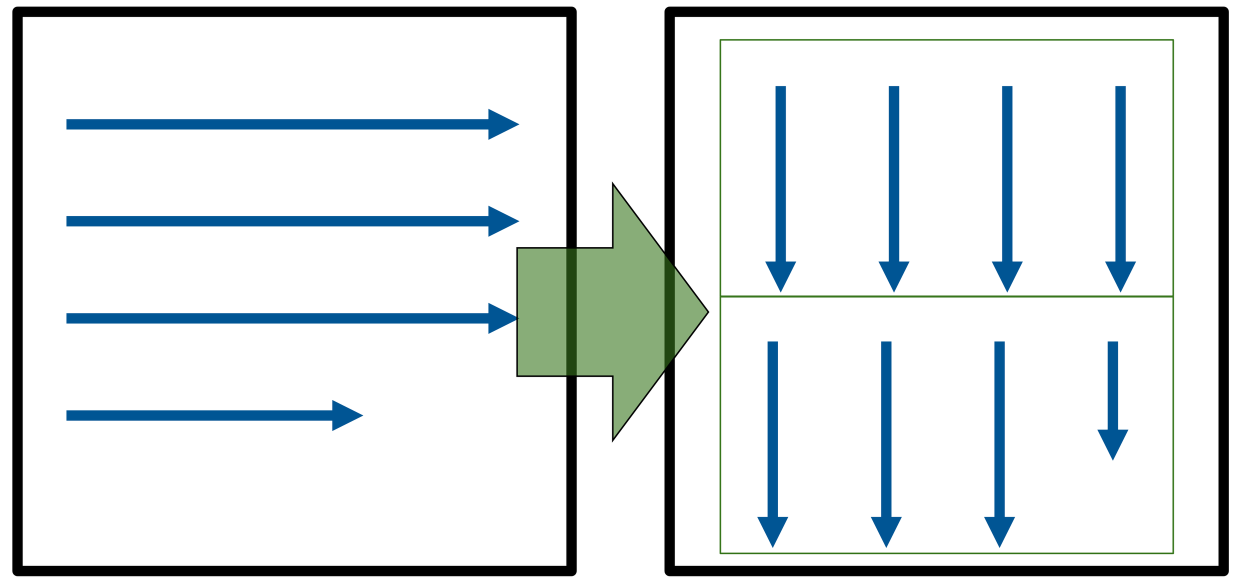
Hardware Design Space

Parameter	Value Range
# of PEs	1~64
# of MAC units	1~64
Accum. buffer size	0~1MB
Weight buffer size	0~1MB
Input buffer size	0~1MB
Global buffer size	0~32MB

$\sim 10^{17}$

×

Mapping Space



$\sim 10^{5}$

×

Evaluation Time

Platform	Evaluation Time
Timeloop	**0.01s**
FPGA	2 mins
VCS	10 mins
Power Analysis	6 hrs

