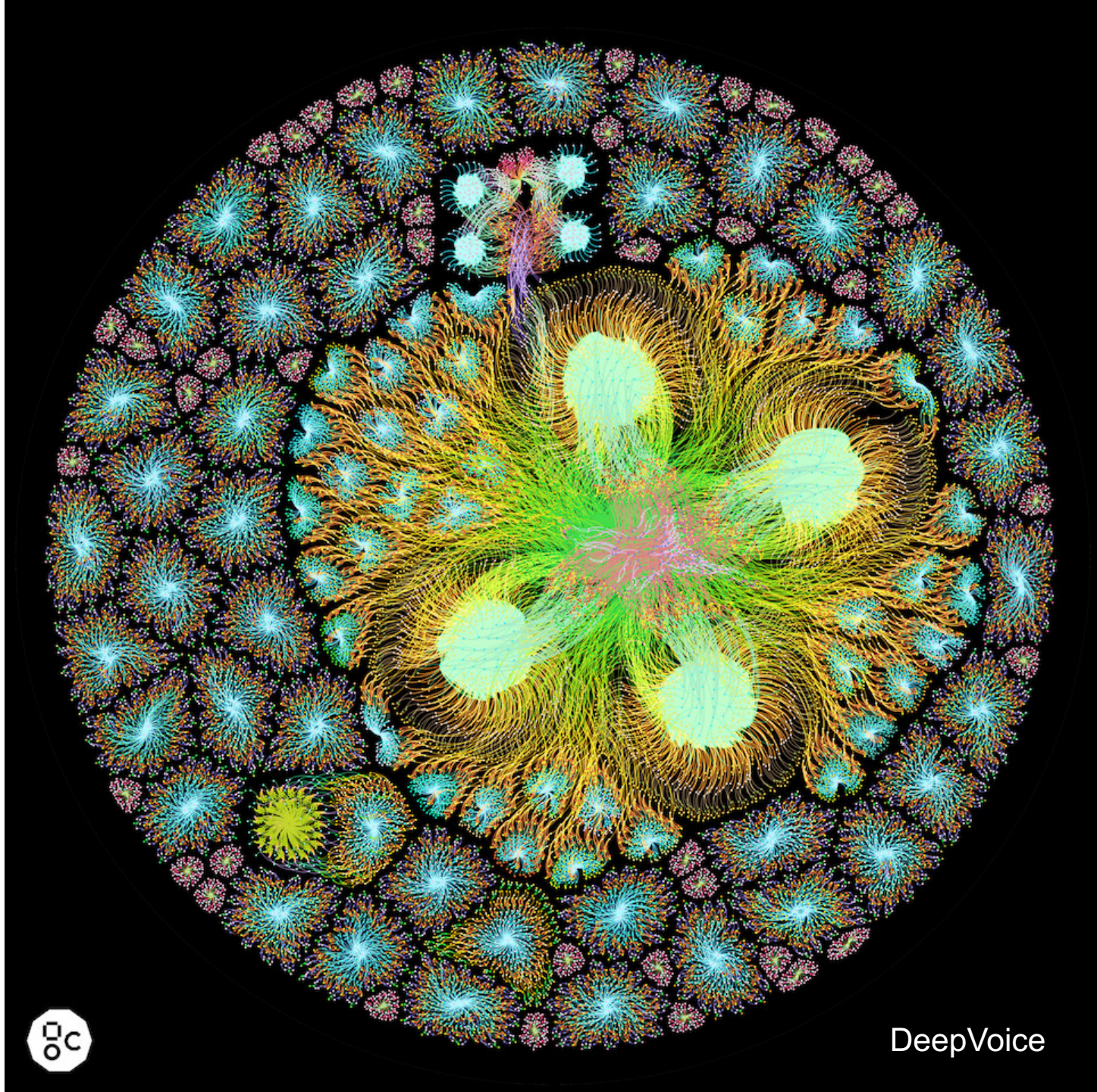
GRAPHCORE

Intelligence Processing Unit



INTELLIGENCE

“The capacity for rational decision-making, ultimately about actions, based on imperfect knowledge, adapted with experience”

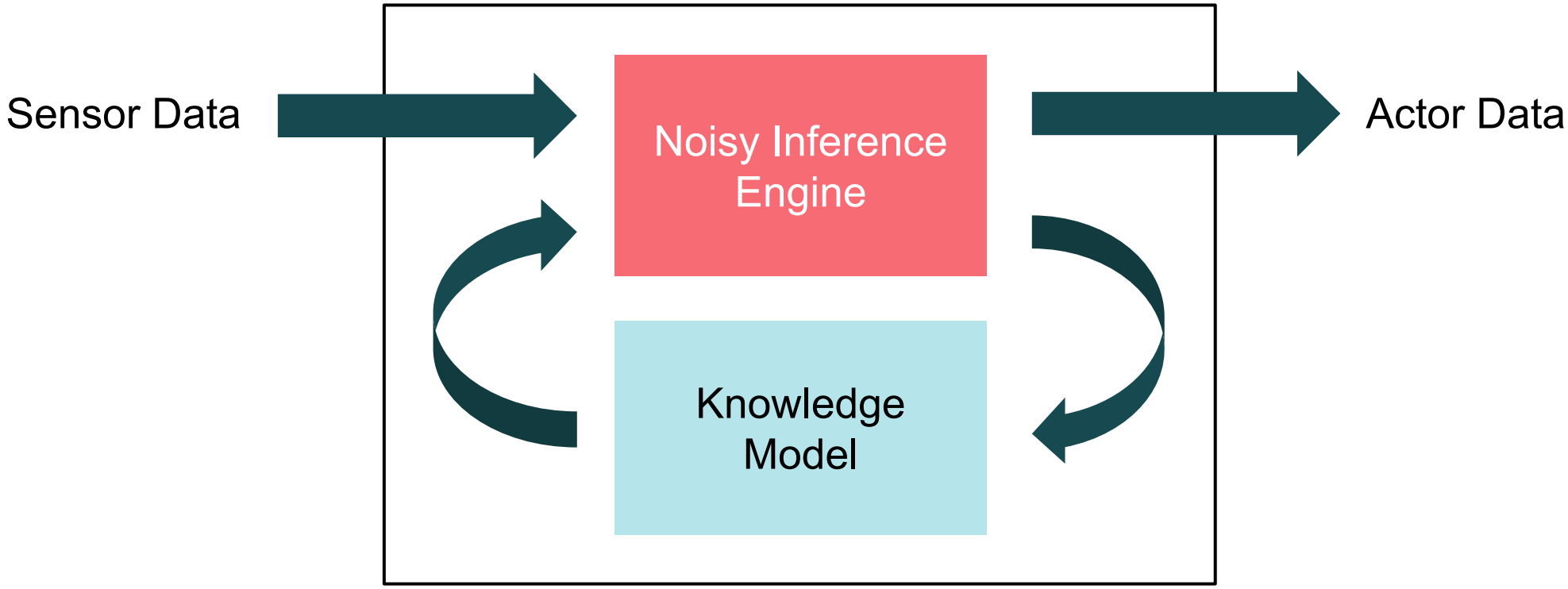
Intelligence is required for any agent operating under uncertainty

Talking with humans

Moving among humans

Learning to do human work

INTELLIGENCE MACHINE

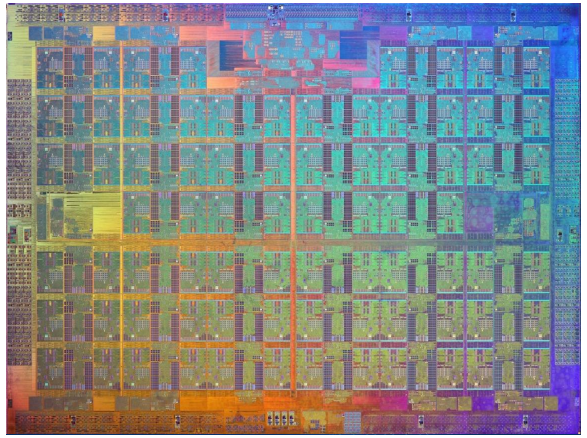


A canonical intelligent agent is a sequence-to-sequence translator.

Learning is inference (of model structure and parameters).

Inference is stochastic optimization, of some cost function.

