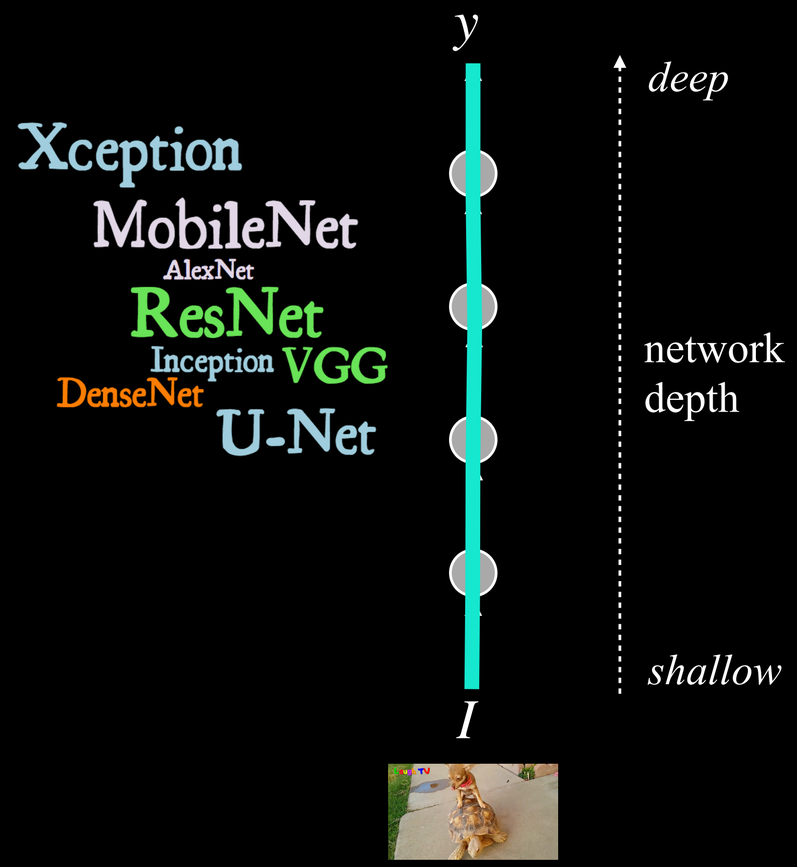
Massively parallel video networks

Viorica Pătrăucean
viorica@google.com

joint work with João Carreira, Laurent Mazare
Andrew Zisserman, Simon Osindero

Processing of visual input

- Vision: dominant sense for humans
- Most works focus on image processing
  - videos: image models applied frame-by-frame



Benefits from working with videos

➡ better accuracy (temporal smoothness; reduced ambiguity)

- action recognition, optical flow estimation etc.

➡ **more efficient**

- e.g. MotionJPEG vs. MPEG

Figure 1. **A still from 'Quo Vadis' (1951).** Where is this going?

Quo vadis, action recognition, Carreira, Zisserman. CVPR2017

DeepMind

Autonomous cars

The Duel: Timo Boll vs. KUKA Robot

Robotics applications

DeepMind

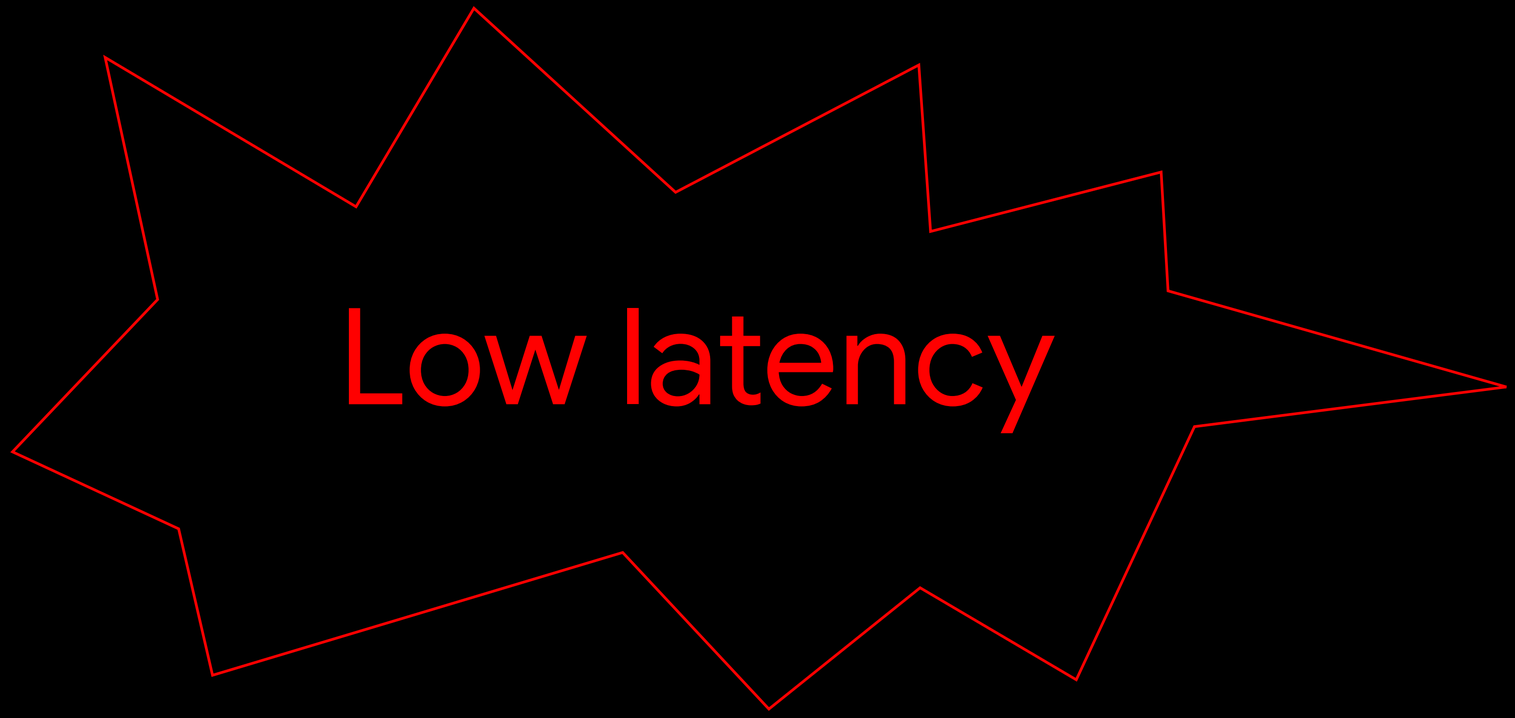
Minority report, Steven Spielberg (2002)

Augmented reality

DeepMind

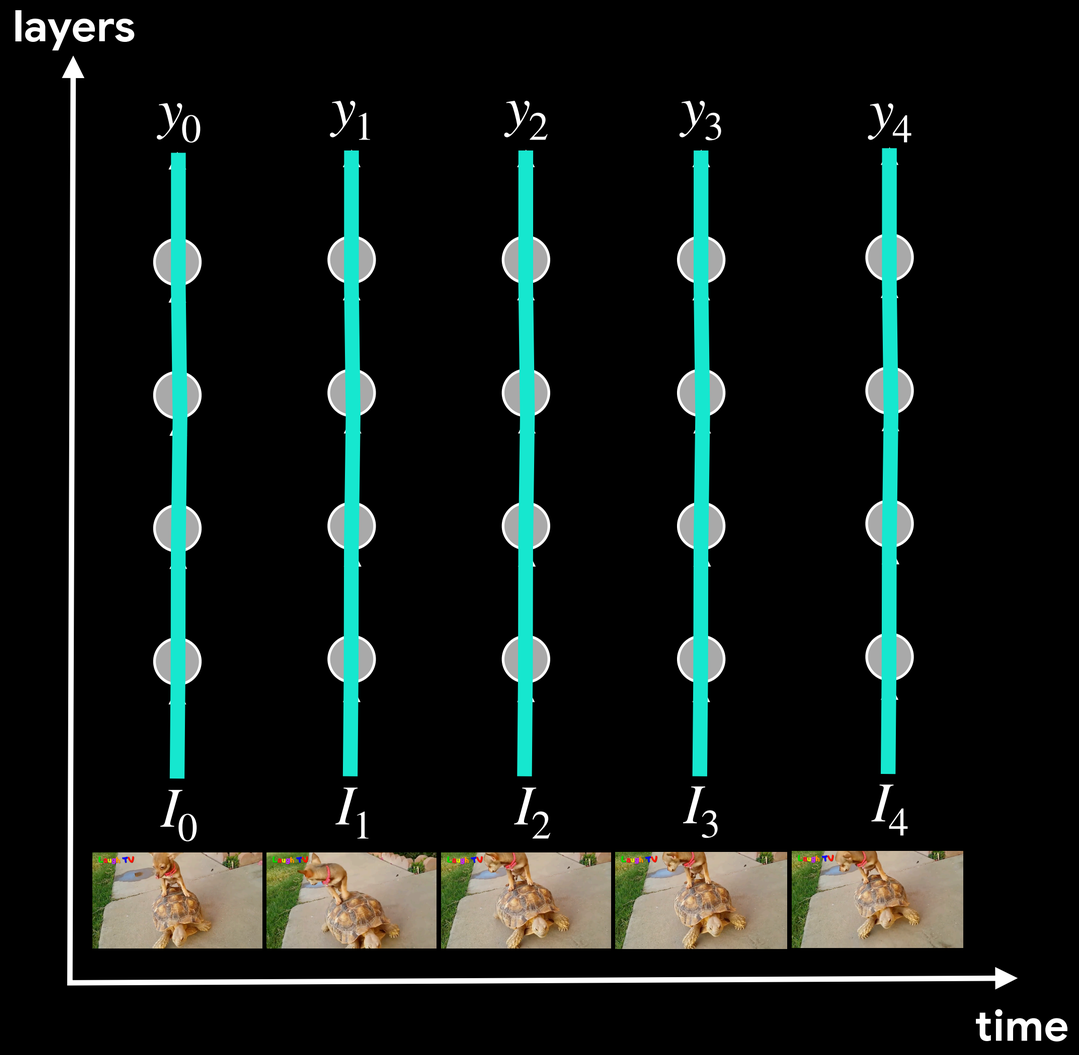
Setting

- online processing, one video stream, can’t batch
- causal, no peeking into the future