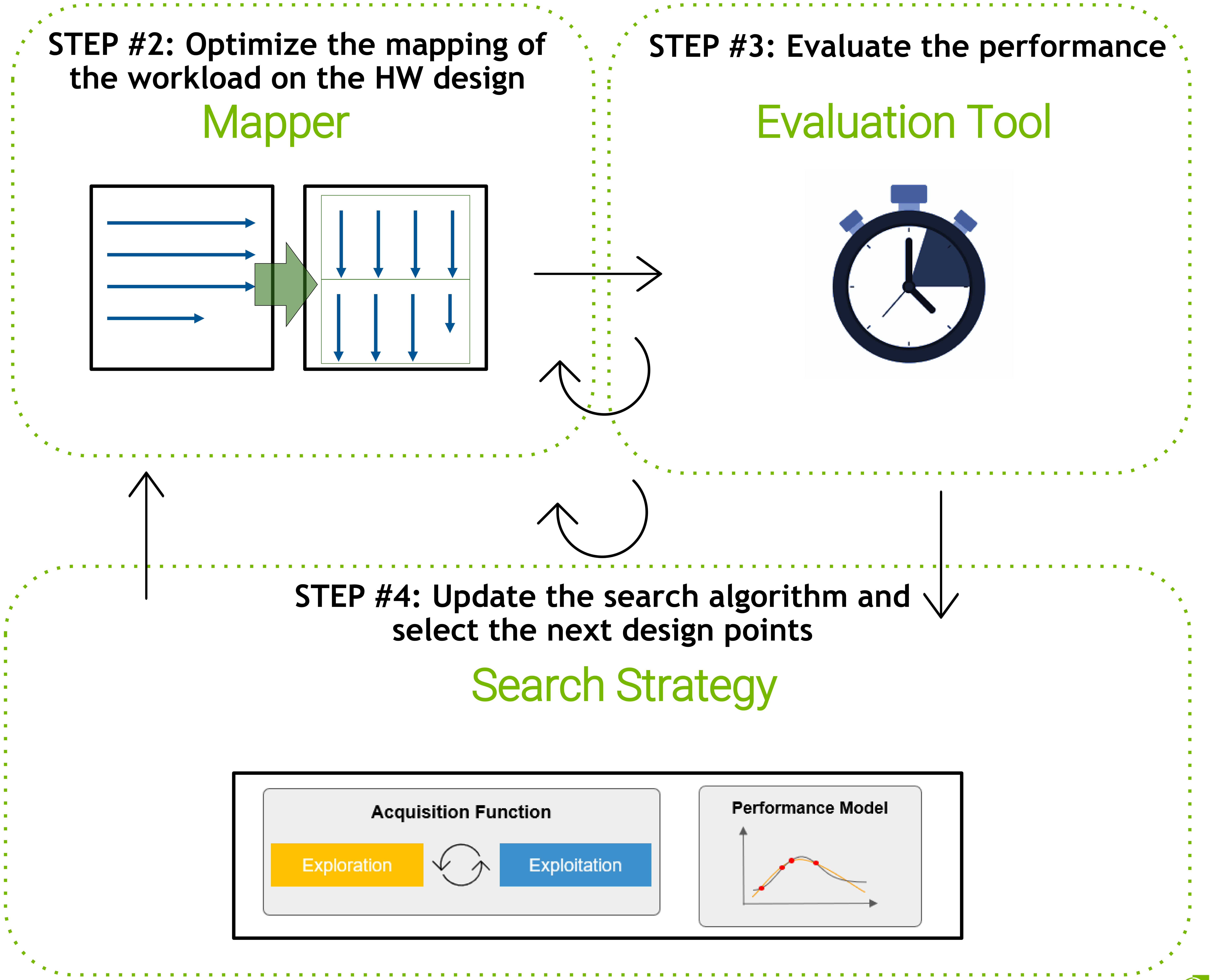
0.01s

Intractable

> 31T logical years

DESIGN SPACE EXPLORATION TOOLS

For more systematic and rapid DSE



A TAXONOMY OF ACCELERATOR EVALUATION TOOLS

Comparisons

Category	Tools	Fidelity	Modeling Speed
Polynomial Model	CoSA	Low	Very Fast
ML Model	PRIME	Medium	Fast†
Static Analysis	Timeloop, MAESTRO	Medium	Fast
Cycle-accurate Model	ScaleSim	High	Slow
RTL Simulation	FireSim, MagNet	Very High	Slow*

† Varies with ML model size
* Varies with workload size

A TAXONOMY OF ACCELERATOR EVALUATION TOOLS

Supported features

Category	Dynamic behavior support	Data/training/ implementation free	Differentiable
Polynomial Model	No	Yes	Yes
ML Model	Yes	No	Yes
Static Analysis	No	Yes	No
Cycle-accurate Model	Yes	Yes	No
RTL Simulation	Yes	No	No

A TAXONOMY OF MAPPERS

Heuristic-Driven

Timeloop
Triton Marvel

Feedback-based

AutoTVM Ansor
Halide Gamma
MindMapping

Constrained Optimization

Polly+Pluto TC
Tiramisu CoSA
IOOpt

- Easy to implement

- More adaptive

- Costly
- Sample invalid space
- Hard to generalize

- More sample efficient

- Limited use case

A TAXONOMY OF ACCELERATOR SEARCH STRATEGIES

	Heuristic-Driven	**Black-box Optimization**	**Gradient-based Optimization**
Original Space	Interstellar	Bayesian Opt: Apollo Evolutionary Algo: PRIME NAAS AutoSA	GD: EDD DiffTune RL: ConfuciuX APN CoexplorationNAS
Latent Space	VAESA		
	- Easy to implement	- Perf model not needed	- Lower evaluation cost
	- Limited DSE	- High evaluation cost	- Data hungry

