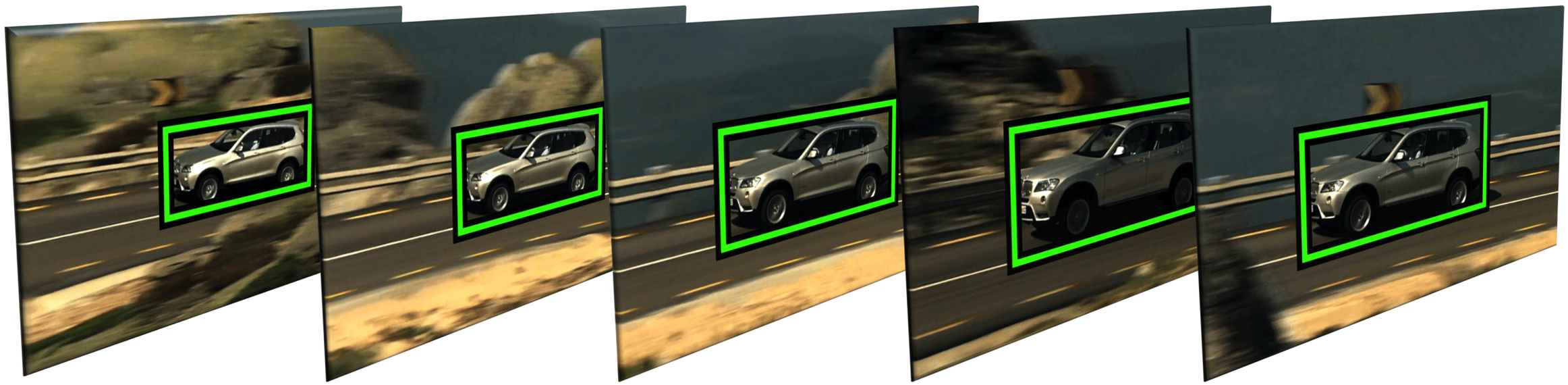
UNIVERSITY of ROCHESTER

Task-Level Modeling

Case study: Continuous Vision Applications

Presenter: Yiming Gan

Application Domain



Continuous Vision

Bottlenecks

- Long Latency
  - 100 ms per frame
  - A. Censi, “Low-latency event-based visual odometry ”, ICRA 2014.
- High Energy Consumptions
  - Drain battery in one hour using camera on iPhone 6s
  - Apple technique report, 2016
- High Memory Bandwidth Requirements
  - 1M Pixels image sensor running at 30 FPS needs16 GB/s
  - Raj Parihar, “Frame Buffer Design For Image Sensor Array”.

