MCDNN: An Approximation-Based Execution Framework for Deep Stream Processing Under Resource Constraints

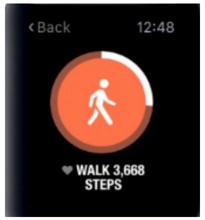
Haichen Shen (haichen@cs.washington.edu)

with Seungyeop Han, Matthai Philipose,

Sharad Agarwal, Alec Wolman, Arvind Krishnamurthy

University of Washington Microsoft Research

Wearable computing → more data



???

When computer vision meets wearable

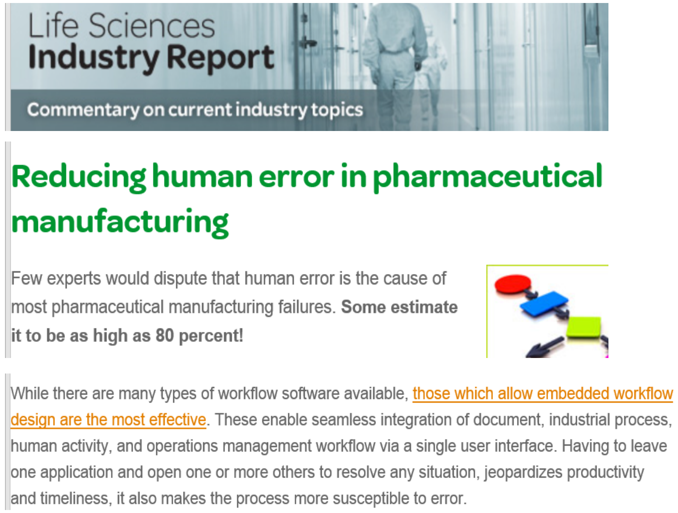
“That drink will get you to 2800 calories for today”

“I last saw your keys in the store room”

“Remind Tom of the party”

“You’re on page 263 of this book”

Consumer



Life Sciences
Industry Report

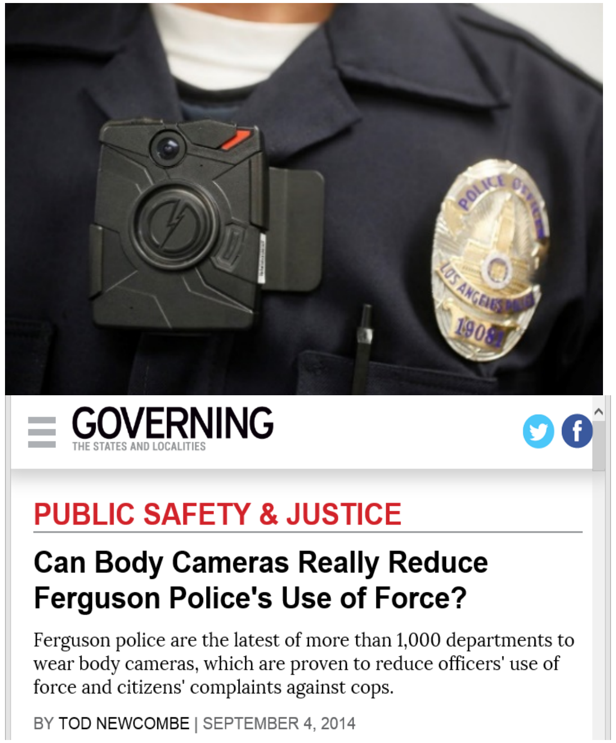
Commentary on current industry topics

Reducing human error in pharmaceutical manufacturing

Few experts would dispute that human error is the cause of most pharmaceutical manufacturing failures. **Some estimate it to be as high as 80 percent!**

While there are many types of workflow software available, those which allow embedded workflow design are the most effective. These enable seamless integration of document, industrial process, human activity, and operations management workflow via a single user interface. Having to leave one application and open one or more others to resolve any situation, jeopardizes productivity and timeliness, it also makes the process more susceptible to error.

Manufacturing



GOVERNING
THE STATES AND LOCALITIES

PUBLIC SAFETY & JUSTICE

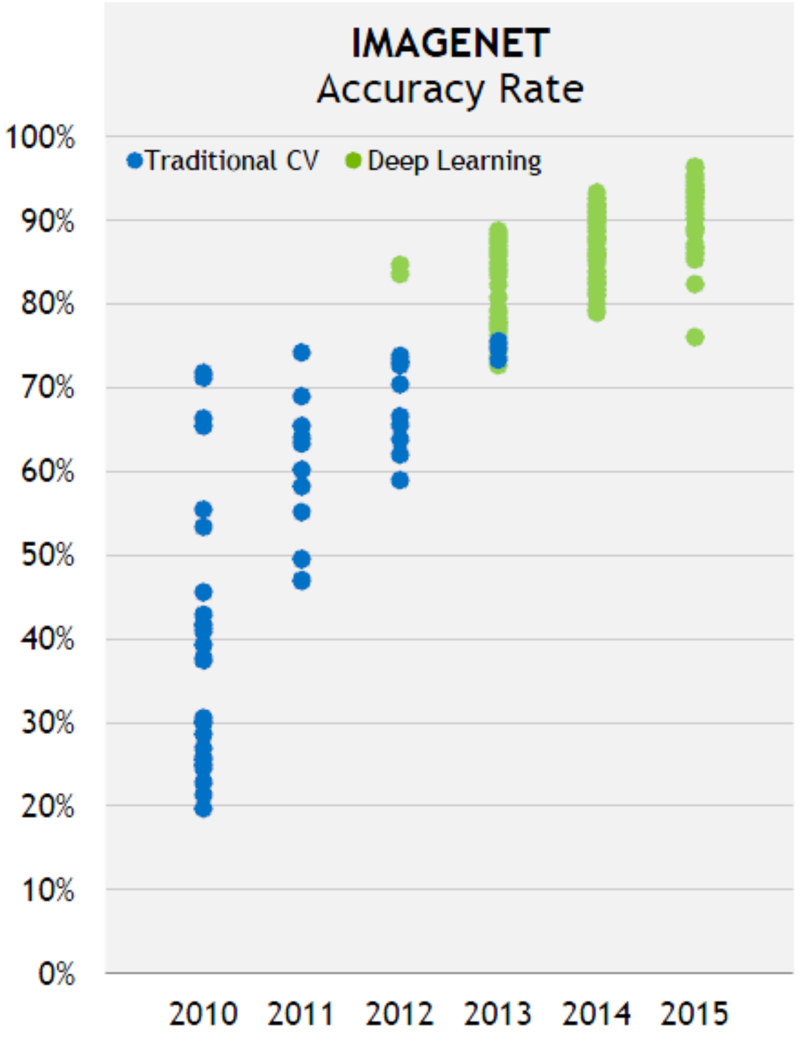
Can Body Cameras Really Reduce Ferguson Police's Use of Force?

Ferguson police are the latest of more than 1,000 departments to wear body cameras, which are proven to reduce officers' use of force and citizens' complaints against cops.

BY TOD NEWCOMBE | SEPTEMBER 4, 2014

Public Safety

Deep learning makes vision work



But...

Recognition Task	face	scene*	object*
Accuracy	97%	88%	92%
Compute/frame (FLOPs)	1.00G	30.9G	39.3G
Compute@1-30fps (FLOPS)	1-30G	30-900G	40G-1.2T

Do we have enough resources to run deep learning?

* top-5 accuracy is shown in the table

Resource usage for continuous vision

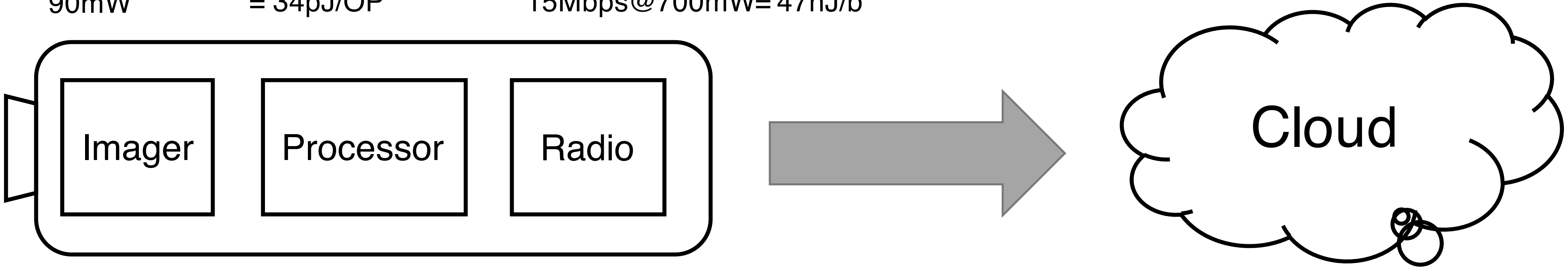
Omnivision
OV2740
90mW

Tegra K1 GPU
290GOPS@10W
= 34pJ/OP

Qualcomm SD810 LTE
>800mW
Atheros 802.11 a/g
15Mbps@700mW= 47nJ/b

Amazon EC2

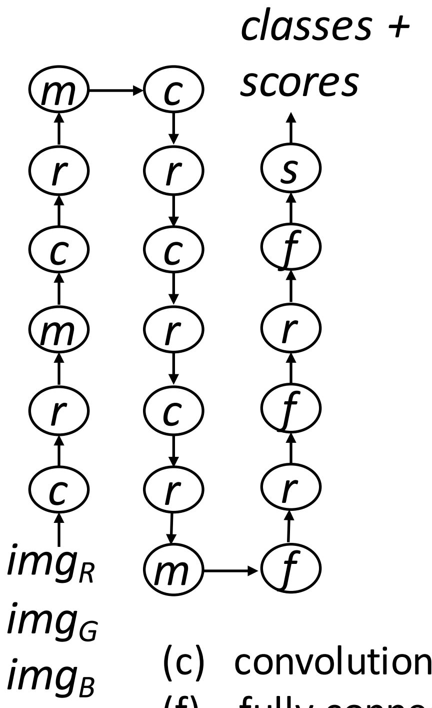
CPU	c4.large	2x400GFLOPS	$0.1/h
GPU	g2.2xlarge	2.3TFLOPS	$0.65/h