DeepMind

Deep video models = image models applied frame-by-frame



sequential in depth and time

latency = $\sum_d$ execution_time_layer$_d$

throughput (output rate) = 1 / latency

depth (accuracy) ↗ ⟶ latency ↗

DeepMind

Current object detectors: ~5fps

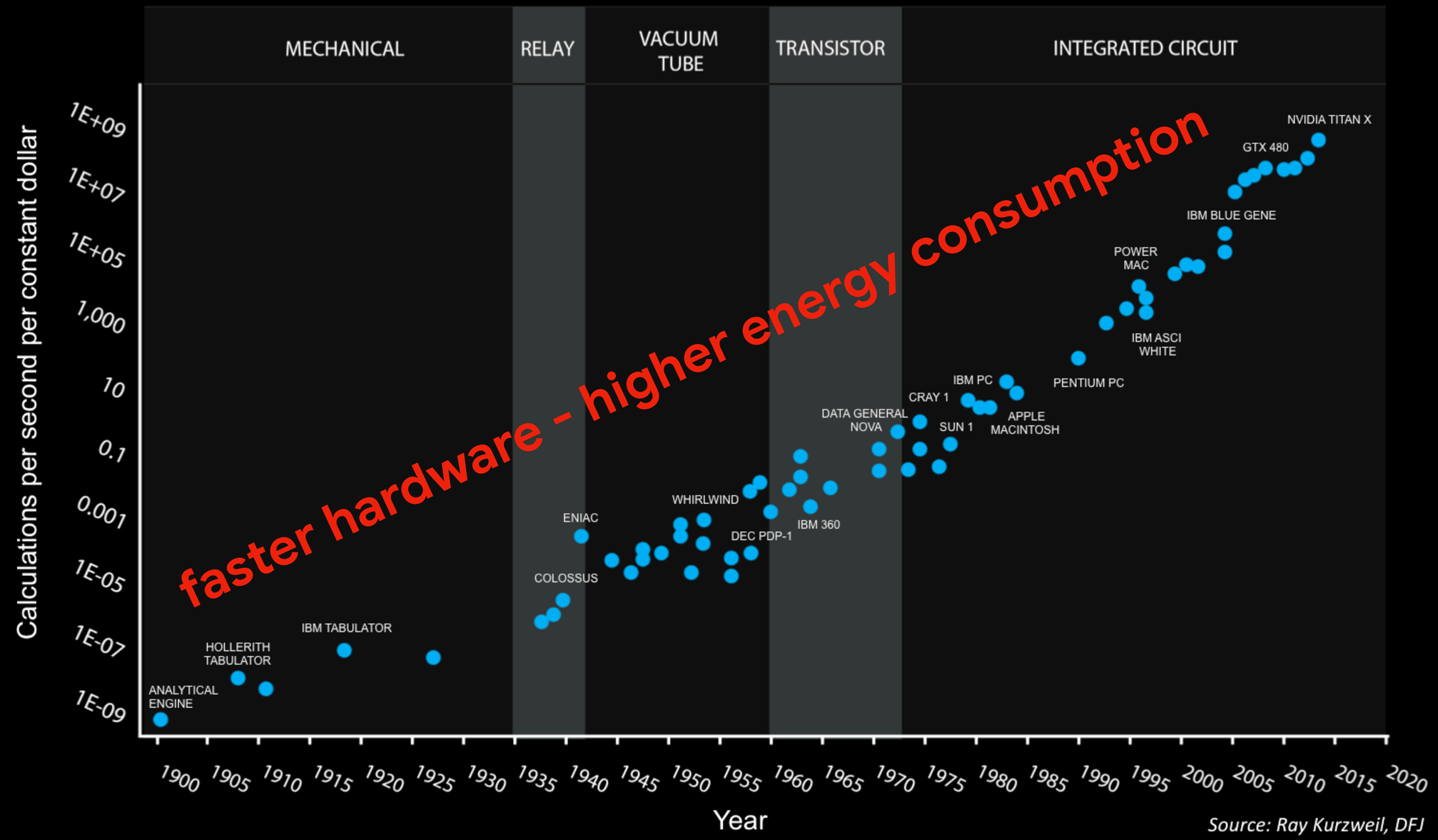
Davis2017 dataset

25 fps

5 fps

Safe autonomous driving at 5fps

120 Years of Moore's Law



Source: Ray Kurzweil, DFJ

More sustainable ideas

Improve efficiency of image models

- model compression (Chen et al.), distillation (Hinton et al.), low representation format (Courbarieux et al), lighter convolutional architectures (Mobilenet, Xception)
- time budget methods: exit when time runs out (Karayev et al., Mathe et al.)

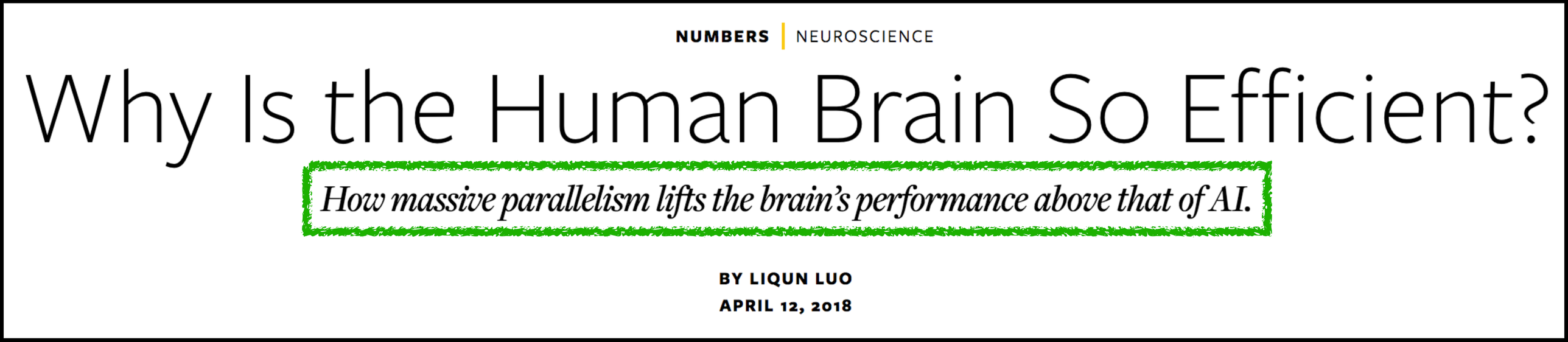
Improve efficiency of video models

- strided 3D conv (Tran et al., Carreira et al.), warping (Zhu et al.)
- different update rates for different features (Shelhamer et al.)

Our approach: orthogonal to these works

DeepMind

Is sequential mode optimal?



NUMBERS | NEUROSCIENCE

Why Is the Human Brain So Efficient?

How massive parallelism lifts the brain's performance above that of AI.

BY LIQUN LUO

APRIL 12, 2018

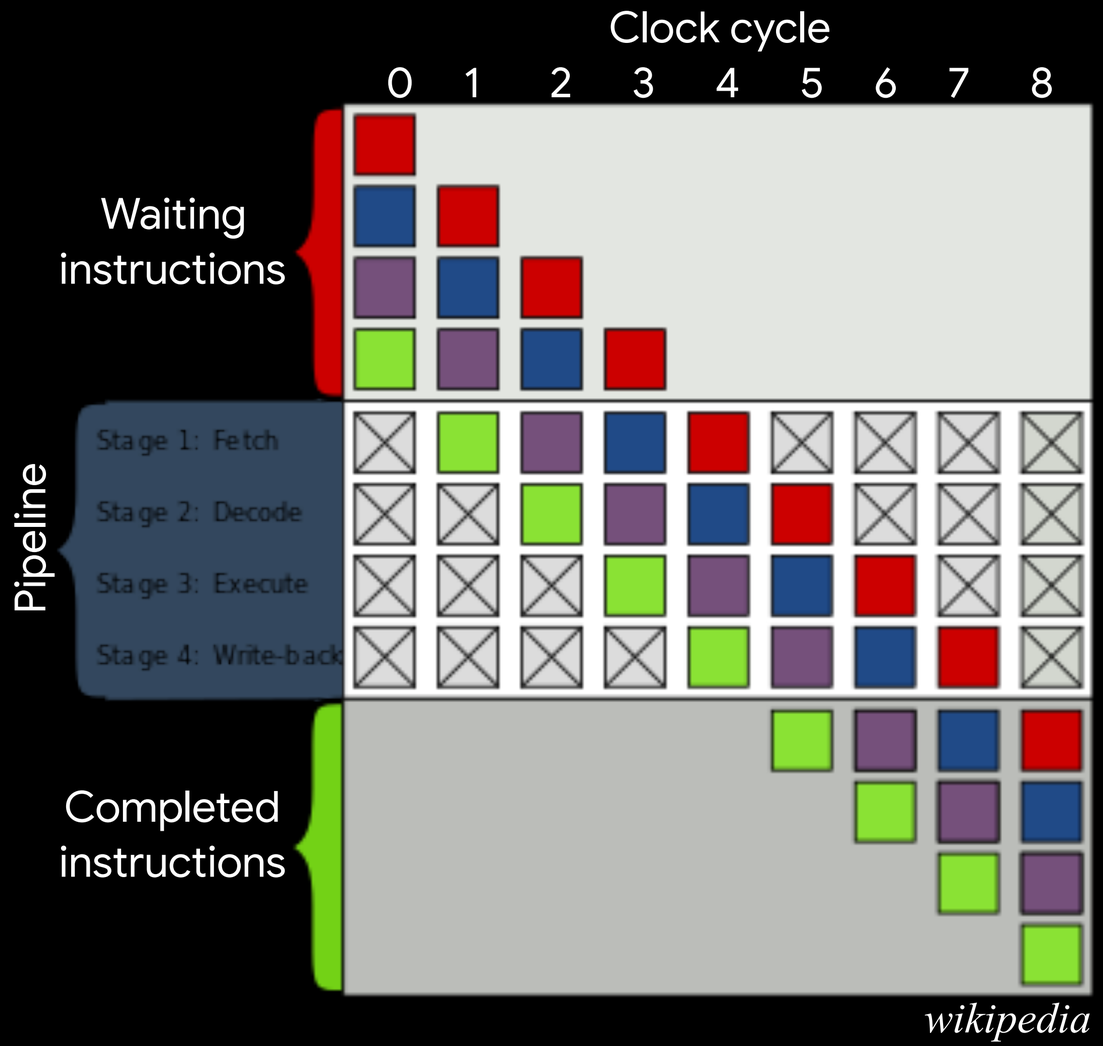
DeepMind

Is sequential mode optimal?

General-purpose processors use **pipelined** instructions enabling **parallel processing**.

