TinyML Is Not Thinking Big Enough

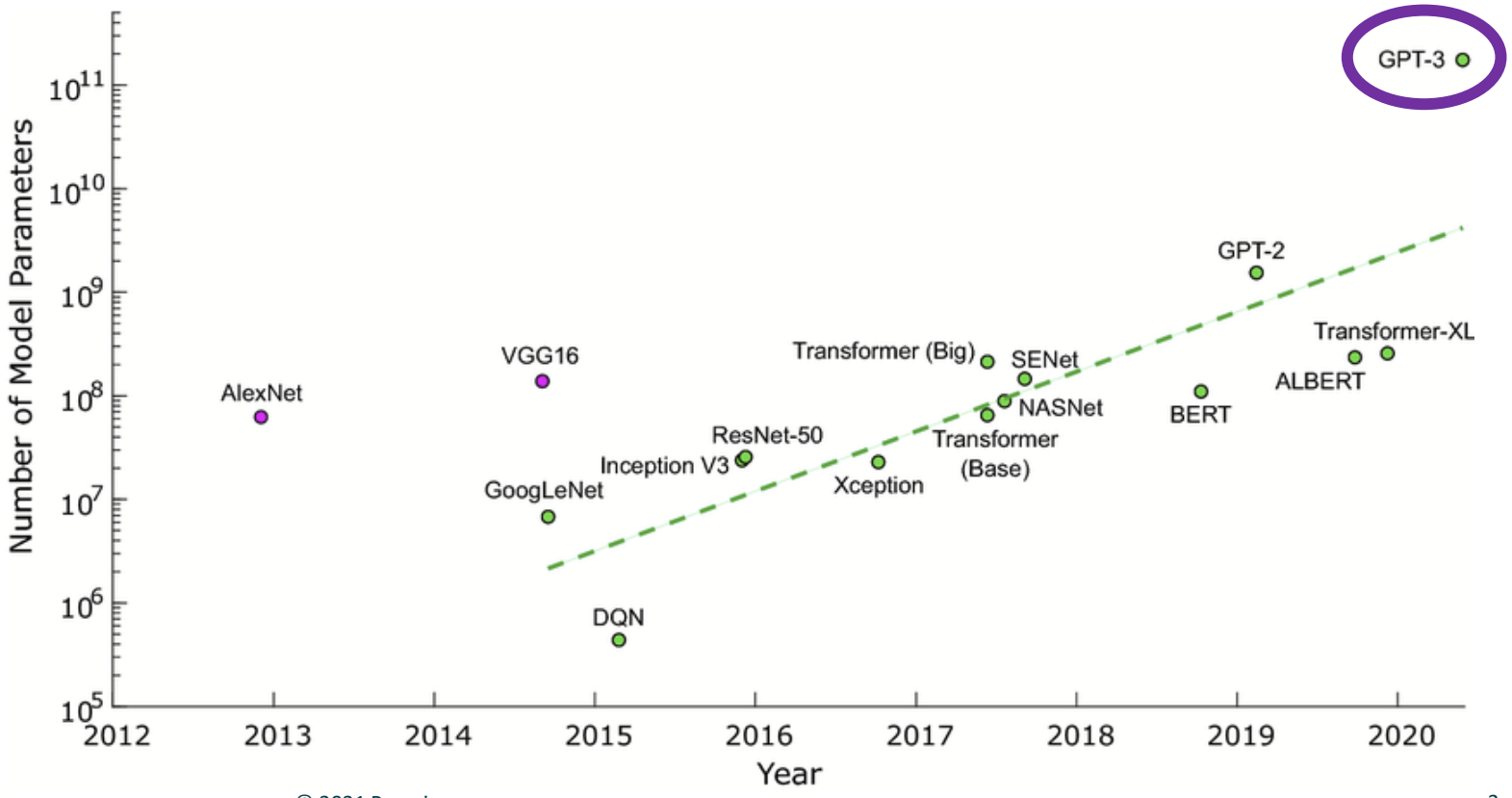
Steve Teig
Perceive

Perceive™

The Bridge of Size

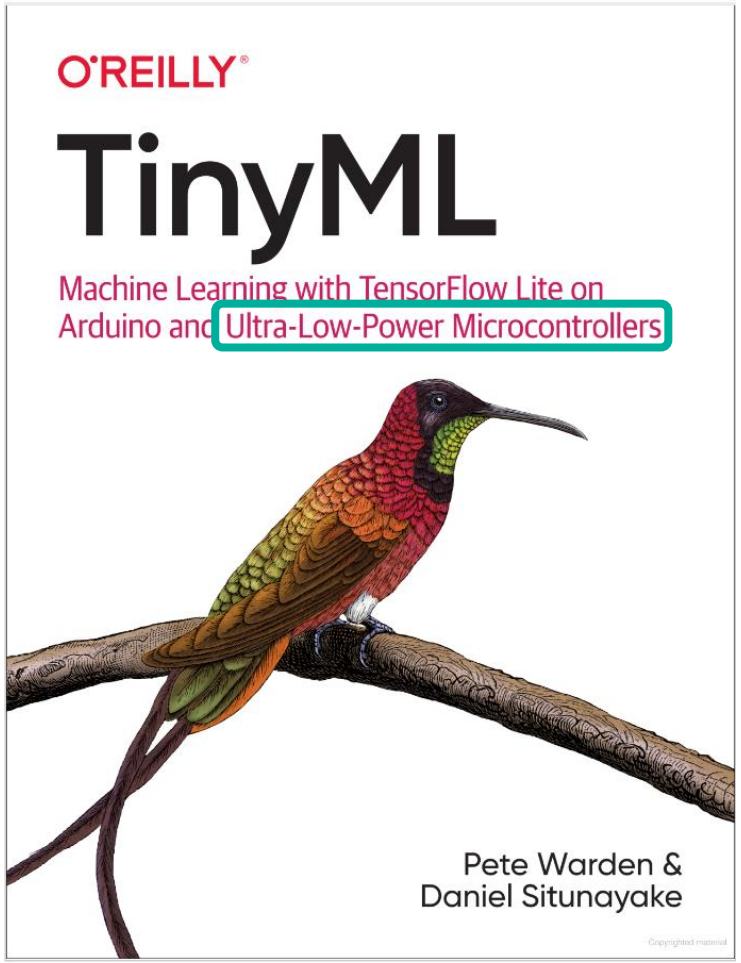
Models are getting bigger every year

- “Networks have to be huge”
- “Significant compression significantly compromises accuracy”



