DASYA

RAD

IT UNIVERSITY OF COPENHAGEN

www.dasya.dk
@dasyaITU

rad.itu.dk

www.itu.dk

different scales of resource-aware deep learning & how to tackle them

Pınar Tözün

Associate Professor, IT University of Copenhagen

pito@itu.dk, pinartozun.com, @pinartozun

INDEPENDENT RESEARCH FUND DENMARK

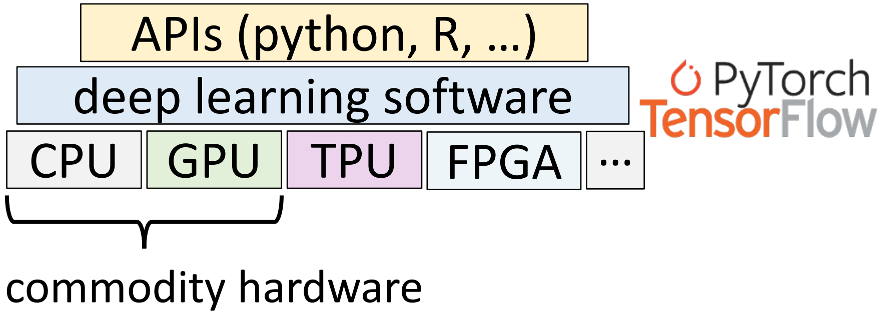
novo nordisk foundation

X_Event/Place_X
xx/xx/2023

unsustainable growth of deep learning

2012 → **2019**

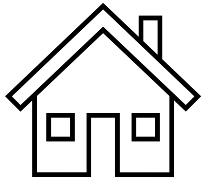
- powerful hardware
- larger datasets
- deep learning frameworks



> 300000x increase in ***computational need*** for deep learning models.

today

estimated carbon footprint for 13% training of a large language model **= average yearly energy of a US home**

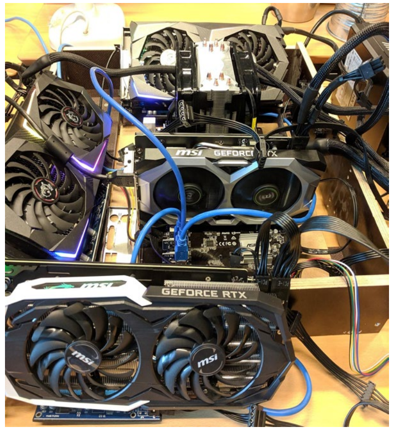


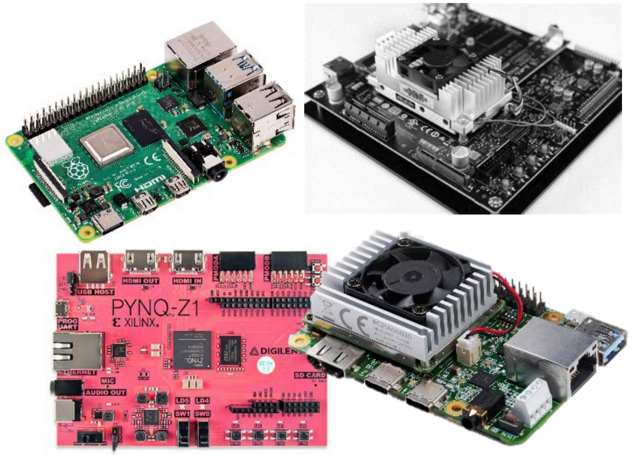
need for higher computational efficiency!

sources: https://openai.com/blog/ai-and-compute/, Dodge et al. "Measuring the Carbon Intensity of AI in Cloud Instances." FAccT 2022

hardware scales for deep learning

large ⟷ **tiny**



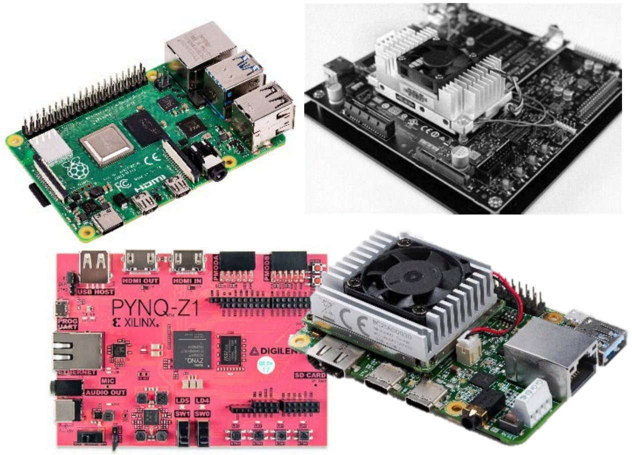


➔can we utilize these hardware well?

➔can we do more with less?

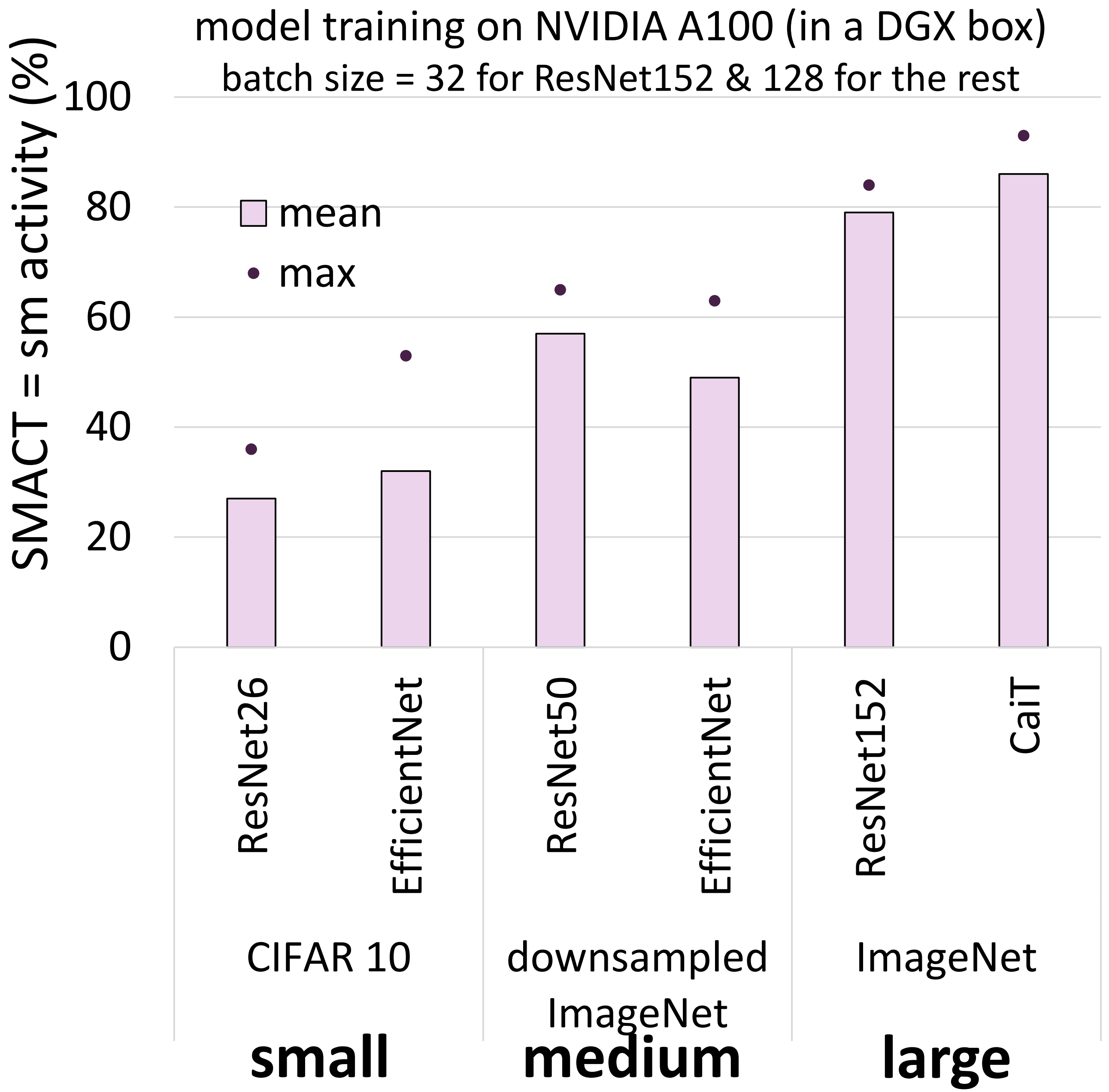
hardware scales for deep learning

large ⟷ **tiny**



➔can we utilize these hardware well?
➔can we do more with less?

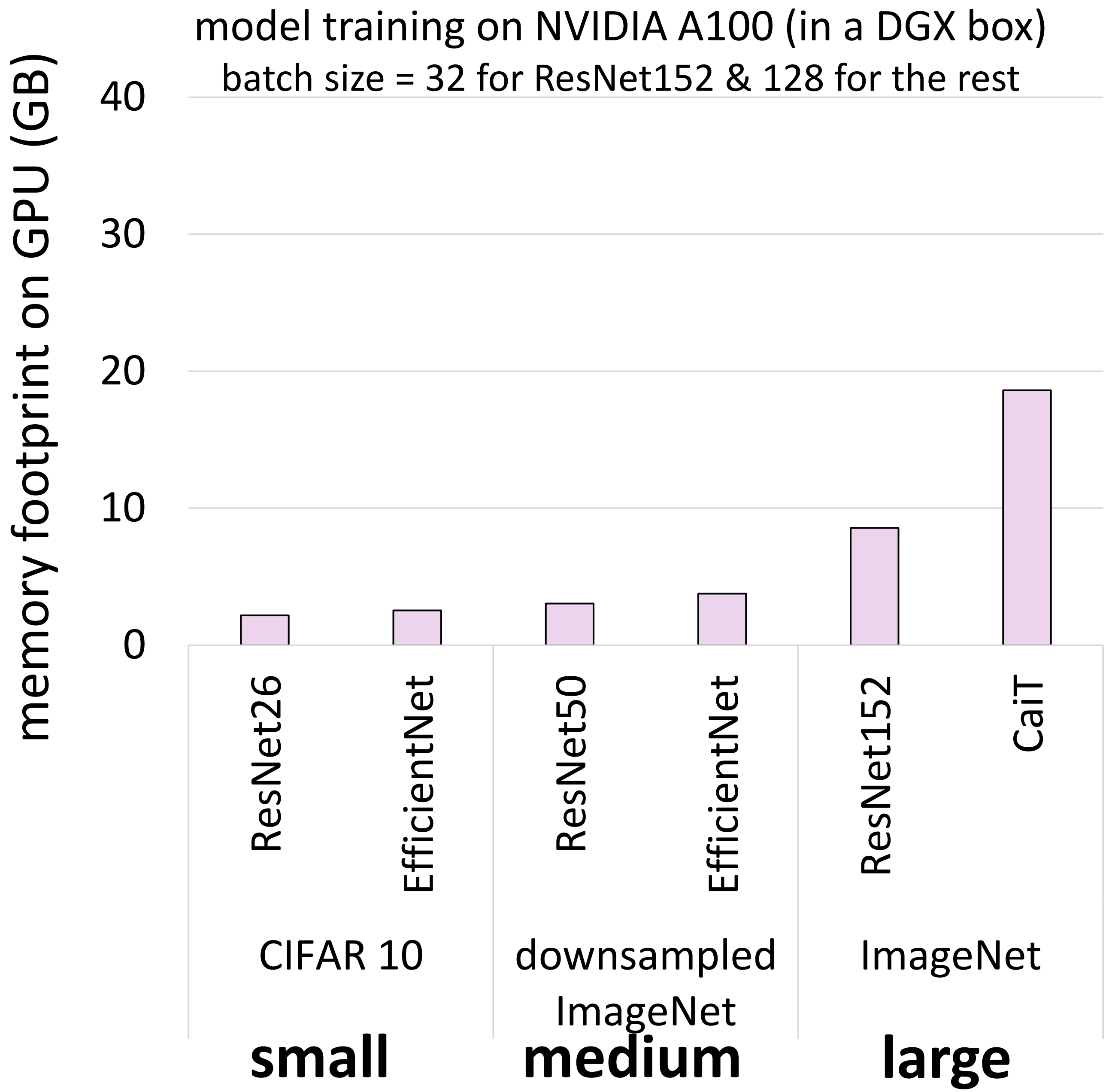
how well we utilize the hardware?



- **@ITU, jobs of data scientists in our lab falls under *small* case – transfer learning, small models**
- **in real-world*, ~52% GPU utilization on average for 100,000 jobs**
- **not all training scenarios fully utilize modern GPUs**

* Jeon et al. "**Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads**." *USENIX ATC* 2019.

how well we utilize the hardware?



- **@ITU, jobs of data scientists in our lab falls under *small* case – transfer learning, small models**
- **in real-world*, ~52% GPU utilization on average for 100,000 jobs**
- **not all training scenarios fully utilize modern GPUs**

* Jeon et al. "**Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads**." *USENIX ATC* 2019.

workload collocation

= multiple workloads sharing hardware resources

benefits when a single workload cannot utilize available resources well / fully

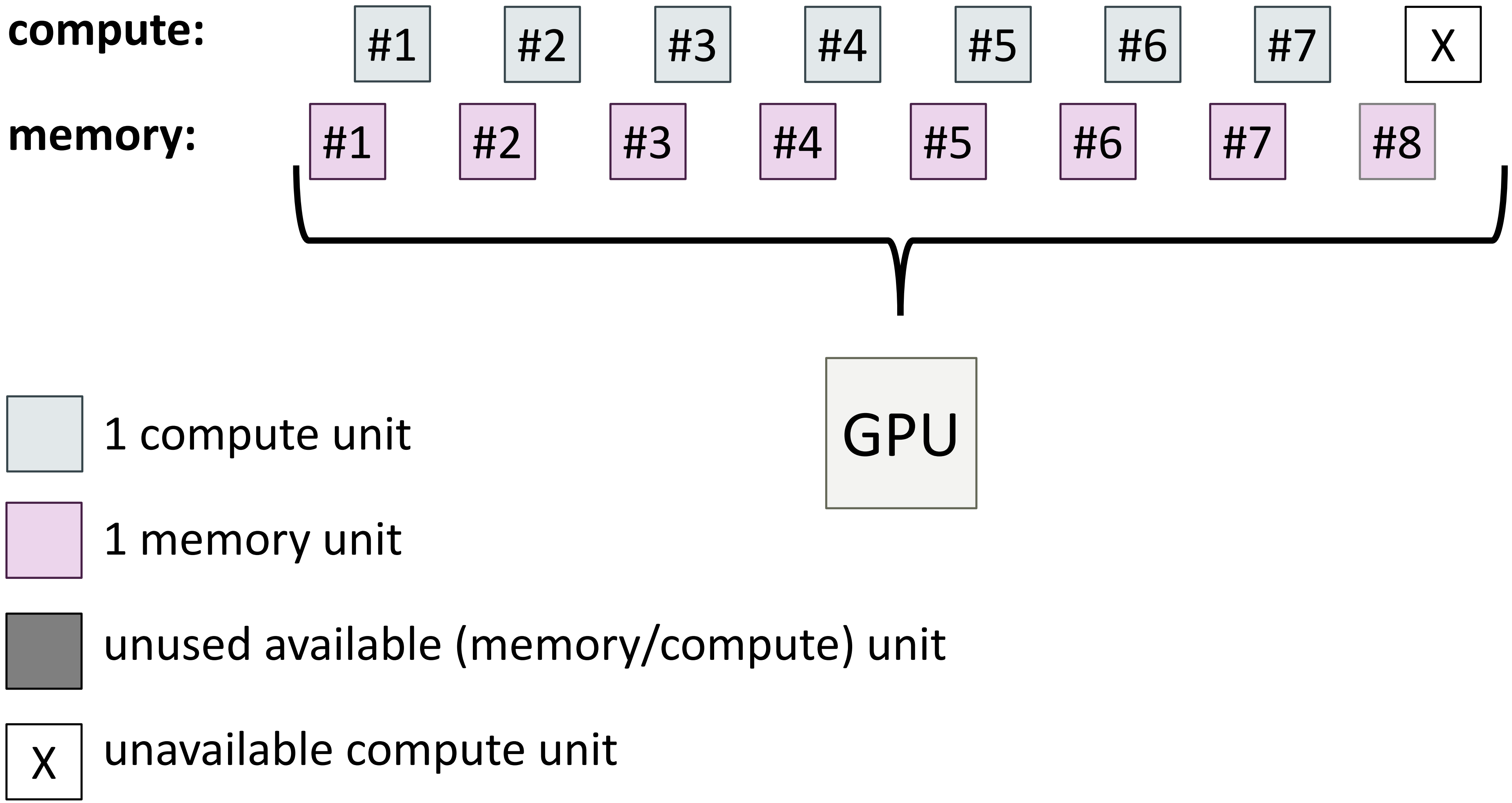
usual challenge → interference across workloads