Maximise throughput

Efficient use of hw resources



wikipedia

Predictive coding: parallelism

DeepMind

Increase throughput

Reduce latency

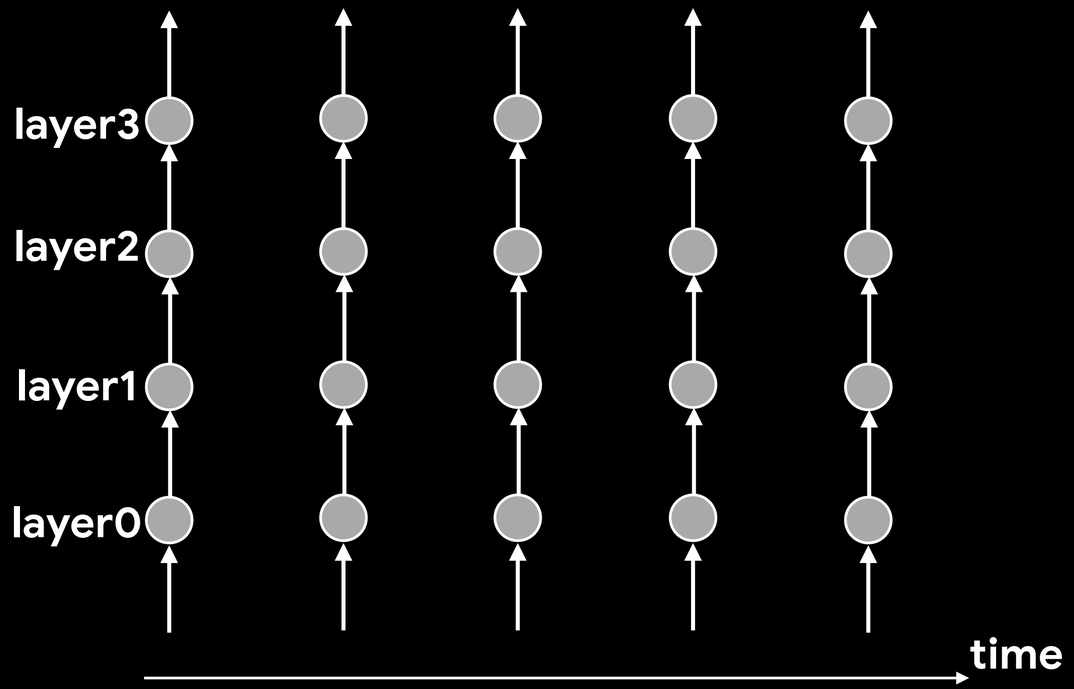
Reduce clock cycles

Increase throughput - option 1

Reduce latency

Reduce clock cycles

Setting



Network unrolled over time

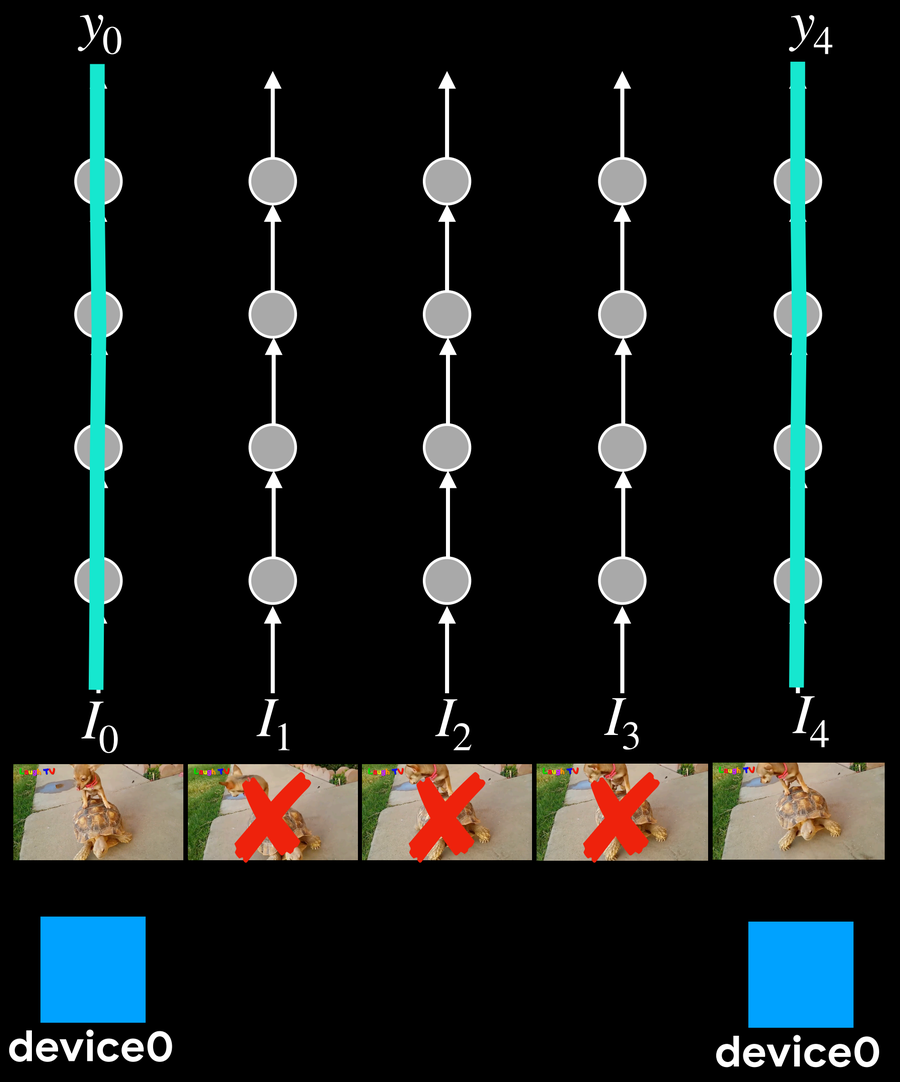
device0

hardware resources

DeepMind

Image model frame-by-frame

non real-time model



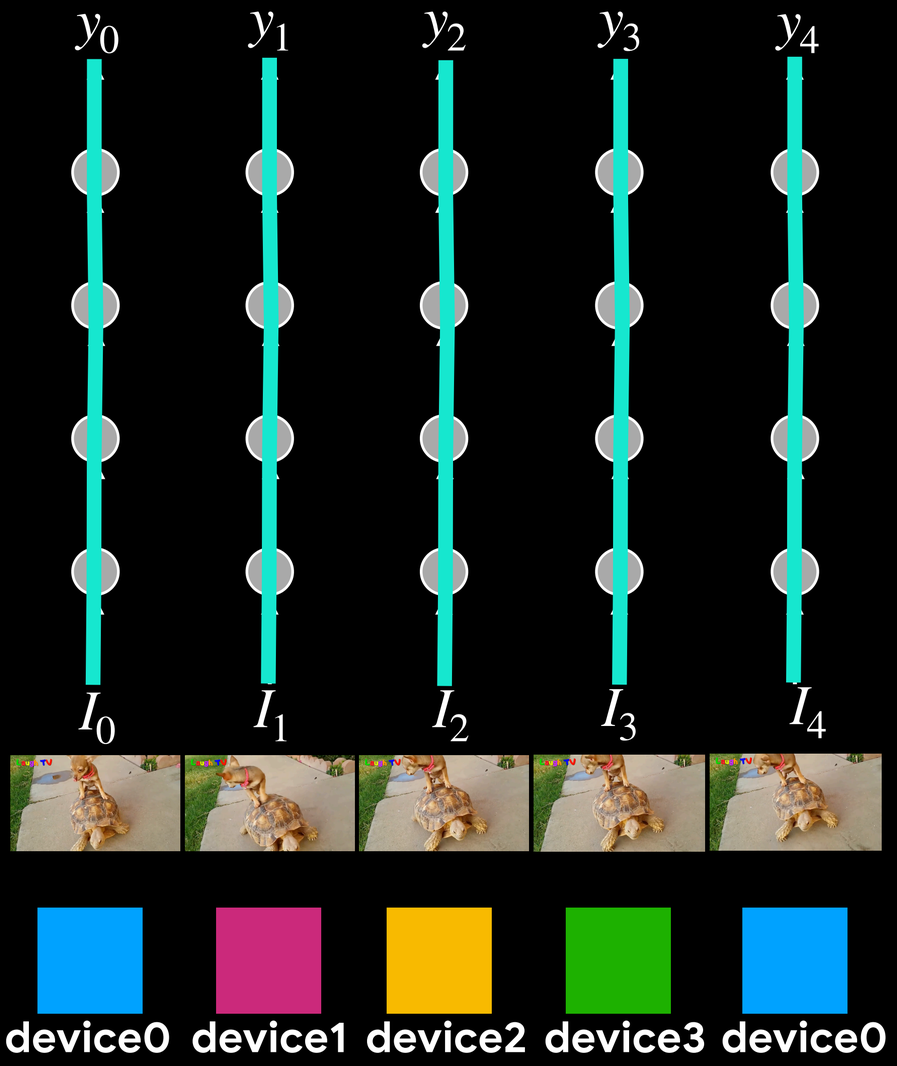
Con:
low throughput
high latency

DeepMind

Image model frame-by-frame

non real-time model

parallel resources



Pro:
high throughput

Con:
same (high) latency

