INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
ICPP/2021/CHICAGO/USA
acm In-Cooperation
sighpc
AUGUST 9–12, 2021

Joint Optimization of DNN Partition and Scheduling for Mobile Cloud Computing

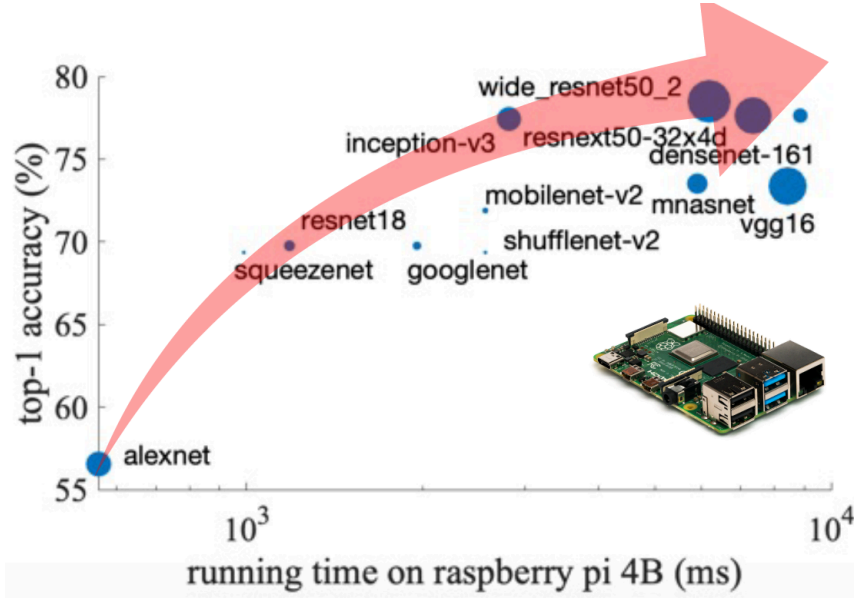
Yubin Duan and Jie Wu

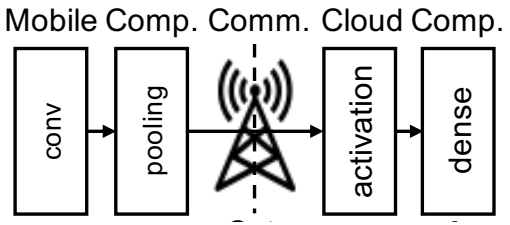
Temple University

TEMPLE UNIVERSITY

Background

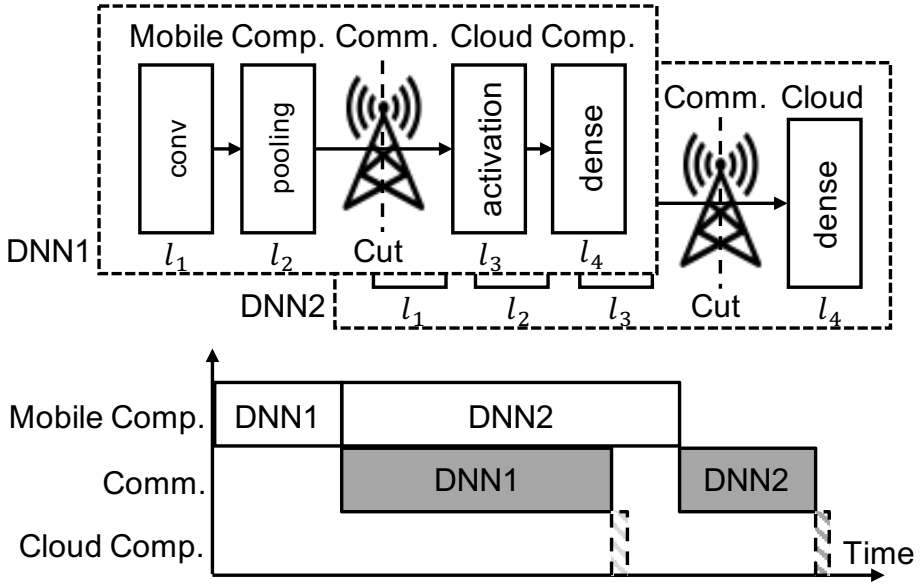
- DNN Inference on Mobile Devices
  - Inference latency matters
  - Use pre-trained DNNs
  - Execute forward propagation
- Cooperative Deep Inference
  - Powerful cloud servers
  - Fast communication channels
  - Offload computation workload