INTELLIGENCE REQUIRES A NEW ARCHITECTURE

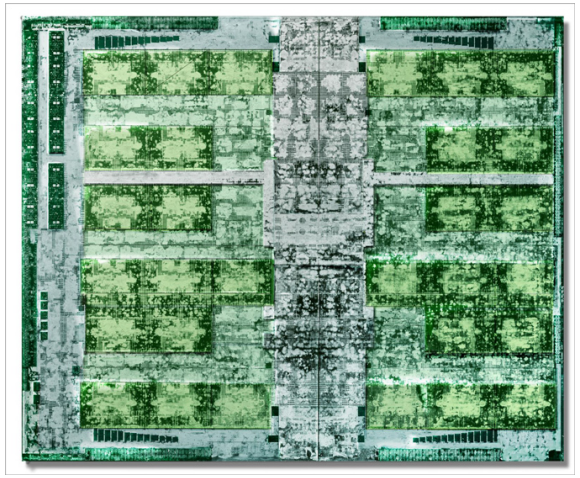


CPU

Scalar

Designed for office apps

Evolved for web servers

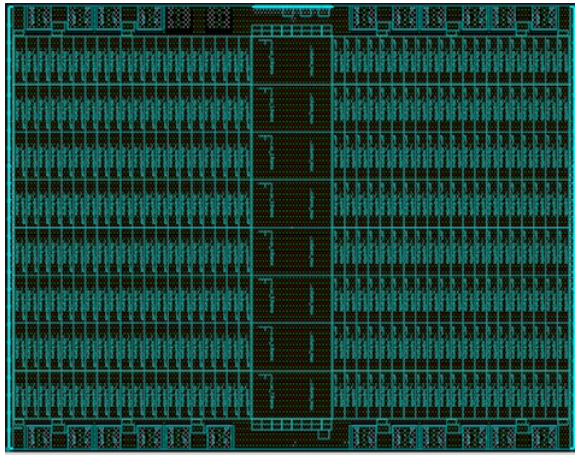


GPU

Vector

Designed for graphics

Evolved for HPC



IPU

Graph

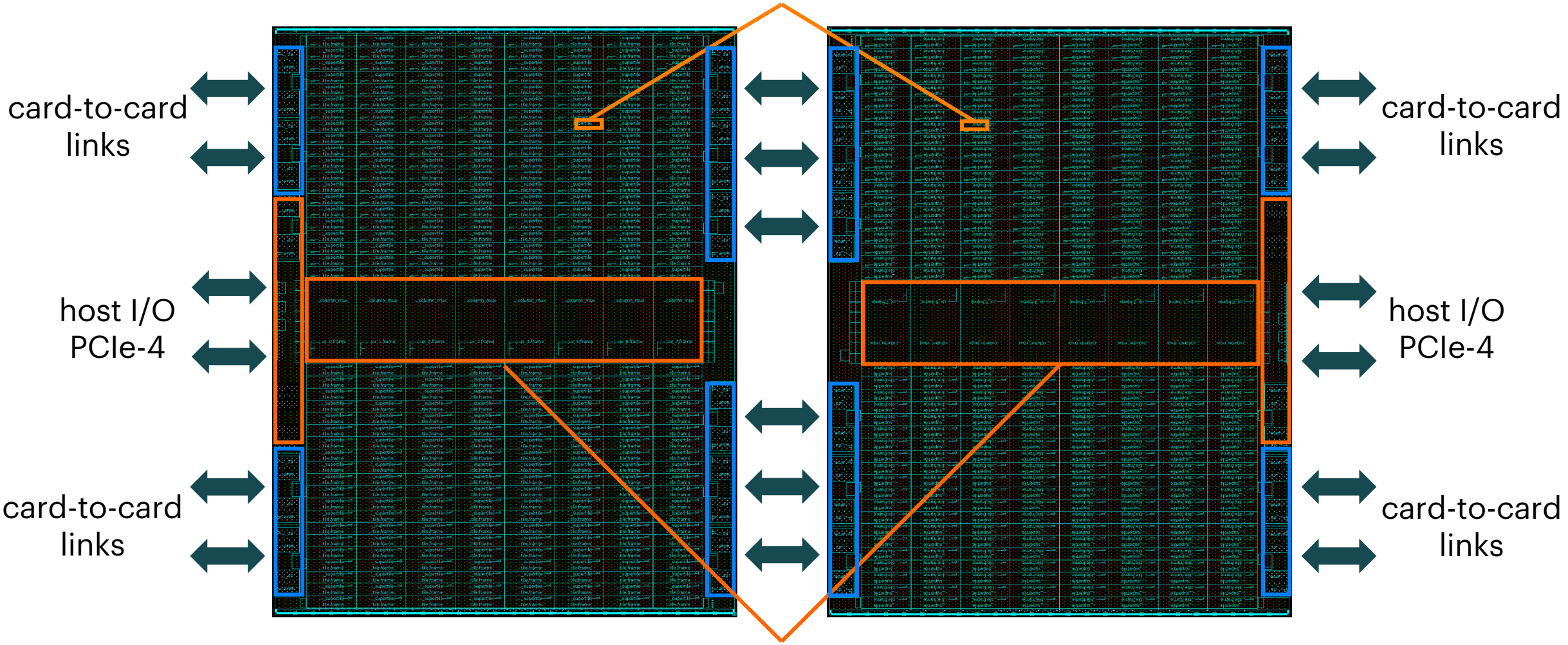
Designed for intelligence

INTELLIGENCE MACHINE CHARACTERISTICS

- Computation on graphs
  - massive parallelism
  - distributed memory
- Low precision, wide dynamic range arithmetic
  - mixed-precision float 16.32 (and smaller?)
- Static graph structure
  - compiler can partition work, allocate memory, and schedule messages
  - bulk-synchronized, address-less communication
- Entropy generative
  - noise in hardware

