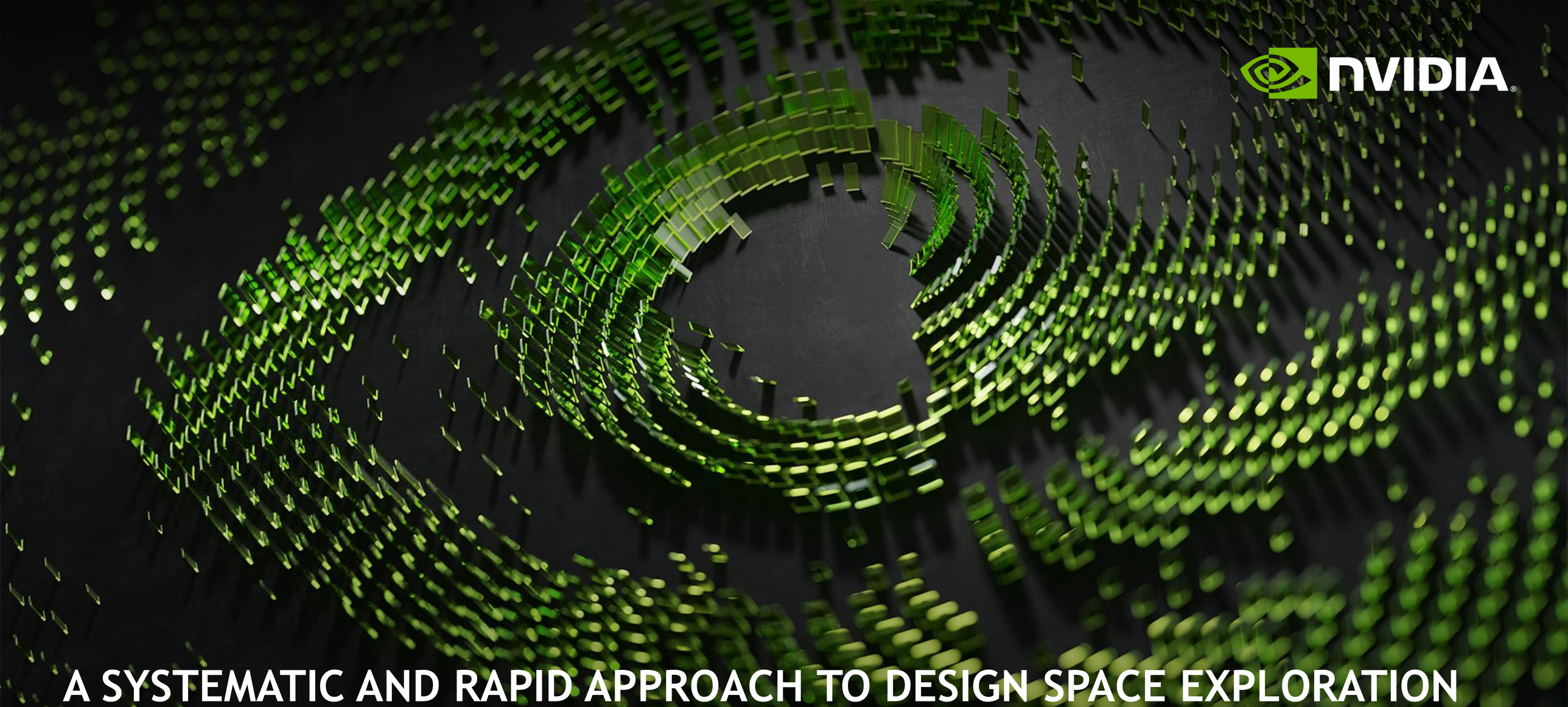


A SYSTEMATIC AND RAPID APPROACH TO DESIGN SPACE EXPLORATION FOR TENSOR ALGEBRA ACCELERATORS

QIJING JENNY HUANG, NVIDIA
jennyhuang@nvidia.com

* The views and opinions expressed in this presentation are those of the speakers and do not necessarily reflect the views or positions of any entities they represent.

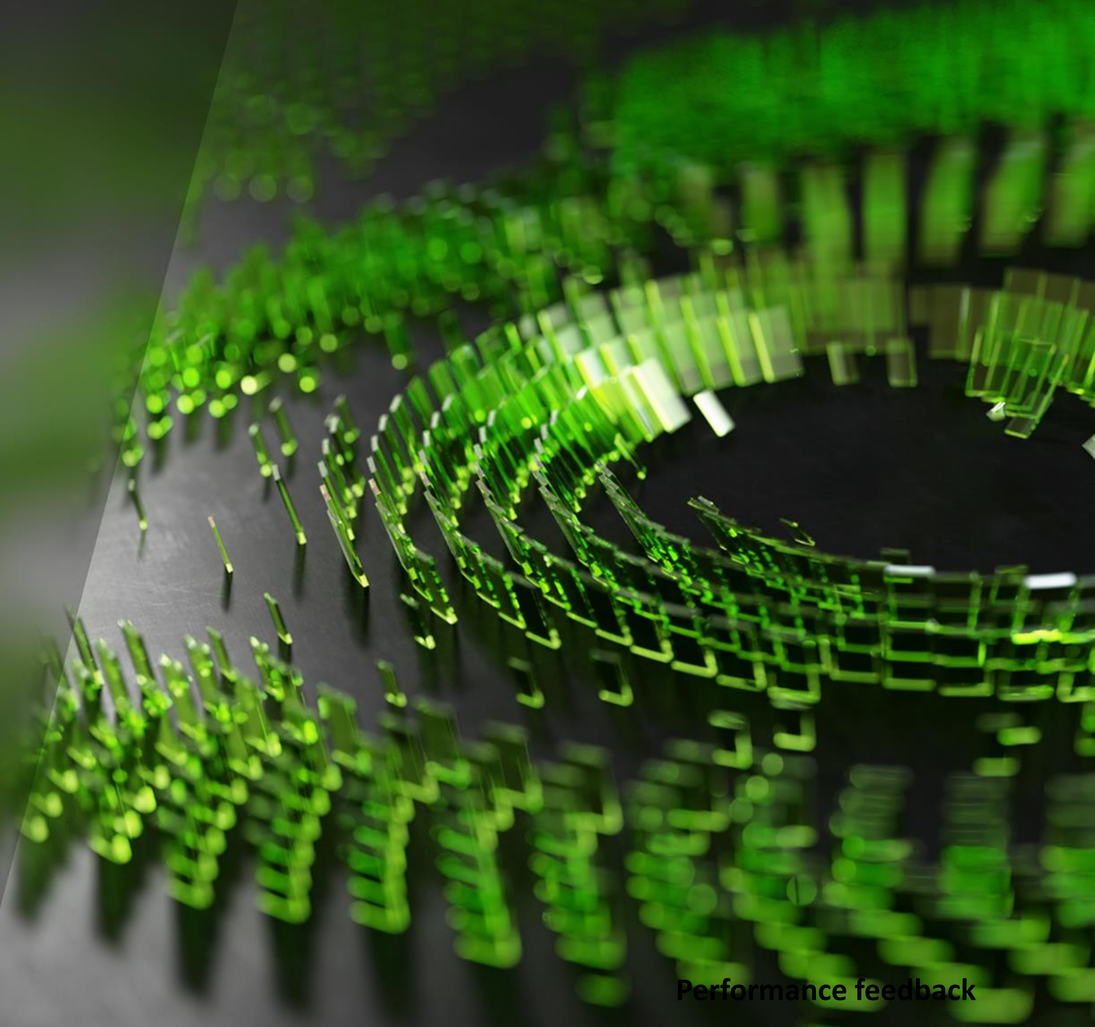
Outline

1. Accelerator design space exploration (5mins)

2. Taxonomy of DSE Tools (5mins)

3. An overview of our approach (10mins)
   - A systematic analysis tool: Timeloop
   - An optimization-driven mapper: CoSA
   - An ML-based search strategy: VAESA

4. Open challenges and opportunities (5mins)



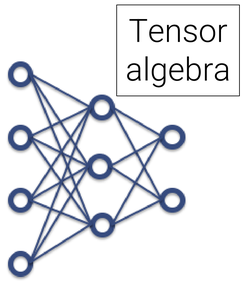

“Design is not just what it looks like and feels like. Design is how it works.”
Steve Jobs


“Design is not just what it looks like and feels like. Design is how it works. And it should be done using design space exploration (DSE)!”

