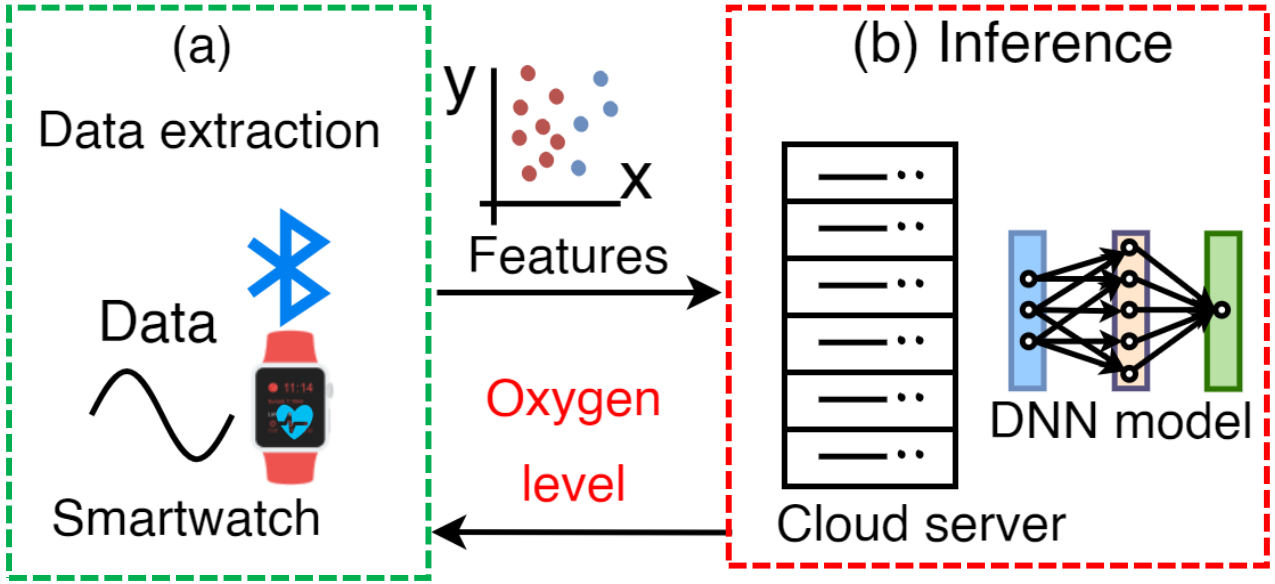
Adaptive Workload Distribution for Accuracy-aware DNN Inference on Collaborative Edge Platforms

Zain Taufique[1], Antonio Miele[2], Pasi Liljeberg[1], Anil Kanduri[1]

[1] University of Turku, Finland

[2] Politecnico di Milano, Italy

DNN Inference on Cloud

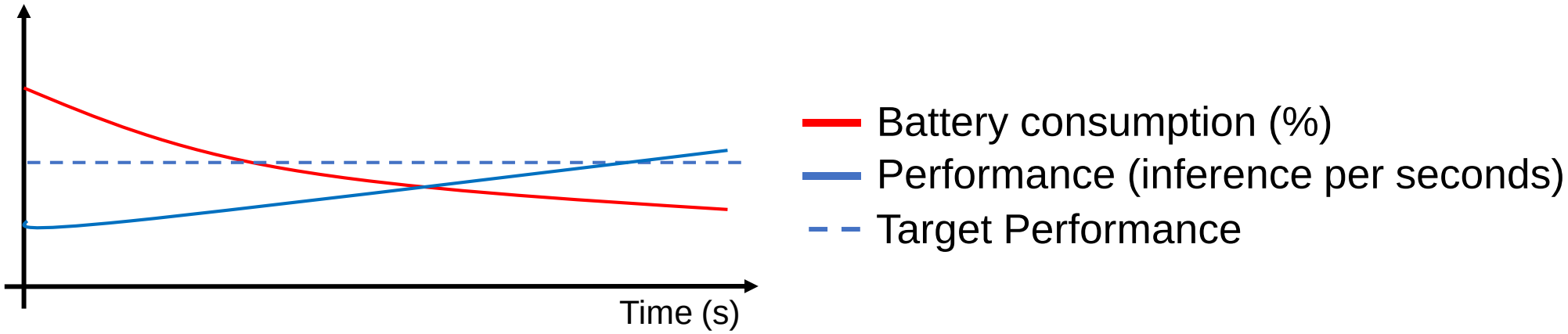


Benefits:

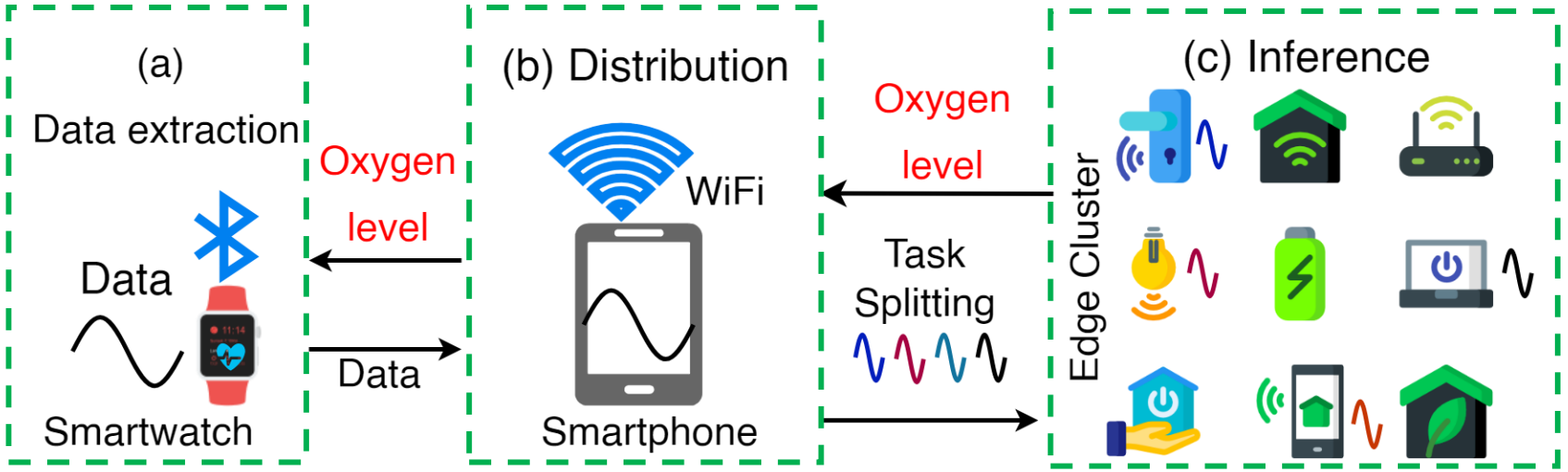
1. High Performance
2. Battery Saving

Drawbacks:

