H4H: Hybrid Convolution-Transformer Architecture Search for NPU-CIM Heterogeneous Systems for AR/VR Applications

Yiwei Zhao

Jinhui Chen, Sai Qian Zhang, Syed Shakib Sarwar, Kleber Stangherlin, Jorge Gomez, Jae-Sun Seo, Barbara De Salvo,Chiao Liu, Phil Gibbons, Ziyun Li

Carnegie Mellon University; Meta Reality Labs

Overview

- Background: AR/VR devices.
- Edge AI/ML Accelerations: NPU and CIM.
- Our Automated Workflow: H4H-NAS.
- Experimental Evaluations.

Overview

- **Background: AR/VR devices.**
- Edge AI/ML Accelerations: NPU and CIM.
- Our Automated Workflow: H4H-NAS.
- Experimental Evaluations.

AR/VR Devices

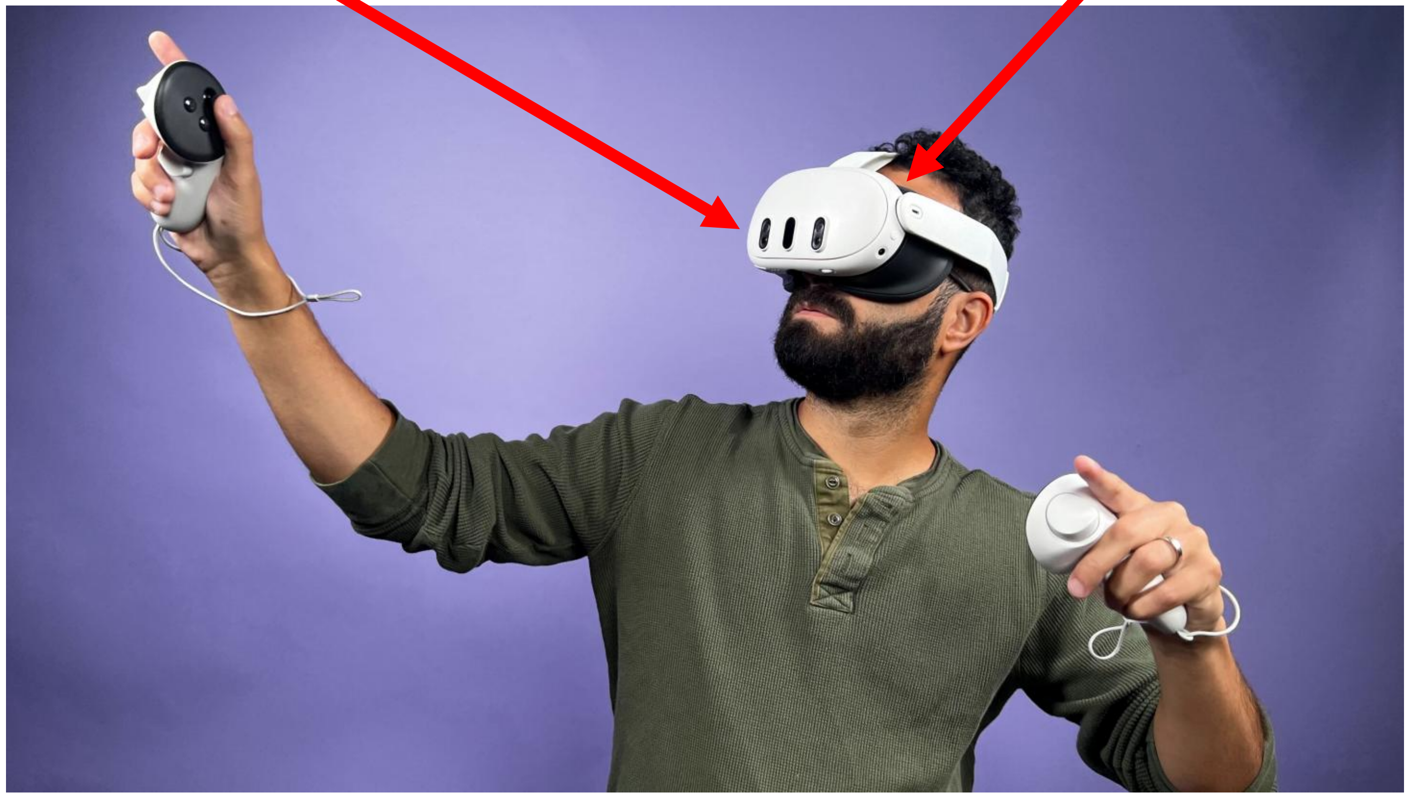
- AR/VR: Next generation human-oriented computing.
- Heavy on AI/ML-driven vision tasks.

AR/VR Devices

- AR/VR workloads are latency-sensitive !!!
- 20ms latency constraint

latency $\leq$ 20ms

Input stream **Output stream**

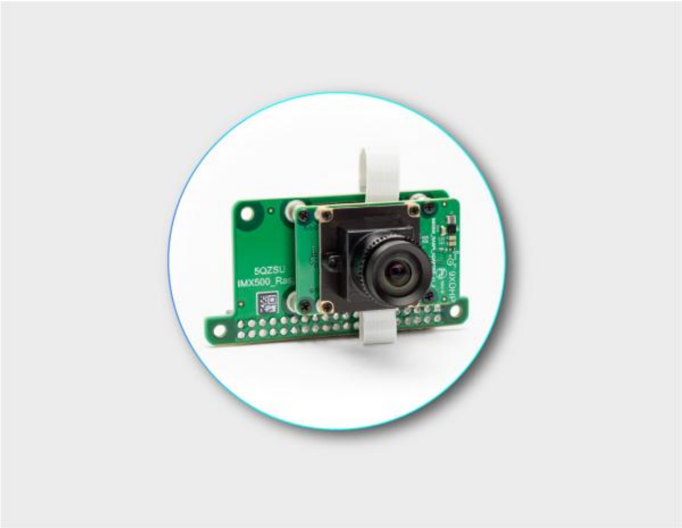


AR/VR Devices

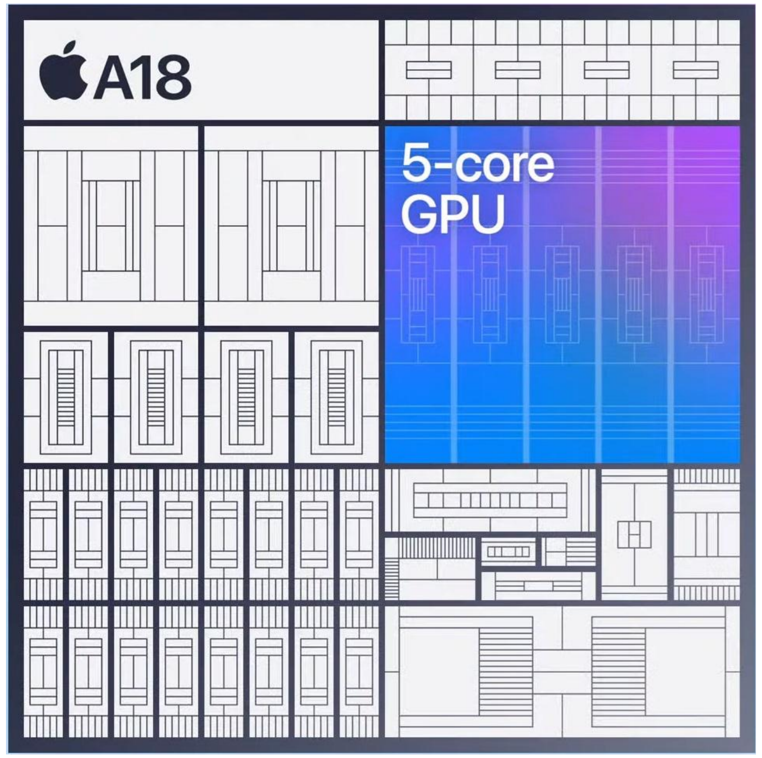
- AR/VR workloads are latency-sensitive !!!
- AR/VR devices are energy-bounded.

AR/VR Designs

Sony IMX500
Avg <1W [1]



