Mainstream: Adaptive compute sharing for video analysis

Angela Jiang, Christopher Canel, Daniel Wong, Ishan Misra, Michael Kaminsky (Intel Labs), David Andersen, Greg Ganger

Overview

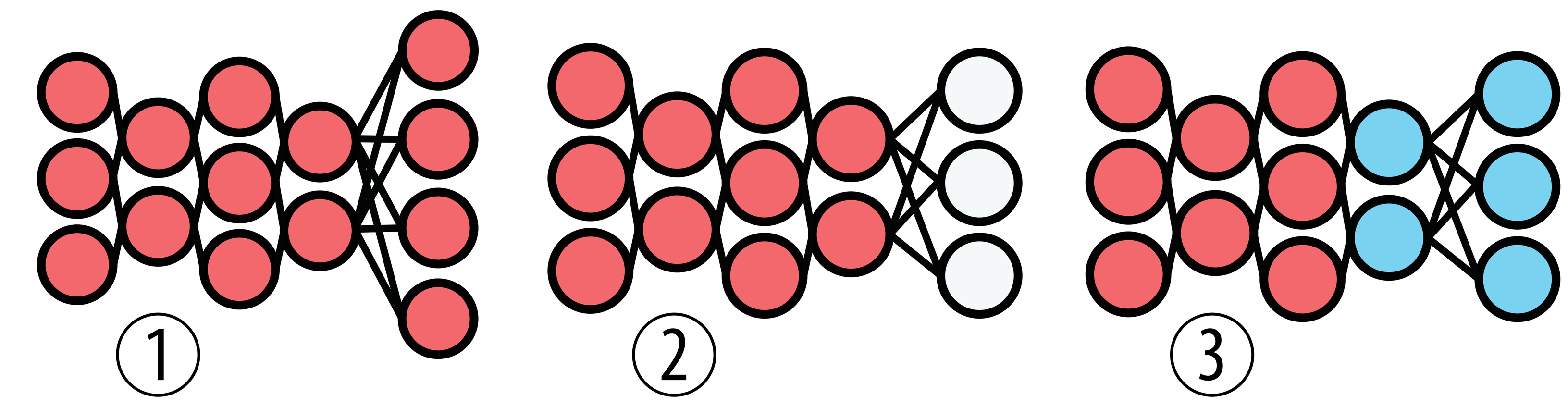
Goal:
- Efficiently run concurrent streaming video analysis apps

Problem:
- Most video analysis apps perform DNN inference
  - Running several full DNNs becomes very slow

Mainstream:
- Identifies and shares redundant DNN computation
  - By exploiting nature of fine-tuned DNNs
- Decides at runtime how much to share
  - Balances specialization vs. sharing trade-off
  - Optimizes when hardware and set of apps is known