round-robin over frames

DeepMind

Increase throughput

Reduce latency

} **our approach**

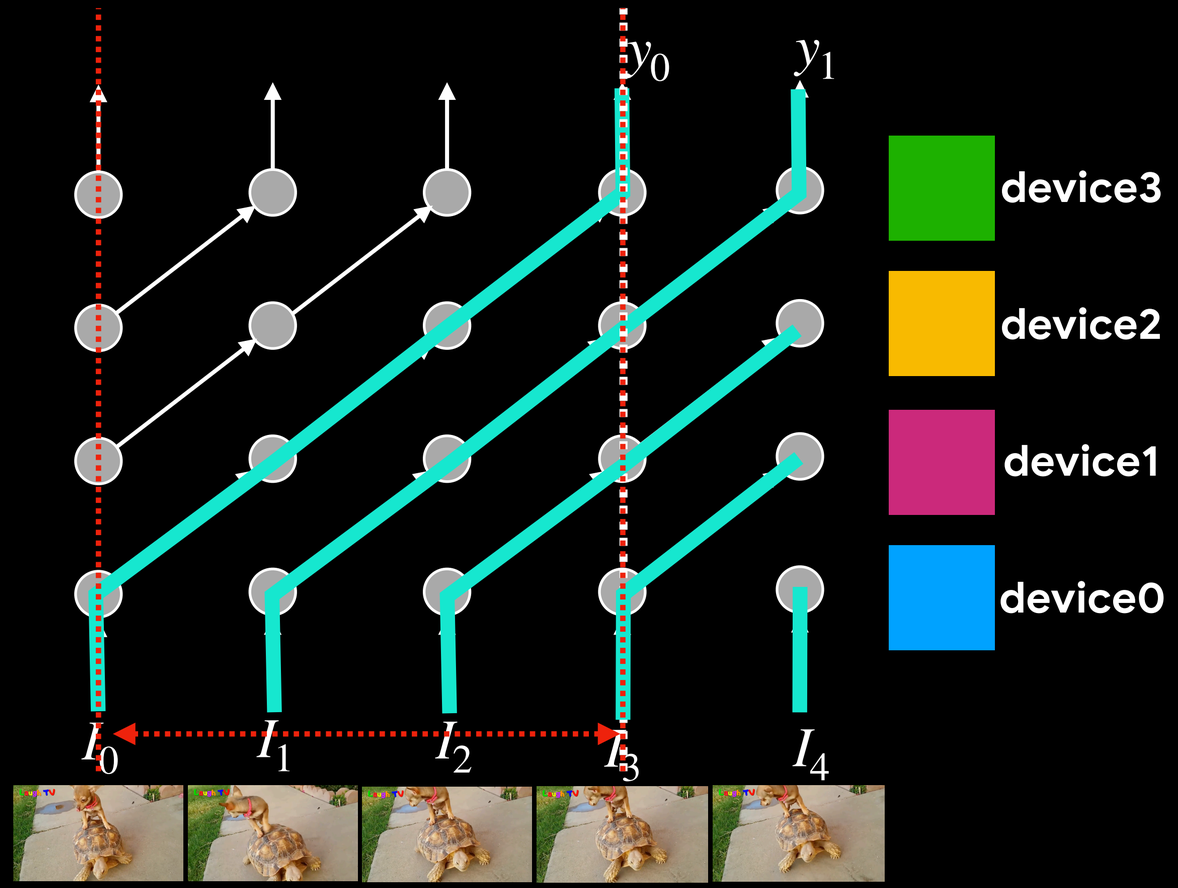
Reduce clock cycles

DeepMind

Predictive pipelined model

Pro:
high throughput

Con:
same latency



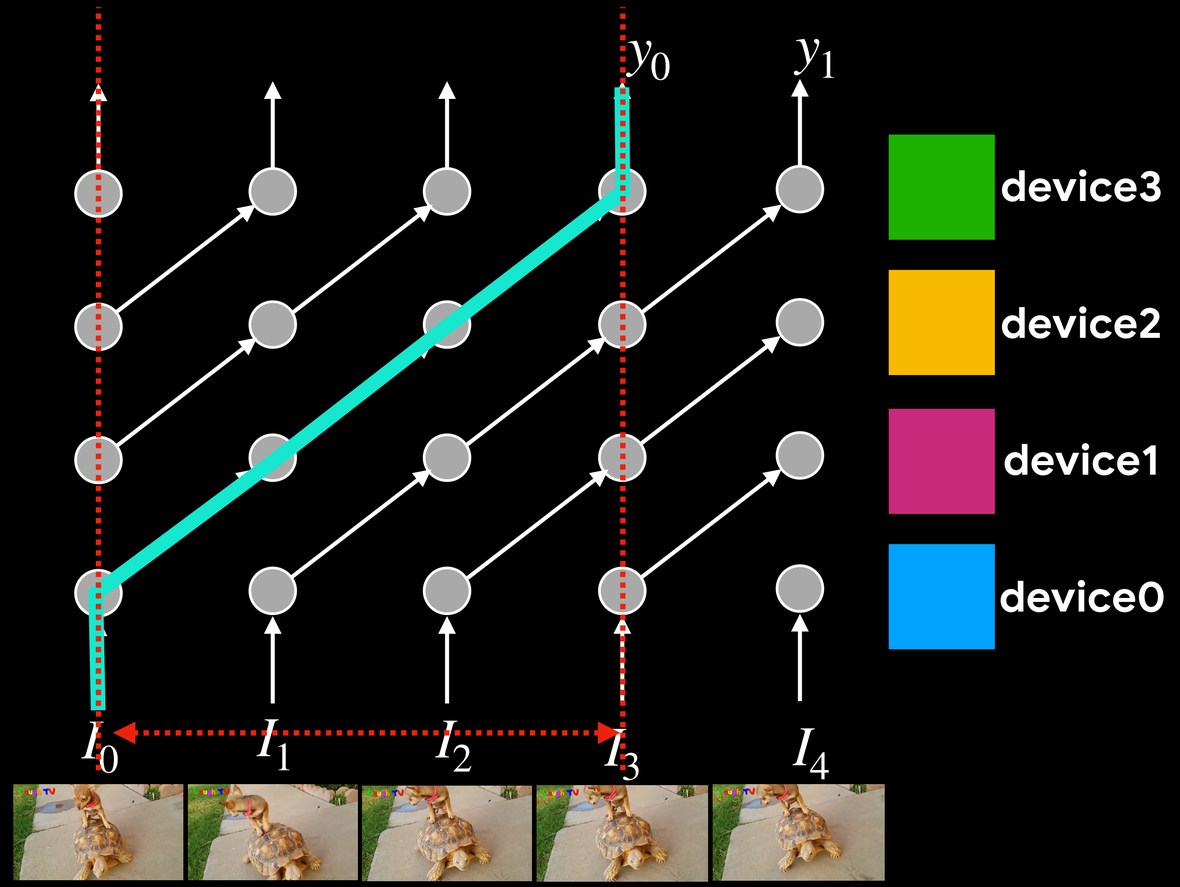
round-robin over layers

DeepMind

Predictive pipelined model

Pro:
high throughput
low latency

Con:
more difficult task
reduced accuracy

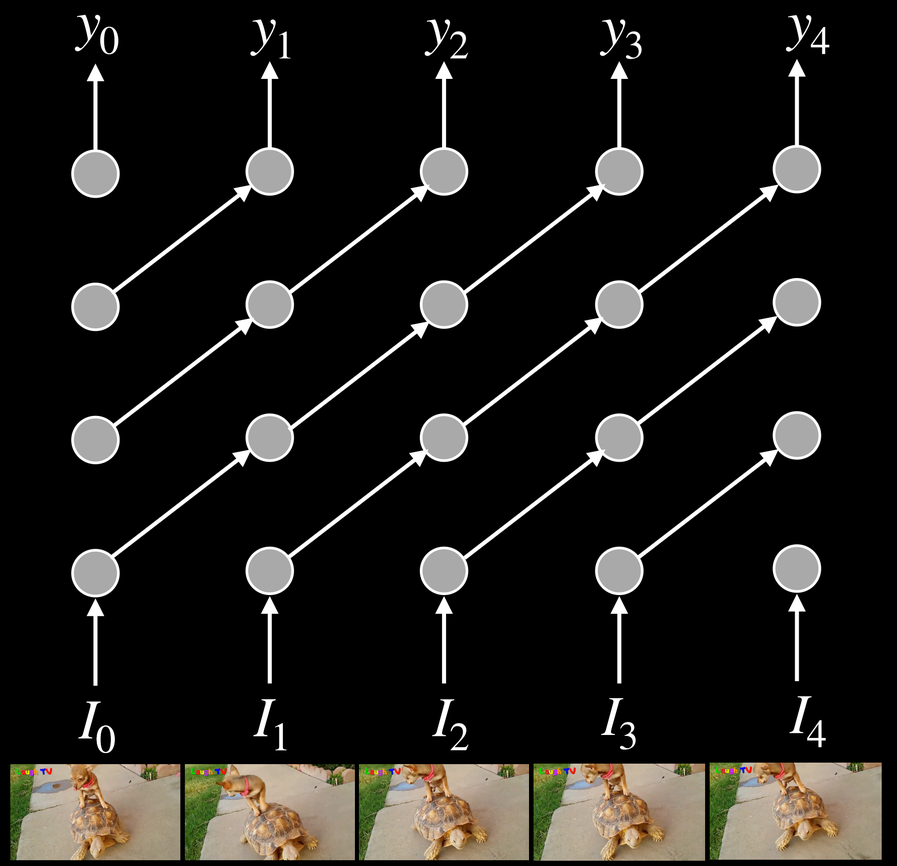


round-robin over layers

DeepMind

1. Partial pipelining

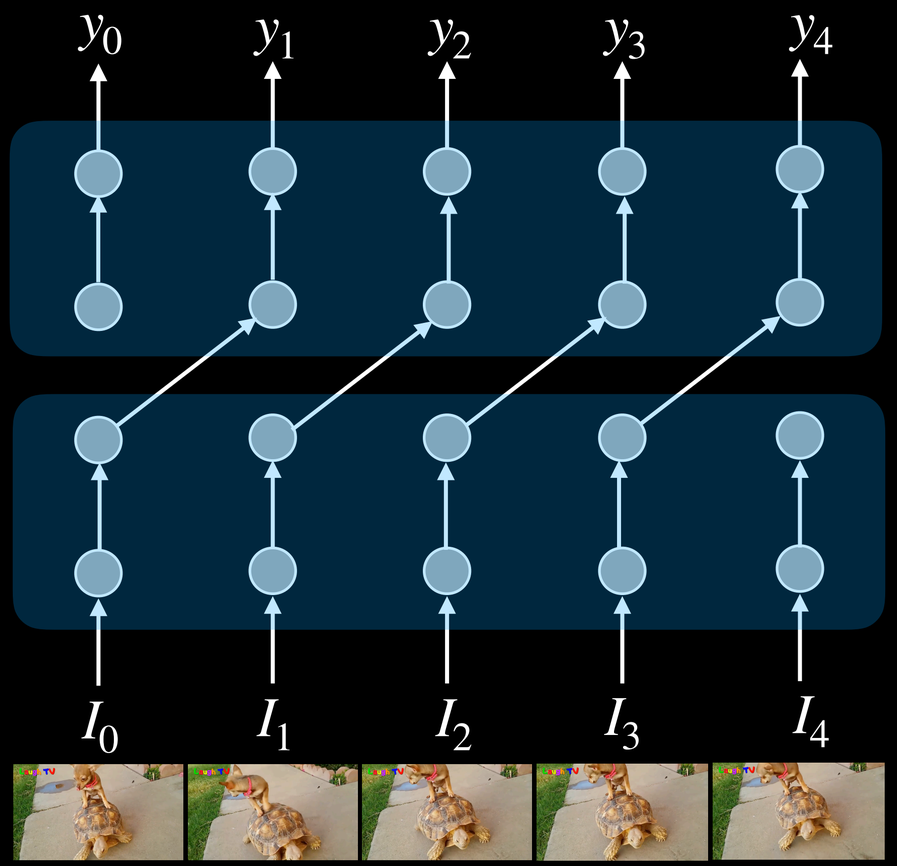
Fully-parallel model



DeepMind

1. Partial pipelining

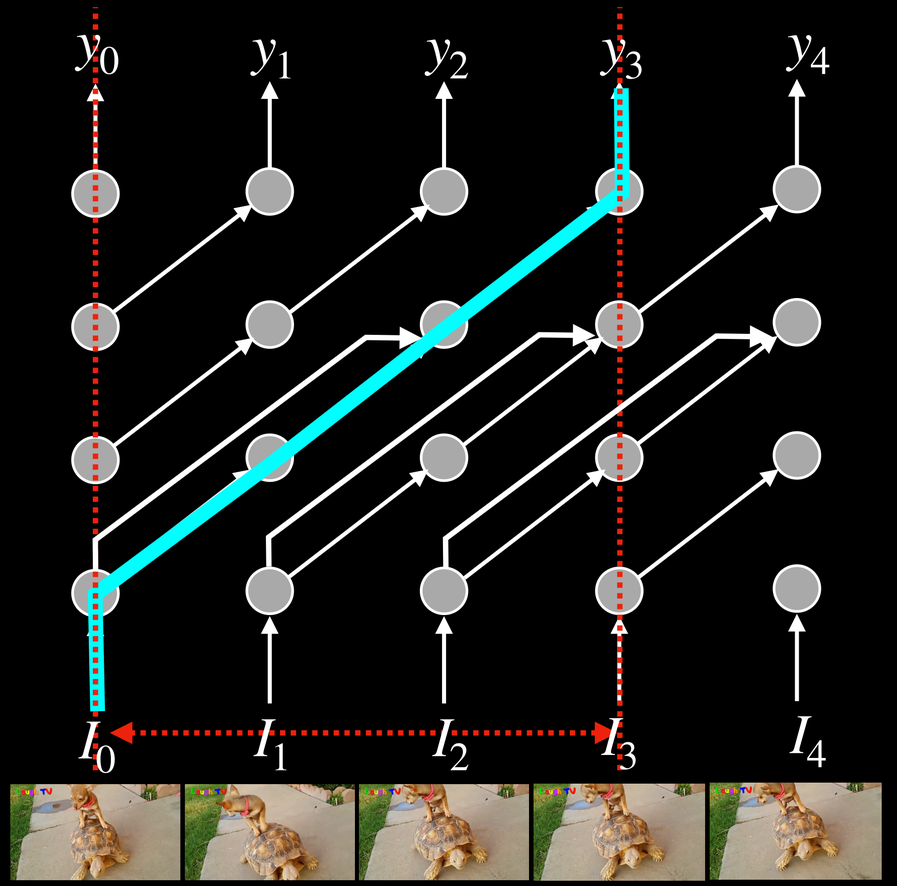