Continuous Vision Pipeline

Continuous Vision Pipeline

Continuous Vision Pipeline

Continuous Vision Pipeline

Mobile SoCs for Continuous Vision Pipeline

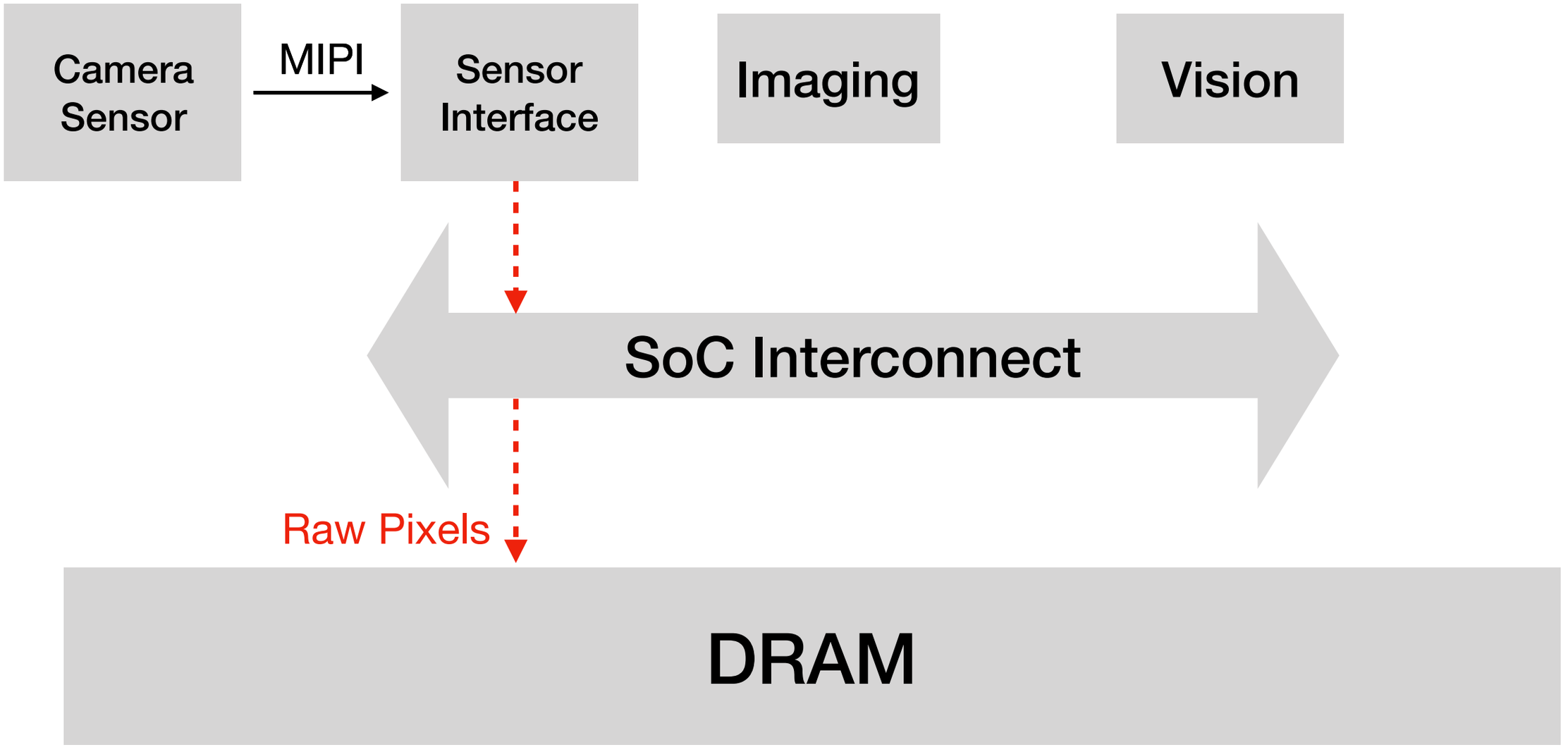
Sensing

Imaging

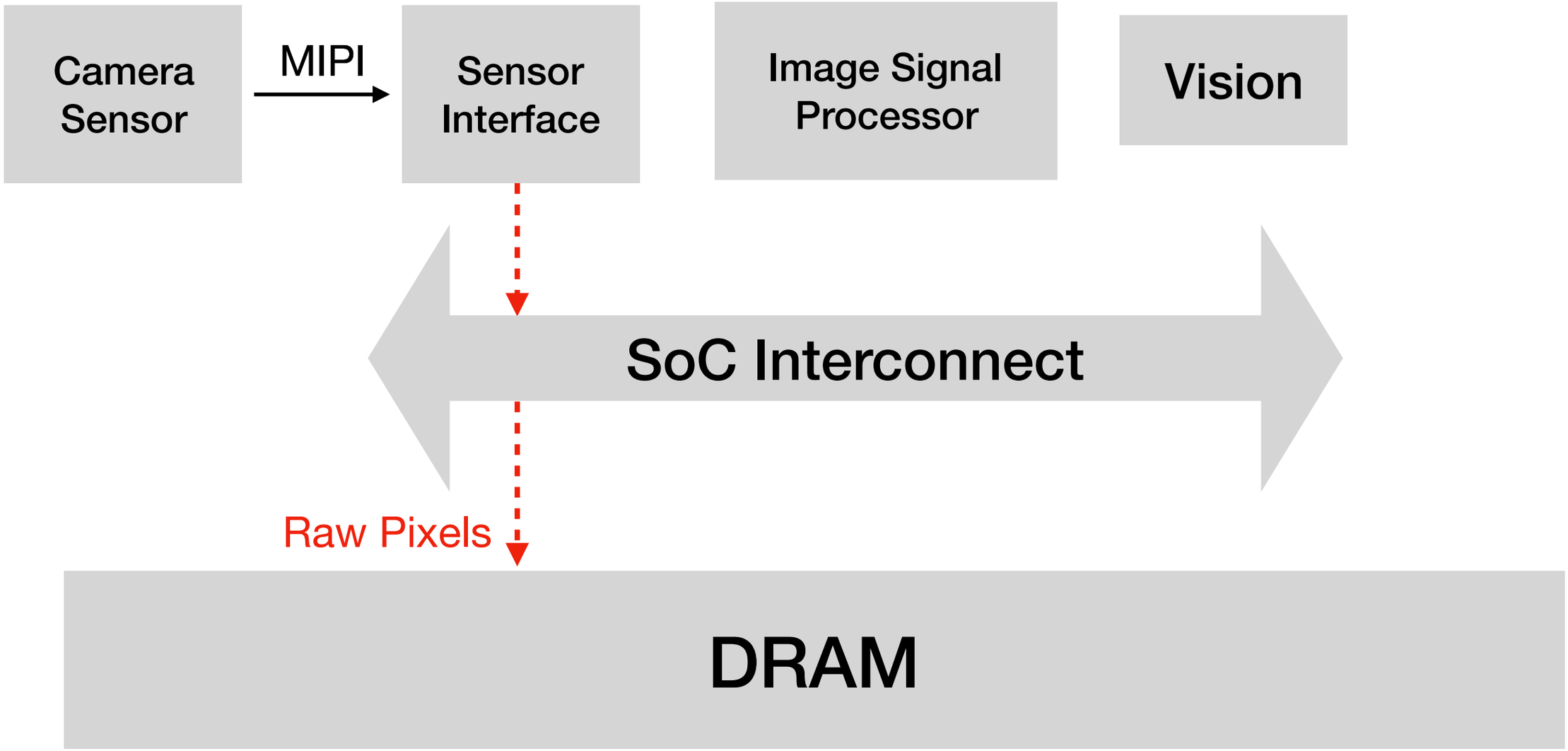
Vision

