IMAGENET TRAINING IN A FEW MINUTES

BENCE TILK
AI APPLICATION ENGINEER

GRAPHCORE

Neural network visualization from POPLAR™

OUTLINE:

- What is IPU?
- What is MLPerf?
- Training ResNet50
- Enabling new architectures

GRAPHCORE

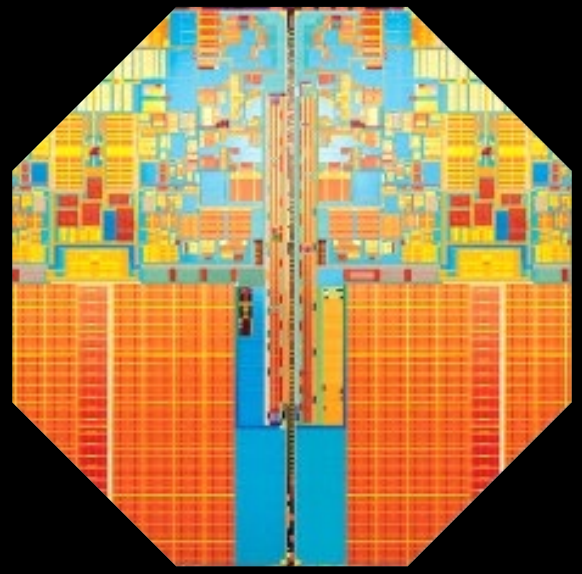
WHAT IS IPU?

MEMORY BANDWIDTH IS LIMITING PERFORMANCE

If 10x faster…

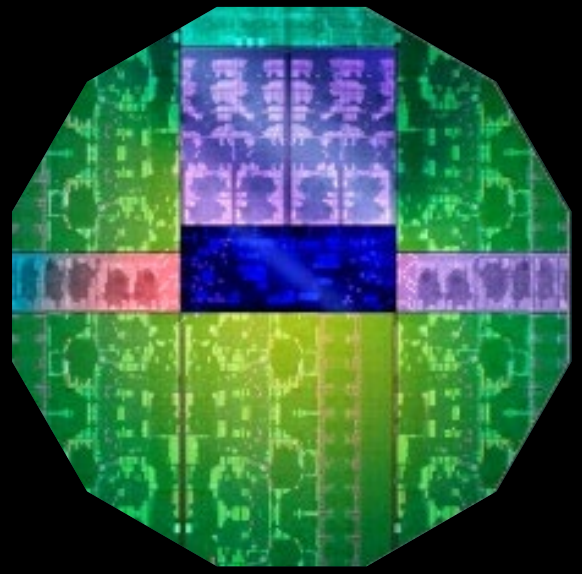
…10x faster how?

A NEW PROCESSOR IS REQUIRED



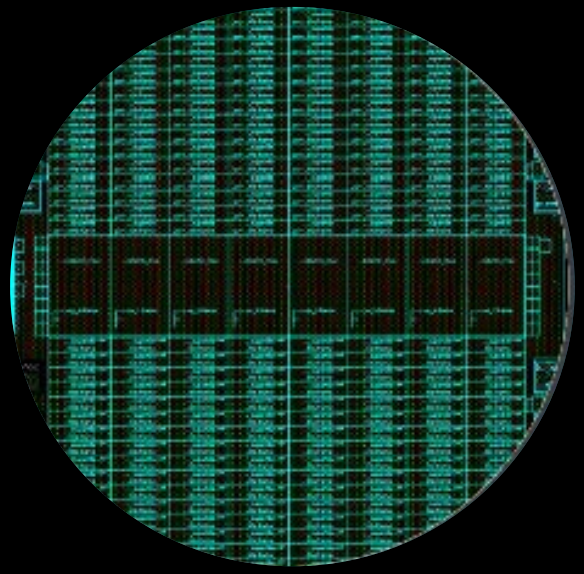
CPU

Apps and Web/
Scalar



GPU

Graphics and HPC/
Vector



IPU

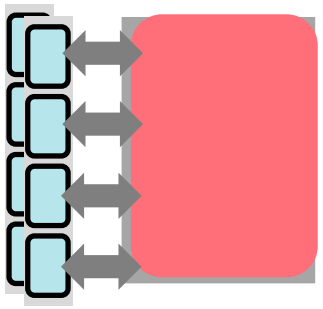
Artificial Intelligence/
Graph

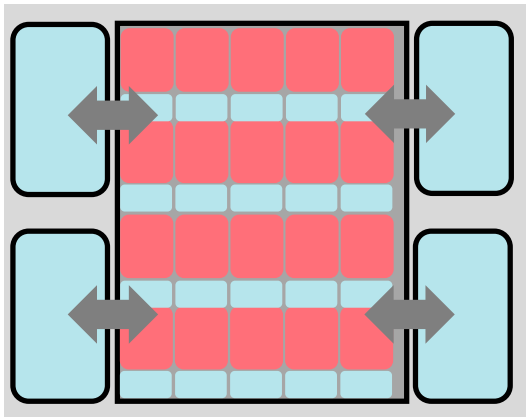
MASSIVE PARALLELISM WITH ULTRAFAST MEMORY ACCESS

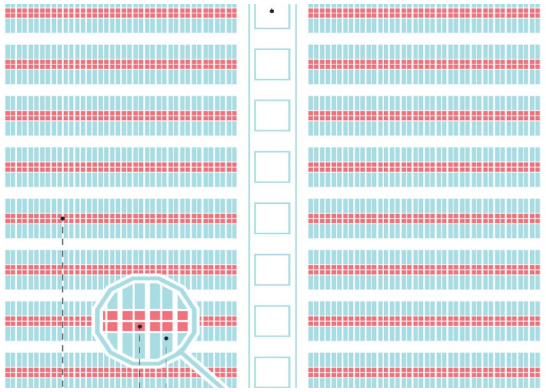
	CPU	GPU	IPU
Parallelism	Designed for scalar processes	SIMD/SIMT architecture. Designed for large blocks of dense contiguous data	**Massively parallel MIMD. High performance/efficiency for future ML trends**
Memory Access	Off-chip memory	Model and Data spread across off-chip and small on-chip cache and shared mem.	**Model & Data in tightly coupled large locally distributed SRAM**

Processor

Memory







INTRODUCING THE BOW IPU
WORLD'S FIRST 3D WAFER-ON-WAFER PROCESSOR

3D silicon wafer stacked processor

350 TeraFLOPS AI compute

Optimized silicon power delivery

0.9 GigaByte In-Processor-Memory @ **65TB/s**

1,472 independent processor cores

8,832 independent parallel programs

10x IPU-Links™ delivering 320GB/s

BOW-2000 IPU MACHINE

4 x Bow 3D Wafer-on-Wafer IPUs

1.4 PetaFLOPS AI Compute

3.6 GB In-Processor-Memory @ **260TB/s**

Up to 256 GB IPU Streaming Memory

2.8 Tbps IPU-Fabric™

Same 1U blade form factor

GRAPHCORE

WHAT IS MLPERF?