1. Internet Dependency
2. Communication overheads
3. Privacy Concerns
4. Server Cost



DNN Inference on Edge

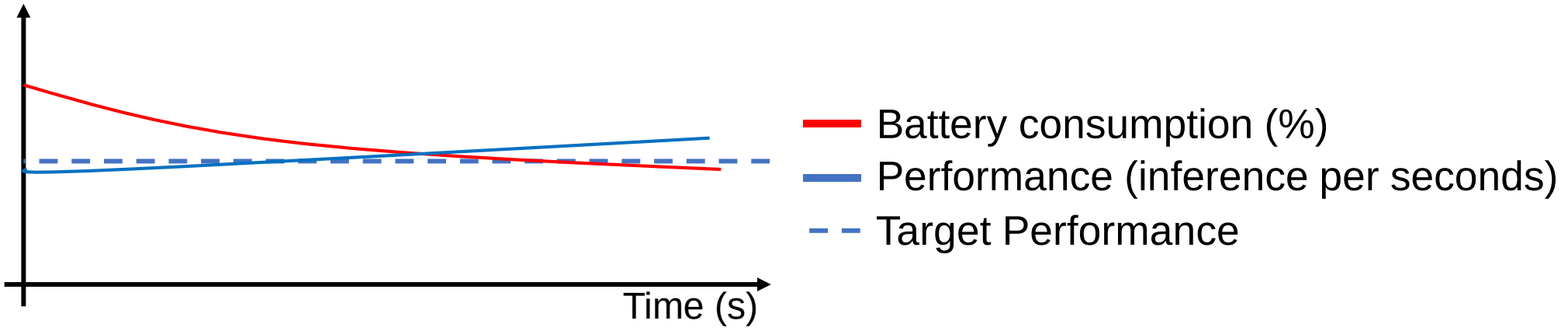


Benefits:

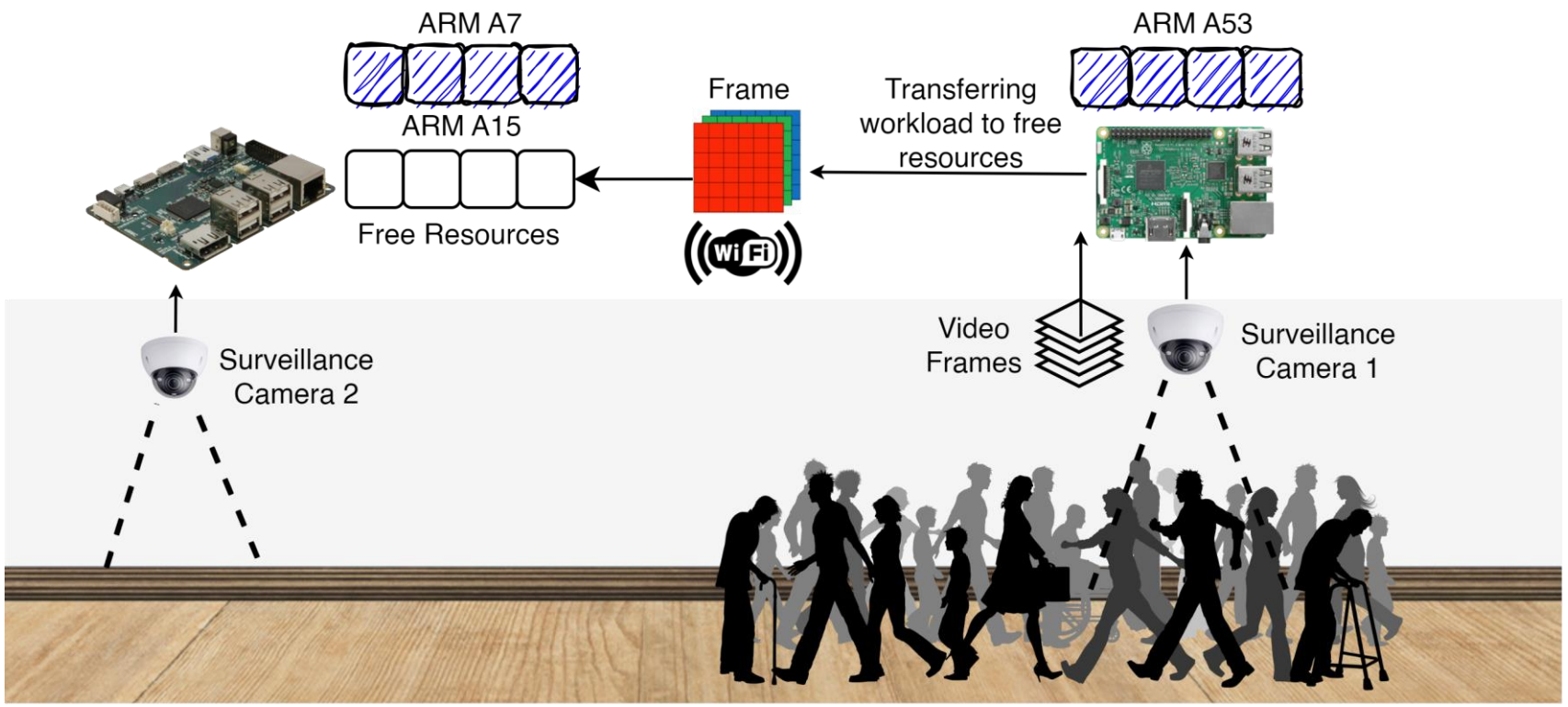
1. High Performance
2. Battery Saving
3. Fast response
4. Secure

Drawbacks:

1. Complex Design
2. Limited Resources



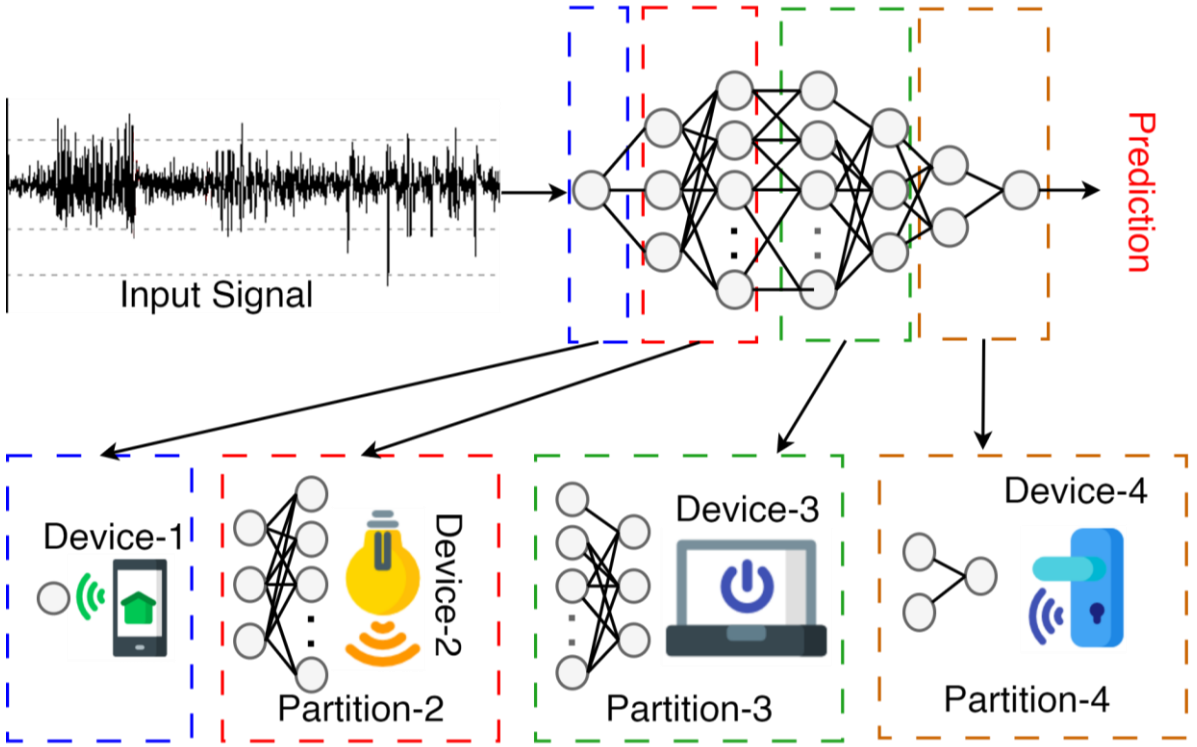
Collaborative Edge Inference



Opportunistic resource sharing to meet the performance

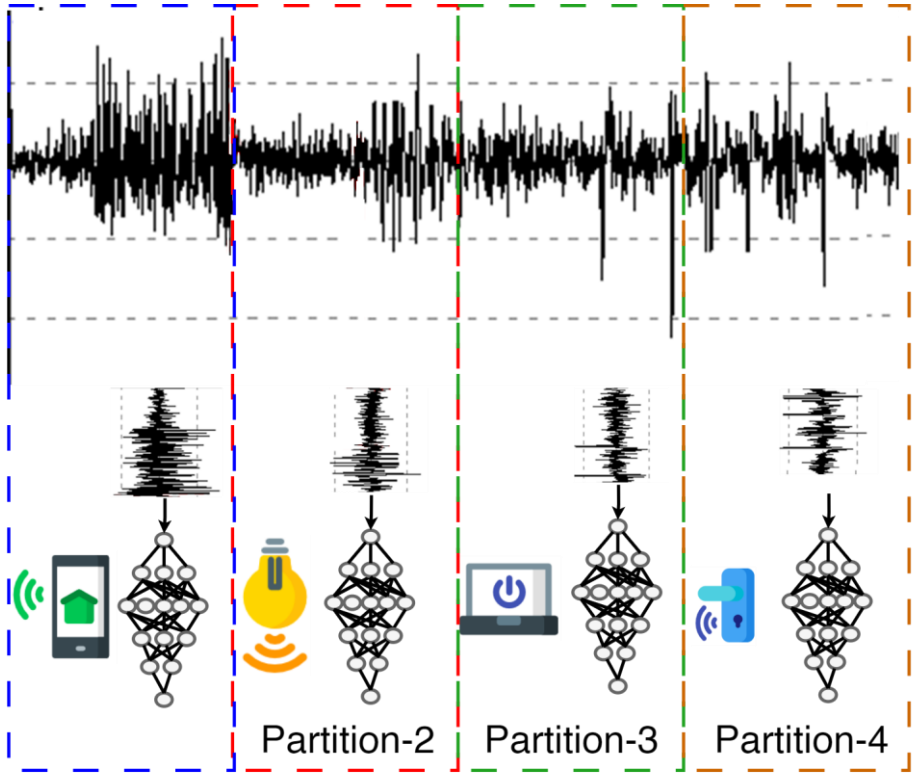
Workload Partitioning Methods

(a). Model Partitioning



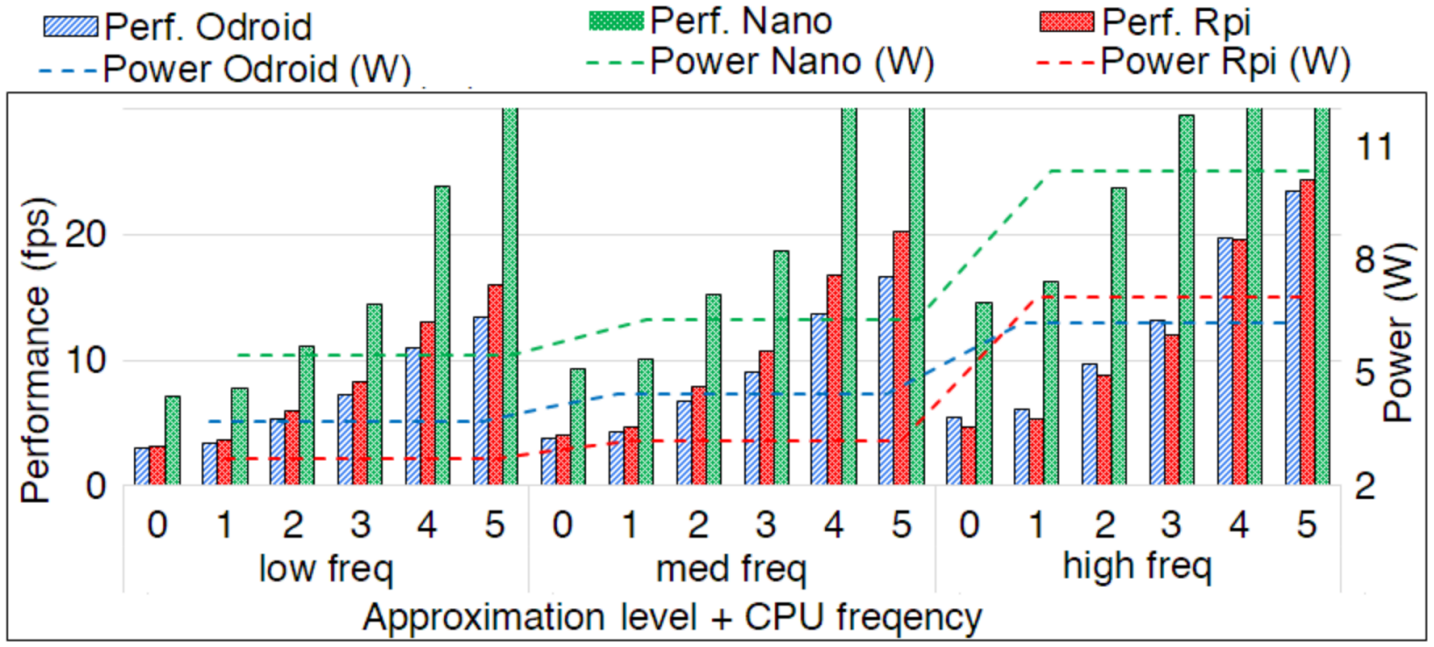
1. Complex Design
2. Architectural Limitations
3. Difficult to scale