GRAPHCORE IPU

2000> processor tiles >200Tflop ~600MB



SILICON EFFICIENCY IS THE FULL USE OF AVAILABLE POWER

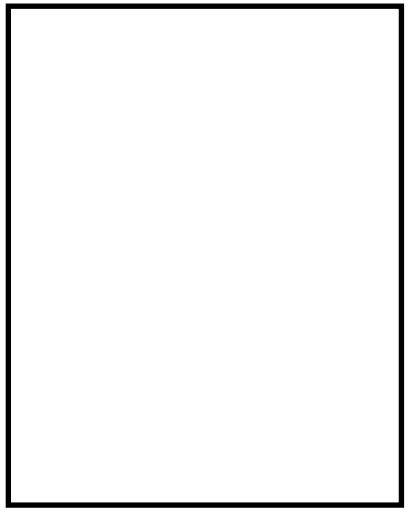
Keep memory local

Serialise communication and compute

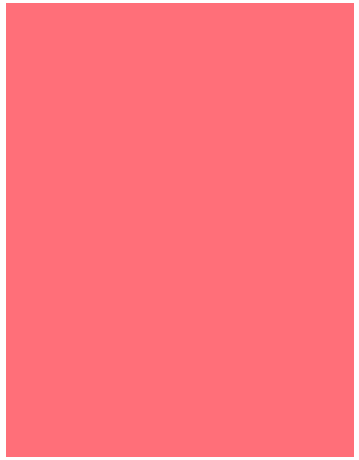
Re-compute what you can’t remember

Proximity of memory is defined by transport energy, more than latency

ALL LOGIC CHIPS ARE POWER LIMITED

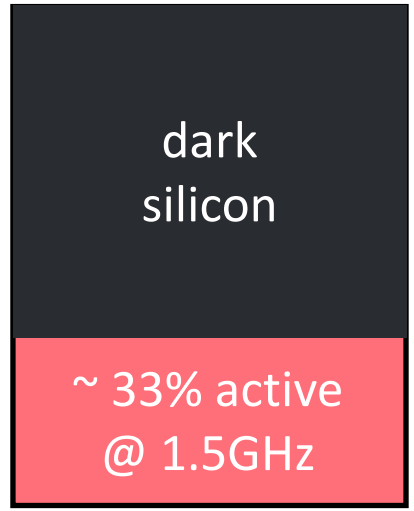


Largest manufacturable
die ~825mm^2



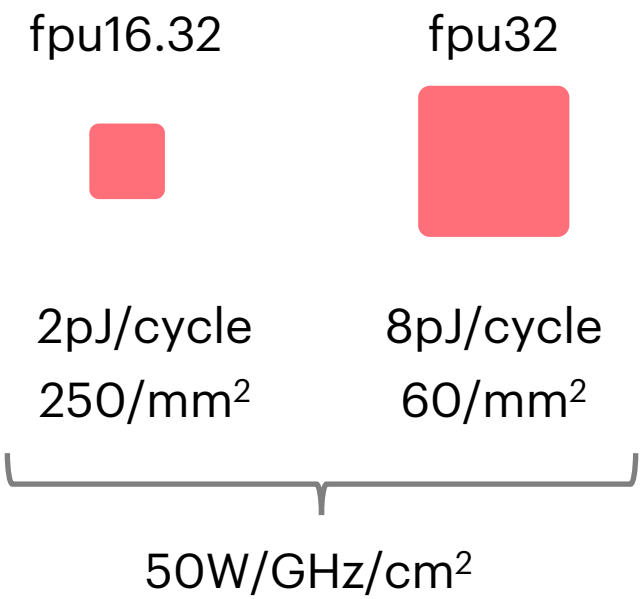
16nm 1Pflop16.32
<700mm^2

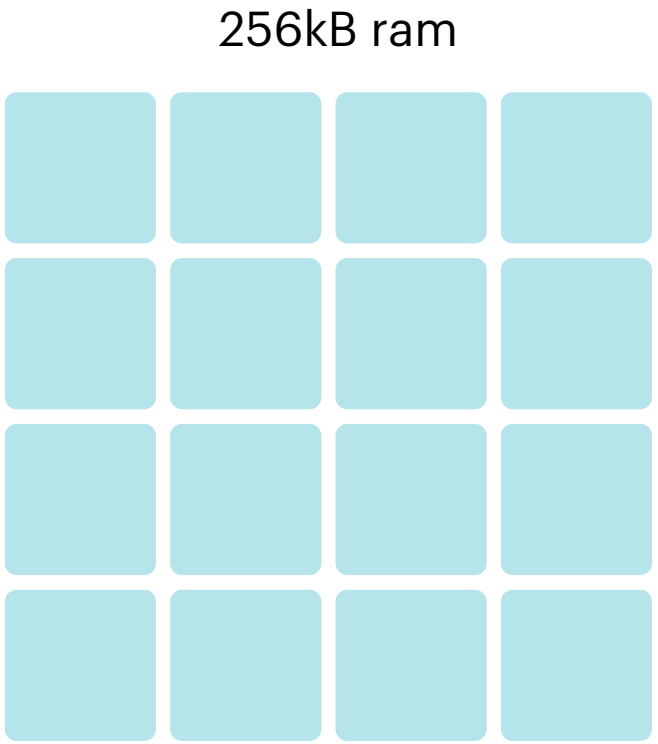
1000W



8cm^2 die @ 200W

POWER DENSITY





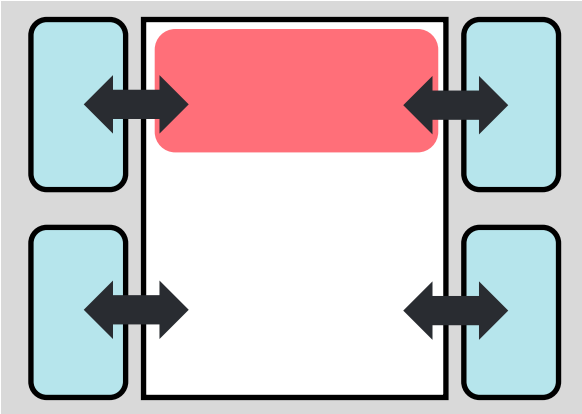
3 accesses x 8pJ/cycle

~1/5 power density of logic

MEMORY BANDWIDTH @ 240W

DRAM on interposer

180W GPU + 60W HBM2

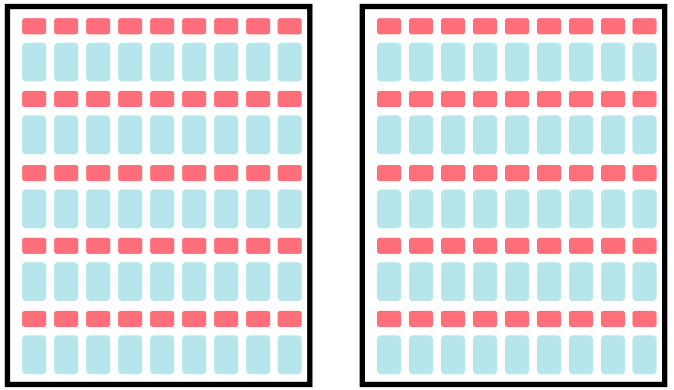


16GB @ 64pJ/B

900GB/s

Distributed SRAM on chip

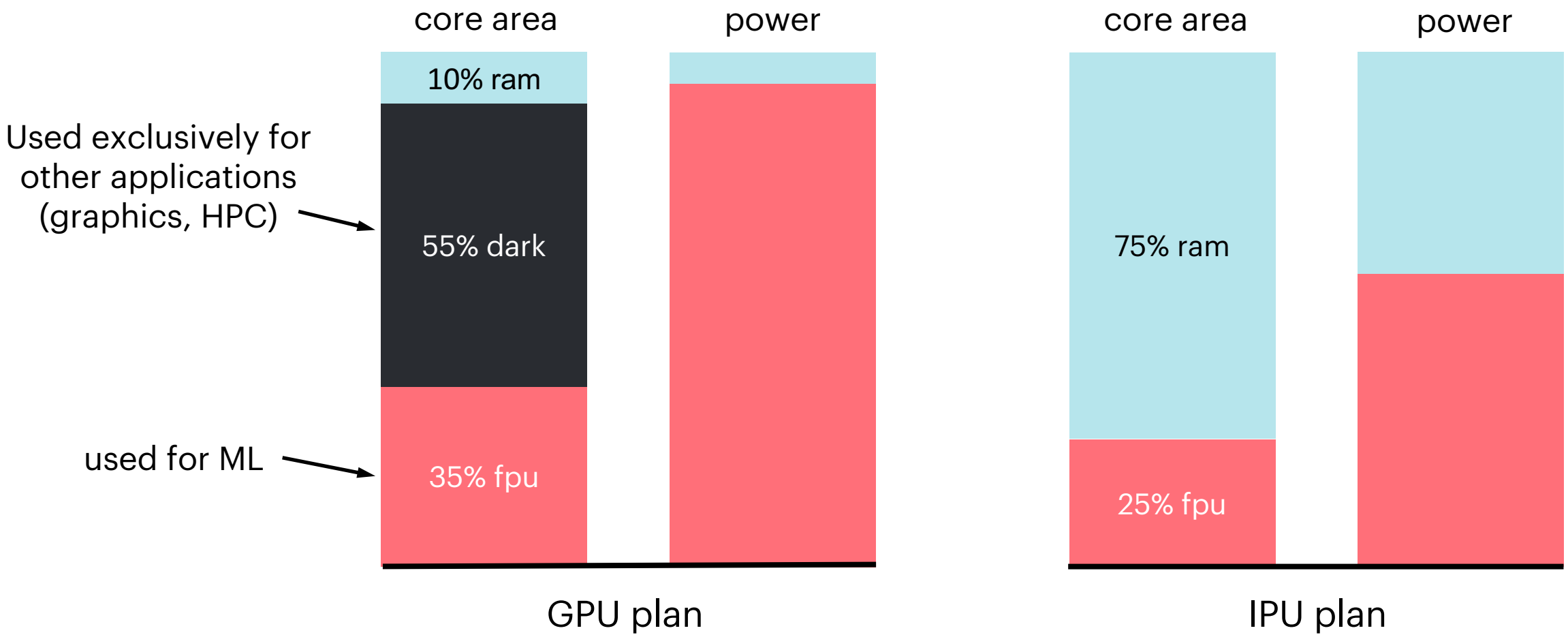
2x IPU (75W logic + 45W ram)