Cell Phone Designs



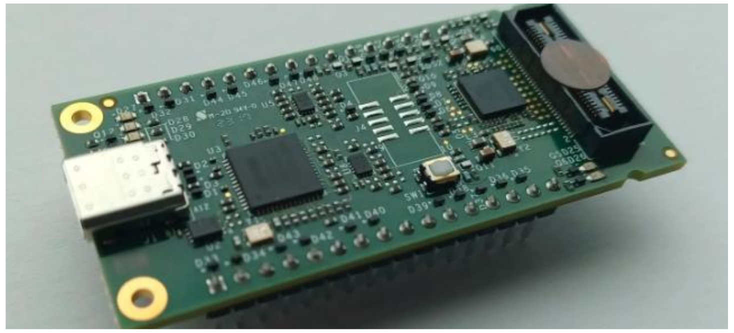
A18 in iPhone 16
Avg ~10W

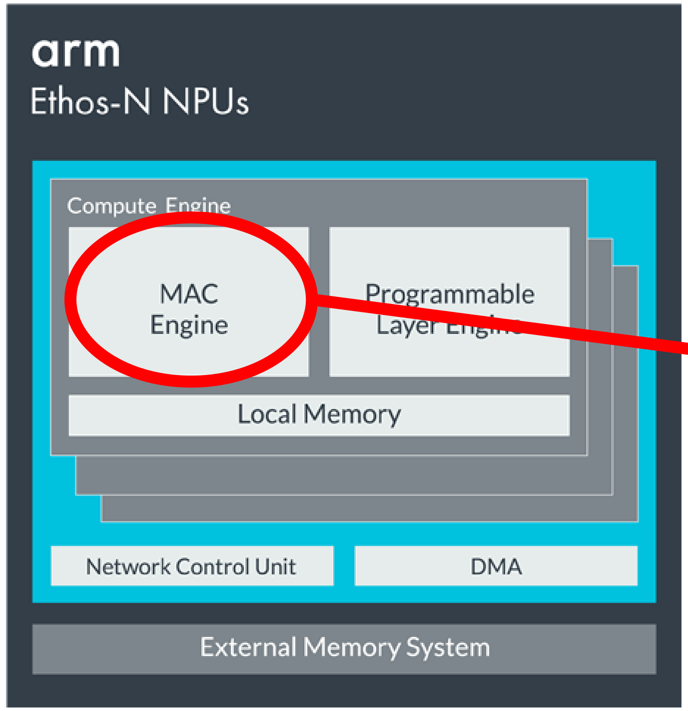
Overview

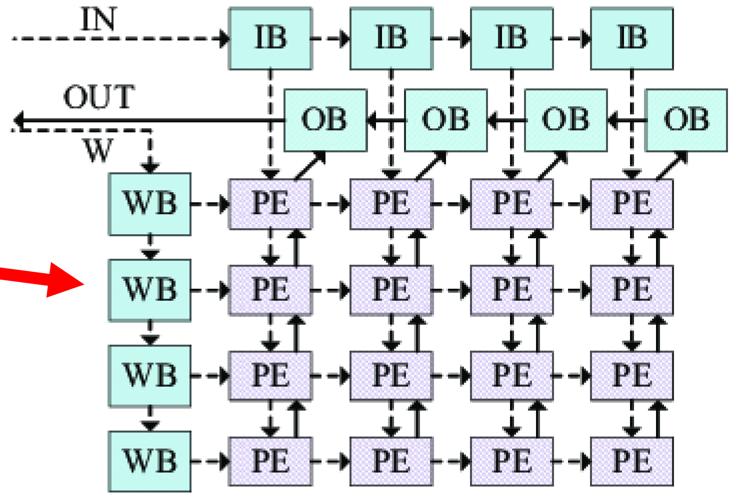
- Background: AR/VR devices.
- **Edge AI/ML Accelerations: NPU and CIM.**
- Our Automated Workflow: H4H-NAS.
- Experimental Evaluations.

NPUs for On-Device Acceleration

- Neural Processing Unit (NPU)
  - Main architecture: Systolic array.
  - SOTA method on the market (e.g., ARM Ethos-U65).
  - Suitable for <u>low-latency low-energy</u> acceleration.

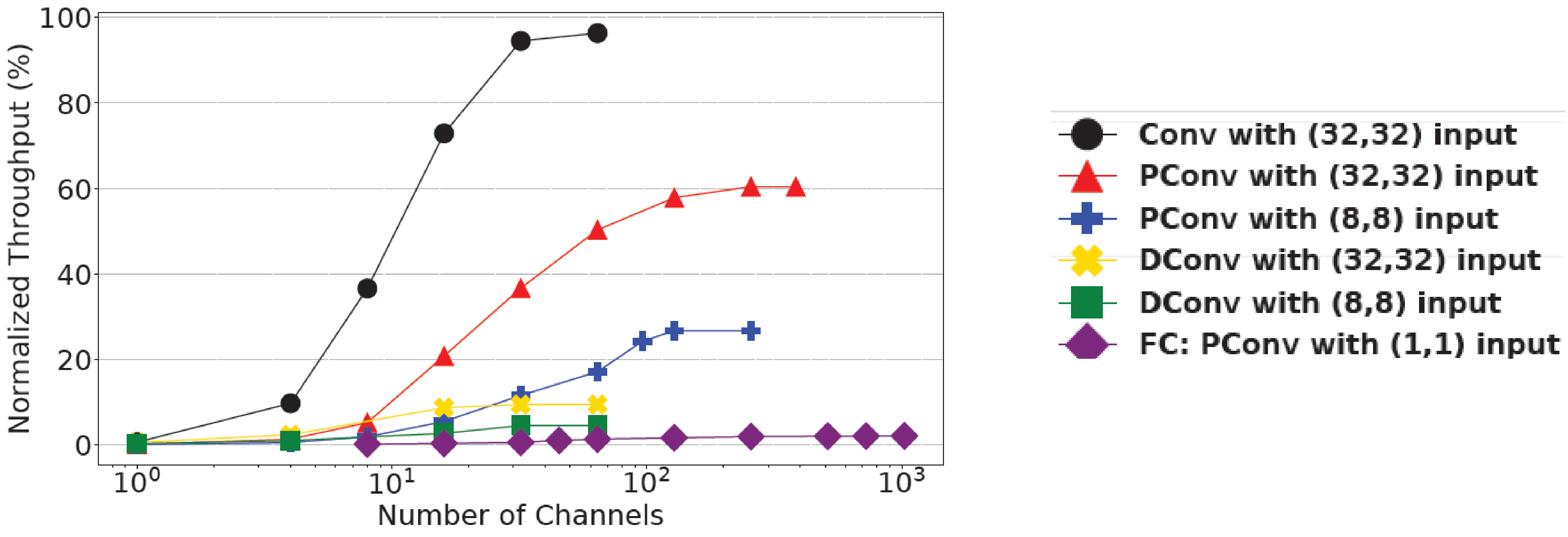






NPUs for Vision Tasks

- NPU illustrates different performances on different layer types.
- Memory Wall: NOT so good at accelerating memory-intensive layers (in latest vision transformers).

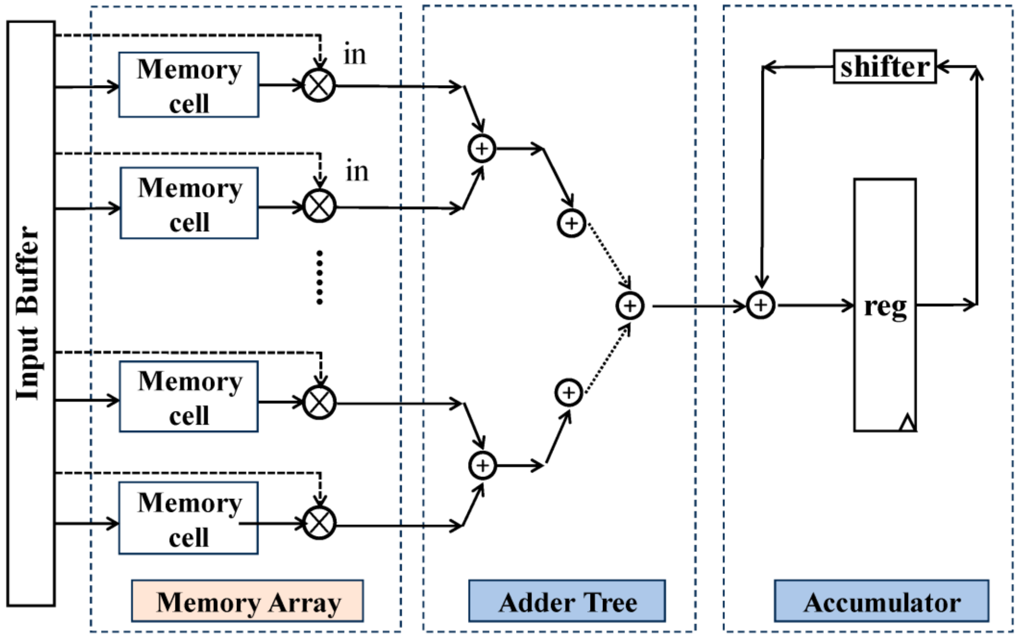


Carnegie Mellon
Parallel Data Laboratory

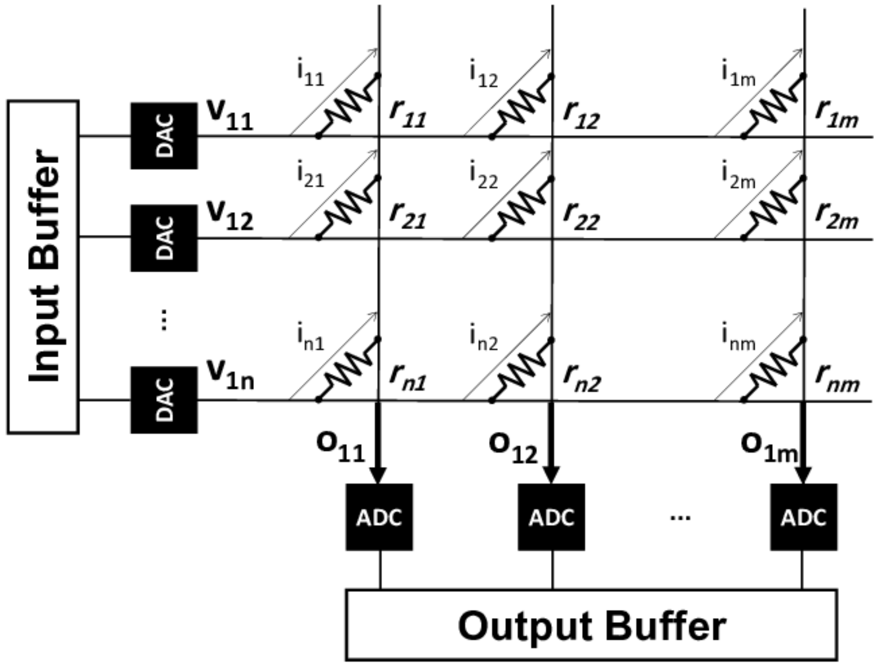
Compute-In-Memory (CIM)

- New architecture to resolve <u>memory wall</u> problem.
  - NMC: Brings computing elements close to memory.
  - CUM: Merges computing elements with memory.