GPU-specific challenge → no fine-grained & flexible resource sharing mechanism

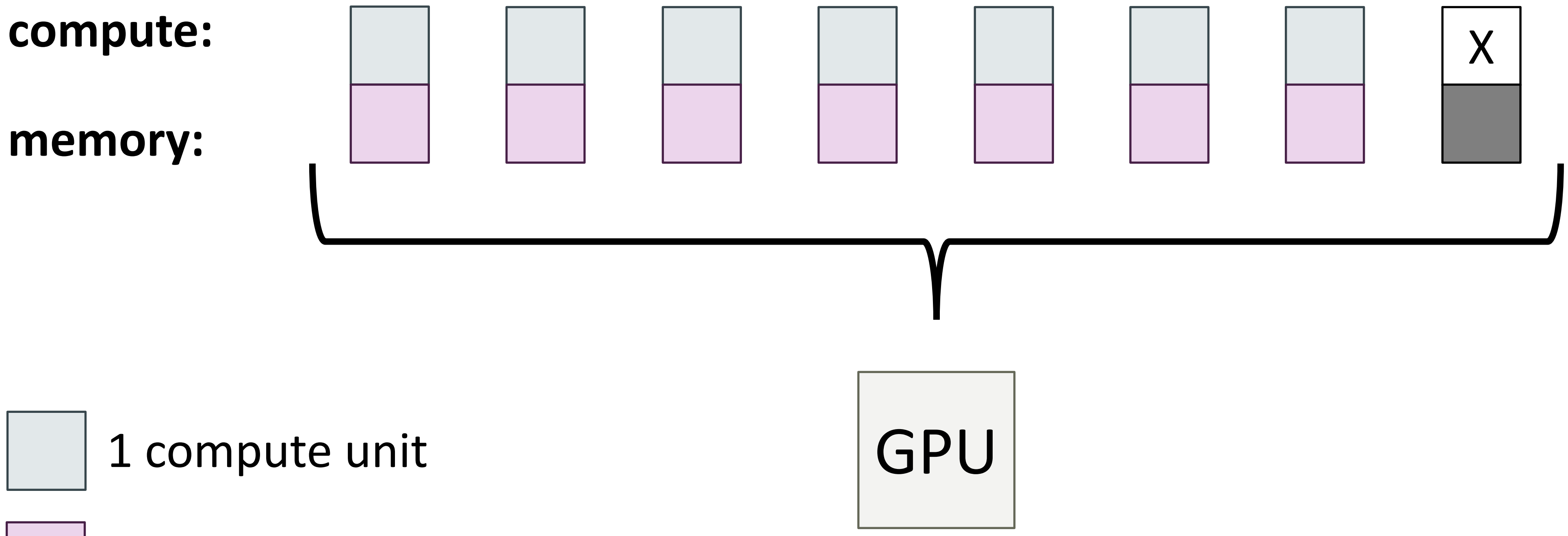
workload collocation on (NVIDIA) GPUs

- ***naïve collocation & virtualization***
  - kernels of different applications are serialized
  - × provides limited parallelism
- ***multi-process service (MPS)***
  - GPU resources are split (manually or automatically) across applications
  - ✓ kernels of different applications can run simultaneously
  - × allowed for one user only (for safety reasons)
- ***multi-instance GPU (MIG)***
  - hardware support for resource split, introduced with NVIDIA A100
  - ✓ prevents interference & can do all the above in a MIG partition
  - × rigid partitioning of GPU resources

multi-instance GPU



multi-instance GPU



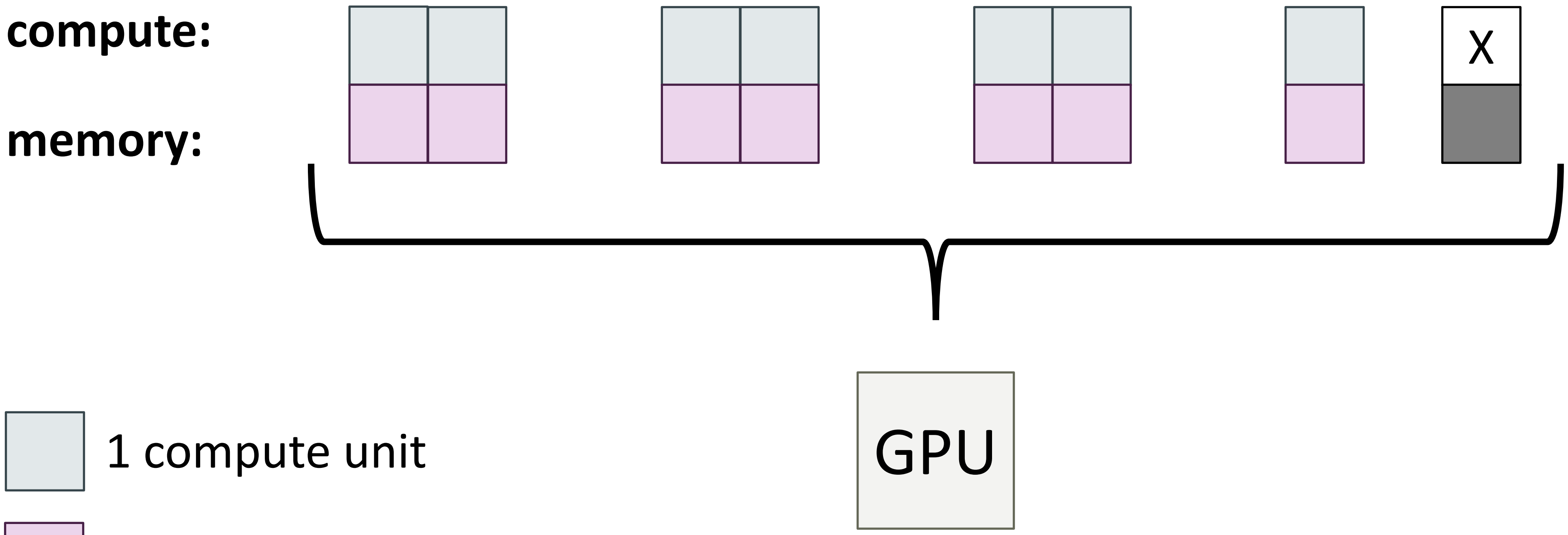
1 compute unit

1 memory unit

unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU



1 compute unit

1 memory unit

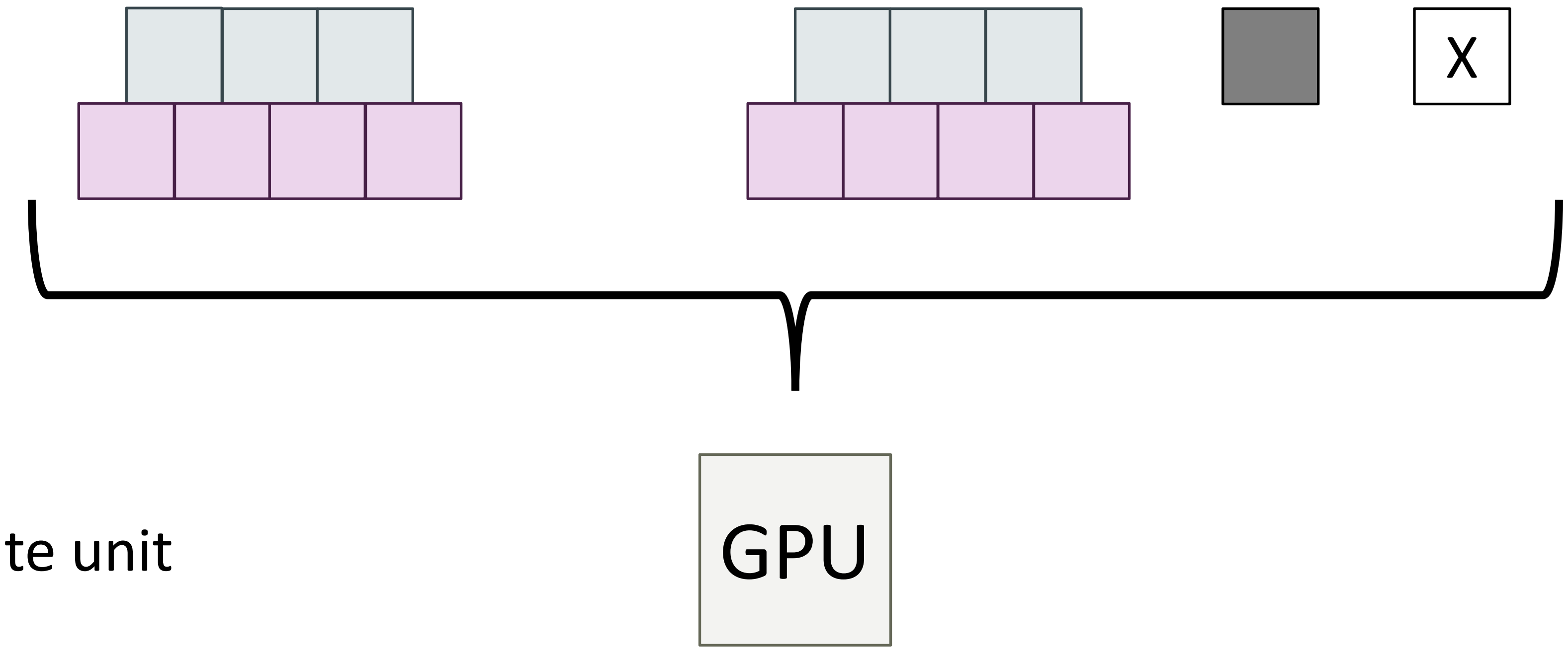
unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU

compute:

memory:



1 compute unit

1 memory unit

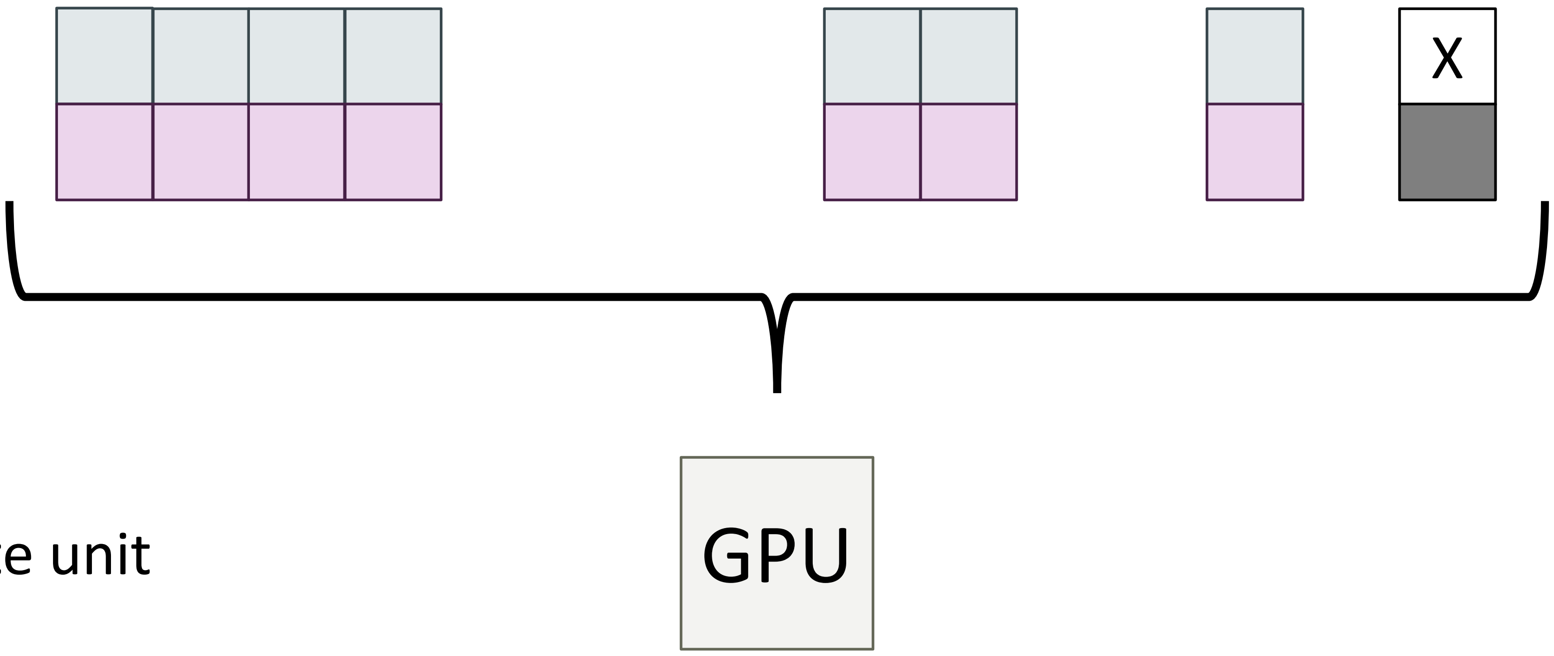
unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU

compute:

memory:



1 compute unit

1 memory unit

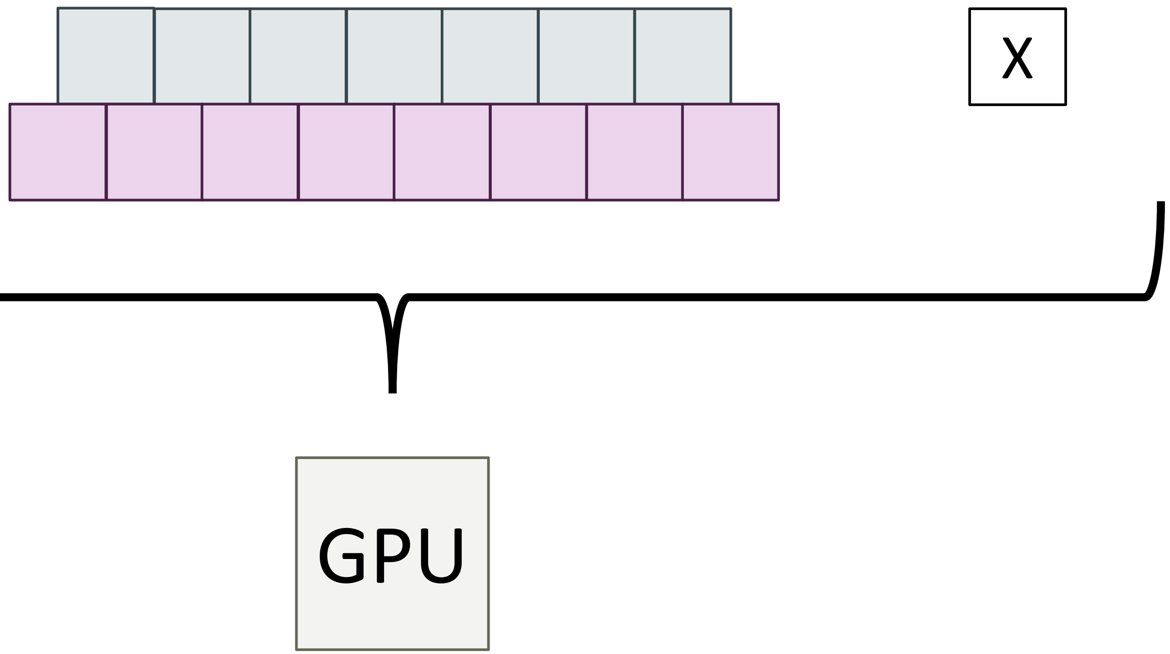
unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU

compute:

memory:



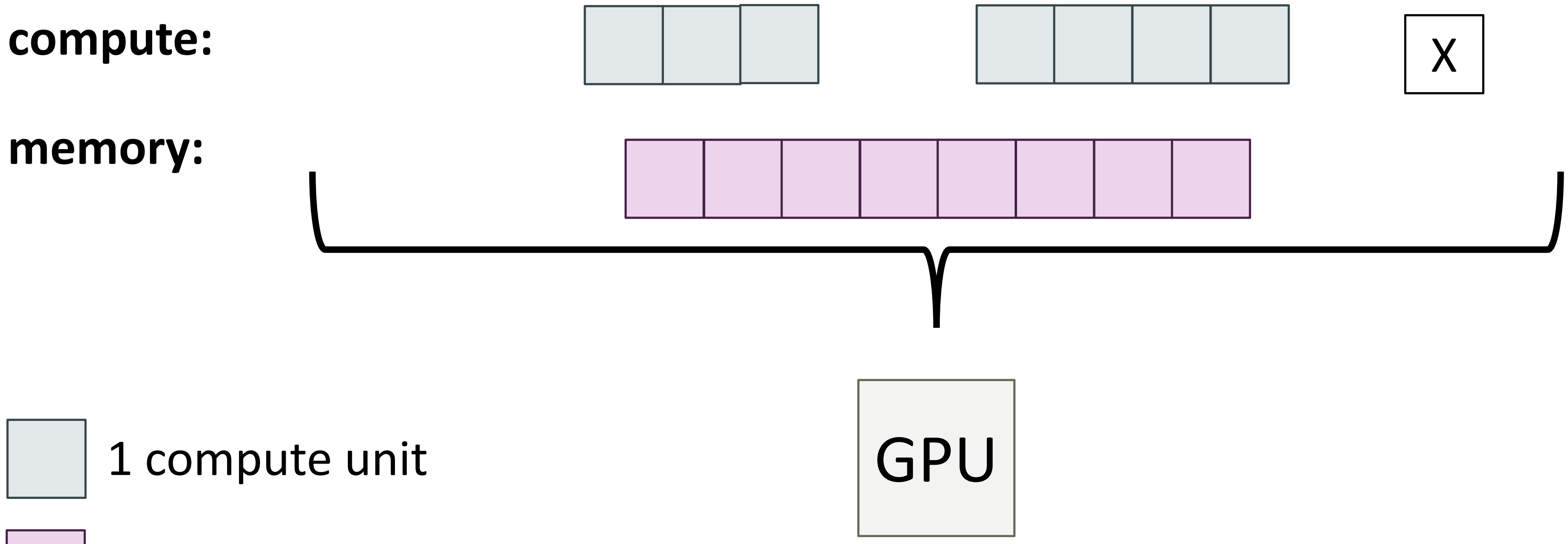
1 compute unit

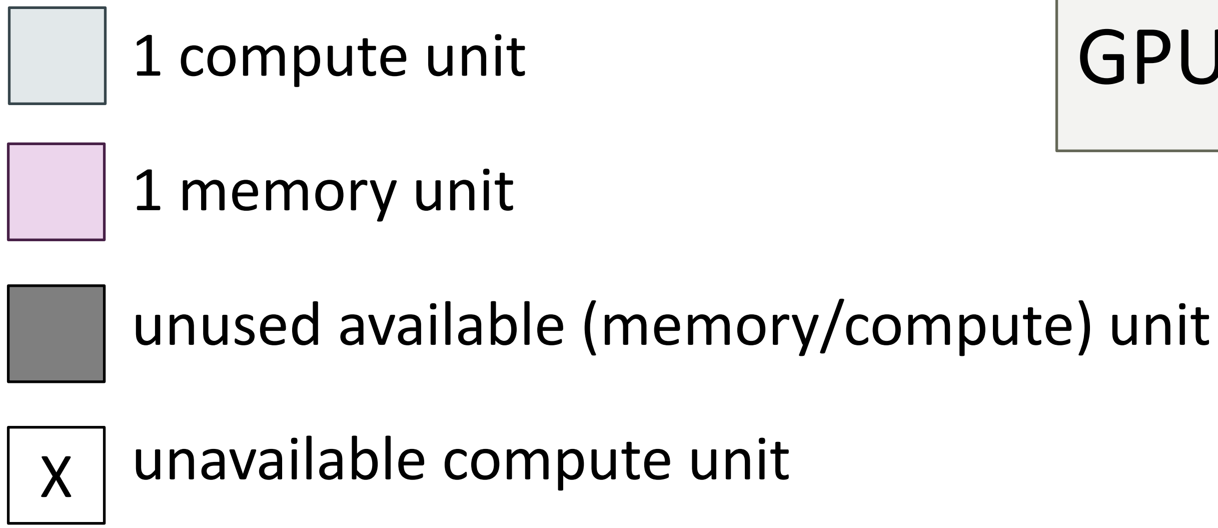
1 memory unit

unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU

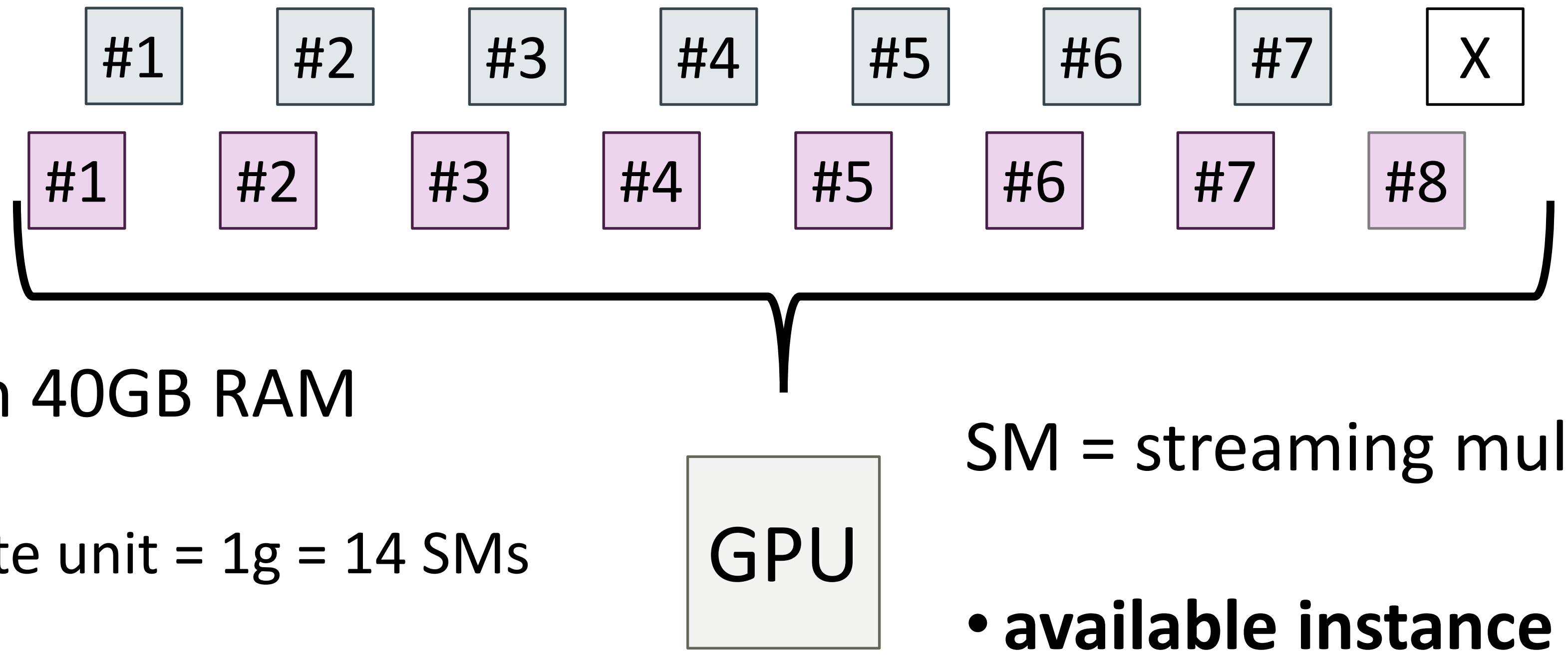




multi-instance GPU

compute:

memory:



on A100 with 40GB RAM

- 1 compute unit = 1g = 14 SMs
- 1 memory unit = 5GB
- unused available (memory/compute) unit
- X unavailable compute unit = 10 SMs

SM = streaming multiprocessor

- **available instance profiles differ across GPUs that support MIG**
- **doesn't allow distributed training on A100s**

performance impact of collocation?

NVIDIA DGX Station A100

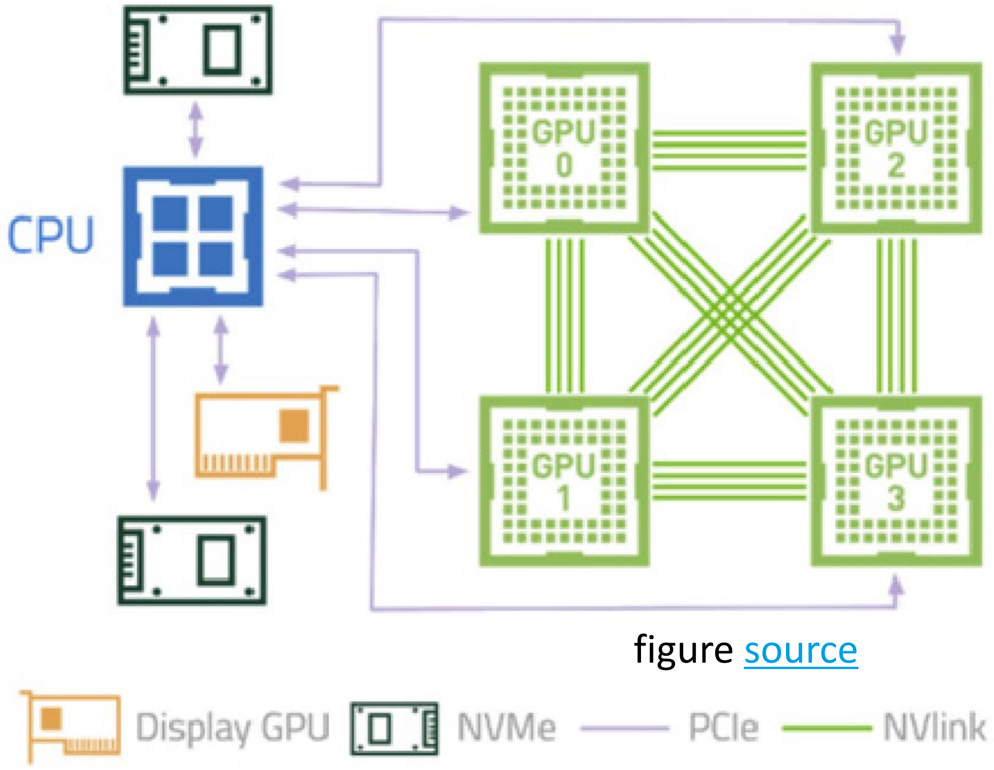


figure source

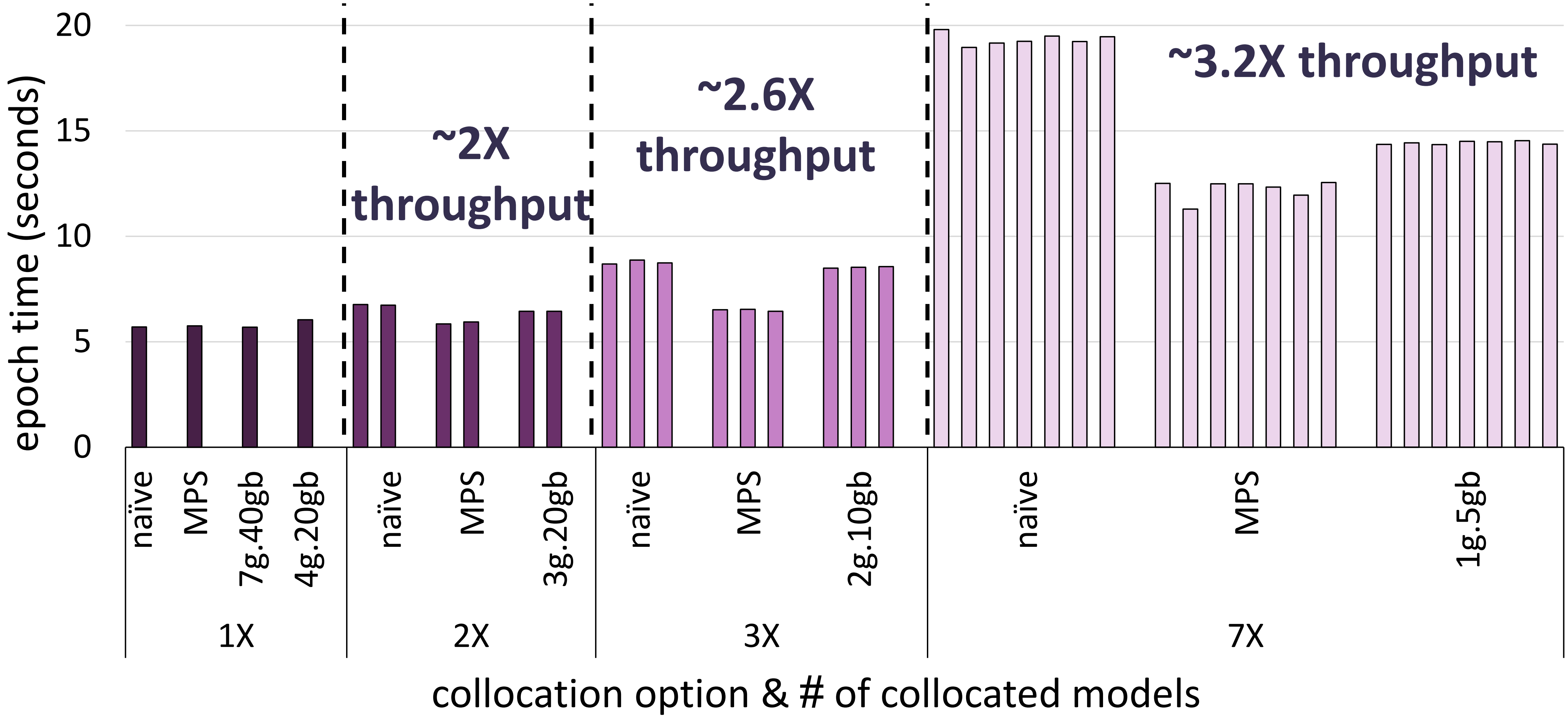
CPU = AMD 7742 – 512 GB RAM
64 physical cores
GPU = NVIDIA A100 – 40 GB RAM

workloads	model	batch size	dataset
small	**ResNet26** EfficientNet	128	CIFAR-10
medium	**ResNet50** EfficientNet	128	downsampled ImageNet*
large	**ResNet152** CaiT	32 128	ImageNet (2012)
xlarge	**DLRM**	1	Criteo Terabyte