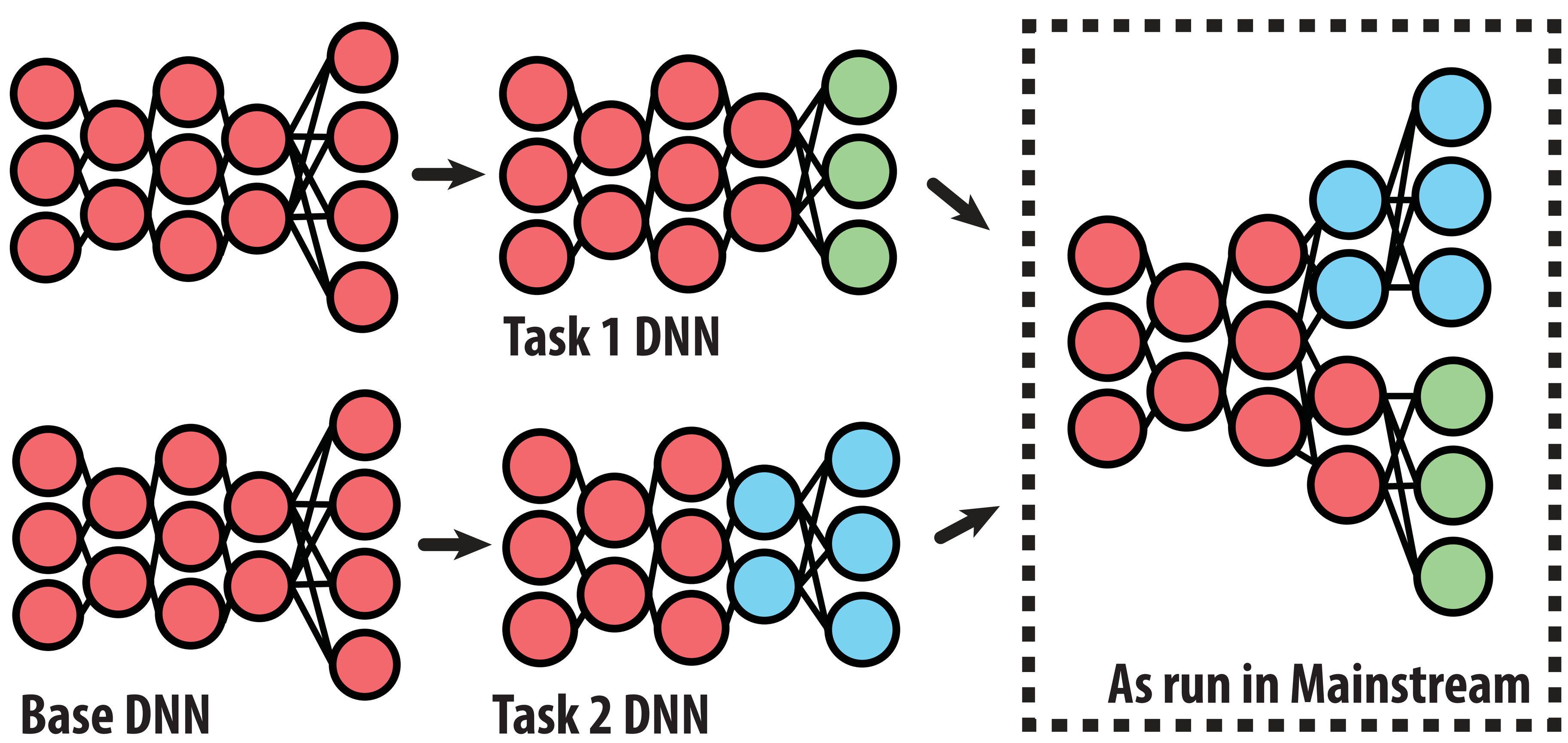
Transfer Learning

- When training task B, use DNN pre-trained for task A
  - Common practice for training networks