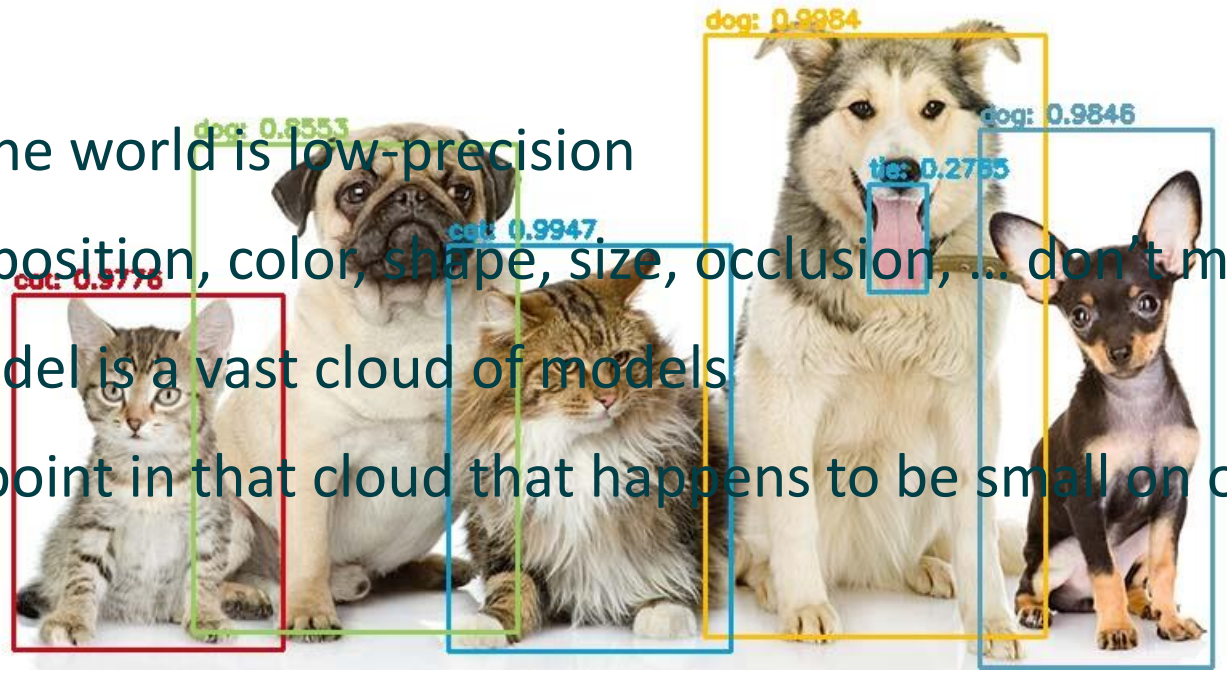
TinyML

- “Tiny networks should be deployed on CPUs and microcontrollers”



Why are today's networks so large?

- NAS-FPN (Ghaisi et al., 2019): 166.5 M parameters
  - 166.5×10^6 parameters x 32b/parameter = ~5.3 Gb
- Every bit corresponds to a yes/no question
- Do we really need 5.3 billion questions and 2.6 TFLOPS to find the dog in the picture?



- Our model of the world is low-precision
- Slight shifts of position, color, shape, size, occlusion, … don't matter to us
- Our mental model is a vast cloud of models
- We can pick a point in that cloud that happens to be small on our hardware