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, …

NVIDIA.

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

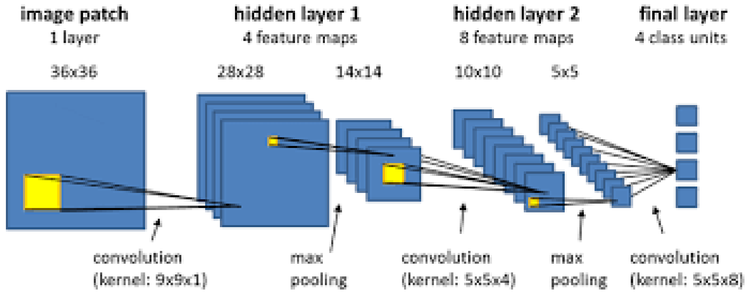
1. Known iteration space bounds
2. Regular memory access patterns
3. Repeated control flow

These properties give rise to many optimizations in accelerator DSE

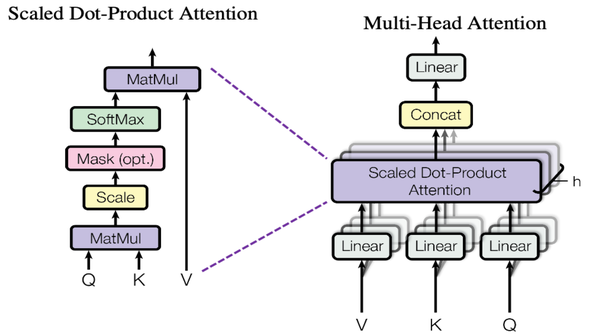
NVIDIA.

TARGET WORKLOADS

- Machine learning

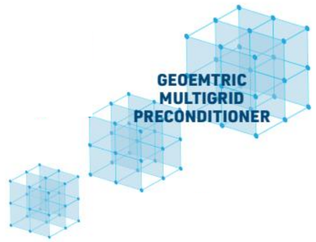


CNNs

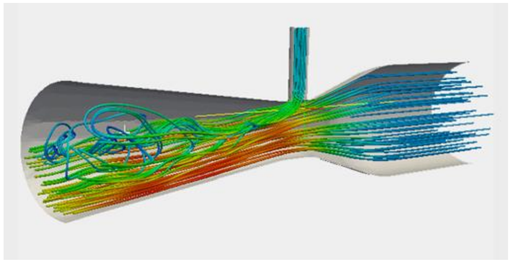


Transformers

- High performance computing



Linear system solver



Fluid dynamics

Physics simulation

- Image and video processing

Correcting illumination

Detecting Edges

Tensor algebra is adopted in a wide range of applications

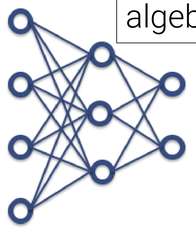
NVIDIA.

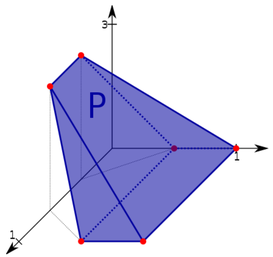
ACCELERATOR DESIGN SPACE EXPLORATION

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

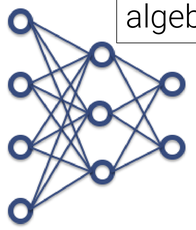
NVIDIA.

ACCELERATOR DESIGN SPACE EXPLORATION

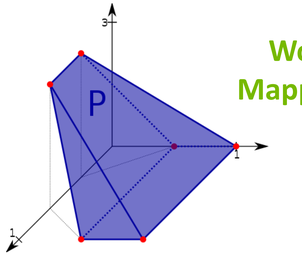
Four key steps

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

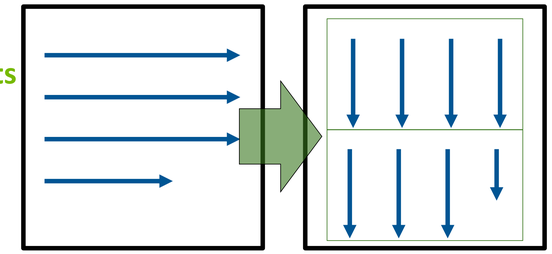
Arch design space

Arch design space

Workload specs
Mapping constraints

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

Mapper



- Tiling facrtors
- Spatial / temporal mapping
- Loop permutation

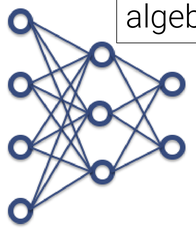
New arch configs

ACCELERATOR DESIGN SPACE EXPLORATION

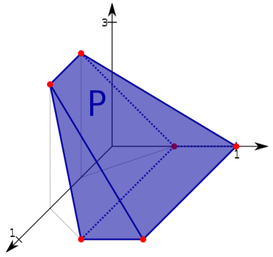
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

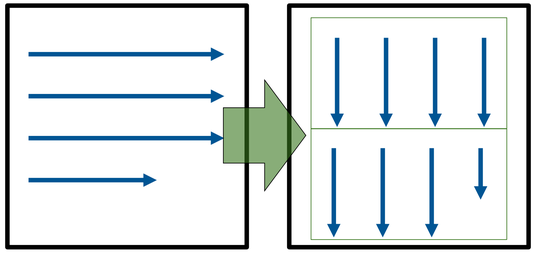
Objectives

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

Workload specs
Arch configs
Objectives
Mapping

STEP #3: Evaluate the performance

Evaluation Tool

- Latency
- Energy
- Area
- ...

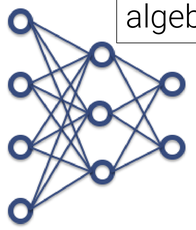
NVIDIA.

ACCELERATOR DESIGN SPACE EXPLORATION

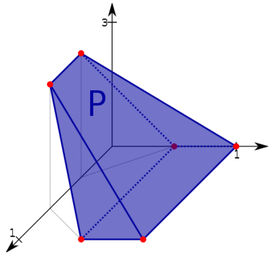
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

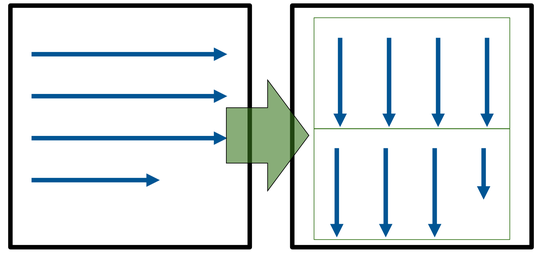
Objectives

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

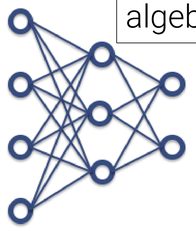
- Latency
- Energy
- Area
- ...

ACCELERATOR DESIGN SPACE EXPLORATION

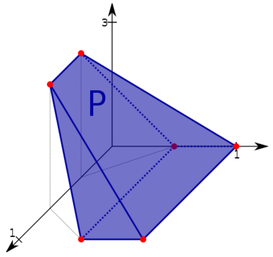
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

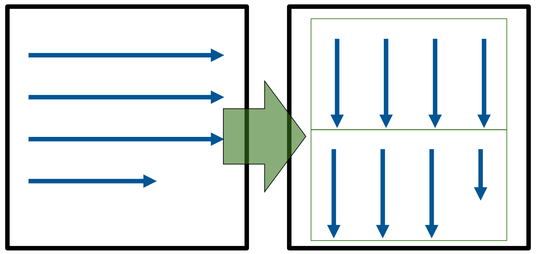
Objectives

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

Workload specs
Arch configs
Objectives
Mapping

STEP #3: Evaluate the performance

Evaluation Tool

- Latency
- Energy
- Area
- ...

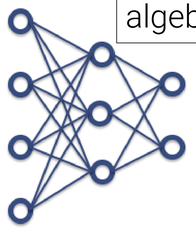
NVIDIA.