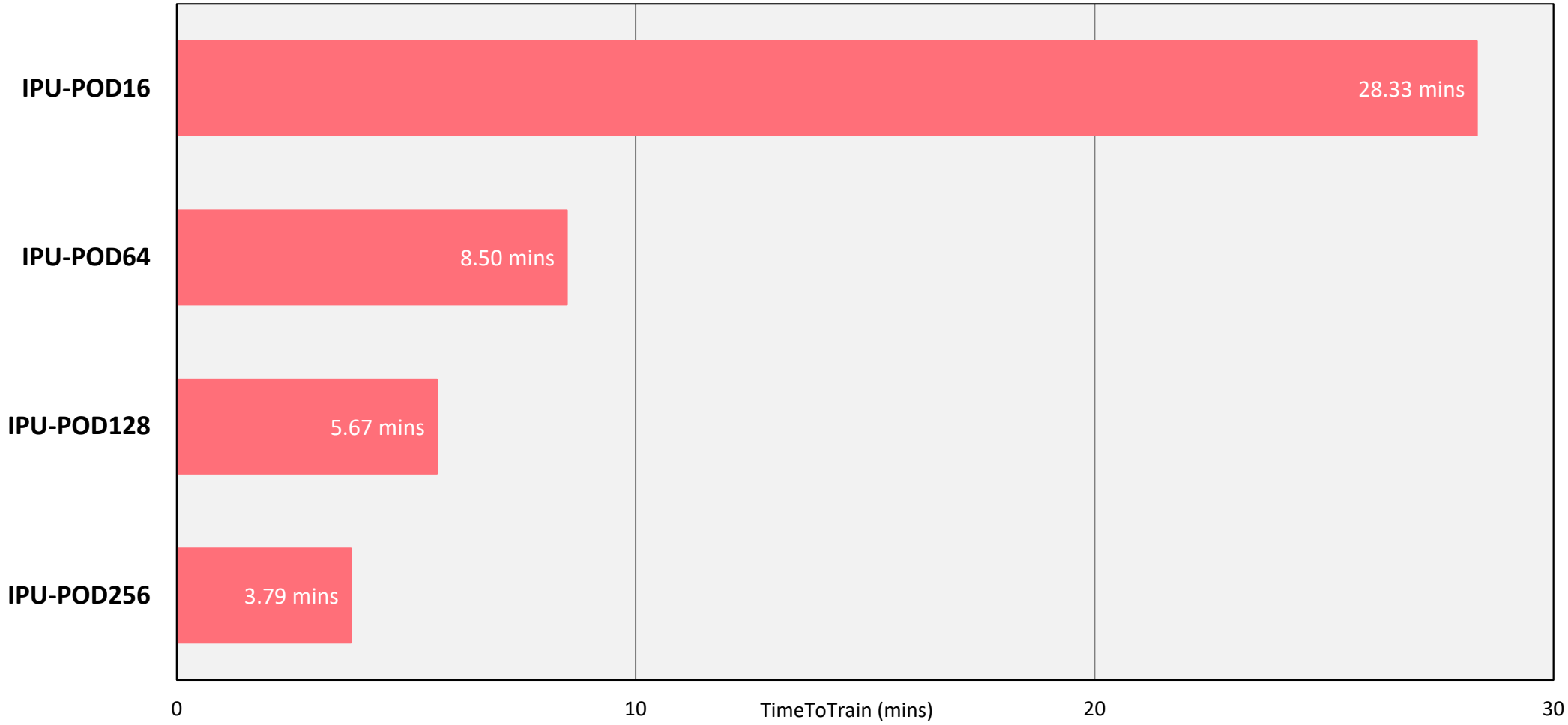
MLPERF TRAINING

MLPerf Training measures the time it takes to train machine learning models to a standard quality target in a variety of tasks including image classification, object detection, NLP, recommendation, and reinforcement learning.

Image classification

- Model:ResNet50
- Dataset: ImageNet 1k
- Target: validation accuracy >=75.9%
  - Closed: Reference implementation must be followed (SGD or LARS optimiser, learning rate schedule and norm layer is fixed)
  - Open: The training model/loop can be modified arbitrarily.