Workload	Deep learning **300GFLOPS** @ 30GFLOPs/frame, 10fps	
Budget	Device power 30% of 10Wh for 10h = 300mW	Cloud cost $10 person/year
Compute power	**9GFLOPS**	**3.5GFLOPS** (GPU) / **8GFLOPS** (CPU)

Huge gap between workload and budget

Neural network



(c) convolution

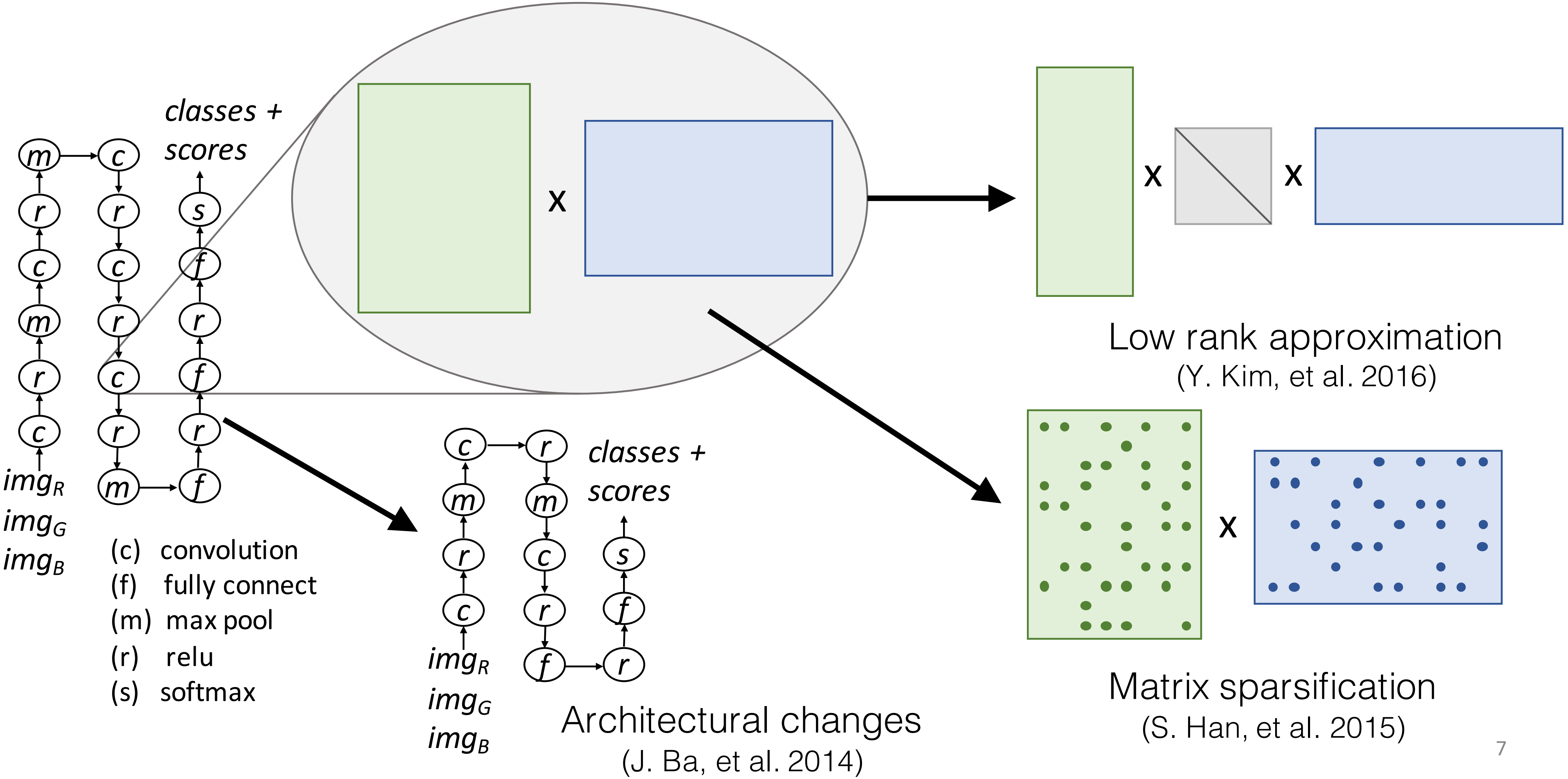
(f) fully connect

(m) max pool

(r) relu

(s) softmax

Neural network ≈ matrix multiplications



(c) convolution
(f) fully connect
(m) max pool
(r) relu
(s) softmax

Architectural changes
(J. Ba, et al. 2014)

Low rank approximation
(Y. Kim, et al. 2016)

Matrix sparsification
(S. Han, et al. 2015)

Managing the approx. / resource trade-off

- Detailed characterization of the approximation / resource trade-off for many optimizations

- Two new optimizations for streaming, multi-application settings

- New scheduling problem, Approximate Model Scheduling, with a heuristic solution

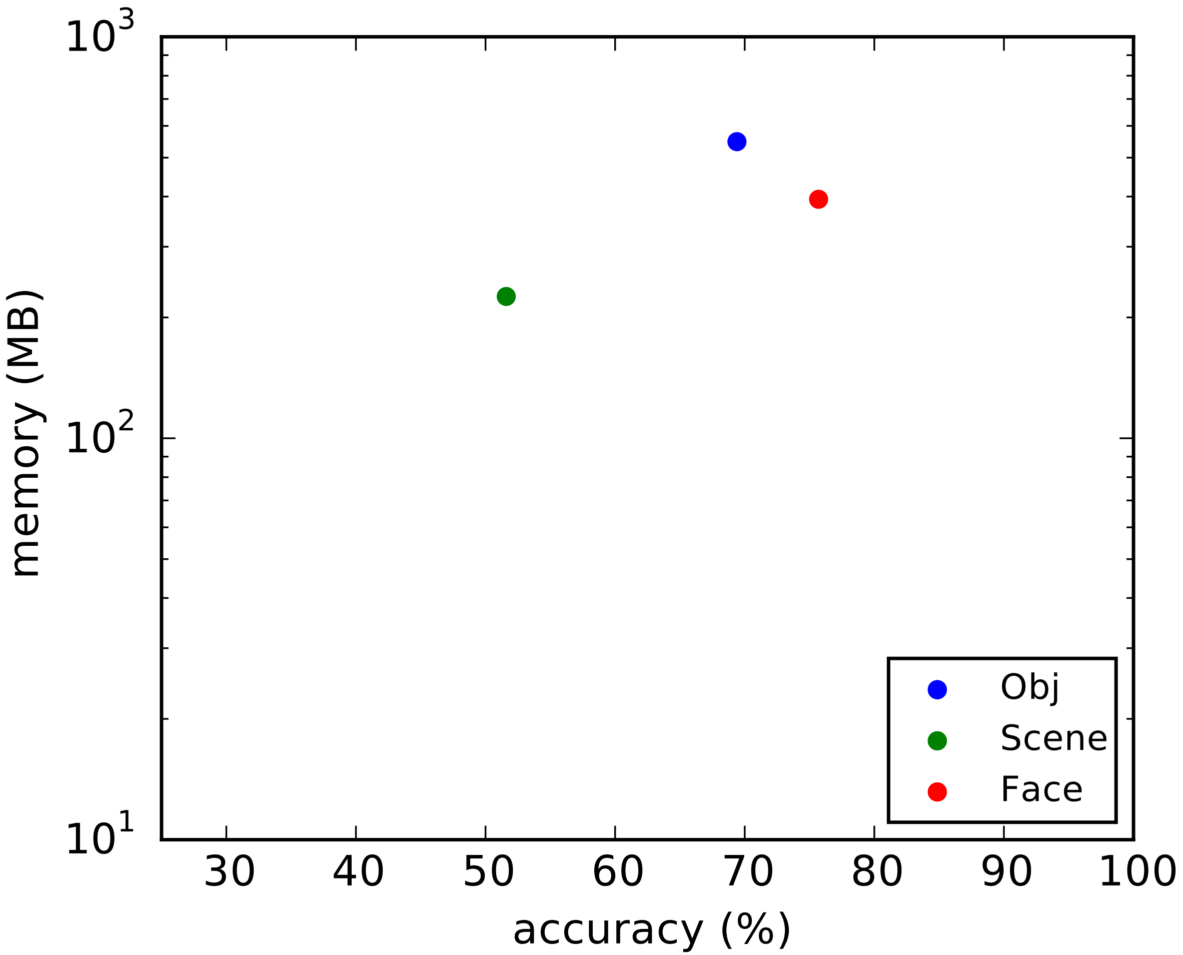
Outline

- Detailed characterization of the approximation / resource trade-off for many optimizations