ACCELERATOR DESIGN SPACE EXPLORATION

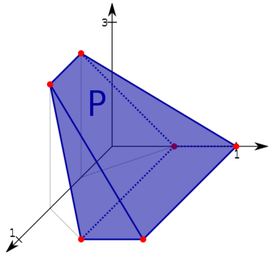
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

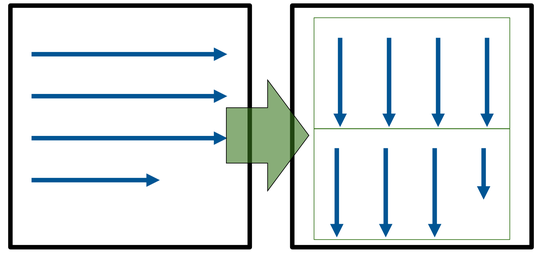
Objectives

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

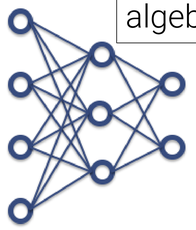
- Latency
- Energy
- Area
- ...

ACCELERATOR DESIGN SPACE EXPLORATION

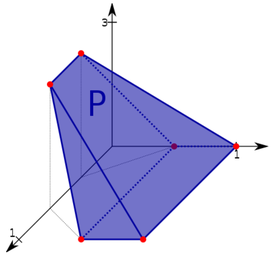
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

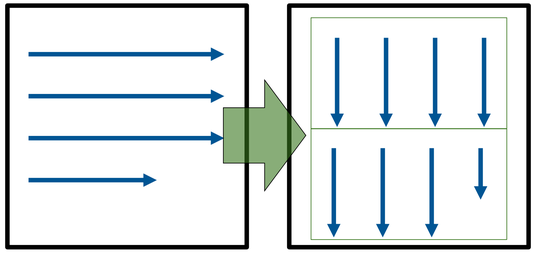
Objectives

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

Workload specs
Arch configs
Mapping

STEP #3: Evaluate the performance

Evaluation Tool

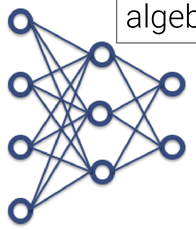
- Latency
- Energy
- Area
- ...

ACCELERATOR DESIGN SPACE EXPLORATION

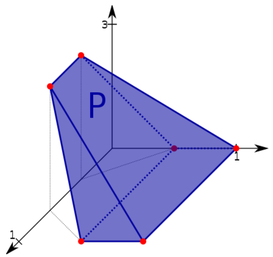
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

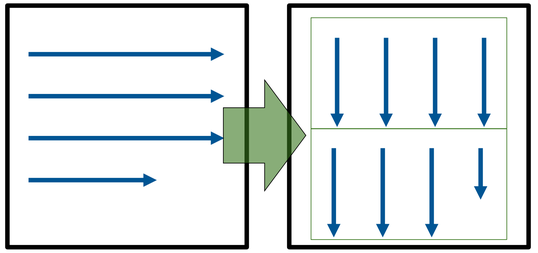
Objectives

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

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
- ...

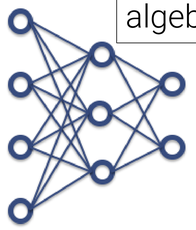
NVIDIA

ACCELERATOR DESIGN SPACE EXPLORATION

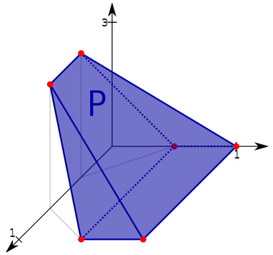
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

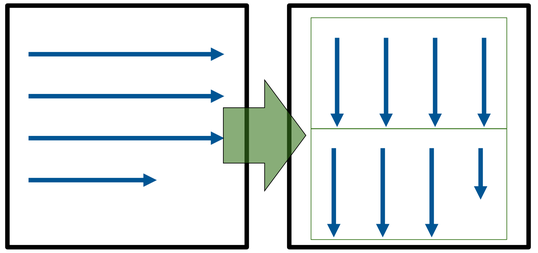
Objectives

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
- ...

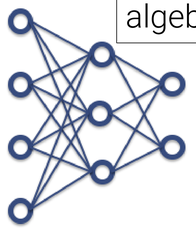
NVIDIA

ACCELERATOR DESIGN SPACE EXPLORATION

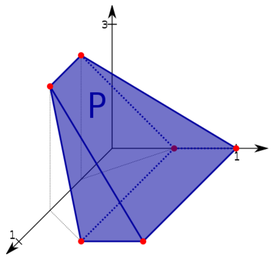
Four key steps

Step #1: Define the design space and objectives

Tensor algebra



Workload specs



Mapping constraints

Metrics
Latency
Energy
Area
EDP
...

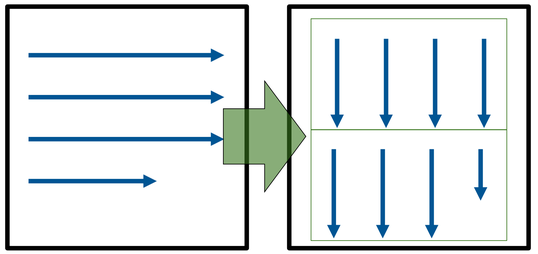
Objectives

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

Performance feedback

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

Search Strategy

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

NVIDIA

KEY CHALLENGES IN DSE

Large design space and costly evaluation

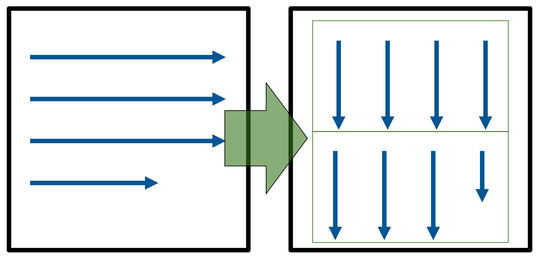
Hardware Design Space

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

NVIDIA.