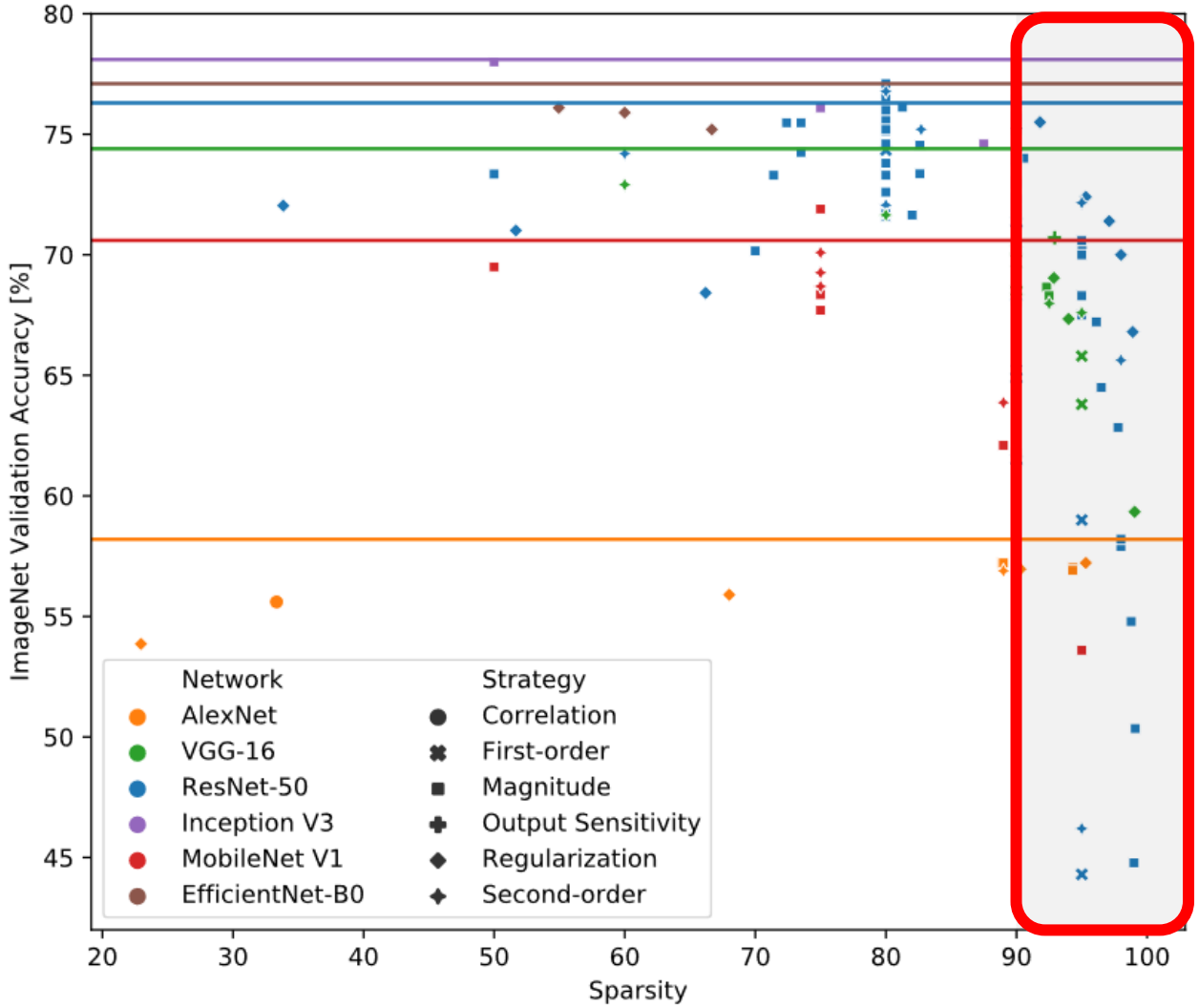
Why are today's networks so large?, cont.

