香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

MACO: A HW-Mapping Co-optimization Framework for DNN Accelerators

Speaker: Wujie Zhong

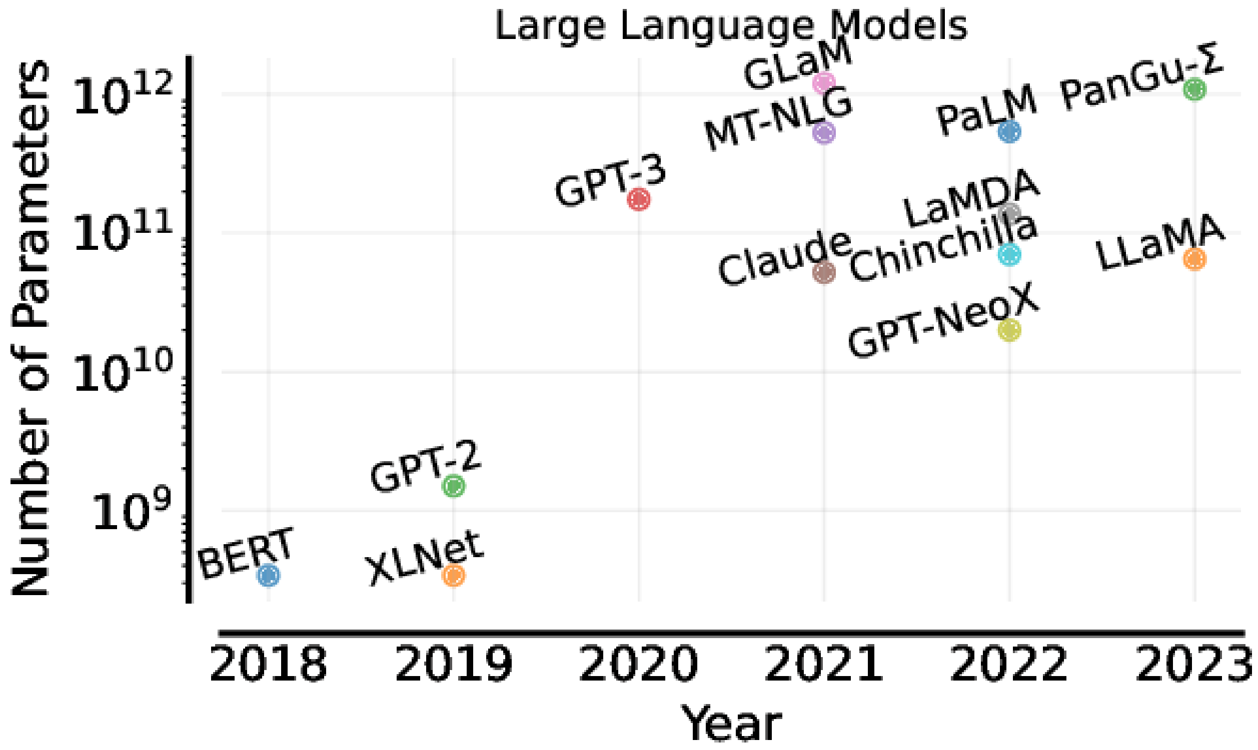
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