- image models: CNN & transformers + recommender model
- on single GPU with PyTorch v2.0
- results reported from 2nd epoch of training
- nvidia-smi & dcgm as monitoring tools

full set of results @ Robroek et al. "An Analysis of Collocation on GPUs for Deep Learning Training"

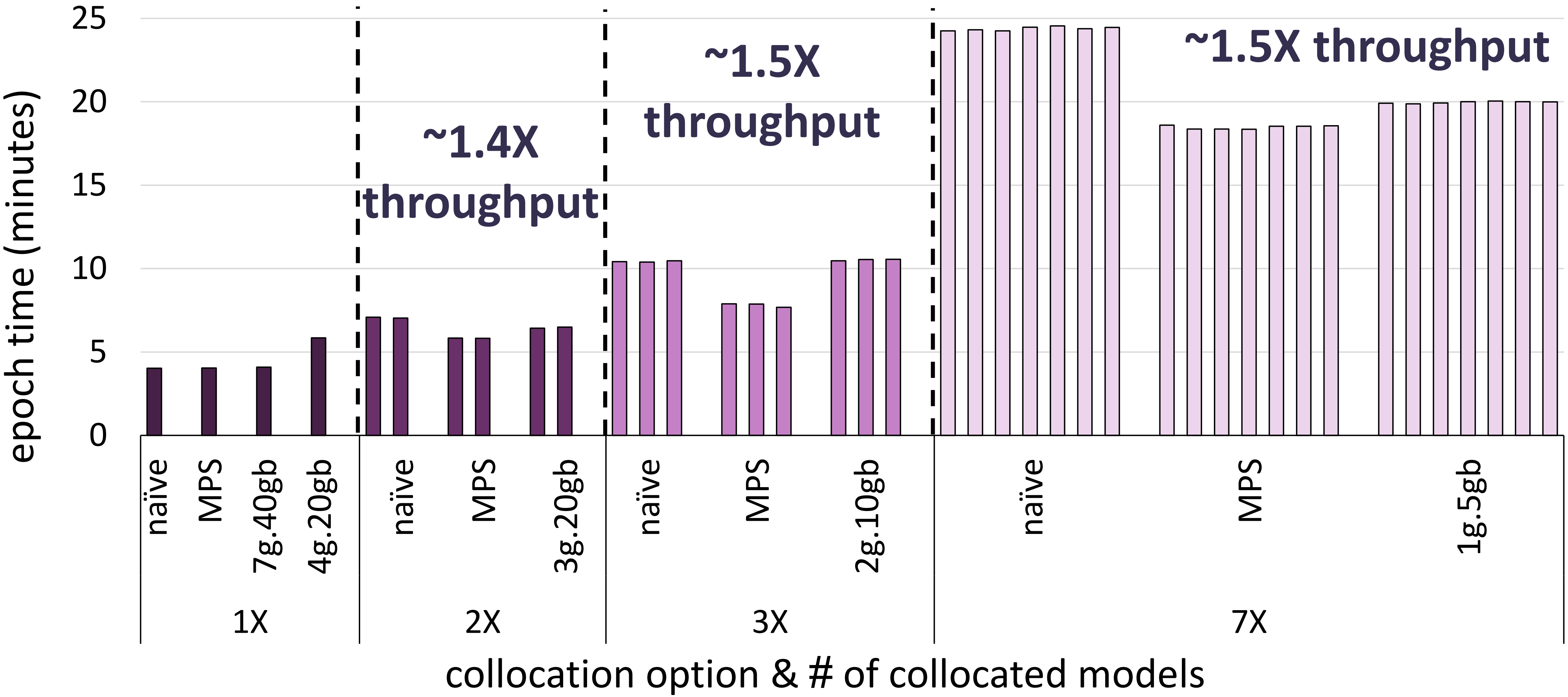
time per epoch – *small case – ResNet26*



collocation benefits despite increased epoch time

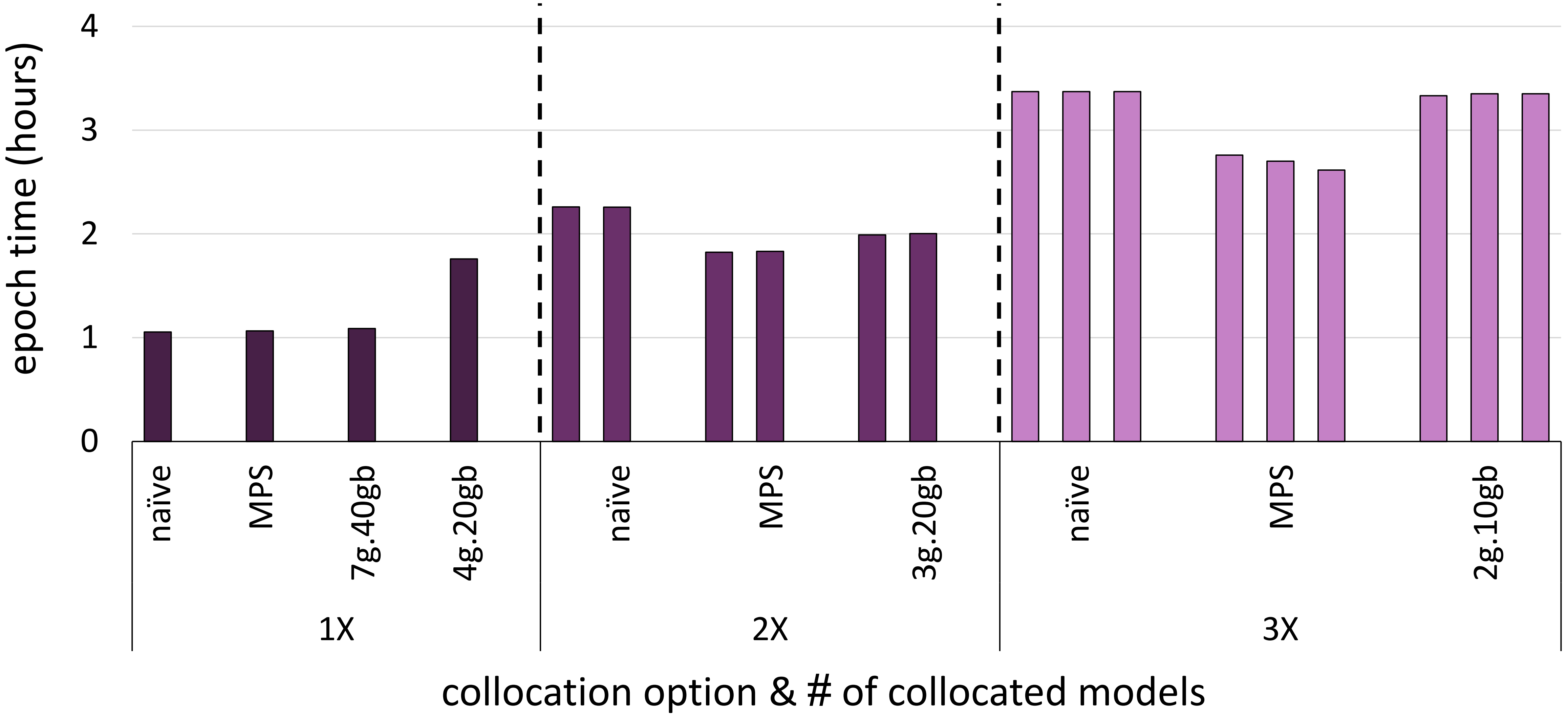
MPS > MIG > naïve

time per epoch – *medium case – ResNet50*



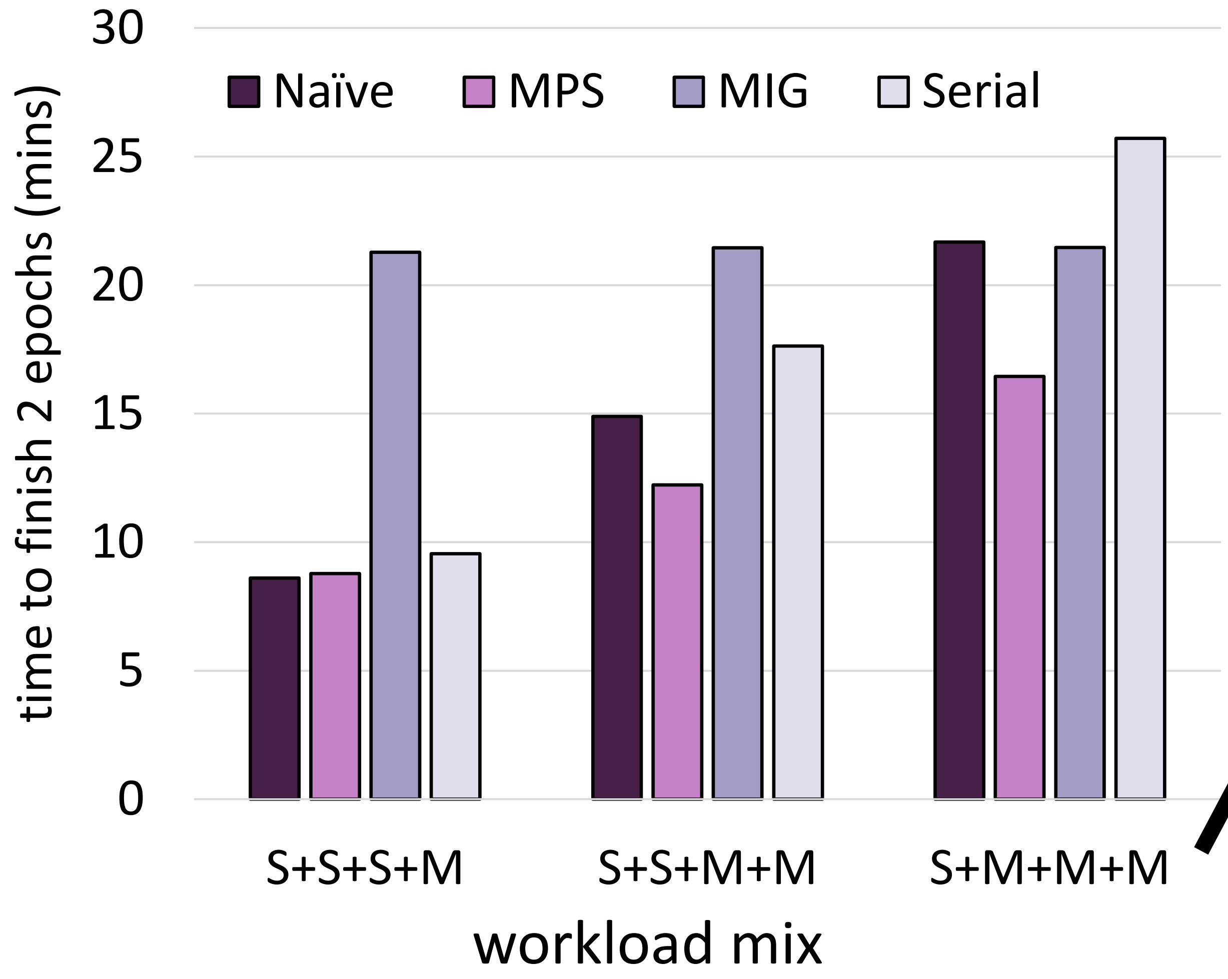
still some throughput benefits
but diminishing returns for increased collocation