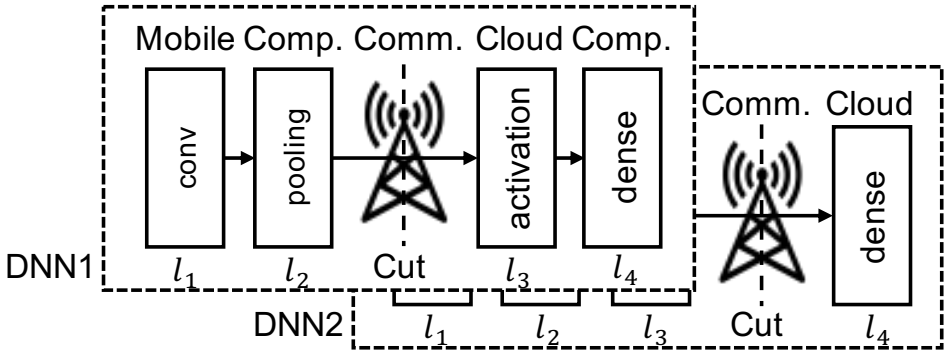
Motivation

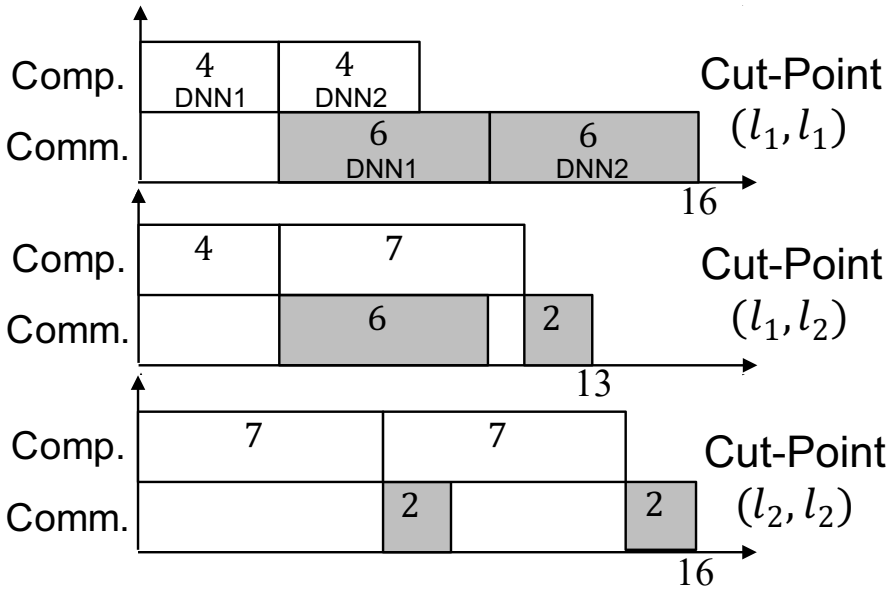
- Duplicated Inference Tasks
  - Simultaneously arrive
  - Auto pilot, AR/VR
- Cooperative Inference Pipeline
  - Reduce inference latency
  - Hide comm. behind mobile comp.
  - Cloud comp. is negligible
  - Partition and scheduling problem
    - Minimize overall latency for duplicated inference tasks



Challenges

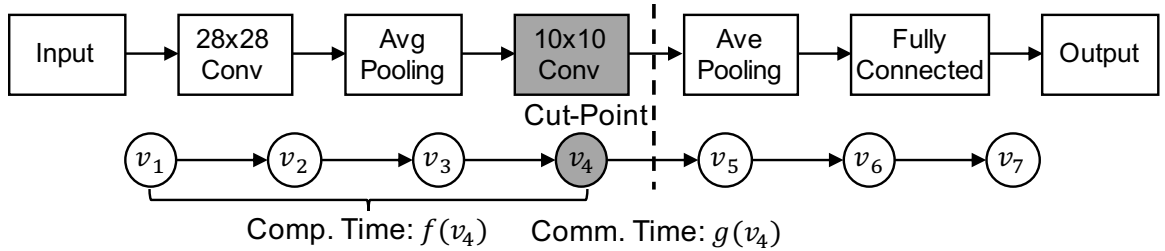
- Exponential Partition Plans
  - Each DNN has # layers cut points
  - Partitions are individual
- Partition-Scheduling Correlation
  - Best individual partition
  - \+
  - Best pipeline scheduling
  - ≠
  - Optimal overall latency