KEY CHALLENGES IN DSE

Large design space and costly evaluation

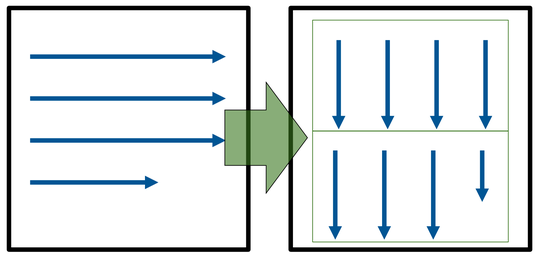
Hardware Design Space

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

$\sim 10^{13}$

×

Mapping Space



$\sim 10^{5}$

×

Evaluation Time

- Latency
- Energy
- Area
- …

0.01s

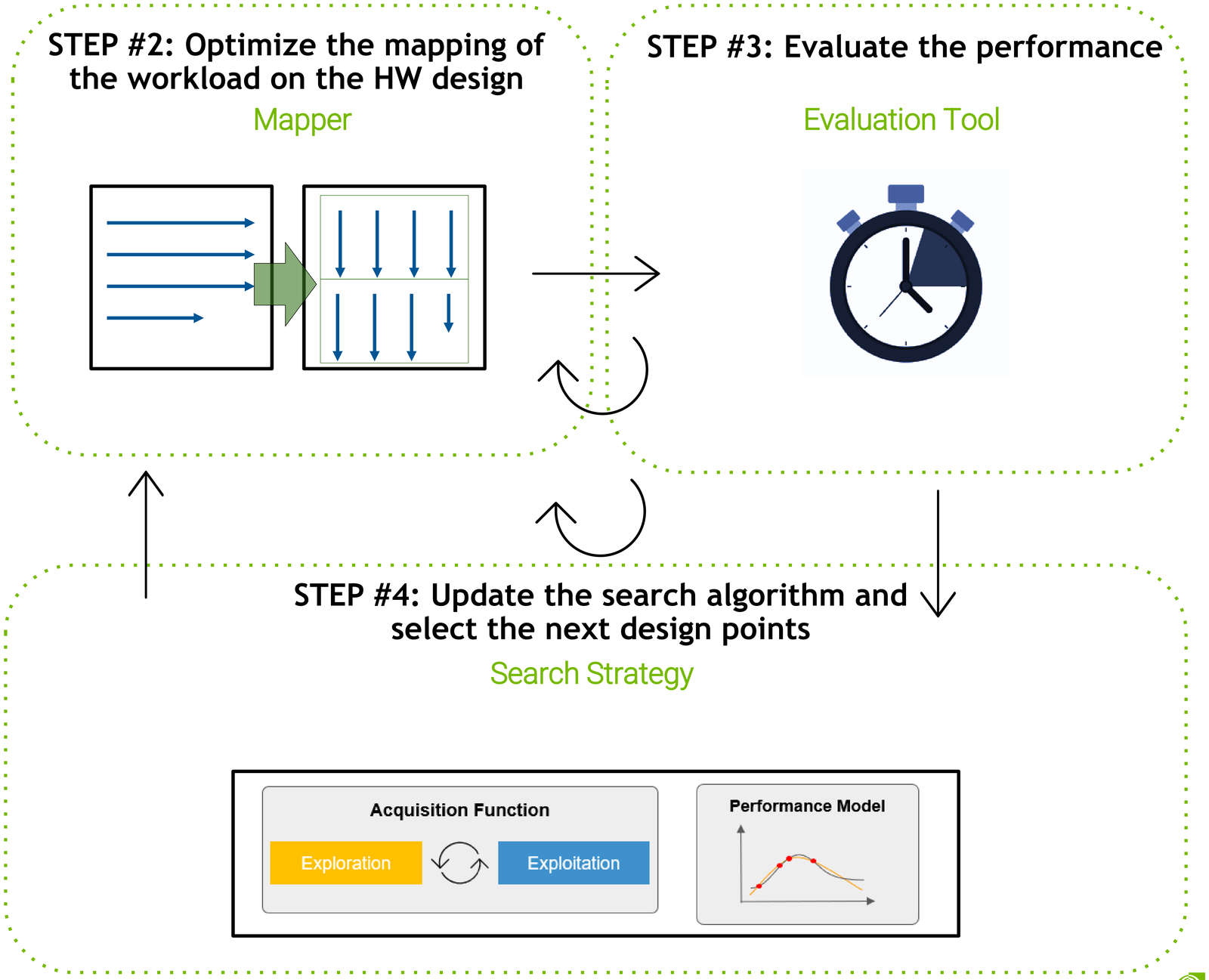
Intractable

> 317M logical years

NVIDIA

DESIGN SPACE EXPLORATION TOOLS

For more systematic and rapid DSE



NVIDIA.

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

NVIDIA.

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

NVIDIA.

A TAXONOMY OF MAPPERS

Heuristic-Driven

Timeloop
Triton Marvel

- Easy to implement

Feedback-based

AutoTVM Ansor
Halide Gamma
MindMapping

- More adaptive

Constrained Optimization

Polly+Pluto TC
Tiramisu CoSA
IOOpt

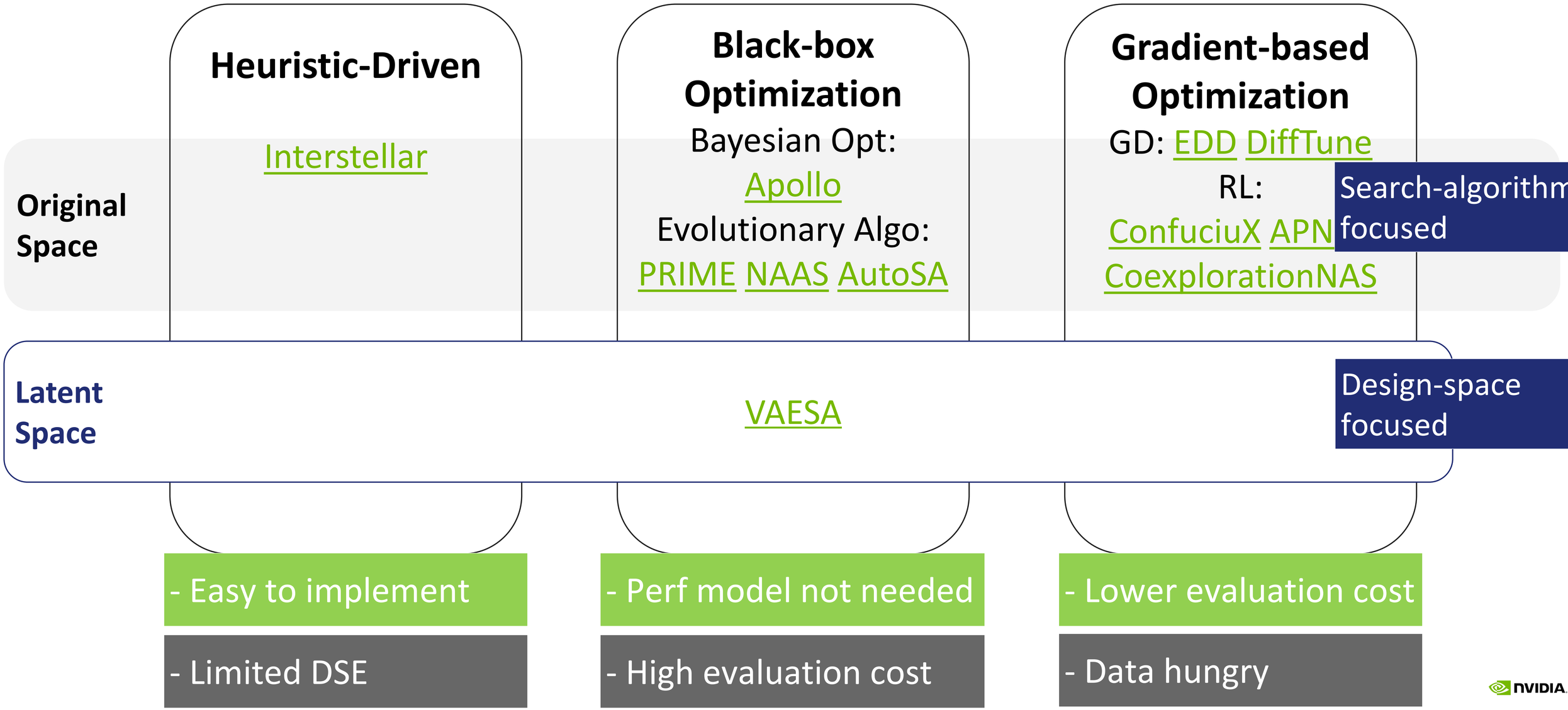
- More sample efficient

- Costly
- Sample invalid space
- Hard to generalize

- Limited use case

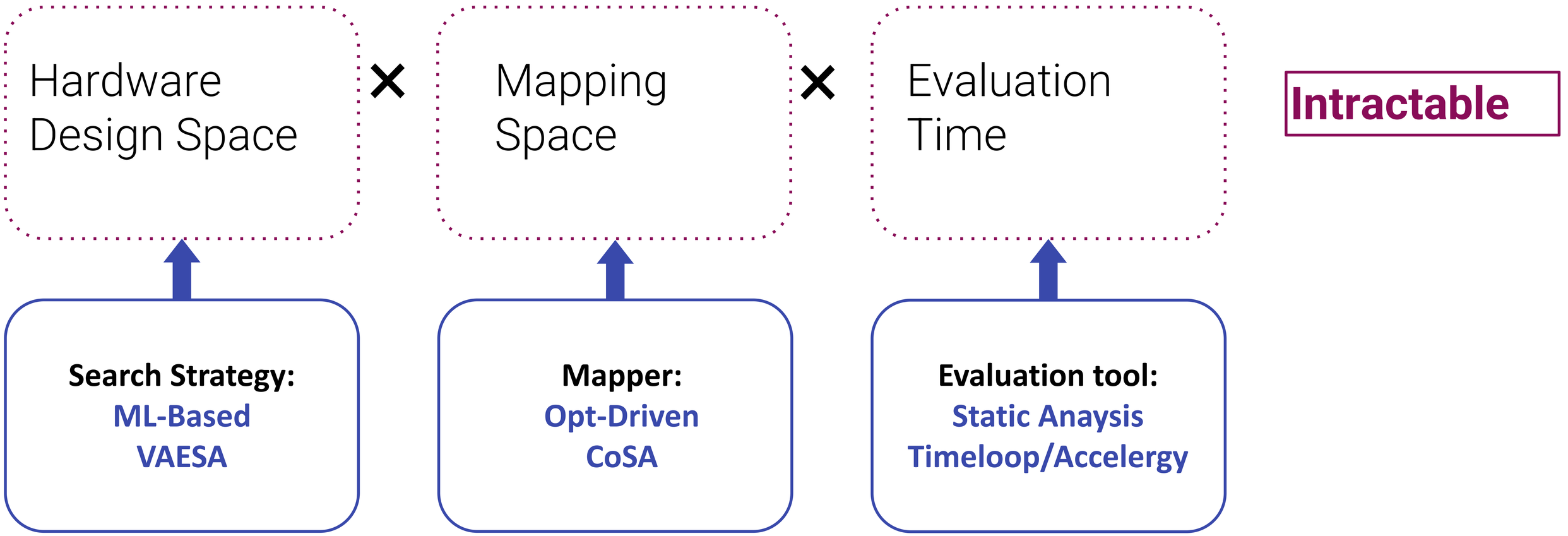
NVIDIA.

