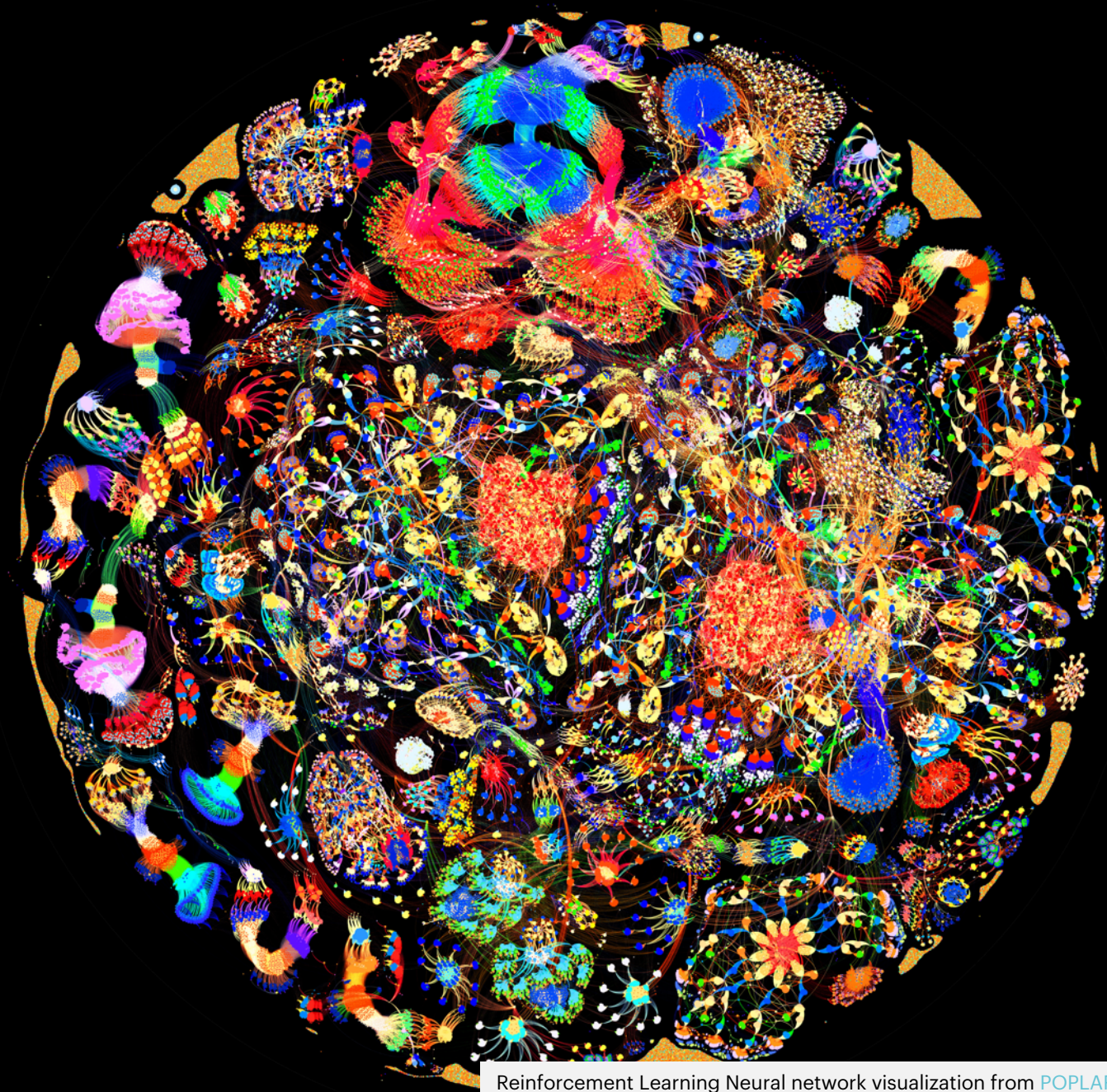
GRAPHCORE

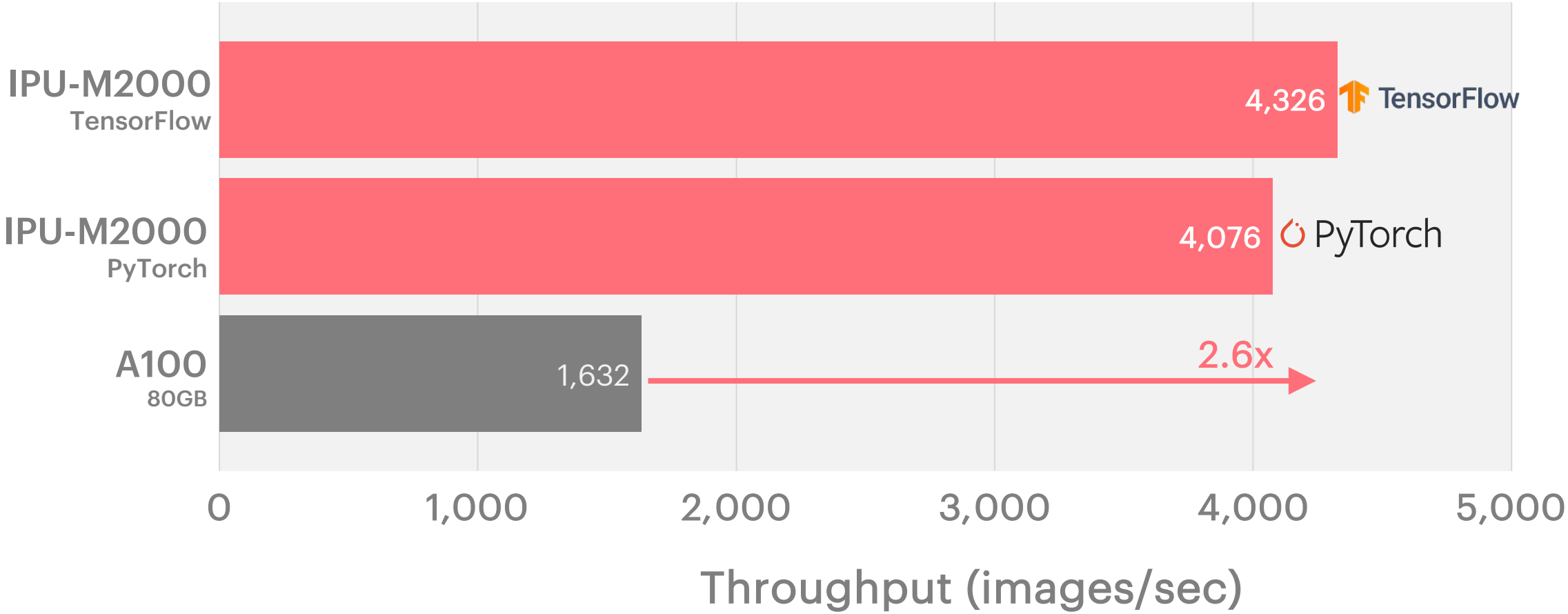
SDK 1.4 BENCHMARKS



Reinforcement Learning Neural network visualization from POPLAR™

RESNET-50 : TRAINING