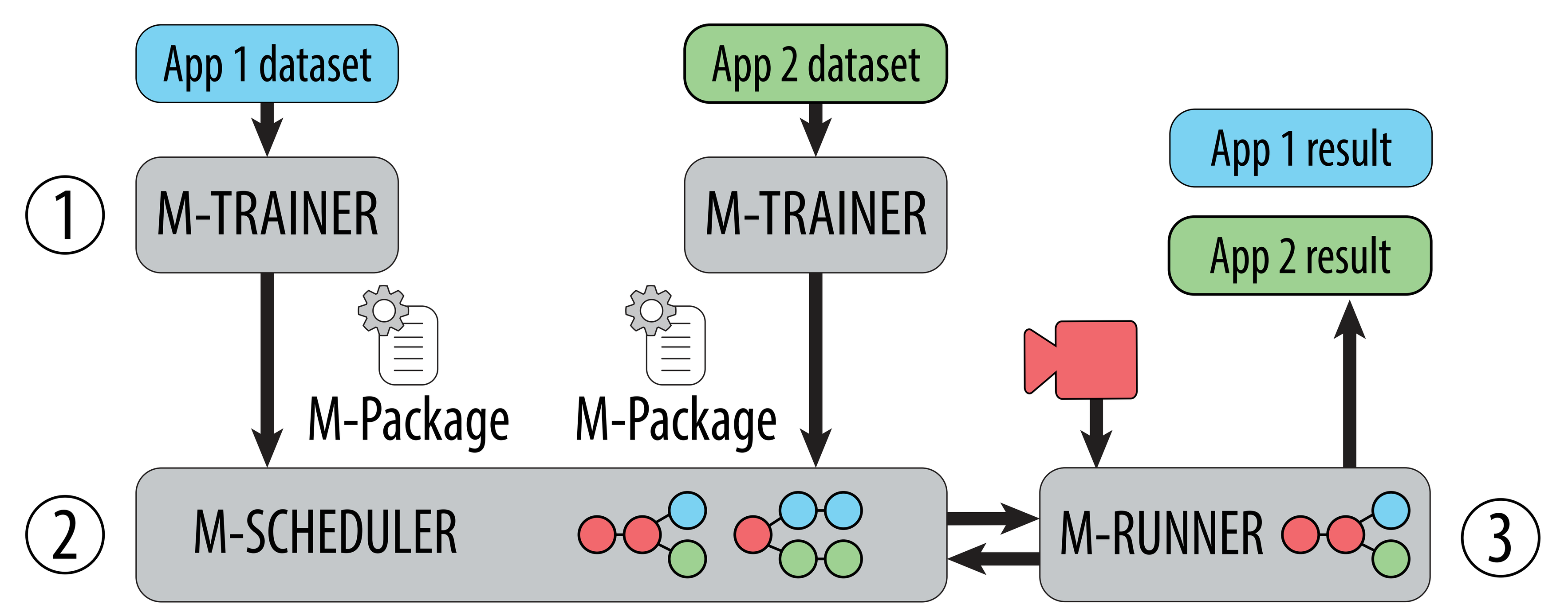


1. Network is trained from scratch for task A (e.g., ImageNet)
2. Replace A-specific final layer with B-specific final layer
3. Fine-tune part of network for task B, other layers held frozen