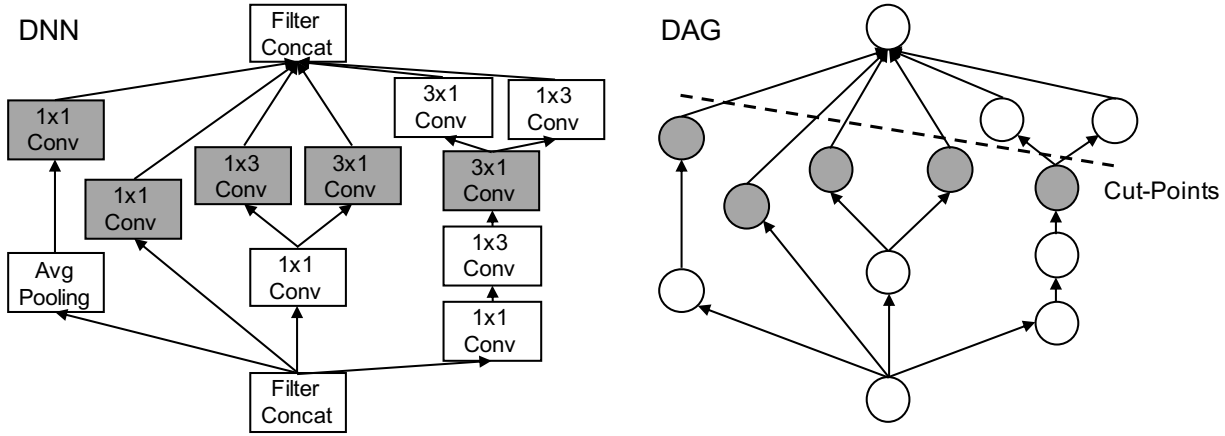
Model

- Chain Structure DNNs
  - E.g.: AlexNet
  - Job shop scheduling
  - NP-hard
- DAG Structure DNNs
  - E.g.: GoogLeNet
  - DAG shop scheduling
  - NP-hard



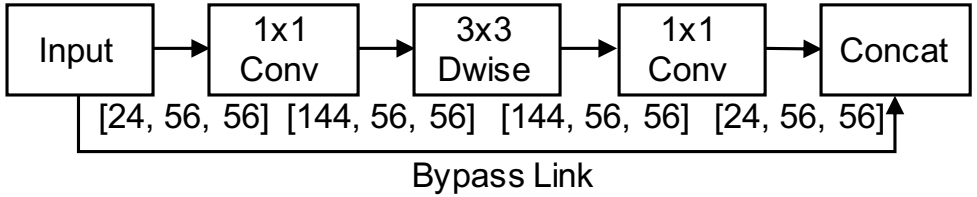
Comp.: forward propagation before cut-point layers

