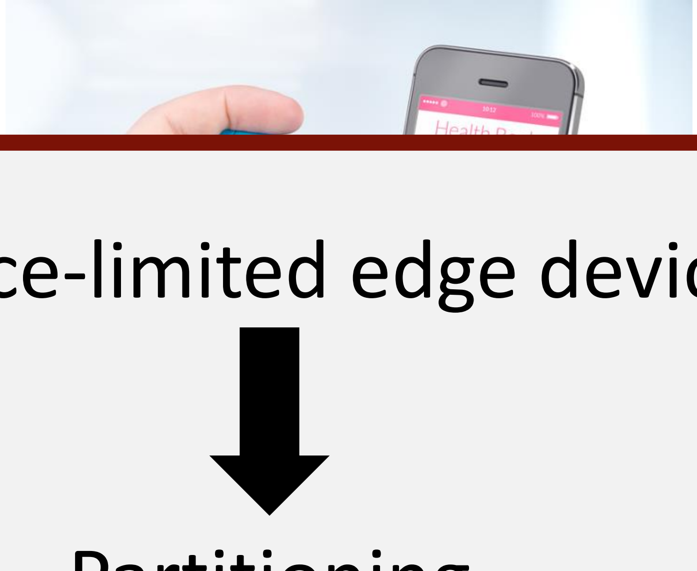
PORTEND: A Joint Performance Model for Partitioned Early-Exiting DNNs

Maryam Ebrahimi
Alexandre da Silva Veith
Moshe Gabel
Eyal de Lara

UNIVERSITY OF
TORONTO

The 29th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2023)