Sharing Computation



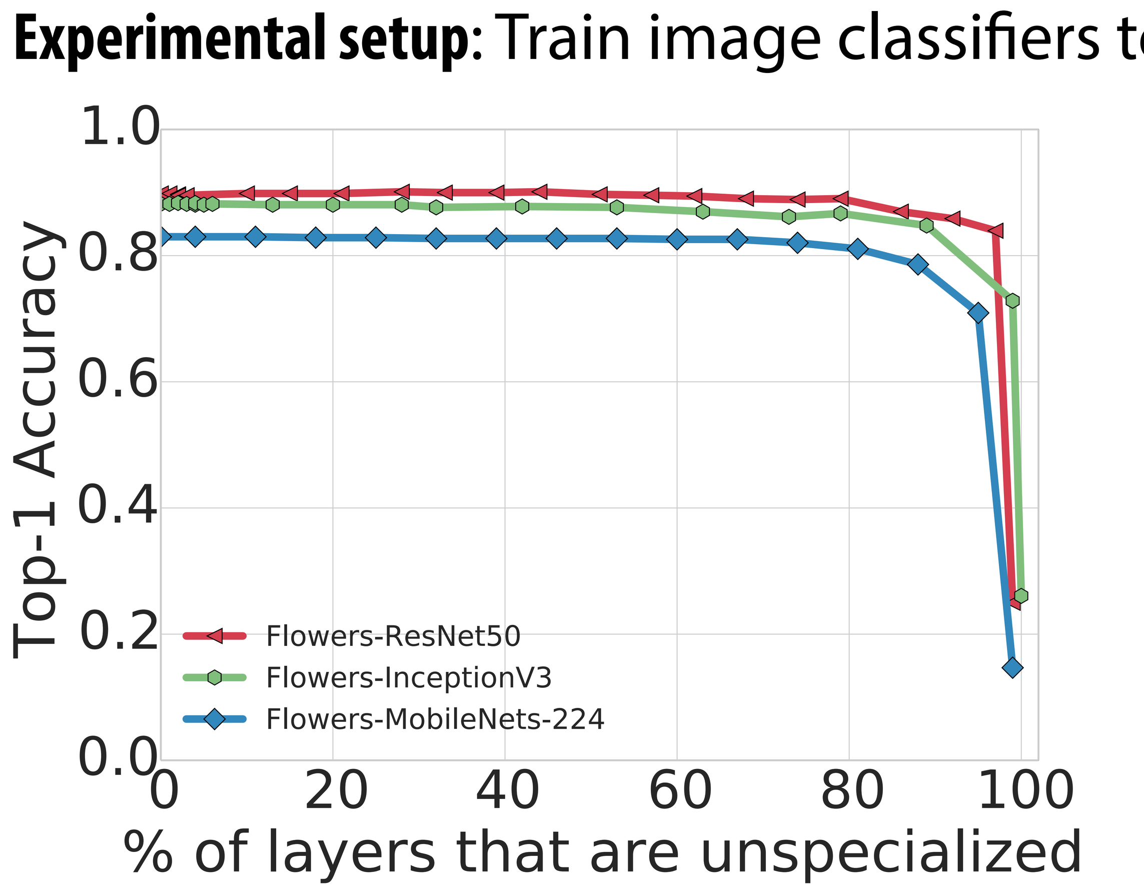
Mainstream Architecture



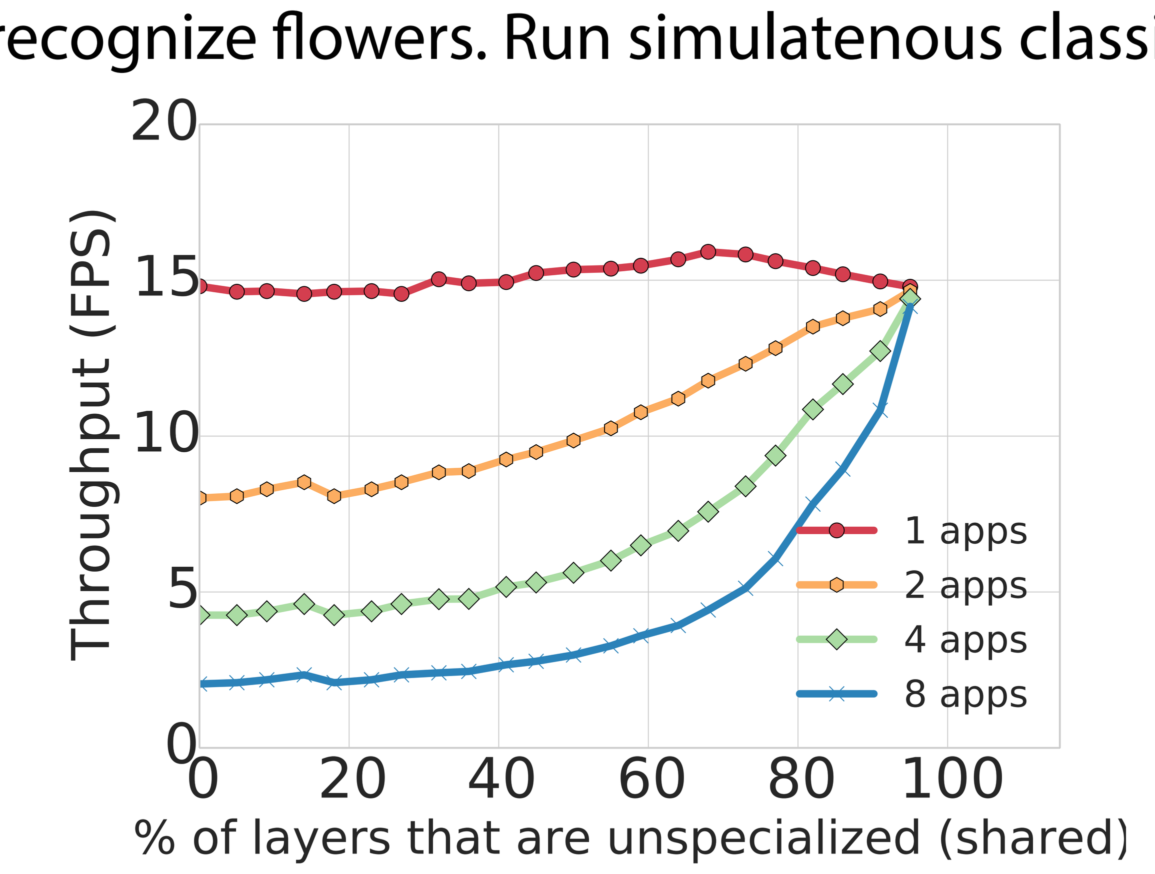
1. **M-trainer** trains DNNs with varying % of network held frozen
2. **M-Scheduler** determines amount of DNN to share for each app
3. **M-Runner** processes video stream using deployed DNNs