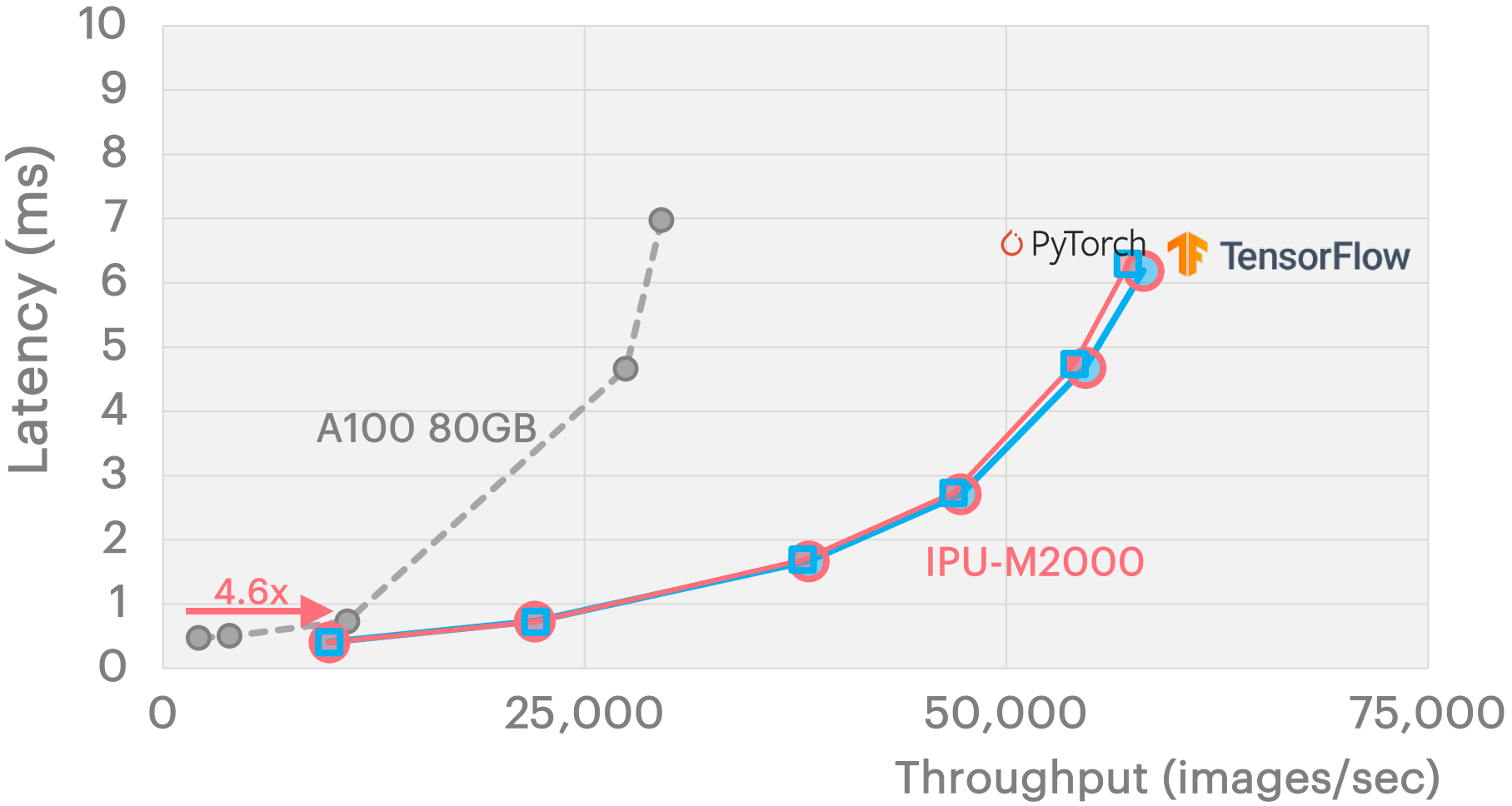
2.6x Higher Throughput



NOTES:
ResNet-50 v1.5 Training Throughput | ImageNet2012 Dataset
1x IPU-M2000 | FP 16.16 | SDK 1.4.0 | TensorFlow Batch Size 1024 | PyTorch Batch Size 1036
1x A100 (A100-SXM4-80GB) in DGX A100 Platform using TensorFlow | Mixed Precision | Batch Size 256
A100 results published by NVIDIA (https://developer.nvidia.com/deep-learning-performance-training-inference)