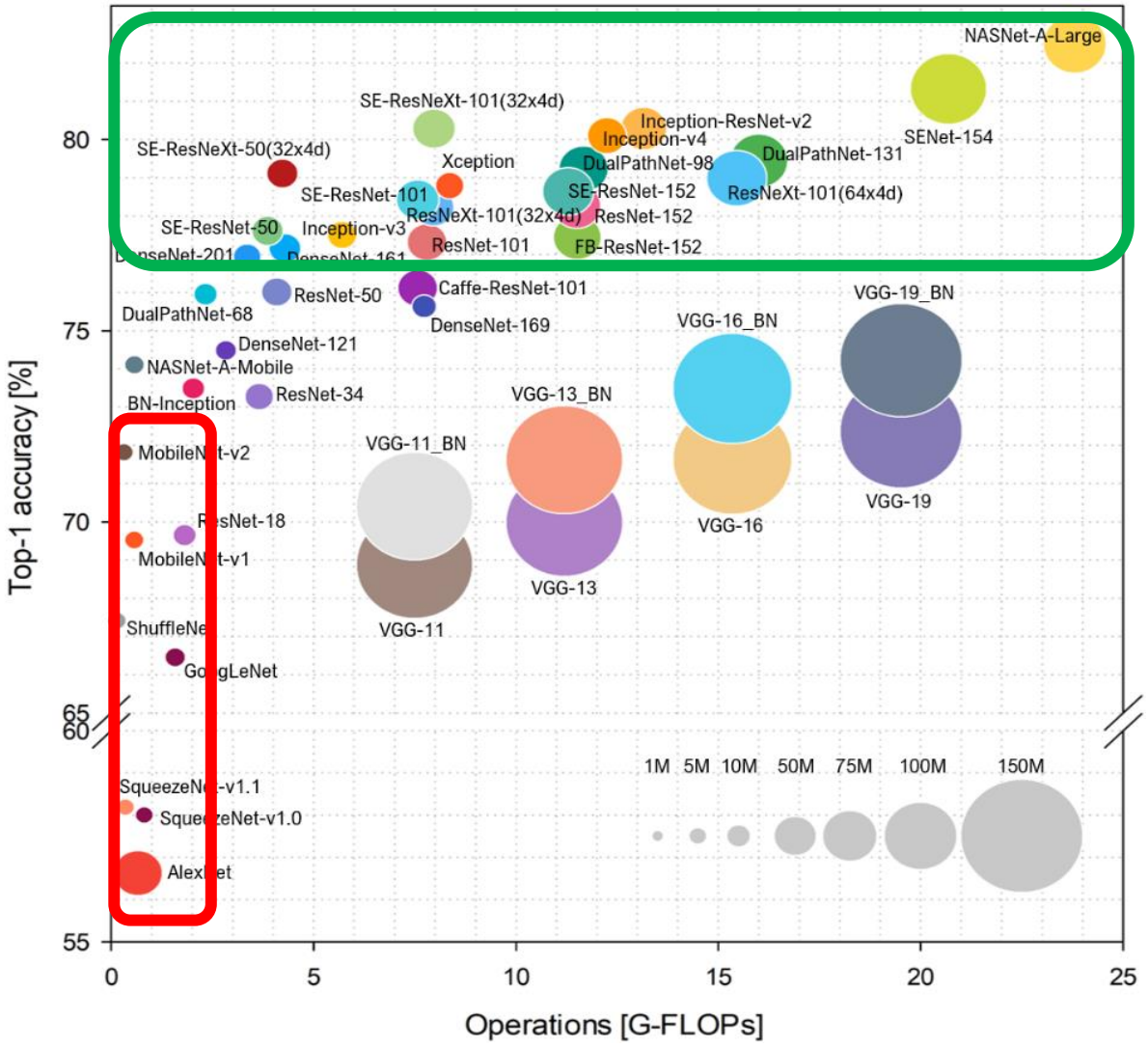
- GPT-3: 175 B parameters
  - At 32 bits per parameter: 700 GB just for parameters!
  - Even with 8-bit parameters: 175 GB = 1,400,000,000,000 bits
- Do we really need 1.4 trillion questions… for anything?

- 175 B param
  - Matrix mult
- Terrible repr

uting hardware

Accuracy degradation with compression and/or sparsity





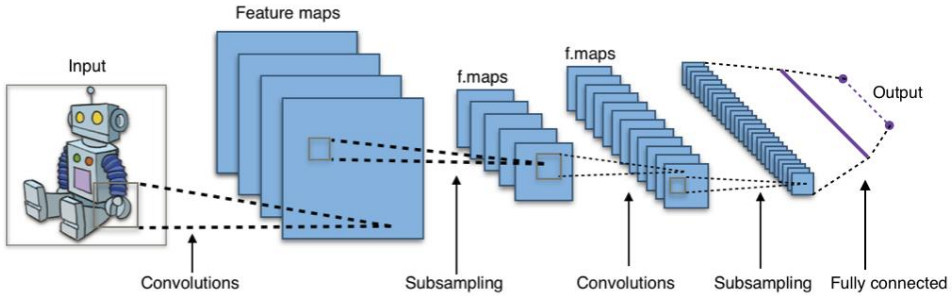
Neural network obesity… and a principled path to curing it

- What if we had a program, P0, that generates GPT-3's parameters as output?
- What if we had a program P1 that generates P0? P2 that generates P1? Etc.
- Computation is cheap, but memory storage and movement are expensive → compress!
- Number of parameters is a poor description of complexity → Kolmogorov is better