Mobile SoCs for Continuous Vision Pipeline

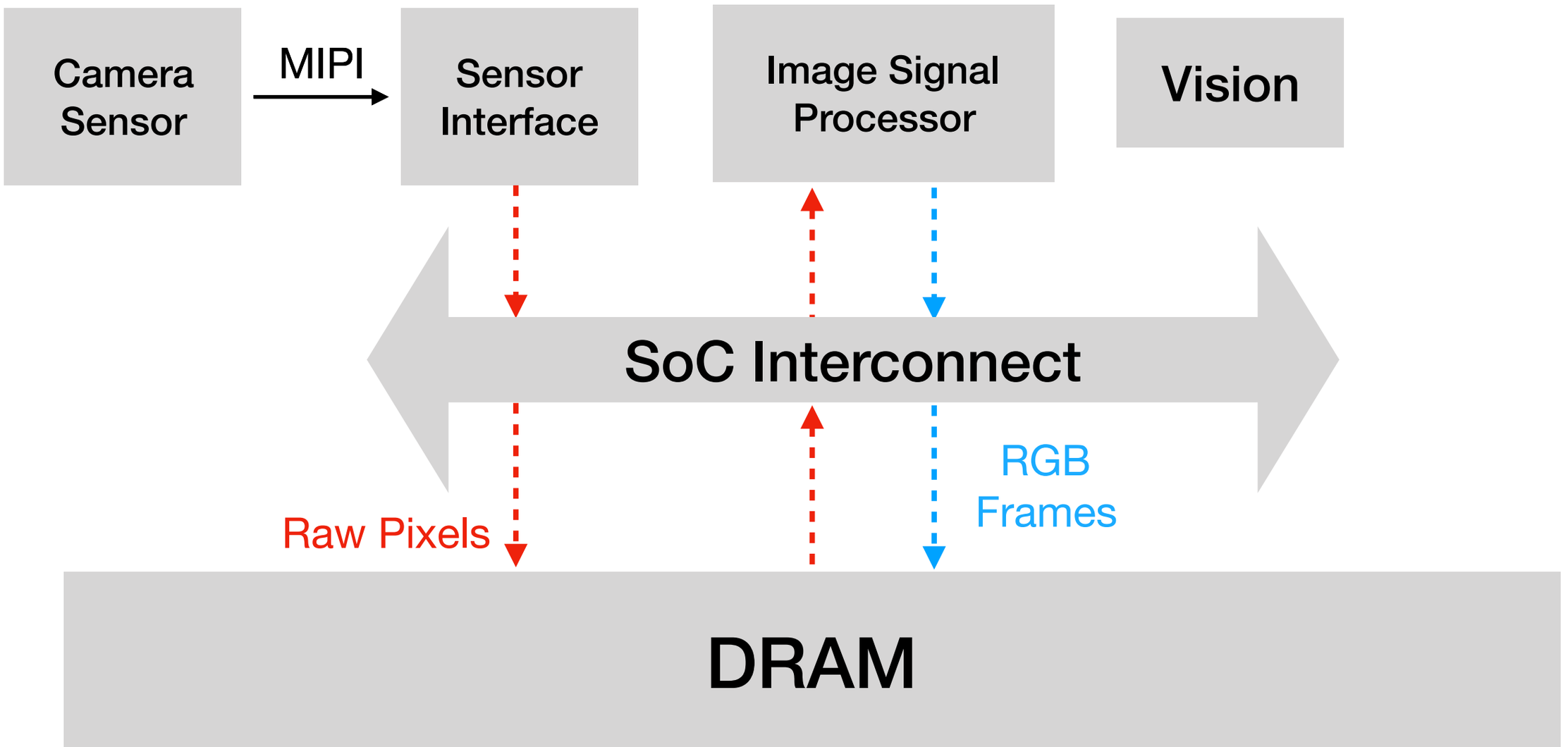
Mobile SoCs for Continuous Vision Pipeline



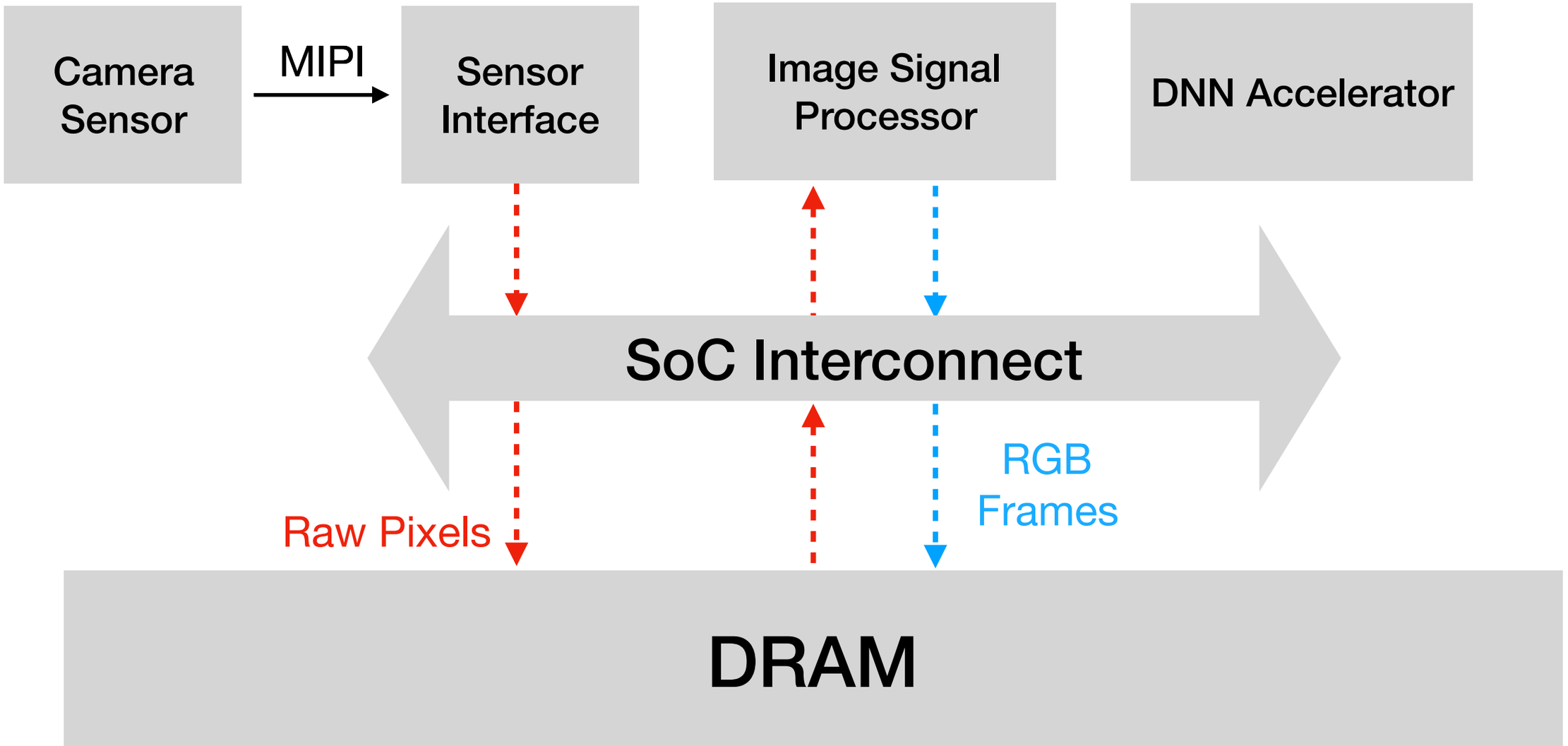
Mobile SoCs for Continuous Vision Pipeline



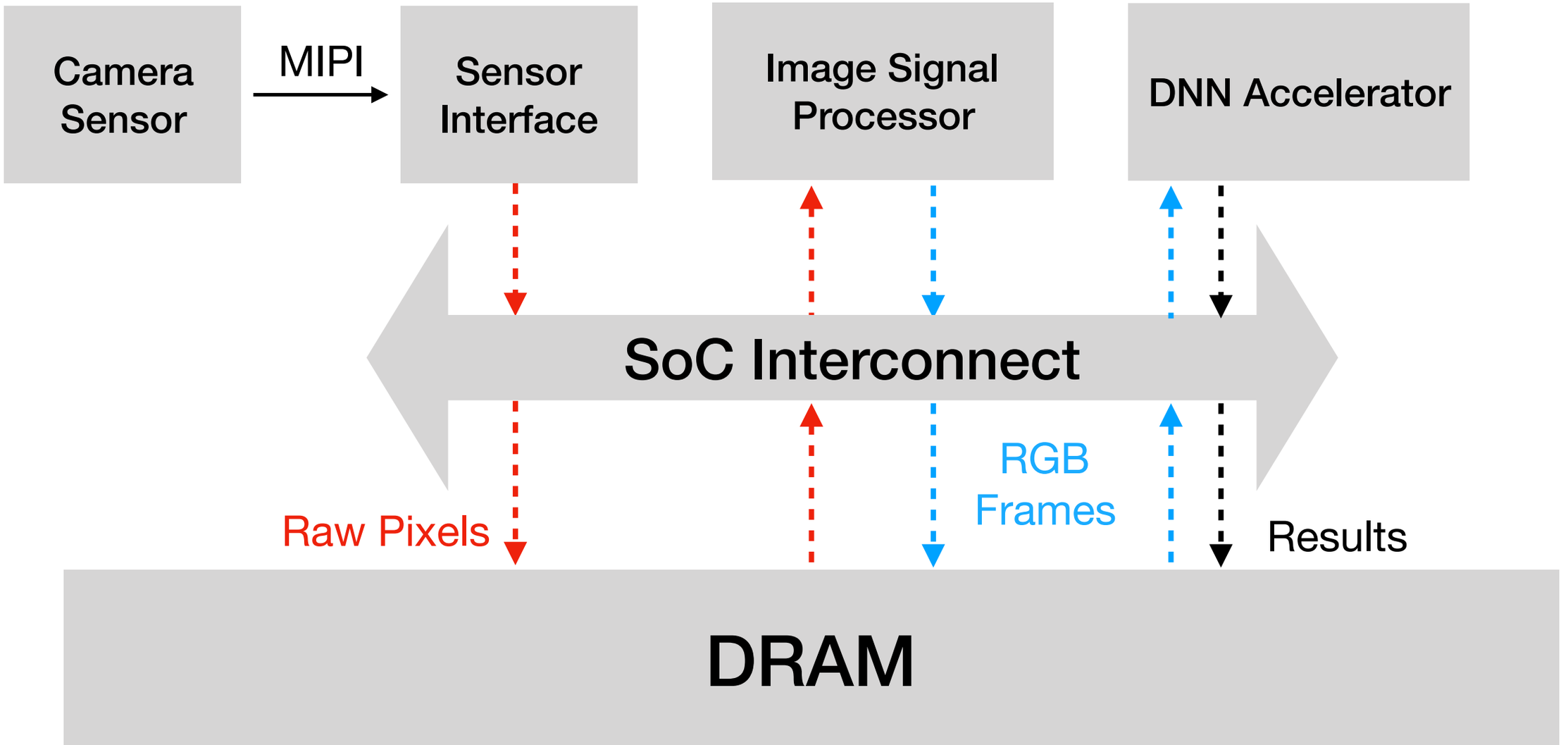
Mobile SoCs for Continuous Vision Pipeline