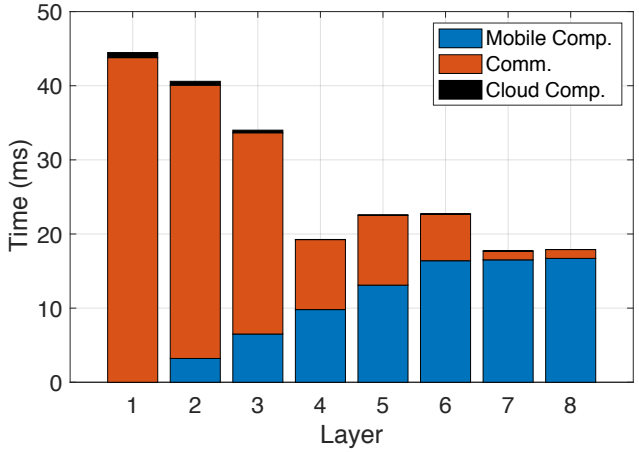
Comm.: send outputs of cut-point layers

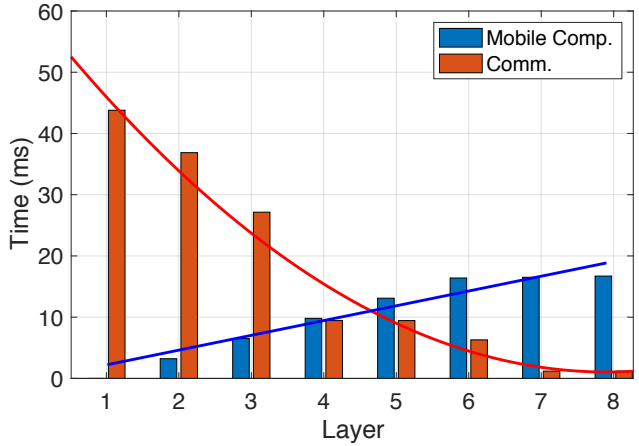


Chain Structure

- Observations
  - Mobile comp. time increases with layers
    - comp. workloads of layers are similar
  - Comm. time decreases with layers
    - pooling usually reduces tensor size by half
    - group layers if tensor sizes are not reduced
  - Cloud comp. time is negligible

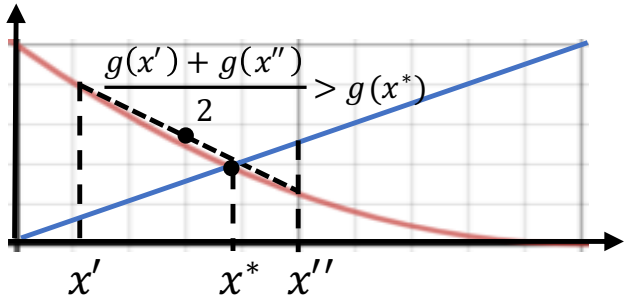


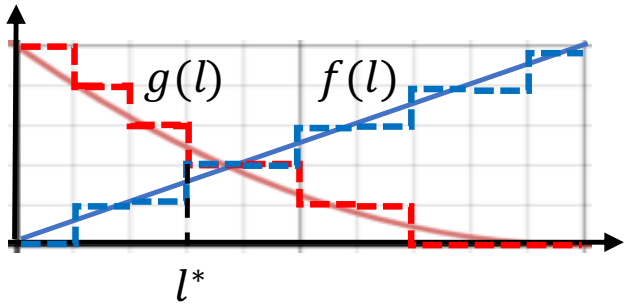




Chain Structure (cont'd)

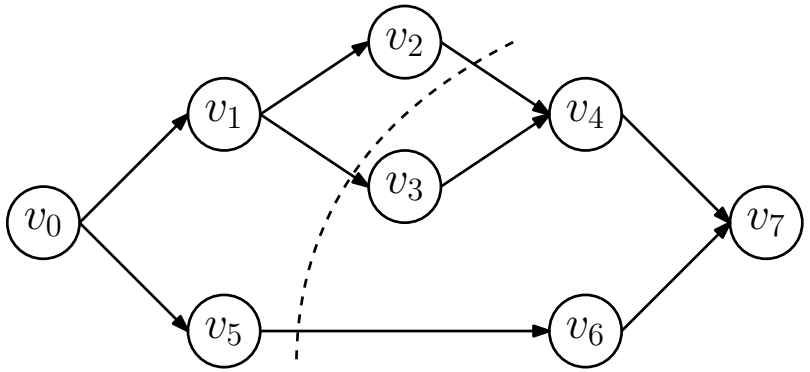
- Assumptions
  - Mobile comp. time: linear functions
  - Comm. time: convex functions
- Relax Problem to Continuous Domain
  - Optimal cut-point x^* for all DNNs:
    - mobile comp. time = comm. time
- Original Problem in Discrete Domain
  - Two types of cuts:
    - Comm.-heavy: left to x^*
    - Comp.-heavy: right to x^*
    - Adjust ratio of two types of cuts to fill the gap