Edge and mobile applications, need fast DNN inference

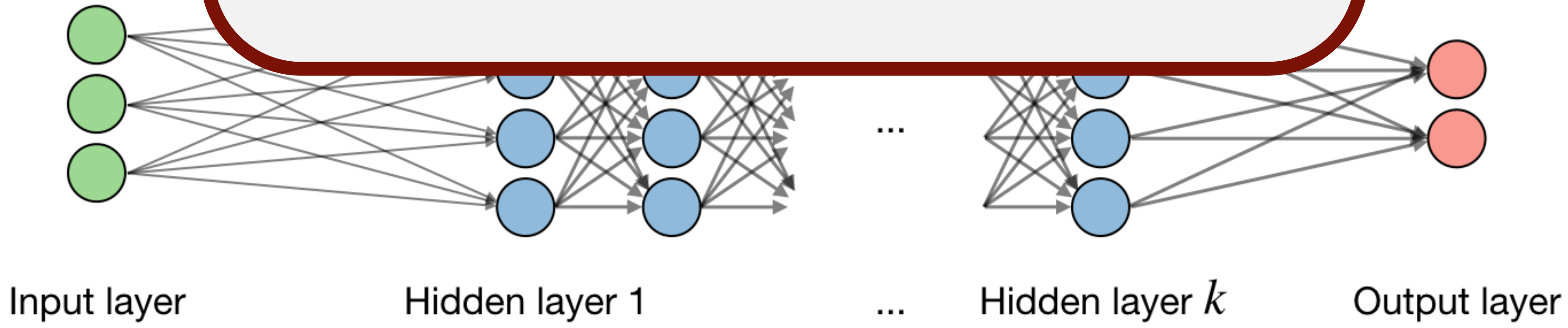


Resource-limited edge devices

↓

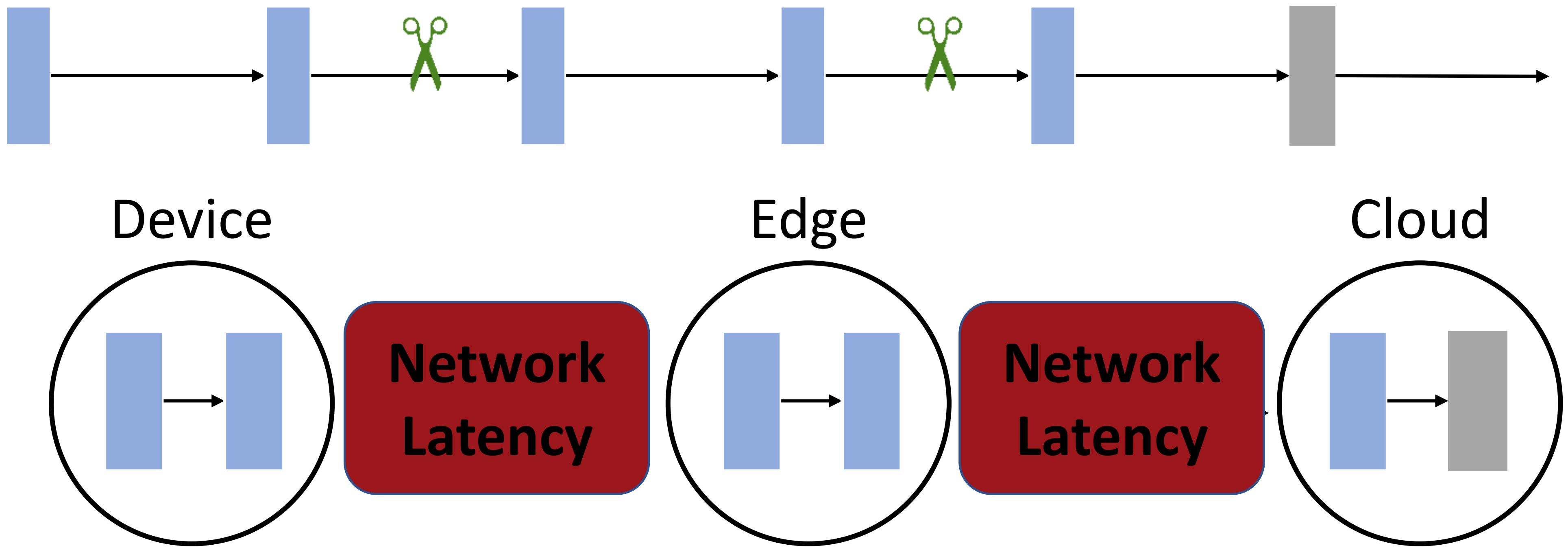
Partitioning

Early Exit



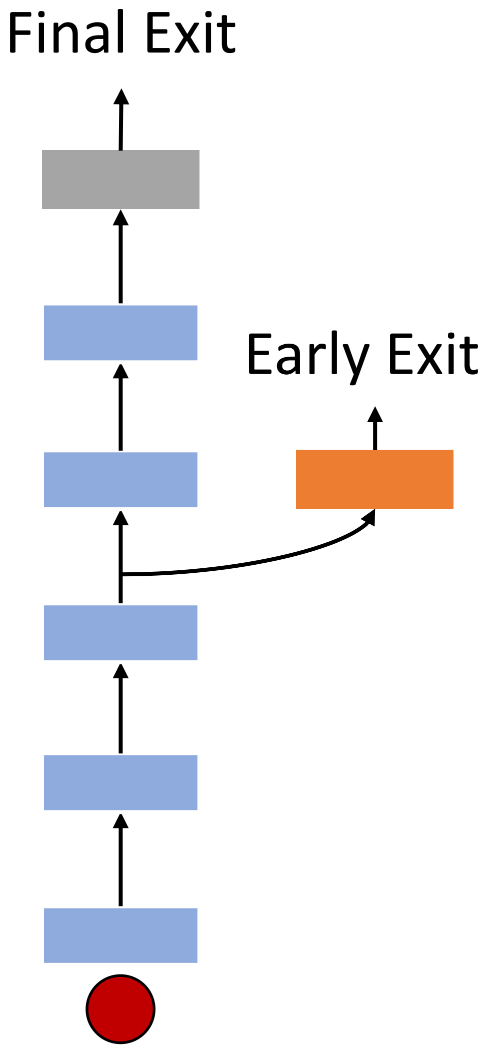
Background: Partitioning

Split pre-trained DNN across multiple servers



This approach trades-off computation time for network latency

Background: Early Exit



Add early exit branches as auxiliary classifiers

Use entropy to measure the confidence of the prediction

Entropy(output) < Threshold ⟶ Exit

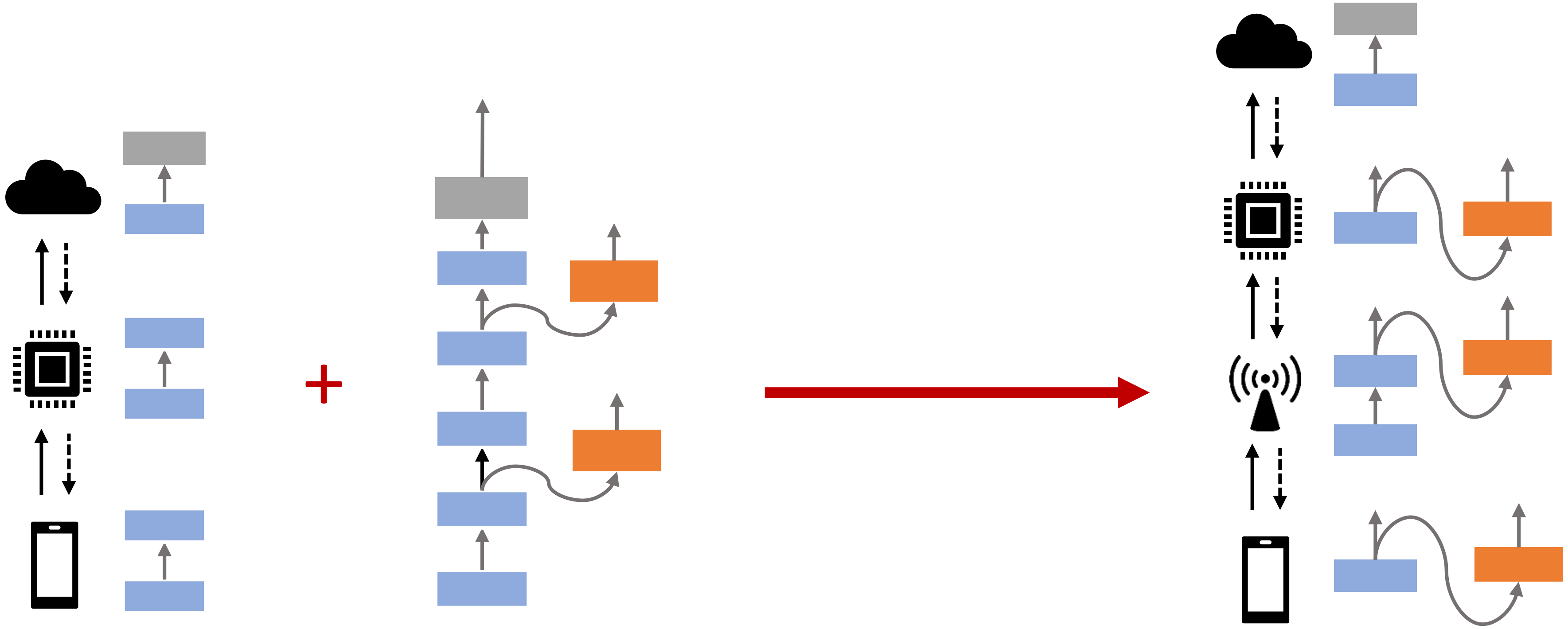
Entropy(output) > Threshold ⟶ Stay

This approach trades-off accuracy for computation latency

Combining DNN partitioning and early exit

Consider partitioning and early exit holistically

- A performance model that jointly optimizes them



Partitioning

Early Exit

Partitioning + Early Exit

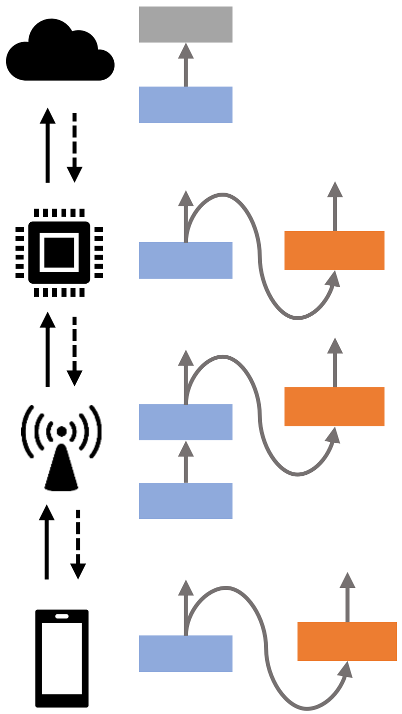
Combining DNN partitioning and early exit

1. Where to *partition* the model

   and attach the early exit branches?

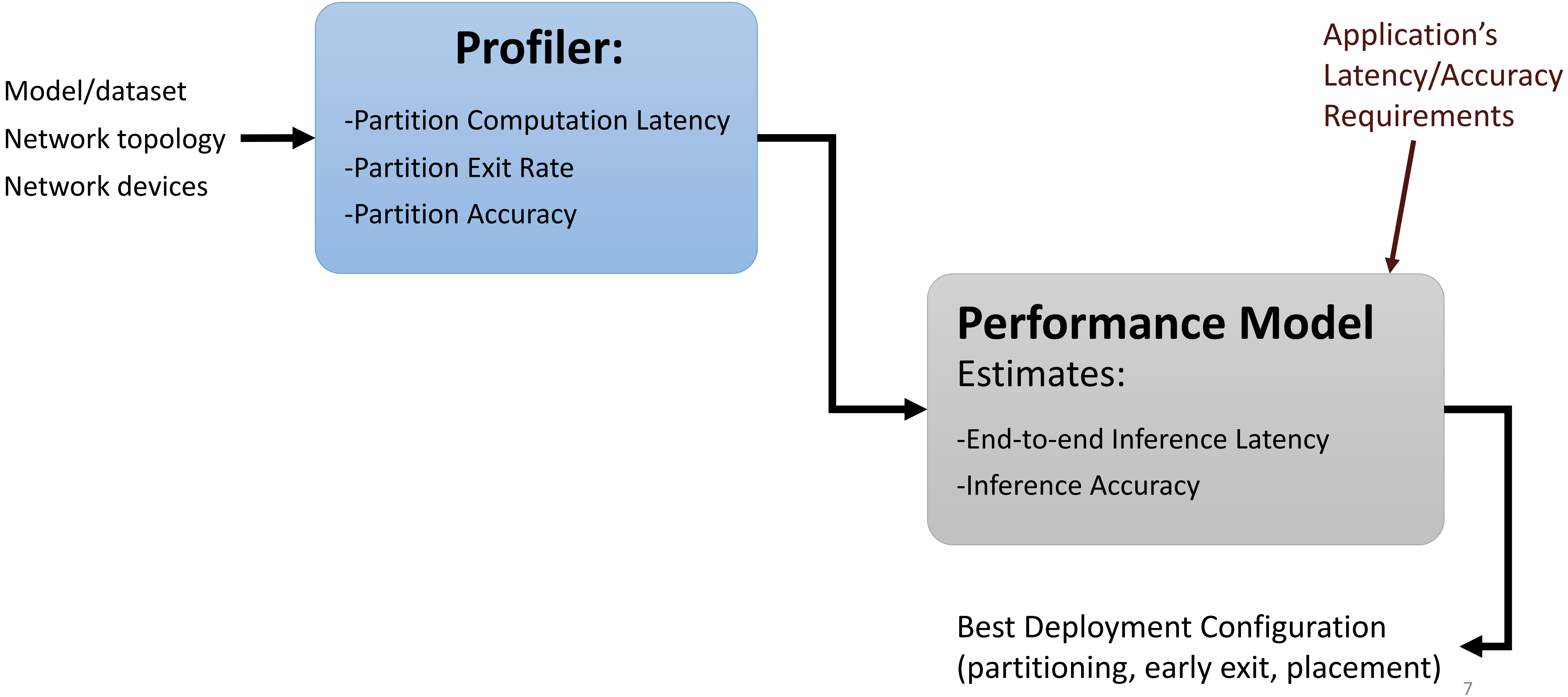
2. How to *place* these partitions on network devices?