time per epoch – *large case – ResNet152*

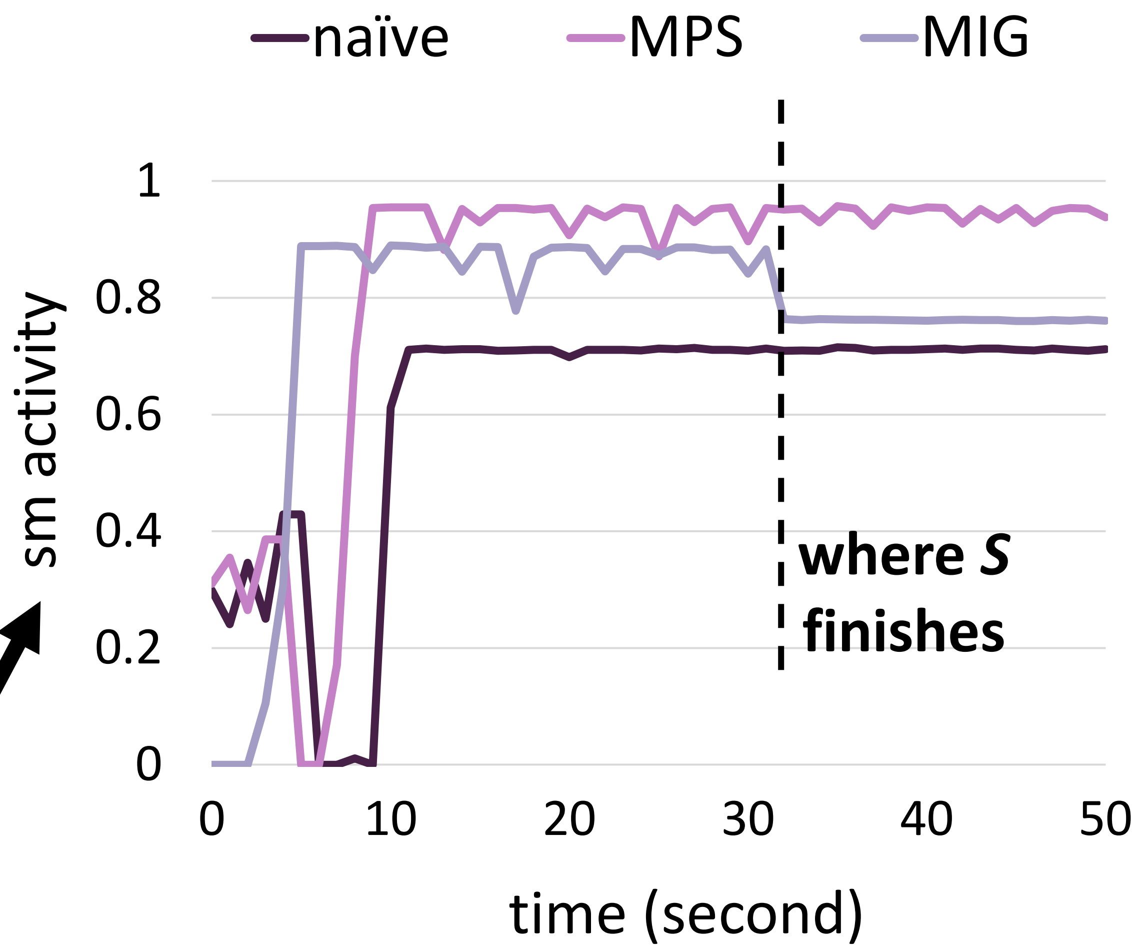


no more throughput benefits – 80% utilization when training alone
better to collocate with smaller or less compute heavy tasks

mixed workloads: all compute-heavy



(s=small – 1g.5GB, m=medium – 2g.10GB)



with MPS ➔ the small training is for free near the medium one

with MIG ➔ isolation at the cost of inflexible resource distribution

mixed workloads: compute- & memory-heavy

	DLRM – time per training block		ResNet152 – time per epoch		sm activity	memory footprint
DLRM alone	5.36 h		-		5%	29.14 GB
ResNet152 alone	-		1.05 h		82%	8.47 GB
naïve	6.09 h	(+14%)	1.11 h	(+5%)	81%	37.75 GB
MPS	5.57 h	(+5%)	1.10 h	(+4%)	81%	37.62 GB
MIG: **3compute – DLRM** **4compute – ResNet** **shared memory**	5.60 h	(+5%)	1.40 h	(+33%)	39%	37.86 GB

collocation can lead to (almost) free lunch
when workloads stress hardware different resources

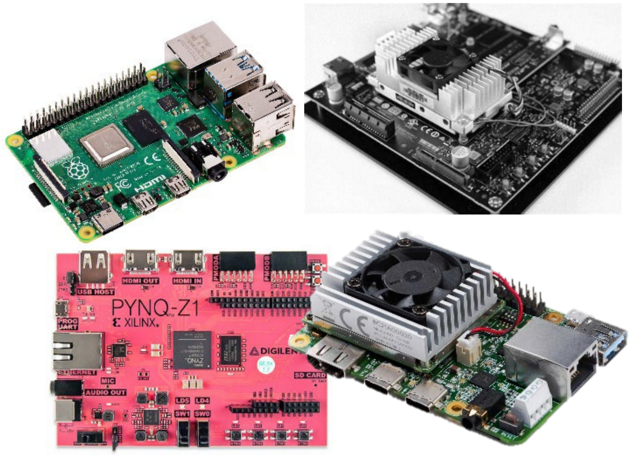
collocation for deep learning

- not all training needs all the resources of a single GPU
- collocation on GPUs benefits when the aggregate compute & memory needs of the collocated training runs fit in the GPU
- MPS performs better thanks to its flexibility
  - wasn't the case pre-PyTorch v2.0 (with CUDA 11.7)
- MIG is the only option if more strict separation is needed
  - if the workload resource needs known ahead of time, can be configured to achieve performance close to MPS

need to build schedulers that incorporate GPU collocation!

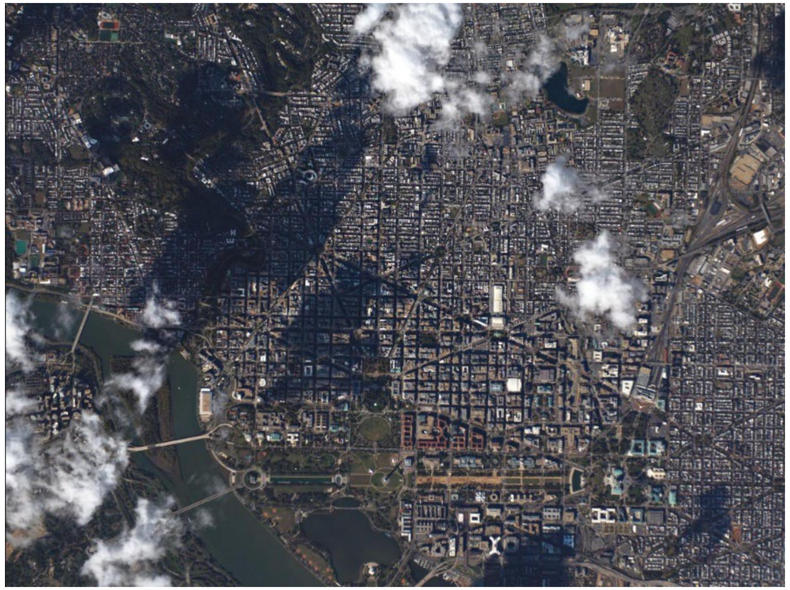
hardware scales for deep learning

large ⟷ **tiny**

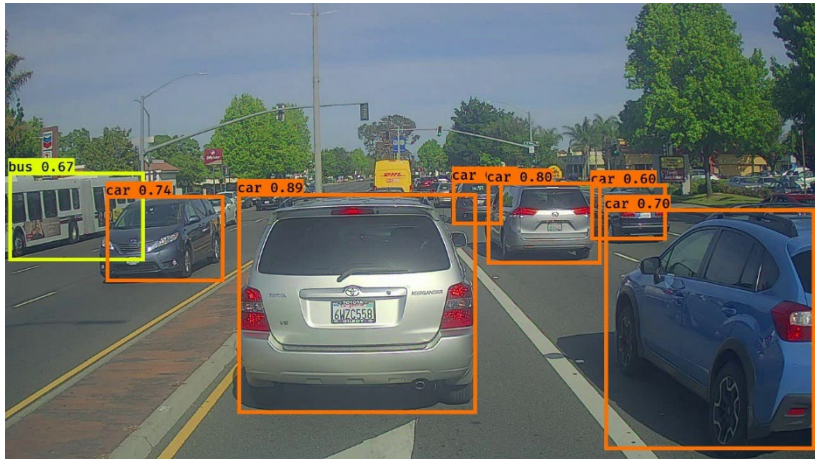


➔can we utilize these hardware well?
➔can we do more with less?

machine learning @ the edge



- low-latency & real-time applications
- poor / non-existing connectivity
- legal restrictions & privacy

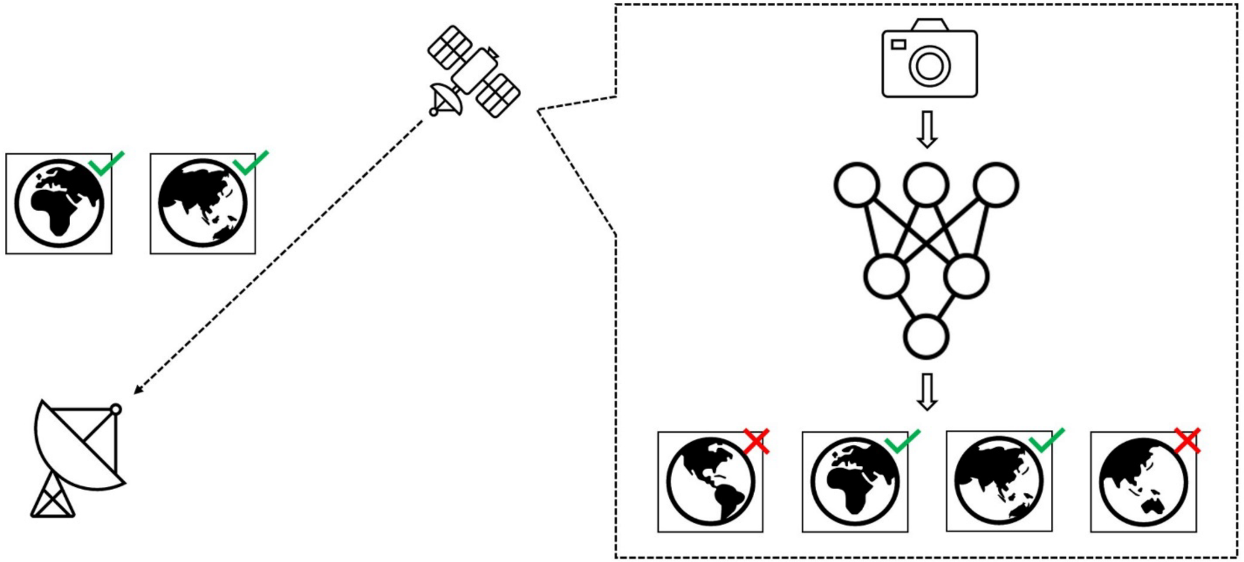


need for efficient & complex data processing closer to data sources!

DISCO: Danish student CubeSat program

https://discosat.dk/

- collaboration across Danish universities
- **use-case:** build a CubeSat satellite for observation of landmasses (especially snow, ice …) in the Arctic
- **goal:** ML-based image classification to send only the relevant images to ground (minimize data movement)



our task: build the image processing unit on the satellite

➔ which edge device can satisfy the *requirements* for this task?

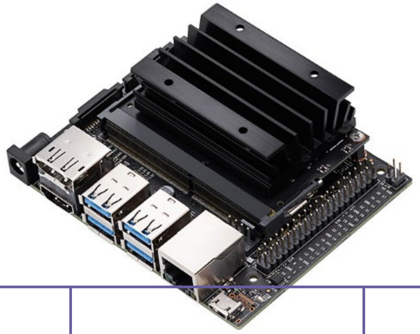
image processing unit requirements

quantitative requirements	
real-time imaging latency	4.42 seconds
peak power draw	5 Watts
mass	150 grams
dimensions	10x70x80 mm

calculations for the latency requirement & full set of results @ Bayer et al. "Reaching the Edge of the Edge: Image Analysis in Space"

qualitative requirements
➔ students should be able to upload / update their code for image analysis on the satellite over time
➔ short time-to-booth
➔ robustness against crashes and failures

devices



	ARM Cortex-M7	**Jetson Nano**	**CoralAI Micro**	**CoralAI Mini**	**CoralAI USB Stick**	**Neural Compute Stick 2**
CPU	ARM Cortex-M7 @300MHz	ARM A57 @1.43GHz	ARM Cortex-M7 @800MHz, ARM Cortex-M4 @400MHz	ARM Cortex-A35 @1.5GHz	Raspberry Pi 3 BCM2837 ARM @1.2GHz	
RAM	384KB SRAM, 32KB FRAM	4GB	64MB	2GB	1GB	
Accelerator	none	128-core Maxwell GPU	CoralAI Edge TPU (4 TOPS)			Intel Movidius Myriad X VPU

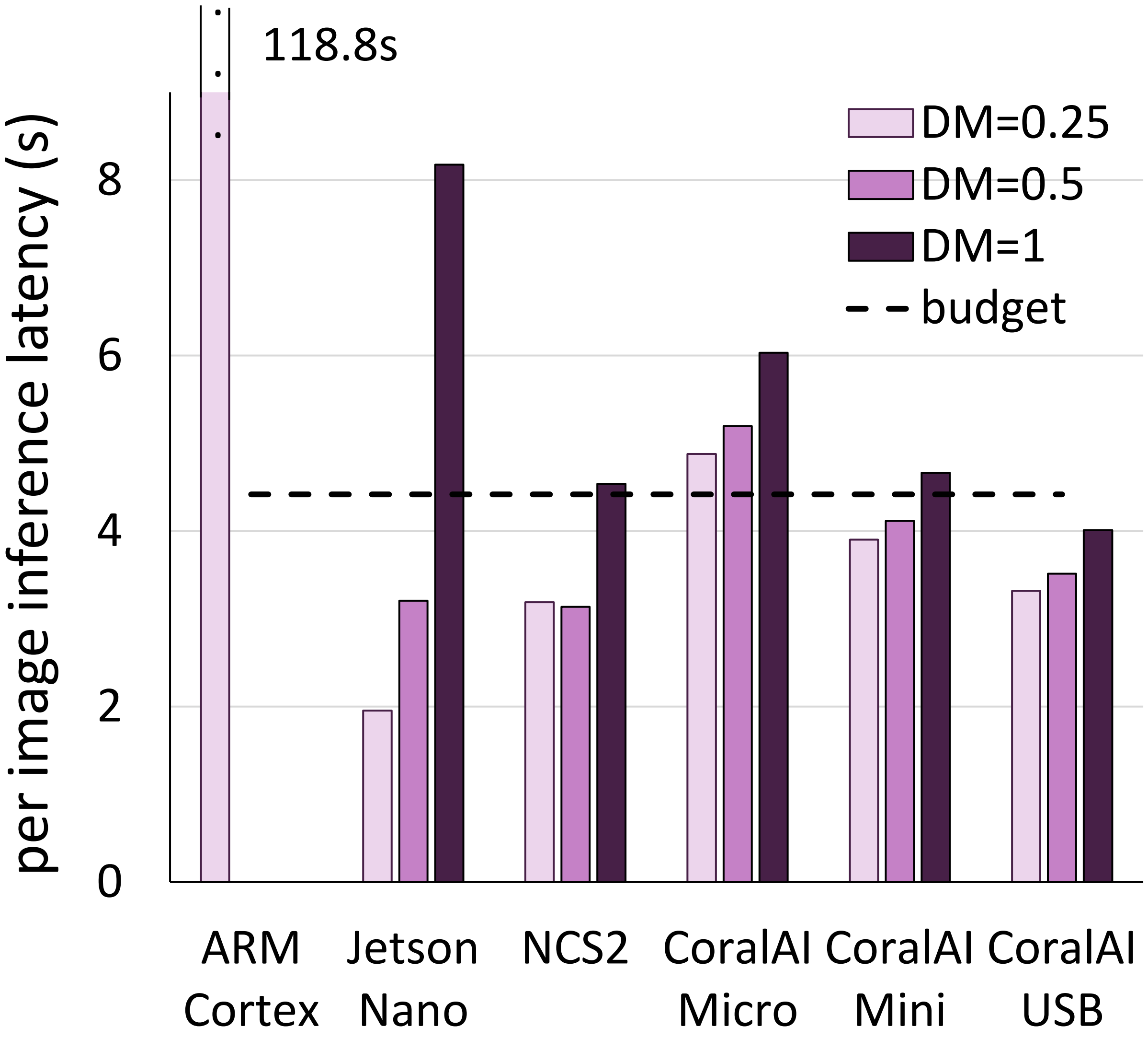
general-purpose ⟷ **higher specialization**