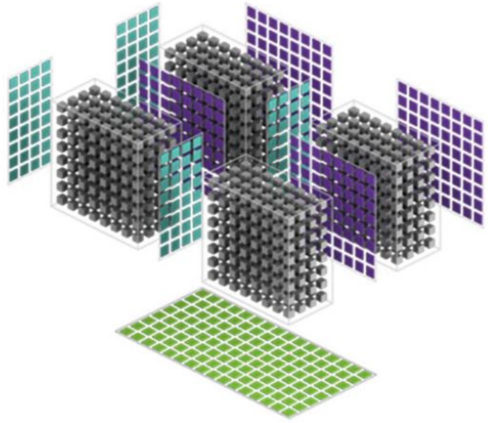
Catalogue

- Introduction
- Related Works
- MACO
- Experiment
- Conclusion

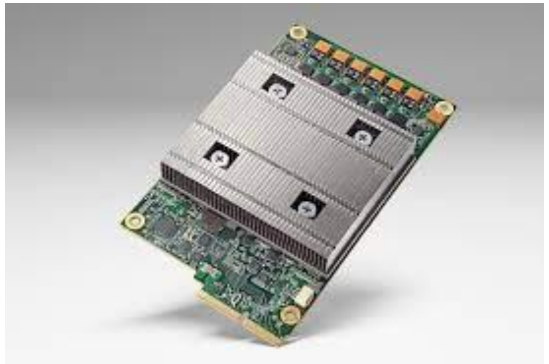
Introduction

- DNN accelerators





GPU
Tensor Core



TPU

Introduction

- Design Space Exploration
  - The capacity of data buffers
  - The number of PEs
  - The number of MACs
  - Loop boundaries
  - Loop order

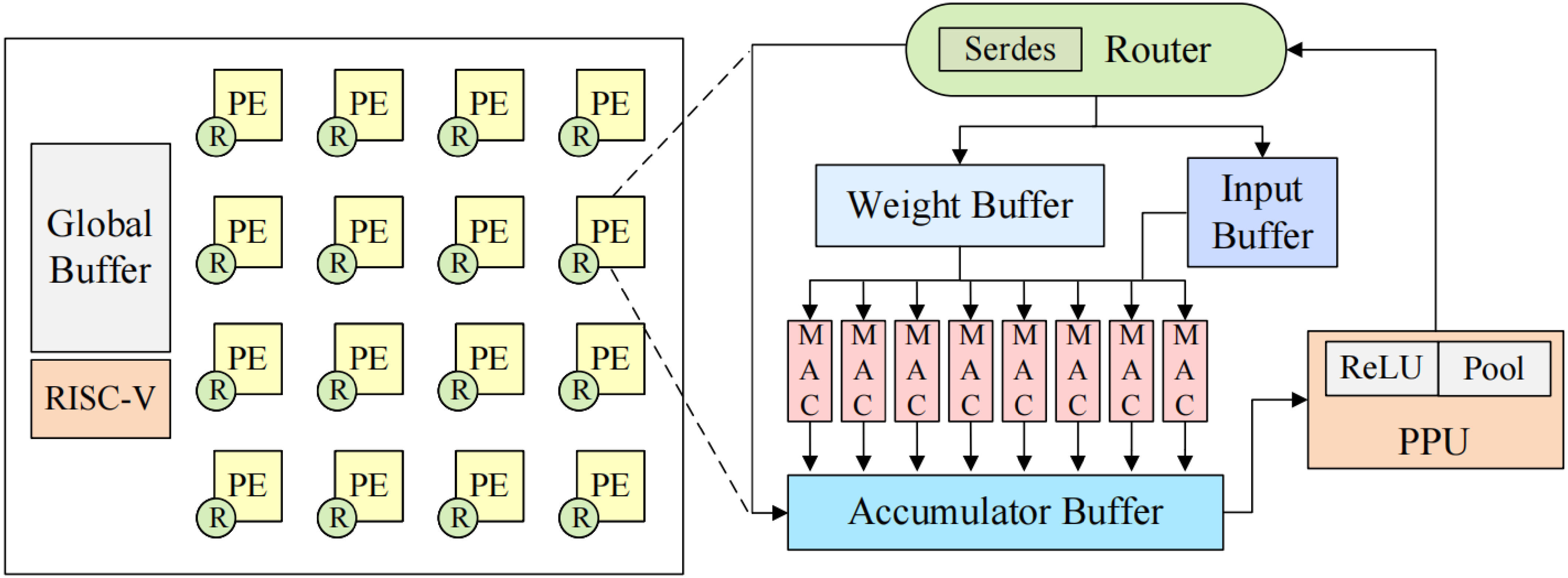
Hardware Space

Mapping Space

- Tradeoff between power, performance and area (PPA)