2 parallel subnetworks

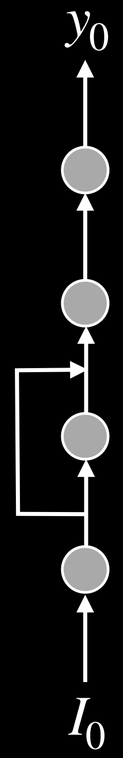


Pro: trade-off latency vs accuracy

DeepMind

2. Skip connections



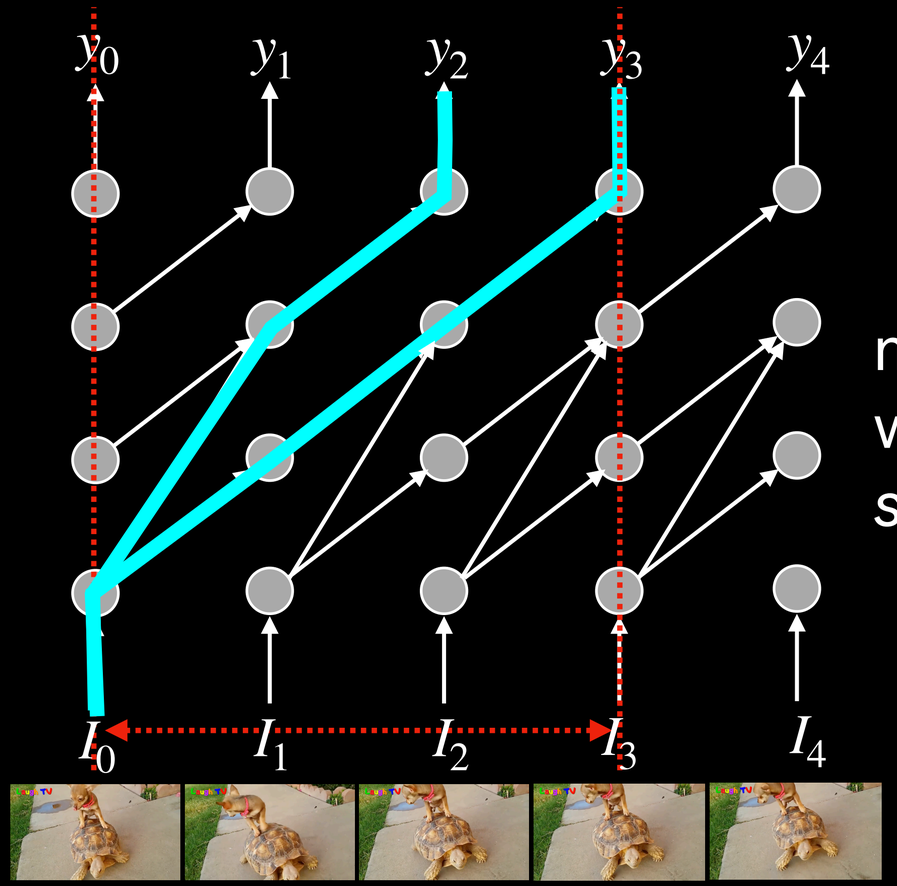


standard skip connection

DeepMind

2. Skip connections

Temporal
skip connections



merge fresh shallow features
with old deep features
slowness principle [1]

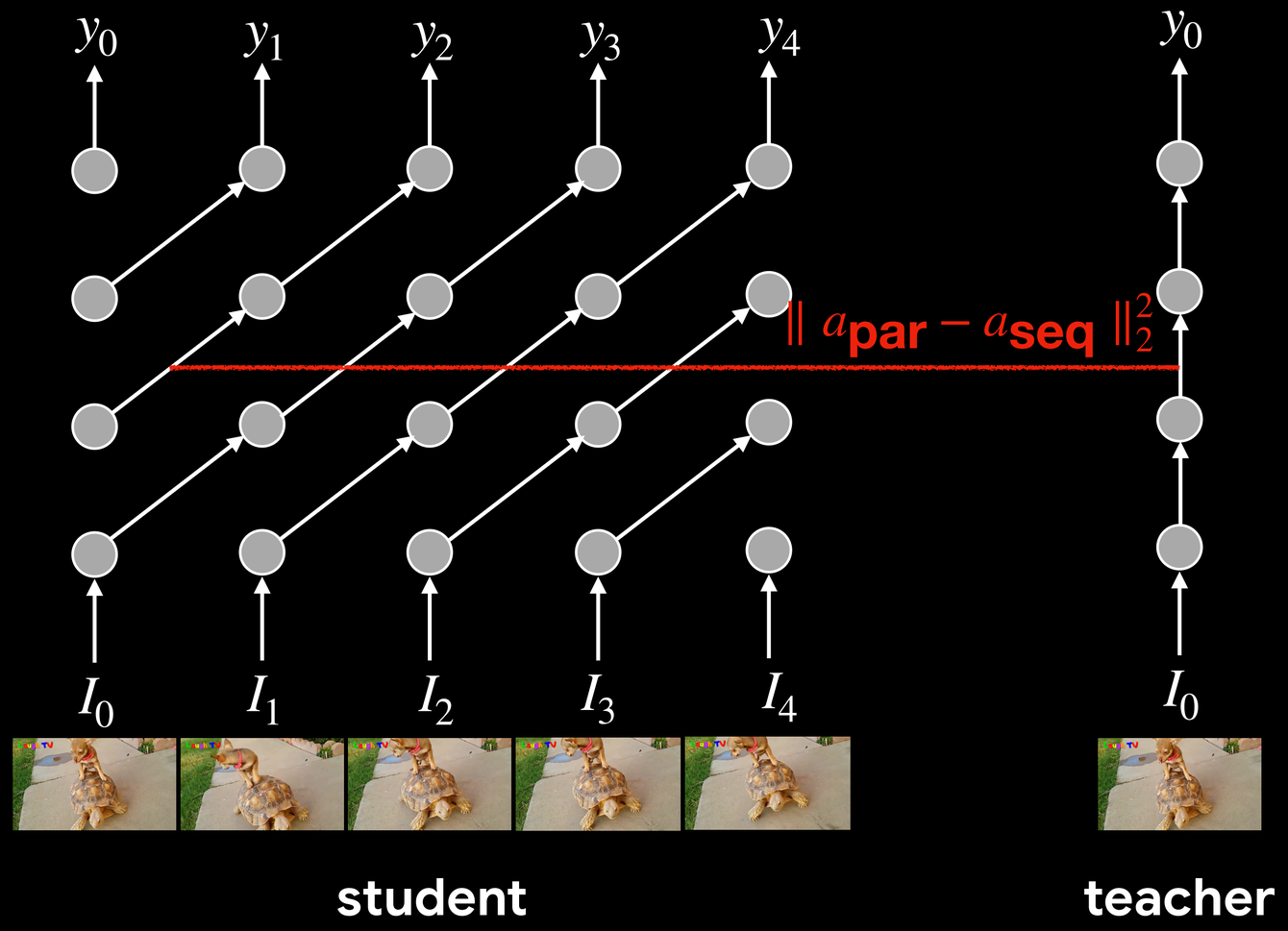
[1] Wiskott and Sejnowski, Slow feature analysis: Unsupervised learning of invariances. Neural Computation 14(4) (2002)

DeepMind

3. Distillation

teacher - sequential model, accurate but slow
student - pipelined model, fast but inaccurate