Search-algorithm focused

Design-space focused

NVIDIA

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time **Intractable**

Search Strategy:
ML-Based
VAESA

Mapper:
Opt-Driven
CoSA

Evaluation tool:
Static Anaysis
Timeloop

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

Intractable

Search Strategy:
ML-Based
VAESA

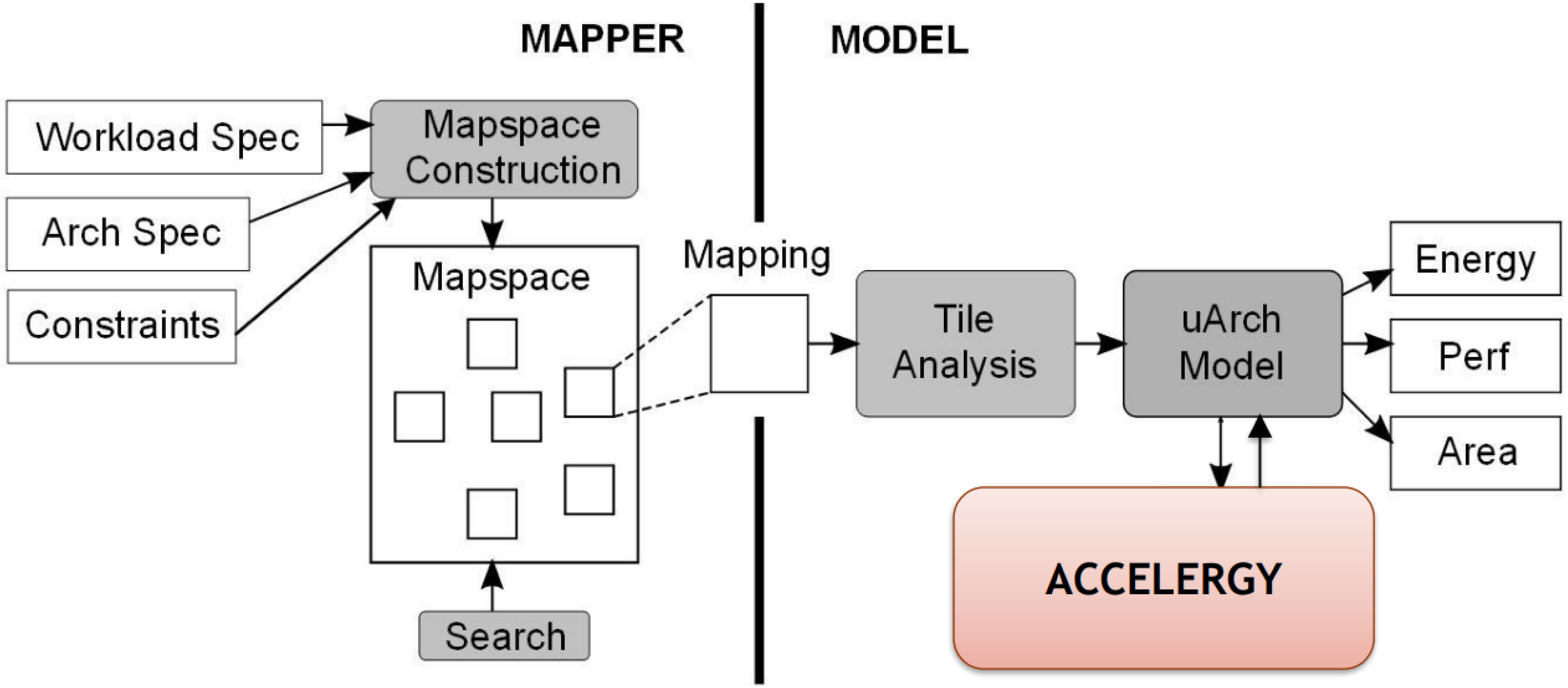
Mapper:
Opt-Driven
CoSA

Evaluation tool:
Static Anaysis
Timeloop [ISPASS'19]

* **Timeloop: A systematic approach to DNN accelerator evaluation**. Parashar A, Raina P, Shao YS, Chen YH, Ying VA, Mukkara A, Venkatesan R, Khailany B, Keckler SW, Emer J. ISPASS'19

A STATIC ANALYSIS TOOL

Timeloop/Accelergy



- **Timeloop** provides a **flexible abstraction** to define a wide range of **applications, architectures** and **constraints**
- **Timeloop** rapidly and accurately reports **latency**, **energy**, **area** using static analysis

* **Timeloop: A systematic approach to DNN accelerator evaluation**. Parashar A, Raina P, Shao YS, Chen YH, Ying VA, Mukkara A, Venkatesan R, Khailany B, Keckler SW, Emer J. ISPASS'19