A TAXONOMY OF ACCELERATOR SEARCH STRATEGIES



DESIGN SPACE EXPLORATION TOOLS

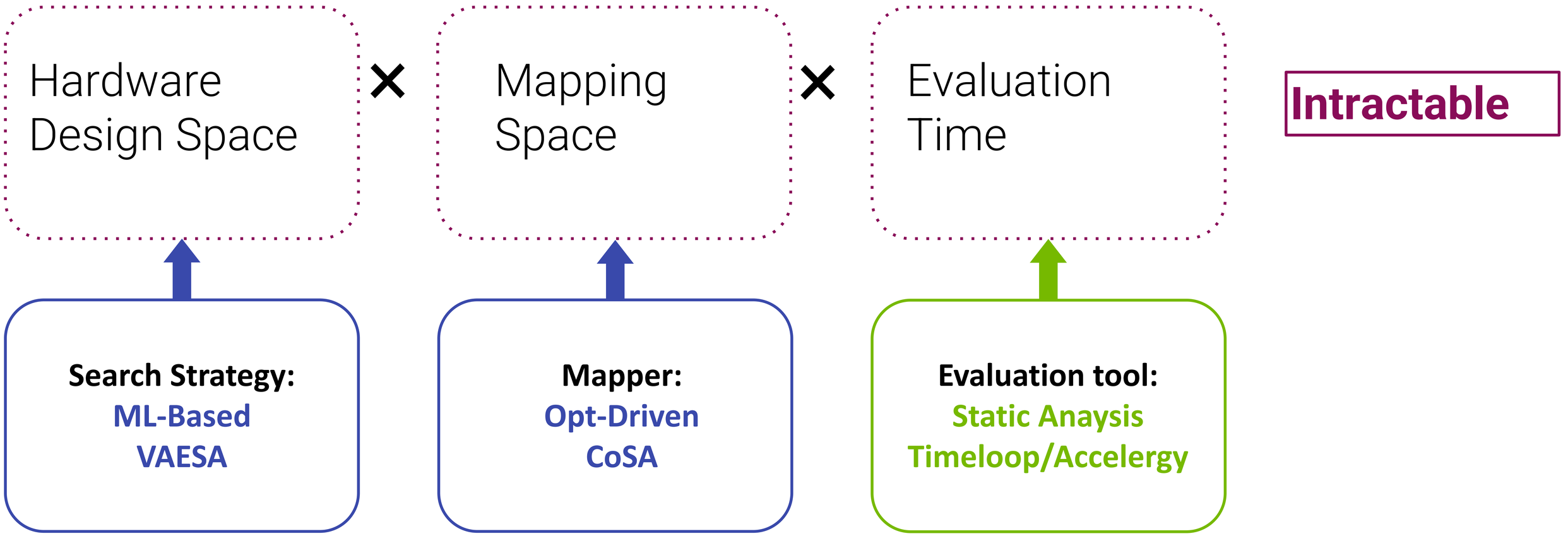
Our approach



NVIDIA.

DESIGN SPACE EXPLORATION TOOLS

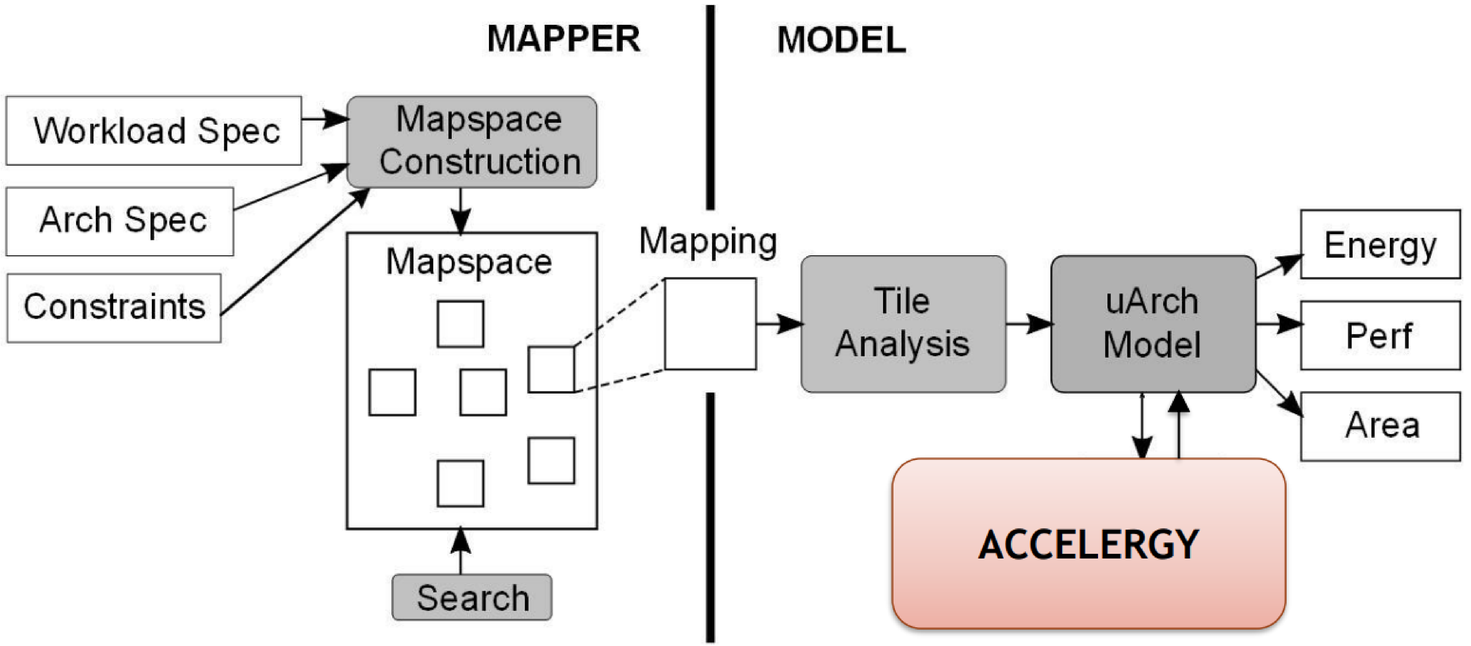
Our approach



NVIDIA.

A STATIC ANALYSIS TOOL

Timeloop/Accelergy



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

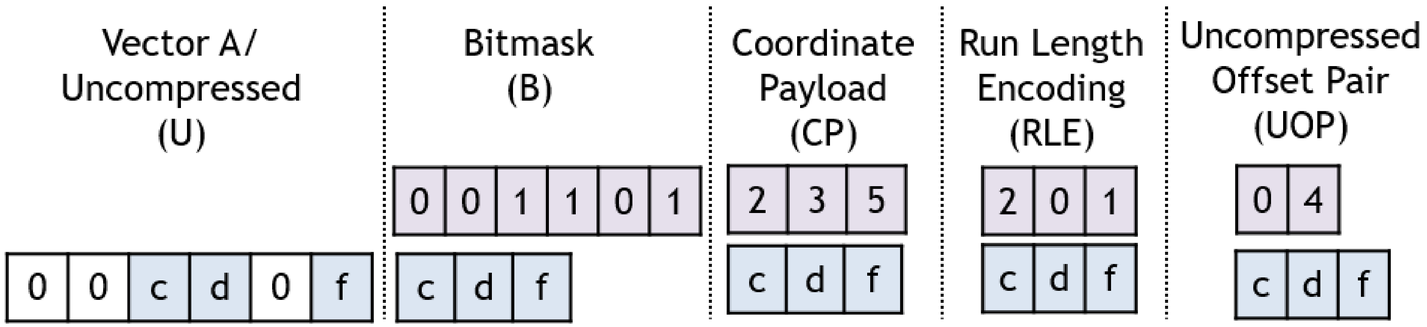
* **Timeloop: A systematic approach to DNN accelerator evaluation**. Parashar A, Raina P, Shao YS, Chen YH, Ying VA, Mukkara A, Venkatesan R, Khailany B, Keckler SW, Emer J. ISPASS'19
* **Accelergy: An architecture-level energy estimation methodology for accelerator designs**. Wu YN, Emer JS, Sze V. ICCAD'19

NVIDIA.