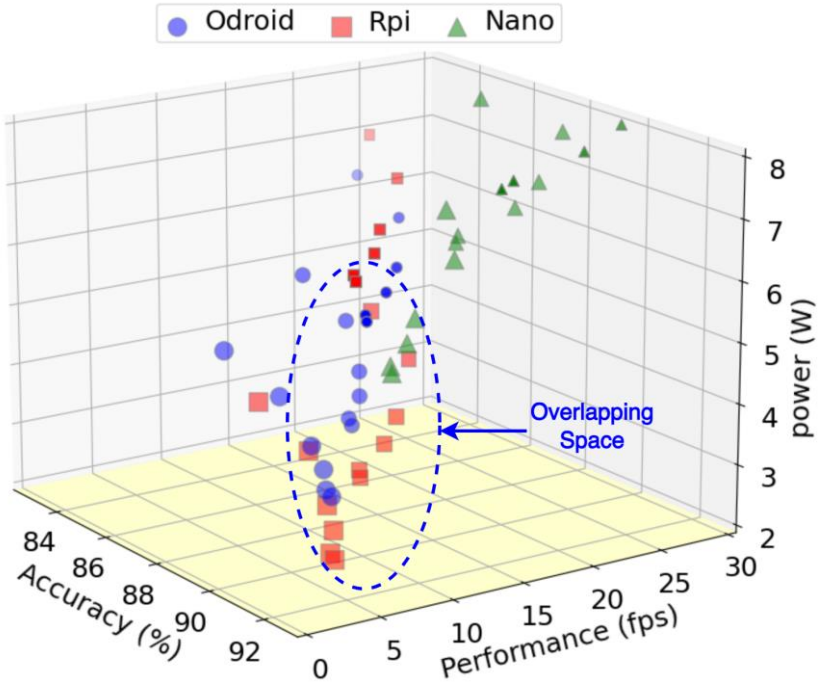
(b). Data Partitioning



1. Scalable
2. Feasible for input parallel applications

Inference on Heterogeneous Edge Nodes

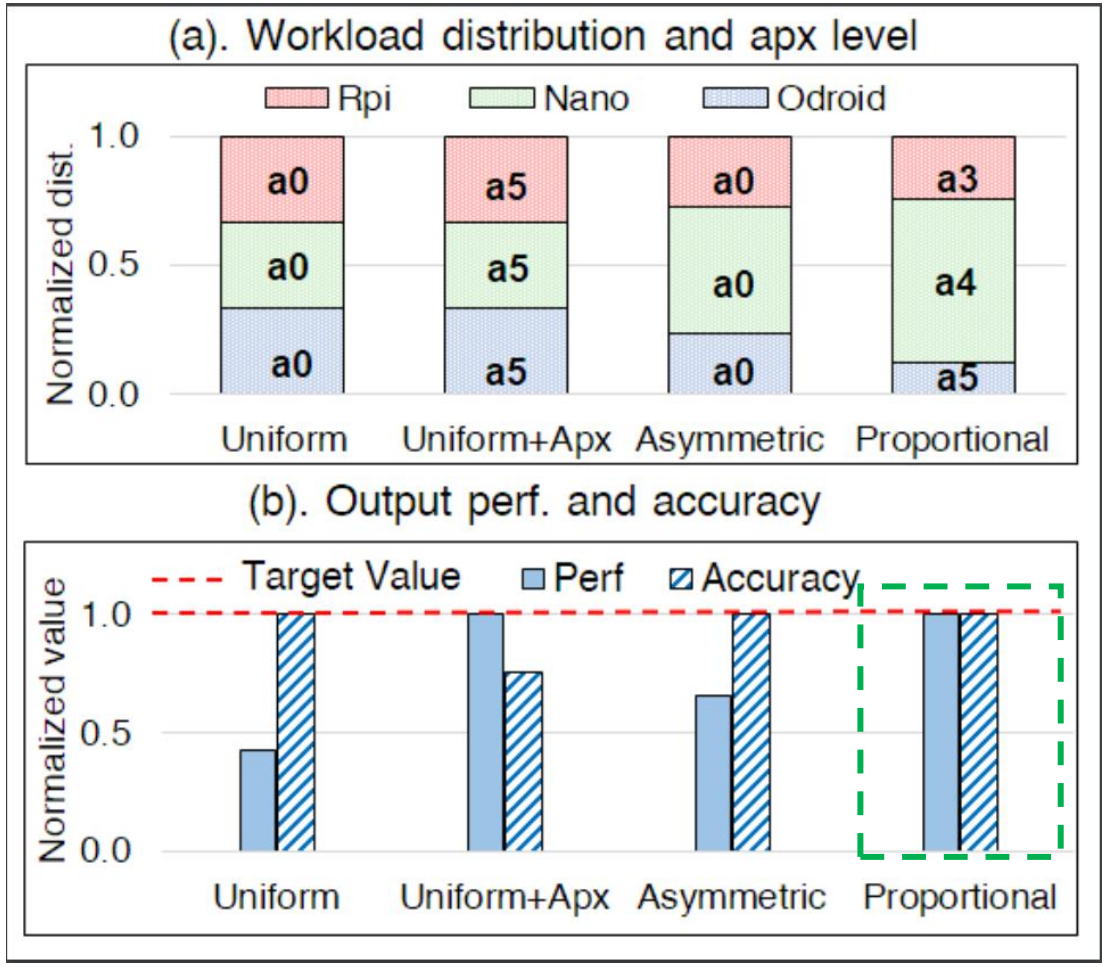




Complex trade-off space between different edge platforms

Possible Pareto-optimal points

Workload Distribution Strategies



Ideal workload distribution:

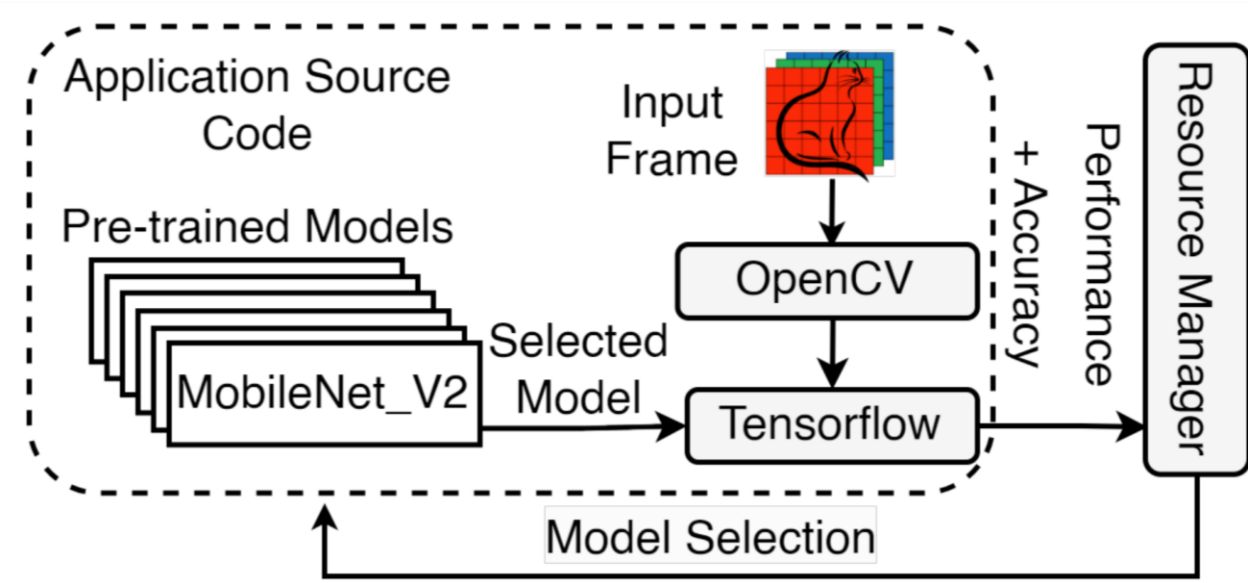
- ✓ Target performance met
- ✓ Target accuracy met

Comparison with State-of-the-Art

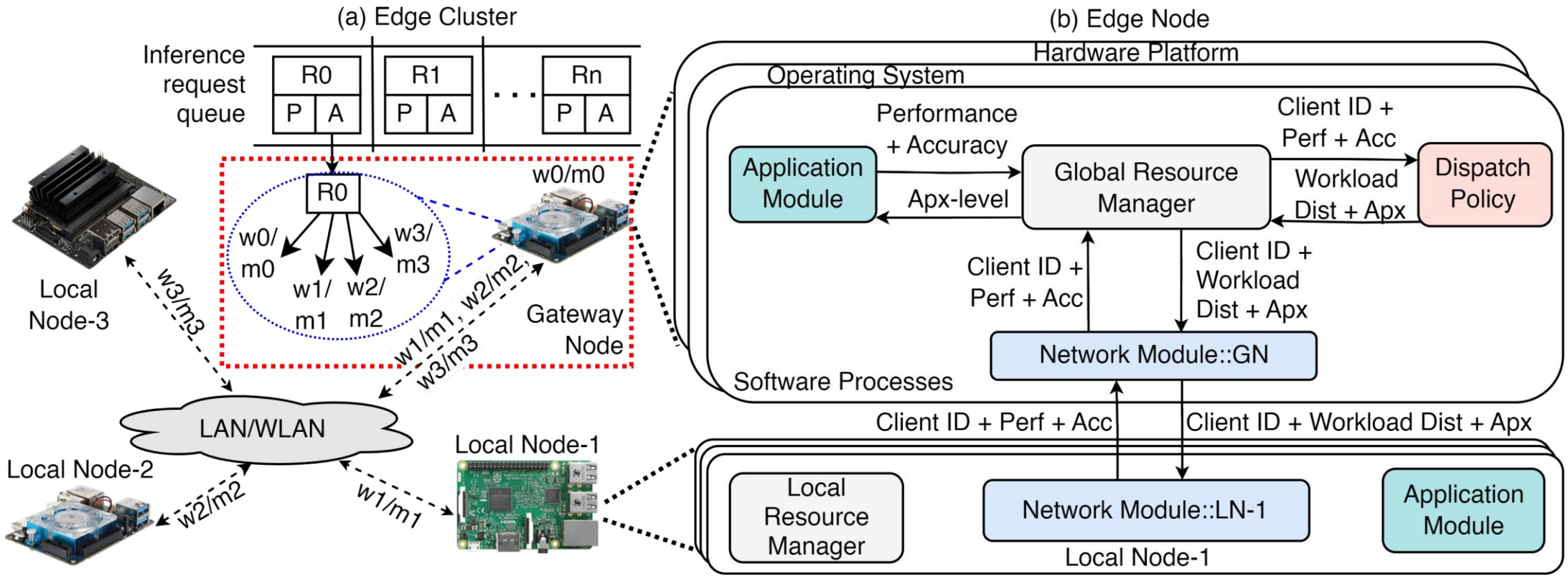
	[1]	[9]	[10]	[11]	[3]	[5]	[12]	[13]	**Our**
Perf. aware	✓	✓	✓	✓	✓	✓	✓	✓	✓
Acc. aware	✓	✘	✘	✓	✓	✓	✘	✘	✓
Heter. aware	✓	✓	✘	✘	✓	✘	✓	✓	✓
Adaptive	✓	✘	✘	✓	✓	✓	✓	✓	✓
Run-time	✘	✘	✘	✘	✘	✓	✘	✘	✓

Approximation with Model Selection

1. Pre-trained light models take less memory
2. Run-time availability to cater varying performance demands
3. Ensures adaptability in dynamic application scenarios