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time **Intractable**

Search Strategy: ML-Based VAESA

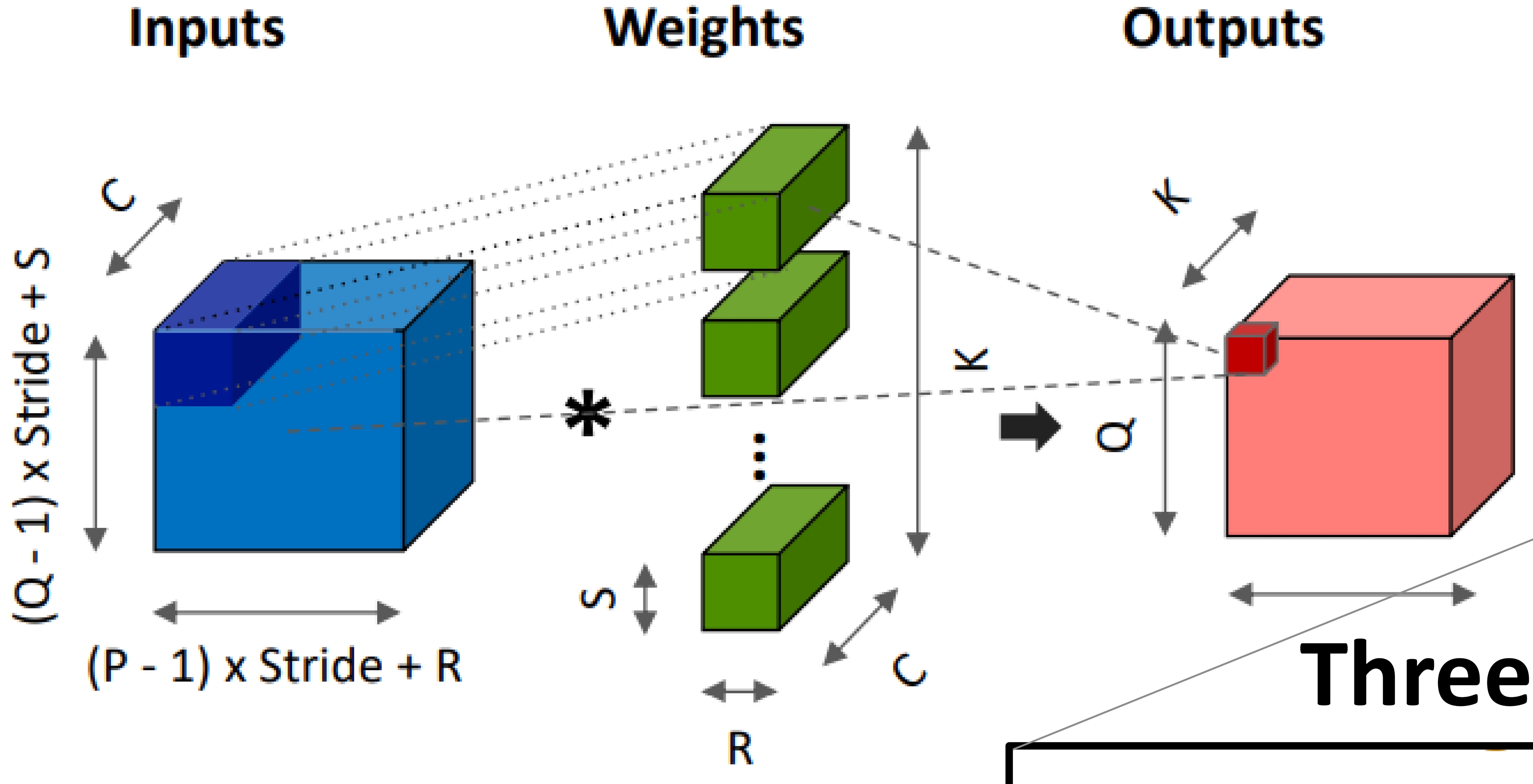
Mapper: **Opt-Driven CoSA [ISCA'21]**

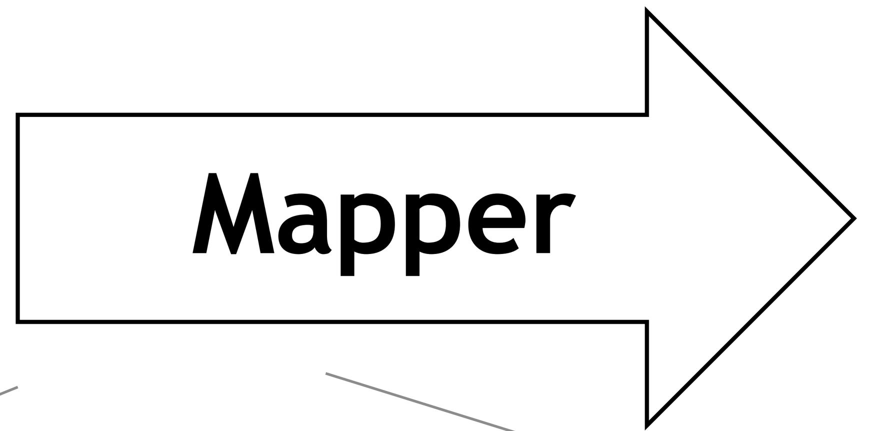
Evaluation tool: Static Anaysis Timeloop/Accelergy

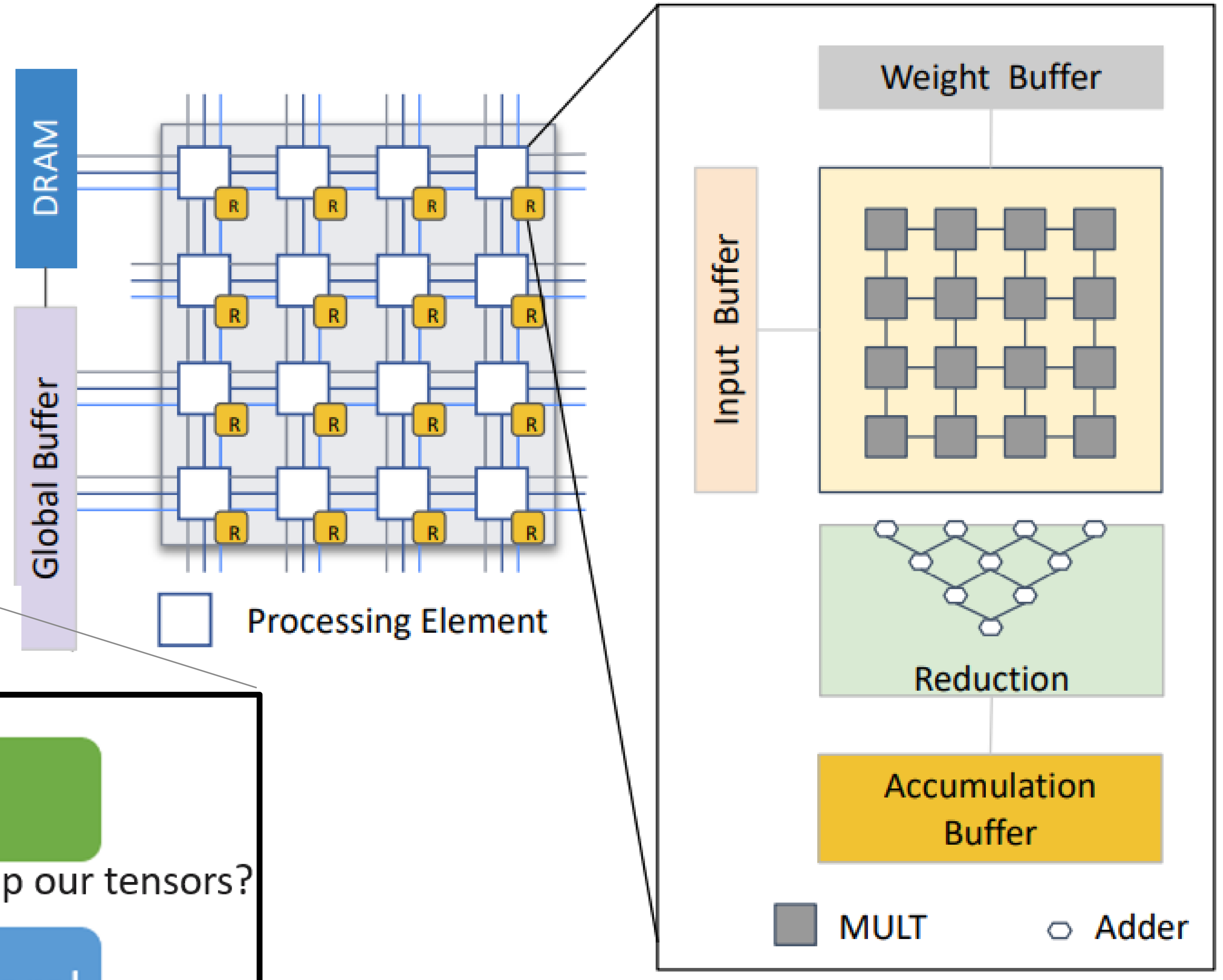
* **CoSA: Scheduling by constrained optimization for spatial accelerators**. Huang Q, Kang M, Dinh G,, Norell T, Kalaiah A, Demmel J, Wawrzynek J, Shao YS. ISCA'21