Near-Memory-Compute (NMC)
An example of MRAM

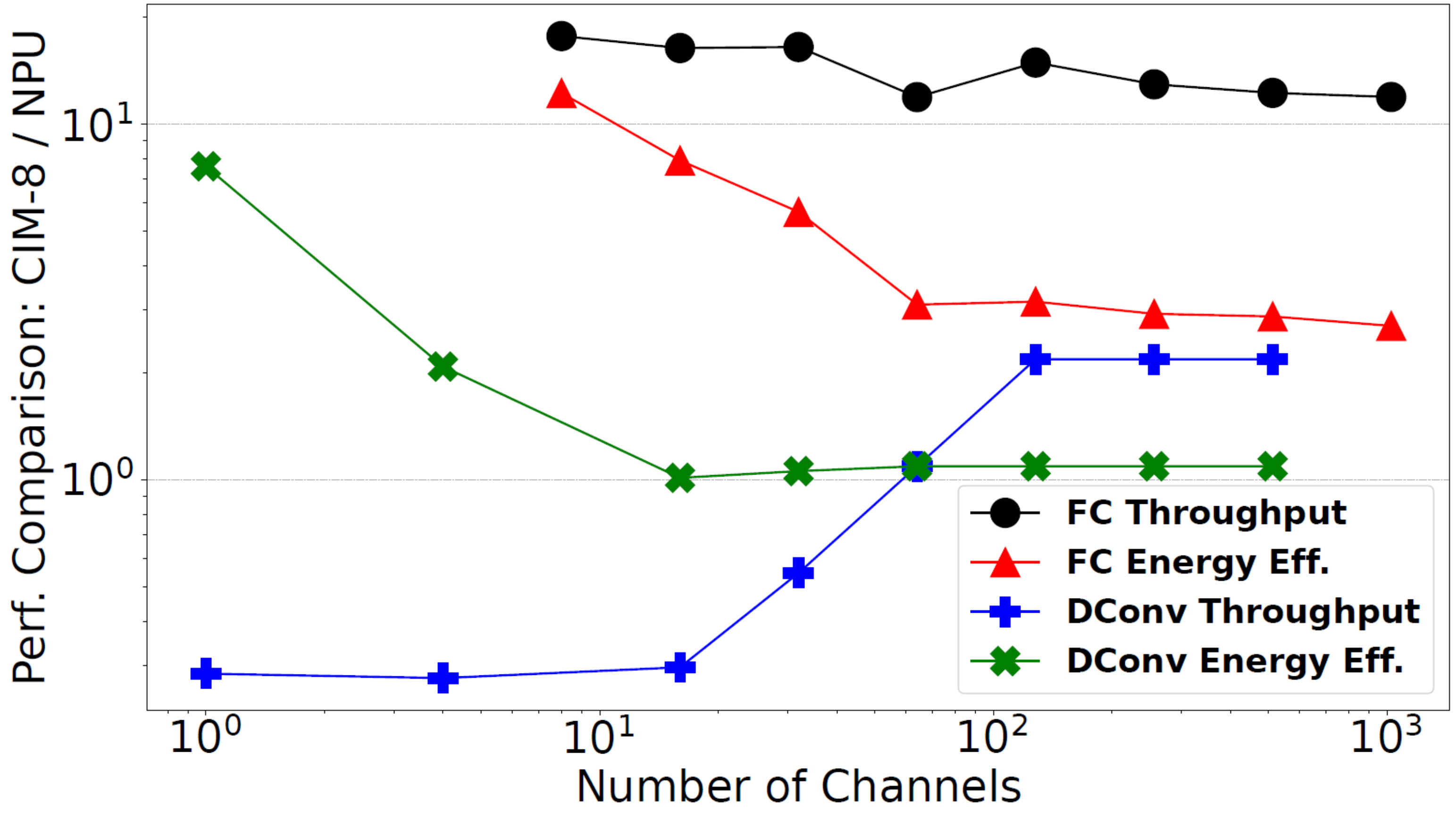


Compute-Using-Memory (CUM)
An example of ReRAM



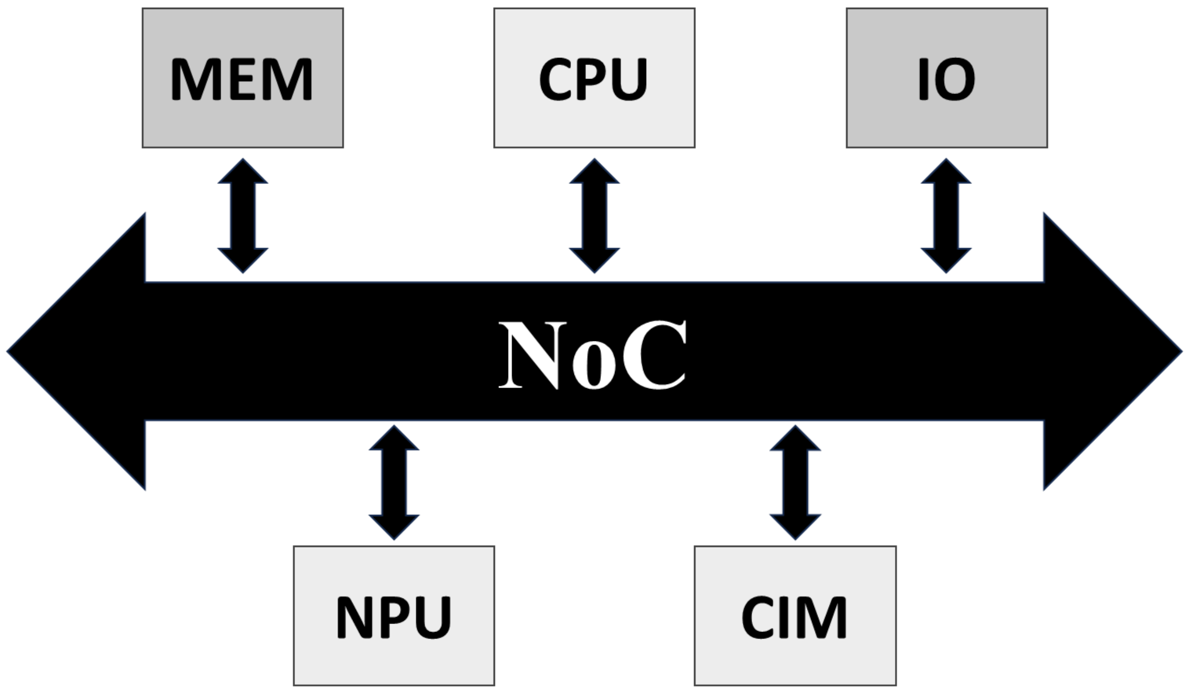
CIMs on Sensors: A Glance

- Evaluations on NMC-based CIM macros.
- CIMs are good at <u>memory-intensive</u> workloads.
  - Both in <u>throughput</u> and <u>energy efficiency</u>.



Why not use them both?

- Exploiting the heterogeneity of hardware.
- Use NPU+CIM on the same device.
- Partition the execution on different accelerators.



Overview

- Background: AR/VR devices.
- Edge AI/ML Accelerations: NPU and CIM.
- **Our Automated Workflow: H4H-NAS.**
- Experimental Evaluations.

Questions to Solve in Our Work

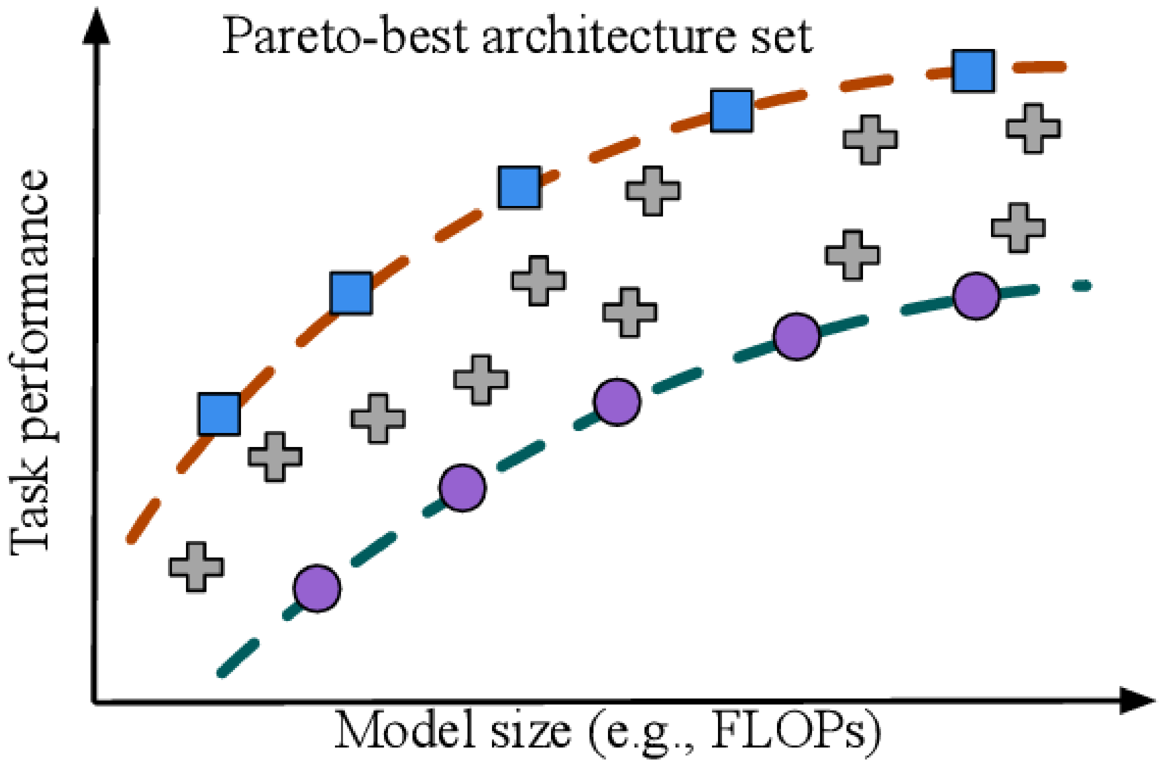
- How to design efficient models on edge devices?
  - New <u>heterogeneous</u> system design: NPU+CIM.
  - Emerging models: Vision transformers.

- How to <u>automate</u> the design process?

- Neural Architecture Search might be a solution.