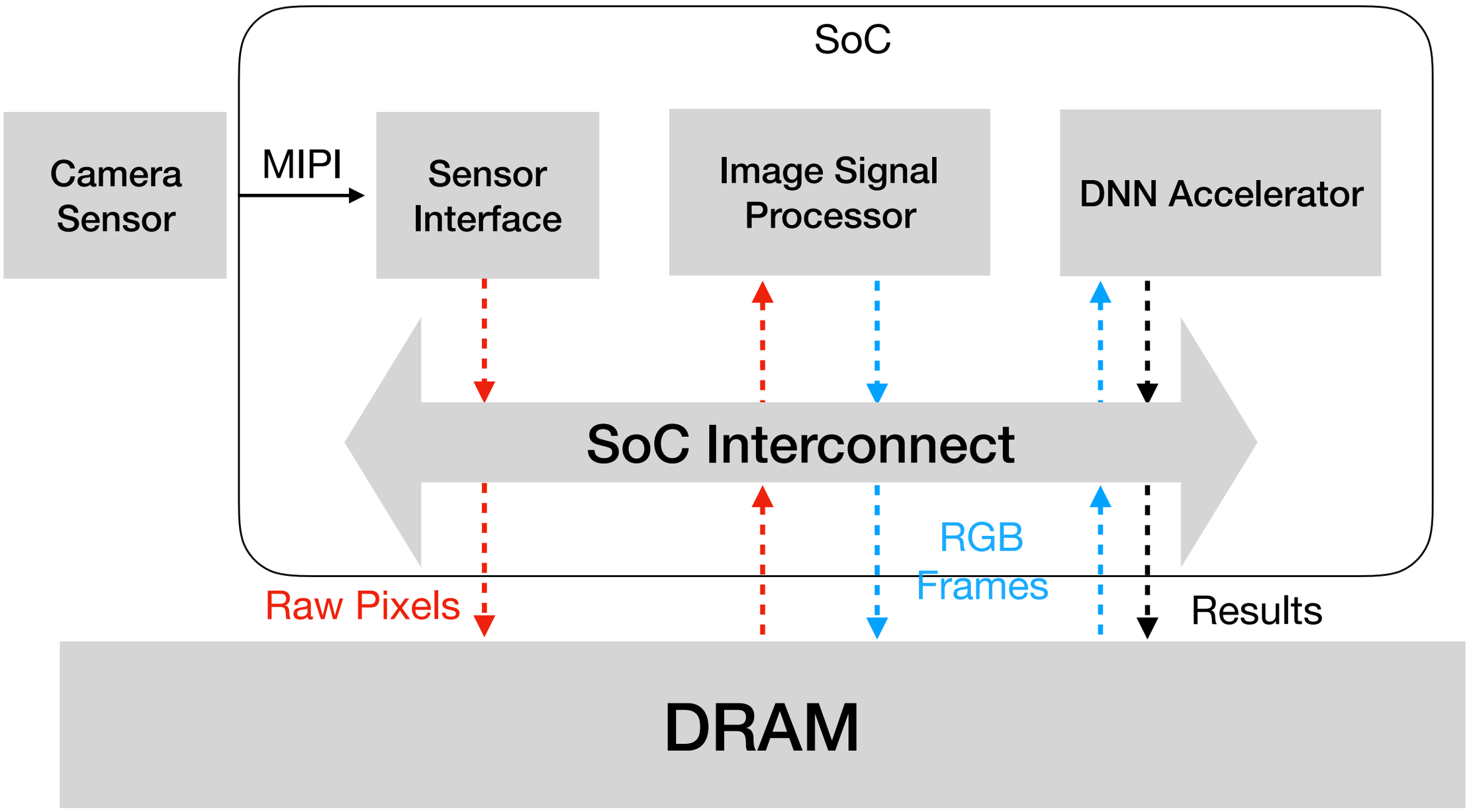
Mobile SoCs for Continuous Vision Pipeline



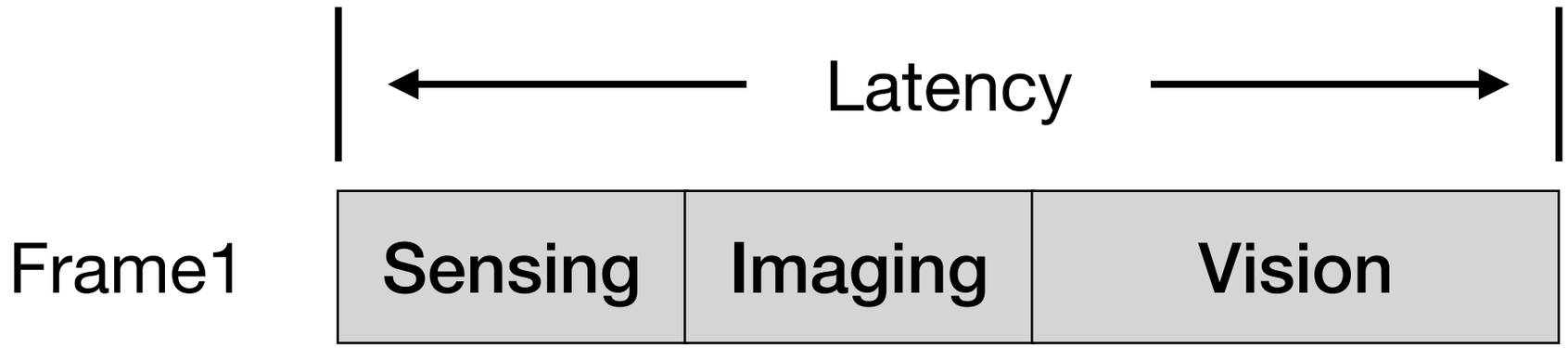
Mobile SoCs for Continuous Vision Pipeline



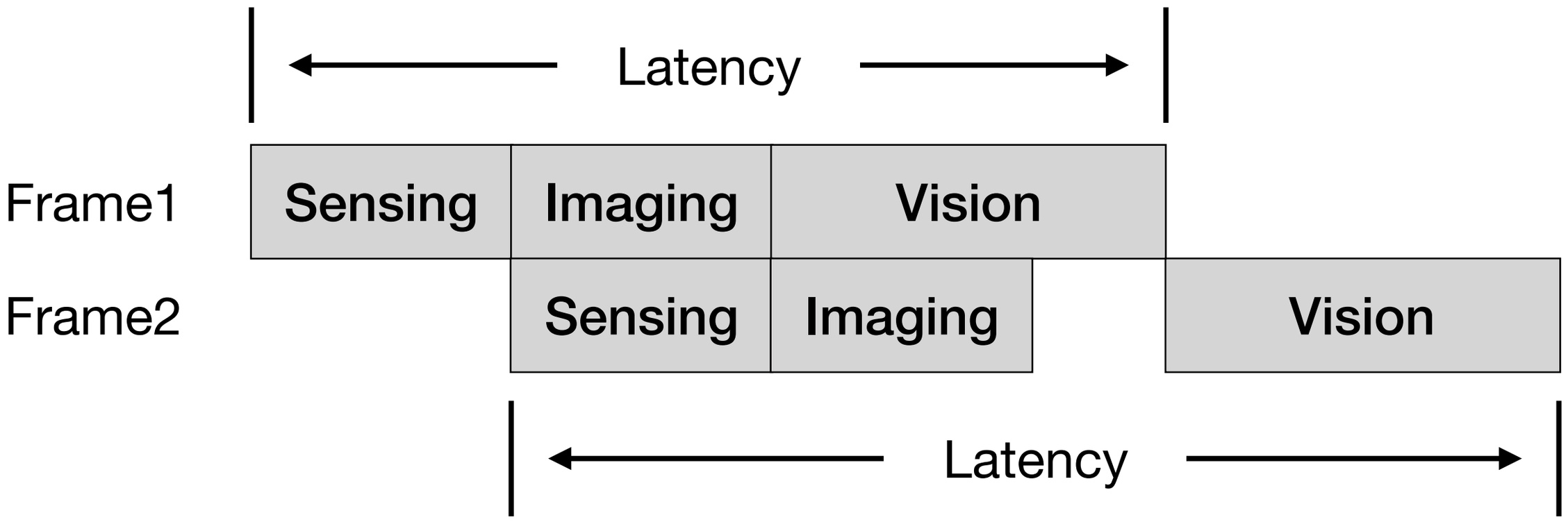
Mobile SoCs for Continuous Vision Pipeline