AN OPTIMIZATION-DRIVEN MAPPER

CoSA







R, S: weight width and height
P, Q: output width and height
C: input channel size
K: output channel size
N: batch size

Three key mapping decisions

DRAM level
for q2 = [0 : 2) :
Global Buffer level
for q1 = [0 : 7) :
for n0 = [0 : 3) :
spatial_for r0 = [0 : 3) :
spatial_for k1 = [0 : 2) :
Input Buffer level
for c1 = [0 : 2) :
for p1 = [0 : 2) :
Weight Buffer level
for p0 = [0 : 2) :
spatial_for k0 = [0 : 2) :

1. Tiling Factors
How do we break up our tensors?

2. Spatial / Temporal
Which loops do we parallelize?

3. Loop Permutation
How do we order the loops?
What dataflow do we use"

- CoSA formulates the mapping decisions into a constrained optimization problem and solves it **in one shot**

* **CoSA: Scheduling by constrained optimization for spatial accelerators**. Huang Q, Kang M, Dinh G,, Norell T, Kalaiah A, Demmel J, Wawrzynek J, Shao YS. ISCA'21

AN OPTIMIZATION-DRIVEN MAPPER

Key idea: problem factor allocation

Matrix-vector mult:

```
for c in [0:C)  // C = 28
  for k in [0:K)   // K = 15
      OA[k] += IA[c] × W[c,k]
```

	C=28			K=15	
Prime Factors	2	2	7	3	5
WeightBuf	✓				
GlobalBuf		✓		✓	✓
DRAM			✓		

GlobalBuf

	C=28			K=15	
Prime Factors	2	2	7	3	5
Spatial				✓	
Temporal	✓				✓

GlobalBuf

	C=28			K=15	
Prime Factors	2	2	7	3	5
rank0	✓				
rank1					✓
rank4					

- **An optimization variable** can be used to represent all three mapping decisions
- CoSA optimizes the variable using the **constraints** and **objectives** formulated in mixed integer programming
- CoSA finds mappings that are **1.5x faster and 1.2x more energy-efficient** while improving the time-to-solution by **90x**

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time **Intractable**

Search Strategy:
ML-Based
VAESA [ISPASS'22]