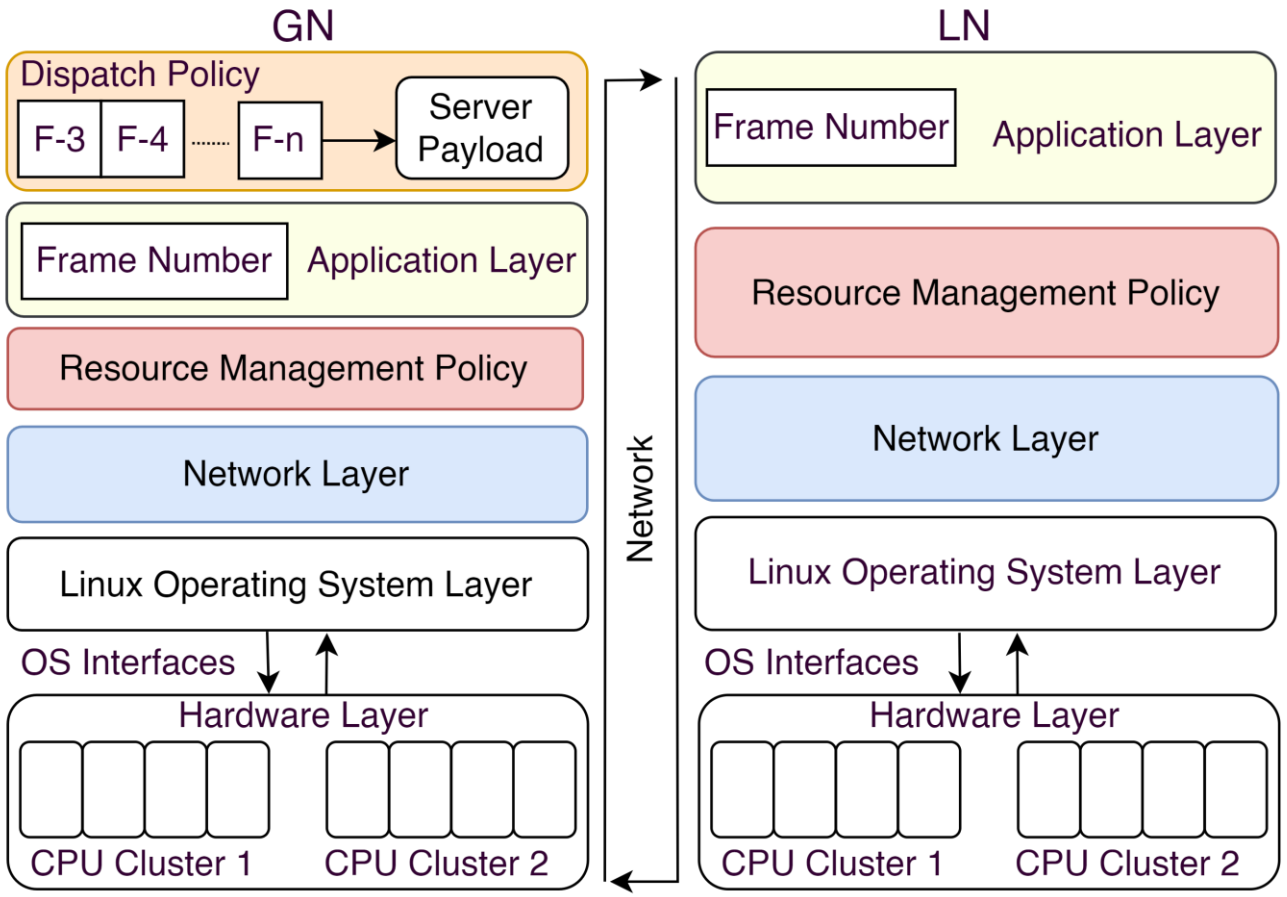


Proposed DNN Inference Framework



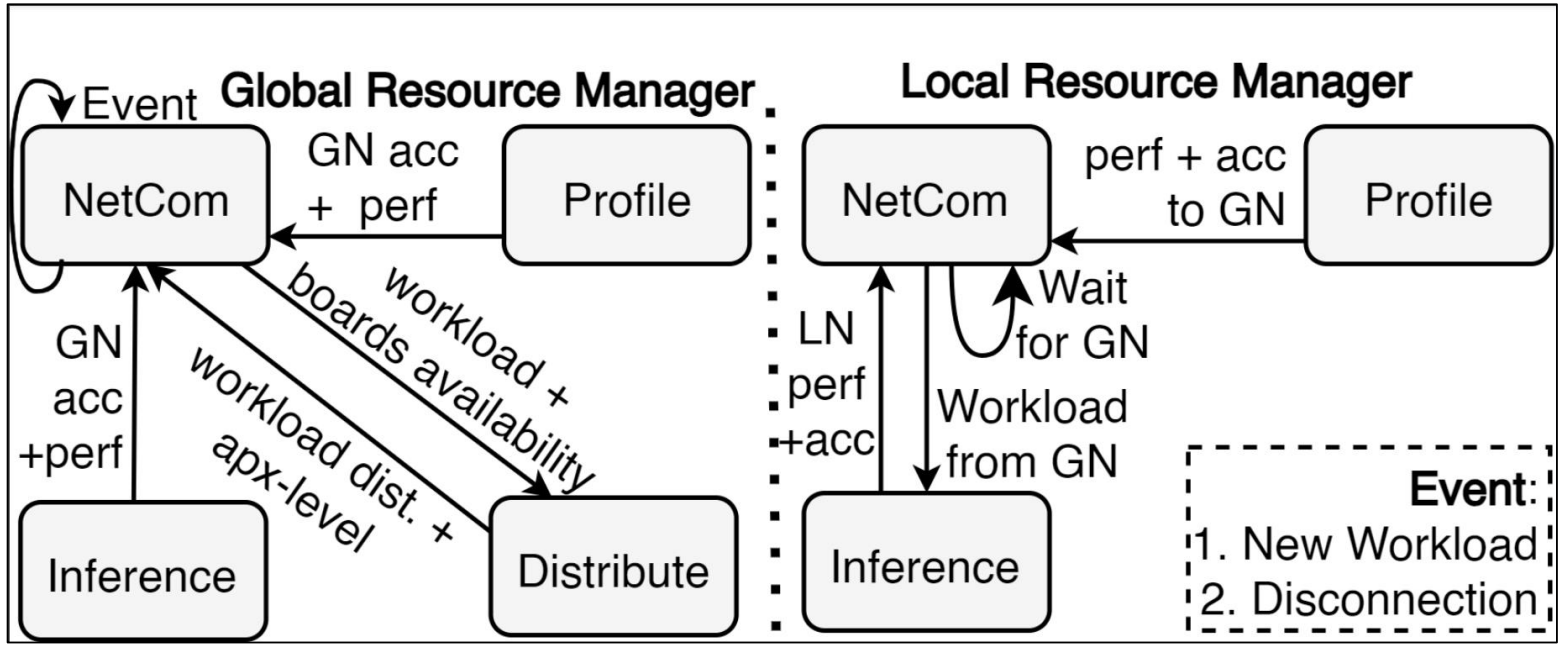
Node Architecture

1. Dispatch Policy is only available in the Global Node
2. All nodes are supported by Linux Operating System
3. Application Layer is responsible for inference
4. Network Layer enables communication between nodes