- Calculate the inference latency and accuracy
- Objective: minimizing latency/maximizing accuracy



Partitioning + Early Exit

PORTEND



Model/dataset

Network topology

Network devices

Profiler:

-Partition Computation Latency

-Partition Exit Rate

-Partition Accuracy

Application’s
Latency/Accuracy
Requirements

Performance Model

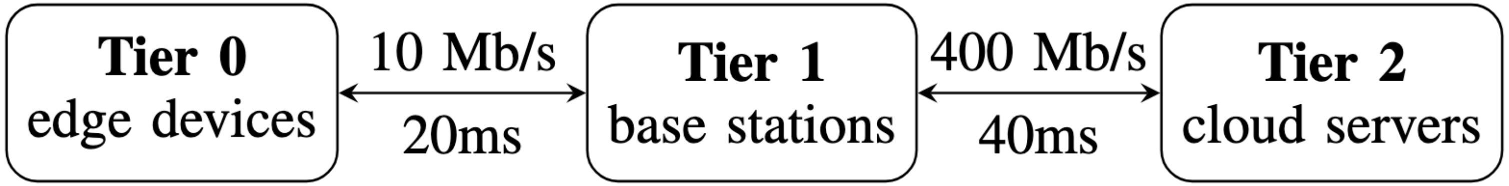
Estimates:

-End-to-end Inference Latency

-Inference Accuracy

Best Deployment Configuration
(partitioning, early exit, placement)

Experimental Setup



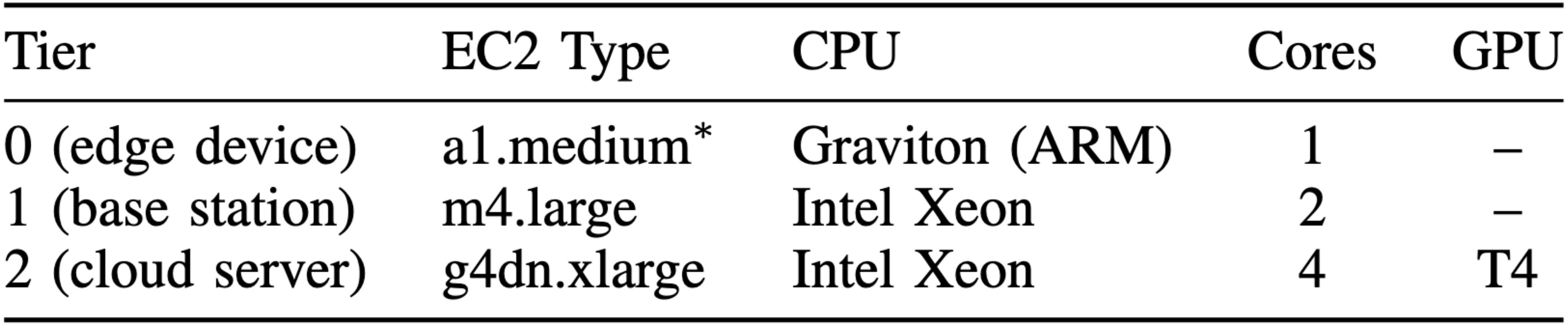
Tier 0 edge devices

10 Mb/s 20ms

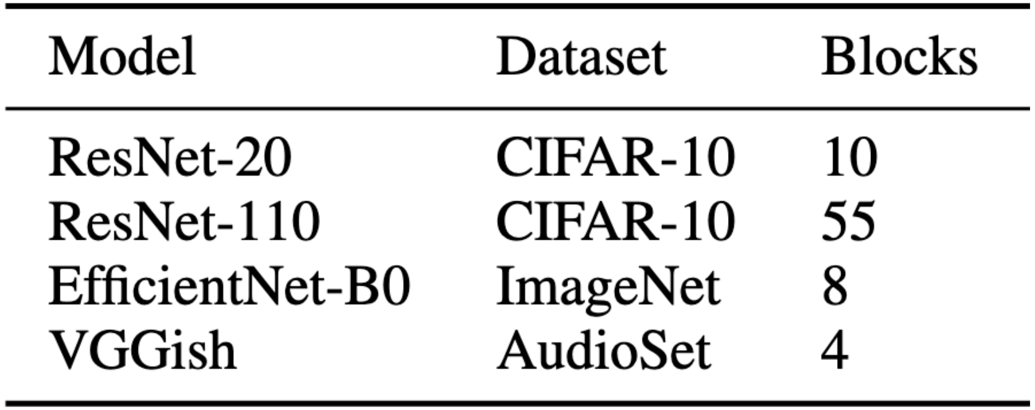
Tier 1 base stations

400 Mb/s 40ms

Tier 2 cloud servers

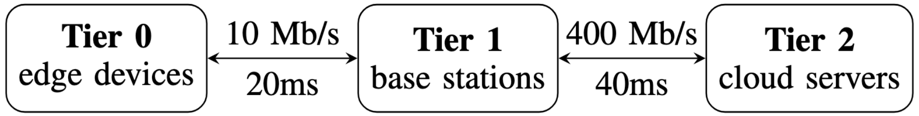


Tier	EC2 Type	CPU	Cores	GPU
0 (edge device)	a1.medium*	Graviton (ARM)	1	–
1 (base station)	m4.large	Intel Xeon	2	–
2 (cloud server)	g4dn.xlarge	Intel Xeon	4	T4



Model	Dataset	Blocks
ResNet-20	CIFAR-10	10
ResNet-110	CIFAR-10	55
EfficientNet-B0	ImageNet	8
VGGish	AudioSet	4