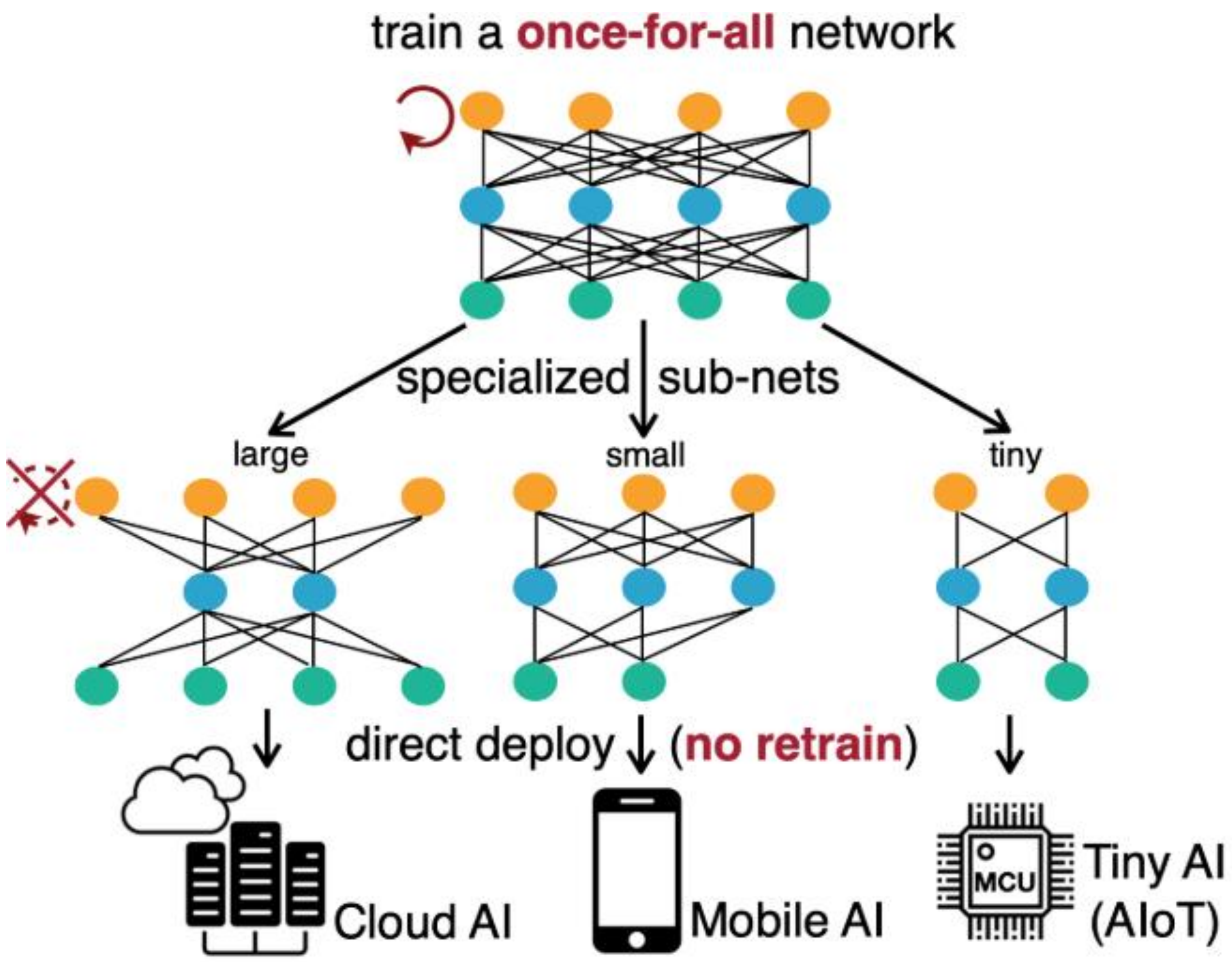
Neural Architecture Search (NAS)

- Input: A model search space.
- Output: A pareto frontier of efficient models with different sizes.
  - Size are measured in resource constraints.
  - E.g., inference latency, energy, model sizes, or FLOPs.



Neural Architecture Search (NAS)

- SOTA method: Two-stage.
- Stage I: Supernet training.
  - Trains all the models in the entire search space.
- Stage II: Searching and pruning.
  - After training, search and prunes out the most efficient sub-model given a specific resource constraint.

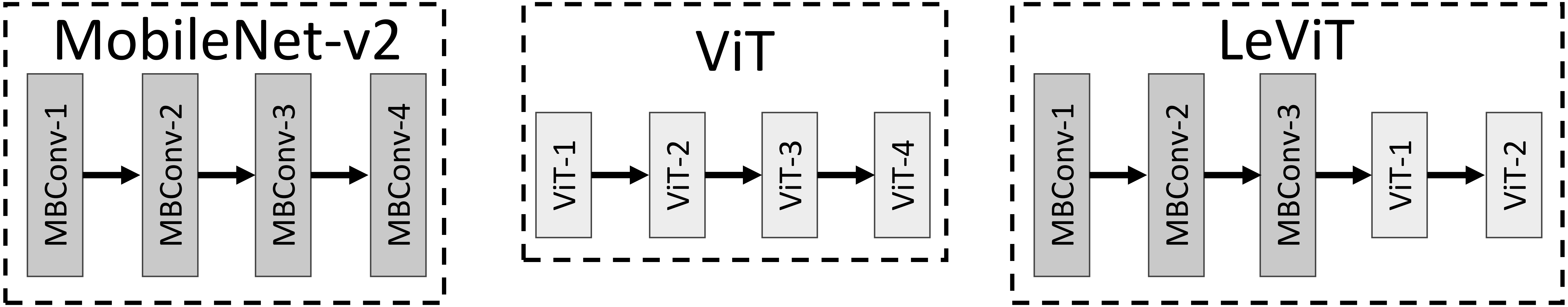


Challenges in NAS

- Challenge I (C1): Inflexible search space.
- Challenge II (C2): Adaptiveness of the training.
- Challenge III (C3): Utilizing new system in searching.

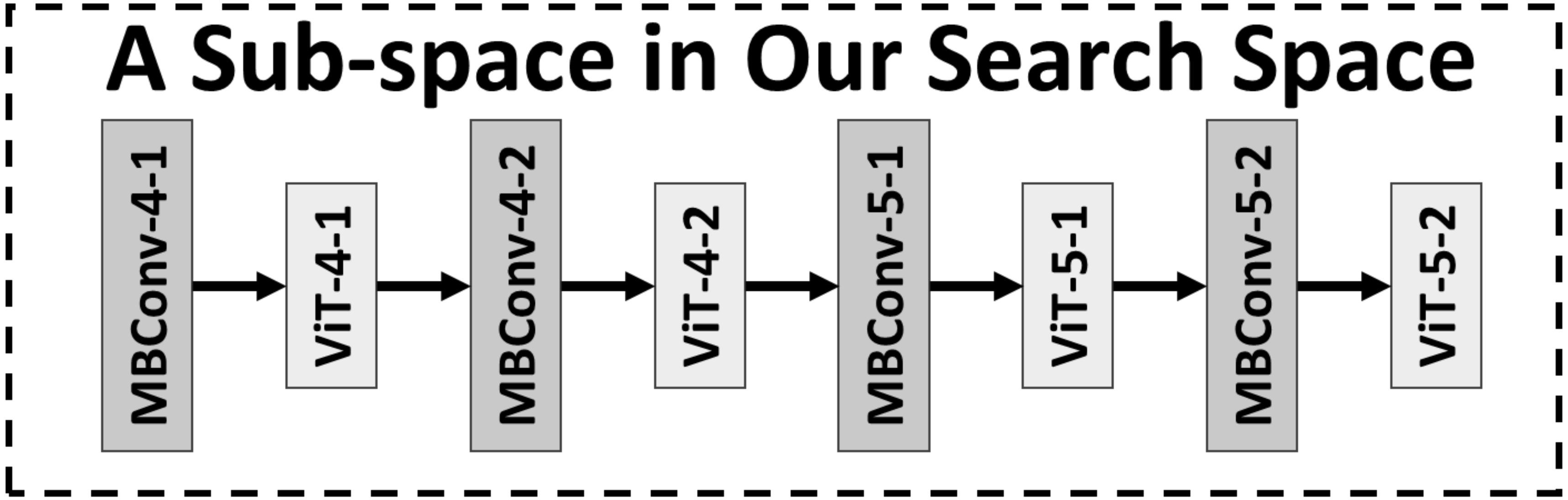
C1: Inflexible Search Space

- Challenge I (C1): Inflexible search space.
  - The main structure of the search space is pre-decided.