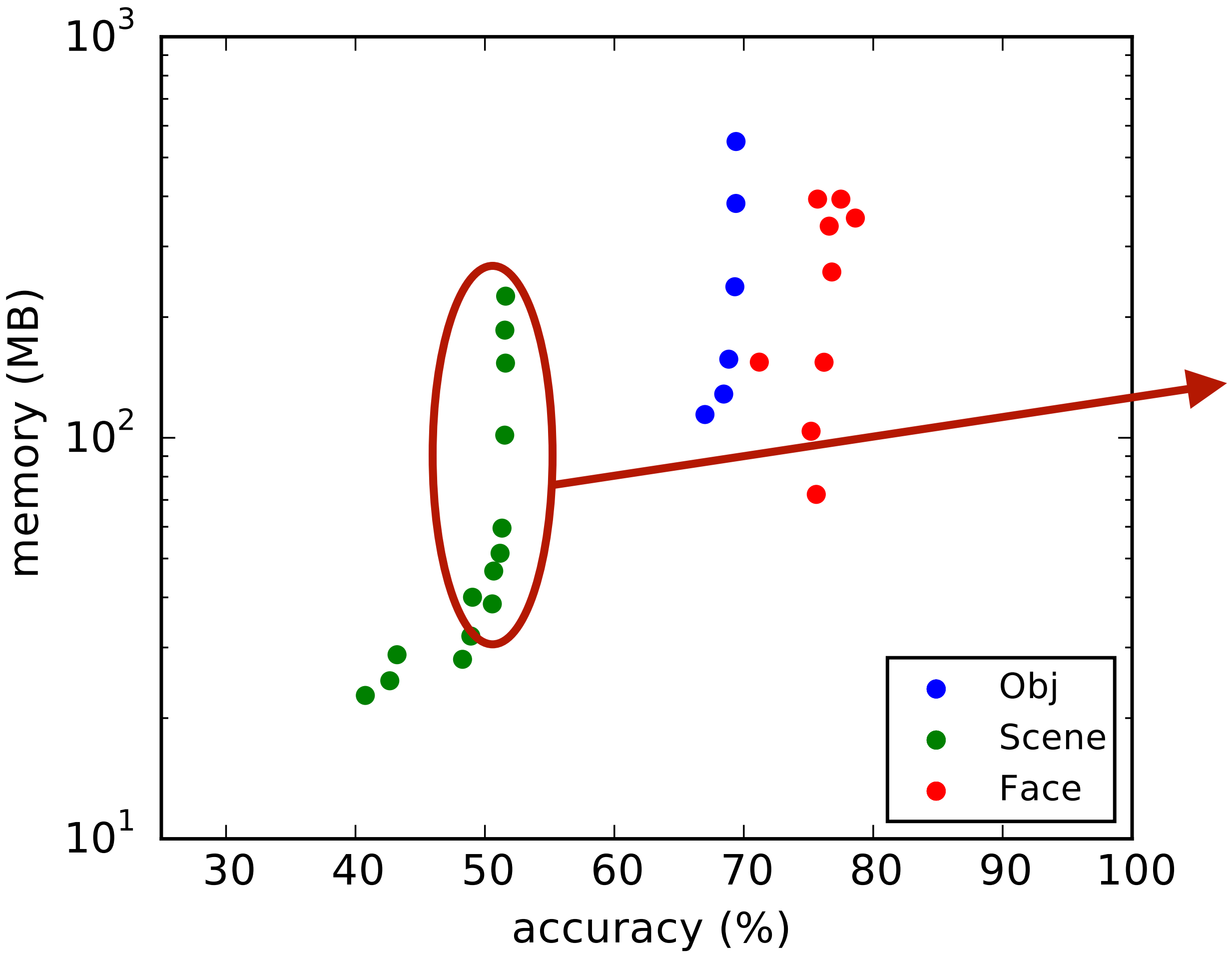
- Two new optimizations for streaming, multi-application settings

- New scheduling problem, Approximate Model Scheduling, with a heuristic solution

Memory / accuracy trade-off

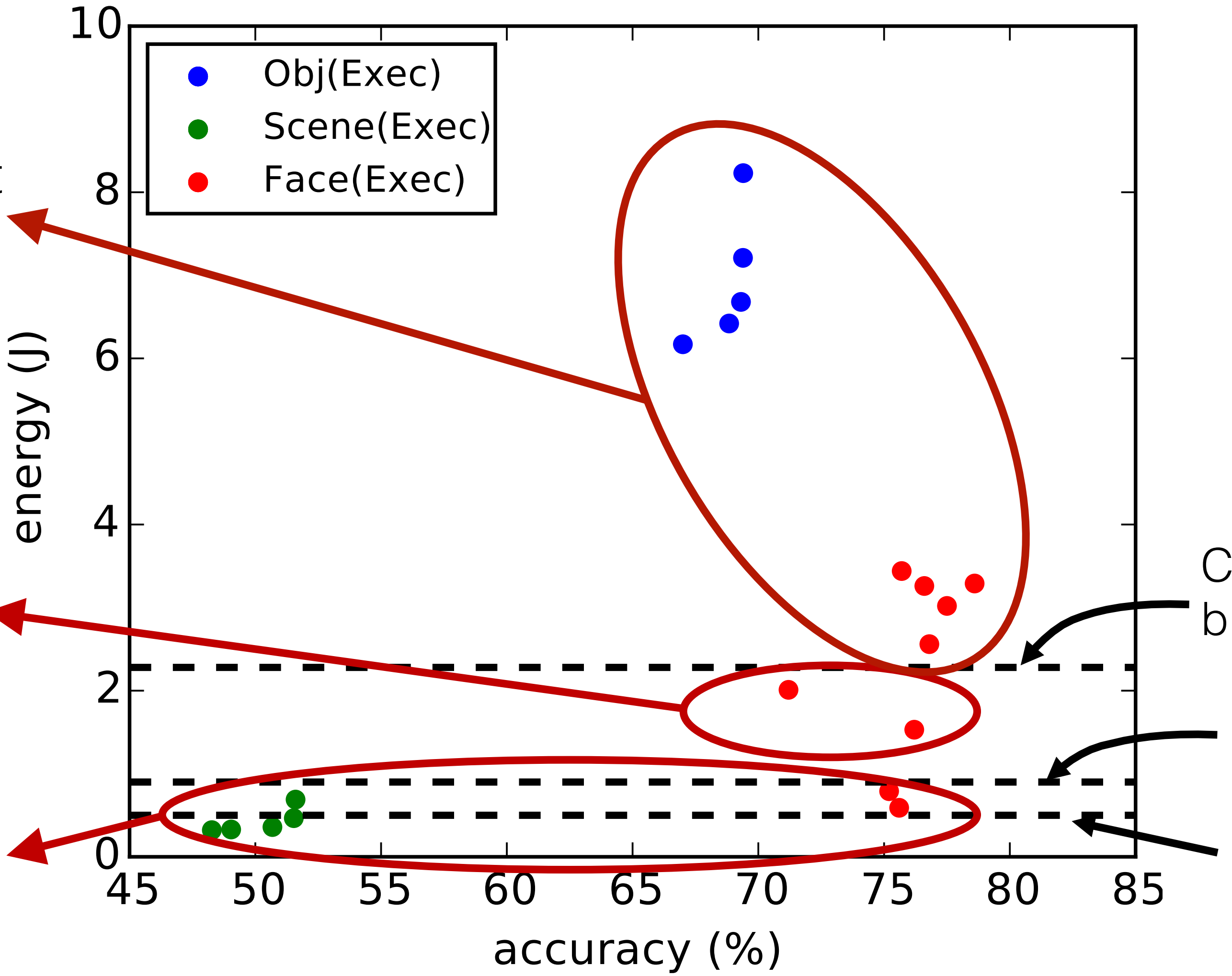


Memory / accuracy trade-off



Substantially reduce memory use with gradual accuracy loss

Energy / accuracy trade-off



Exceed energy budget when execute locally

Can execute locally under energy budget

Always execute locally

Compute energy budget (2.3J)

LTE xmit cost (0.9J)

Wifi xmit cost (0.5J)