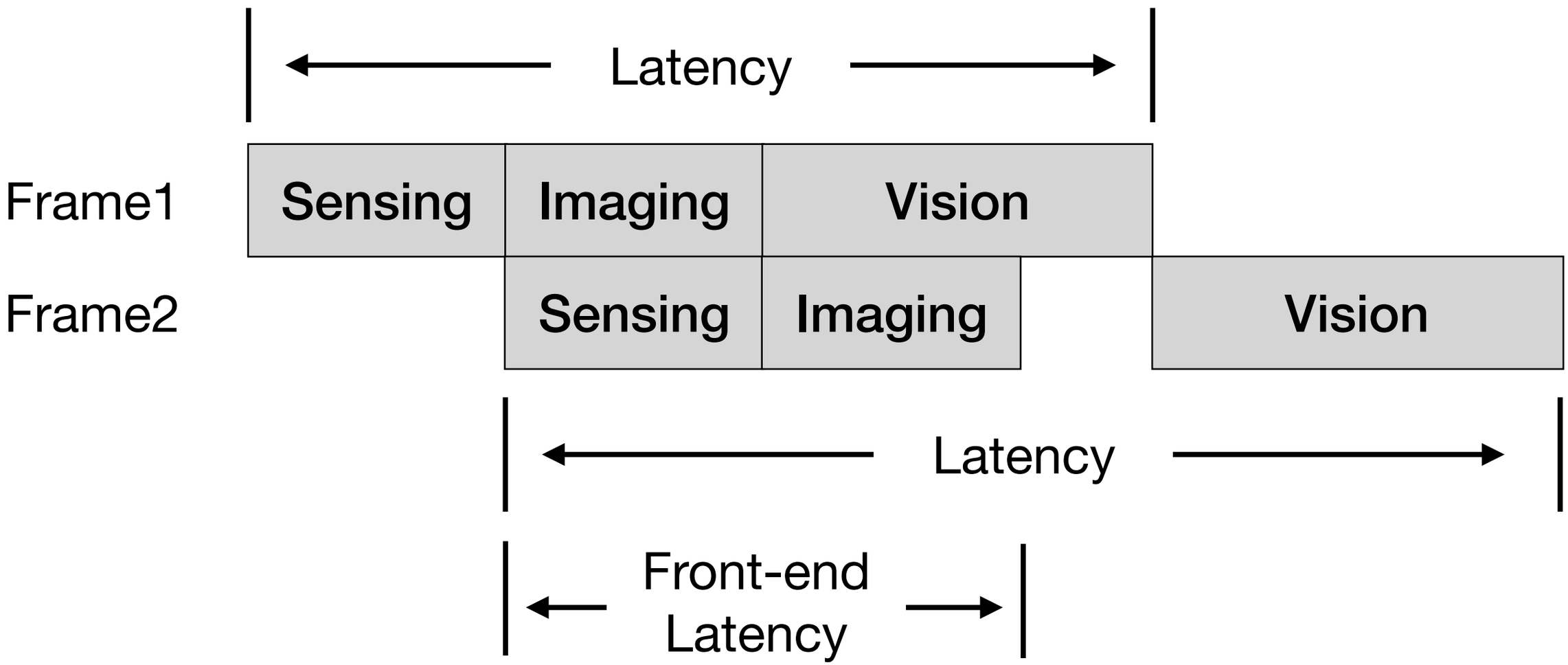
Latency in Continuous Vision Pipeline



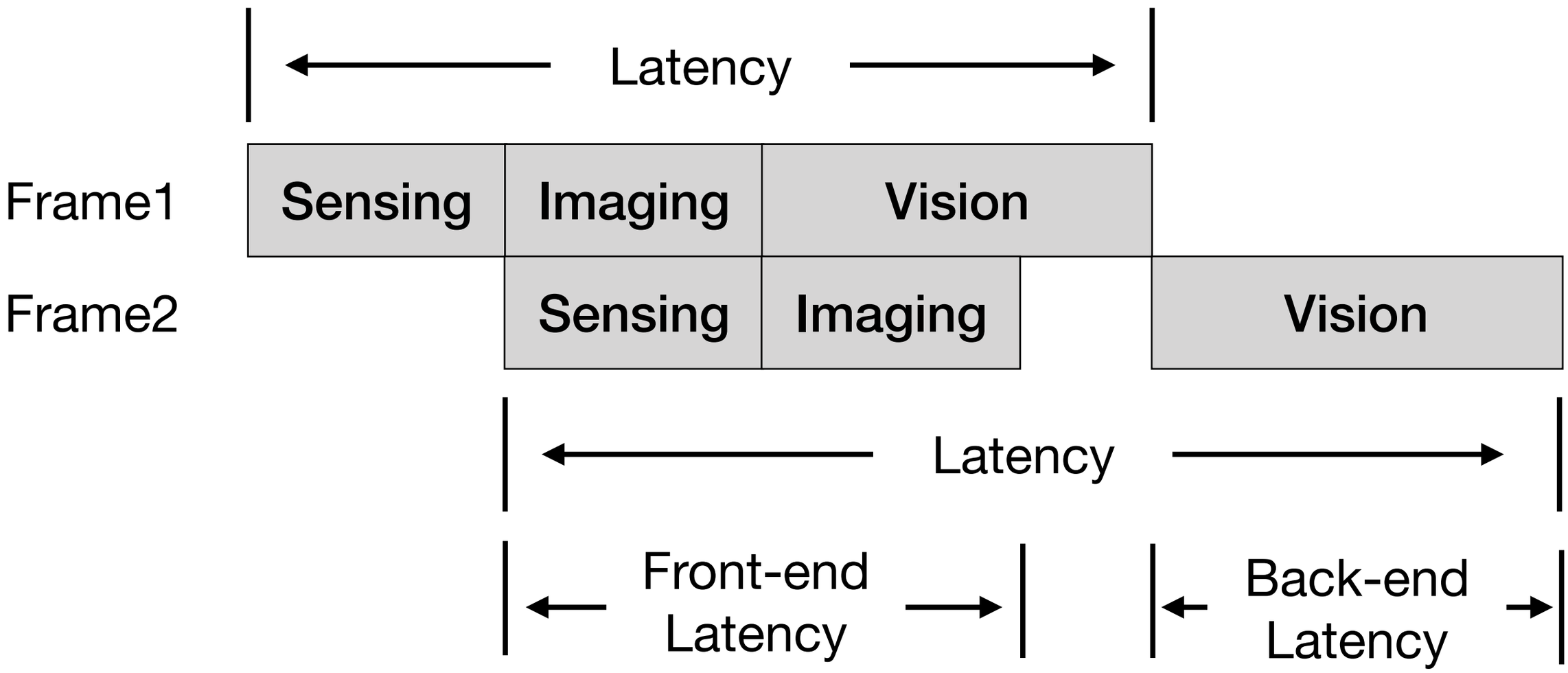
Latency in Continuous Vision Pipeline



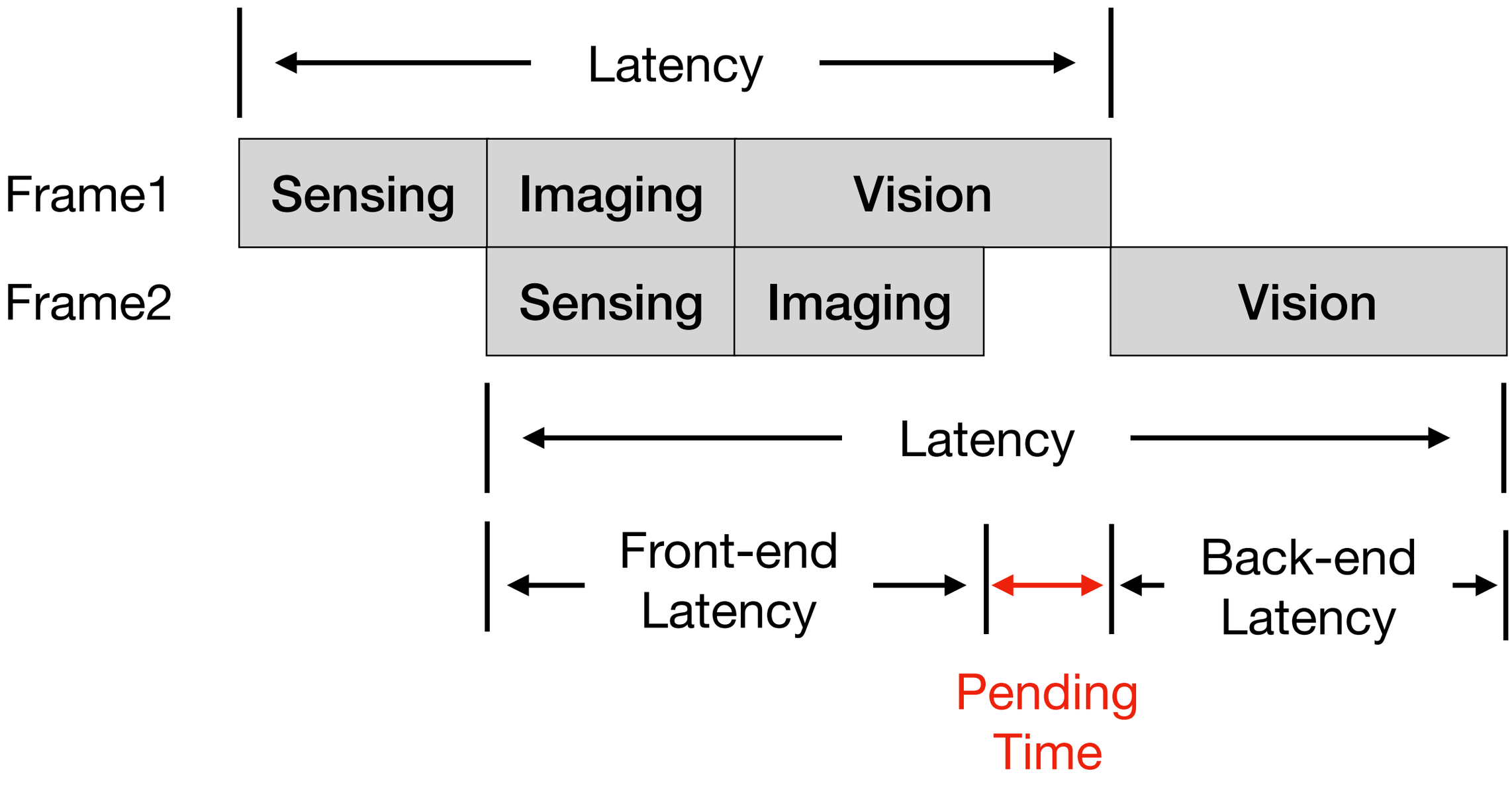
Latency in Continuous Vision Pipeline



Latency in Continuous Vision Pipeline



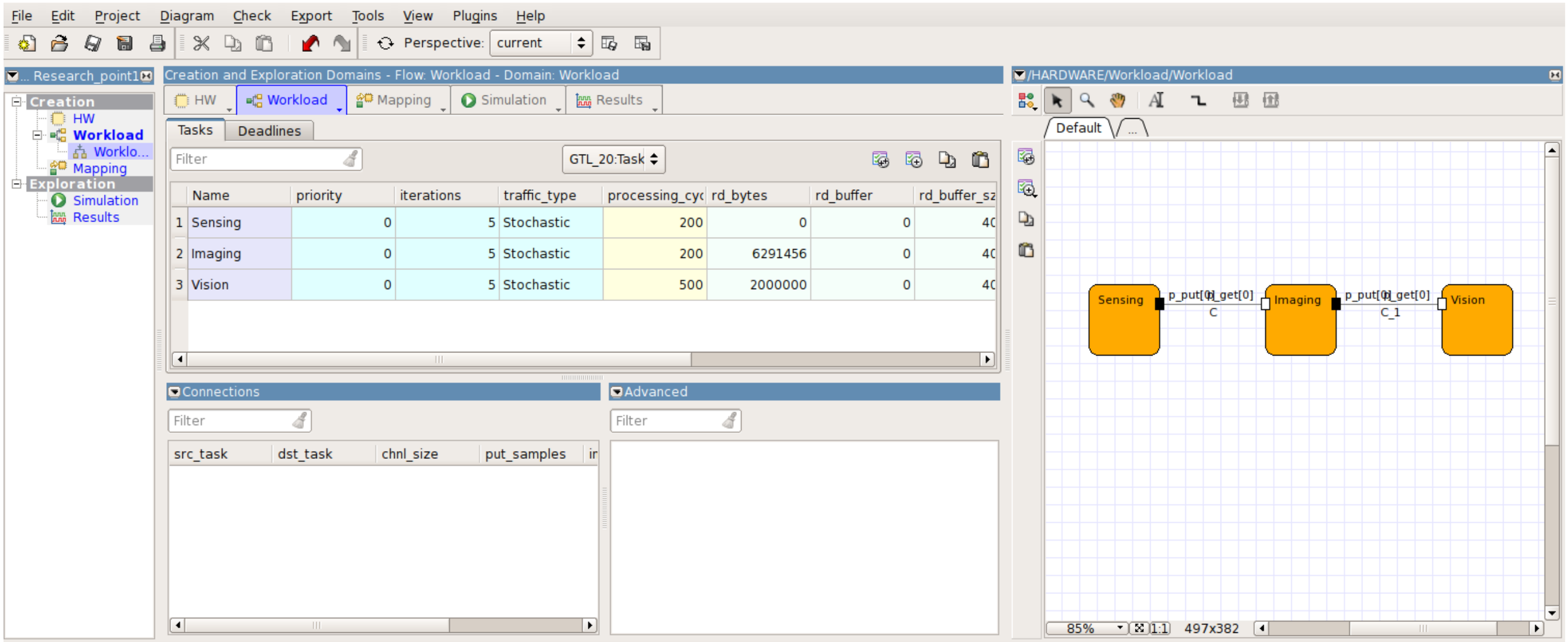
Latency in Continuous Vision Pipeline



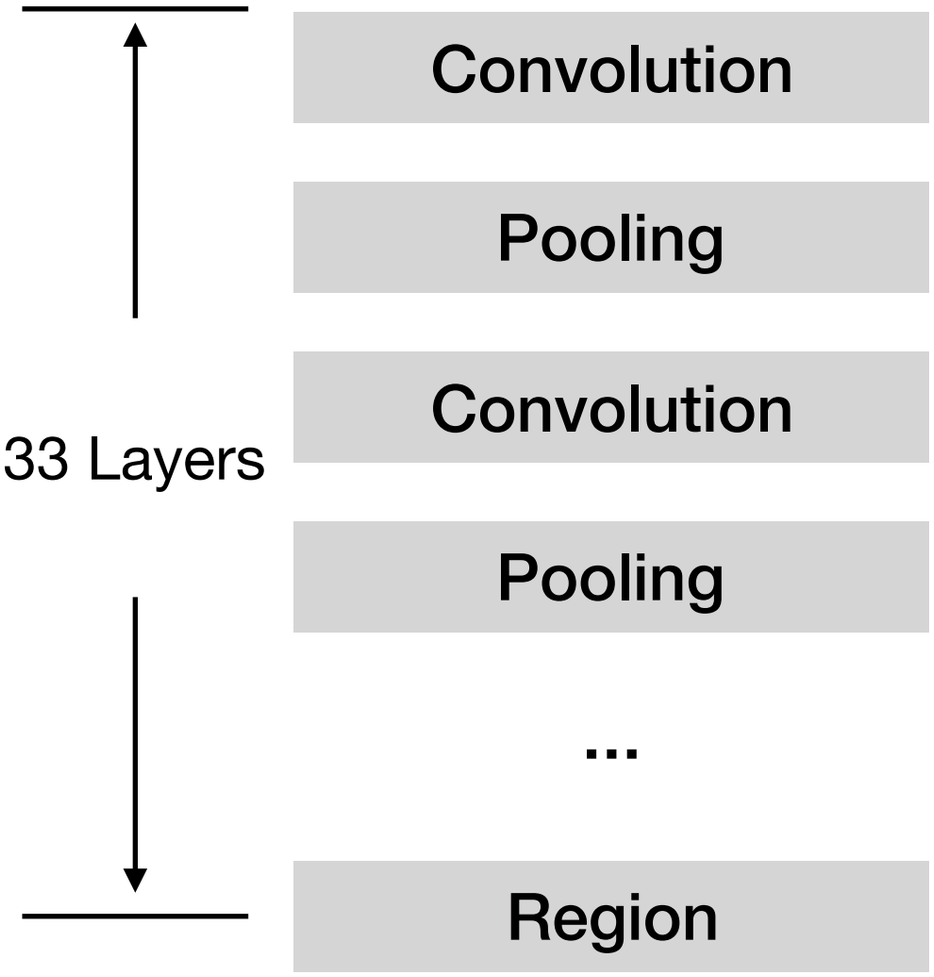
Platform Architect Modeling

Sensor	AR 1335 Sensor, 30 FPS AR1335 Datasheet
ISP	Nvidia TX2 ISP, 30 FPS
DNN Accelerator	20x20 Systolic Array 1.5 MB SRAM, 500 MHz
Bus	AMBA4 AXI4, 32 GB/s