Benefits of Multiple Partitions - ResNet-110



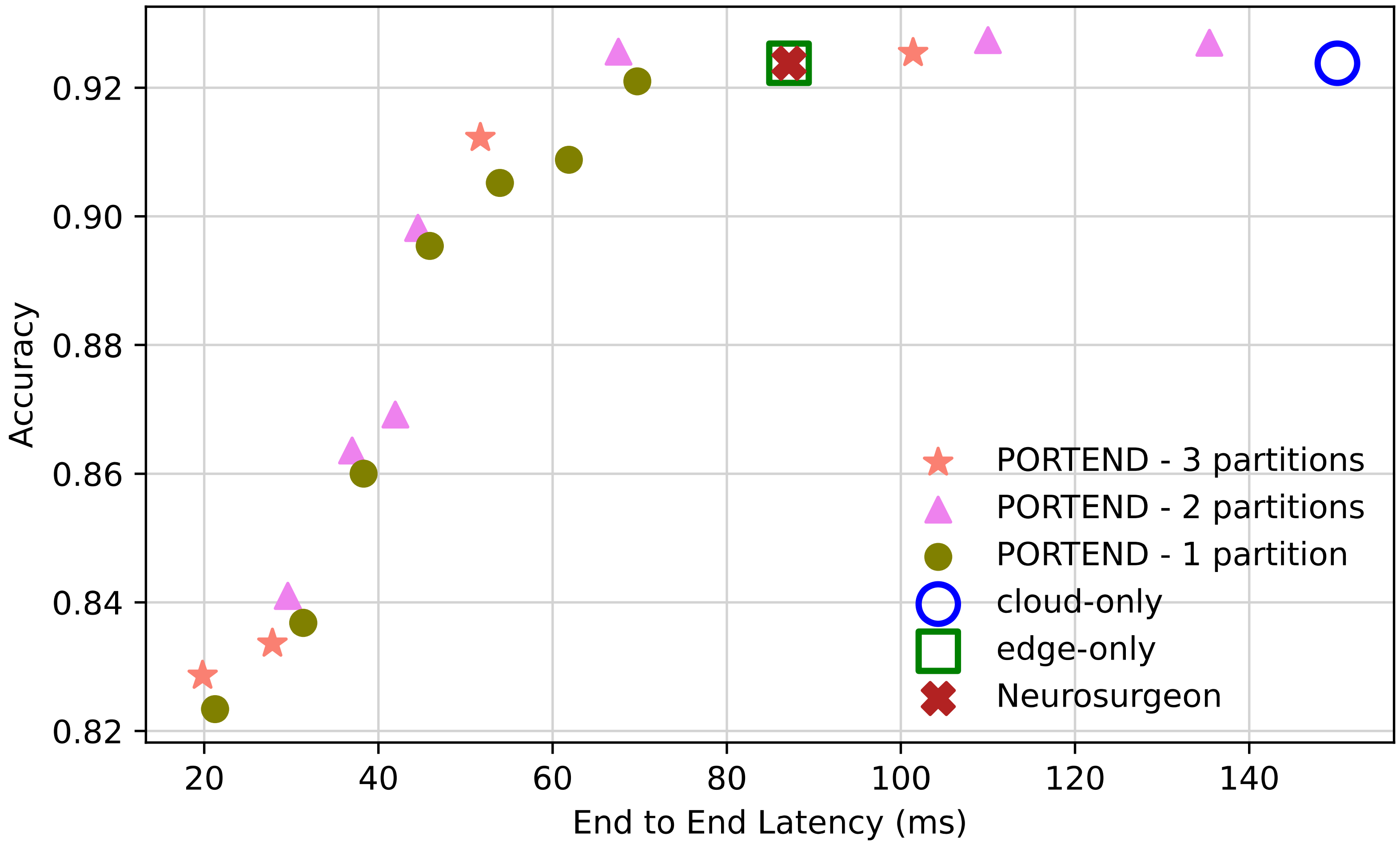
Tier 0 edge devices

10 Mb/s 20ms

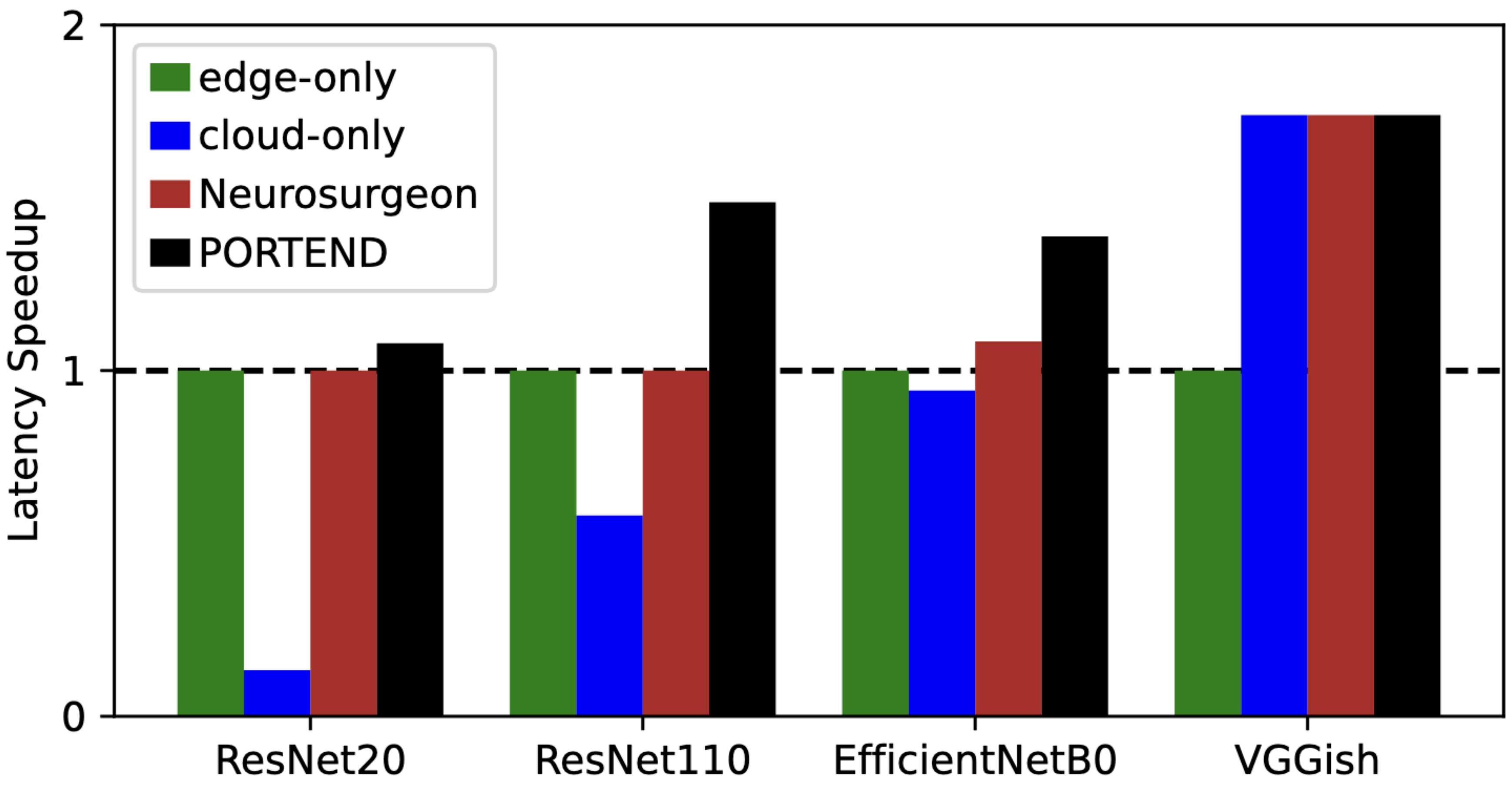
Tier 1 base stations

400 Mb/s 40ms

Tier 2 cloud servers

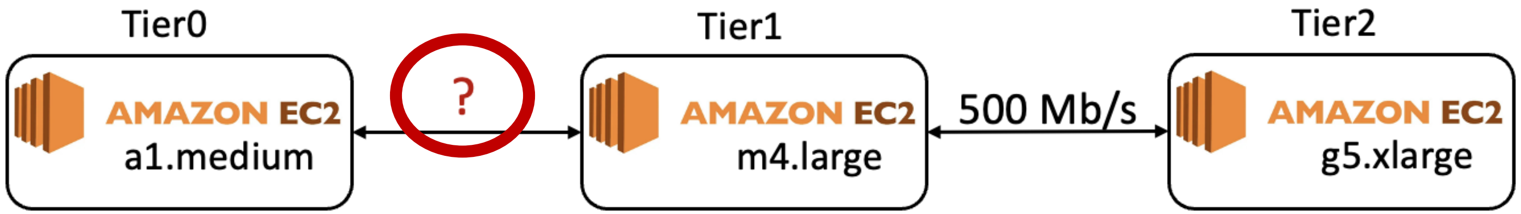


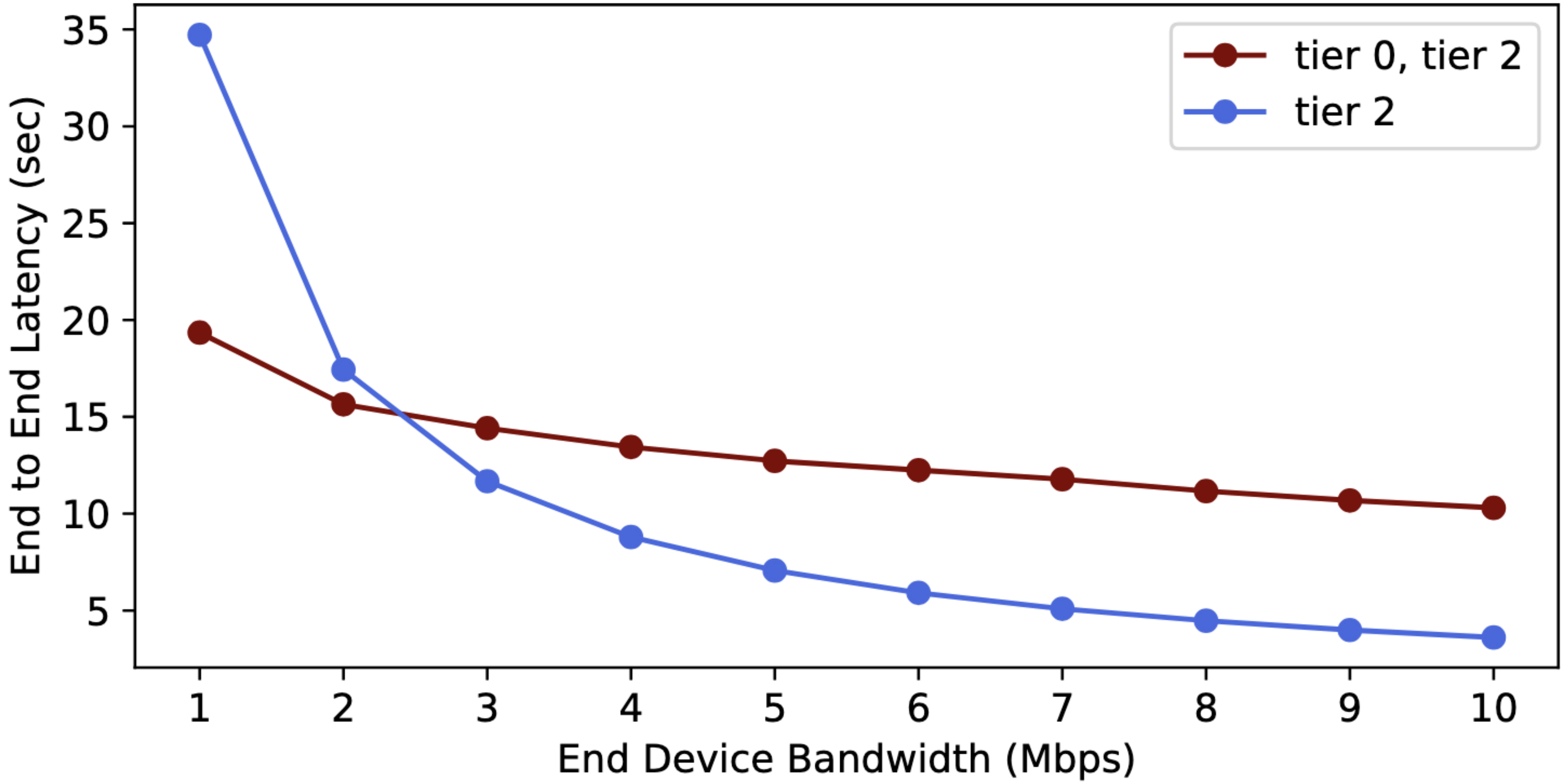
PORTEND Latency Speed-up



Exploring Hypothetical Scenarios

Reducing Bandwidth – EfficientNetB7





Conclusions & Future Work:

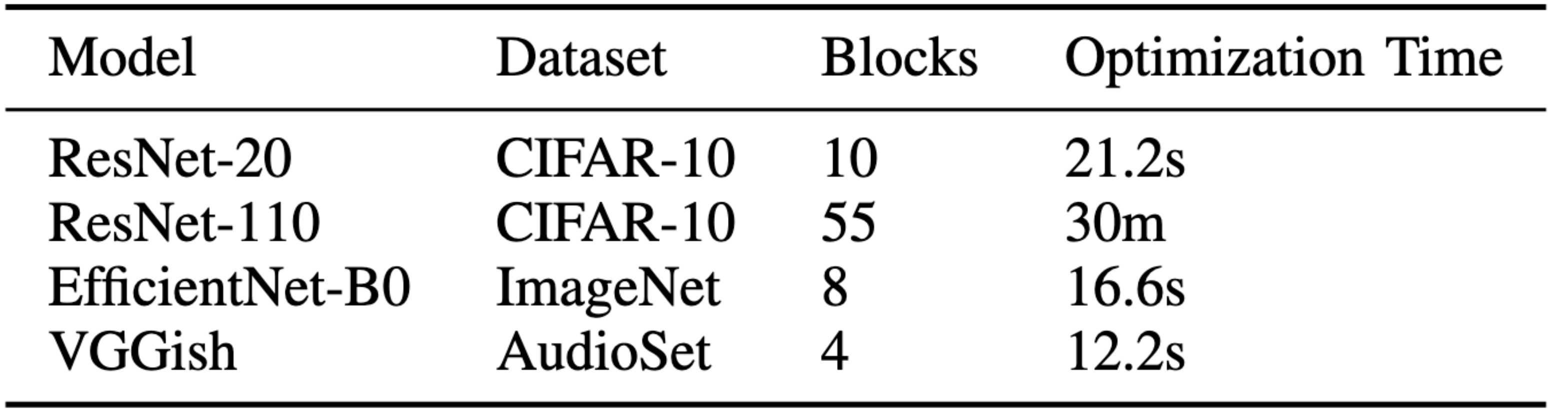
1. Considering multiple partitions with early exits improves the latency-accuracy trade-off.
2. Considering flexible placement is necessary.
3. Future work:
   1. Model compression and quantization
   2. Dynamic scheduling environment

Thank you!

Questions?

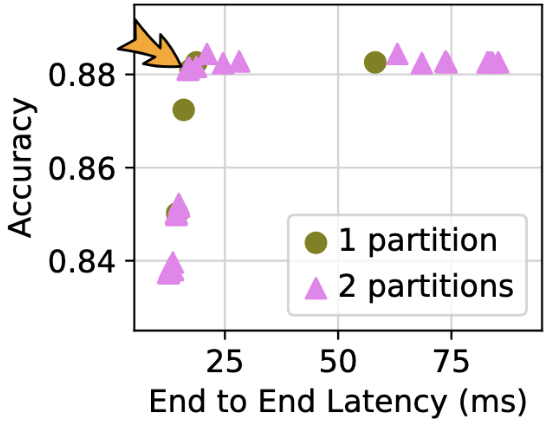
Experimental Setup

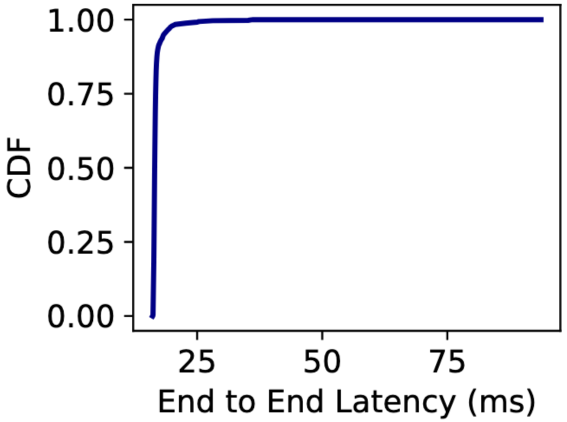
Image and Audio Classification Models

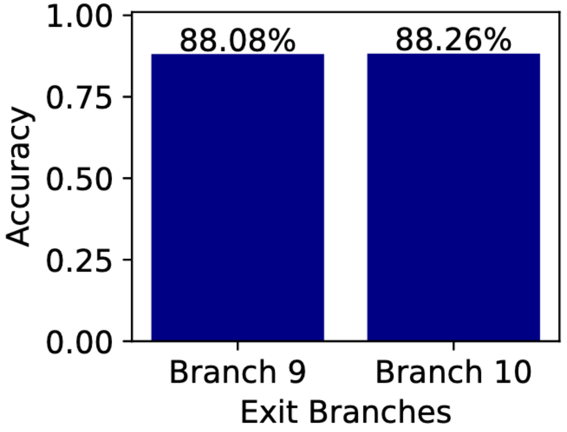


Model	Dataset	Blocks	Optimization Time
ResNet-20	CIFAR-10	10	21.2s
ResNet-110	CIFAR-10	55	30m
EfficientNet-B0	ImageNet	8	16.6s
VGGish	AudioSet	4	12.2s

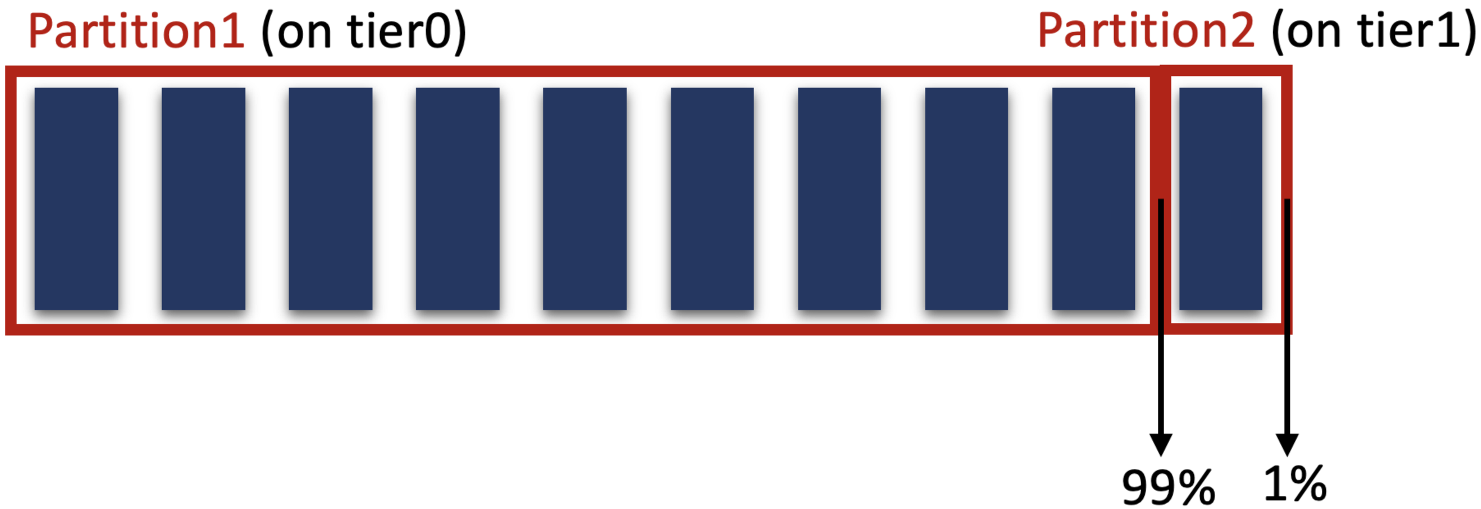
Finding Optimal Configurations- ResNet20







Model	Tiers [blocks]	Threshold	Exit rates
ResNet-20	0 [1–9] , 1 [10]	0.7	99% , 1%