Nvidia Jetson TK1

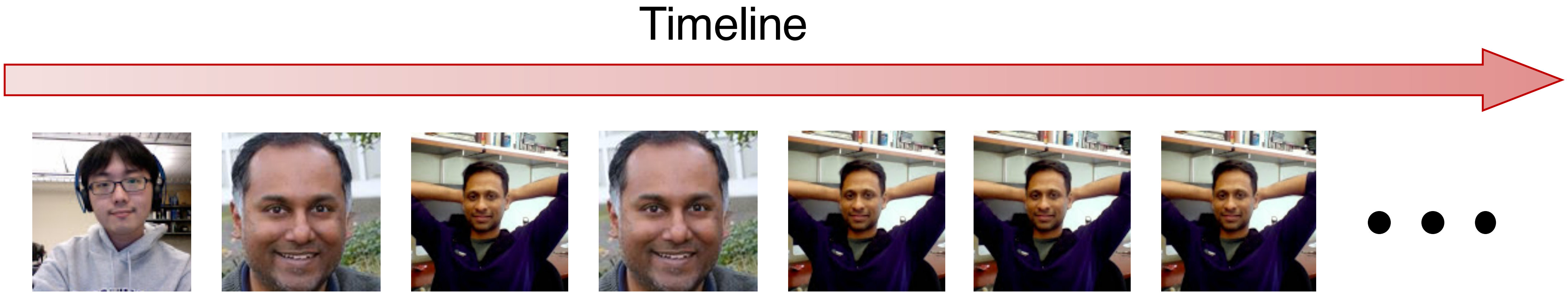
energy budget = total energy / total time(10h) / requests per second(1 req/sec)

Outline

- Detailed characterization of the approximation / resource trade-off for many optimizations
- Two new optimizations for streaming, multi-application settings
  - Specialization
  - Model sharing
- New scheduling problem, Approximate Model Scheduling, with a heuristic solution

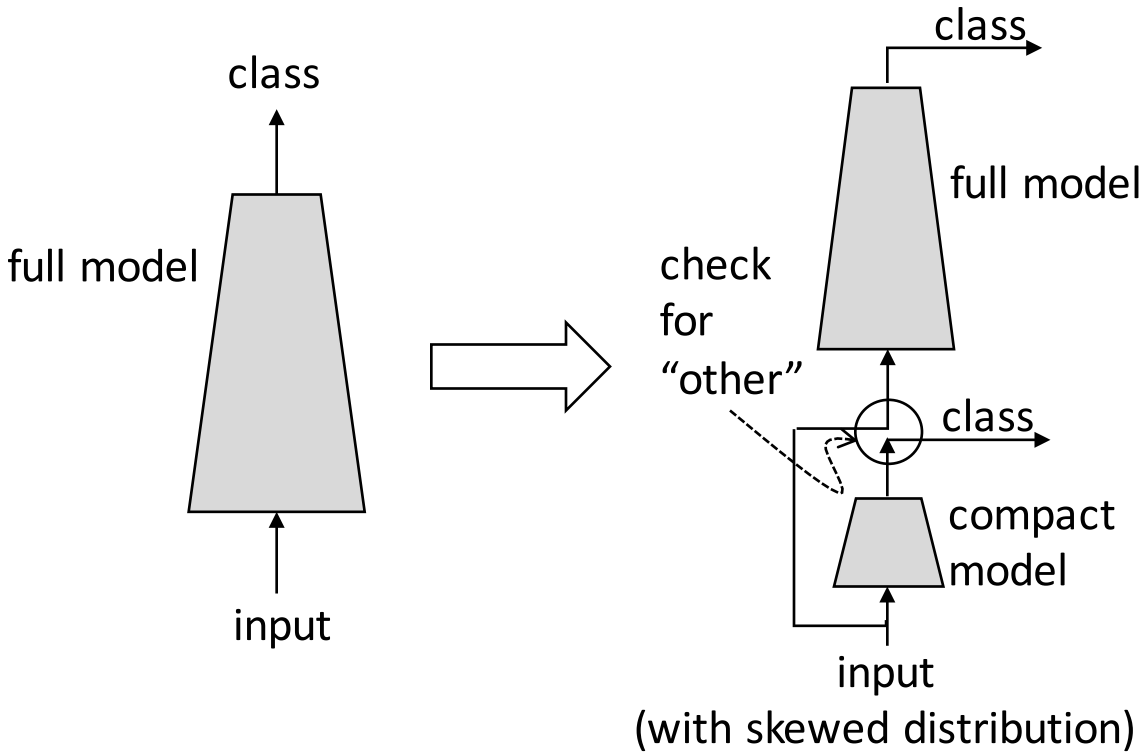
Exploiting stream locality by specialization

- Standard deep neural network recognizes 4000 people
- Most of videos are dominated by less than 10 faces over minutes

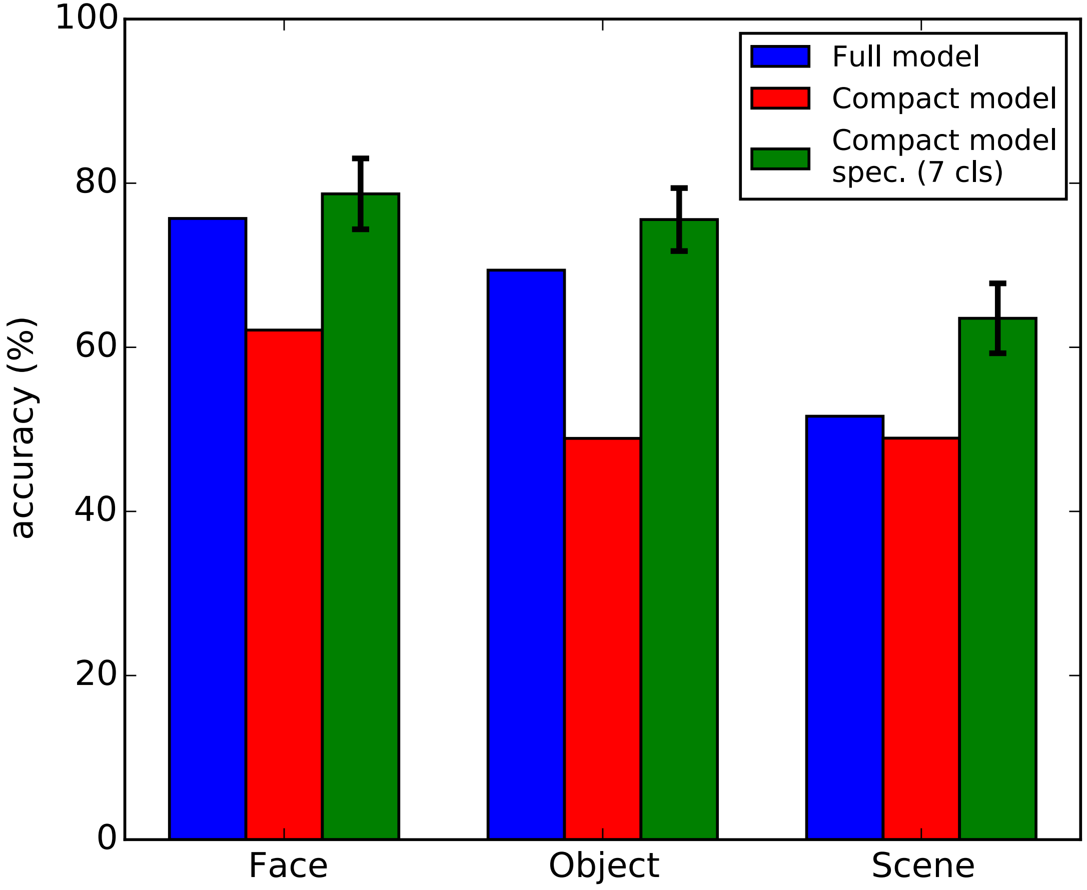


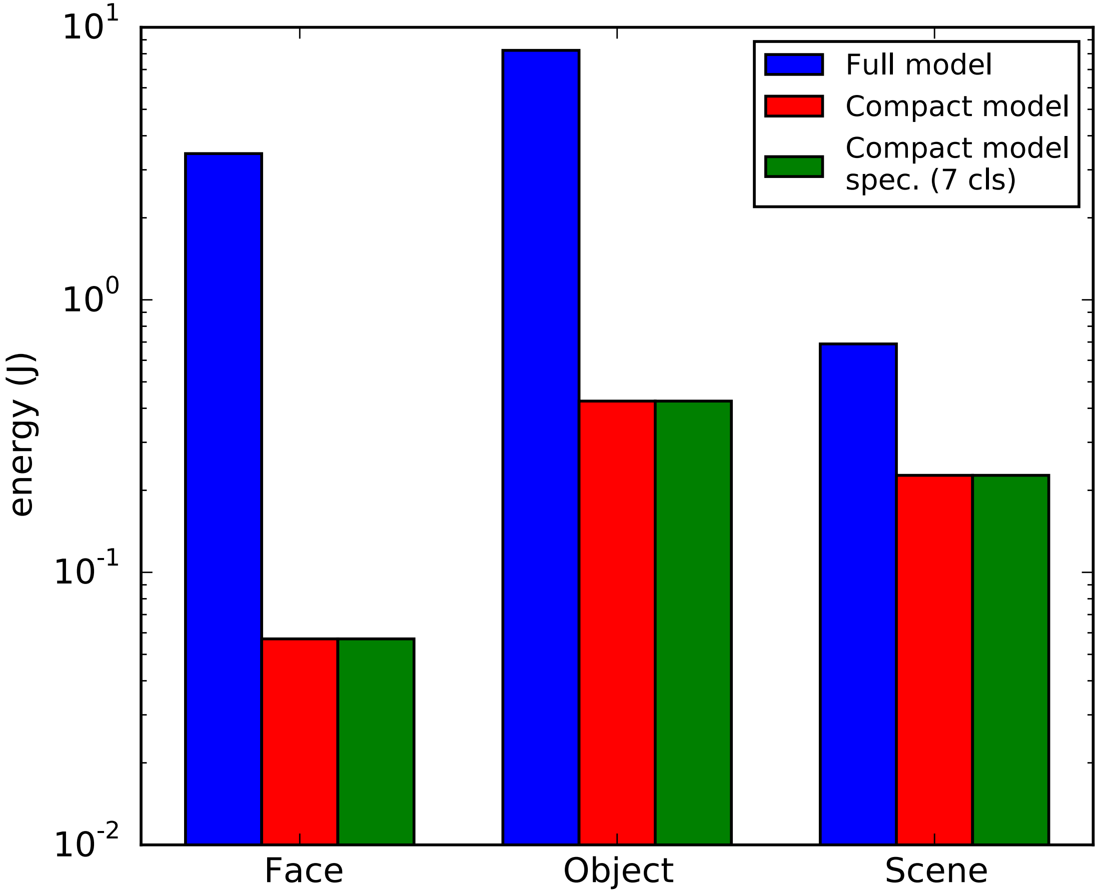
Produce more compact models for skewed classes