Additional loss term: intermediate activations should match



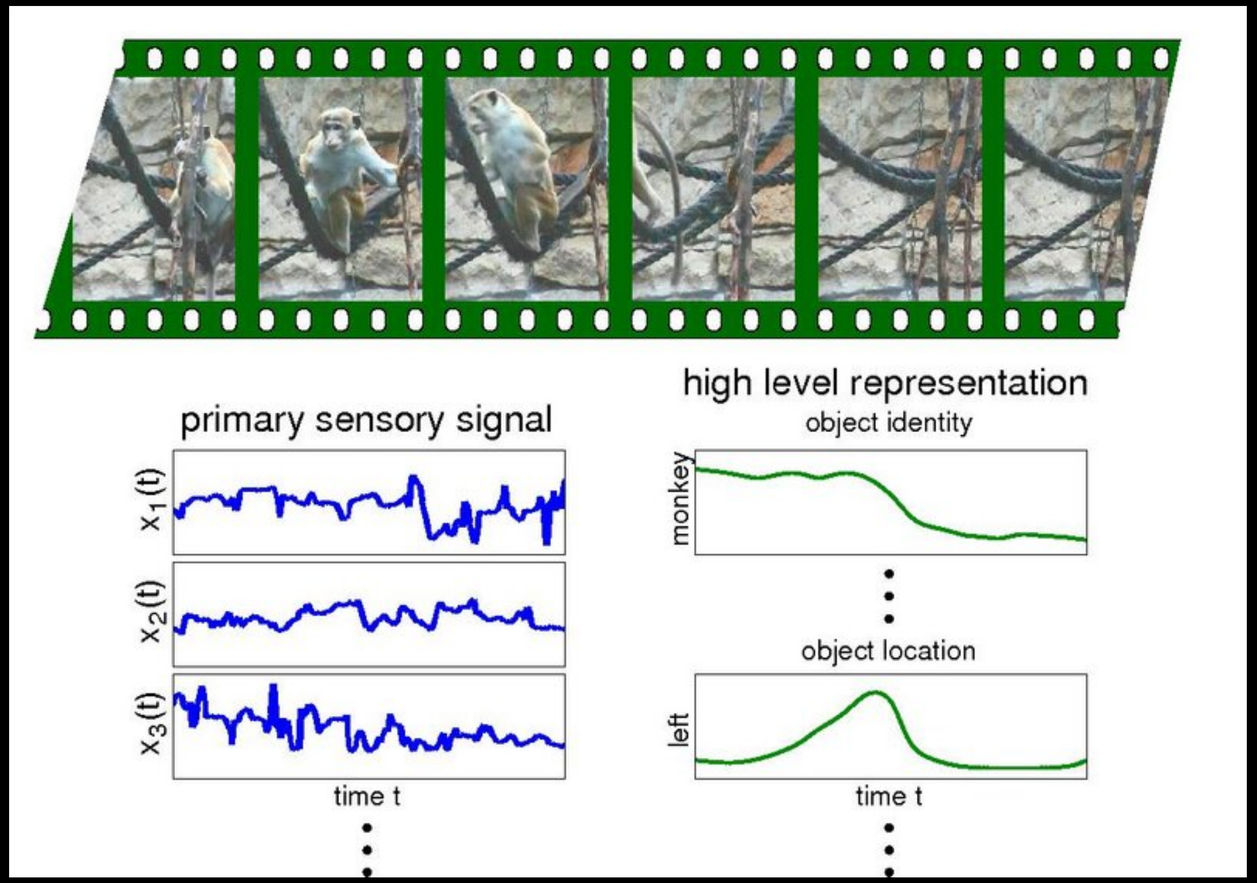
DeepMind

Increase throughput

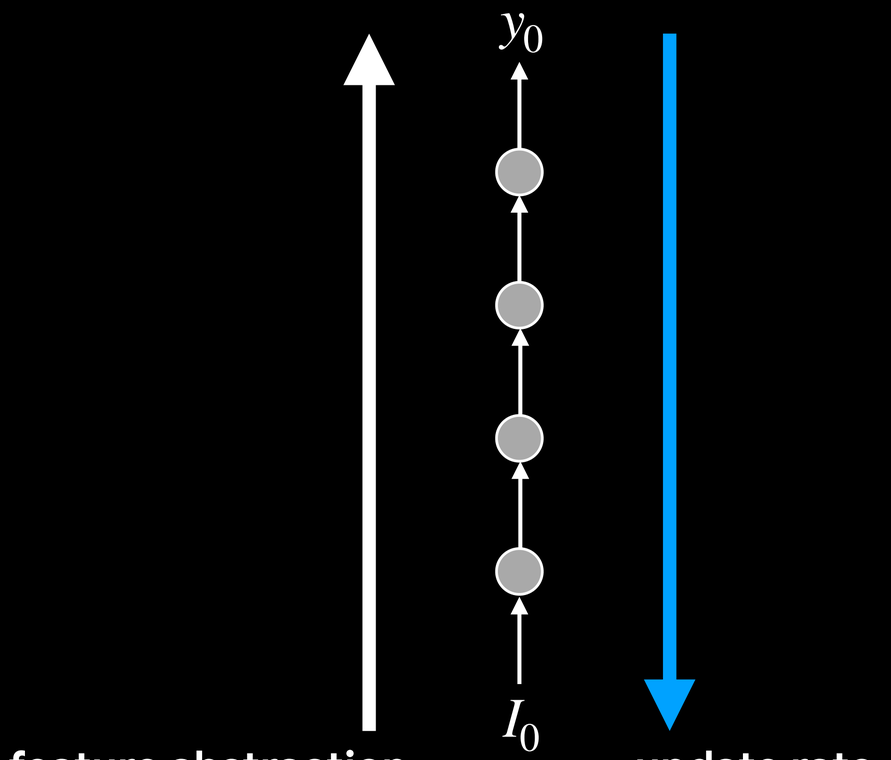
Reduce latency

Reduce clock cycles

Slowness principle



http://www.scholarpedia.org/article/Slow_feature_analysis



Exponentially reduced update rates (Wavenet style) <=> causal conv3D temporal stride > 1

Summary: predictive depth-parallelism

1. pipelined ops
2. temporal skip connections

} changes in temporal connectivity

3. distillation — change in train loss
4. multi-rate clocks — change in graph connectivity

Latency not dependent on depth

DeepMind

Experiments

Investigate the effect of **pipelining** on:

- **accuracy**
- **speedup (throughput)**

Tasks

Slow task
action recognition

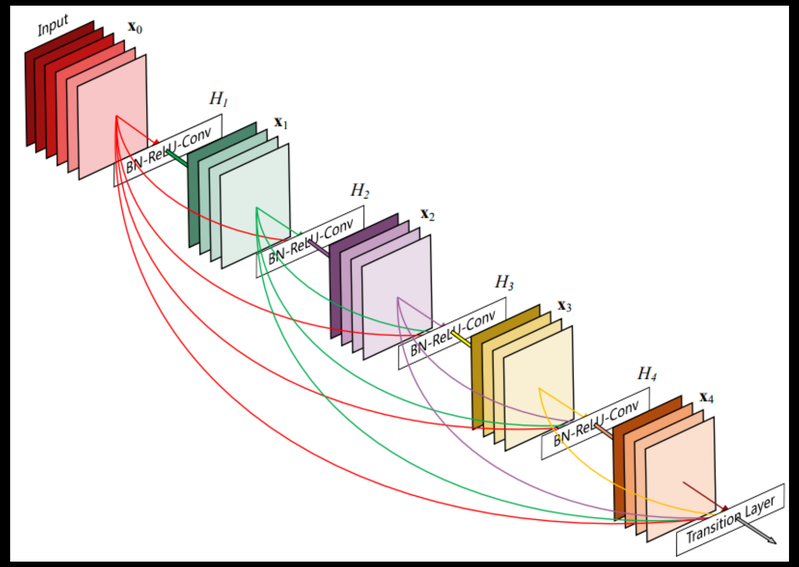
Fast task
human keypoint localisation

per-frame labels: *13D joints heatmaps*

DeepMind

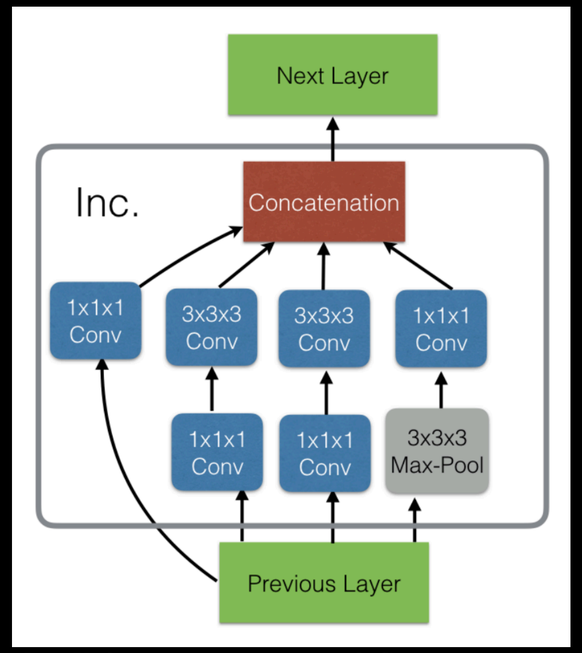
Models

Par-DenseNet



Densely connected convolutional neural networks, Huang et al., CVPR2017

Par-Inception



Quo Vadis, action recognition? Carreira and Zisserman, CVPR2017

DeepMind

Dataset

Mini-Kinetics
200 action classes
80k training videos
5k validation

automatically generated
“ground truth” keypoints [1]

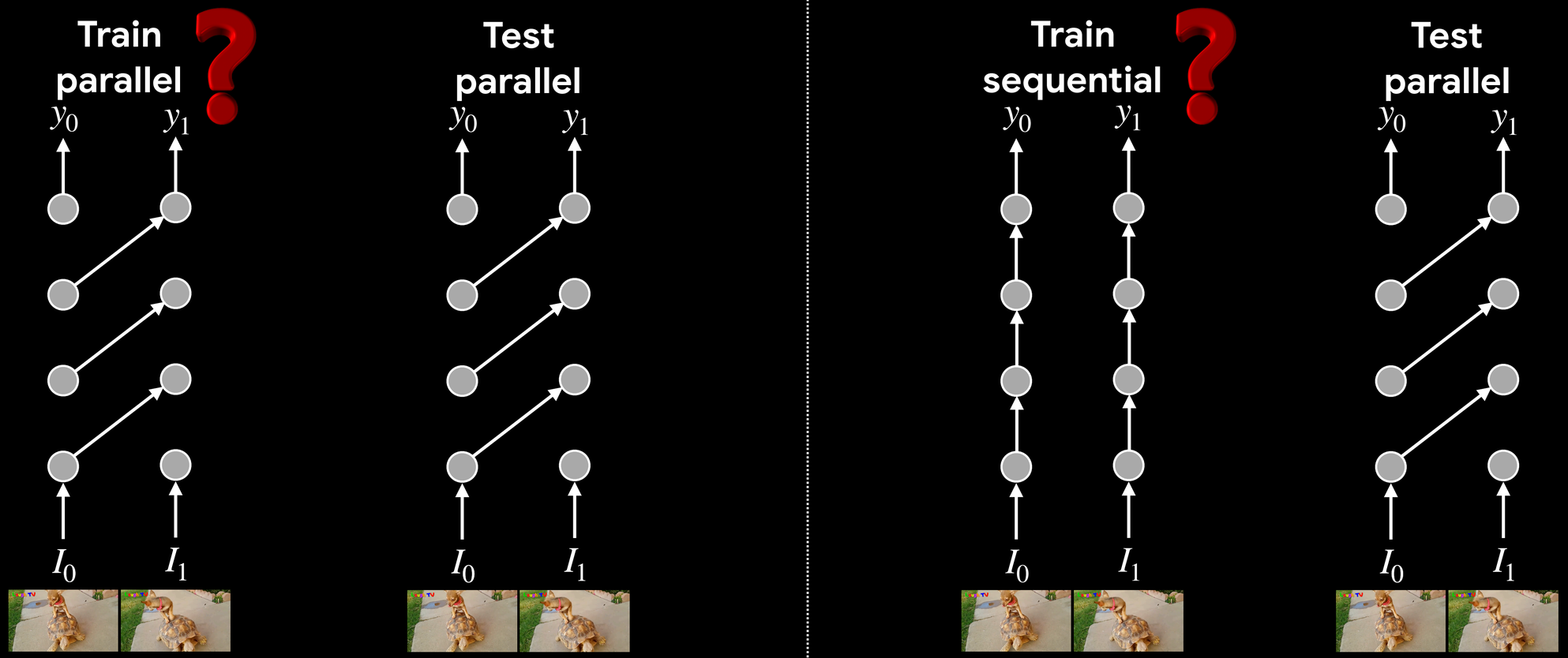
[1] Papandreou et al., Towards Accurate Multi-person Pose Estimation in the Wild, CVPR2017