GRAPHCORE MLPERF V1.1 RESULTS FOR RESNET-50

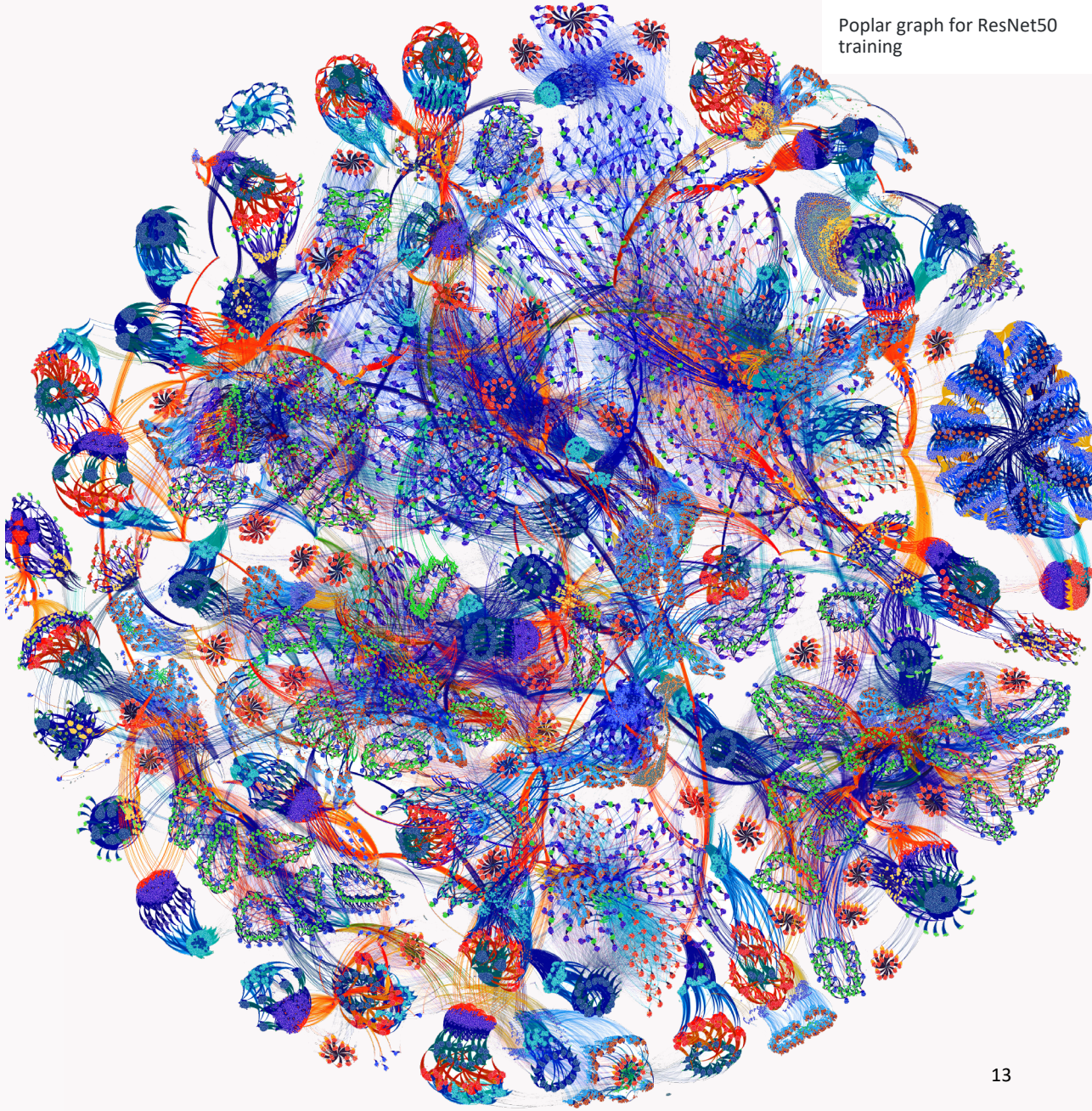


MLPerf v1.1 Training Results | MLPerf ID: 1.1-2040, 1.1-2042, 1.1-2044, 1.1-2045

TRAINING RESNET50

RESNET 50 TRAINING ON IPU

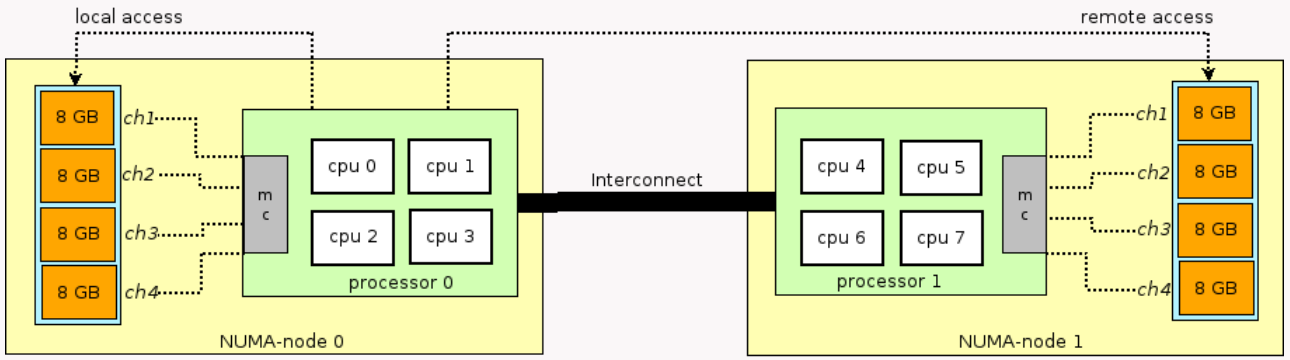
- Speed up dataloaders
- Use recomputation to maximize batch size & throughput
- Distributed Batch Norm to ensure accurate Batch Norm Statistics
- Optimise the collectives for Gradient accumulation
- Stochastic rounding
- Hyperparameter optimisation



Poplar graph for ResNet50 training

SPEED UP DATALOADER / 1

- Determine possible bottlenecks: I/O, Compute, Memory or Bandwidth
- Reduce I/O usage by caching data
- Use int8 to transfer images to the device
- Increase memory bandwidth by using multiple NUMA aware instances



See boost.org for more information

SPEED UP DATALOADER / 2

- **Use fused ops if possible and beneficial**
- Case study 1: load image and crop
    - Decode JPEG is expensive
    - Images stored by 16x16 block --> decode only the selected blocks
    - TensorFlow provide a solution: *tf.io.decode_and_crop_jpeg*
- Case study 2: fuse BatchNorm statistics and affine transformation:
    - TF solution: *tf.compat.v1.nn.fused_batch_norm*