A STATIC ANALYSIS TOOL

Sparseloop

- Sparse tensor algebra
  - Sparsity specification
    - Uniform, Fixed structure, Banded, Real data
  - Compression formats



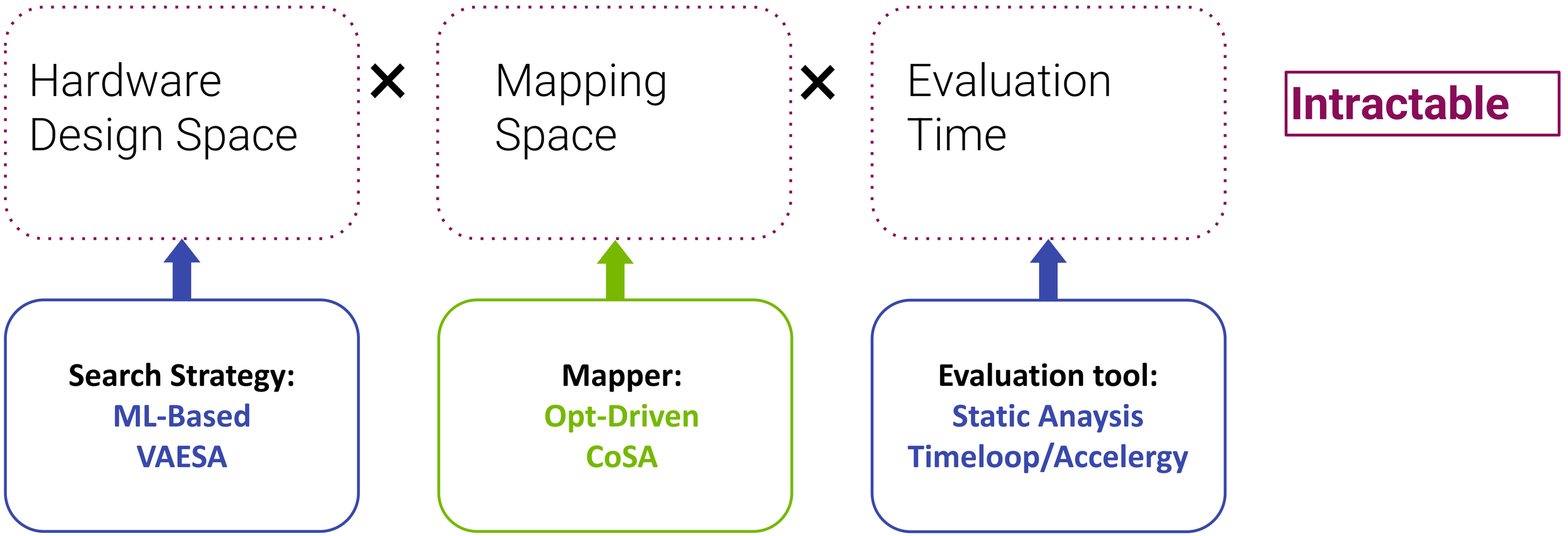
  - Hardware optimizations
    - Compression, Gating, Skipping

*** Sparseloop: An analytical, energy-focused design space exploration methodology for sparse tensor accelerators**. Wu YN, Tsai PA, Parashar A, Sze V, Emer JS. ISPASS'21

NVIDIA.

DESIGN SPACE EXPLORATION TOOLS

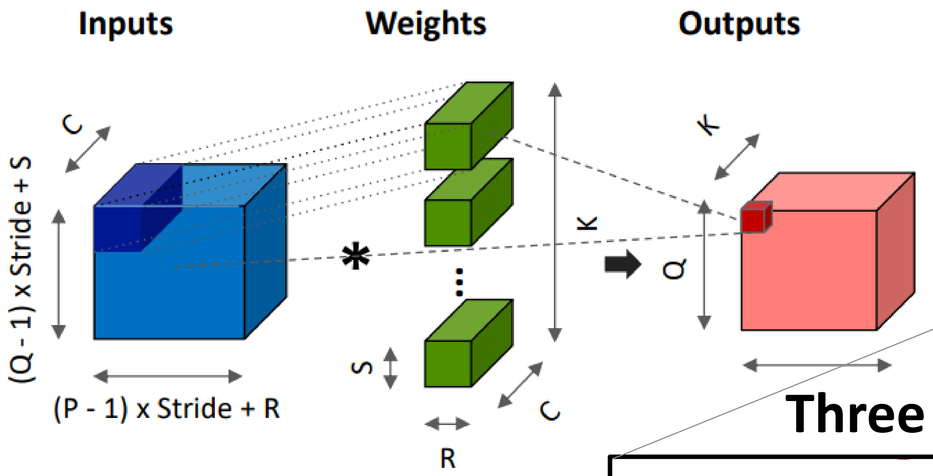
Our approach



NVIDIA.

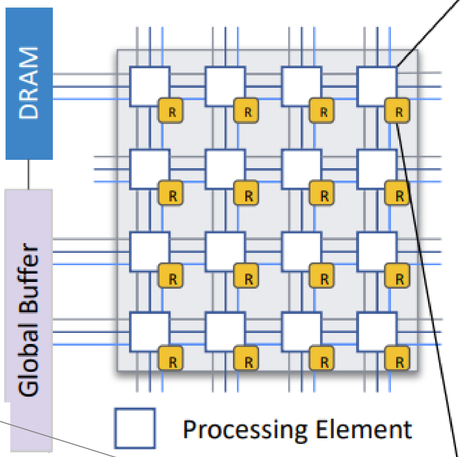
AN OPTIMIZATION-DRIVEN MAPPER

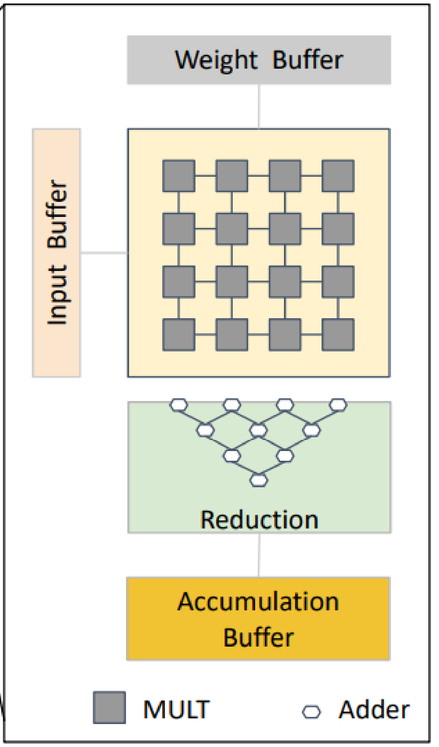
CoSA



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

Mapper





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"