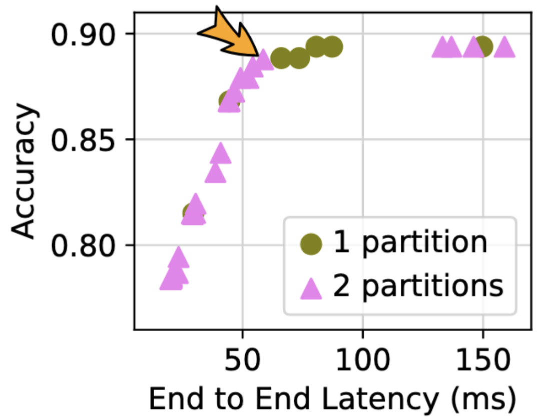
Partition1 (on tier0)

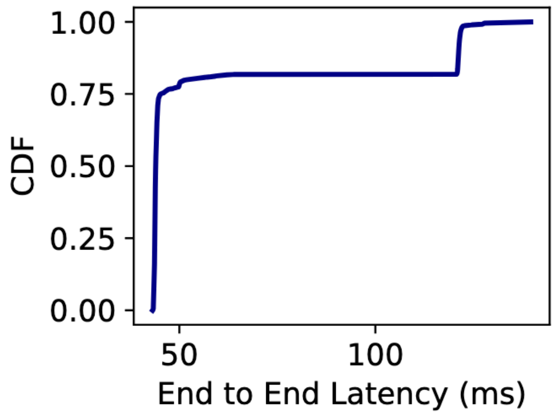
Partition2 (on tier1)

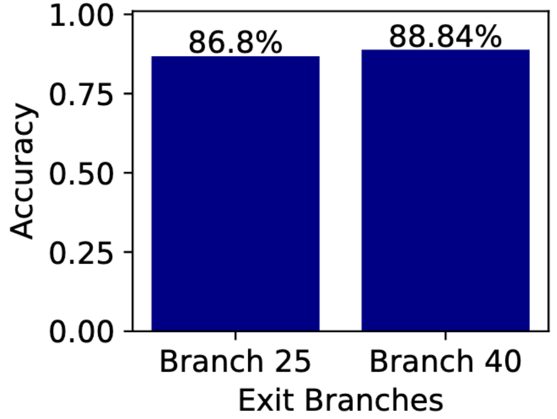
99%

1%

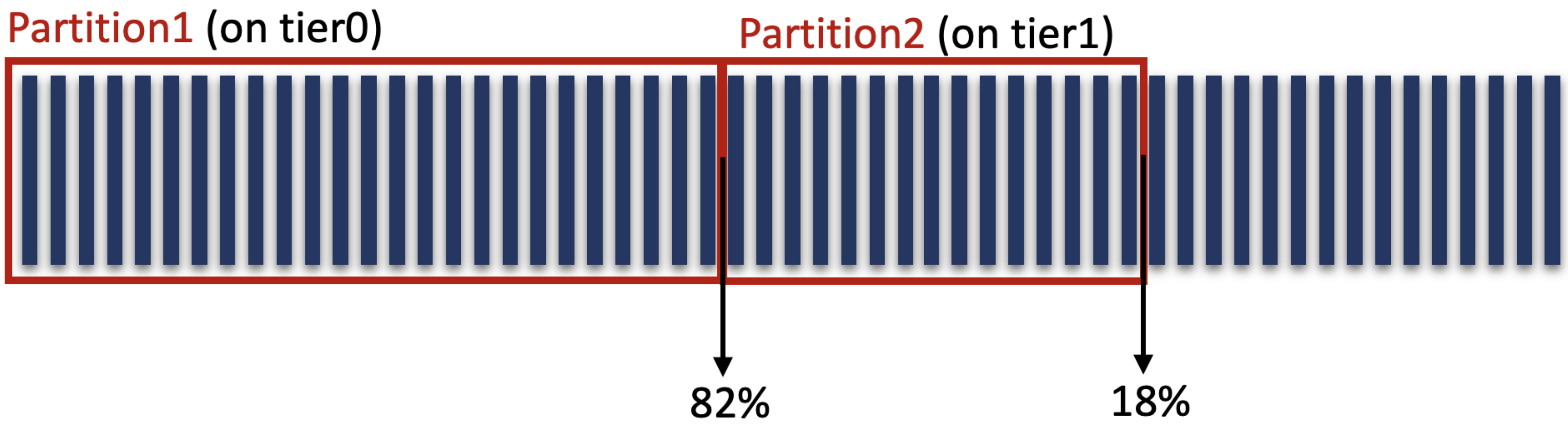
Finding Optimal Configurations- ResNet110







Model	Tiers [blocks]	Threshold	Exit rates
ResNet-110	0 [1–25] , 1 [26–40]	0.1	82% , 18%



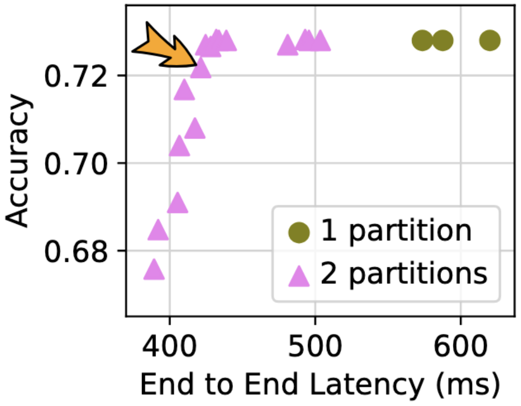
Partition1 (on tier0)

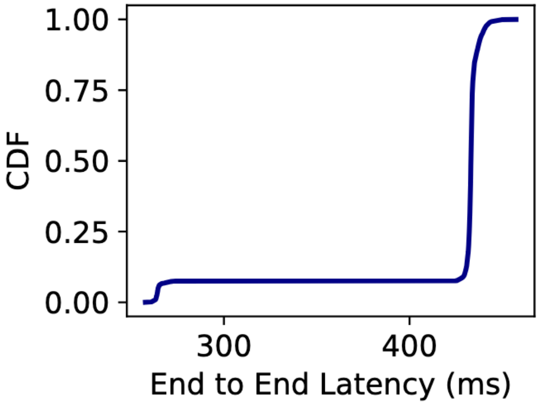
Partition2 (on tier1)

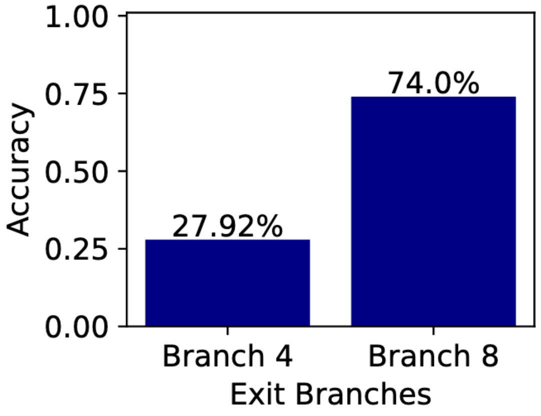
82%

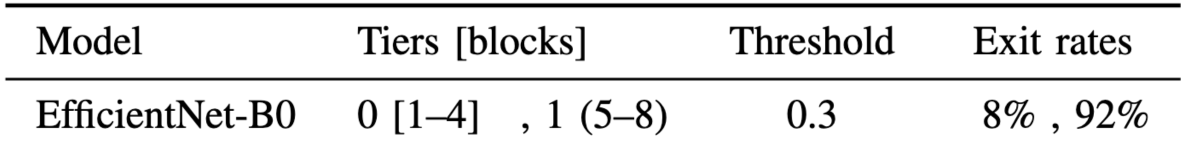
18%

Finding Optimal Configurations- EfficientNetB0

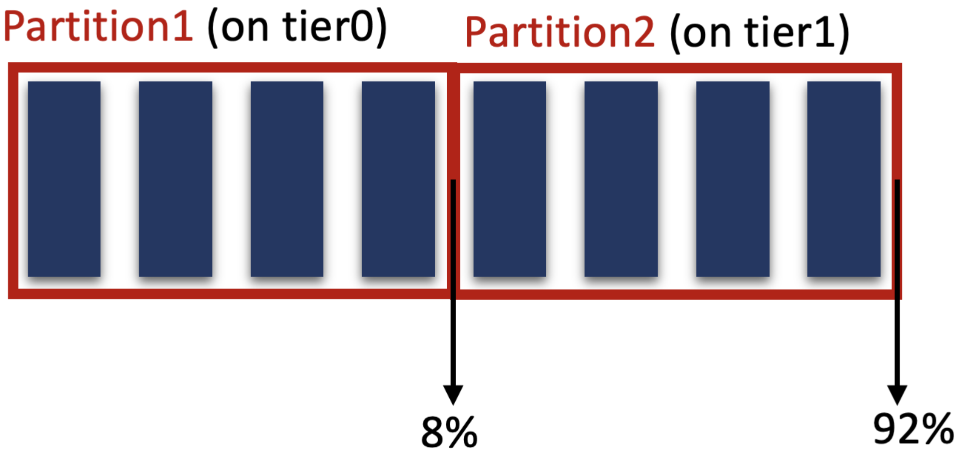








Model	Tiers [blocks]	Threshold	Exit rates
EfficientNet-B0	0 [1–4] , 1 (5–8)	0.3	8% , 92%