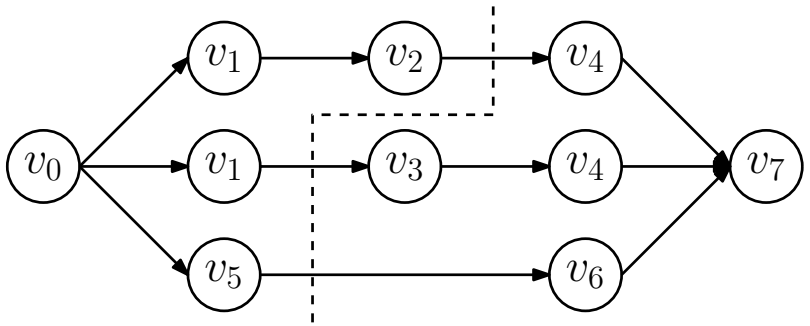




DAG Structure

- Convert to Multi-Path DAG
  - Duplicate fork and join nodes
- Partition and Schedule Each Path
  - Treat each path as an independent task
  - Include duplicated nodes for scheduling
  - Exclude duplicated nodes for execution
    - Memorize layer outputs with hash tables





Experiment

- Testbed
  - Mobile:
    - Raspberry Pi 4 model B
  - Cloud:
    - Lab server (i7, GTX1080, 32GB)

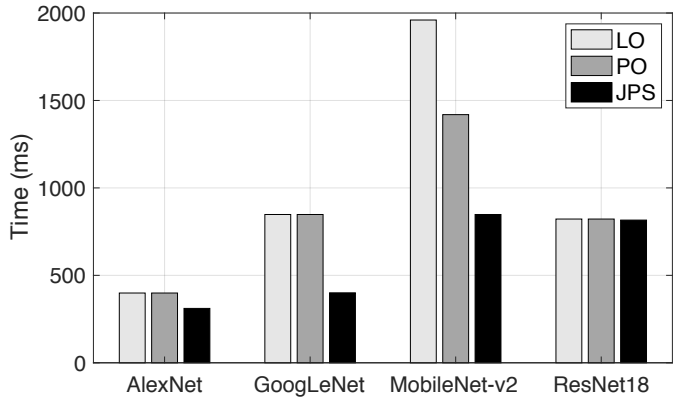
- Comparison Algorithms
  - LO: local only
  - CO: cloud only
  - PO: partition only (state-of-the-art)
  - JPS: joint partition and scheduling

Model	3G		4G		Wi-Fi	
	PO	JPS	PO	JPS	PO	JPS
AlexNet	0	**22.06**	33.33	**42.11**	63.91	**73.43**
MobileNet-v2	27.60	**56.73**	60.00	**78.83**	82.81	**84.69**
GoogLeNet	0	**52.83**	56.13	**71.93**	66.63	**72.17**
ResNet18	0	**0.73**	1.46	**28.22**	58.52	**58.52**

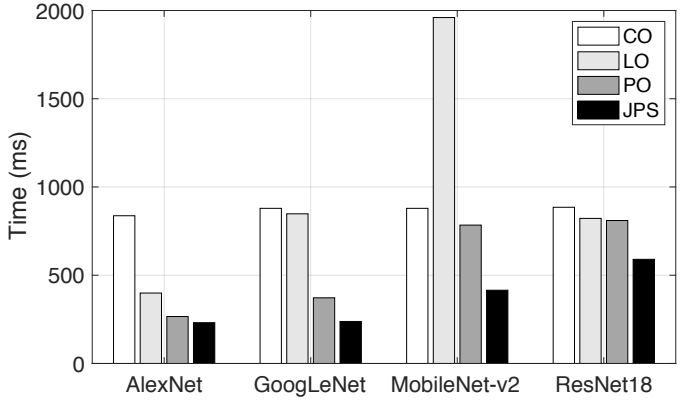
Latency reduction ratio compared with LO

Experiment Results

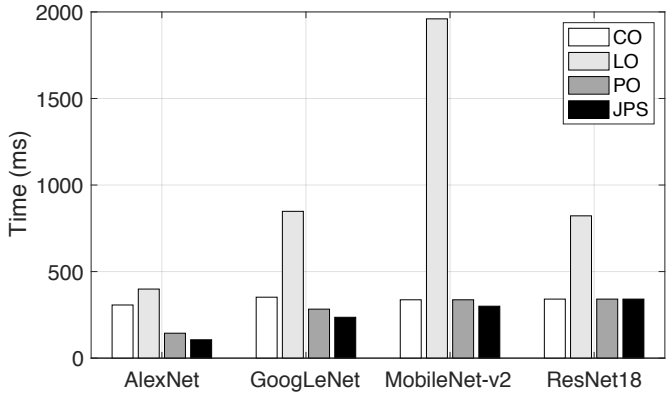
- Significantly reduced latency with negligible scheduling overhead