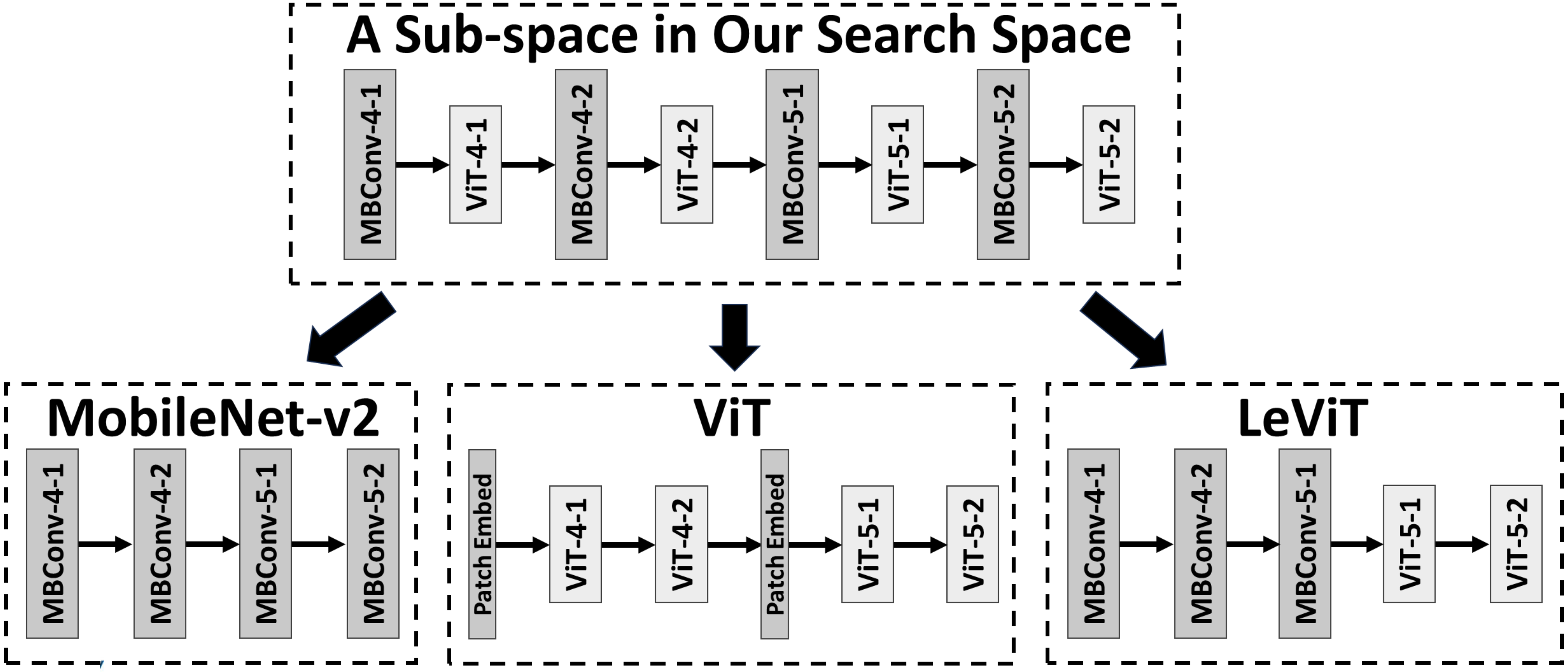
Addressing C1: Flexible Search Space

- Repeated "CNN + ViT" blocks.



Addressing C1: Flexible Search Space

- Repeated “CNN + ViT” blocks.
- Key idea: Enable the search space to be flexibly reduced.



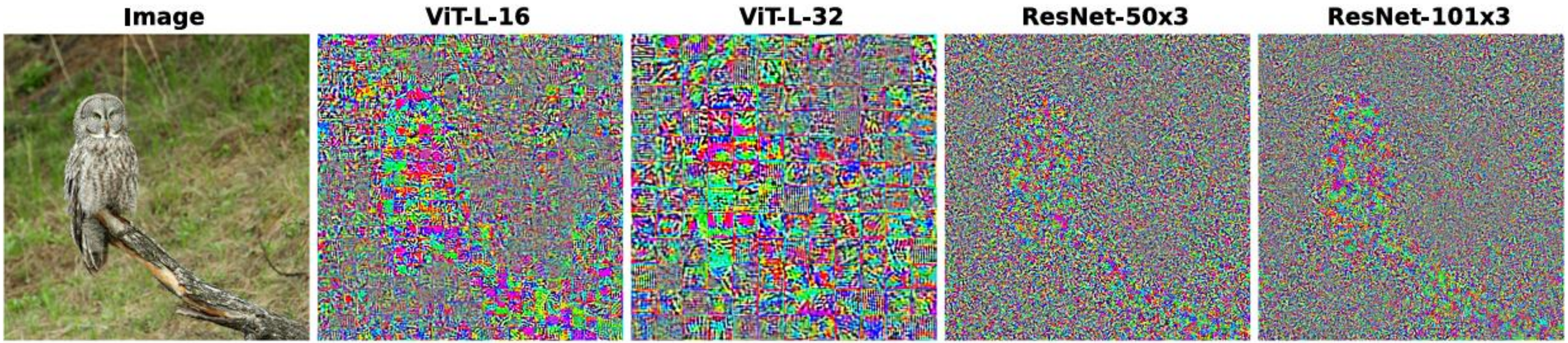
Challenges in NAS

- Challenge I (C1): Inflexible search space.
- **Challenge II (C2)**: Adaptiveness of the training.
  - Previous training method is naive sample-based training.
  - Sample subnets + weighted <u>average the gradients</u>.
  - Might not be suitable for more flexible search spaces [2].
  - Existing problem: <u>Conflicts</u> between CNN and ViT.
- Challenge III (C3): Utilizing new system in searching.

[2] https://openreview.net/forum?id=Qaw16njk6L

C2: Conflicts between CNN & ViT

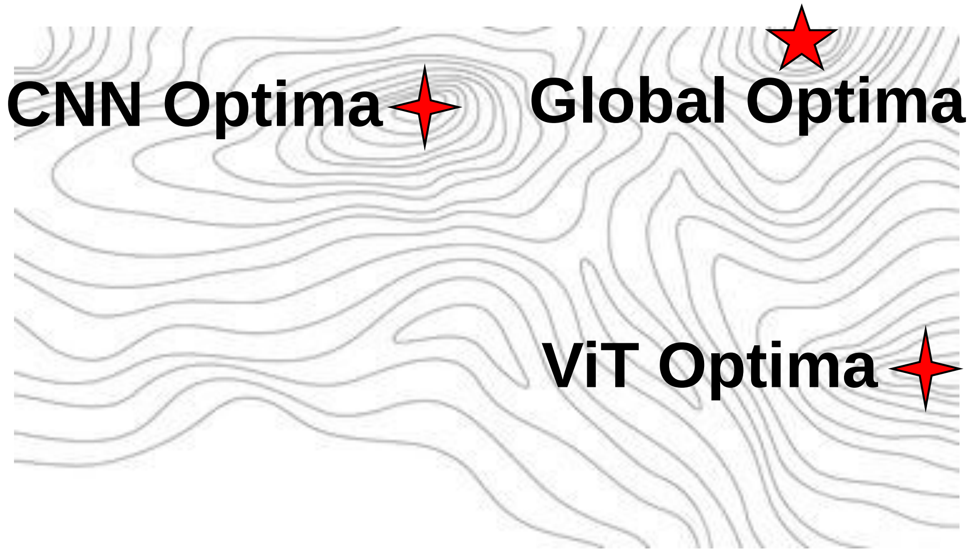
- Despite similar accuracy, ViT and CNN acts differently.
  - ViT: Low-pass filters.
  - CNN: High-pass filters.
- Adversarial perturbations shown below [3].

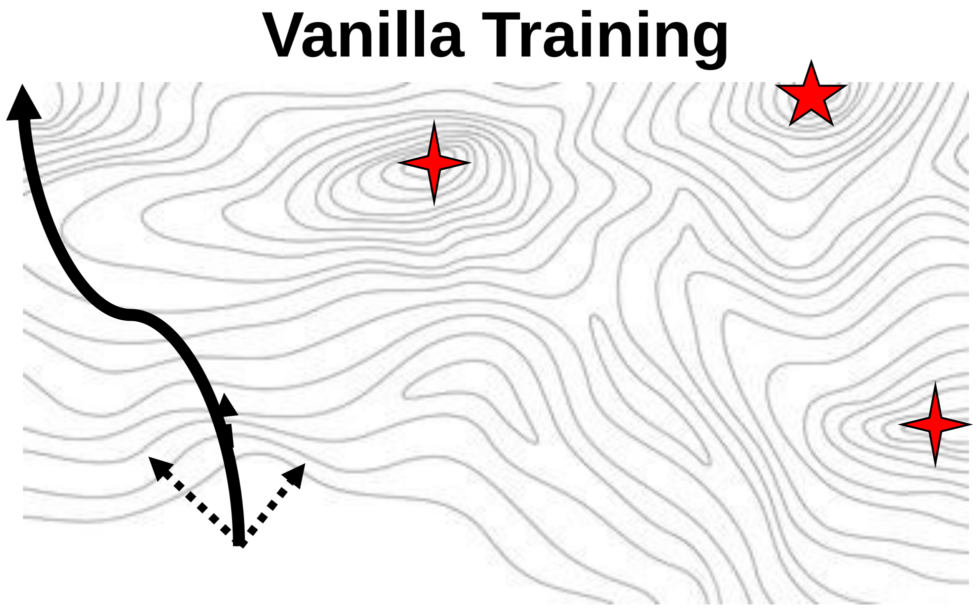


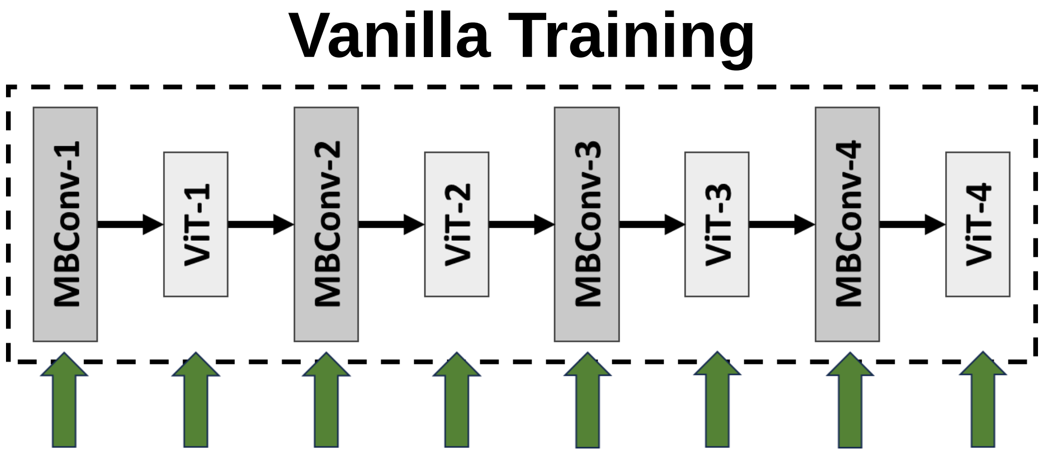
[3] https://arxiv.org/abs/2103.14586

C2: Conflicts between CNN & ViT

- Despite similar accuracy, ViT and CNN acts <u>differently</u>.
- Their gradient will <u>conflict</u> in supernet training in NAS.