Partition1 (on tier0)

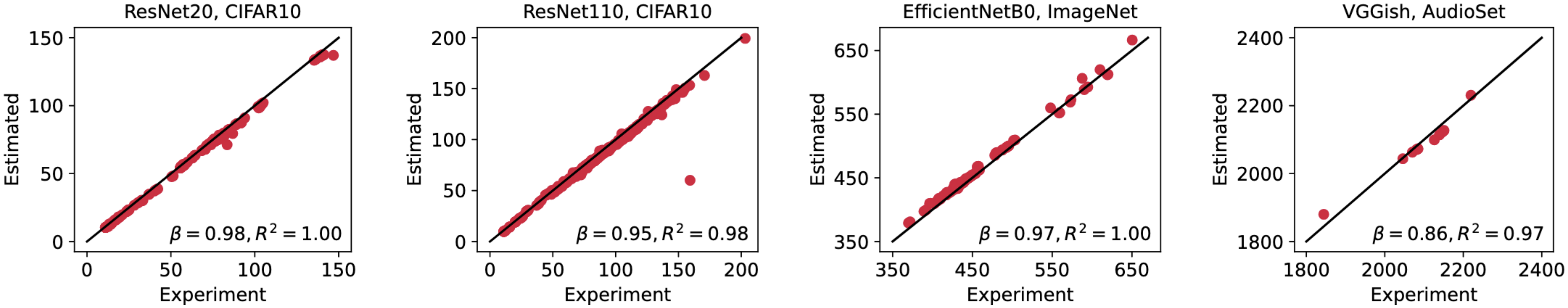
Partition2 (on tier1)

8%

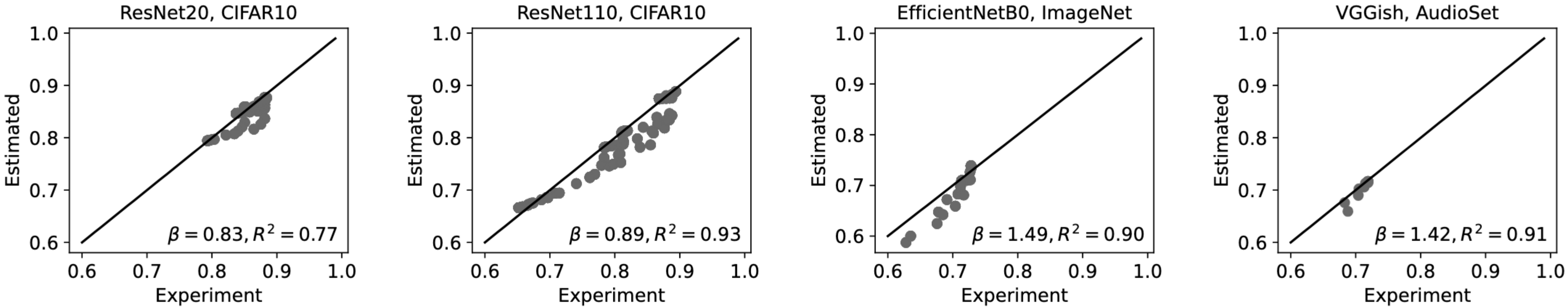
92%

Validity of the Performance Model

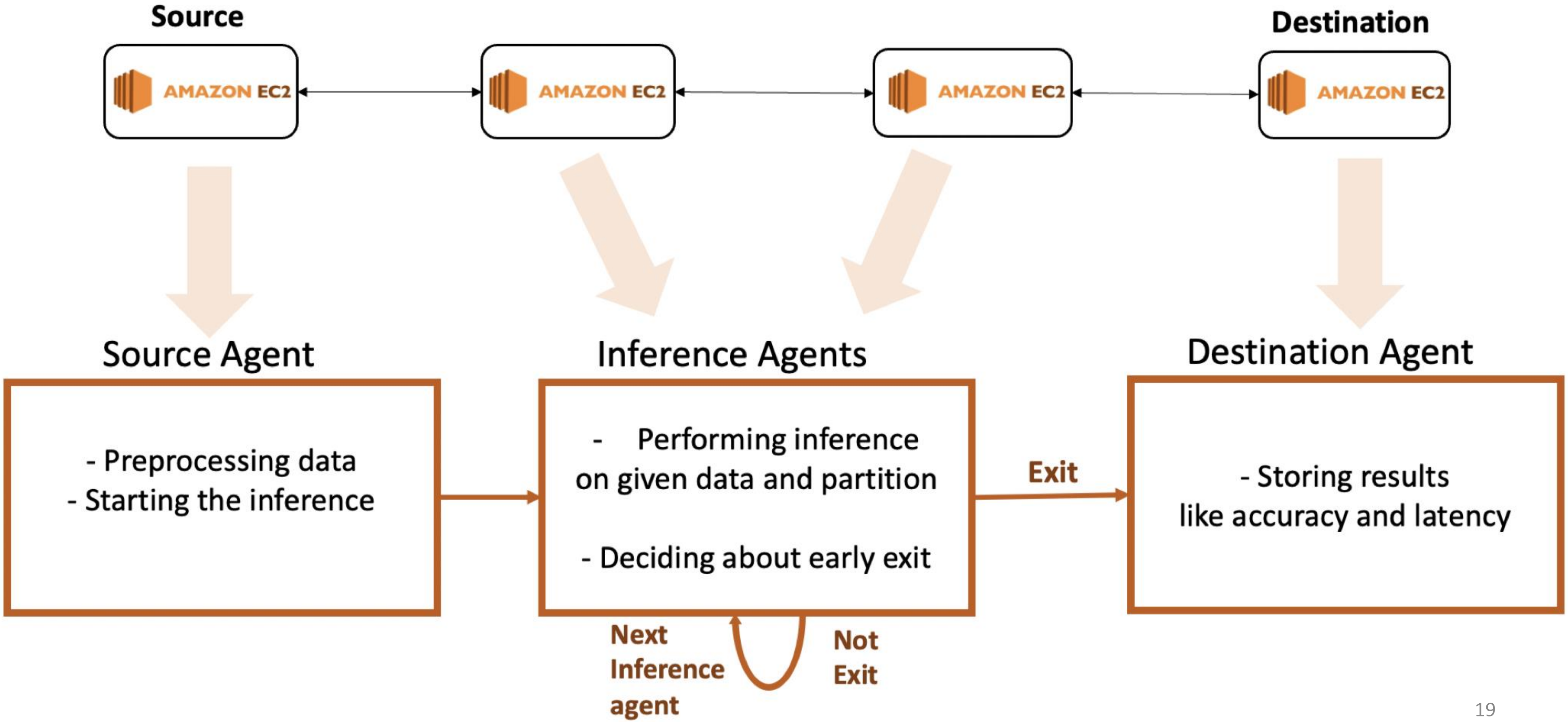
End-to-End Inference Latency



Inference Accuracy



Experimental Setup



Source Agent

- Preprocessing data
- Starting the inference

Inference Agents

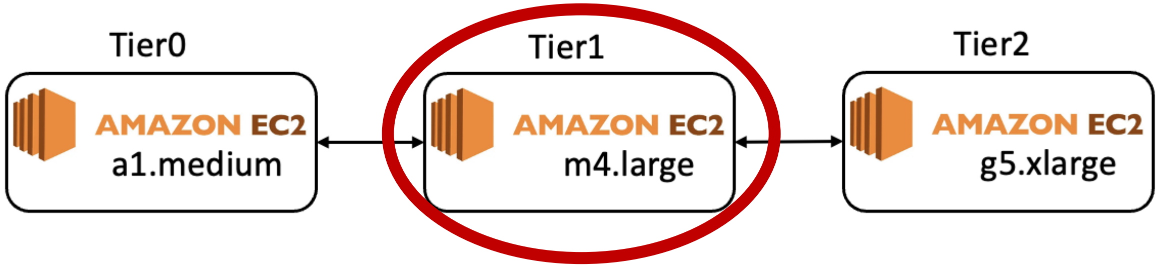
- Performing inference on given data and partition
- Deciding about early exit