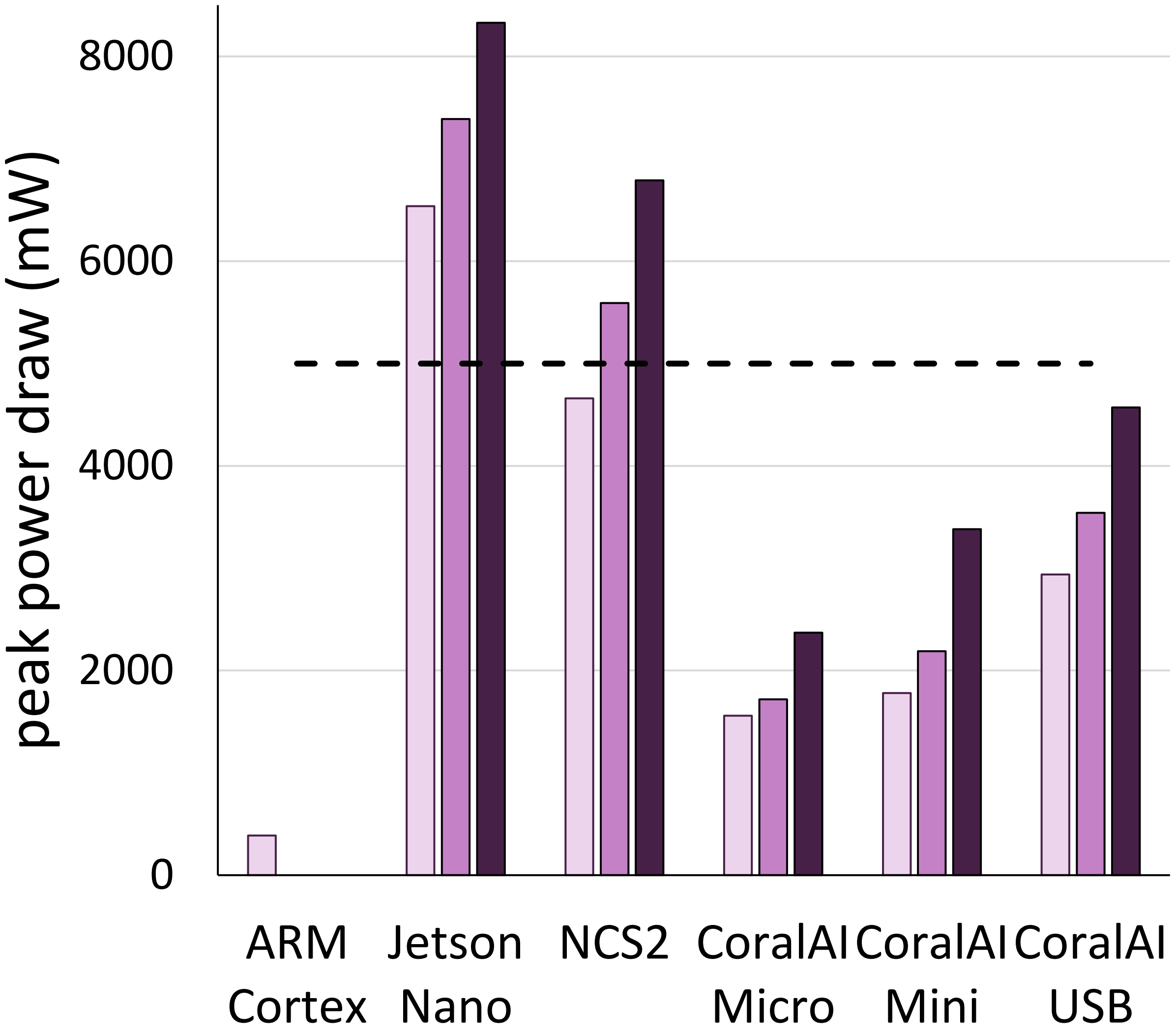
setup

- **model:** MobileNet v1 ➔ small model size, can be scaled down further (DM = 1, 0.5, 0.25)
- **dataset:** Flowers* ➔ fits the input size of the model (224x224)
  ➔ number of classes to classify fit our use case
- model trained on ImageNet, fine tuned on Flowers dataset
- **input:** 4512 x 4512 images from the satellite camera → 400 224x224 image patches
- **per-image inference latency =** time to infer 400 patches
  - helps with optimizations per-inference & sending only the relevant part of an image

	ARM Cortex-M7	Jetson Nano	CoralAI *	Neural Compute Stick 2
framework	TensorFlow Lite for Microcontrollers	TensorRT	TensorFlow Lite	OpenVino
quantization	8bit (to fit the device memory)	16bit	8bit (only supports 8bit ints)	16bit (only supports 16bit floats)
batching	not enough memory to do batching	batch size per inference = 16	doesn't support batching	number of concurrent inference requests = 4

latency & power draw

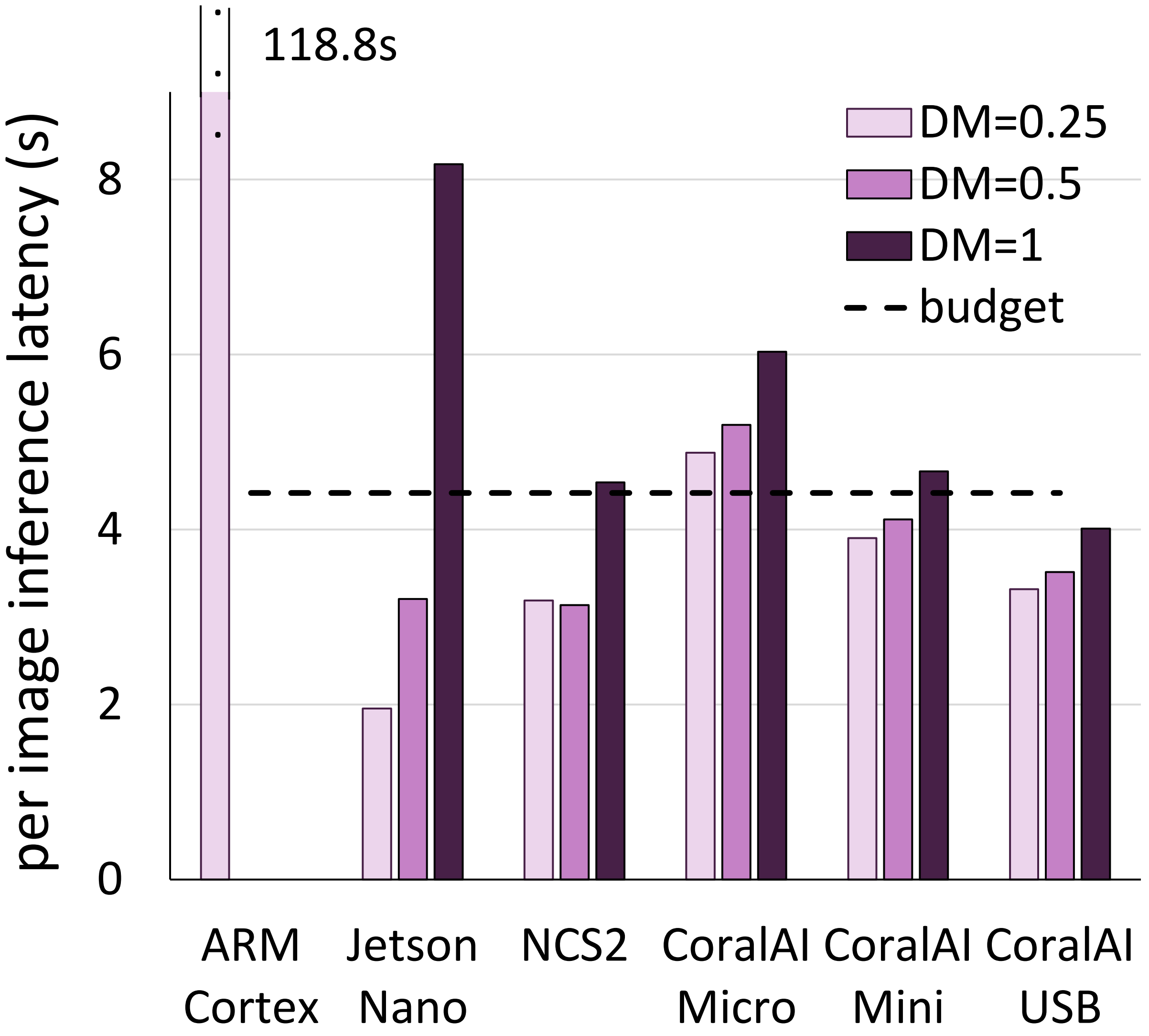


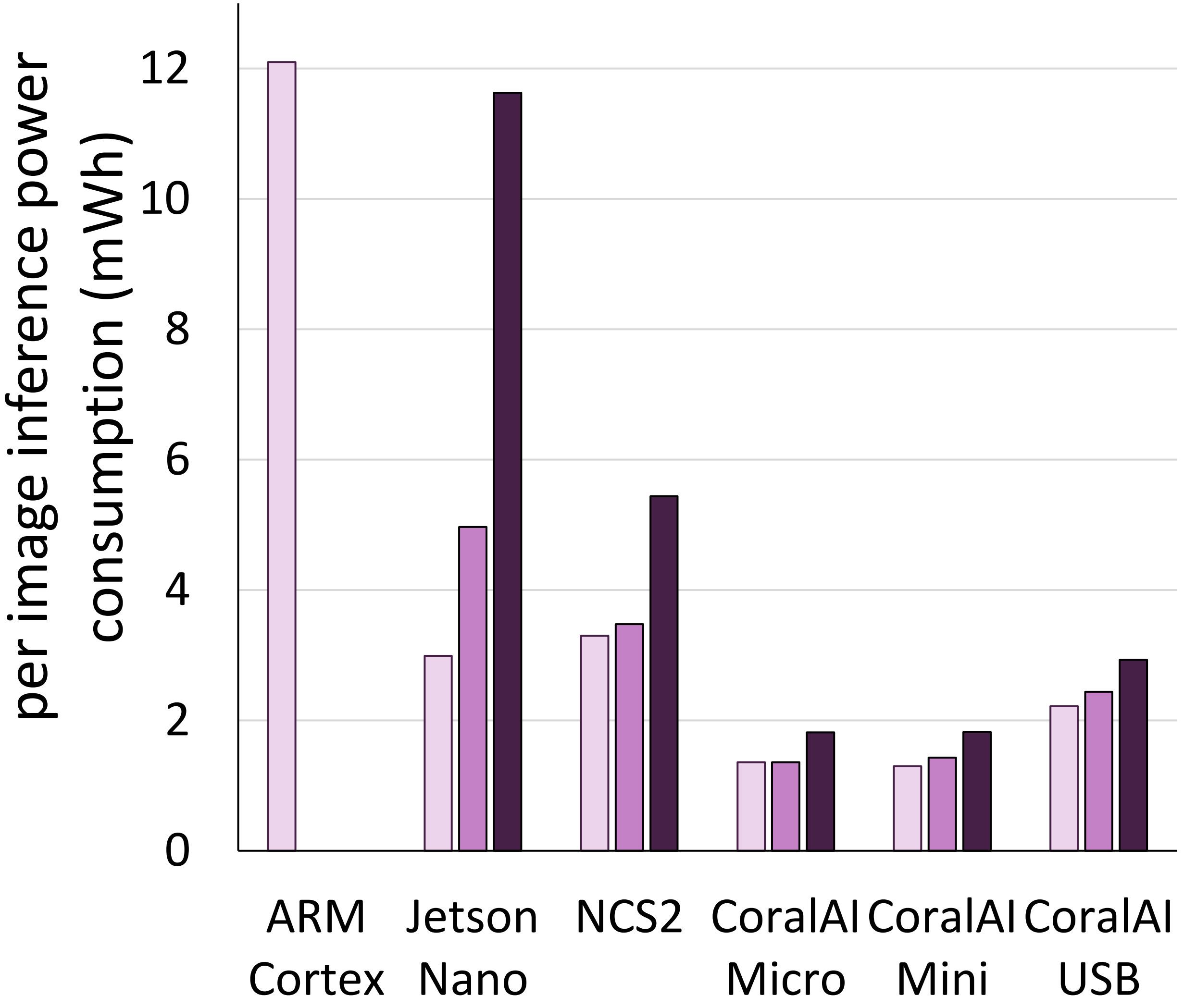


Jetson Nano & NCS2 are faster for smaller models, Coral scales better.

Jetson Nano & NCS2 fail the peak power budget.