Introduction

- Hardware Space Exploration



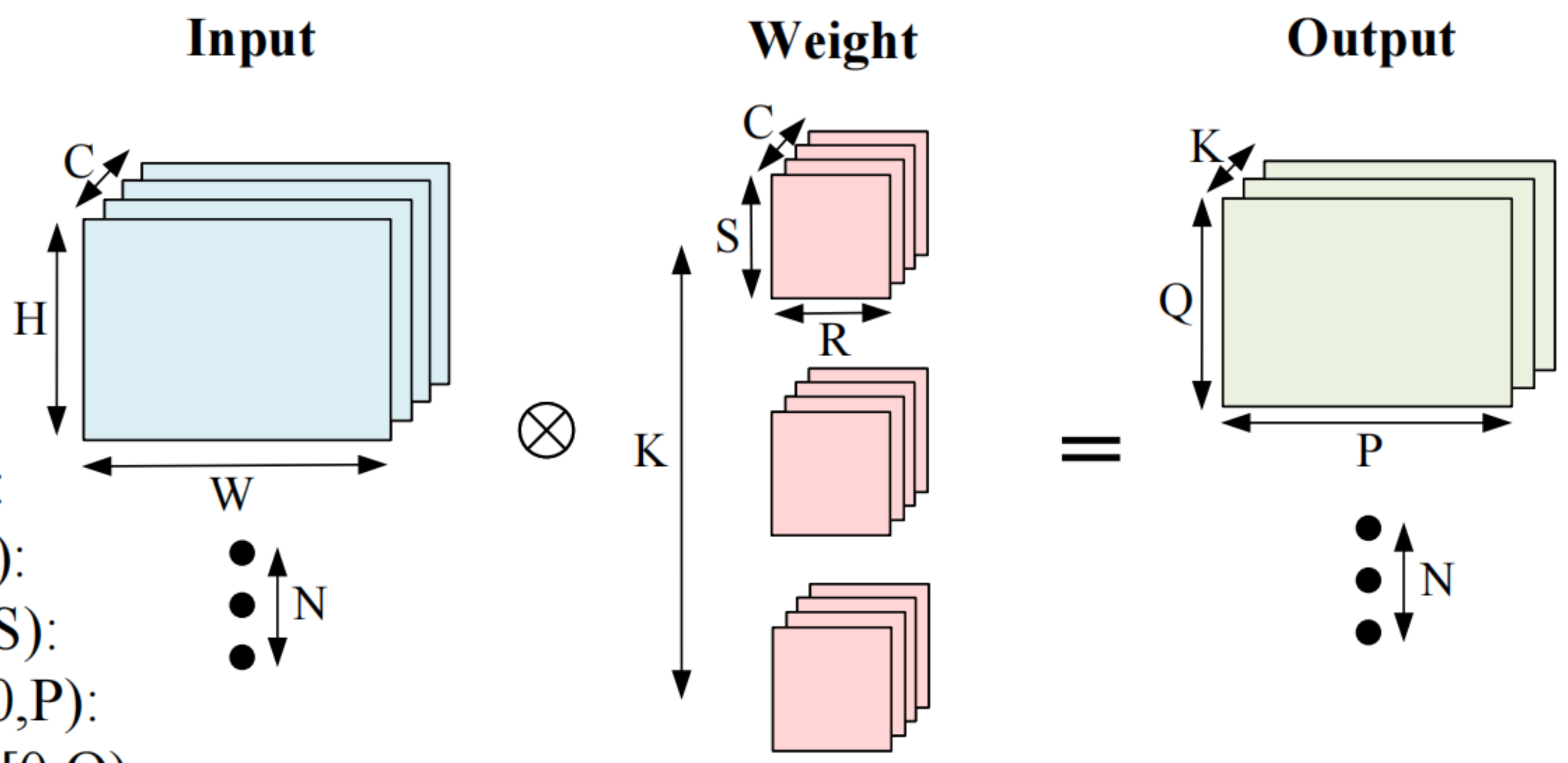
The architecture of a CNN accelerator: Simba [MICRO'19]

Introduction

- Hardware Space Exploration
  - Explore the hardware parameters
  - Computation bound
    - More PEs or more MACs
  - Memory bound
    - Higher bandwidth and larger buffers