Specialization runtime



Better resource/accuracy trade-off





No specialization

Specialization

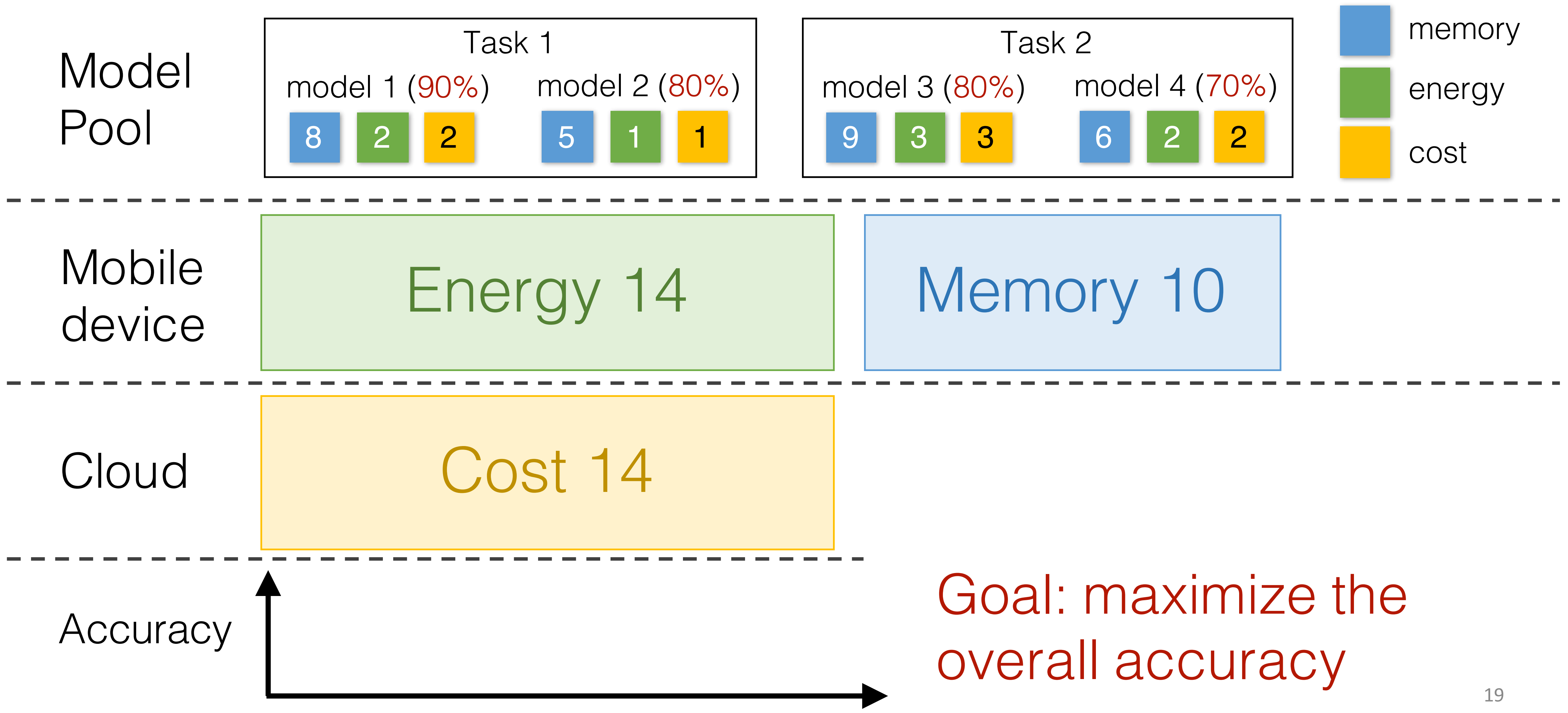
Outline

- Detailed characterization of the approximation / resource trade-off for many optimizations

- Two new optimizations for streaming, multi-application settings

- New scheduling problem, Approximate Model Scheduling, with a heuristic solution