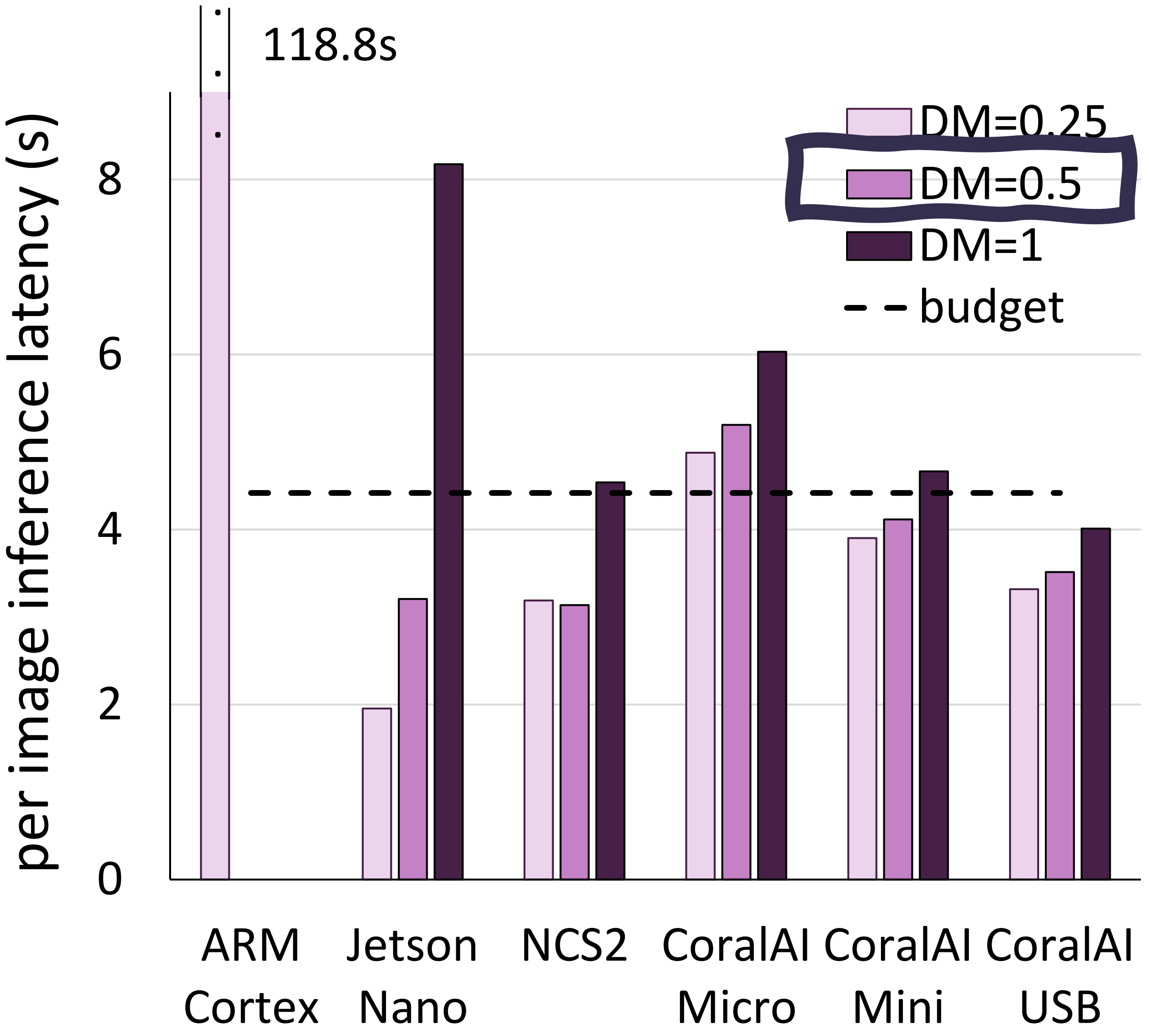
latency & power draw

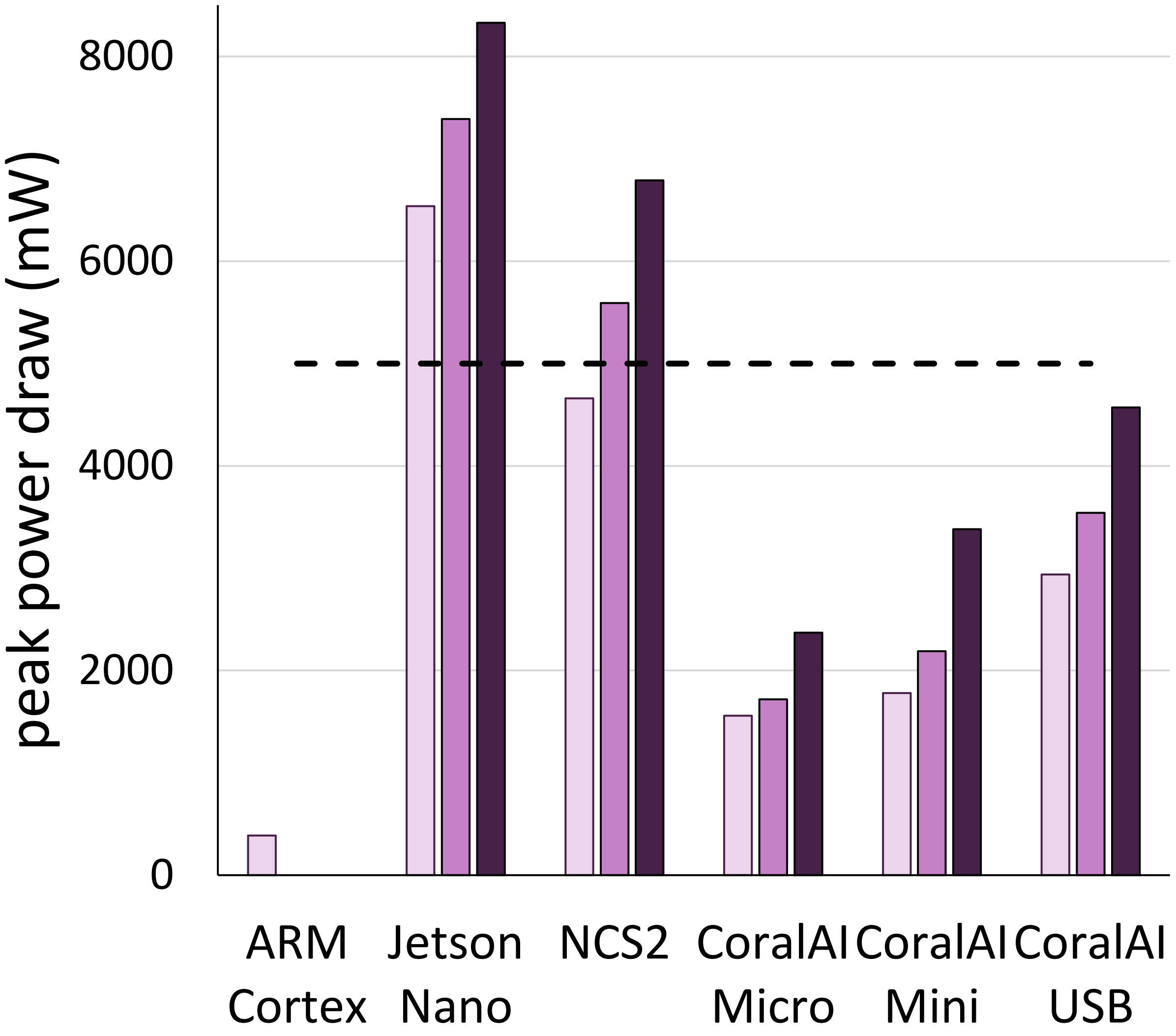




ARM-based microcontroller draws little power per unit time but per inference power need is higher than the rest!

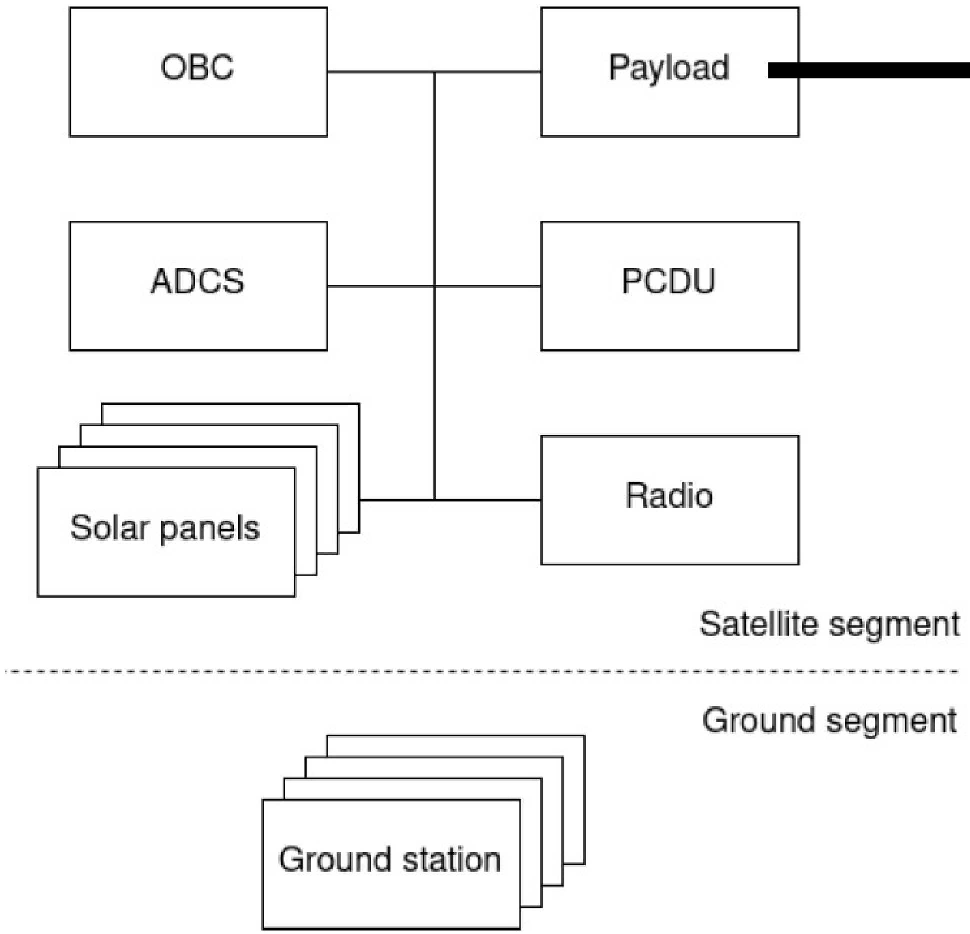
latency & power draw

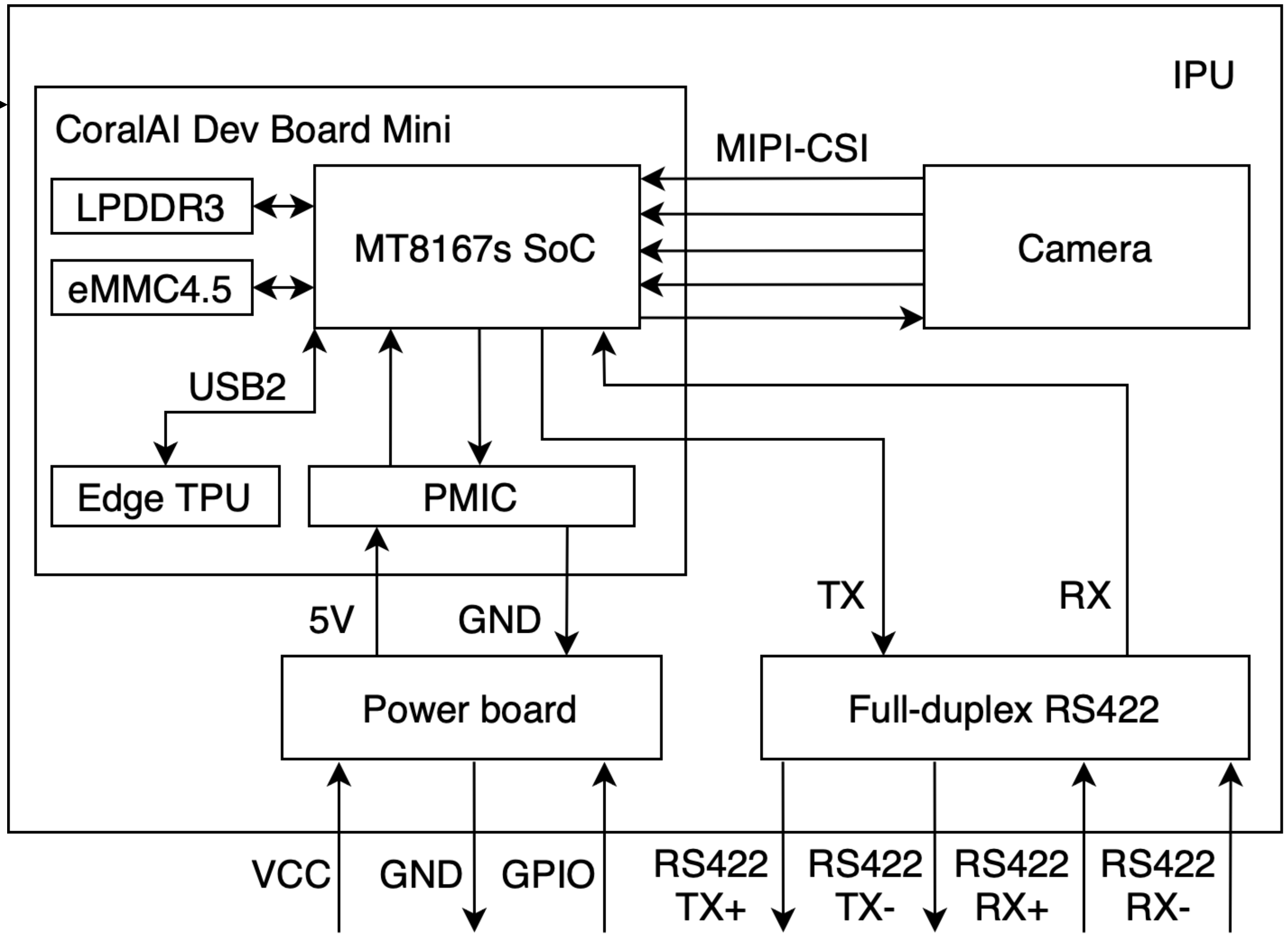




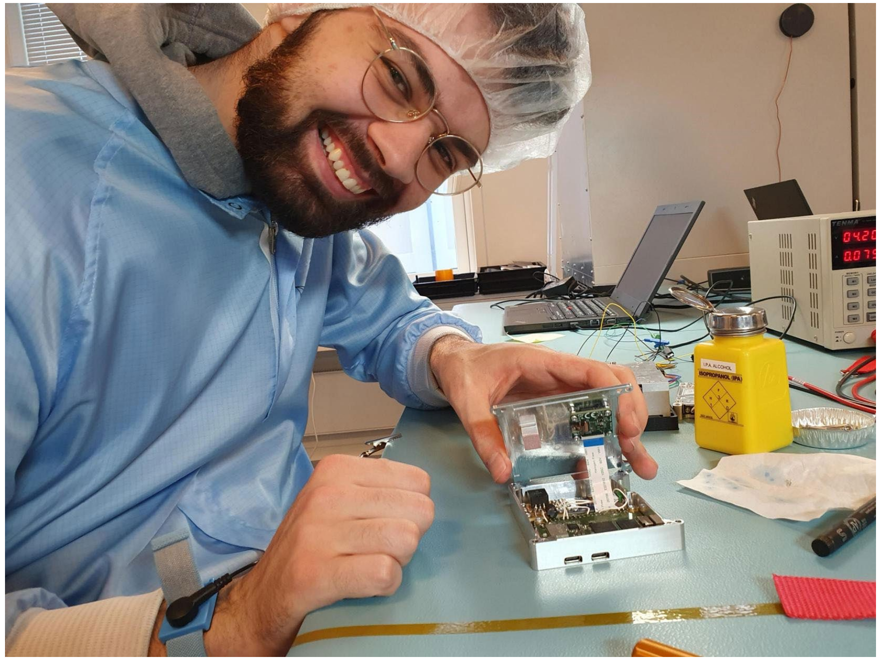
CoralAI Mini & USB satisfy both latency and power budget.