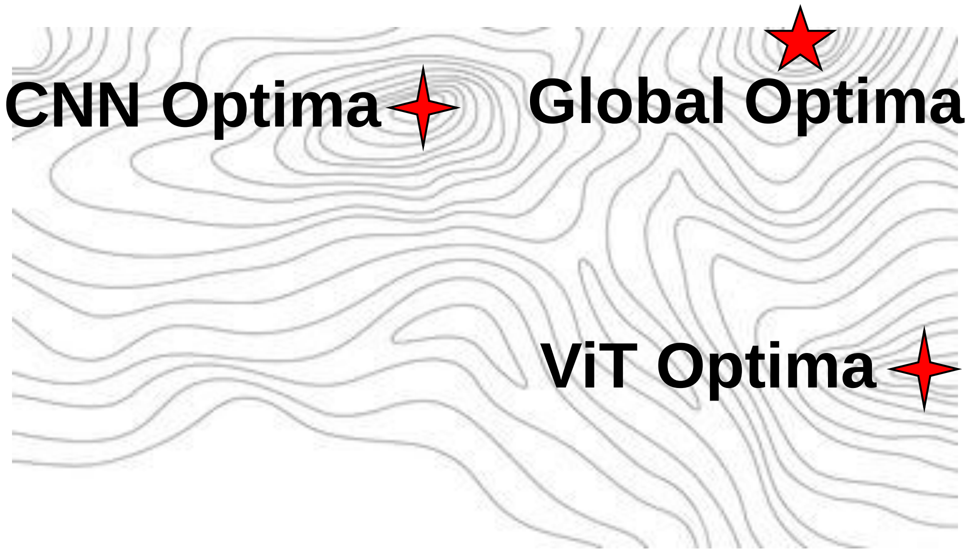


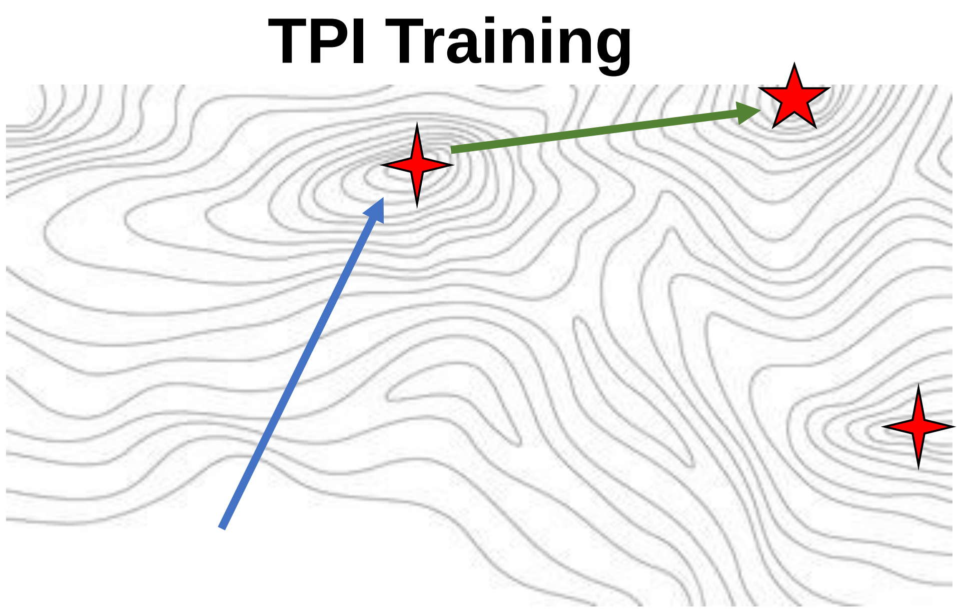


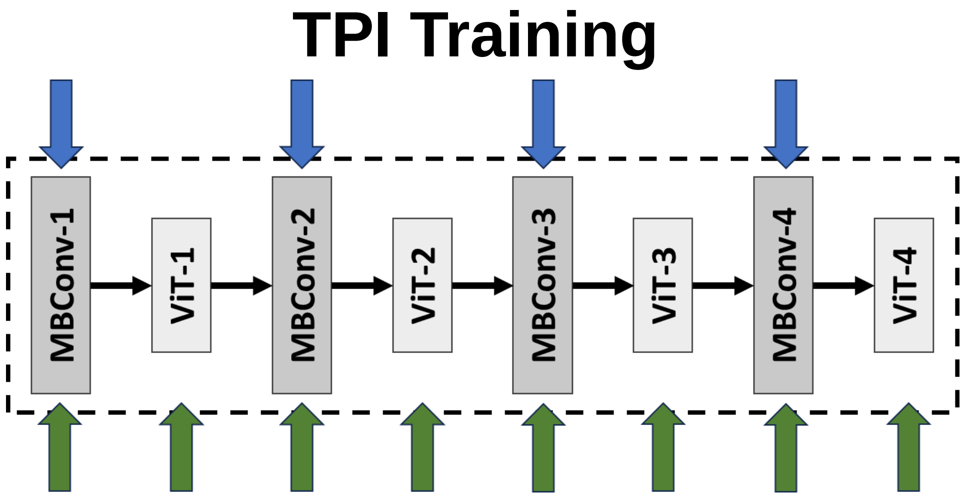
Addressing C2: Two-Phase Training

- Two-Phase Incremental Training (TPI)
  - **Phase 1**: Pre-train a sub-space containing only the CNNs.
  - **Phase 2**: Load the pre-trained CNN and train the entire space.

- Key Idea: Avoid gradient conflict in each phase.



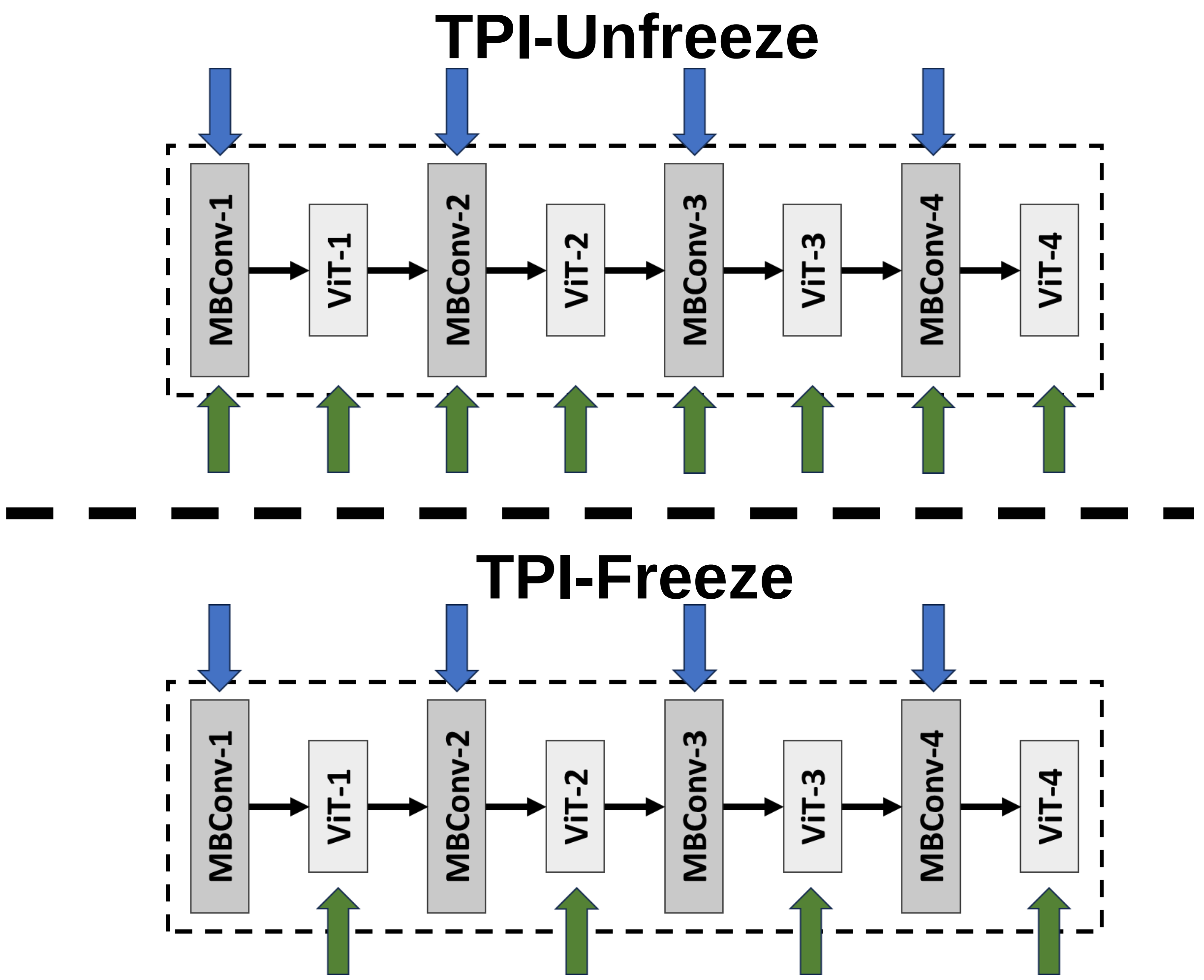




Addressing C2: TPI Results

- “Freeze” here refers to whether to freeze the CNN weights in the 2nd phase of ViT training.