Approximate model scheduling



Model Pool

Task 1

model 1 (90%) 8 2 2

model 2 (80%) 5 1 1

Task 2

model 3 (80%) 9 3 3

model 4 (70%) 6 2 2

🟦 memory

🟩 energy

🟨 cost

Mobile device

Energy 14

Memory 10

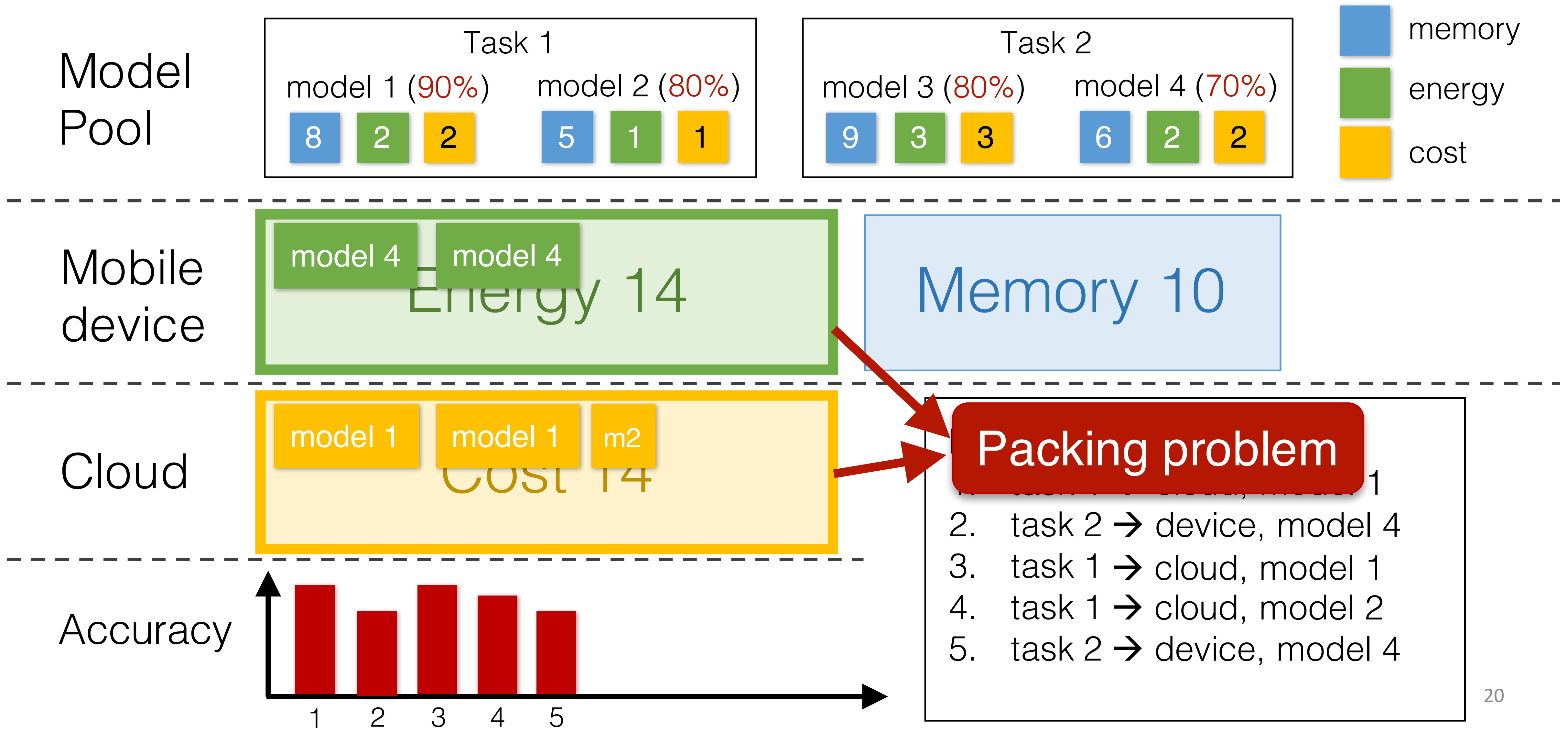
Cloud

Cost 14

Accuracy

Goal: maximize the overall accuracy

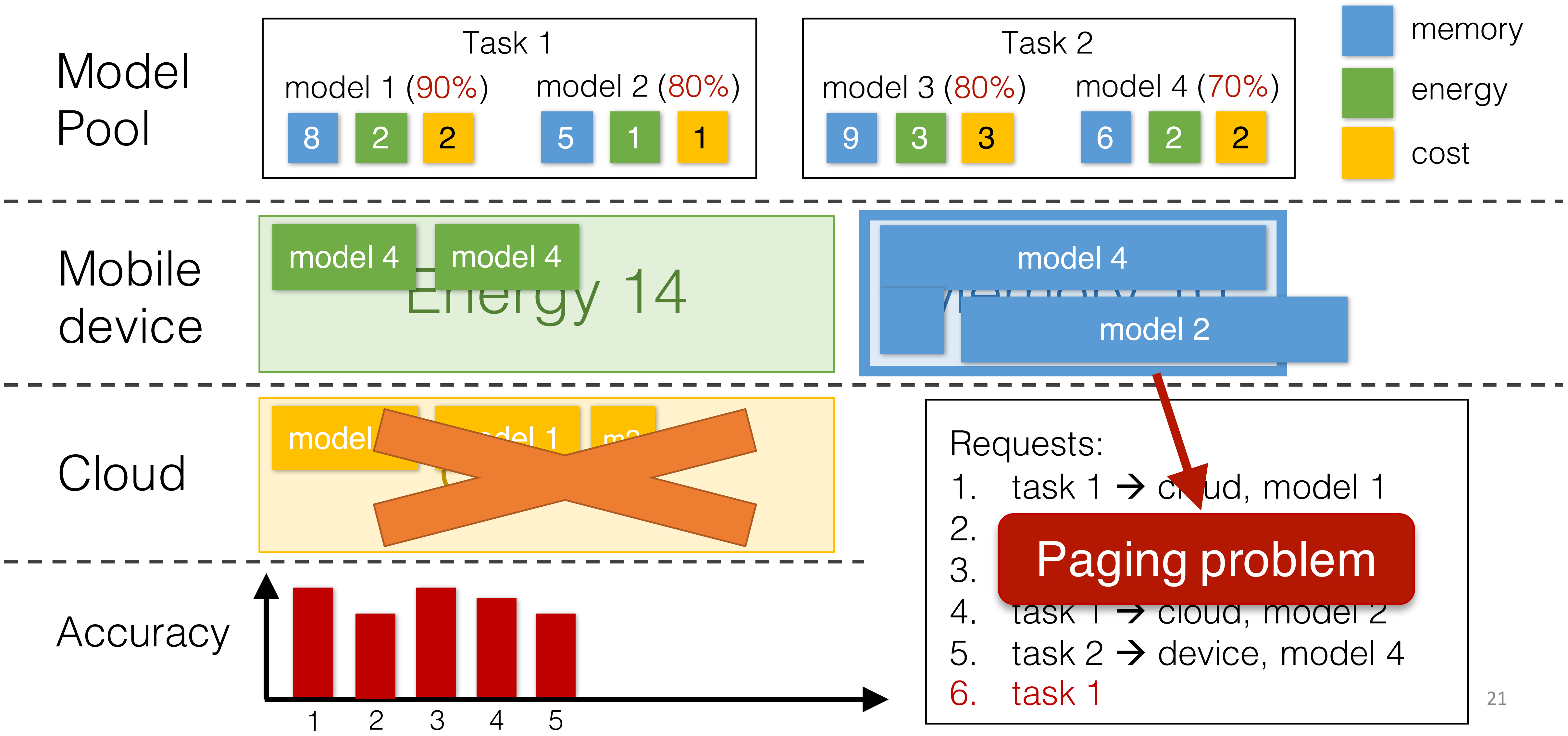
Approximate model scheduling



Packing problem

1. task 1 → cloud, model 1
2. task 2 → device, model 4
3. task 1 → cloud, model 1
4. task 1 → cloud, model 2
5. task 2 → device, model 4

Approximate model scheduling



Requests:

1. task 1 → cloud, model 1
2.
3.
4. task 1 → cloud, model 2
5. task 2 → device, model 4
6. task 1

Approximate model scheduling

- Packing problem: pick versions that satisfy energy/cost budgets

$$\sum_t e_i x_{it} \leq E, \sum_t c_i x'_{it} \leq C \;\; (x_{it}, x'_{it} \in [0,1], x_{it} \cdot x'_{it} = 0)$$