Introduction

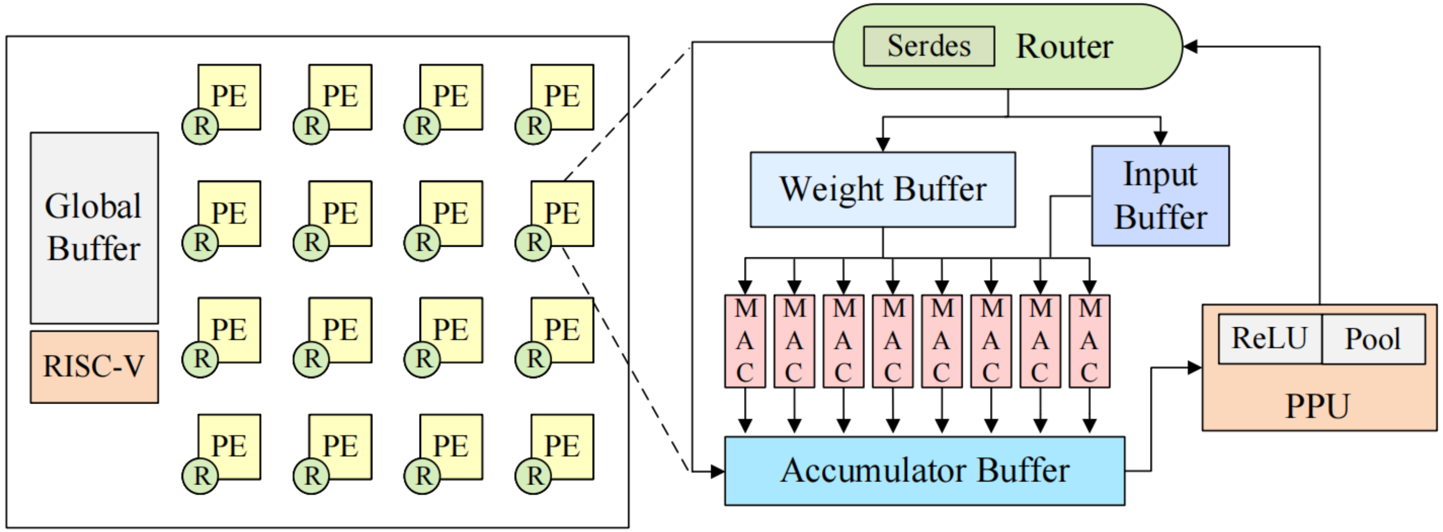
- Mapping Space Exploration
  - Loop nest




```
for n in [0,N):
  for c in [0,C):
    for k in [0,K):
      for r in [0,R):
        for s in [0,S):
          for p in [0,P):
            for q in [0,Q):
              Output[n][k][p][q] += Weight[k][c][r][s] * Input[n][c][p+r][q+s];
```

Introduction

- Mapping Space Exploration
  - Six Memory Levels
  - L0: PE Weight Register Level
  - L1: PE Accumulator Buffer Level
  - L2: PE Weight Buffer Level
  - L3: PE Input Buffer Level
  - L4: Global Buffer Level
  - L5: DRAM Level



Introduction

- Mapping Space Exploration

```
# L2: PE Weight Buffer Level
for k2 in [0,4):                          ← sequential loop
    for s2 in [0,3):
        for r2 in [0,3):
            for c2 in [0,4):
# L1: PE Accumulator Buffer Level
for q1 in [0,7):
    for p1 in [0,14):
        spatial for c1 in [0,8):
# L0: PE Weight Register Level
for n0 in [0,1):
    Output[n][k][p][q]+= Weight[k][c][r][s] * Input[n][c][p+r][q+s]
```

A part of an example about mapping a convolution layer into a Simba-like chiplet

Introduction

- Mapping Space Exploration
  - Buffer Capacity Constraint

PE Weight Buffer:

$$K2 * S2 * R2 * C2 \leq Buf_w$$

```
# L2: PE Weight Buffer Level
for k2 in [0,4):
    for s2 in [0,3):
        for r2 in [0,3):
            for c2 in [0,4):
# L1: PE Accumulator Buffer Level
for q1 in [0,7):
    for p1 in [0,14):
        spatial for c1 in [0,8):
# L0: PE Weight Register Level
for n0 in [0,1):
```