Model & Recipe	Min_net	Max_net
CNN-only	71.691	78.802
Hybrid: Vanilla	71.346 (−0.345)	79.914 (+1.112)
Hybrid: TPI-unfreeze	72.140 (+0.449)	79.248 (+0.446)
Hybrid TPI-freeze	72.201 (**+0.510**)	79.782 (**+0.980**)



Challenges in NAS

- Challenge I (C1): Inflexible search space.
- Challenge II (C2): Adaptiveness of the training.
- **Challenge III (C3)**: Utilizing new system in searching.
  - Needs to provide optimal mapping & scheduling during NAS searching stage.

Addressing C3: Profiler and Scheduler

- Hardware Profiler for heterogeneous system
  - Silicon-based NPU and communication traffic profiler.
  - Simulator-based CIM profiler.

- System scheduler
  - Workflow partitioned and executed on the better device.
  - Pipelined execution between NPU and CIM.

Please refer to our paper for more details !

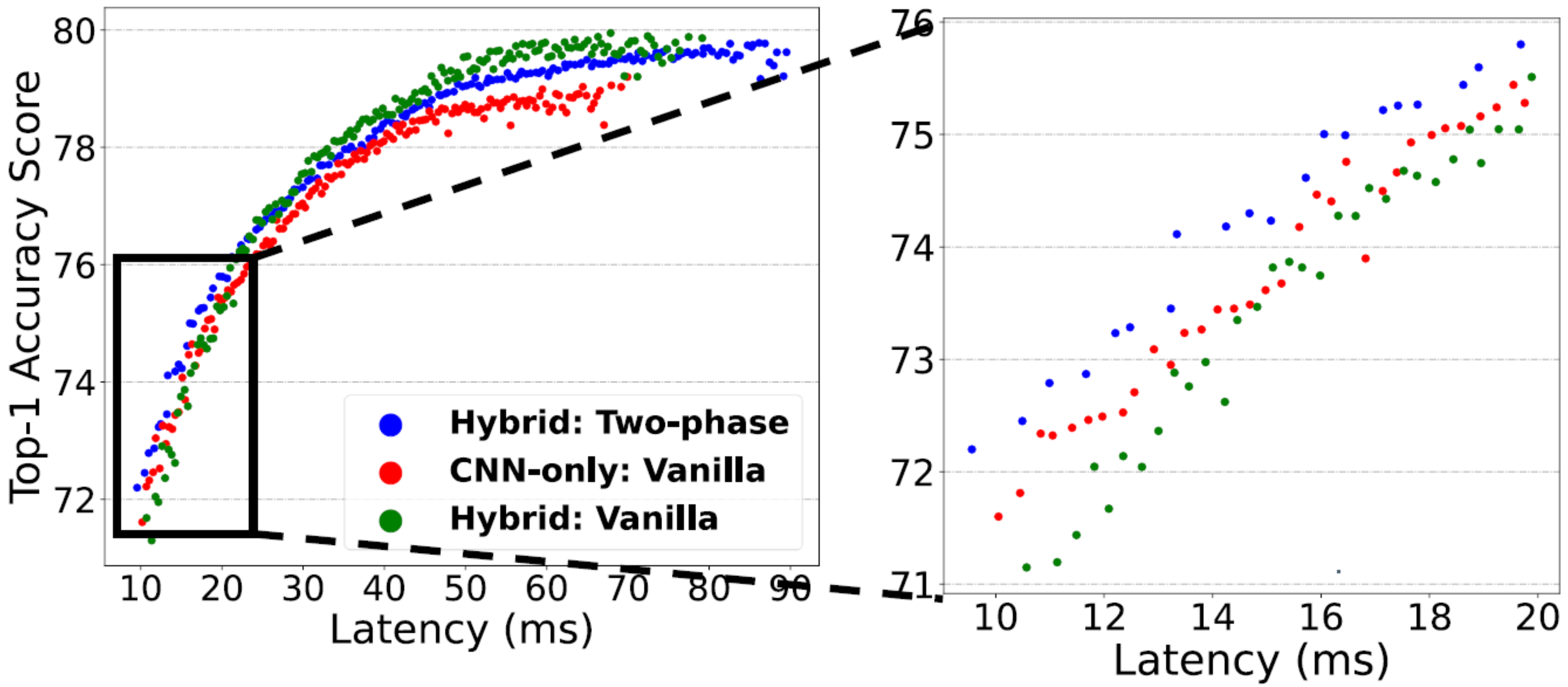
Overview

- Background: AR/VR devices.
- Edge AI/ML Accelerations: NPU and CIM.
- Our Automated Workflow: H4H-NAS.
- **Experimental Evaluations.**

Main Results: Accuracy Improvement

TPI training improves accuracy by 0.98%.

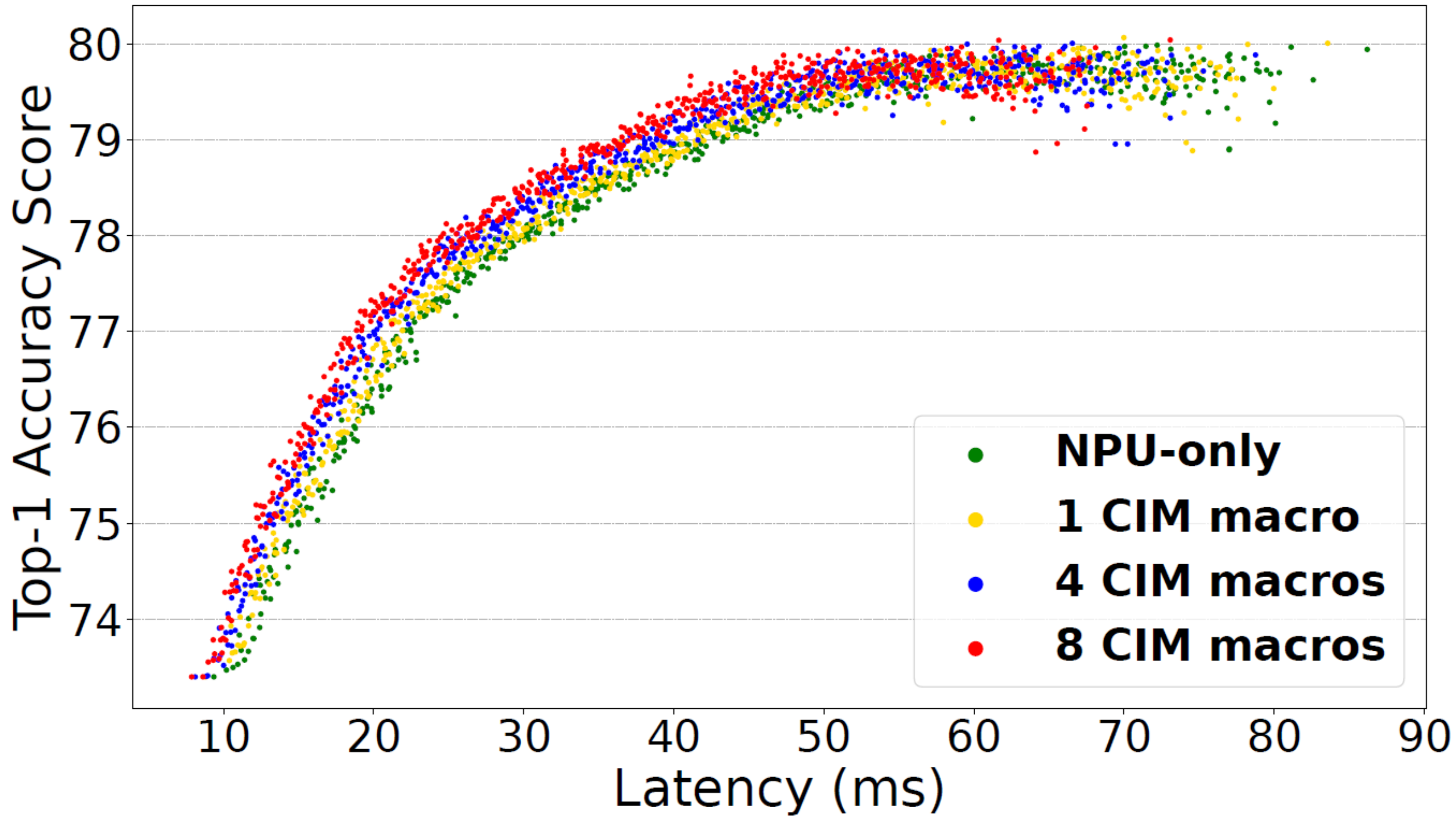
Model & Recipe	Min_net	Max_net
CNN-only	71.691	78.802
Hybrid: Vanilla	71.346 (−0.345)	79.914 (+1.112)
Hybrid: TPI-unfreeze	72.140 (+0.449)	79.248 (+0.446)
Hybrid TPI-freeze	72.201 (**+0.510**)	79.782 (**+0.980**)