Specialization vs. Sharing Trade-off

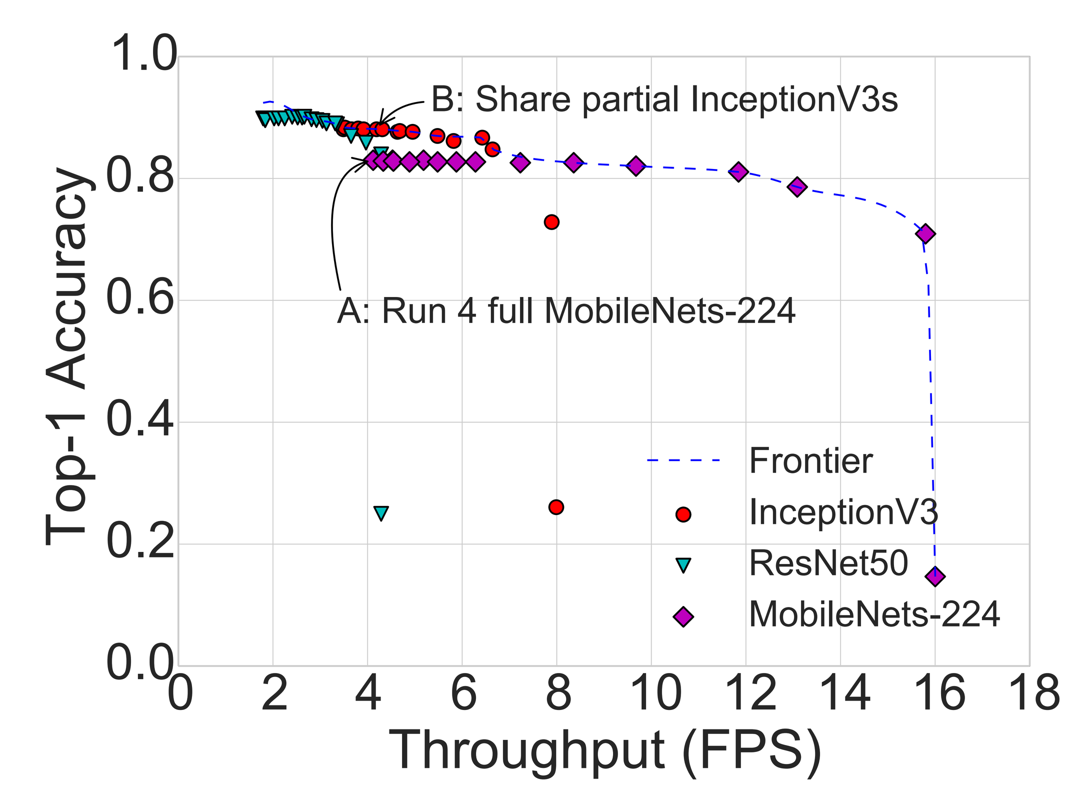
Experimental setup: Train image classifiers to recognize flowers. Run simulatenous classification pipelines on an Intel NUC.



Less specialization -> Lower per-frame acc.