Platform Architect Modeling

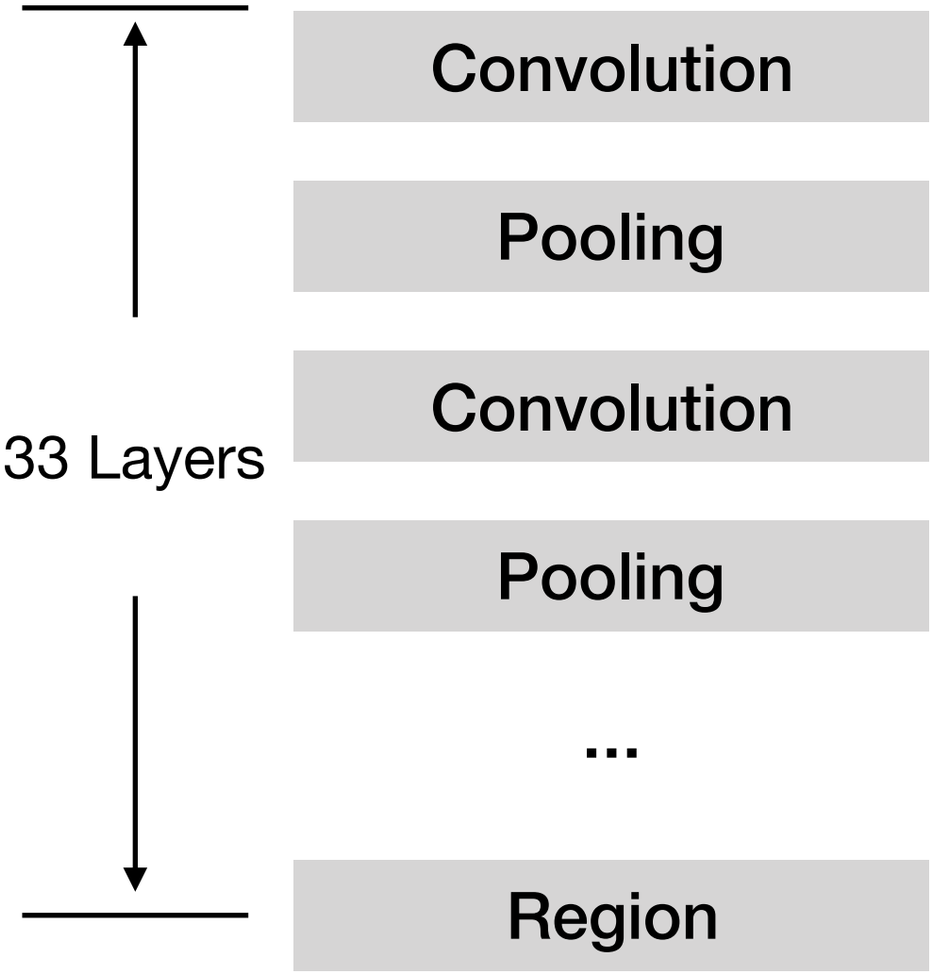


Different Vision Algorithms

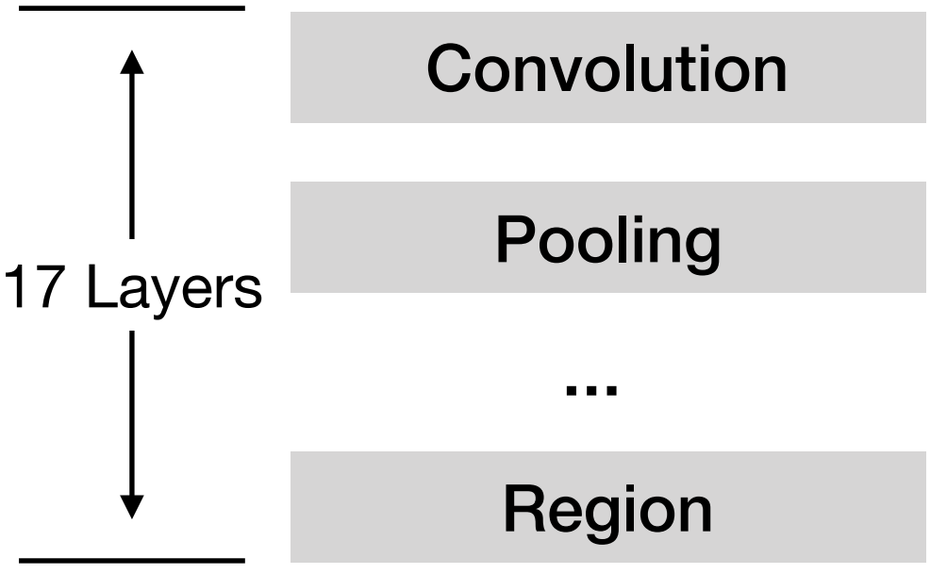


YOLOv2: 448x448 Input
36 Billion MAC Ops

Different Vision Algorithms



YOLOv2: 448x448 Input
36 Billion MAC Ops

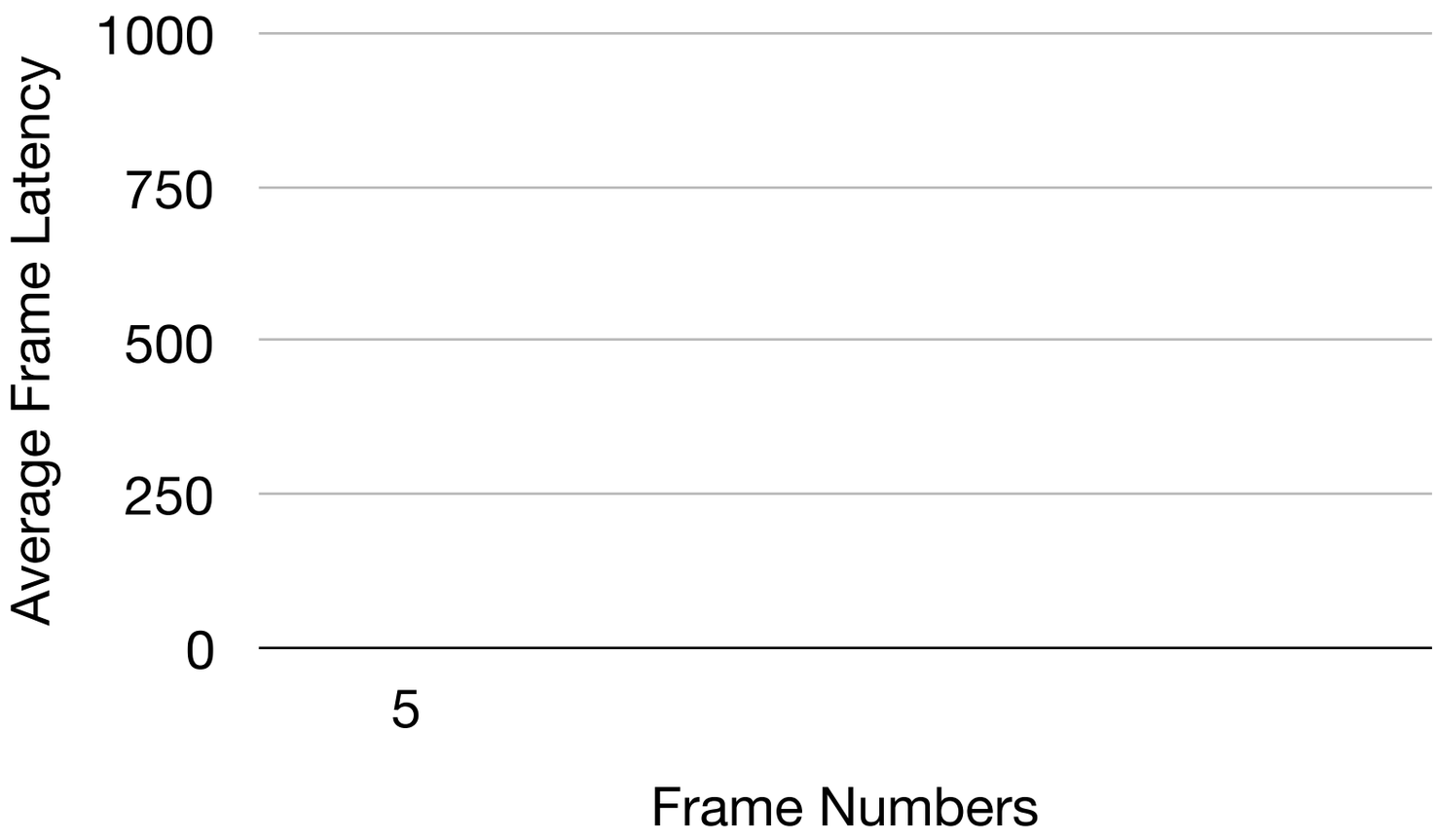


TinyYOLO: 448x448 Input
7 Billion MAC Ops

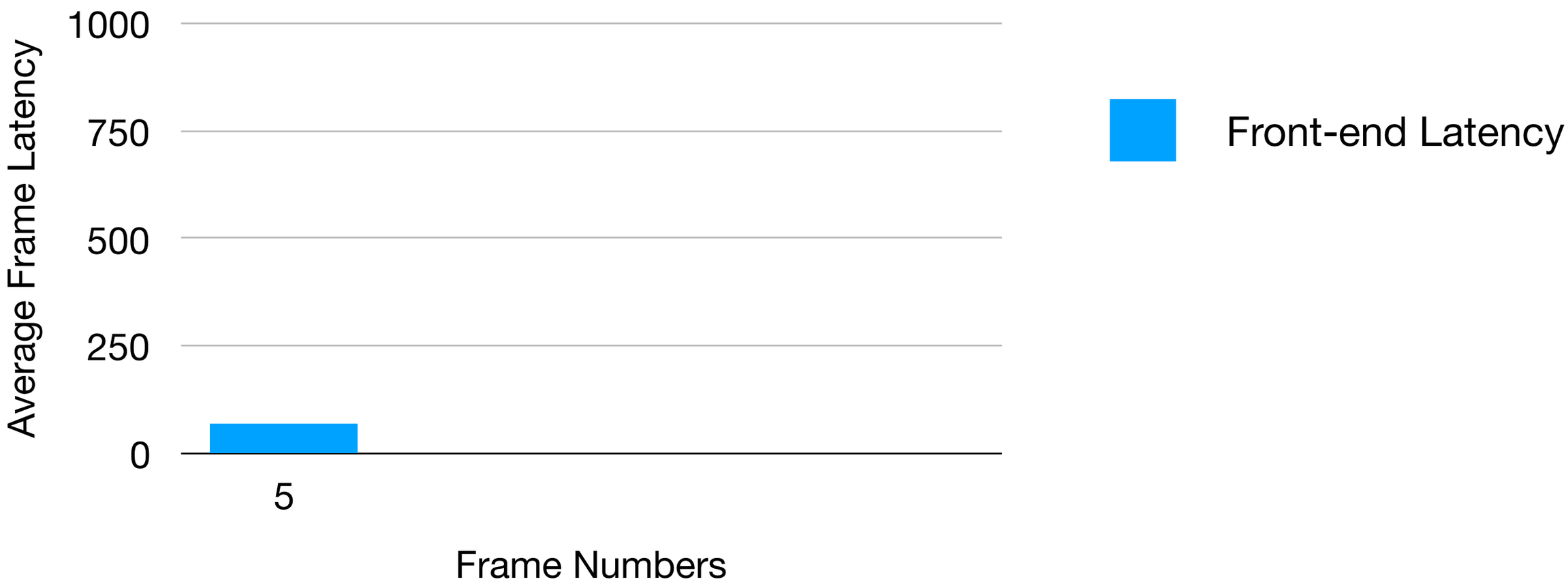
Task Modeling for YOLOv2

Task	Sensing	Imaging	Vision
Latency (ms)	34	34	102

Latency Results for YOLOv2