600MB @ 1pJ/B

90,000GB/s

GPUS USE DARK SILICON TO SERVE MULTIPLE MARKETS, IPUS USE IT TO LOCALIZE MEMORY

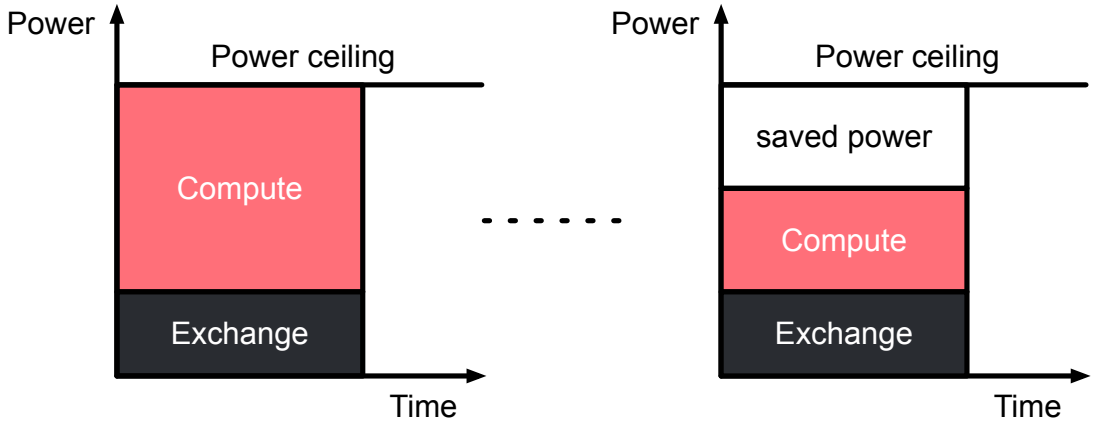


SERIALIZE COMPUTE AND COMMUNICATION

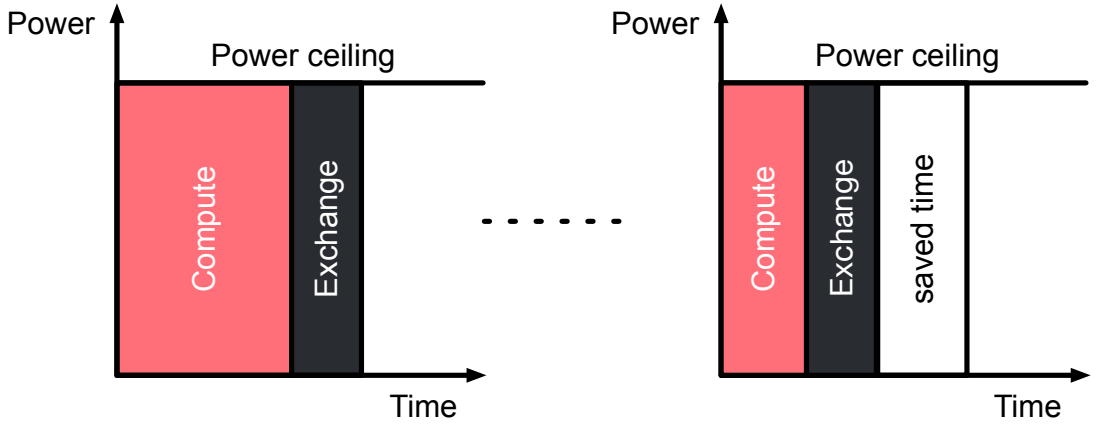
Design point

Operating point

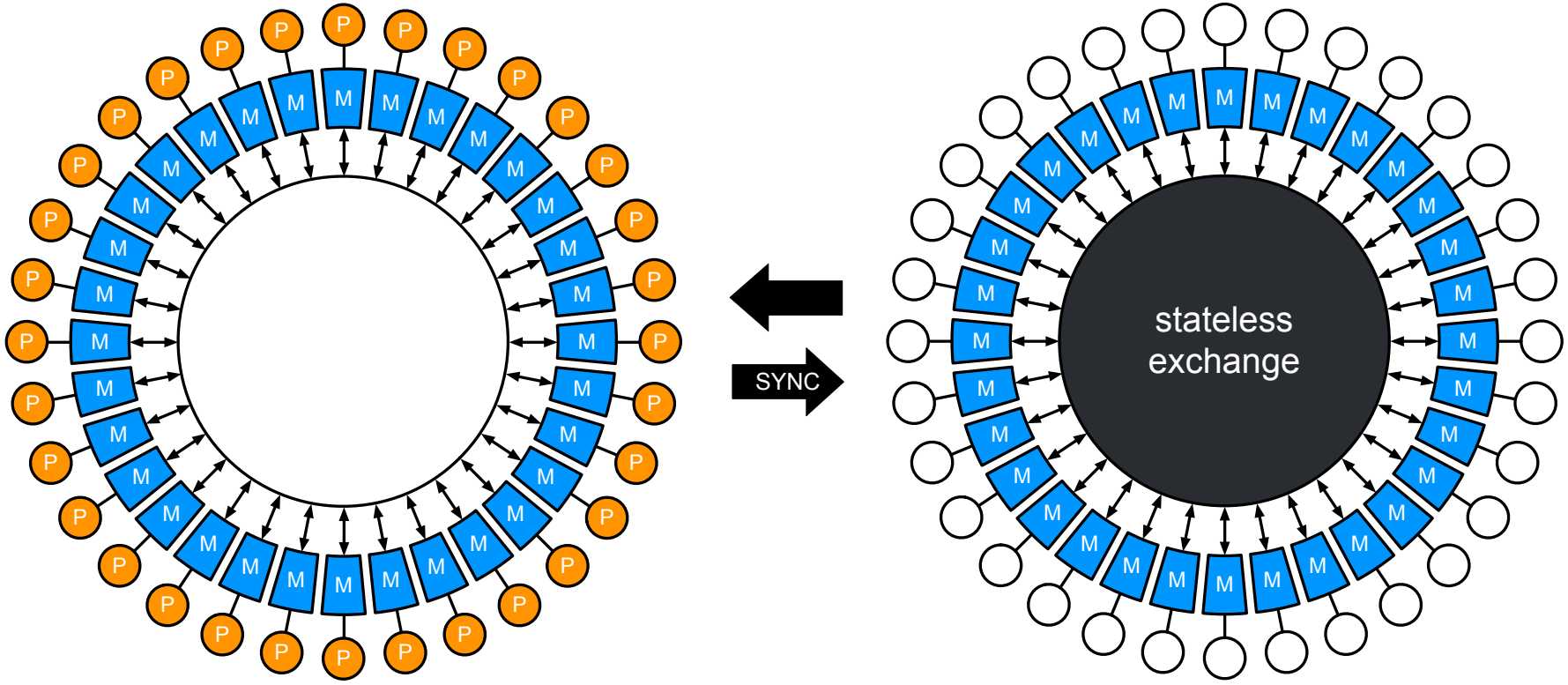
Concurrent compute and communication



Serialized compute and communication



BULK SYNCHRONOUS PARALLEL



Compute Phase

stateful codelets execute on local memory state

Exchange Phase

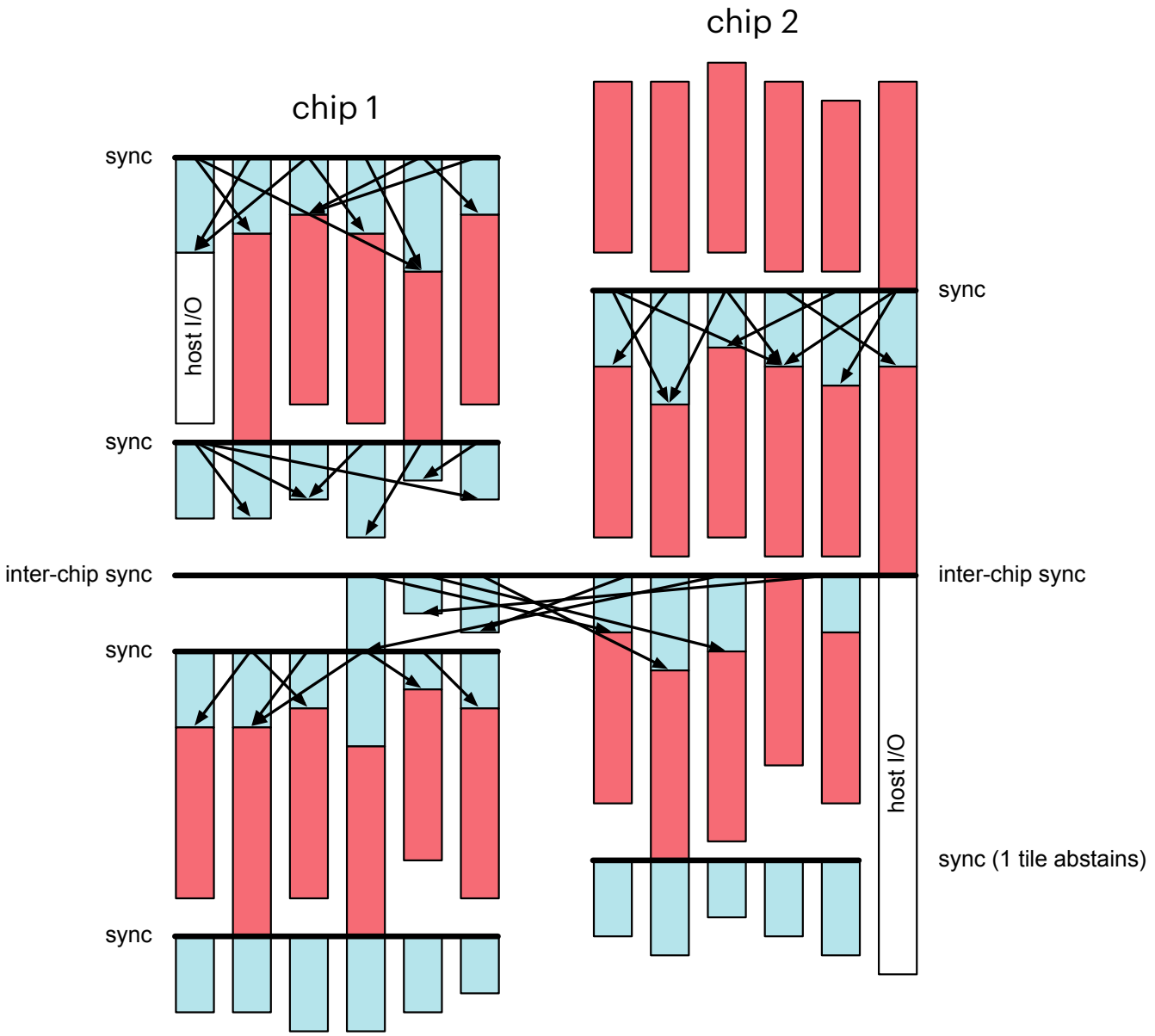
memory-to-memory data movement, no compute, no concurrency hazards