Related Works

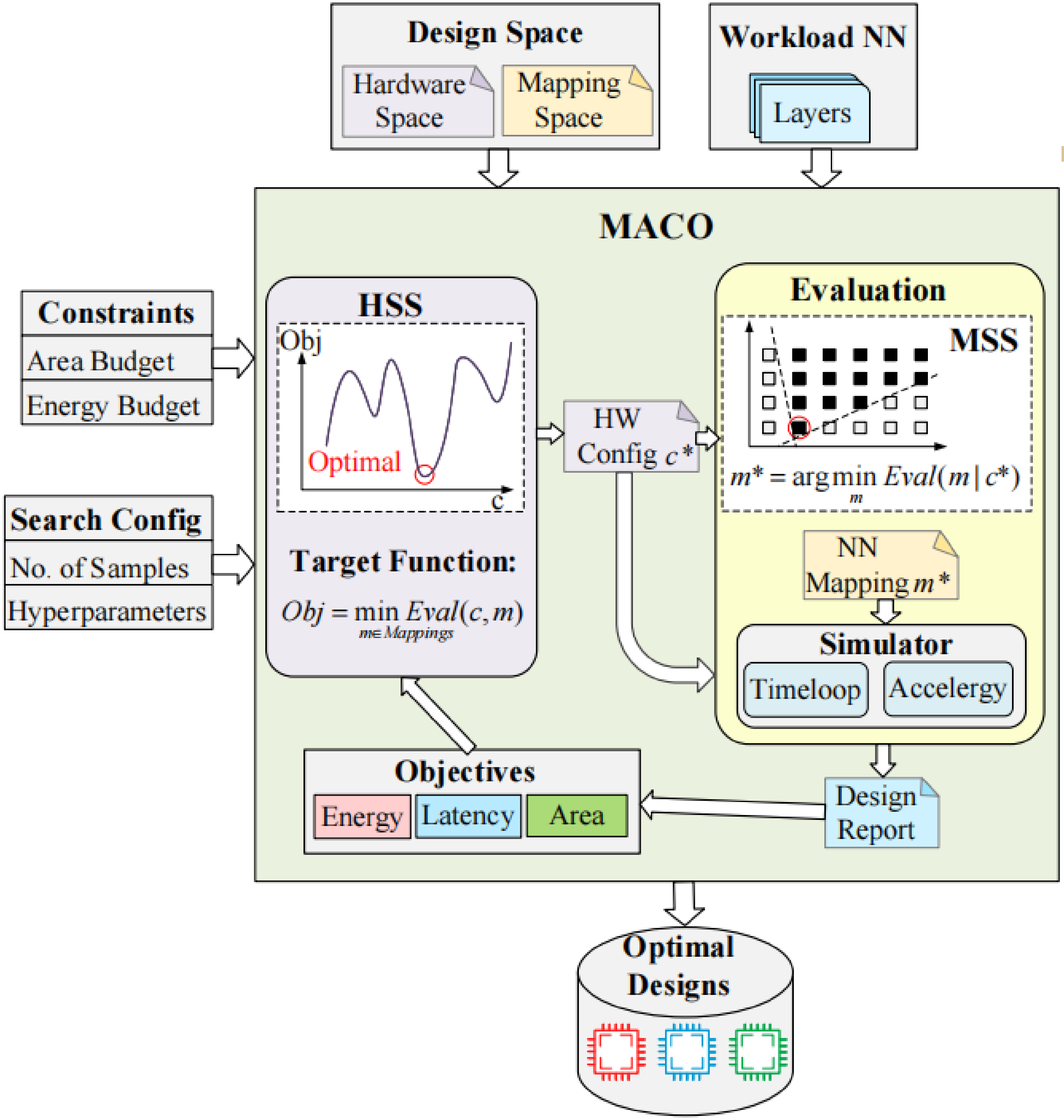
- Mapping Space Exploration
  - Timeloop [ISPASS'19]: exhaustive and random search
    - Challenge: huge design space
  - GAMMA [ICCAD'20]: genetic algorithm
  - CoSA [ISCA'21]: Mixed Integer Programming (MIP)
  - LEMON [CF'23]: Mixed Integer Programming (MIP)

Related Works

- Hardware-Mapping Co-optimization
  - DiGamma [DATE'22]: genetic algorithm
    - Target on a two-level memory hardware
  - DOSA [MICRO'23]: gradient-based methods
    - Target on a single objective
  - MEDEA [DATE'22]: genetic algorithm
    - Suboptimal solutions