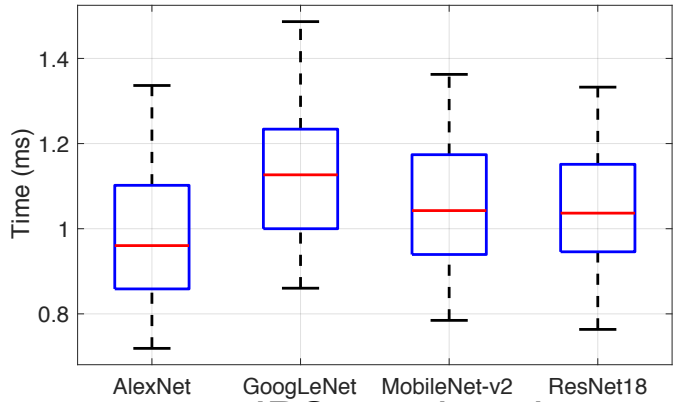
3G (1.1 Mbps)



4G (5.85 Mbps)



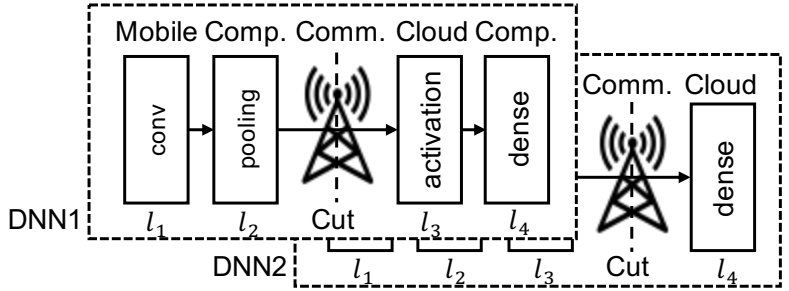
Wi-Fi (18.88 Mbps)

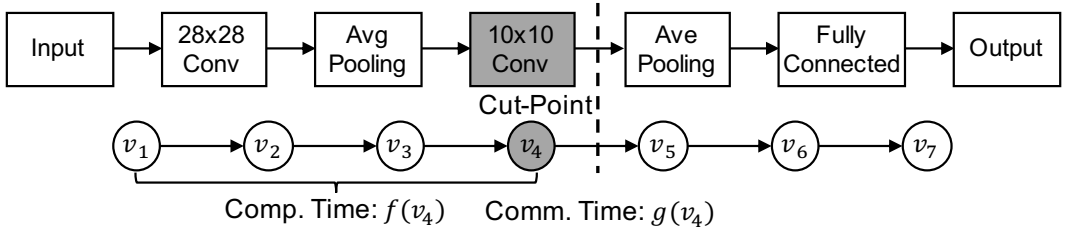


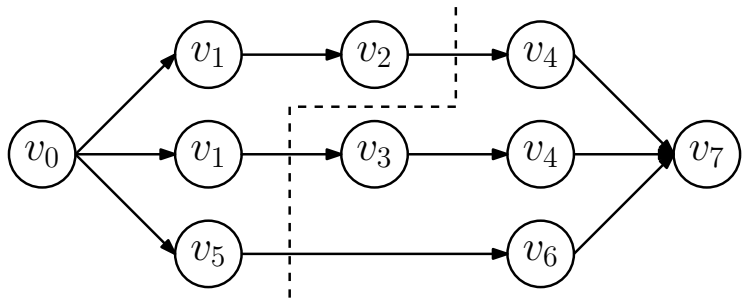
JPS overhead

Conclusion

- Joint Partition and Scheduling
  - Cooperative DNN inference
  - Reduce DNN inference latency
- Chain-Structure DNNs
  - Relax problem with optimal solution
  - Two types of DNN cuts
- DAG-Structure DNNs
  - Multi-path DNN conversion







Q&A

yubin.duan@temple.edu