DeepMind

Results: Human keypoint localisation

Results: Predictive parallelism

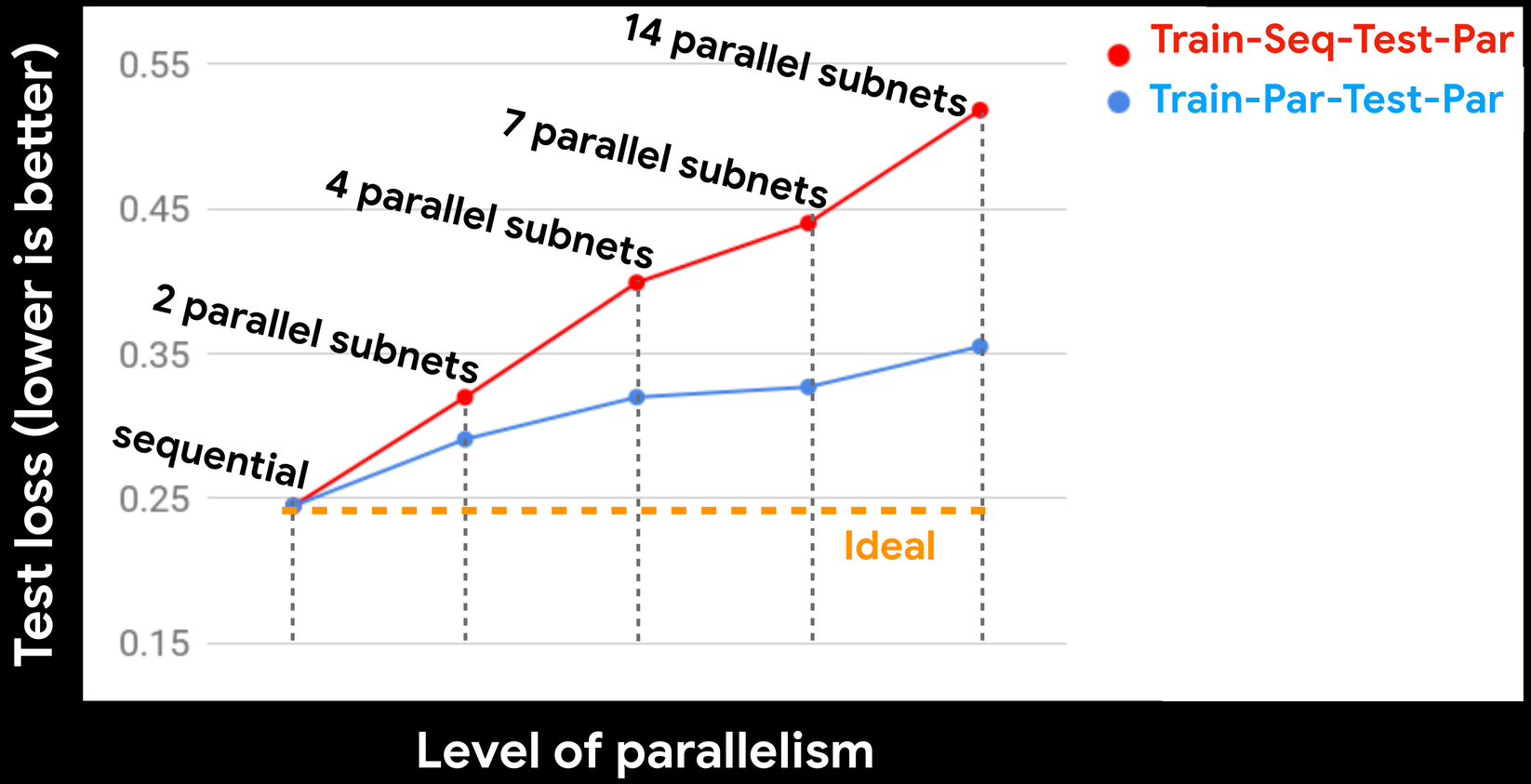
Q: Can we just parallelise at test time?



DeepMind

Results: Predictive parallelism

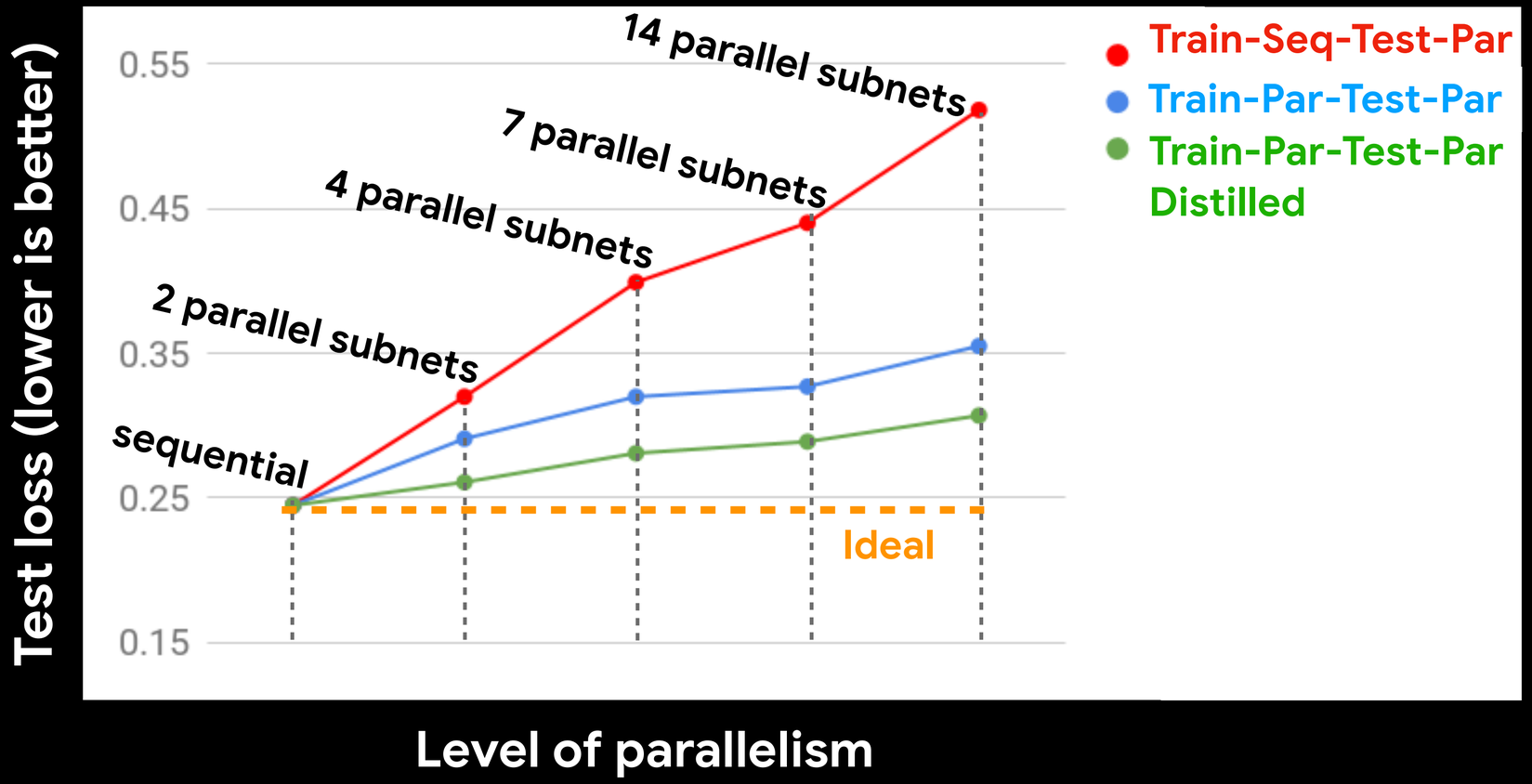
Q: Can we just parallelise at test time? A: NO