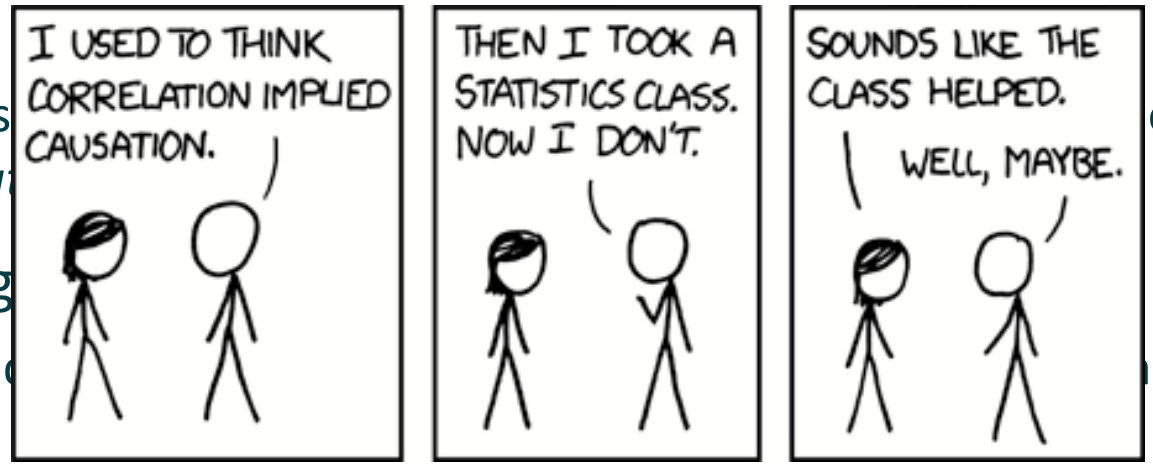
GPT-3's parameters

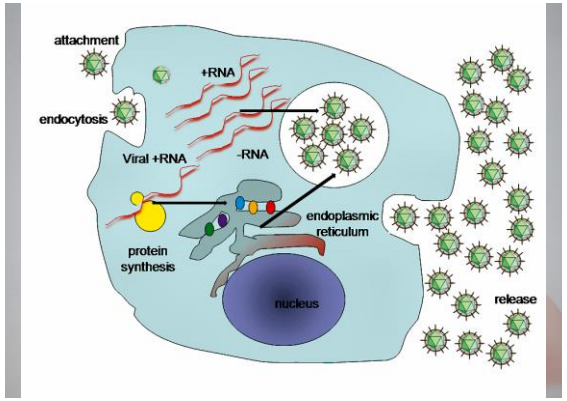
Compute-intensive decompression → better compression
Not as exotic as it sounds

- Convolutional neural networks
  - Assume the universe is translationally invariant
  - Store one set of weights → *apply* to every pixel or region



- Causality
  - No way of knowing whether the universe is causal
  - Assuming causality – present is *function*(past) – massively compresses our models
- Viruses
  - Viruses e "computation" (i.e., *replication* and *transla*
- Human g
  - Just ad !





Small is beautiful

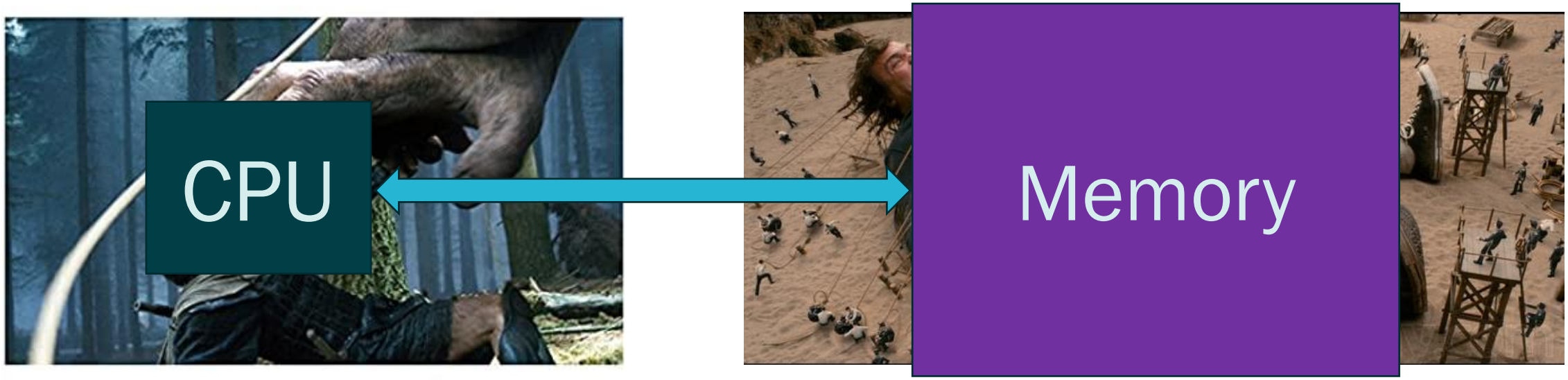
- Hutter: fastest program that computes f is among the shortest programs that (provably) compute f
  - https://arxiv.org/abs/cs/0206022
- Kolmogorov complexity: K(f) = length of shortest possible program to compute f