RESNET-50 : INFERENCE

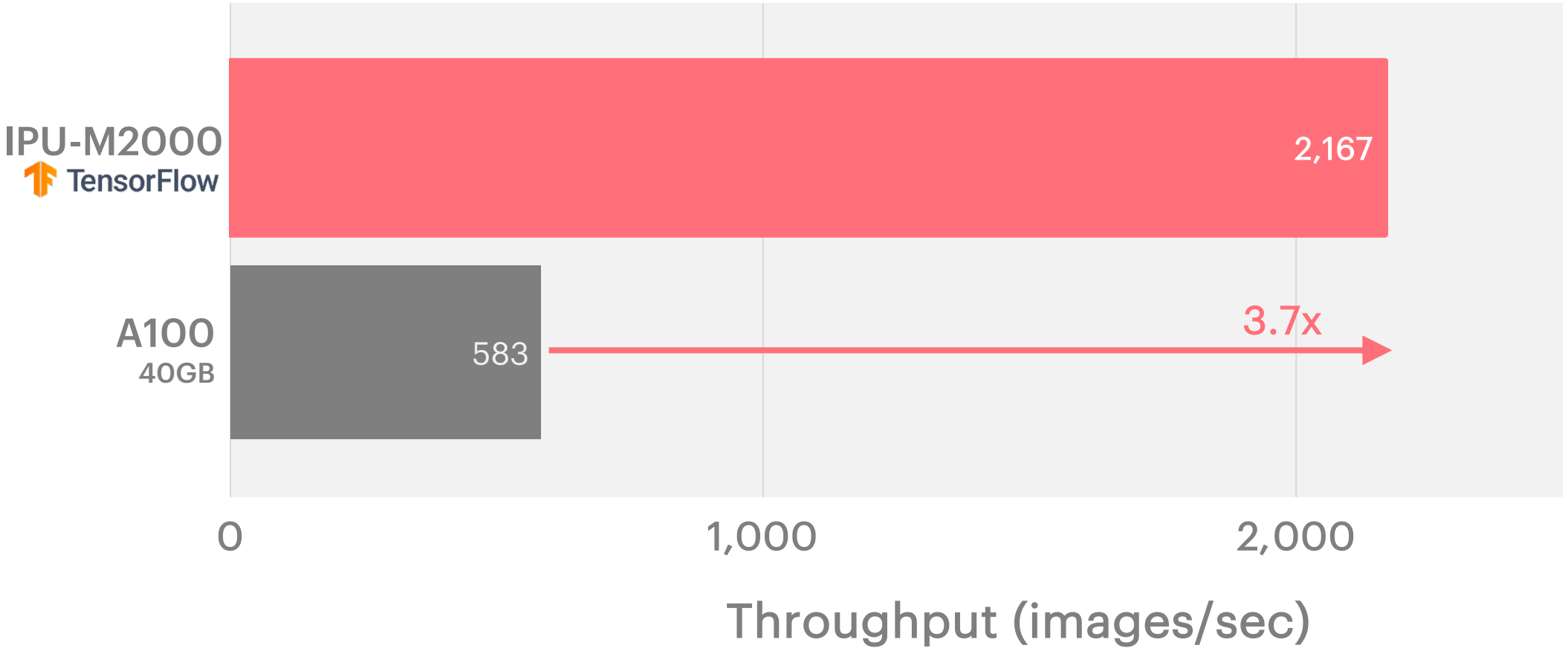
4.6x Higher Throughput



NOTES:
ResNet-50 v1.5 Inference | Synthetic Data | Throughput comparison at lowest latency
1x IPU-M2000 using TensorFlow & PyTorch | FP 16.16 | SDK 1.4.0 | Batch size 4 through 360 | Replicated scaling across IPU-M2000
1x A100 (A100-SXM4-80GB) results using TensorRT 7.2.1 | INT8 | Batch size 1 through 206
A100 results published by NVIDIA (https://developer.nvidia.com/deep-learning-performance-training-inference)