MACO

- Overview



MACO

- Hardware Space Search Block
  - Multi-objective Bayesian optimization (MOBO)

$$Obj(c) = \min_{m \epsilon Mappings} Eval(c, m)$$

- Evaluation Block
  - MIP solver: LEMON [1]

$$m^* = \arg\ \min\ Eval(m|c^*)$$

[1] Memory-Aware DNN Algorithm-Hardware Mapping via Integer Linear Programming, Computing Frontiers, 2023

MACO

- Hardware Design Space
  - Simba accelerator
  - 2.9 million parameters

Components	Depth	Number	Space Size
Global Buffer	512:3584:128	1	24
Input Buffer	128:1920:64	16	28
Weight Buffer	32:992:32	128	30
Acc Buffer	32:224:16	128	12
MAC	-	512:2048:128	12

MACO

- Hardware Sparce Search Algorithm

Algorithm 1 Pseudo-code for the MOBO Algorithm

Input: hardware design space C, the number of iterations N
Output: optimal configuration and objective pairs (C^*, O^*)

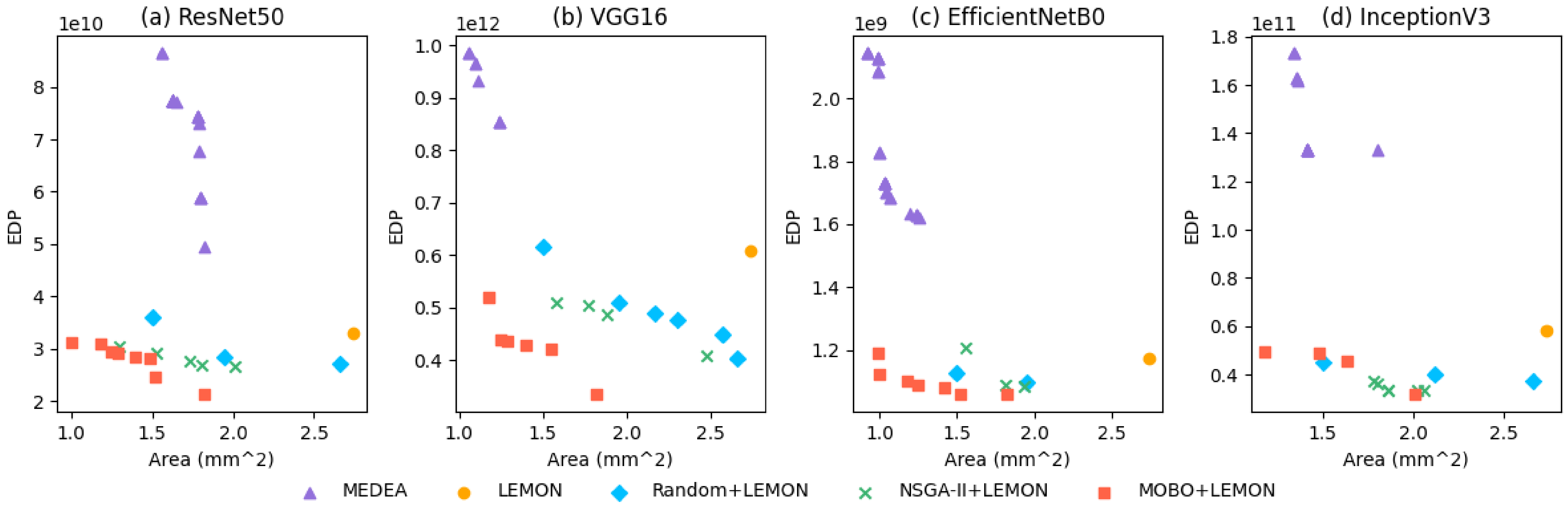
Sample initial dataset D from the design space C.
for i **from** 0 **to** N **do**
▷ Exploration Stage
  Fit the Gaussian Process model GP using dataset D.
  Update the posterior distribution $p(o|c, D)$ on GP.
  Sample dataset (Cs, Os) using the acquisition function $Acq(p(o|c, D))$.
  Get the Pareto optimal set (Cp, Op) from (Cs, Os).
  Randomly select a value c^* from Cp as the next hardware design choice.
▷ Evaluation Stage
  Get the optimal mappings m^* from the MSS block for c^*.
  Calculate the objectives o^* using the evaluation tools with the hardware configuration c^* and the mapping strategy m^*.
  Update dataset D: $D \leftarrow D \cup (c^*, o^*)$.
end for
Calculate the Pareto optimal set (C^*, O^*) from D.
Return (C^*, O^*).