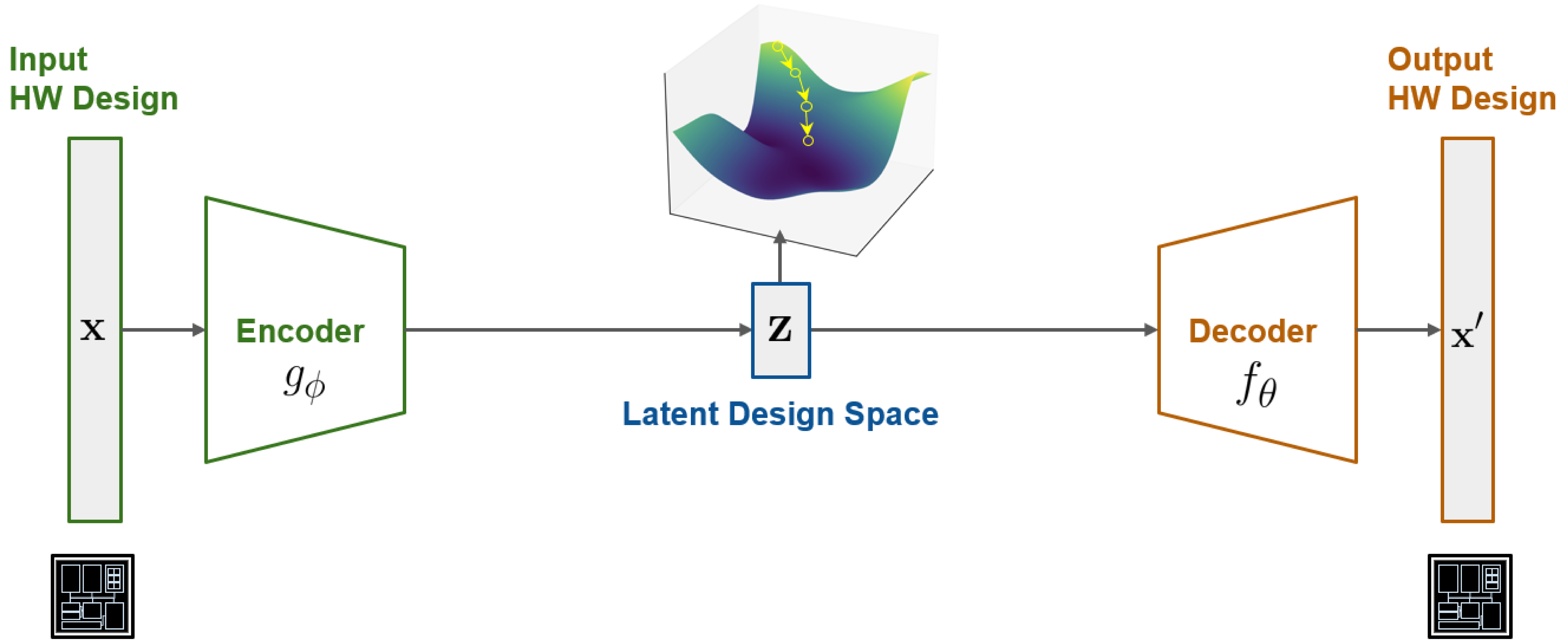
Mapper:
Opt-Driven
CoSA

Evaluation tool:
Static Anaysis
Timeloop/Accelergy

* **VAESA: Learning A Continuous and Reconstructible Latent Space for Hardware Accelerator Design**. Huang Q, Hong C, Wawrzynek J, Subedar M, Shao YS. ISPASS'22

A ML-BASED SEARCH STRATEGY

VAESA

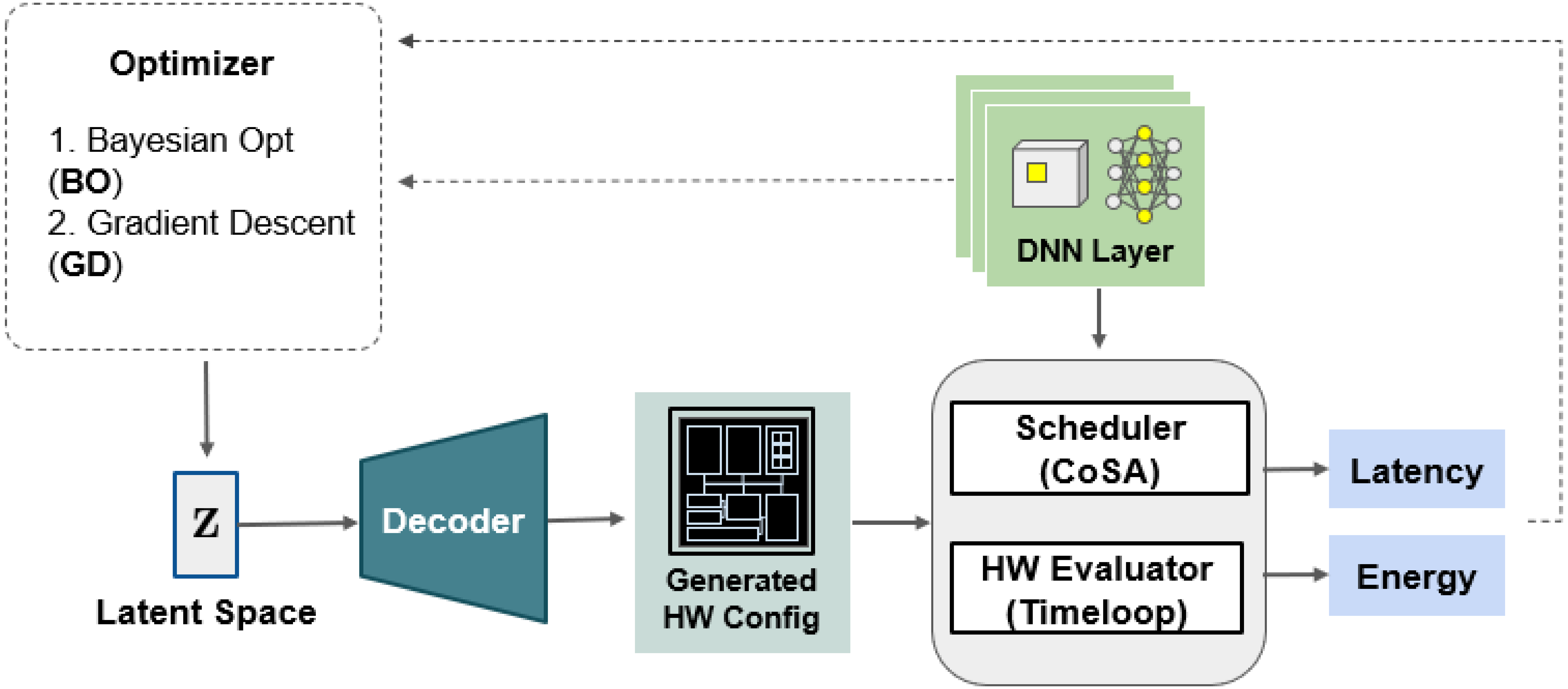


- **VAESA** learns a **low dimensional, continuous, reconstructible** latent space to facilitate accelerator DSE using **Variational Autoencoder (VAE)**

* **VAESA: Learning A Continuous and Reconstructible Latent Space for Hardware Accelerator Design**. Huang Q, Hong C, Wawrzynek J, Subedar M, Shao YS. ISPASS'22

A ML-BASED SEARCH STRATEGY

VAESA Inference



- The search algorithms are applied to the latent space and evaluated on the original search
- The latent space **reduces the search complexity** and provides a **smoother performance surface**
- Both Bayesian Optimization and Gradient Descent **achieve better sample efficiency**