DeepMind

Results: Predictive parallelism

Q: Can we just parallelise at test time? A: NO



DeepMind

Results: Human keypoint localisation

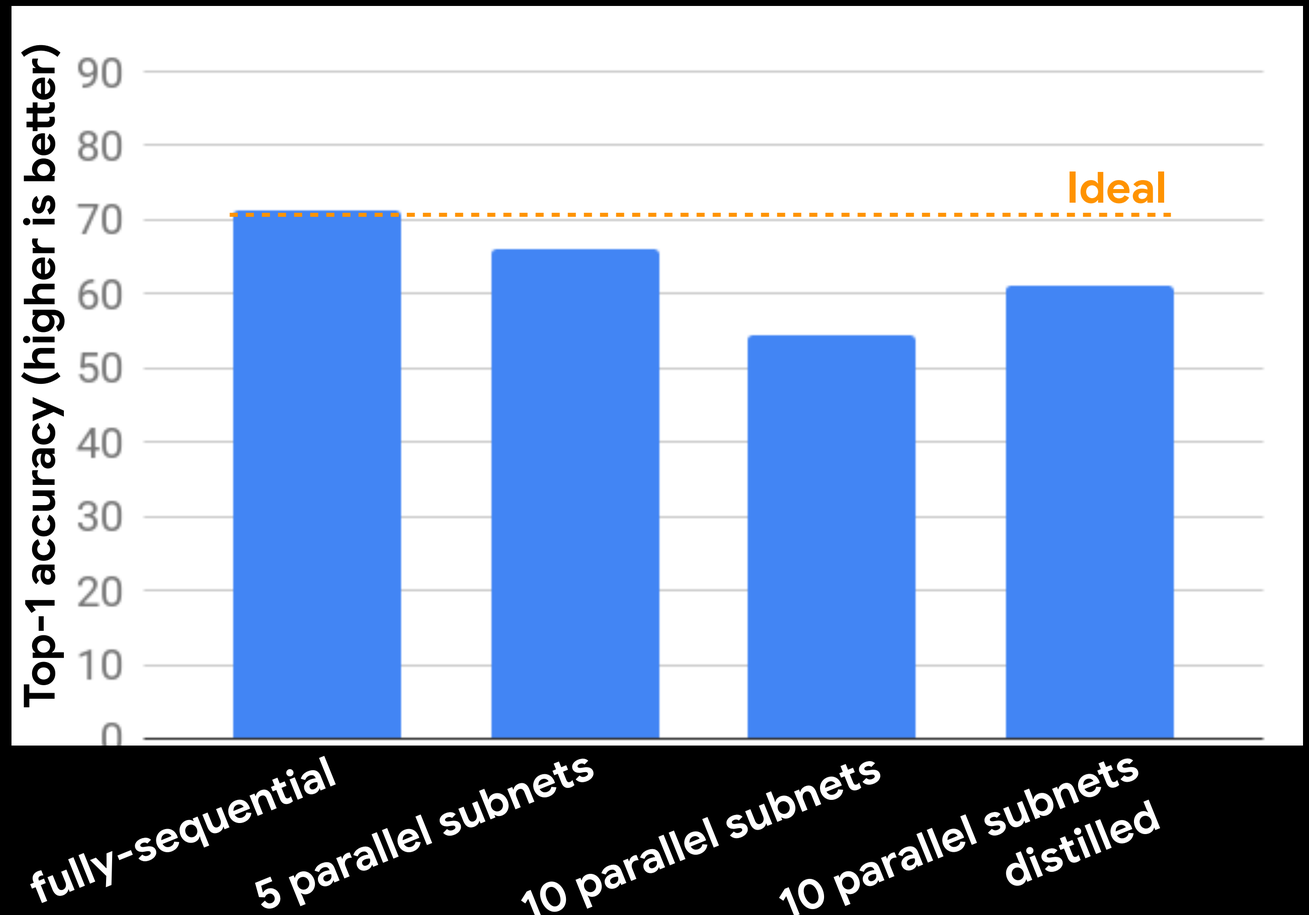
Input

Fully-sequential model
running offline

Fully-parallel model
running online

DeepMind

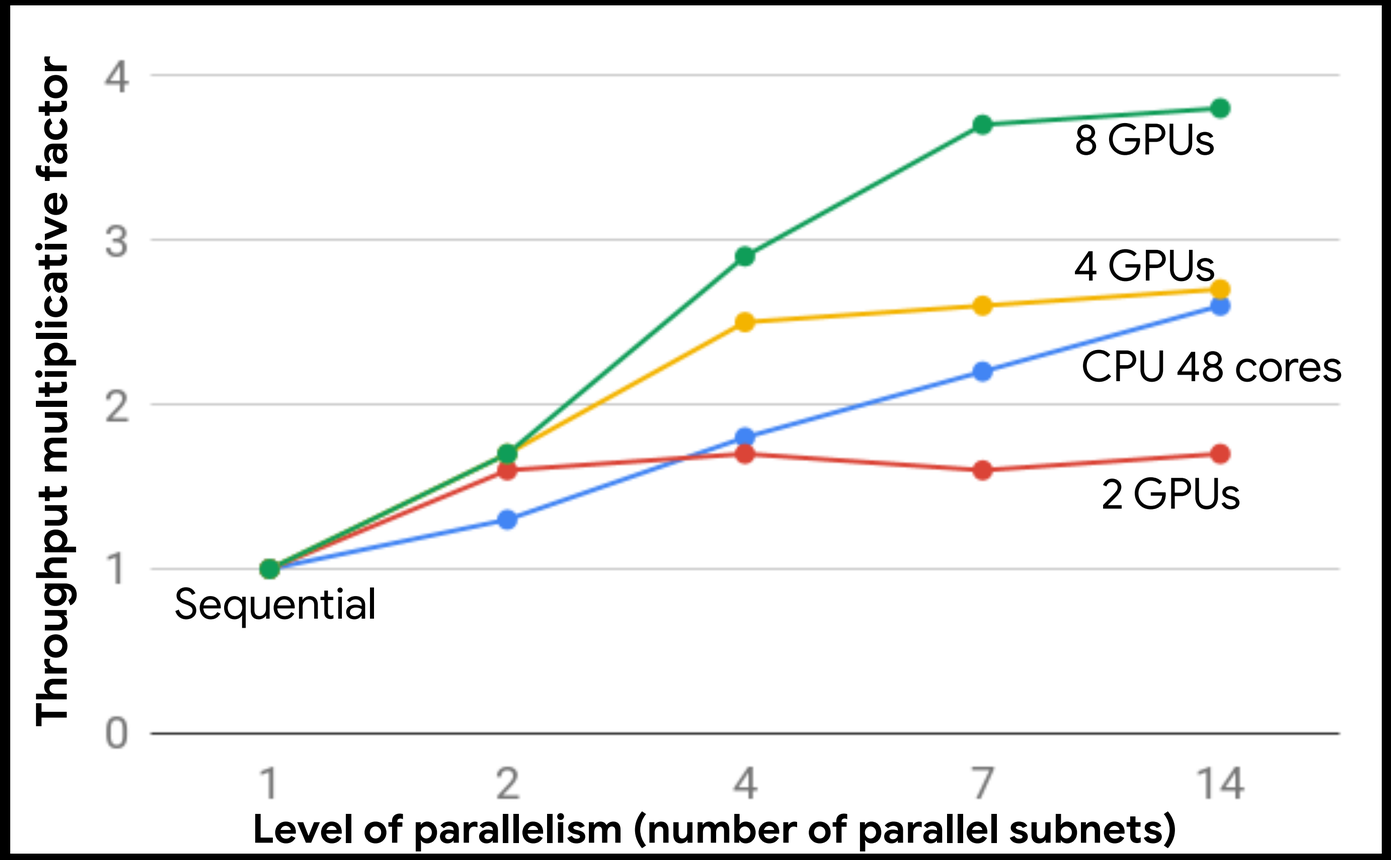
Results: Action recognition



DeepMind

Speedup

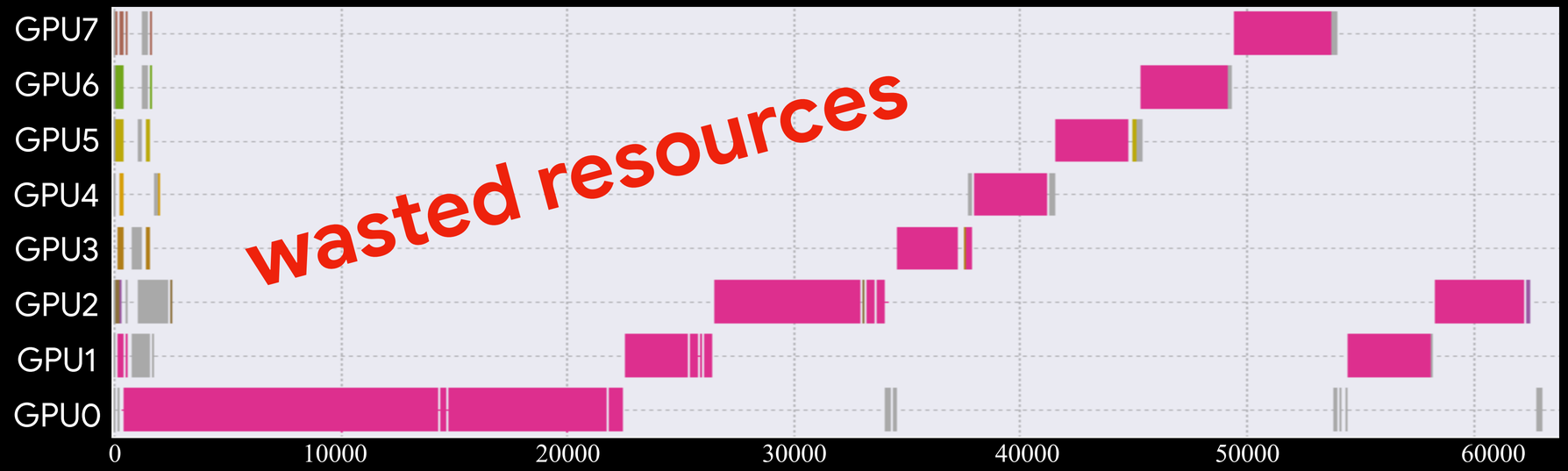
Speedup - CPU or GPU box



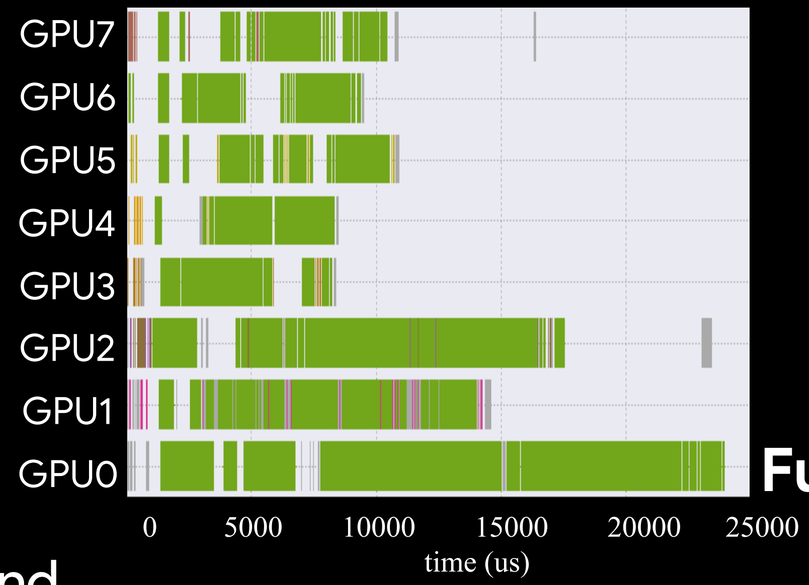
Physically colocated hardware

DeepMind

GPU utilisation



Fully sequential



Fully parallel

DeepMind

Conclusion

Promising new avenue for scalable video networks design

Important speed-up gain: 4x - 8x

Interesting aspect: latency reduction shapes representations

More investigation of architecture space needed (evolution strategies, Hebbian learning)