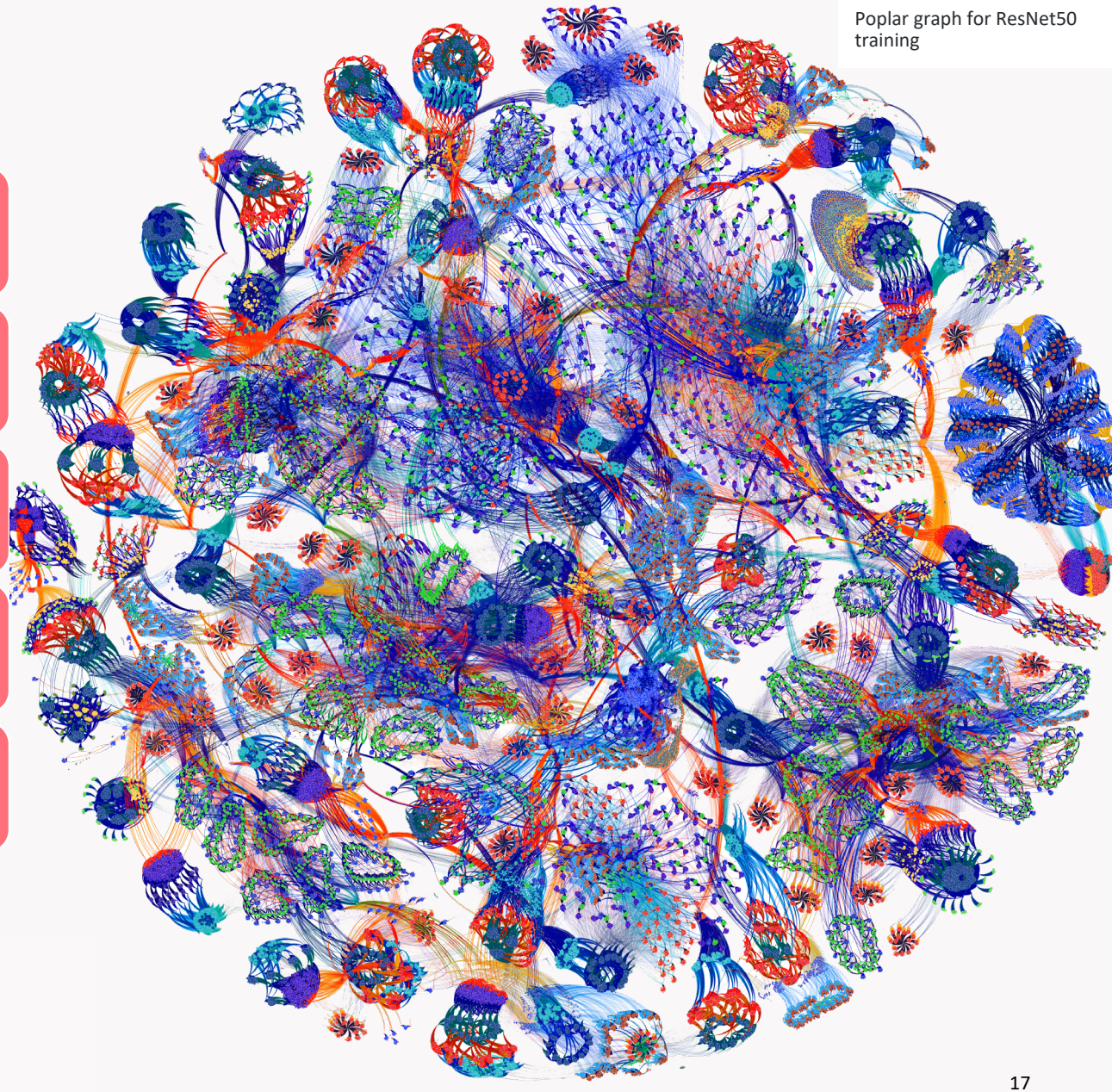
RECOMPUTATION

- IPU has really fast memory, but only 900MB / chip
- Higher compute batch size results in increased efficiency
- Recomputation of the activation
  - Lowers the memory usage
  - Increase compute costs

Tensors are stored in memory (referred to as "live") for as long as they are required. By default, we need to store the activations from the forward pass until they are consumed during backpropagation. We can reduce this liveness, and therefore the memory requirements, by recomputing the activations.

DISTRIBUTED BATCHNORM

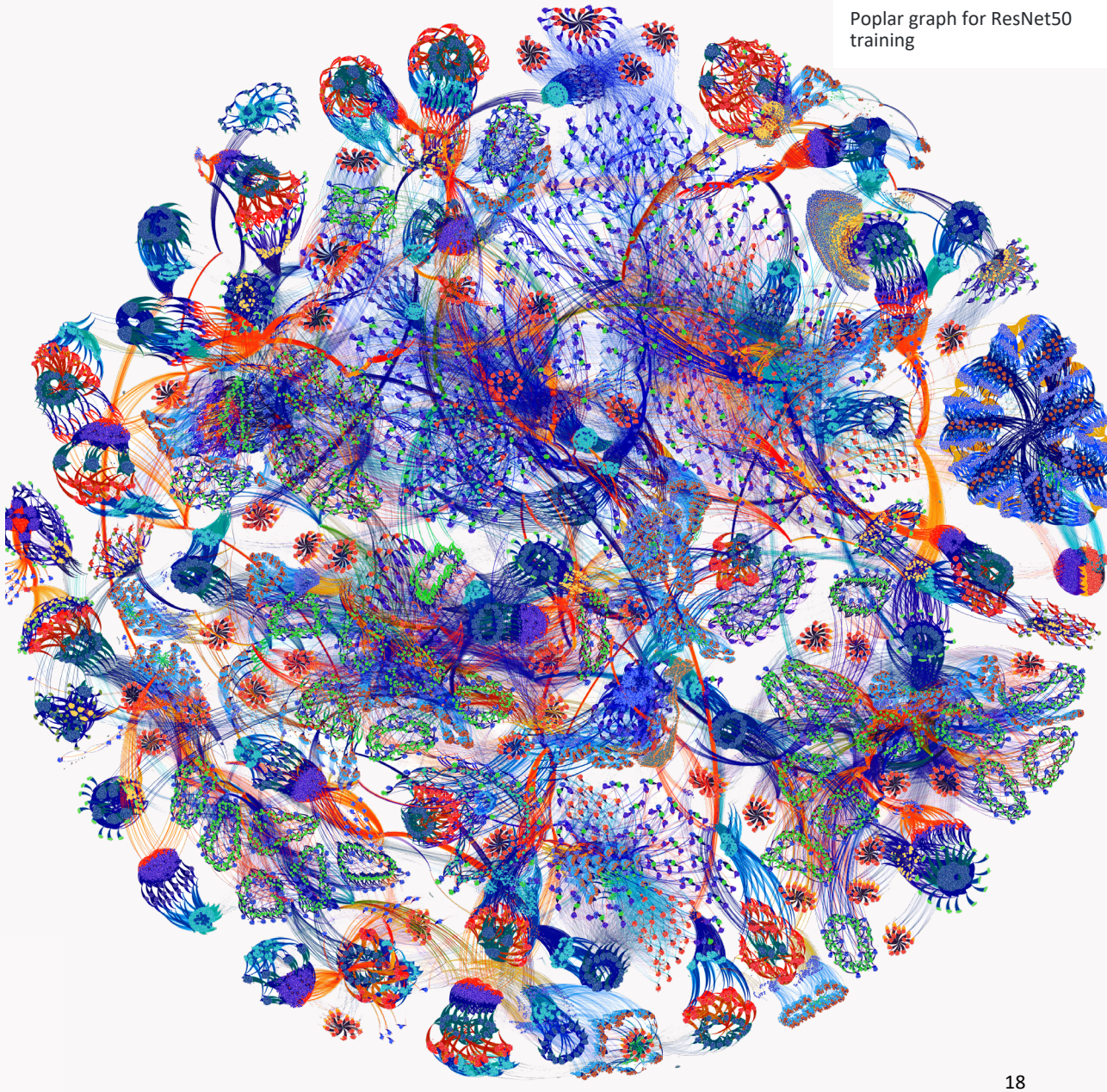
- Limited memory results in limited compute batch size
- BatchNorm layers work well with big batch sizes
- Sync the statistics between multiple devices → Distributed BatchNorm
- Communication overhead introduced
- Less epoch to convergence