Resource Manager

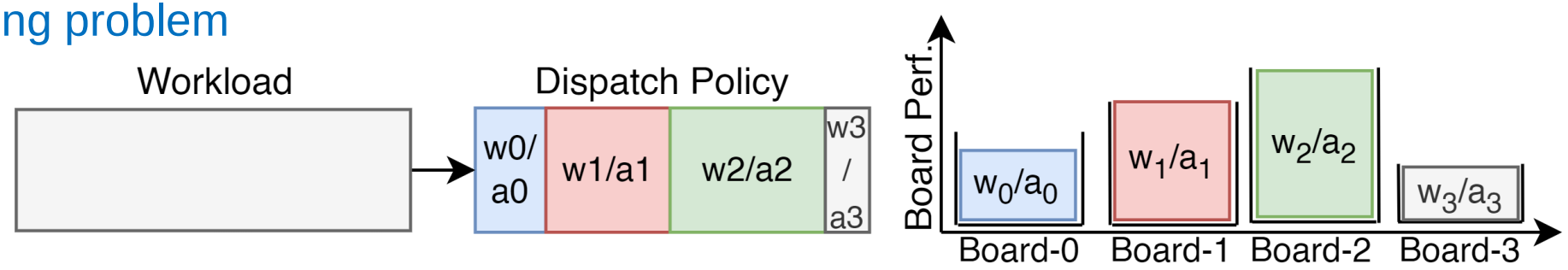
1. Implemented as a finite state machine
2. Triggers in case of new inference request
3. Initial profiling is done to populate the exploration table



Design Space Exploration for Workload Distribution

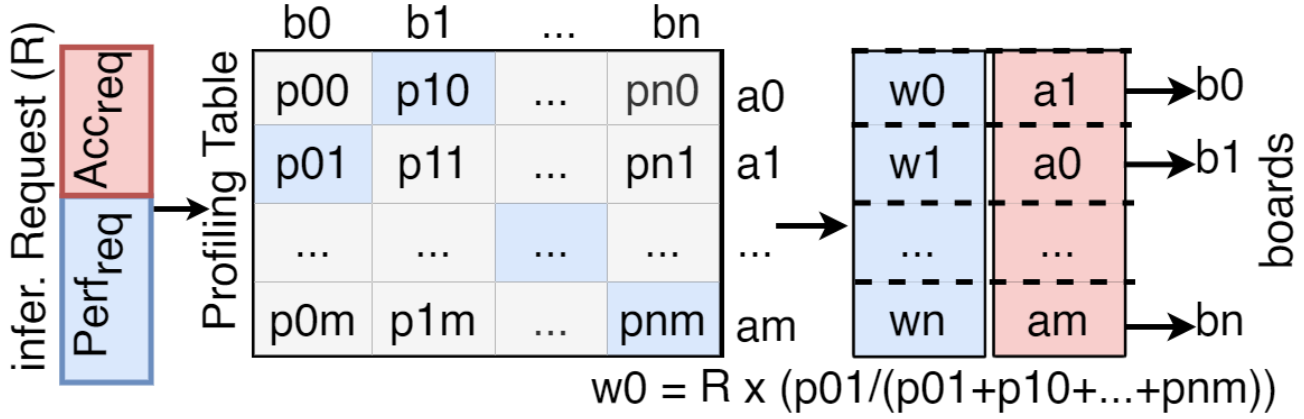
Formulated as a bin-packing problem

(a). Problem formulation



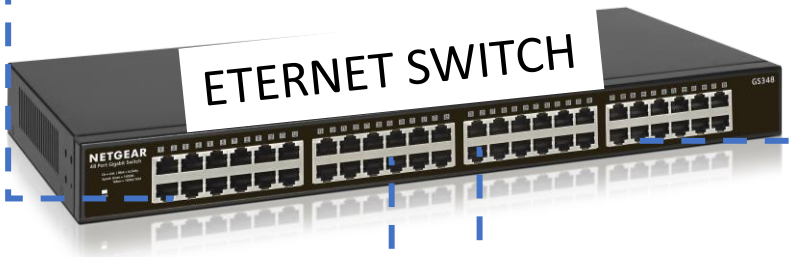
Exploration space is populated in the profile phase