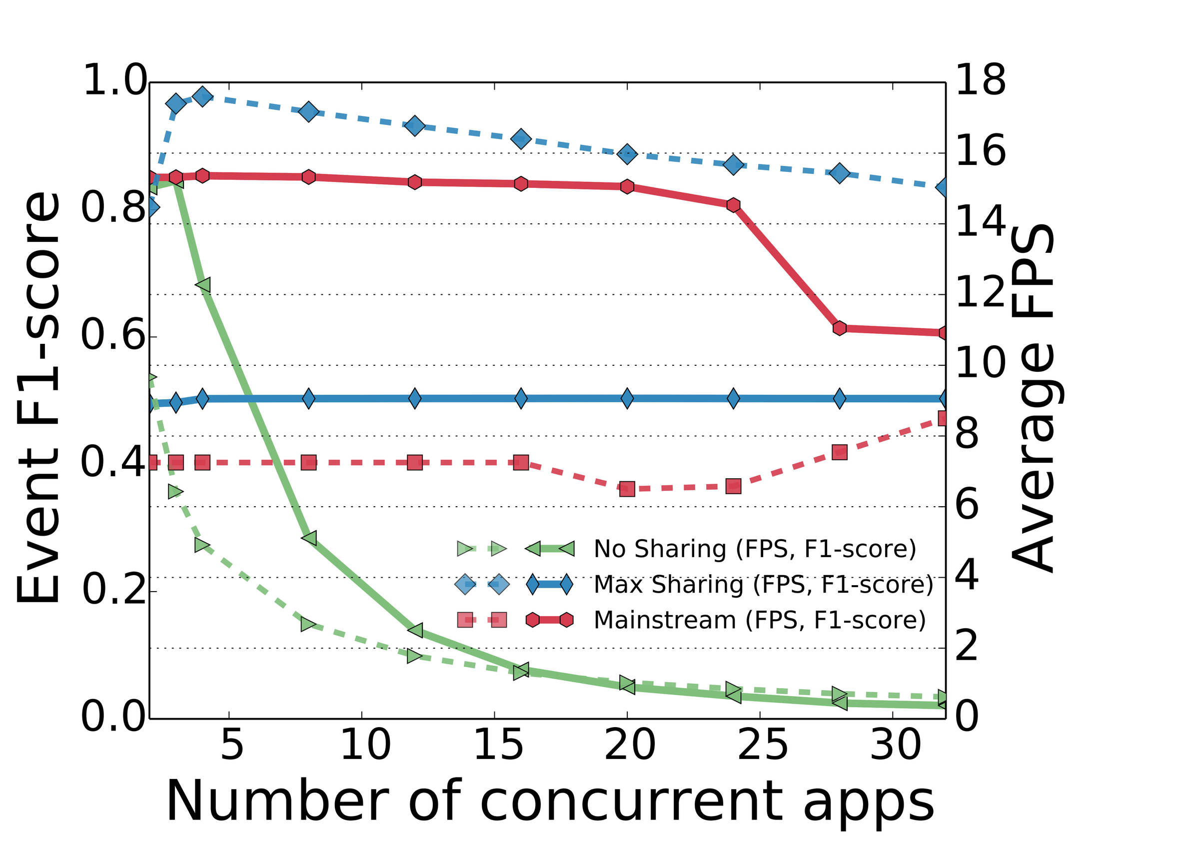
Less specialization -> Higher throughput