Carnegie Mellon
Parallel Data Laboratory

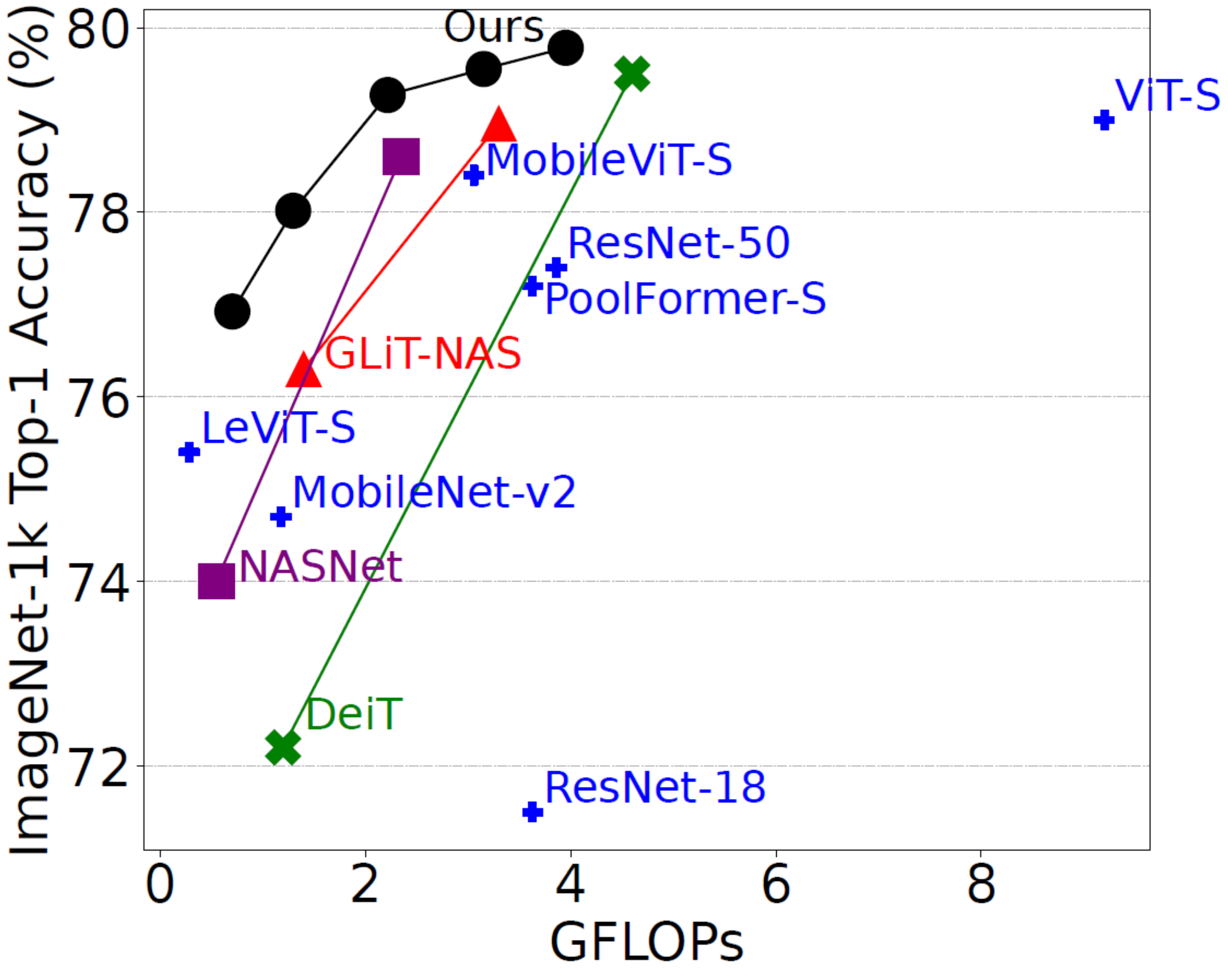
Main Results: System Performance

- H4H reduces latency: Avg 21.9% and up to 56.1%.
- H4H improves energy: Avg 19.2% and up to 41.8%.



End-to-end Comparison

- Outperforms all previous methods.
  - Either hand-crafted or NAS-based.



Interesting Finding I

- Heterogeneous edge system prefers the existence of both CNN and ViT in the same model.

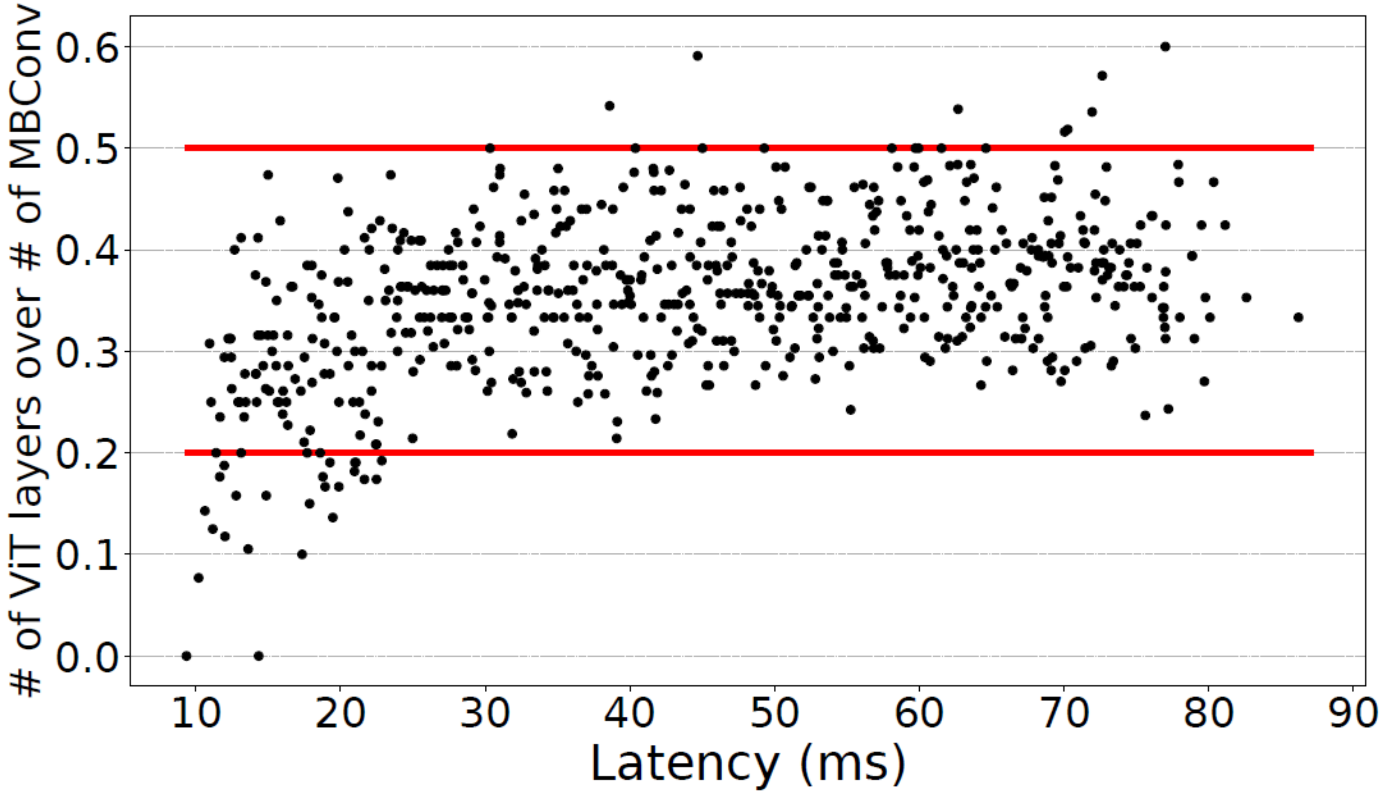
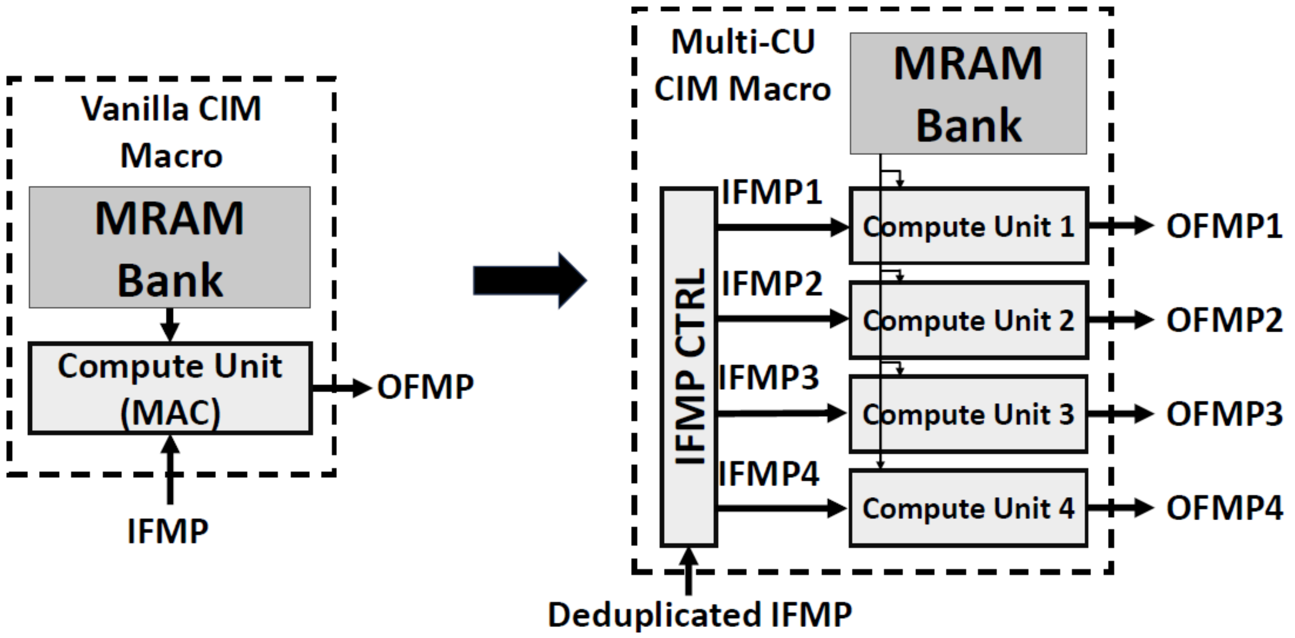


Figure 9: Ratio between number of ViT layers and number of MBConv layers, in each subnet.

Carnegie Mellon
Parallel Data Laboratory

Interesting Finding II

- Current CIM macro products are not targeting at transformer components.
- Possible solution: Add multiple compute units in the same CIM macro.



- Additionally 10.1% faster.
- Additional 9.34% energy improvement.

Key Takeaways

- AR/VR requires low-latency low-power acceleration.
- NPU + CIM serve as paradigms for edge computing.
- NAS automates the design flow, but needs changes.
- Key techniques:
  - Highly-flexible hybrid search space.
  - Two-phase supernet training for hybrid models.
  - Integrated simulator and workflow-dataflow scheduler.
  - 0.98% accuracy, 56.1% throughput and 41.8% energy improvement.