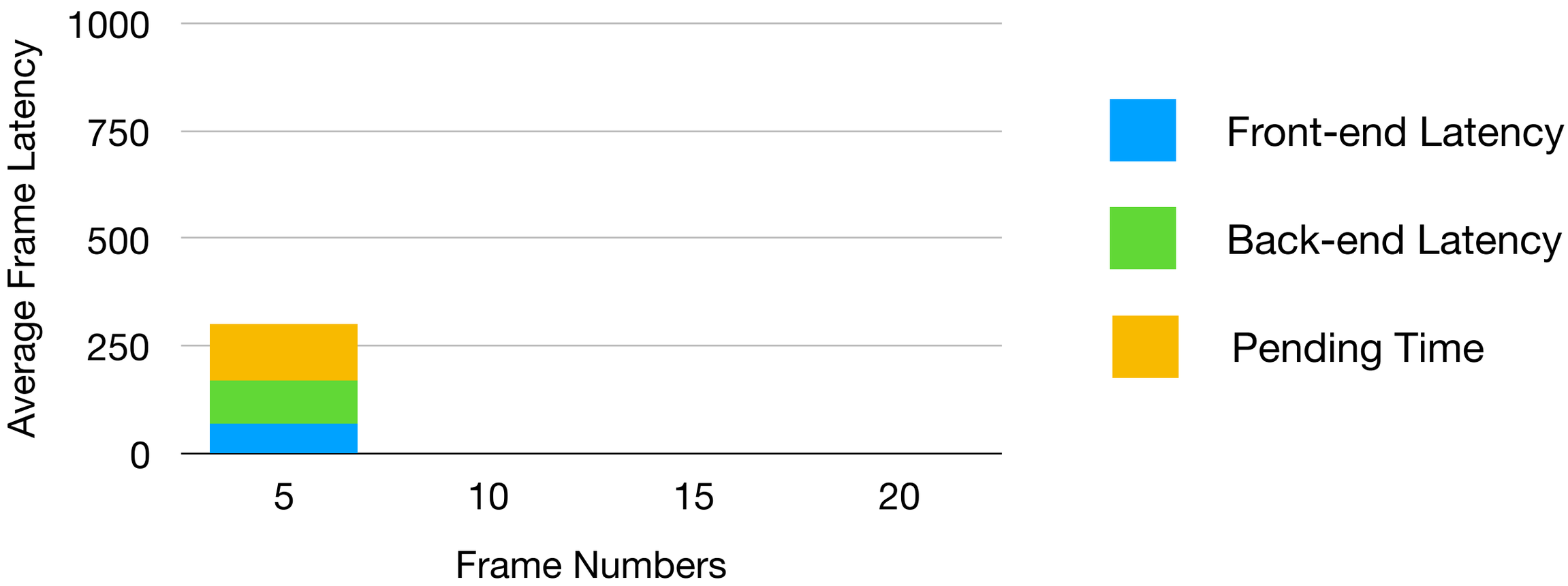
Latency Results for YOLOv2



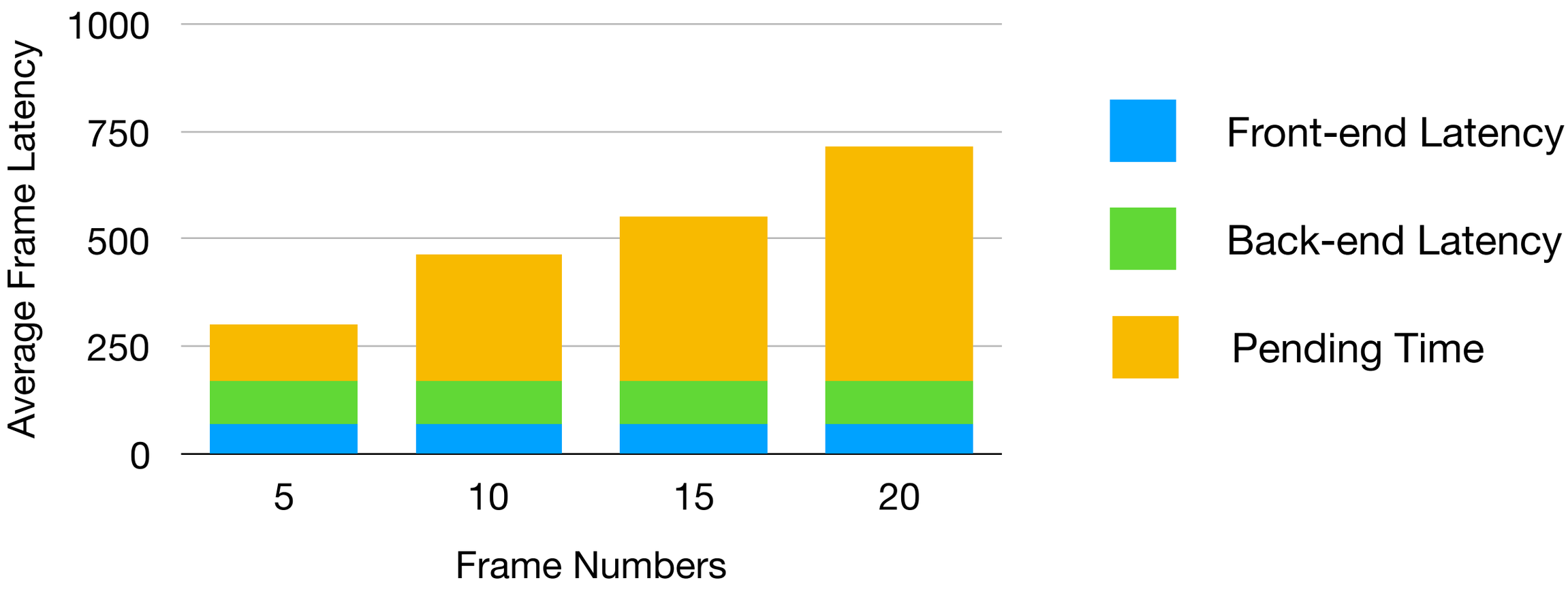
Latency Results for YOLOv2

Latency Results for YOLOv2

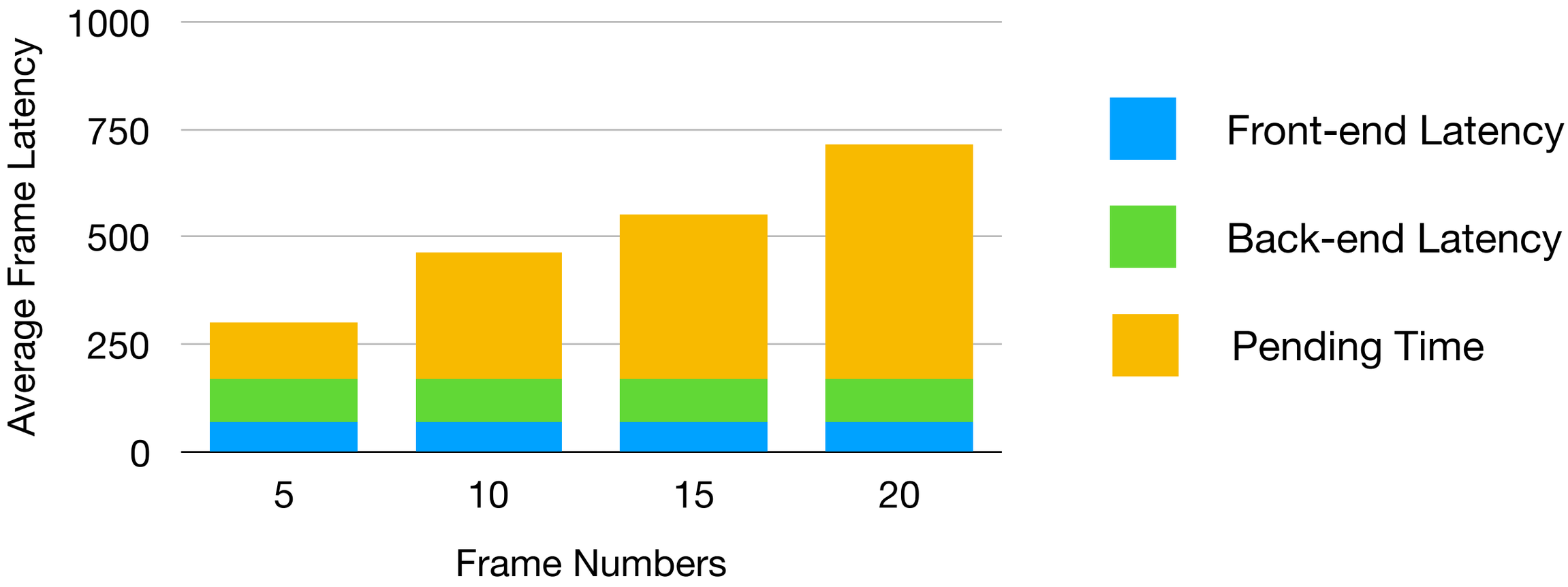
Latency Results for YOLOv2



Latency Results for YOLOv2



Latency Results for YOLOv2

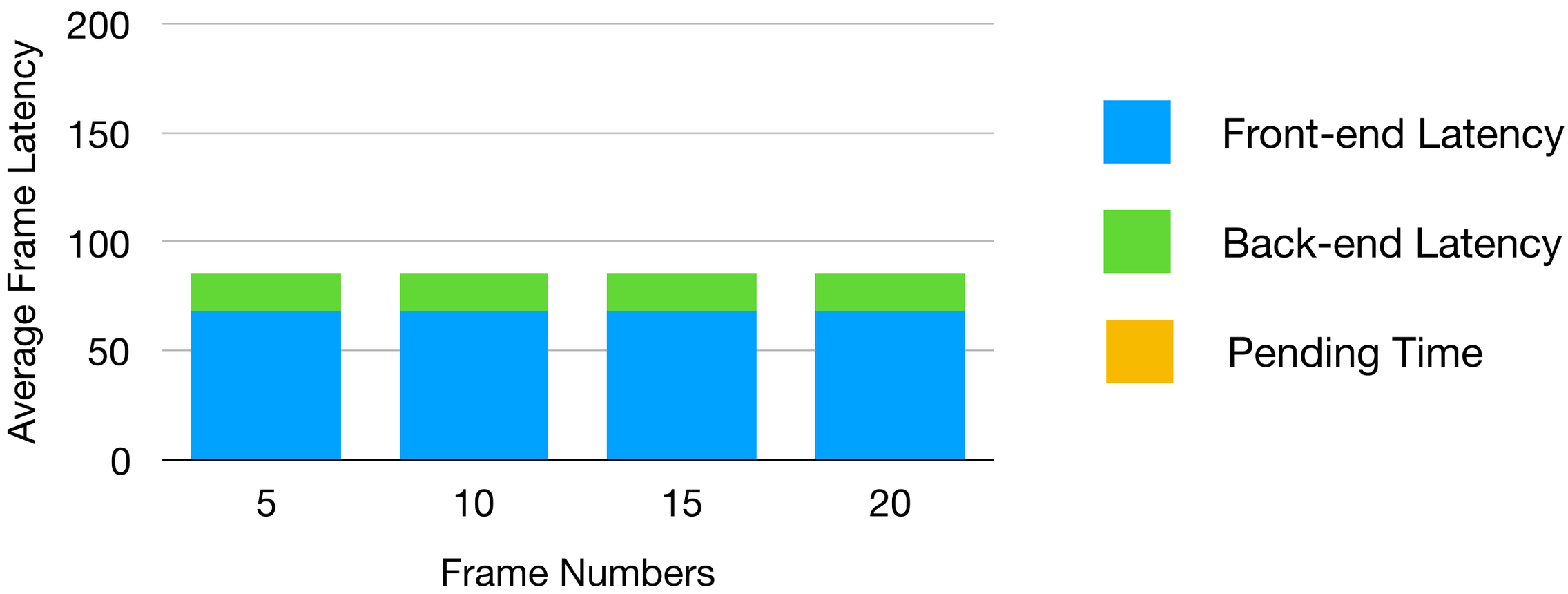


Conclusion 1: When the front-end is faster than the back-end, pending time is the bottleneck. Reducing pending time can significantly reduce the per frame latency.