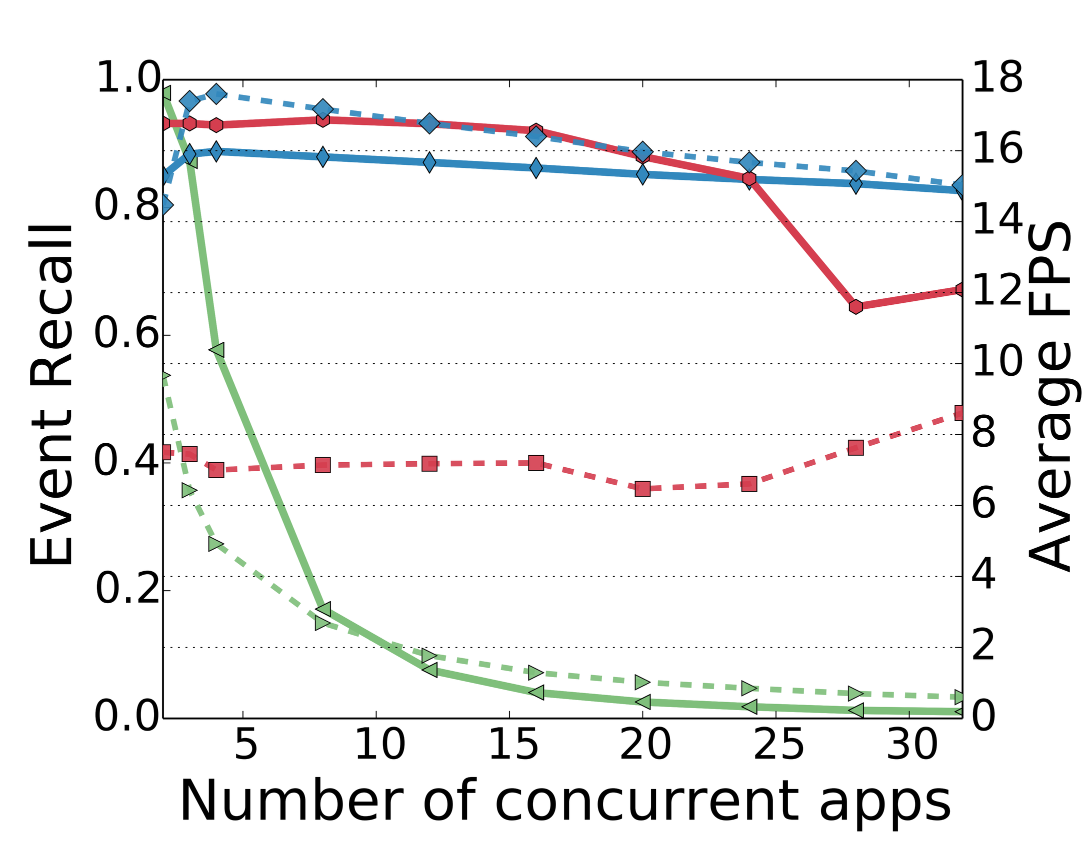


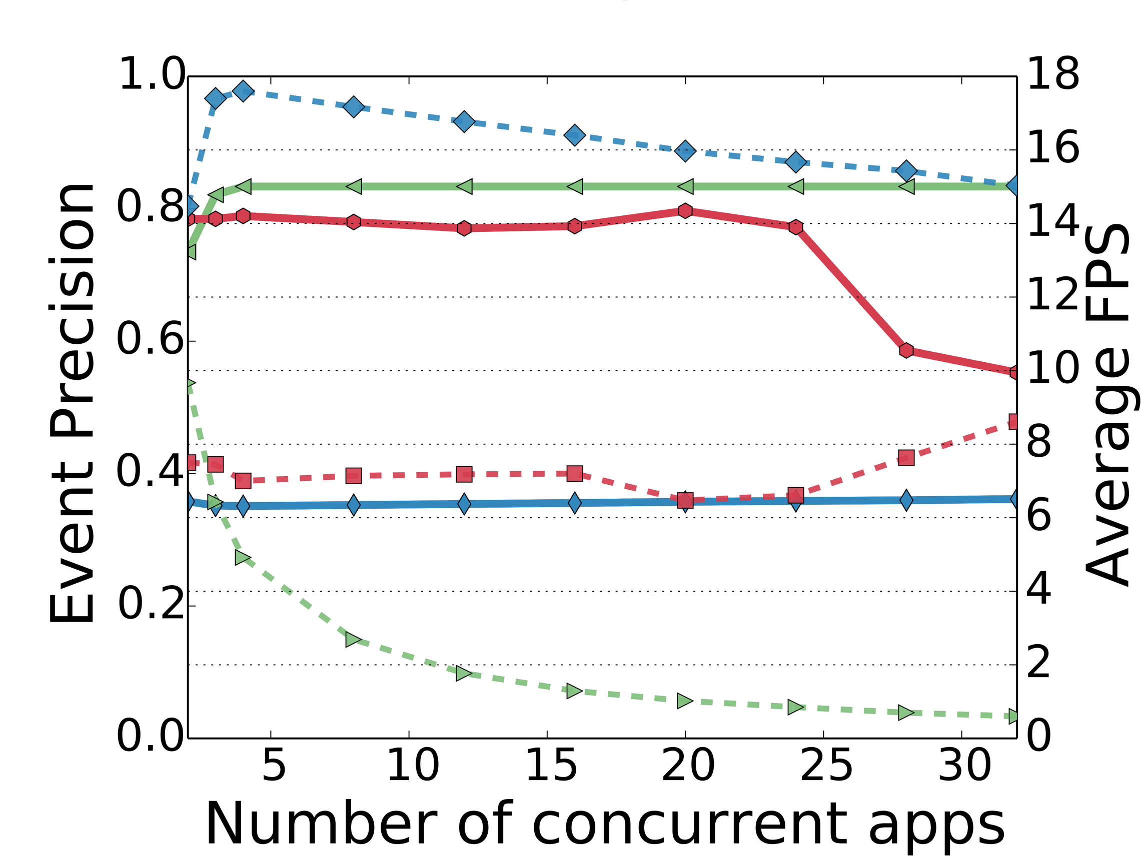
Optimal pts in trade-off space use sharing

Application Performance

Recall: % of events detected; **Precision:** % of detected events that are correct; **F1 score:** Harmonic mean of precision and







- "No Sharing" deploys full DNN for each app
- " Max Sharing"shares all but final layer
- Mainstream gives up to 28X higher F1
- "No Sharing" (NS) has low FPS, high acc.
- "Max Sharing" has high FPS, low acc.
-