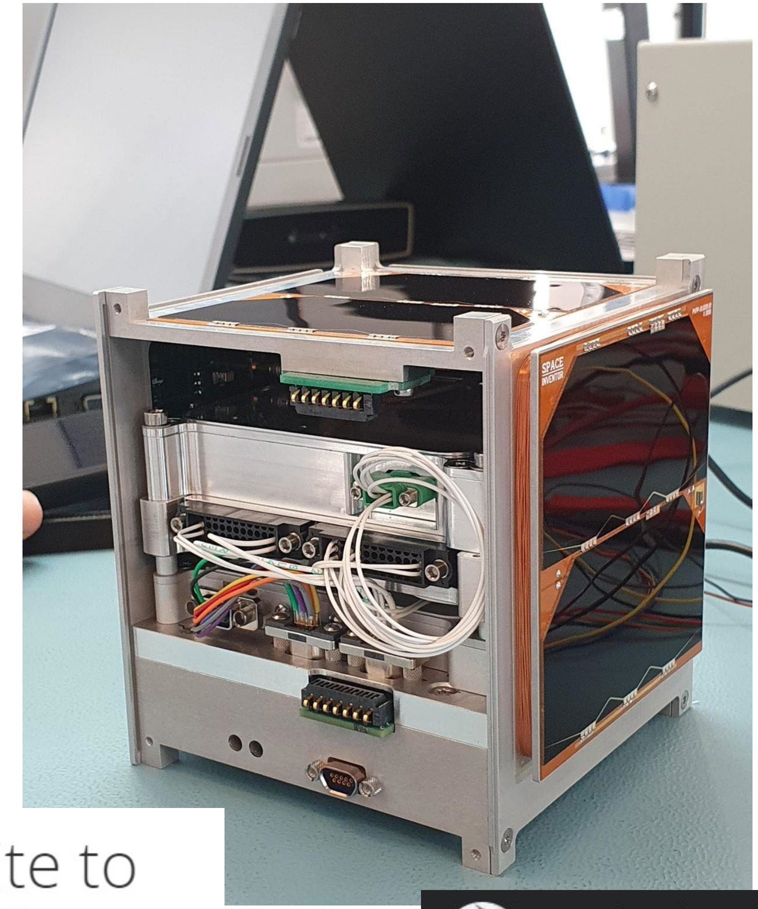
image processing unit = IPU





DISCO satellite





Students launch a satellite to test artificial intelligence in space

On April 14, students from ITU will contribute to writing space history. The satellite, DISCO-1, is launched into space and it carries a microcomputer to test artificial intelligence outside the atmosphere. The satellite is developed by the space program, DISCO, which is a collaboration between students from four Danish universities.

IT-Universitetet i København
April 15 ·

Så lykkedes det! 🚀🛰️✨

Satellitten DISCO-1, udviklet af danske studerende fra bl.a. ITU, blev her til morgen sendt ud i rummet med SpaceX' raket fra Californien.

Satellitten indeholder en mikrocomputer, der skal teste kunstig intelligens i rummet. 🤗

Læs mere om projektet her 👉 https://www.itu.dk/.../Studerende-opsender-satellit-der...

📷 Julian Priest (CC BY-NC 3.0)

ML @ the edge

- demand for more data analysis closer to the data source
  - reduces data movement & privacy concerns
  - helps with real-time decisions
- variety of edge devices to choose from offering increasingly powerful hardware but still resource-constrained
  - requires not just latency-efficient,
    but also energy-efficient data processing
- hardware specialization helps with latency & power budget
  - though, we need more flexibility

need for methods that can deal with resource management & program updates at the edge!

team*RAD* - resource-aware data systems

rad.itu.dk

phd students

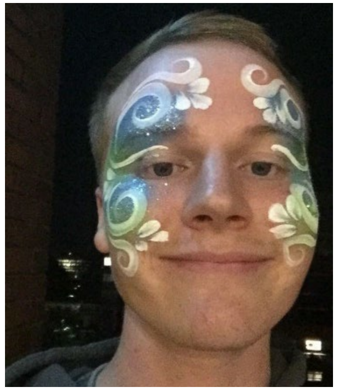
Ties Robroek

Ehsan Yousefzadeh-Asl-Miandoab

Robert Bayer

research assistants

Neil Kim Nielsen



Joachim Moe Osterhammel

collaborators & helpers

Julian Priest

Lottie Greenwood

Sebastian Büttrich

DASYA

Data-Intensive Systems and Applications

www.dasya.dk

@dasyaITU











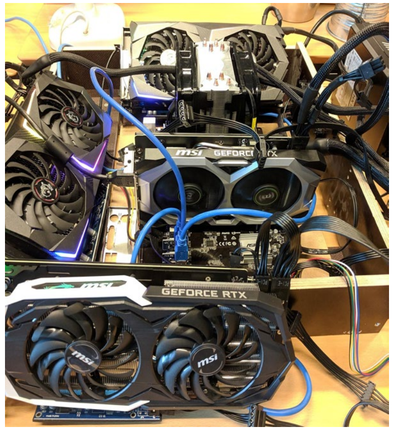


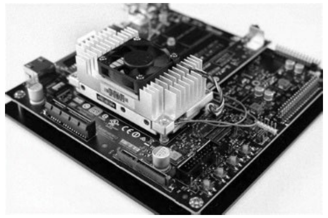
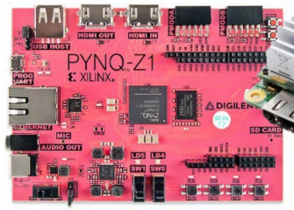


hardware scales for deep learning

thank you!

large ⟷ **tiny**

➔can we utilize these hardware well? ➔ not always

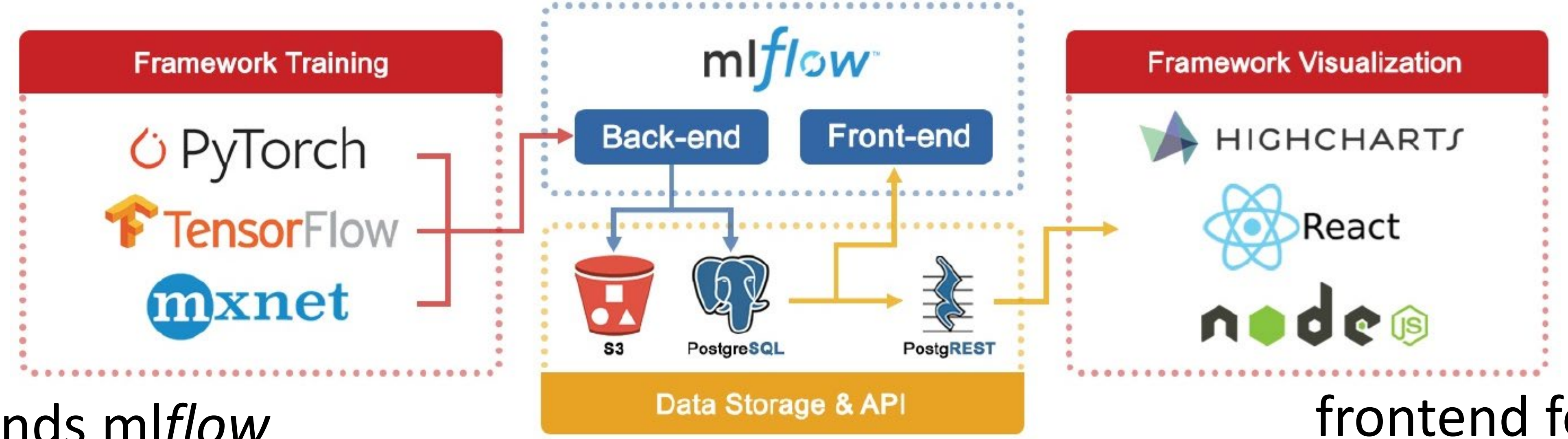
- need more effective workload collocation on accelerators
- energy-efficiency must be part of the utilization analysis

➔can we do more with less? ➔ yes, but it isn't free lunch

- need to understand better the capabilities of different devices
- every scale requires its own dynamic resource managers

backup

radT



- extends ml*flow*
- incorporates collocation
- allows easy, extensible, and scalable tracking of hardware metrics on CPUs & GPUs
  - listeners for monitoring (dcgm, nvidia-smi, top) & profiling (nsys, ncu, pytorch profiler) tools

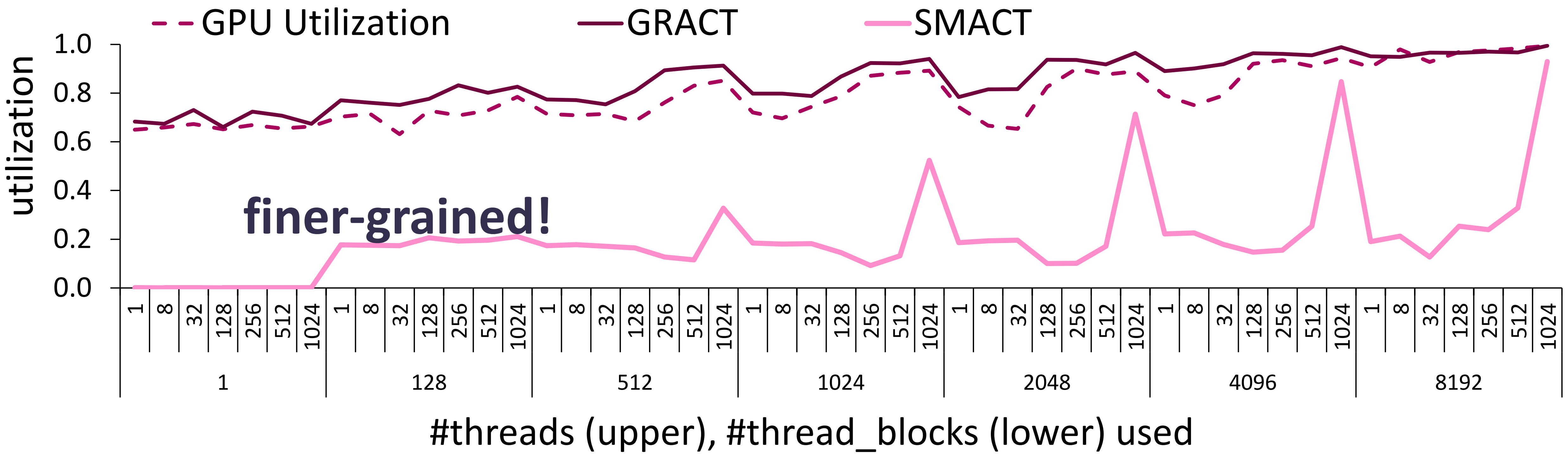
frontend for data exploration

used by several members of our group including data scientists for systematic benchmarking of deep learning training

Robroek et al. "[Data Management and Visualization for Benchmarking Deep Learning Training Systems](#)", DEEM 2023
https://github.com/Resource-Aware-Data-systems-RAD/radt & https://www.youtube.com/watch?v=oaGfzYjKJ1Q

GPU utilization

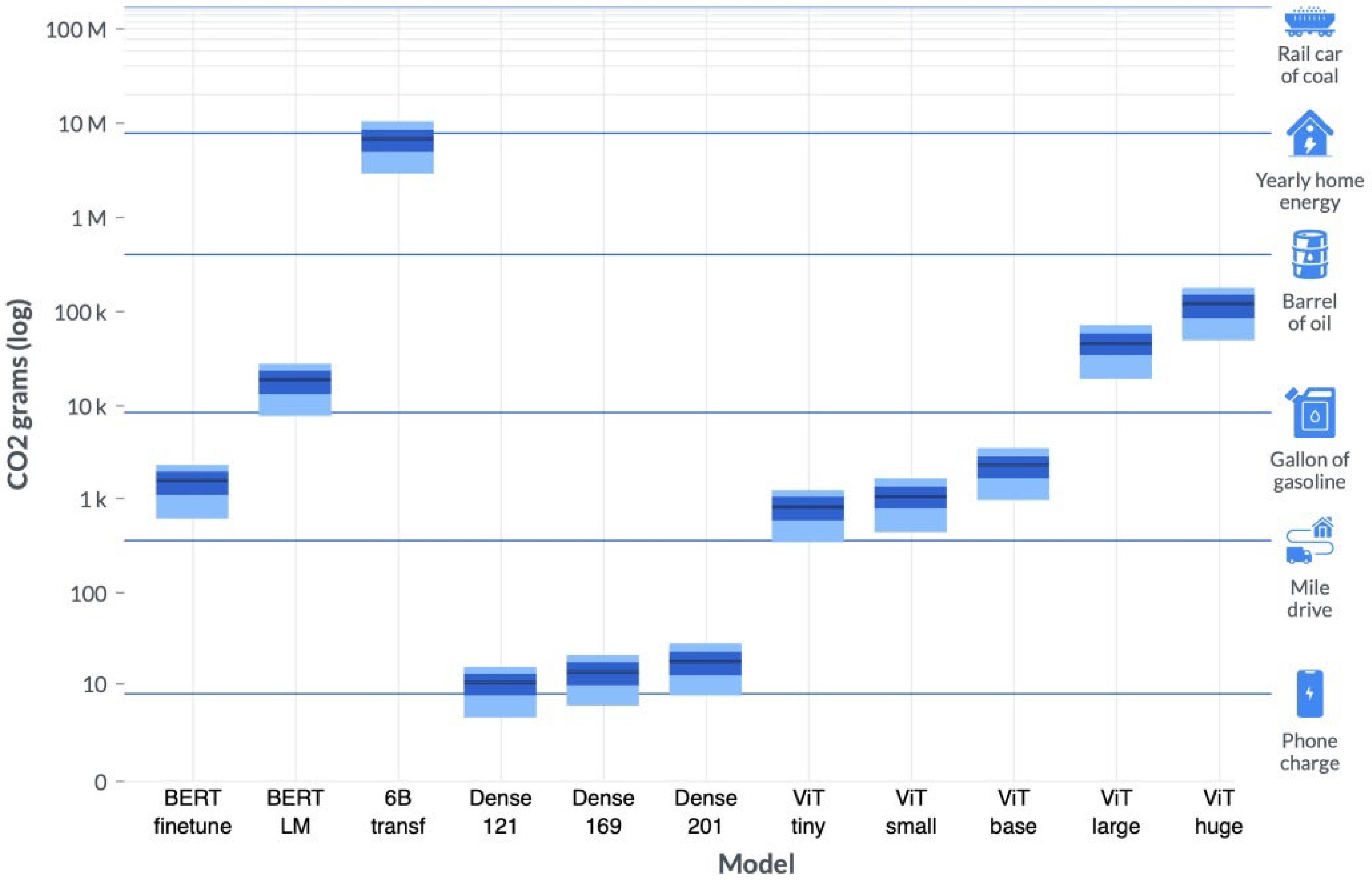
- **GPU utilization**: % of time one or more kernels were executing on the GPU
- **GRACT**: % of time any portion of the graphics or compute engines were active
- **SMACT:** the fraction of active time on an SM, averaged over all SMs = streaming multiprocessor



coarse-grained GPU utilization metrics could be misleading!

Yousefzadeh-Asl-Miandoab et al. "[Profiling and Monitoring Deep Learning Training Tasks]", EuroMLSys 2023

unsustainable growth of deep learning



Dodge et al. "[Measuring the Carbon Intensity of AI in Cloud Instances](#)." FAccT 2022

DISCO use-cases

	real-time imaging	Arctic region imaging	Greenland imaging
cycle duration	4.42 seconds	95.65 minutes	1 day
# images to infer per cycle	1	80	320
budget for inference per image	4.42 seconds	71.74 seconds	270 seconds

- **higher latency budget**
- **more pressure on the memory / storage resources**
  need to buffer more images

accuracy

on Flowers dataset, with post-training quantization

		MobileNet DM = depth multiplier		
		0.25	**0.5**	**1**
accuracy	**32bit float**	86.92%	90.33%	90.74%
	16bit float	86.78%	90.33%	90.74%
	8bit integer	84.33%	89.78%	91.55%
#params		219,829	832,101	3,233,989

accuracy trade-off becomes noticeable

too big & complex for most resource-constrained devices