Poplar graph for ResNet50 training

COMMUNICATION PATTERN RECONSIDERATION

- Weight update requires mean reduction across replicas
- Communicating each tensor between replicas 1 by 1 is slow. Need to wait the latency between the IPUs for each layer.
- The solution is bulk communication, merging the tensors for the reduce operation
- This can double the throughput with 100s of IPUs



Poplar graph for ResNet50 training

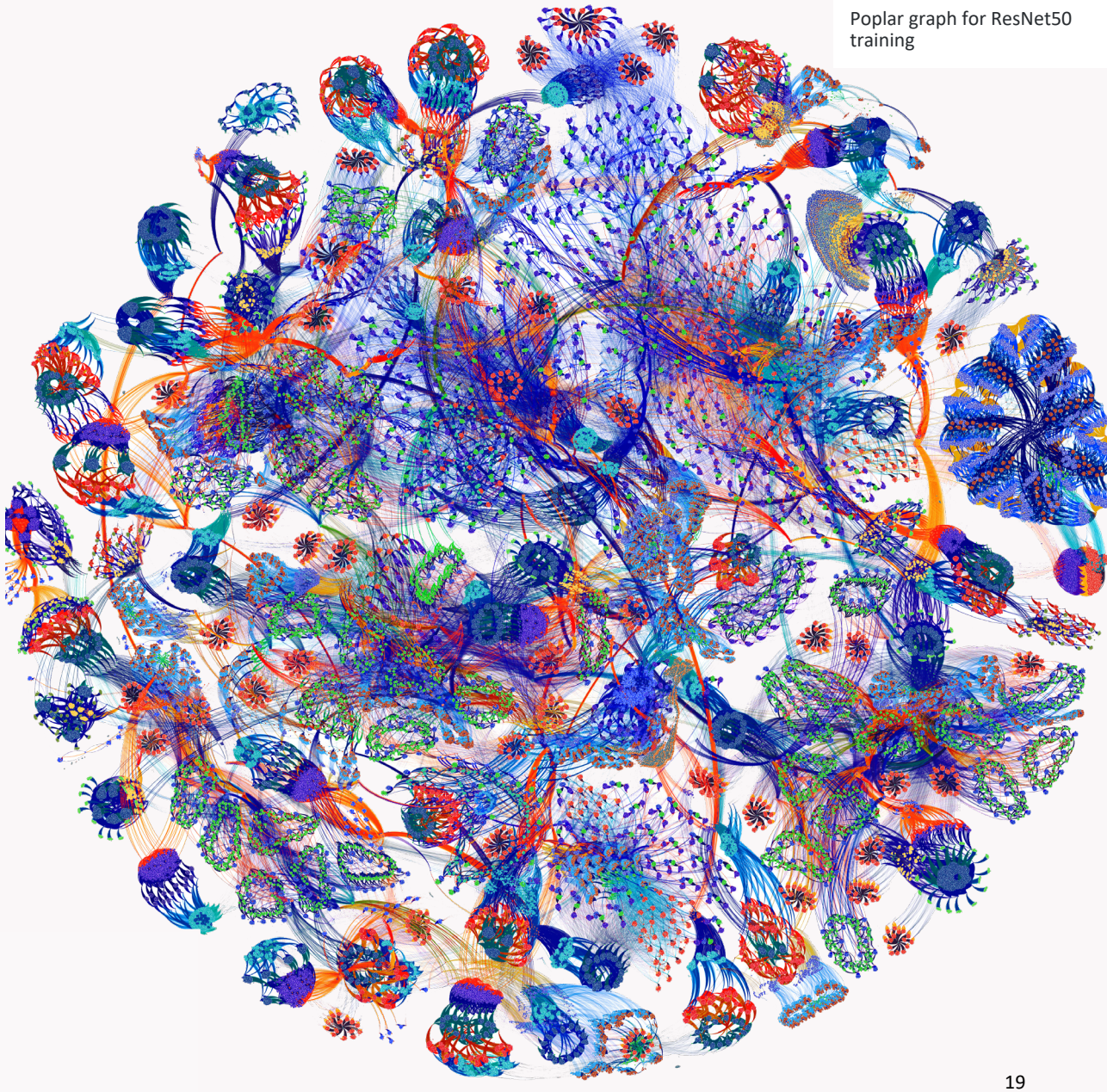
WHAT IS STOCHASTIC ROUNDING?

- 2 bits and represent numbers [0..1]

- Rounded to 0.5
- Stochastic rounding:
  - Round to 0.25 (P=0.4)
  - Round to 0.5 (P=0.6)

Motivation: add 0.4 10 times:

- With SR: expected value is 3.5 = 6*0.25+4*0.5
- Without SR: 10x 0.5 = 5

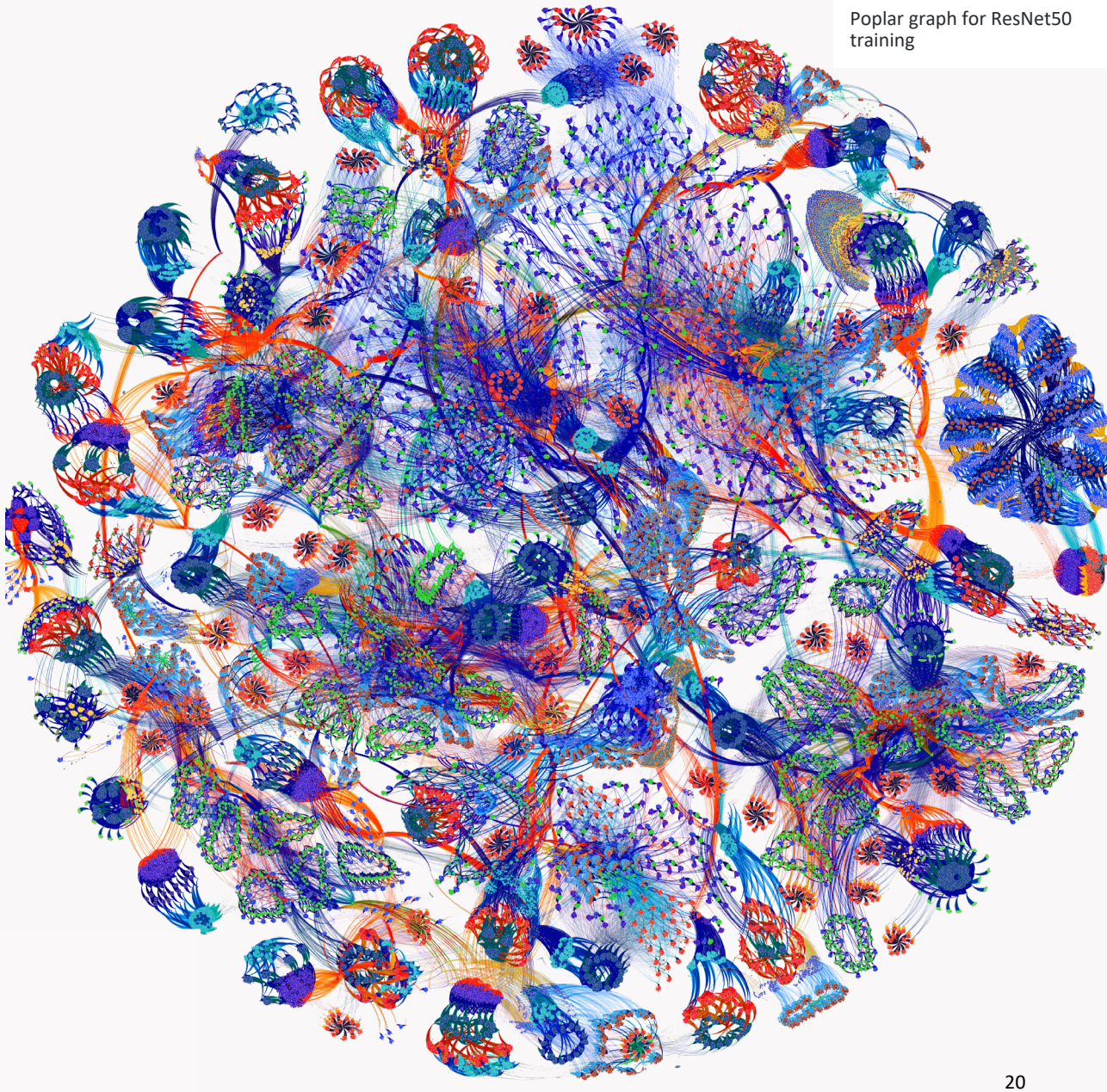


Poplar graph for ResNet50 training

STOCHASTIC ROUNDING

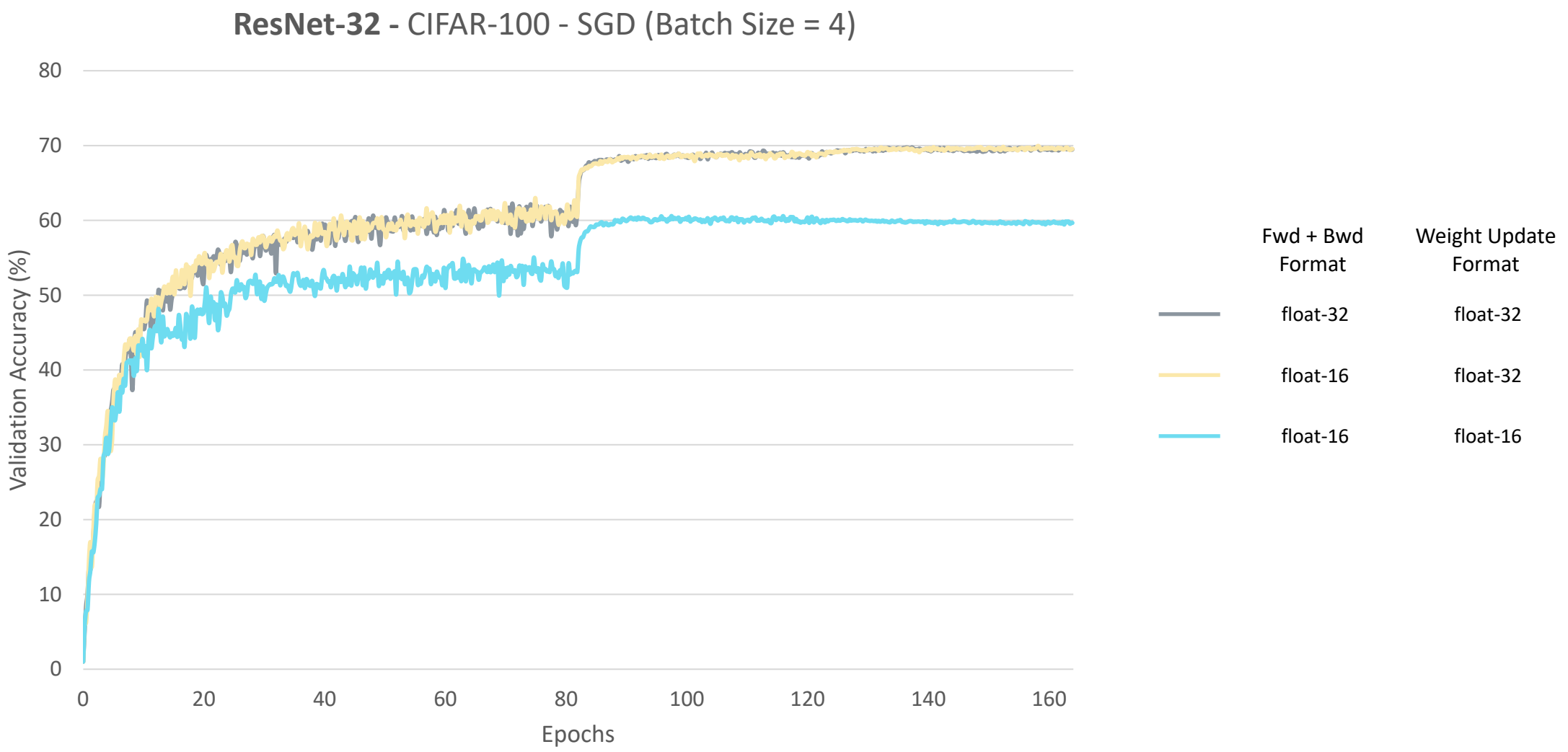
- Recipe for mixed precision training:
  - Activations in fp16
  - **Master weights in fp32**
  - Norm layer and loss in fp32

- SR can help to escape from stucked weight state if gradients underflow.
- It is possible to train ResNet50 with fp16 master weights.