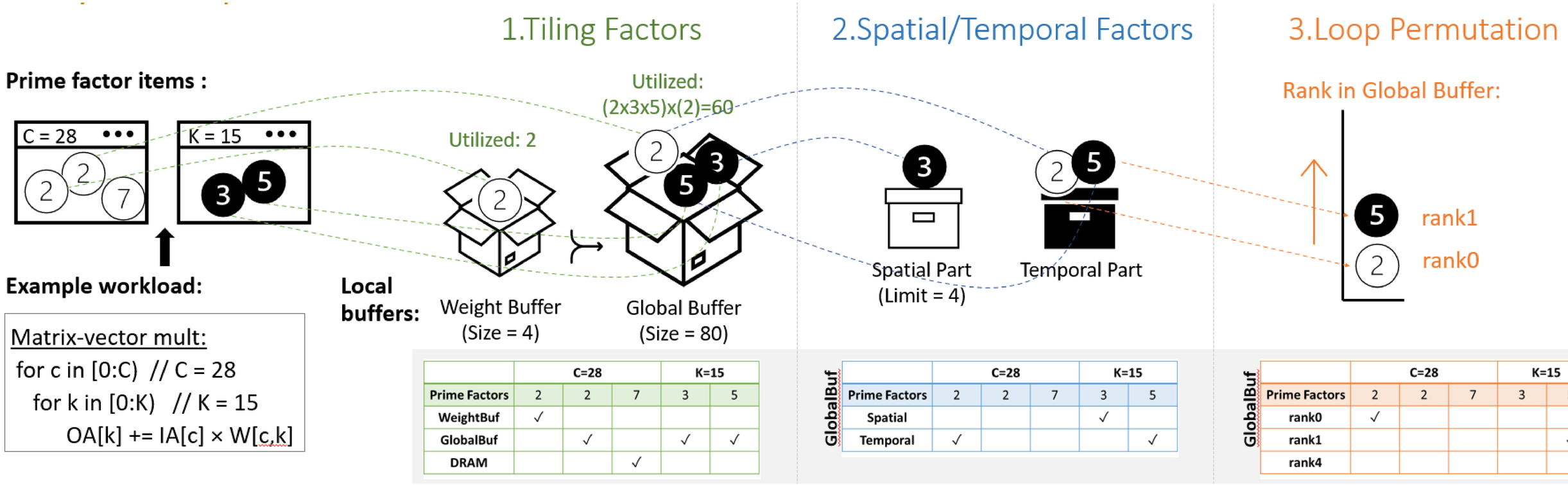
- **CoSA** formulates the mapping decisions into a constrained optimization problem and solves it **in one shot**

*** 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

NVIDIA.

AN OPTIMIZATION-DRIVEN MAPPER

Key idea: tiling factor allocation



	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

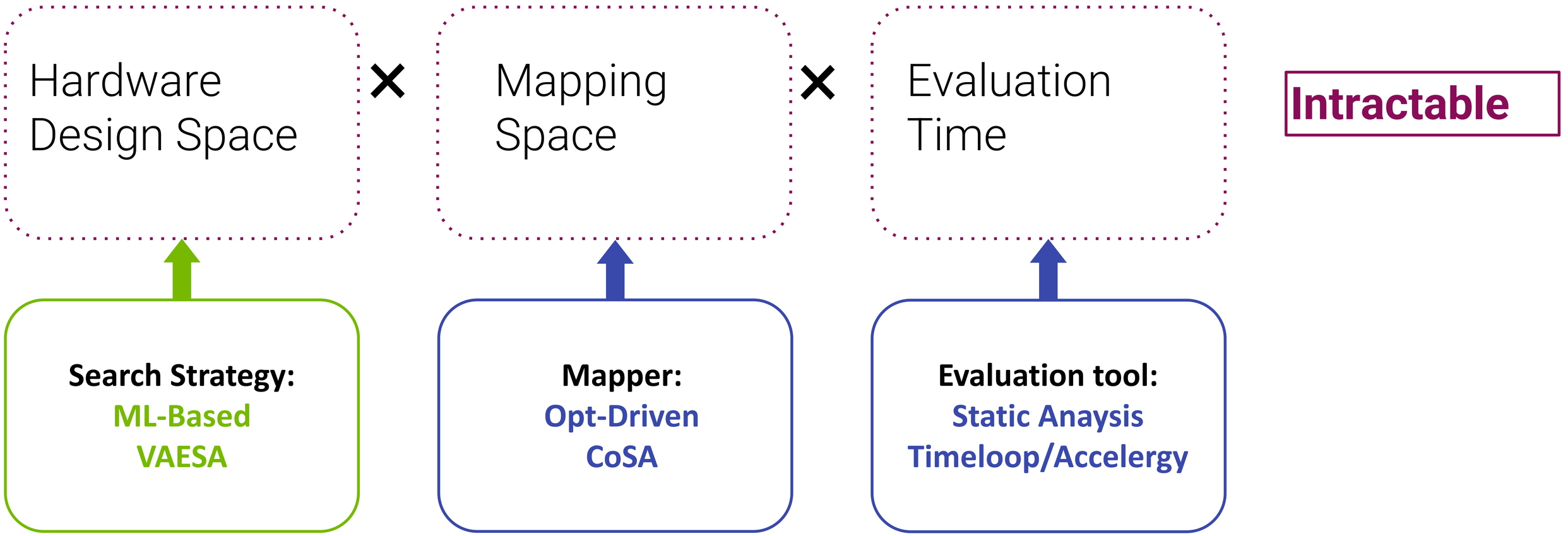
	C=28			K=15	
Prime Factors	2	2	7	3	5
rank0	✓				
rank1					✓
rank4					

- **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**

NVIDIA.