(b). Exploration



Experimental Setup

Odroid XU4
Octa-core ARM
4xA53, 4xA72

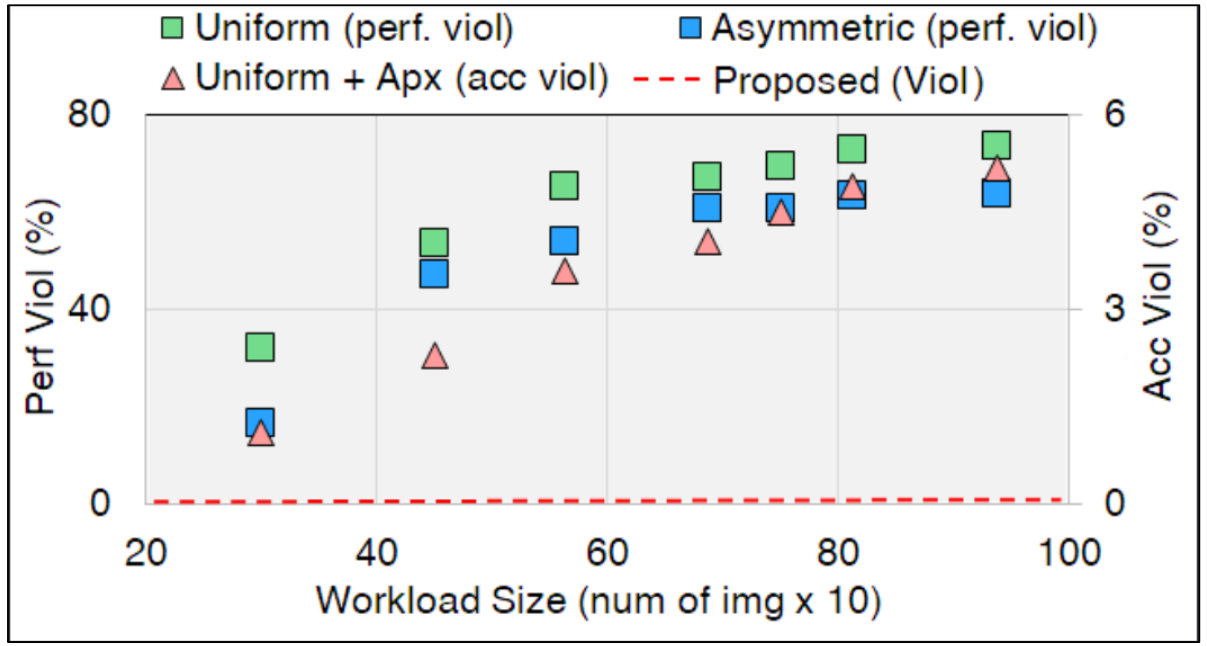


Odroid XU4
Octa-core ARM
4xA53, 4xA72

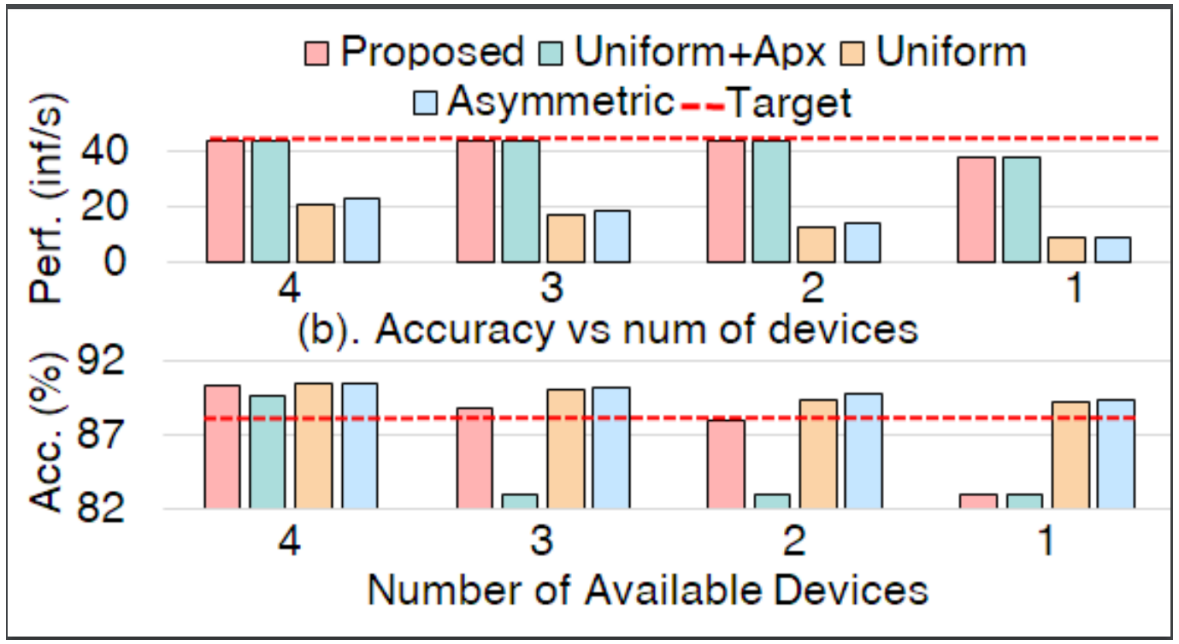
Jetson Nano
Quad-core ARM
4xA57
128 NVIDIA CUDA

Raspberry Pi 4
Quad-core ARM
4xA54

Performance Evaluation

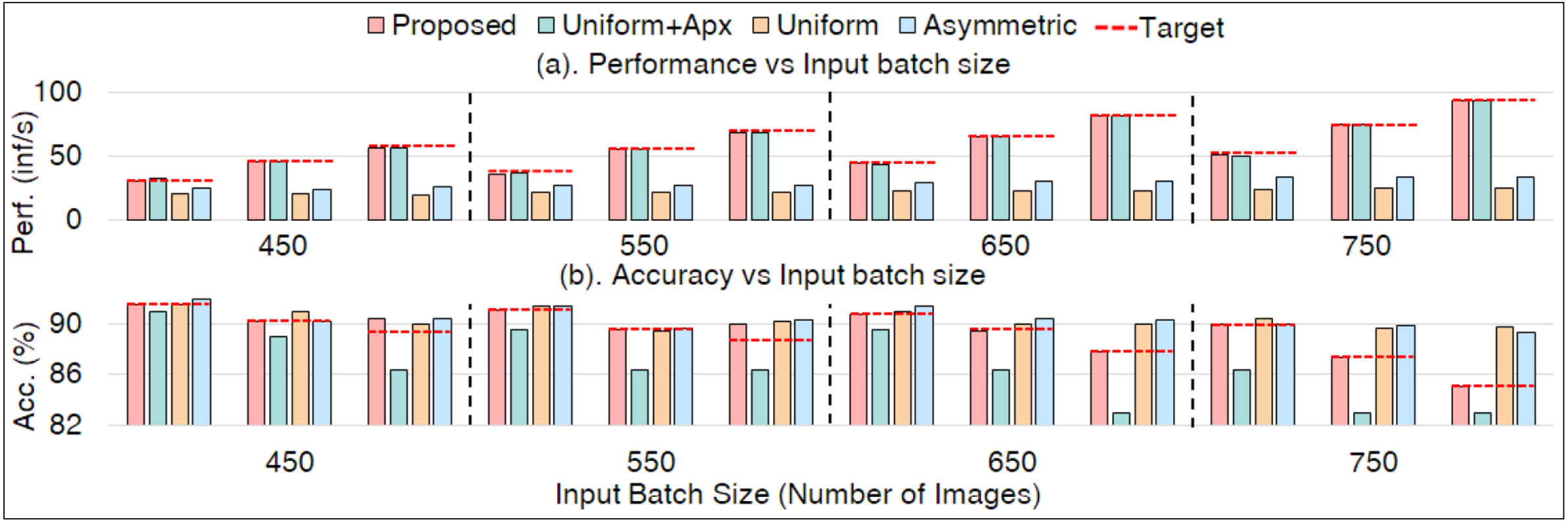


(a). Workload Size vs Performance and Accuracy Violations



(b). Num of devices vs Performance and Accuracy Violations

Performance with variable input sizes



Conclusion

- Adaptive workload distribution policy
  - Partitions DNN inference requests on collaborative heterogeneous edge clusters.
  - Exploits accuracy-performance trade-offs of DNN models
  - Jointly determines optimal partitioning points and accuracy levels of dynamic inference requests
- Strategy implementation on a real hardware testbed
  - Cluster of devices – 2xOdroid Xu4, Raspberry Pi 4, Jetson Nano
  - Workload distribution policy implemented as a middleware
- Evaluation against relevant workload distribution strategies
  - Average performance gain of 42.5%
  - Average accuracy gain of 4.2% against other approximation methods