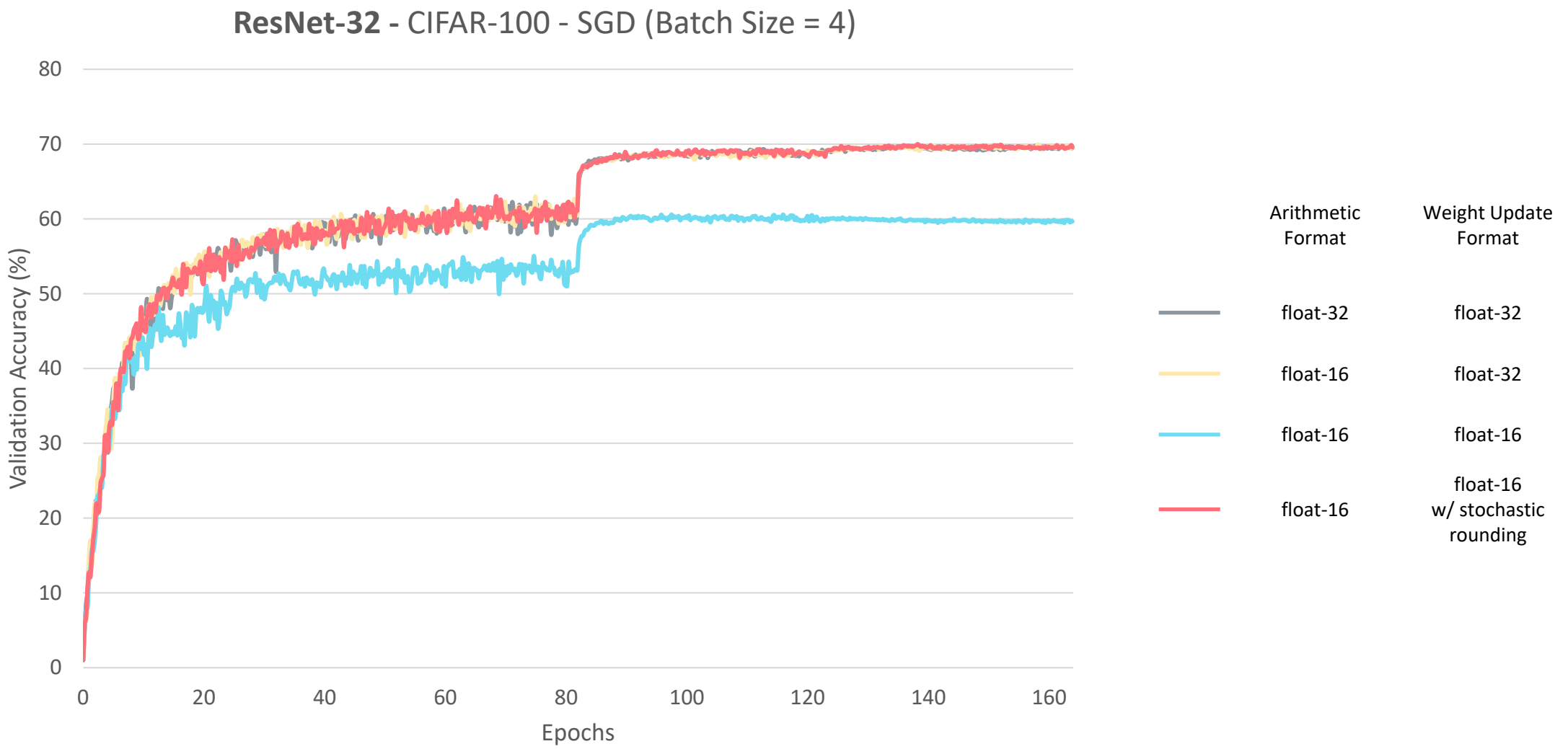


Poplar graph for ResNet50 training

WITHOUT STOCHASTIC ROUNDING

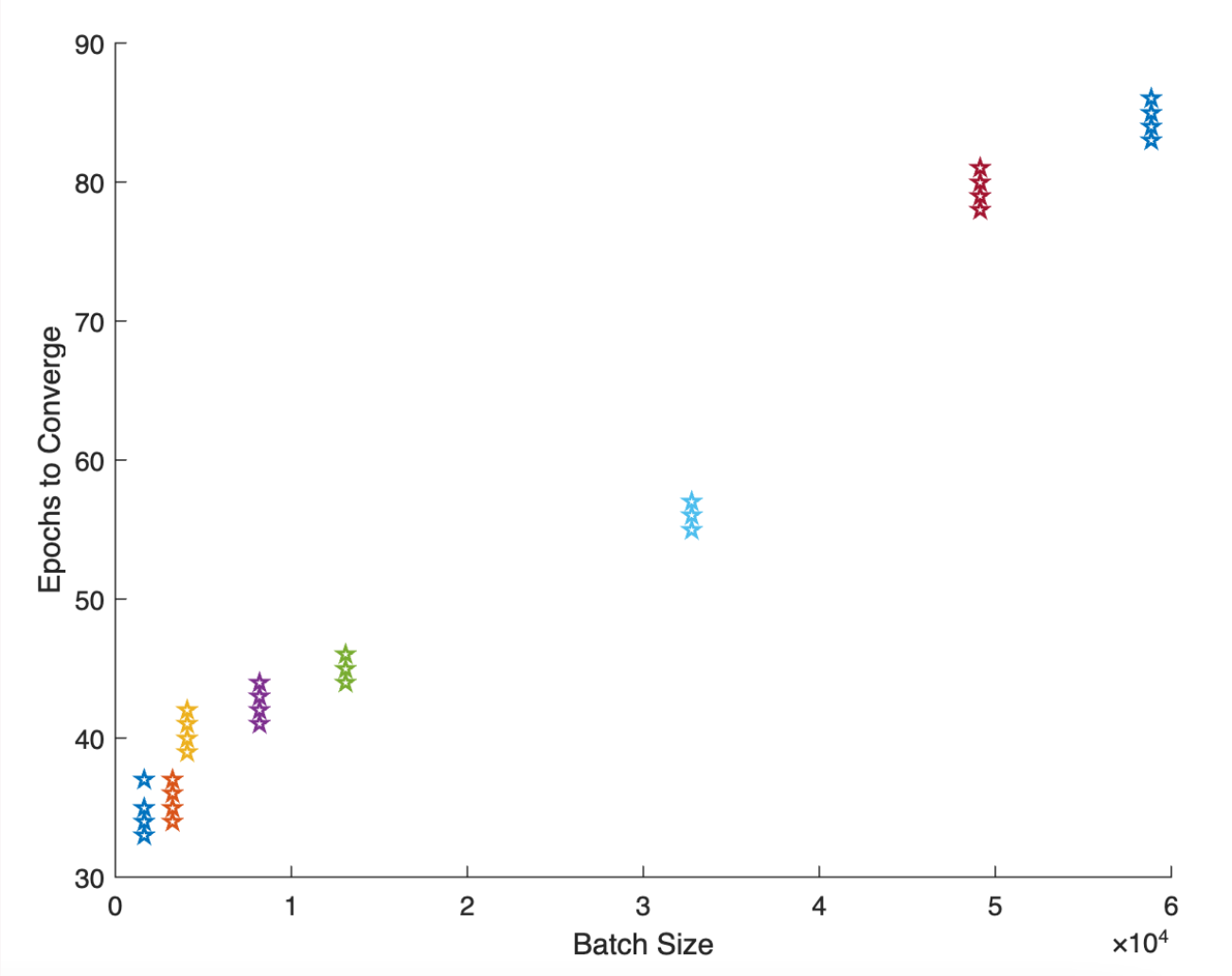


WITH STOCHASTIC ROUNDING



HYPERPARAMETER SEARCH

- Most important metric is time to train
- Tradeoff between throughput and #epochs
- Most important parameter is global batch size
- Learning rate also as important as usual



Source: https://github.com/mlcommons/logging/blob/master/mlperf_logging/rcp_checker/training_1.1.0/rcps_resnet.json

ENABLING NEW ARCHITECTURES

WHAT MIGHT LIMIT THE PERFORMANCE OF A MACHINE INTELLIGENCE PROCESSOR?

Compute

Rate of arithmetic at low precision

Sparsity in Data