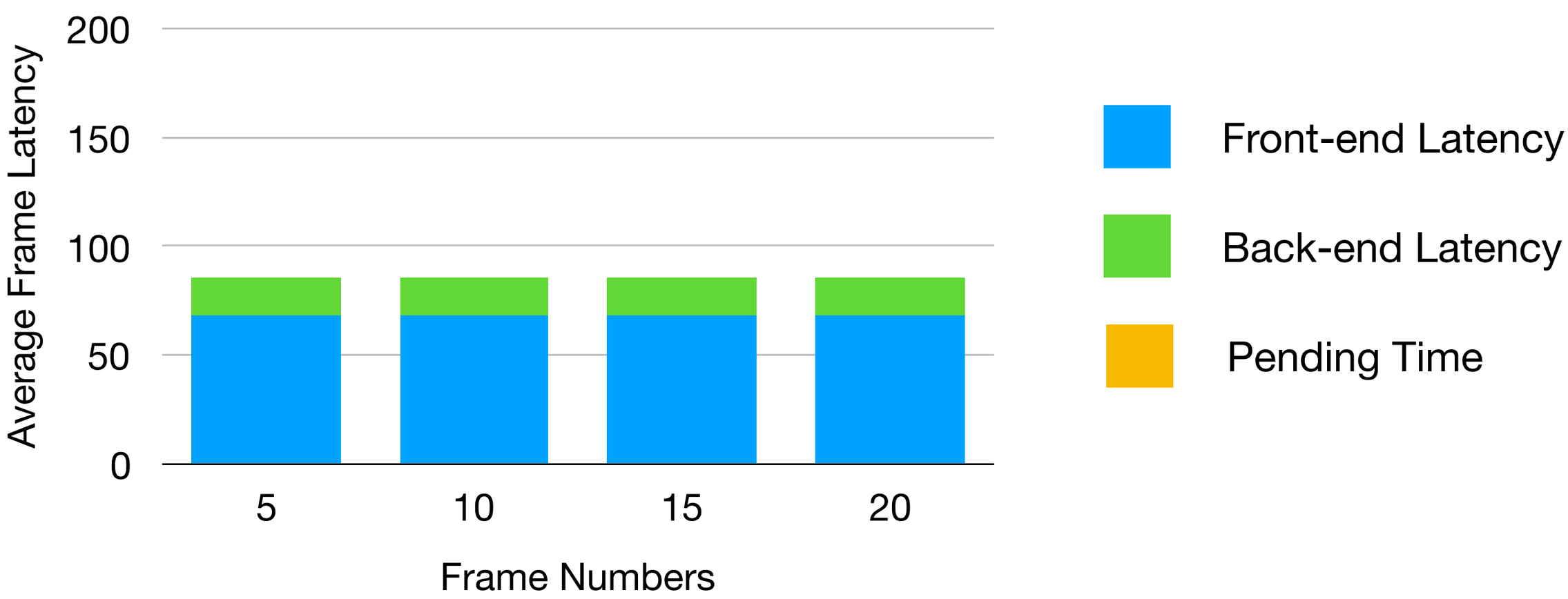
Task Modeling for TinyYOLO

Task	Sensing	Imaging	Vision
Latency (ms)	34	34	19.6

Latency Results for TinyYOLO

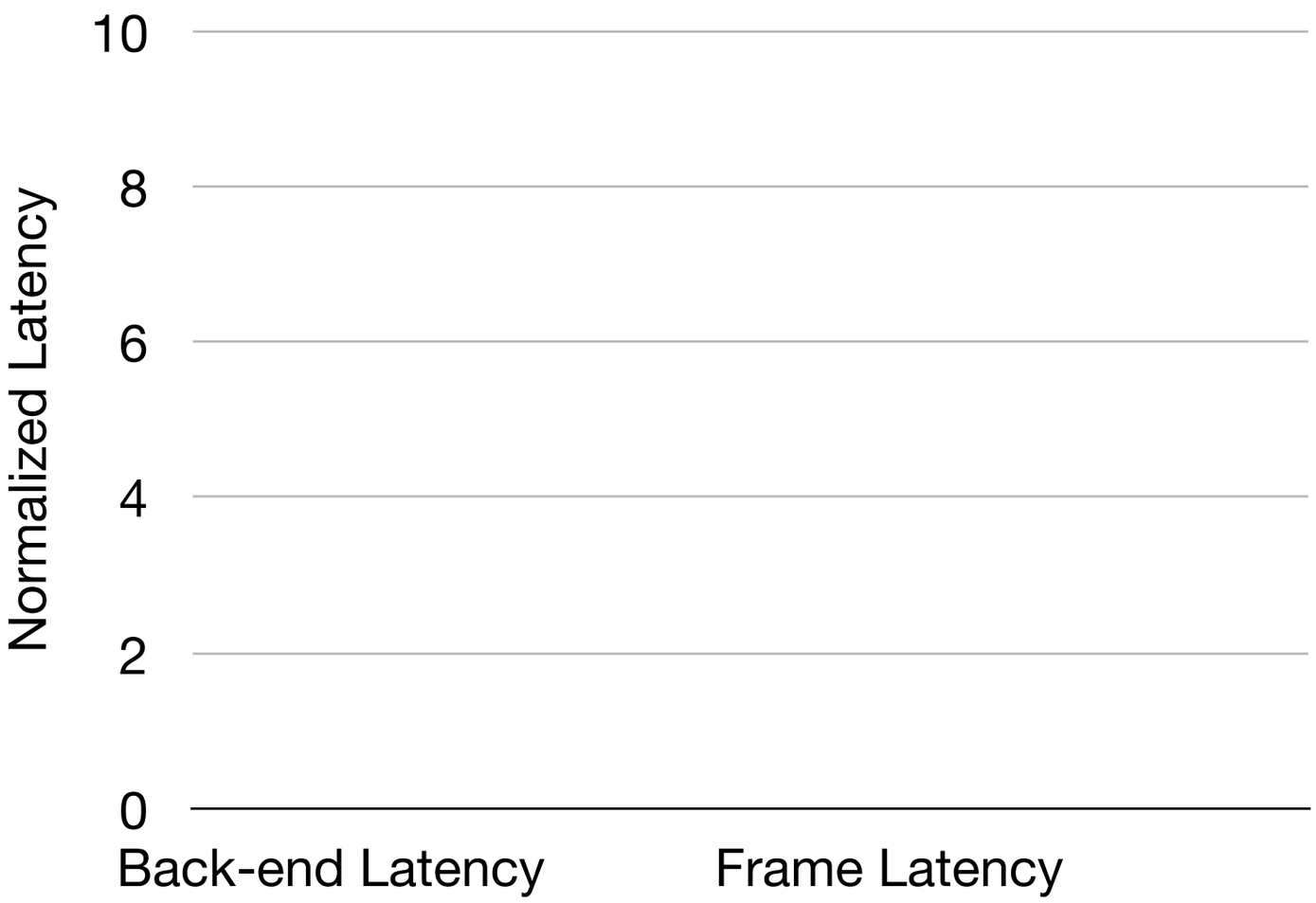


Latency Results for TinyYOLO

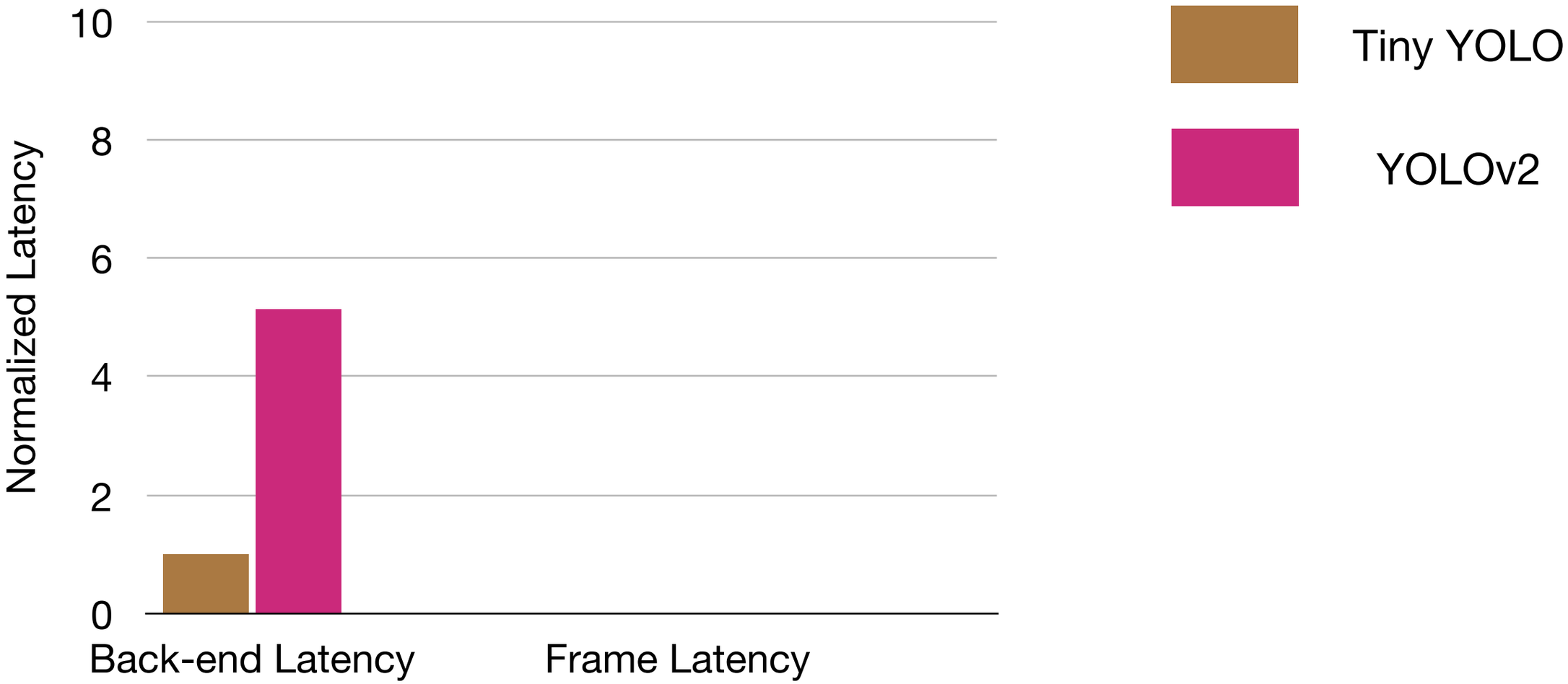


Conclusion 2: When the front-end is slower than the back-end, NPU is not fully utilized. The front-end is the bottleneck and optimizing the vision stage has marginal effect.

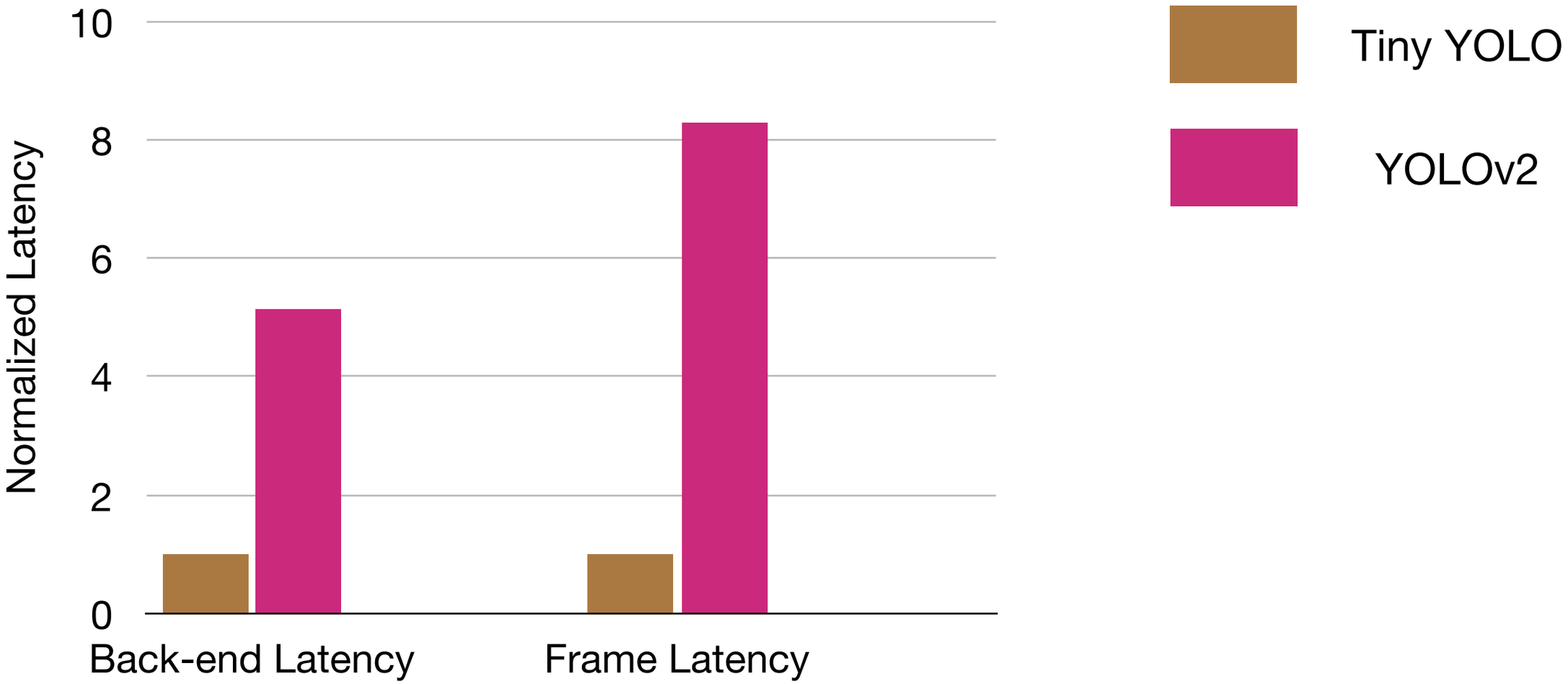
Comparison for 20 Frames



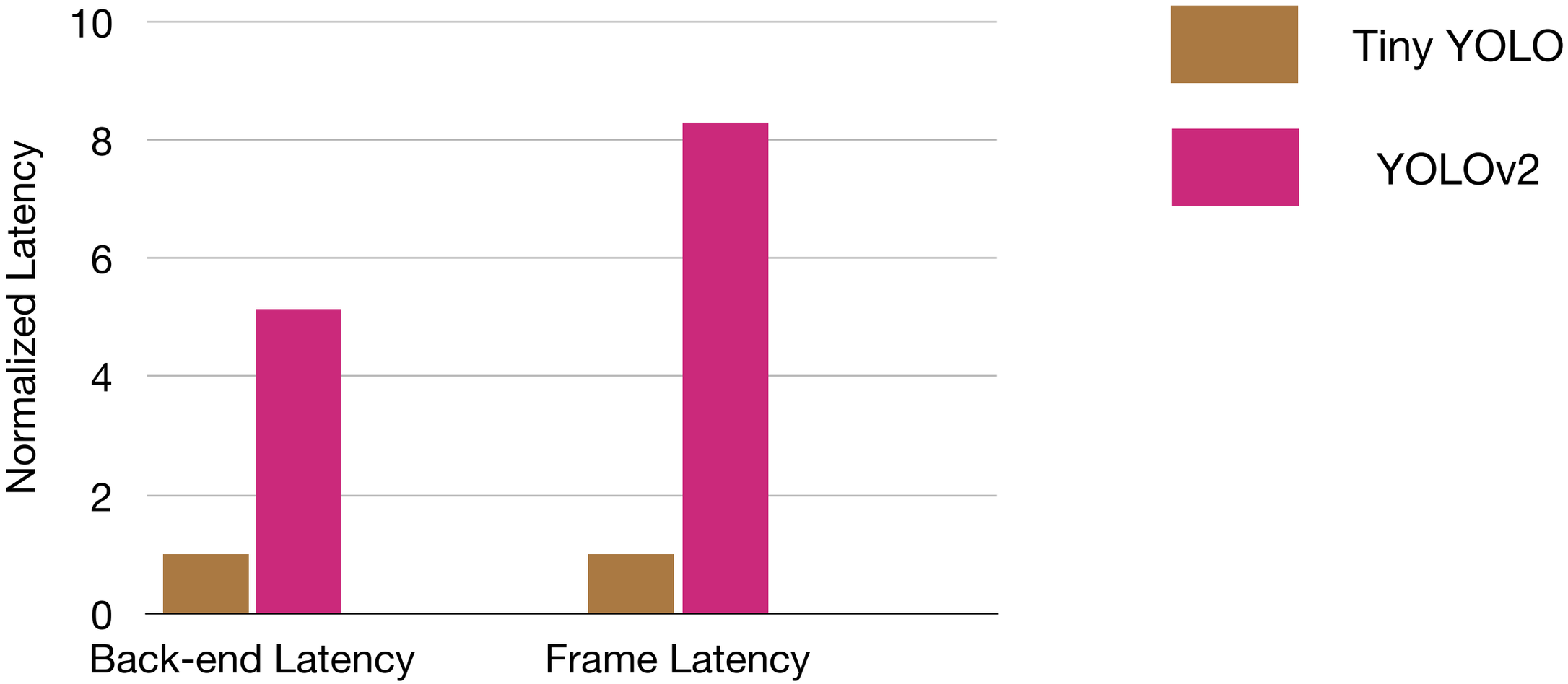
Comparison for 20 Frames



Comparison for 20 Frames



Comparison for 20 Frames



Conclusion 3: Frame latency is not proportional to back-end latency, even with the same front-end. We have to focus on end-to-end latency not just vision stage.