BULK SYNCHRONOUS PARALLEL

Massive parallelism with
no concurrency hazards

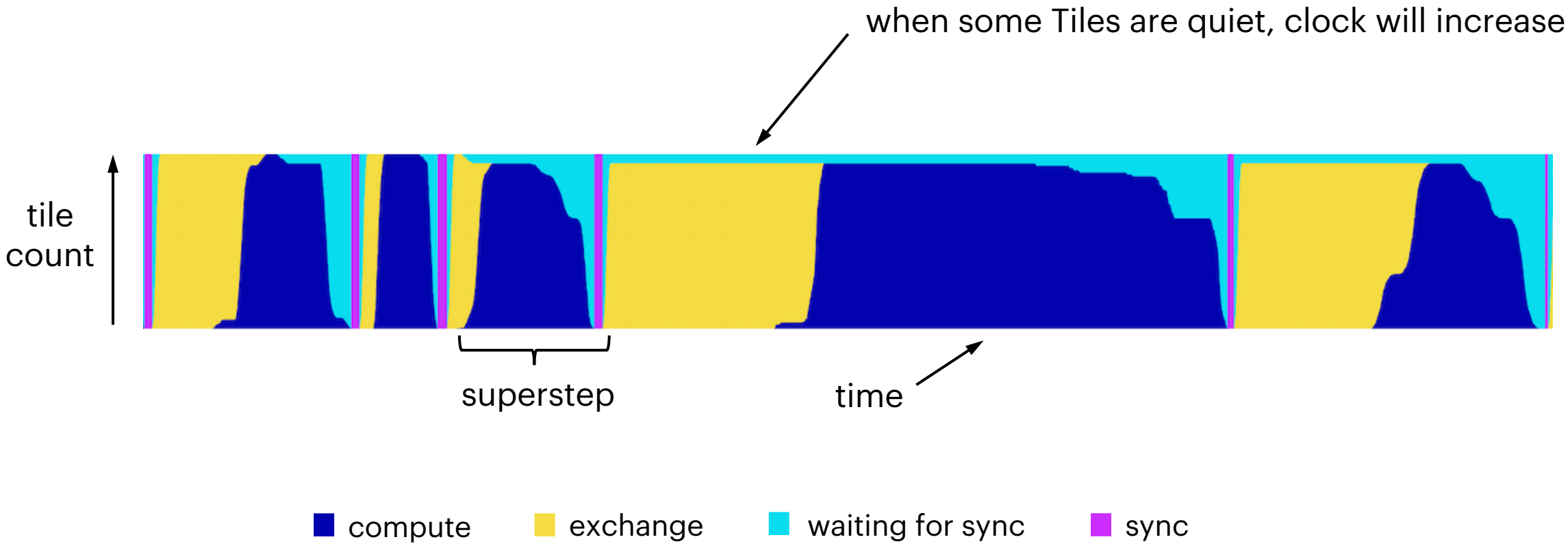
compute phase

exchange phase

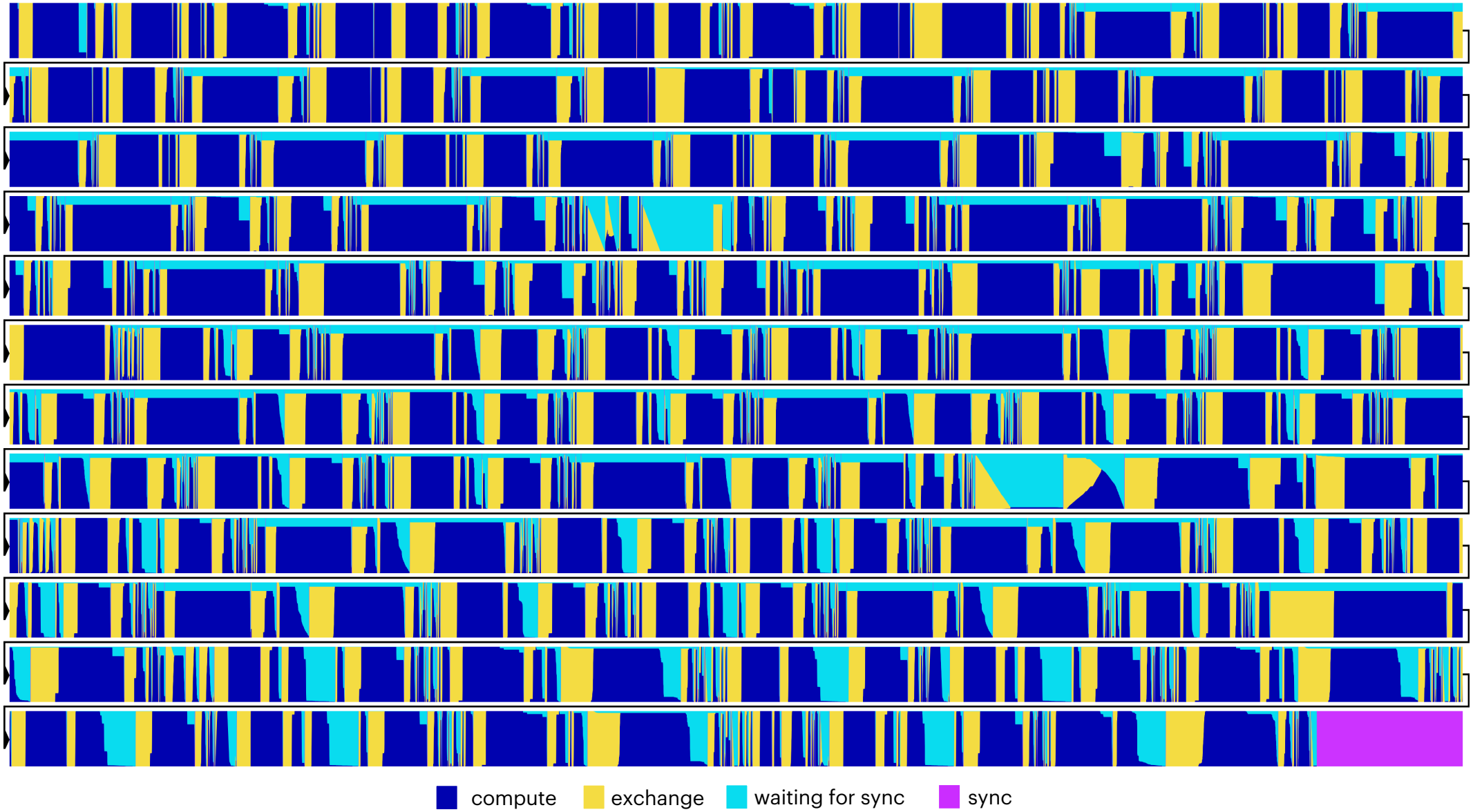


BSP EXECUTION TRACES

Poplar® summary view of workload balance

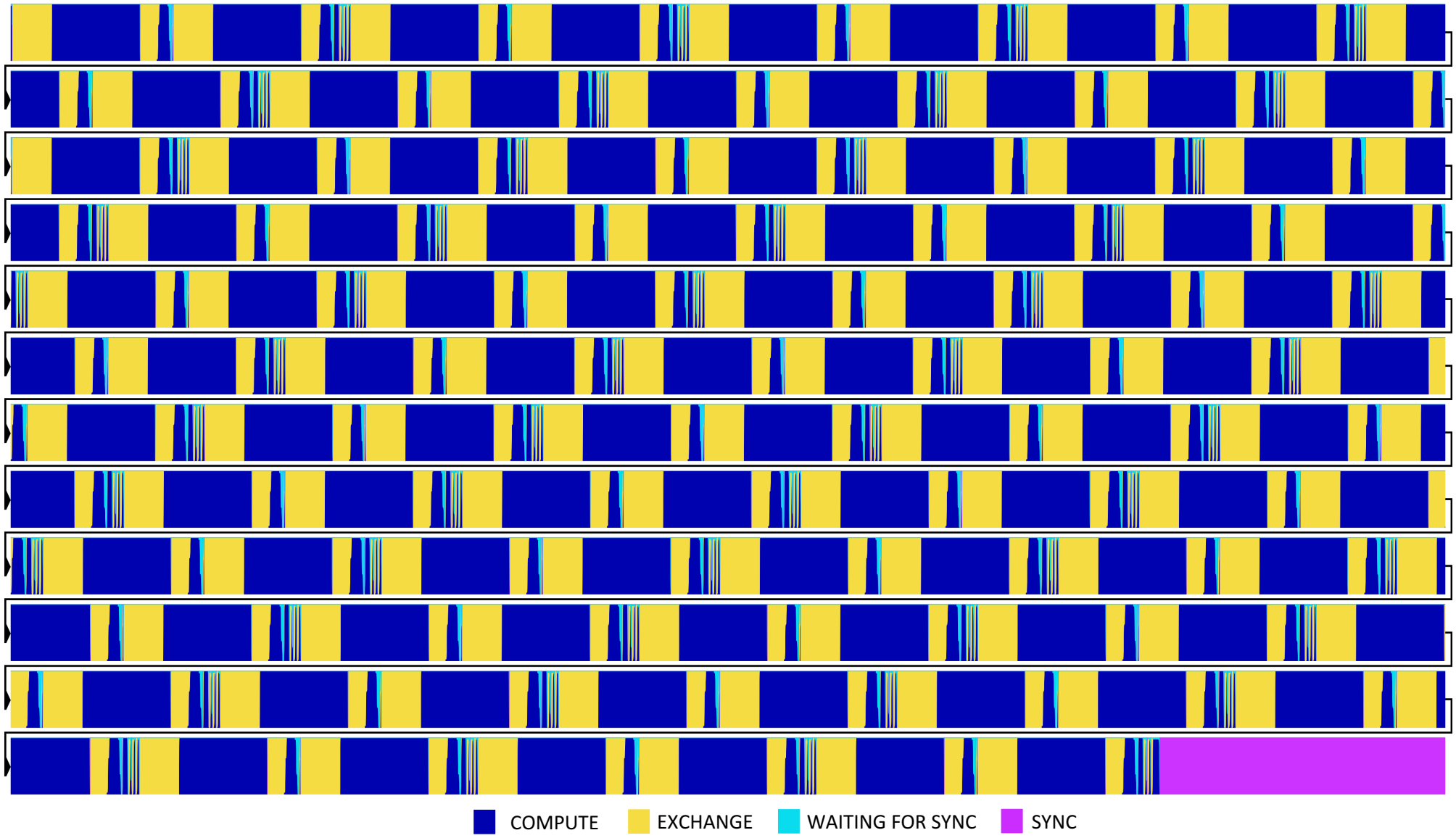


BSP TRACE: RESNET-50 TRAINING, BATCH=4



BSP TRACE: DEEPBENCH LSTM INFERENCE

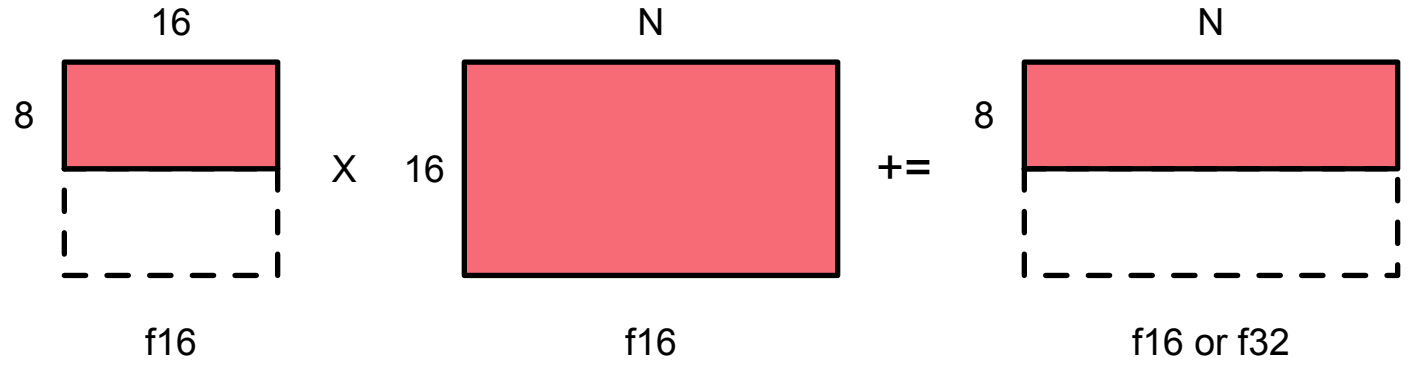
HIDDEN=1536, STEPS=50



CORE ARITHMETIC – LINEAR ALGEBRA

Float16.32 @ 64 flop/cycle
8x16xN matrix mac in 4N cycles

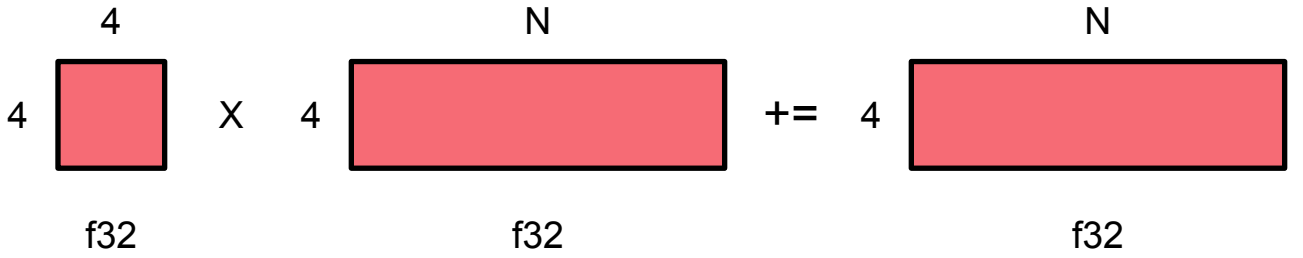
Eg. 1x1 convolve 16->8 maps
N = H.W(.batch)



(if f16, rounded from f32 intermediate every 16 terms)

Float32 @ 16 flop/cycle
4x4xN matrix mac in 2N cycles

Eg. 1x1 convolve 4->4 maps
N = H.W(.batch)



CORE ARITHMETIC – ENTROPY

IPU Tiles have instruction-level hardware for:

Generation of uniformly distributed integers

Generation of vectors of approximately Gaussian distributed floats

Random zeroing of vector elements with specified probability

Stochastic rounding of f32 accumulations to f16

MANAGING MEMORY

State of an Intelligence Processor:

Graph connectivity (model structure) ...small

Vertex code (model behaviour) ...small

Persistent data (model parameters) ...large

Ephemeral data (model activations) ...large x batch size

IPU has smaller memory than GPU, greater compute:

Use small batches per IPU

Trade compute for memory

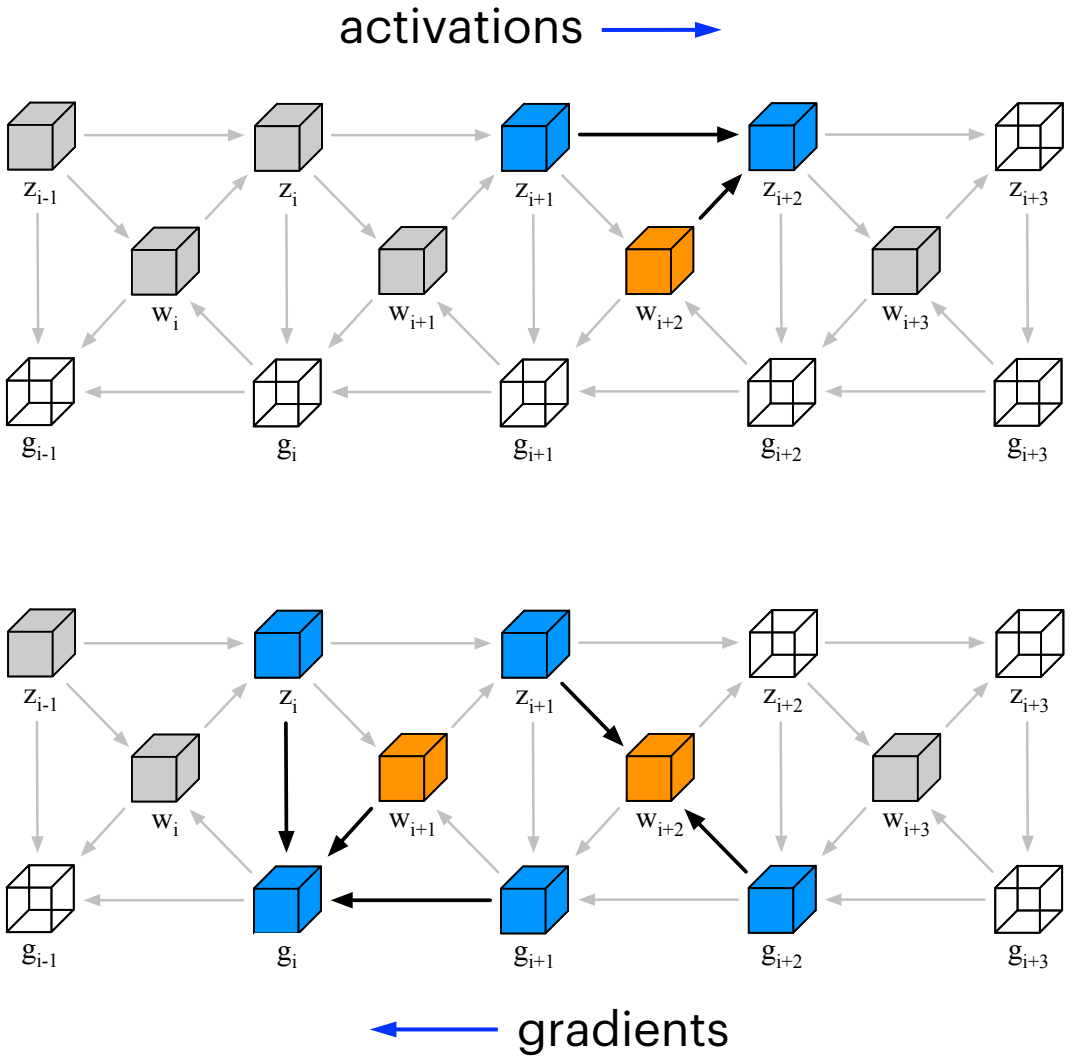
Favour memory-efficient algorithms/models

RE-COMPUTE WHAT YOU CAN'T REMEMBER

In back-propagation, most memory is consumed by storing all activations in forward pass, for gradient and weight update calculation in backward pass.

Alternatively, re-compute the activations from sparse snapshots.

Trades most storage for one repeat of forward pass compute.

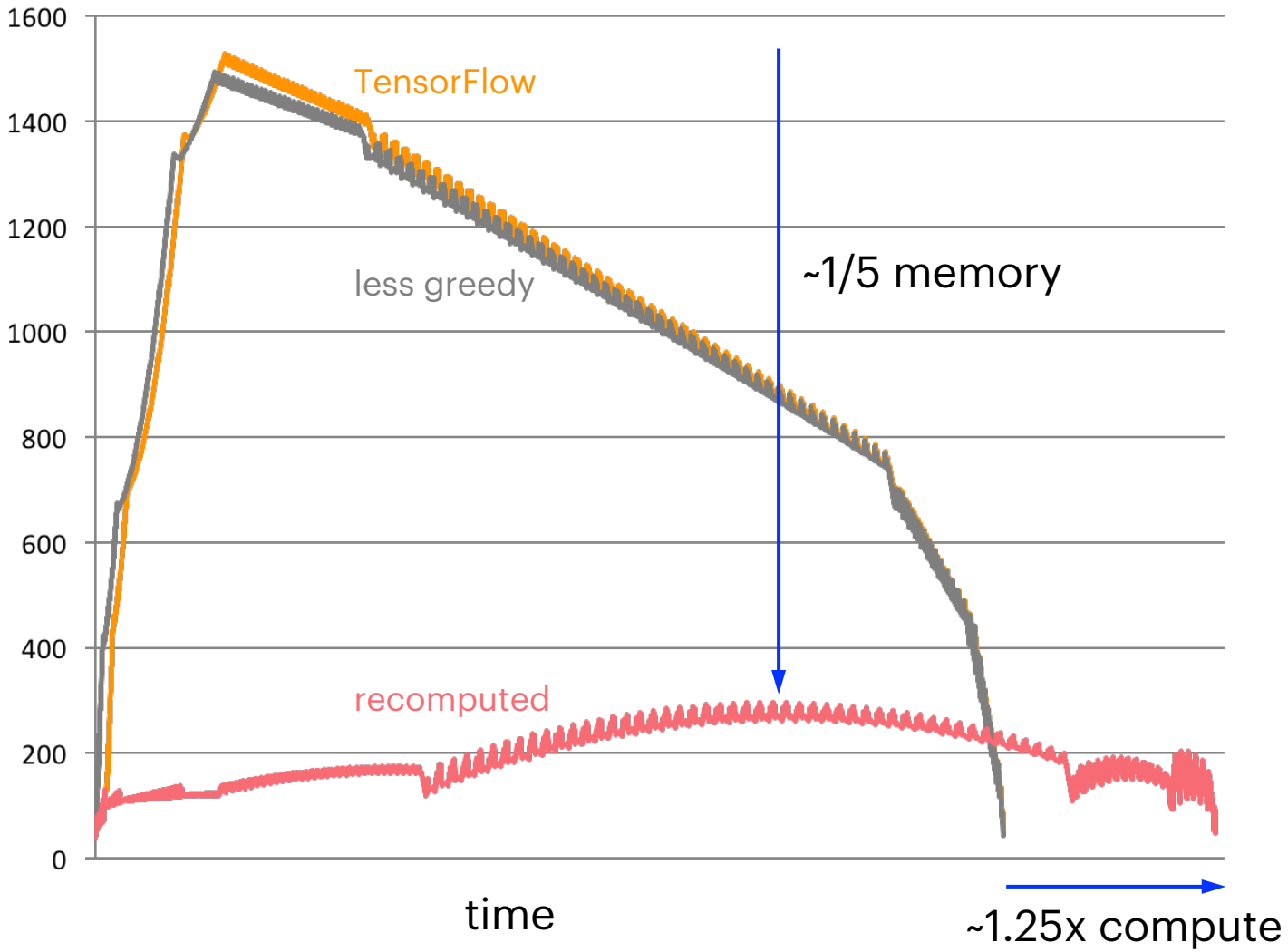


COMPUTE/MEMORY TRADE-OFF IN DENSENET-201 TRAINING

Naive strategy: memorize activations only at input of each residue block

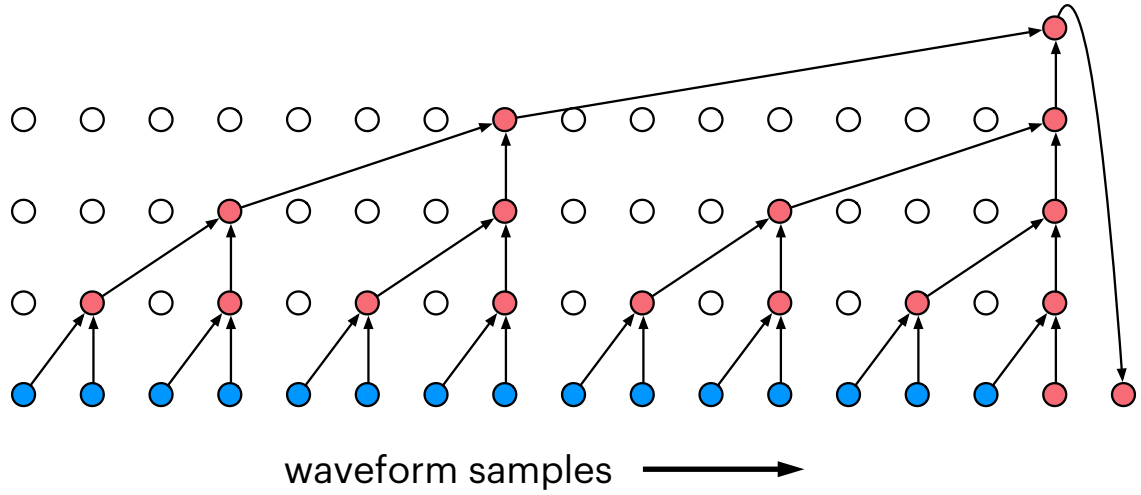
Batch=16 executing on CPU, recording total memory allocated for weights + activations.