- Paging problem: pick versions that fit in memory

$$\forall 1 \leq t \leq T, \sum_{i=1}^{n} s_i x_{it} \leq S$$

- Goal: maximize the accuracy

$$\max_x \sum_t \sum_i a_i (x_{it} + x'_{it})$$

No known optimal online algorithms

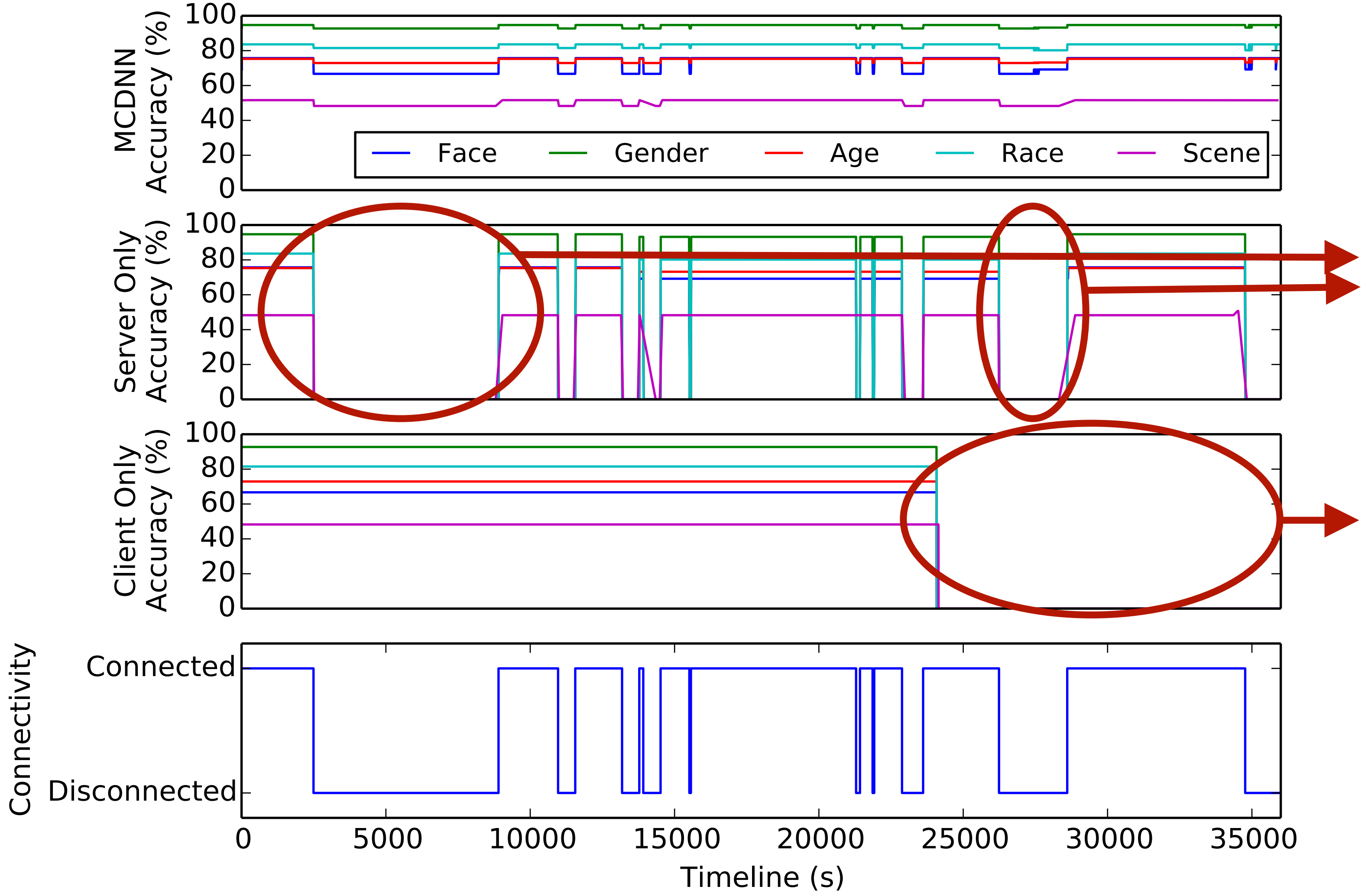
Heuristic scheduler

- Estimate future resource use and compute the budget for each request

- Account for paging cost to reduce oscillations

- Use increasingly more accurate versions of more heavily used models

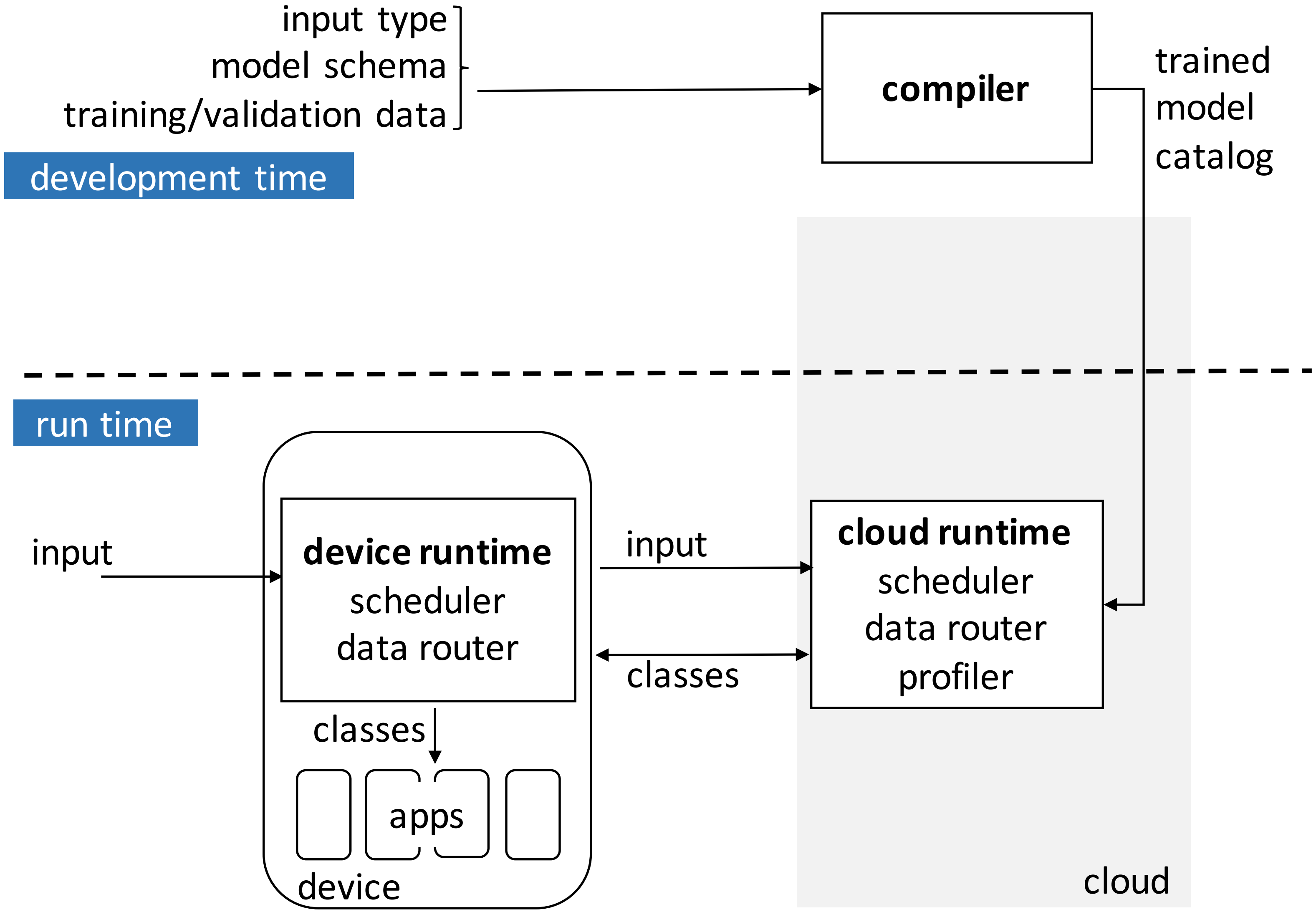
Trace-driven evaluation



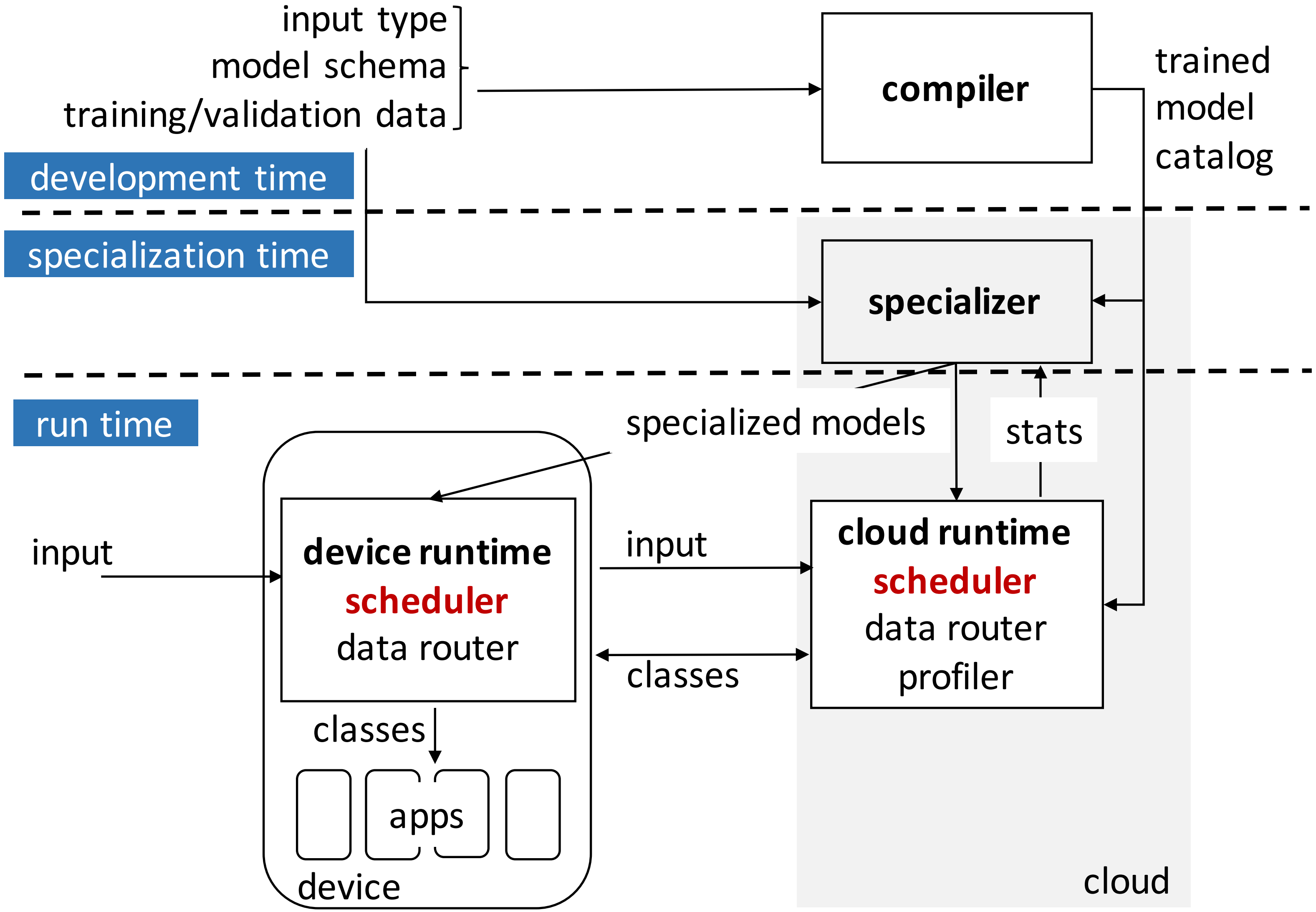
Lose connectivity

Run out of battery

MCDNN framework



MCDNN framework



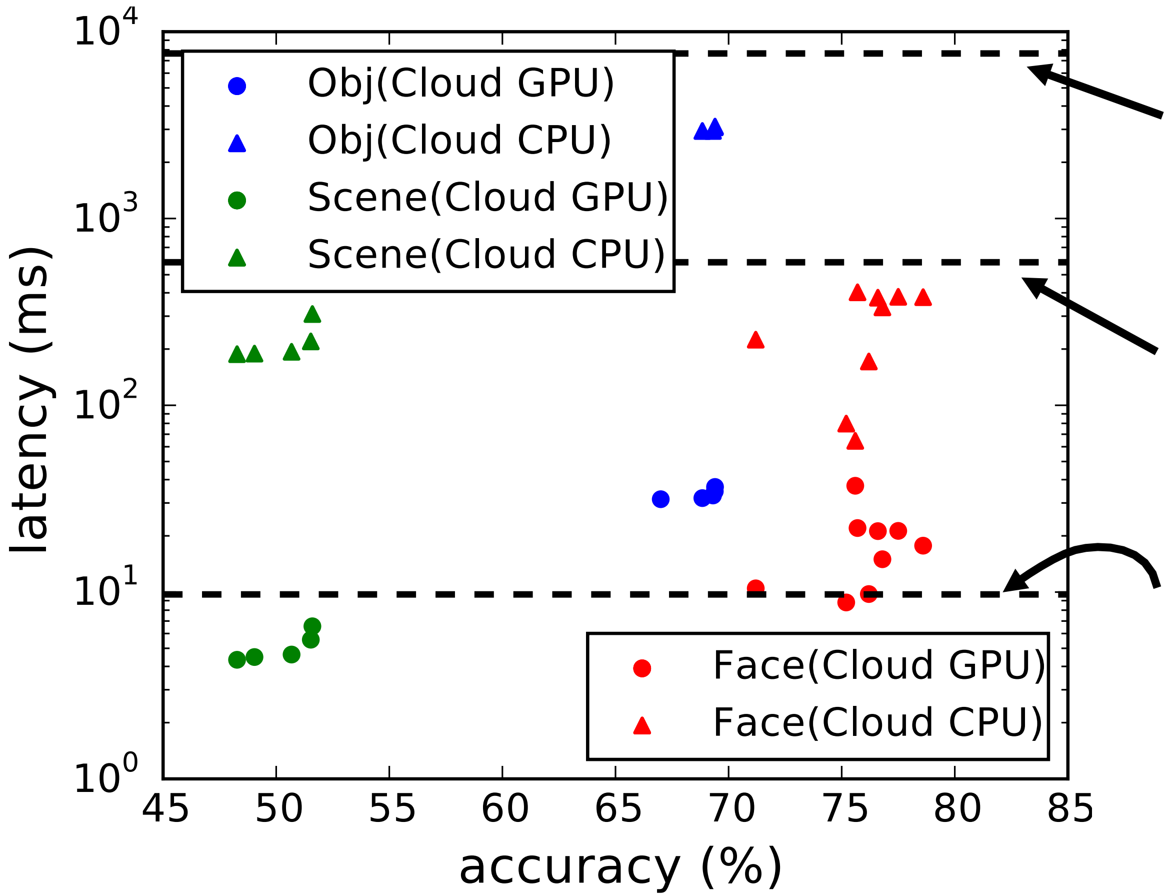
Conclusion

- MCDNN makes efficient trade-offs between resource use and accuracy