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

Search Strategy:
ML-Based
VAESA

$\sim 10^{\text{latent\_dim}}$

Mapper:
Opt-Driven
CoSA

1

Evaluation tool:
Static Anaysis
Timeloop/Accelergy

0.01s

Significantly reduced DSE costs

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

One-loop DSE:

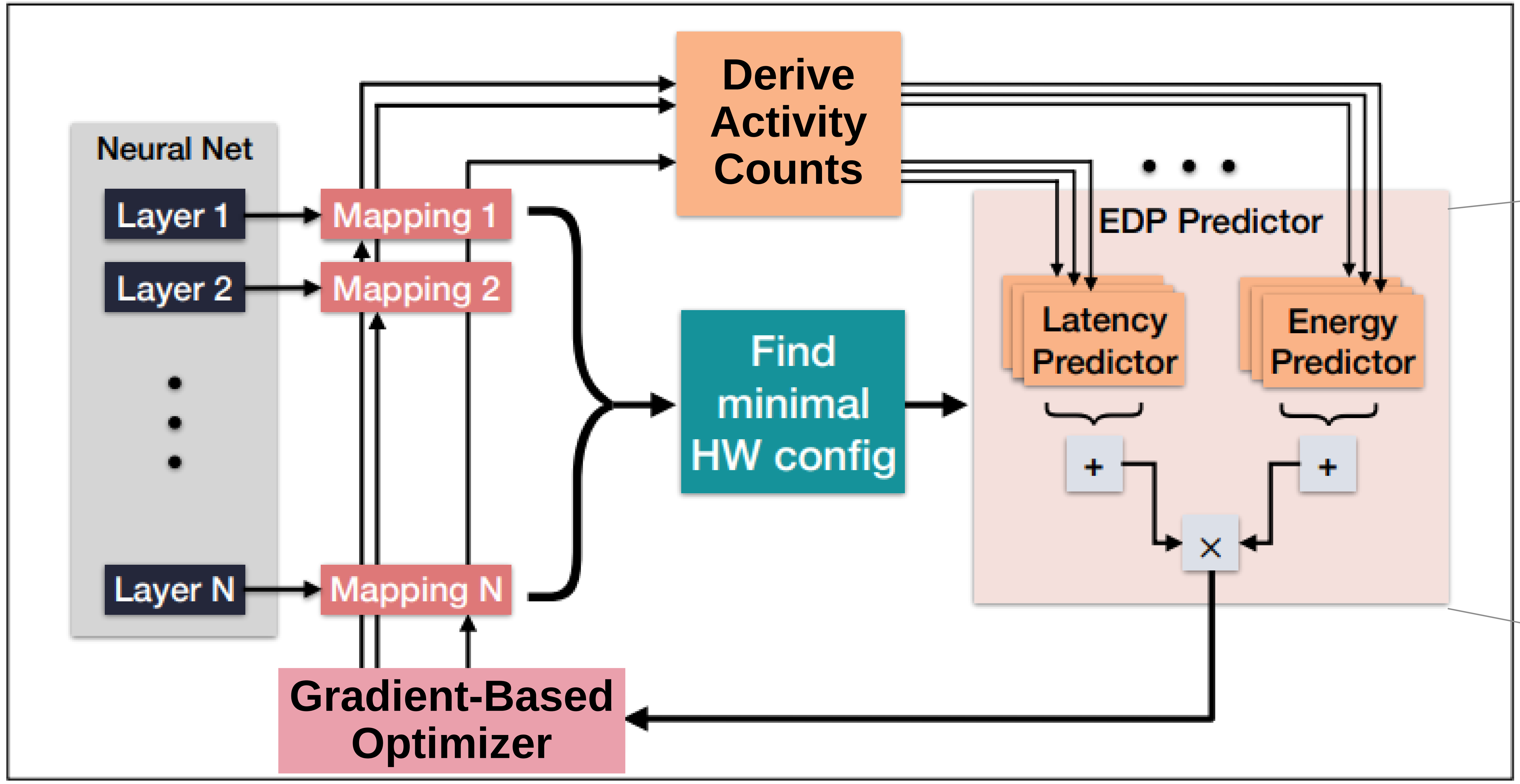
Differentiable Formulation

DOSA [MICRO’23]

***DOSA: Differentiable Model-based One-loop Search for DNN Accelerators.** Hong C, Huang Q, Dinh G, Subedar M, Shao YS. MICRO’23

A DIFFERENTIABLE DSE FORMULATION

DOSA



$$\text{compute latency} = \frac{\#\text{ of MACs in layer}}{\prod(\text{spatial tiling factors})}$$

$$\text{mem latency} = \frac{\text{accesses}}{\text{bandwidth}}$$

$$\text{latency} = \max(\text{compute latency}, \text{mem latency})$$

$$\text{energy} = \sum \text{accesses} \times \frac{\text{energy}}{\text{access}}$$