Experiment

- Setup
  - DNN Models:
    - ResNet50, VGG16, EffienetNetB0, InceptionV3
  - Simulators
    - Timeloop [ISPASS'19] and Accelergy [ICCAD'19]
  - Search algorithms
    - MEDEA [DATE'22]
    - LEMON [CF'23]
    - Random + LEMON, NSGA-II + LEMON and MOBO + LEMON

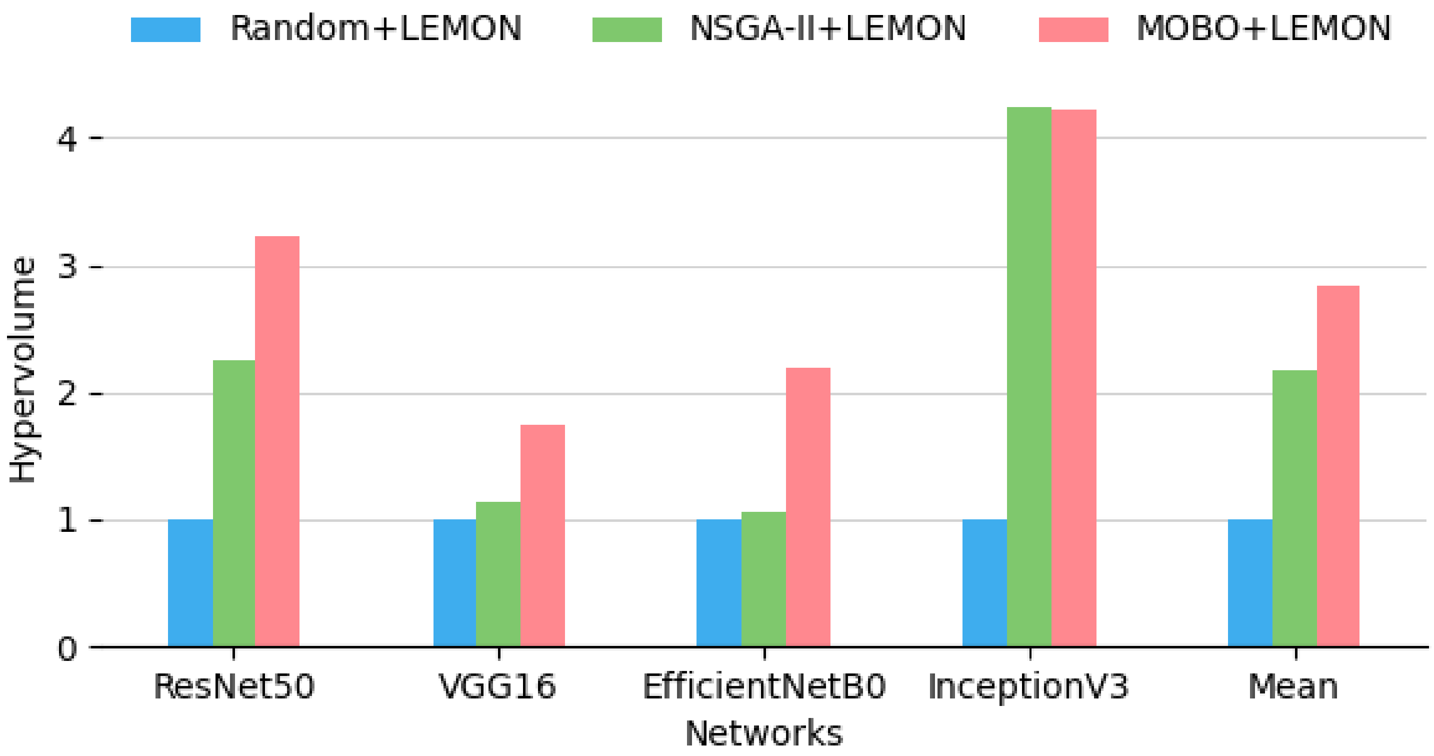
Experiment

Pareto Front



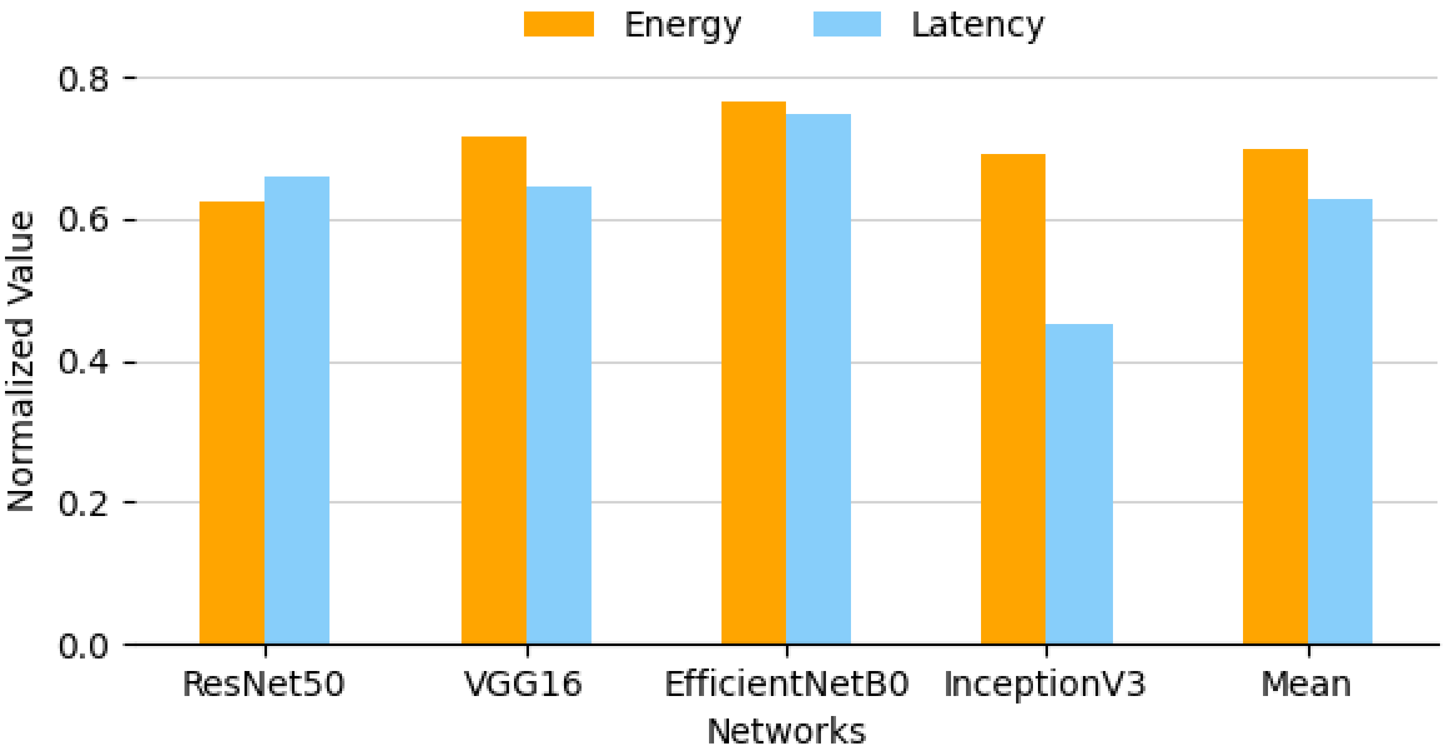
Experiment

- Hypervolume
  - 2.84× speedup over Random + LEMON
  - 1.3× speedup over NSGA-II + LEMON



Experiment

- Compared to MEDEA (same area)
  - 30% reduction in energy consumption
  - 37% reduction in latency



Conclusion

- MACO: A HW-Mapping Co-optimization Framework for DNN Accelerators
  - Hardware Space: Multi-objective Bayesian optimization
  - Mapping Space: MIP model: LEMON
  - Result: better PPA metrics

Thanks