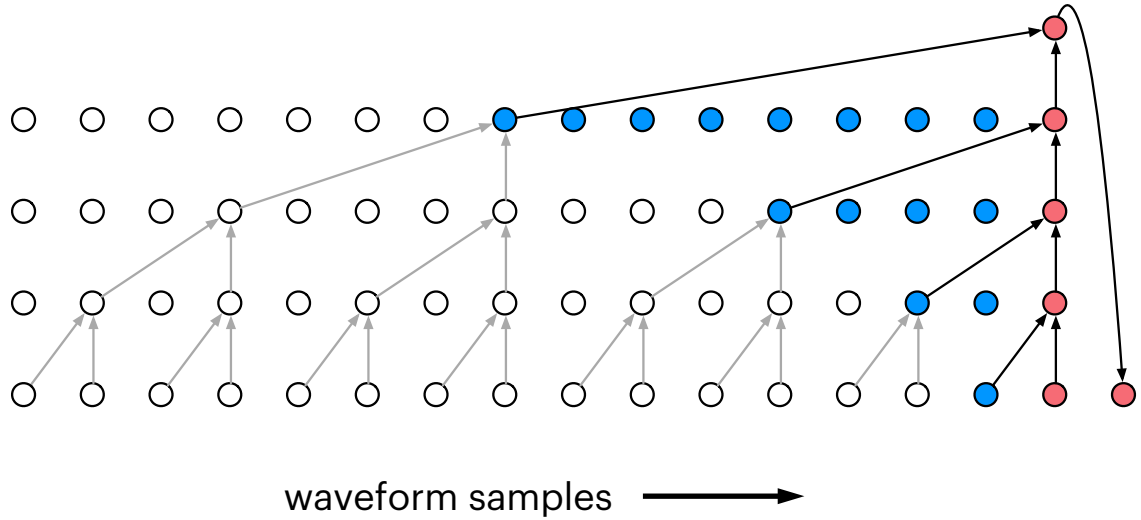
f16 weights and activations, single weight copy.



COMPUTE/MEMORY TRADE-OFF IN WAVENET INFERENCE

- memorized
- (re)computed

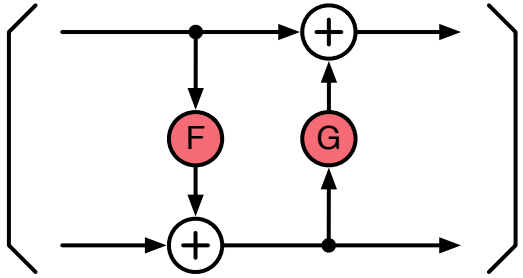




REVERSIBLE NETS

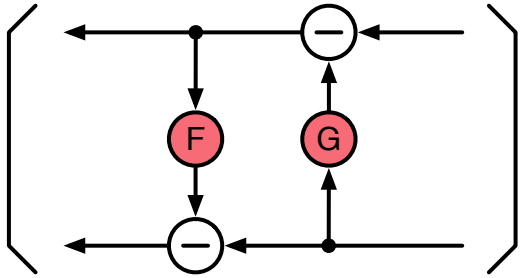
arxiv.org/abs/1707.04585

2 sets of activations, layer N



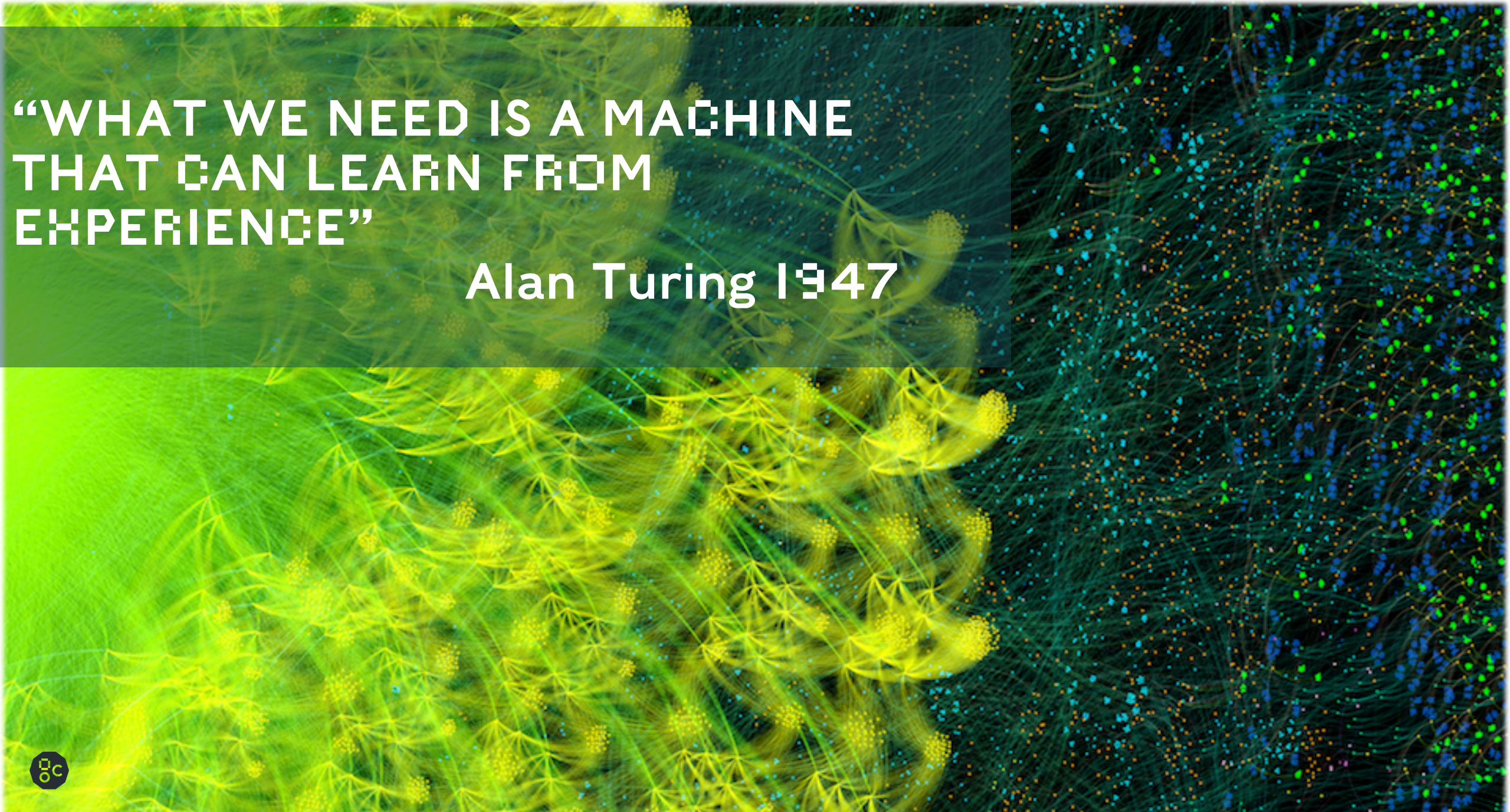
2 sets of activations, layer N+1

2 sets of activations, layer N



2 sets of activations, layer N+1

Layers can be made reversible if they don’t lose information, eg. pool or stride


“WHAT WE NEED IS A MACHINE
THAT CAN LEARN FROM
EXPERIENCE”
Alan Turing 1947