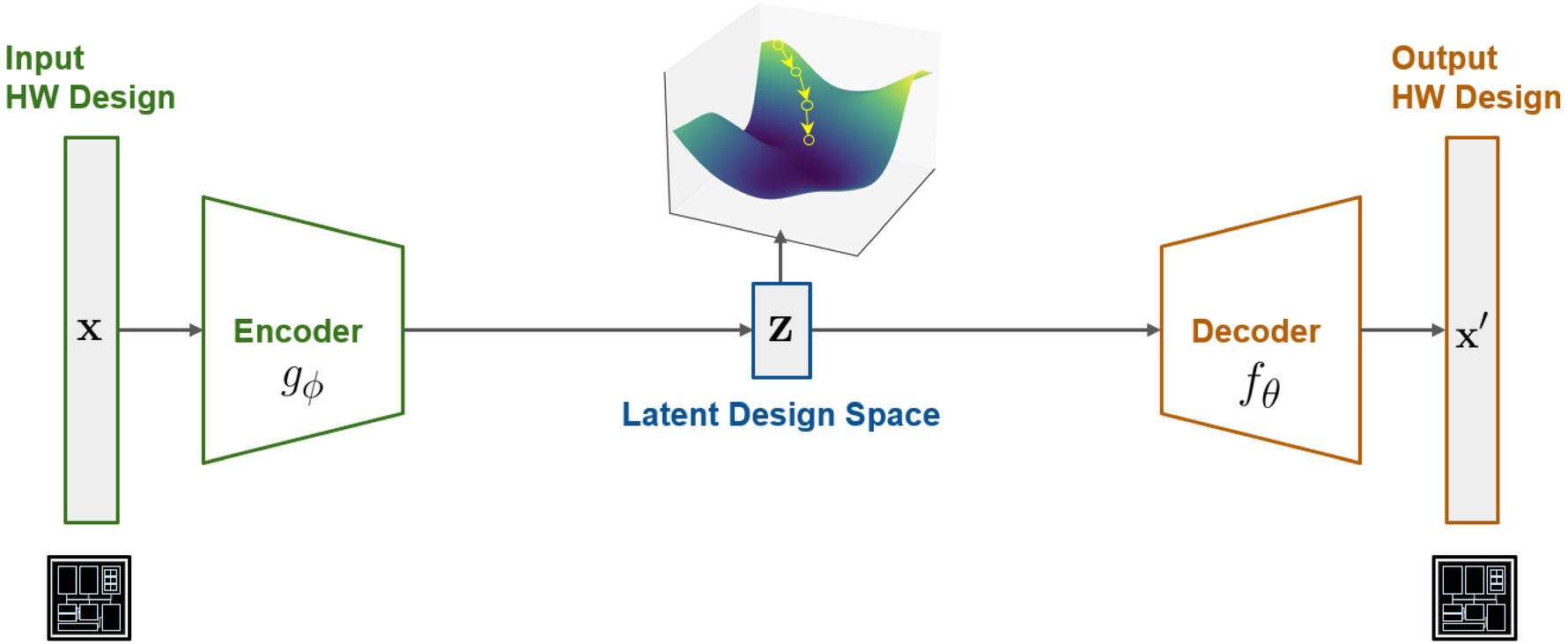
DESIGN SPACE EXPLORATION TOOLS

Our approach



A ML-BASED SEARCH STRATEGY

VAESA



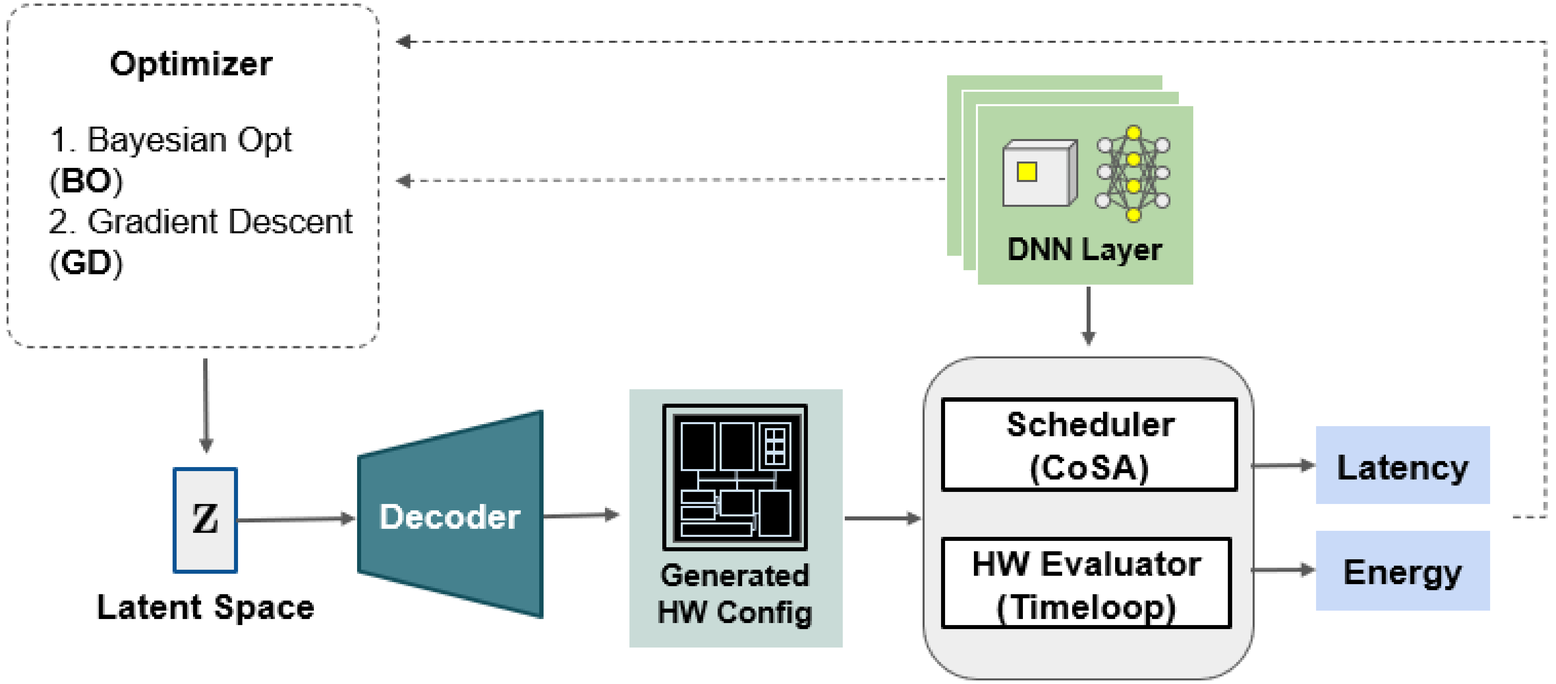
- **VAESA** learns a **low dimensional, continuous, reconstructible** latent space to facilitate accelerator DSE using **Variational Autoencoder (VAE)**

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

NVIDIA.

A ML-BASED SEARCH STRATEGY

VAESA Inference

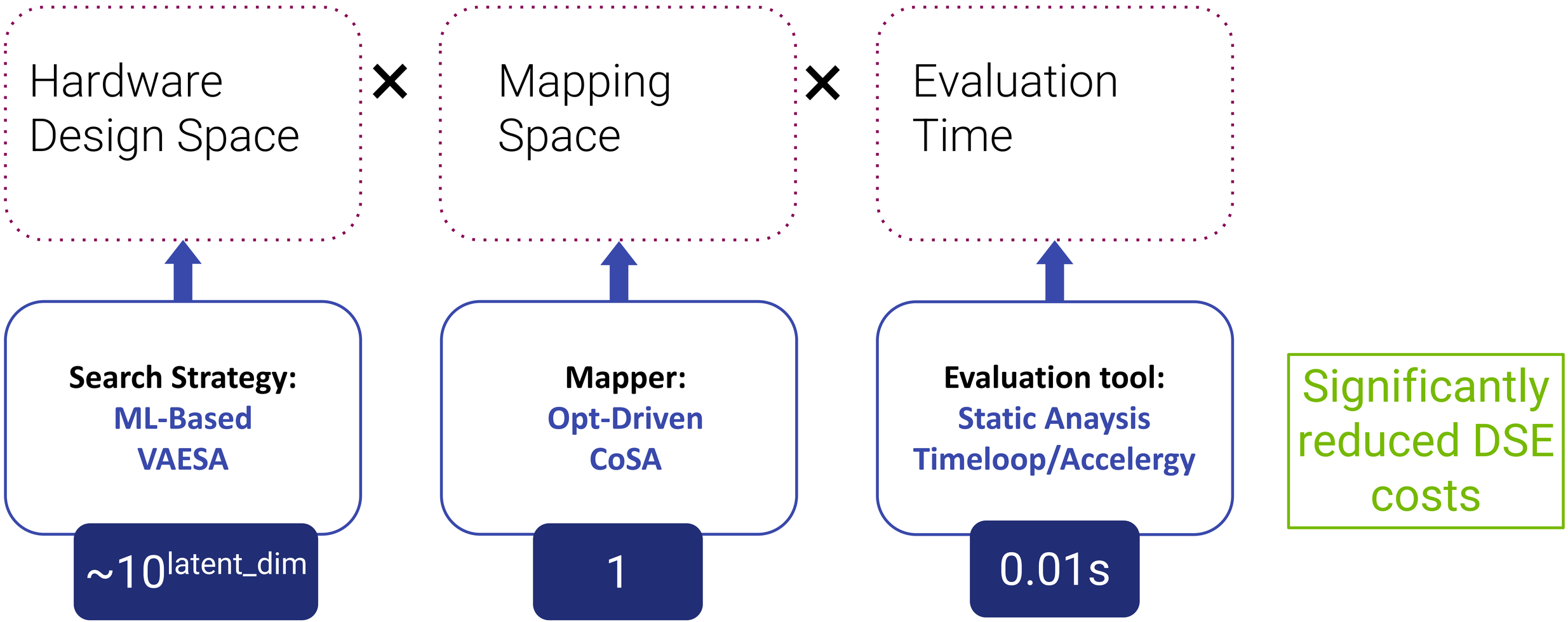


- The search algorithms are applied to the latent space and evaluated on the original search
- The latent space **reduces the search complexity** and provides a **smoother performance surface**
- Both BO and GD **achieve better sample efficiency** on the latent space

NVIDIA.

DESIGN SPACE EXPLORATION TOOLS

Our approach



NVIDIA.

OPEN CHALLENGES AND OPPORTUNITIES

What shall we work on next?

#1 Flexibility of workloads

- irregular
- input-dependent
- multi-tenant

- Leverage the statistical info and profiling traces

#2 Dynamic system components

- SW/OS schedulers
- Caching/paging

- Augment ML with analytical model to provide feedback

NVIDIA.

OPEN CHALLENGES AND OPPORTUNITIES

What shall we work on next?

#3 Limited HW design space and execution models

- Lack of customization vs programmability tradeoffs
- SoC with cpu, vector, tensor units

- Design extensible hardware design abstraction

#4 Transferability under new constraints

- Repeated DSE runs

- Collect datasets of hardware design

NVIDIA.

THANK YOU

QIJING JENNY HUANG, NVIDIA
jennyhuang@nvidia.com