- Levin complexity: adds penalty to Kolmogorov complexity to include execution time
  - L(f) = min[K(f) + log(time(f))] = min[size of program + how long to wait]
- For ML, modified Levin complexity: shortest program given performance requirements

- Everyone wants fast ML → also wants small ML → everyone should want TinyML

CPUs are not well-suited for giant neural networks

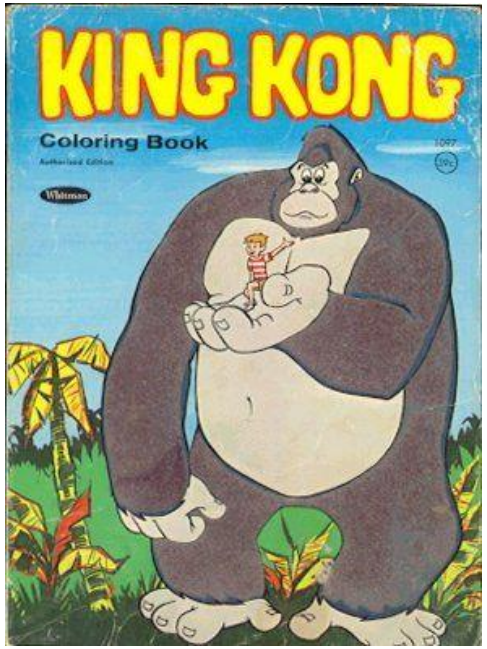
- Essential premises of von Neumann style of computation
  - Every operation loads from memory, computes something, and stores in memory
  - Memory throughput completely dominates system throughput
  - Execution model is (typically) serial and control-heavy → ~1 computing element



Neural networks beg for massive parallelism

- Typical networks run many filters on the same data – the previous layer(s)
- Independent computations can be run in parallel

- Moving data costs much more power and time than computing
- Parallel computing elements increase computing "surface area"
  - Close proximity to much more data → significant reduction in time and power



Improving ML as a whole

- Principled compression can improve ML everywhere
  - Needed for TinyML
  - Kolmogorov/Levin complexity reduces arbitrariness while compressing models
  - It's not about the number of weights or just using fewer bits per weight

- Massively parallel computation and ultra-low-latency memories make ML practical
  - Stop aiming TinyML at CPUs, microcontrollers, and the von Neumann bottleneck
  - Neural networks are highly parallel, non-von Neumann computing devices

TinyML is not thinking big enough

- If the goal of TinyML is…
- High *accuracy* at low power → principled compression + special-purpose hardware
- High *performance* at low power → special-purpose hardware
- High *performance* at high *accuracy* → special-purpose hardware + principled compression

- The goal of ML is high *performance* at high *accuracy* → TinyML is how to get there
  - And you get low power as a bonus!

- Perceive: novel, principled compression and special-purpose hardware

Resources

Perceive

https://www.perceive.io

TinyML

https://www.tinyml.org/

Levin complexity

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

2021 Embedded Vision Summit

"Facing Up To Bias" (Talk)