- Formulate the approximate model scheduling problem and devise a heuristic algorithm
- Design a generic approximation-based execution framework for continuous mobile vision

Thank you! Questions?

Backup Slides

Cloud cost / accuracy trade-off



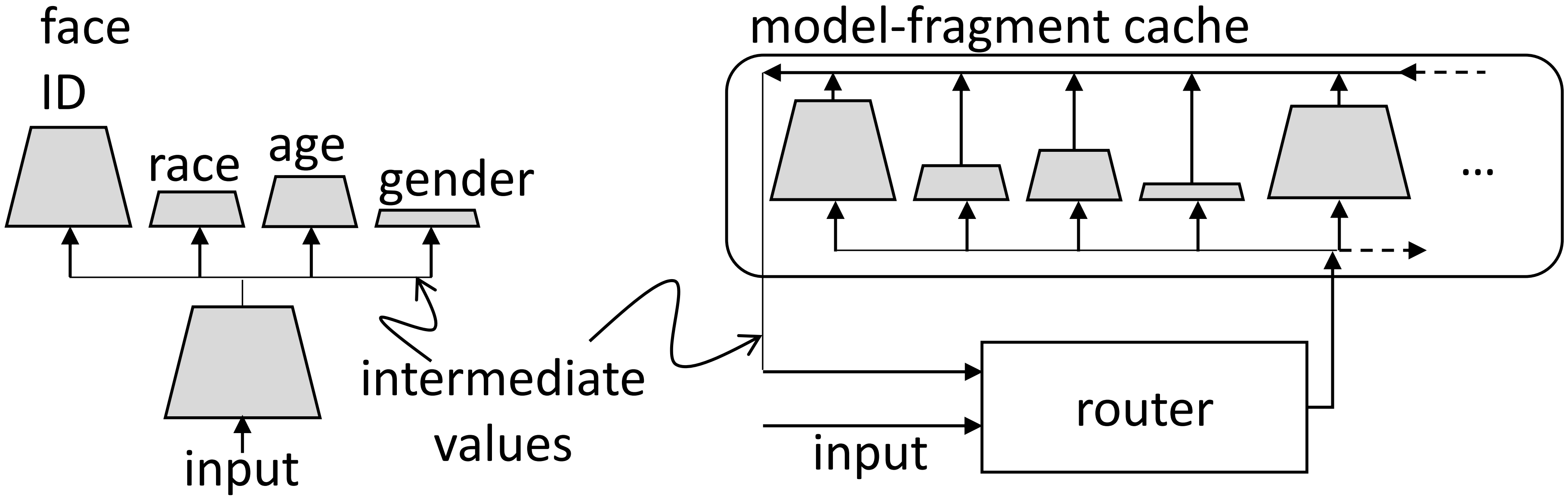
cloud CPU latency budget (7645ms) @ $10/yr, **1 request/min** @ AWS c4.large

cloud GPU latency budget (582ms) @ $10/yr, **1 request/min** @ AWS g2.2xlarge

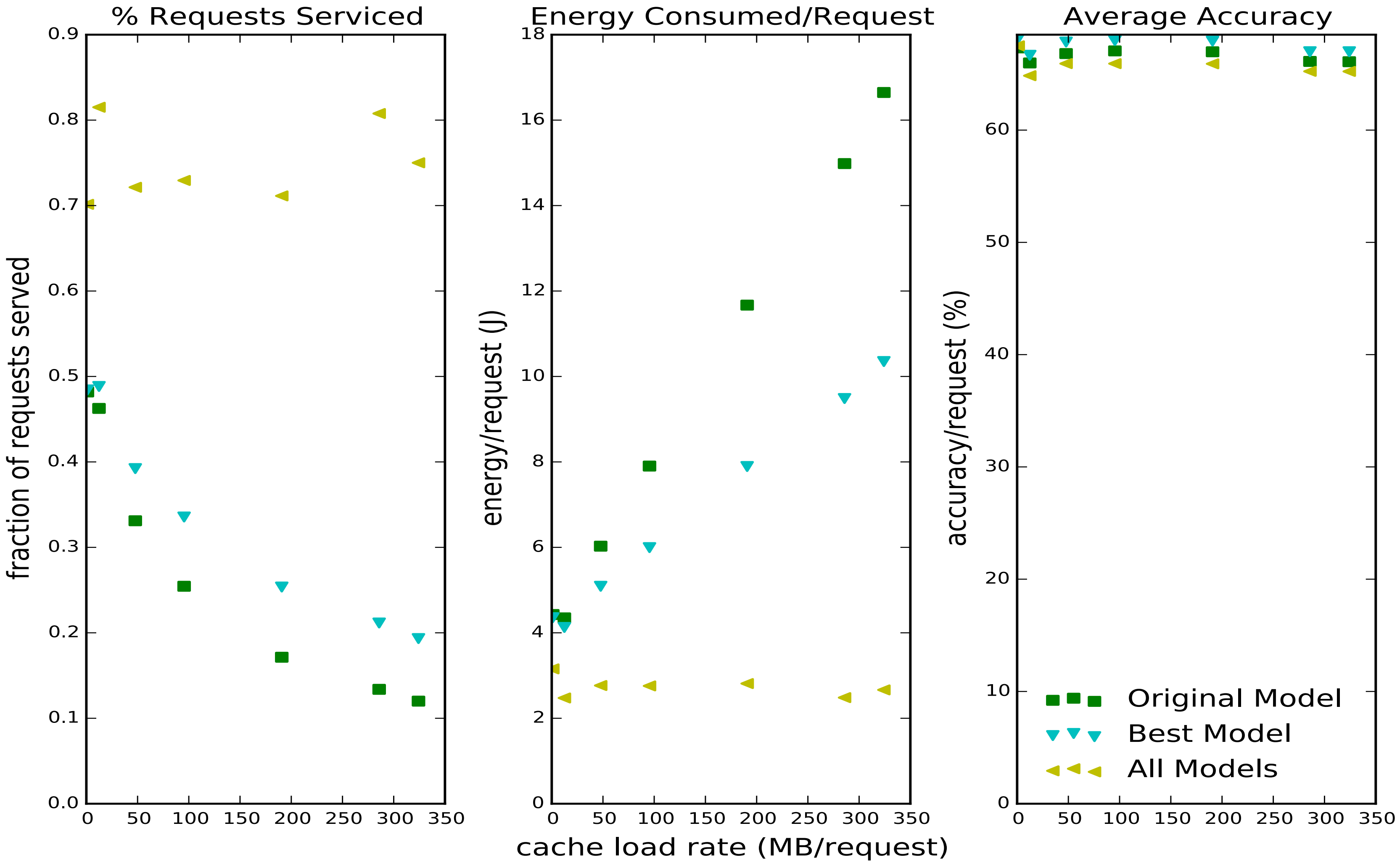
cloud GPU latency budget (9.7ms) @ $10/yr, **1 request/s** @ AWS g2.2xlarge

latency budget = cost budget / cost per hour / #requests

Model sharing



Dynamically-sized caching scheme


% Requests Serviced
fraction of requests served
Energy Consumed/Request
energy/request (J)
Average Accuracy
accuracy/request (%)
cache load rate (MB/request)
Original Model
Best Model
All Models