Destination Agent

- Storing results like accuracy and latency

Exploring Hypothetical Scenarios

Increasing Compute Power – EfficientNetB7



Tier0

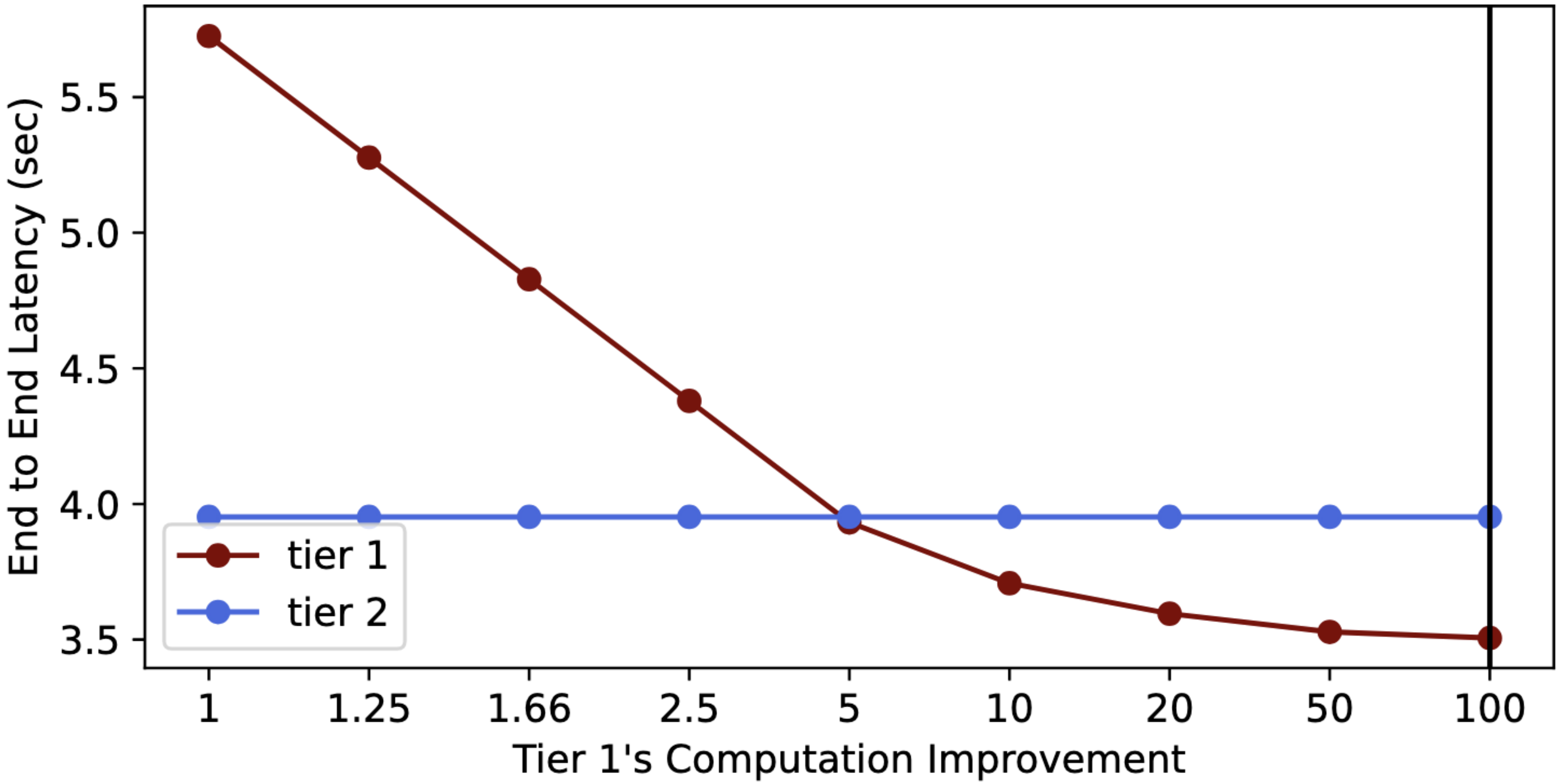
AMAZON EC2
a1.medium

Tier1

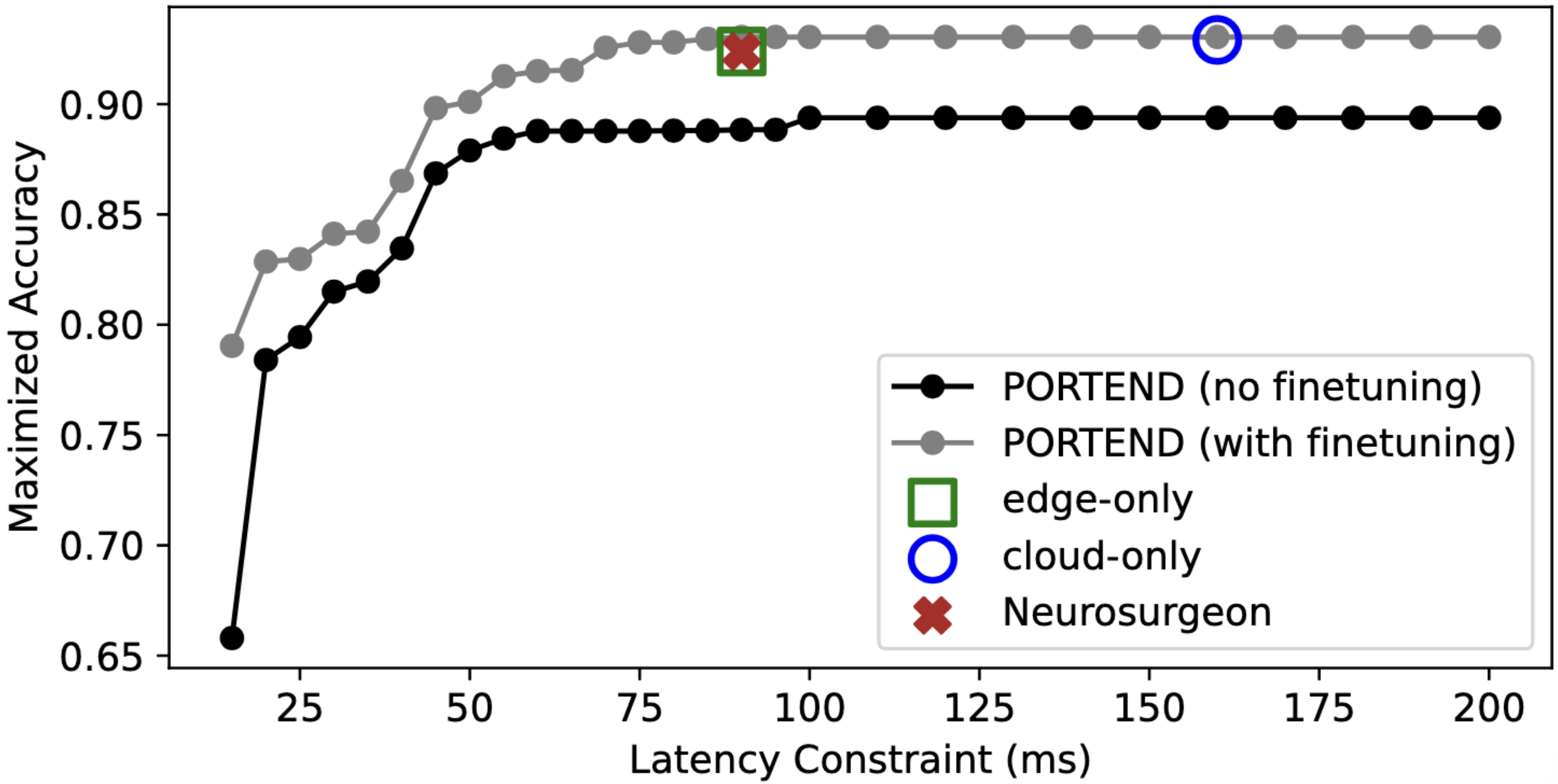
AMAZON EC2
m4.large

Tier2

AMAZON EC2
g5.xlarge



Effect of Fine-Tuning on final models


Maximized Accuracy
Latency Constraint (ms)
PORTEND (no finetuning)
PORTEND (with finetuning)
edge-only
cloud-only
Neurosurgeon