Granularity of functional units, Rate of unique address calculations for accesses

Memory Access

Bandwidth and latency of accesses to memory

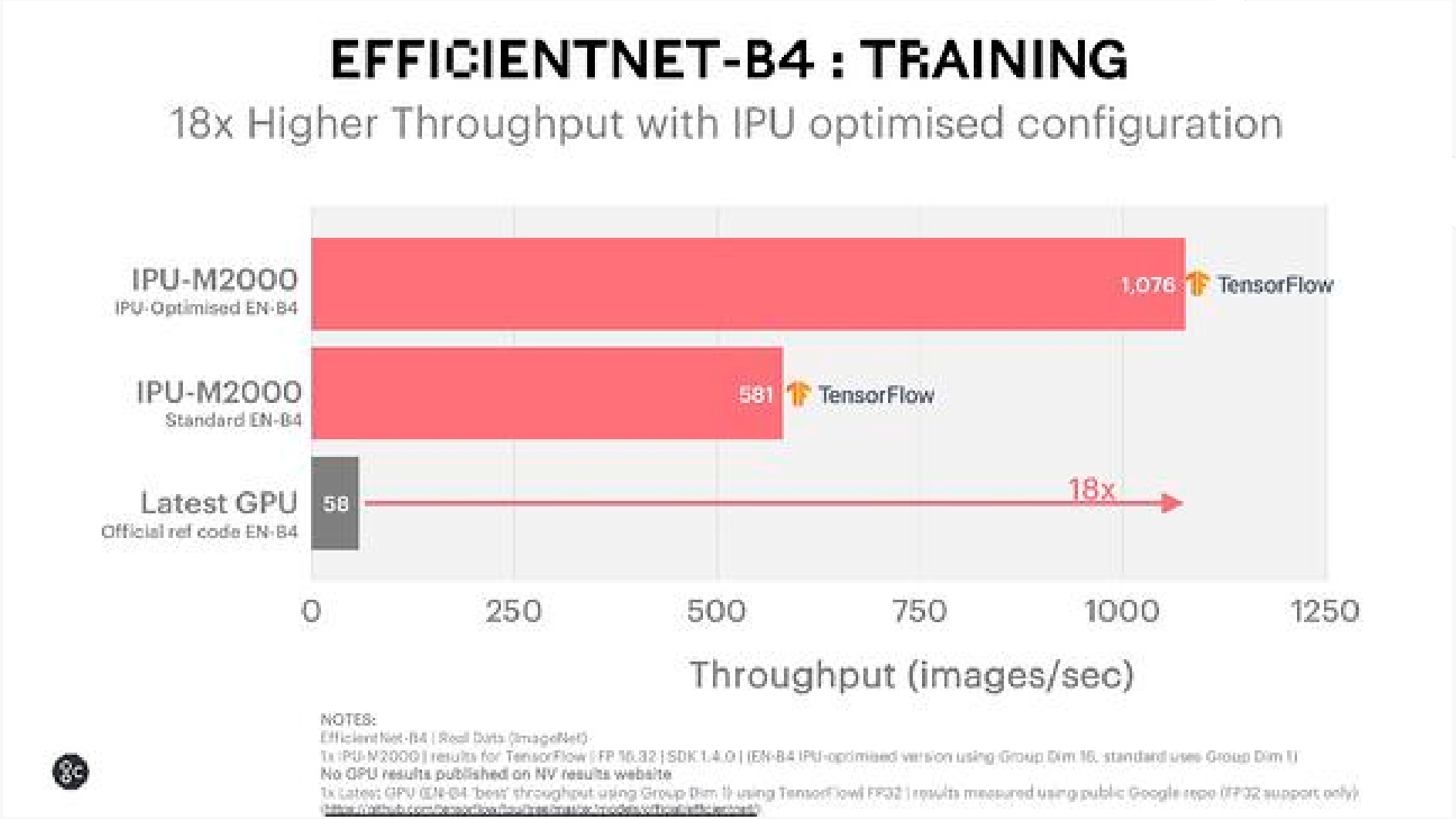
CASE STUDY: EFFICIENTNET

- The model's FLOP count is low
- Requires unregular memory access pattern
- Compute / memory transfer ratio is low

- If you know the IPU tile internals it can be further optimized
- Increase convolution groups from 1 to 16 to fully utilise compute unit
- The expansion rate can be decreased to provide the same accuracy(FLOP).

More information at https://www.graphcore.ai/posts/how-we-made-efficientnet-more-efficient

CASE STUDY: EFFICIENTNET



THANK YOU

Bence Tilk
bencet@graphcore.ai