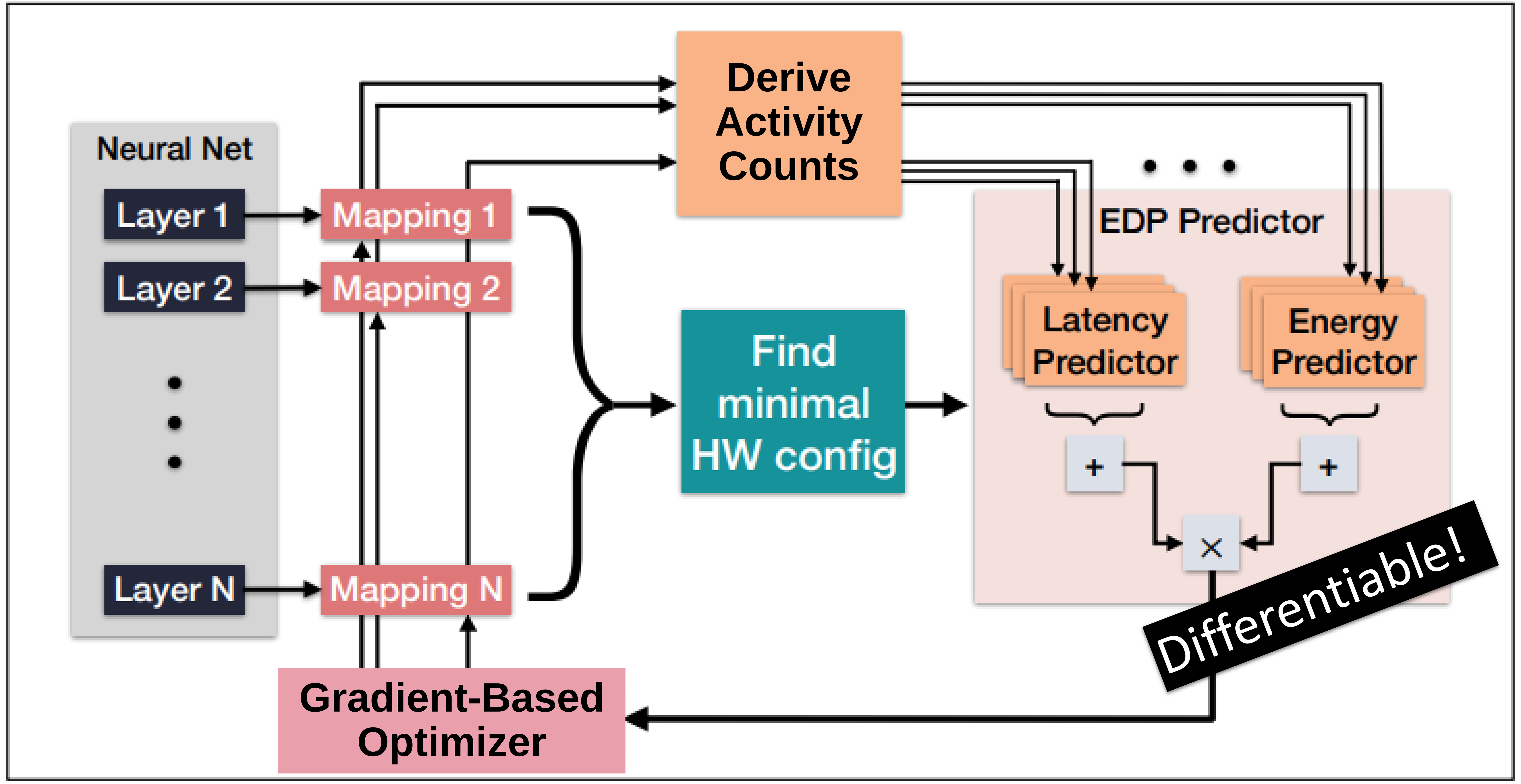
Differentiable!

- DOSA explicitly expresses the latency and energy performance as a **differentiable function** of mappings to enable gradient-based optimization.

***DOSA: Differentiable Model-based One-loop Search for DNN Accelerators.** Hong C, Huang Q, Dinh G, Subedar M, Shao YS. MICRO’23

A DIFFERENTIABLE DSE FORMULATION

DOSA



- Hardware design configurations are derived from the optimized in the one-loop mapping first DSE.
- DOSA's differentiable analytical model accurately predicts the performance and addresses the generalizability issues of the data-driven counter parts.
- It further improves the search efficiency and QoR in DSE.

DOSA: Differentiable Model-based One-loop Search for DNN Accelerators. Hong C, Huang Q, Dinh G, Subedar M, Shao YS. MICRO'23

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

Search Strategy:
ML-Based
VAESA

Mapper:
Opt-Driven
CoSA

Evaluation tool:
Static Anaysis
Timeloop/Accelergy

One-loop DSE:
Differentiable Formulation
DOSA

Significantly reduced DSE costs

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

Search Strategy:
ML-Based
VAESA

Mapper:
Opt-Driven
CoSA

Evaluation tool:
Static Anaysis
Timeloop/Accelergy

One-loop DSE:
Differentiable Formulation
DOSA

Limited design insight

DESIGN SPACE EXPLORATION TOOLS

Our approach

Hardware Design Space × Mapping Space × Evaluation Time

Early Stage DSE:

Performance Bound Derivation

Mind the Gap [ISCA'24]

July 01, Monday 11:15am

OPEN CHALLENGES AND OPPORTUNITIES

#1 DSE for diverse workloads

- irregular
- input-dependent
- multi-tenant

#2 DSE for dynamic system components

- SW/OS schedulers
- caching/paging

#3 DSE for different execution models

- selection among cpu, vector, tensor units
- customization vs programmability tradeoffs

#4 DSE generalizability for new workloads and constraints

- no consensus on DSE tools
- no uniform abstraction
- high-quality data is limited

NVIDIA

THANK YOU

QIJING JENNY HUANG, NVIDIA
jennyhuang@nvidia.com