RESNEXT-101 : TRAINING

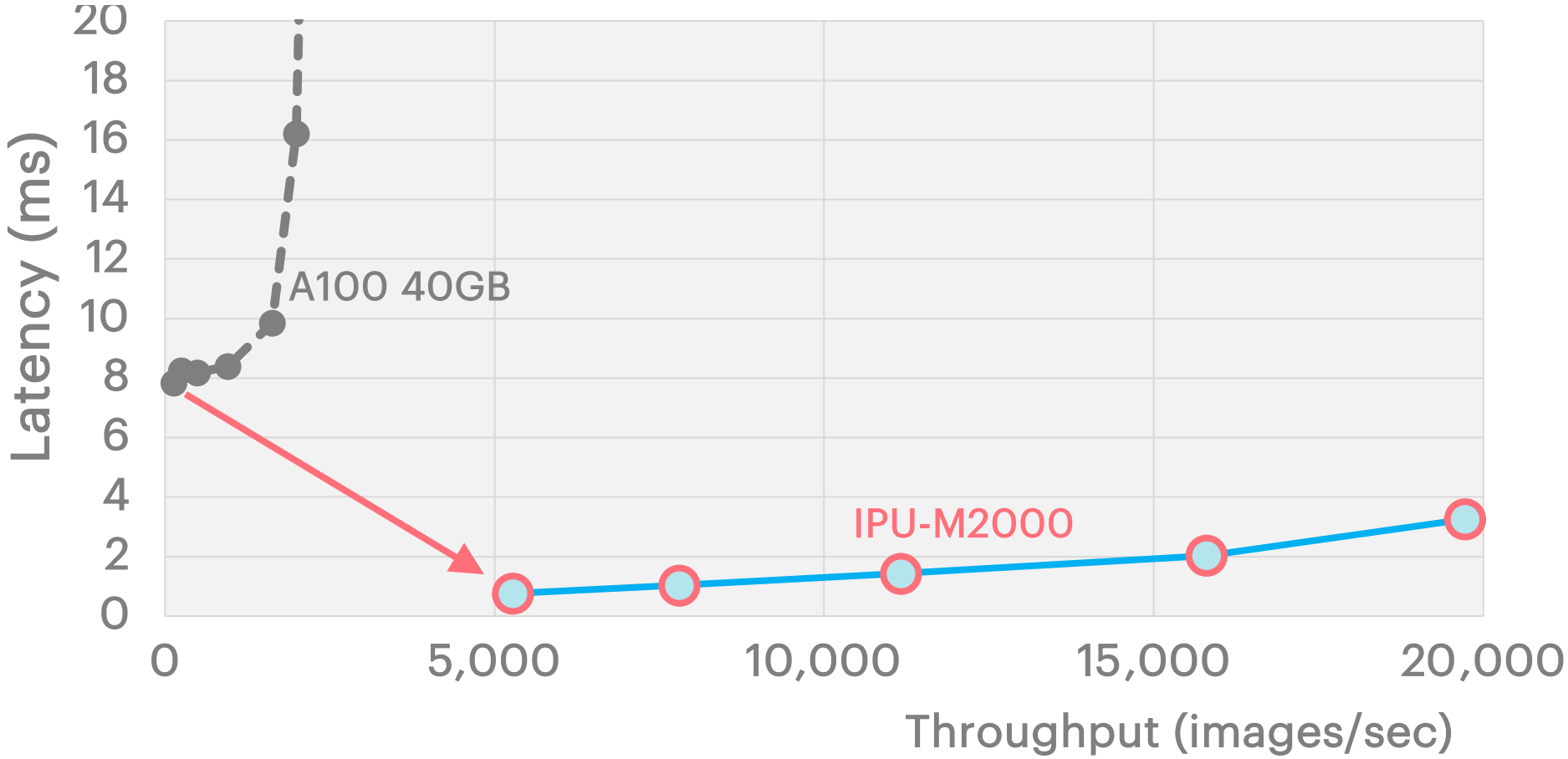
3.7x Higher Throughput



NOTES:
ResNeXt-101 Training Throughput| ImageNet2012 Dataset
1x IPU-M2000 using TensorFlow | FP 16.16 | SDK 1.4.0 | Batch Size 768
1x A100 (A100-SXM4-40GB) GPU results on DGX A100 Platform using PyTorch | Mixed Precision | Batch Size 256
A100 results published by NVIDIA (https://developer.nvidia.com/deep-learning-performance-training-inference)

RESNEXT-101 : INFERENCE

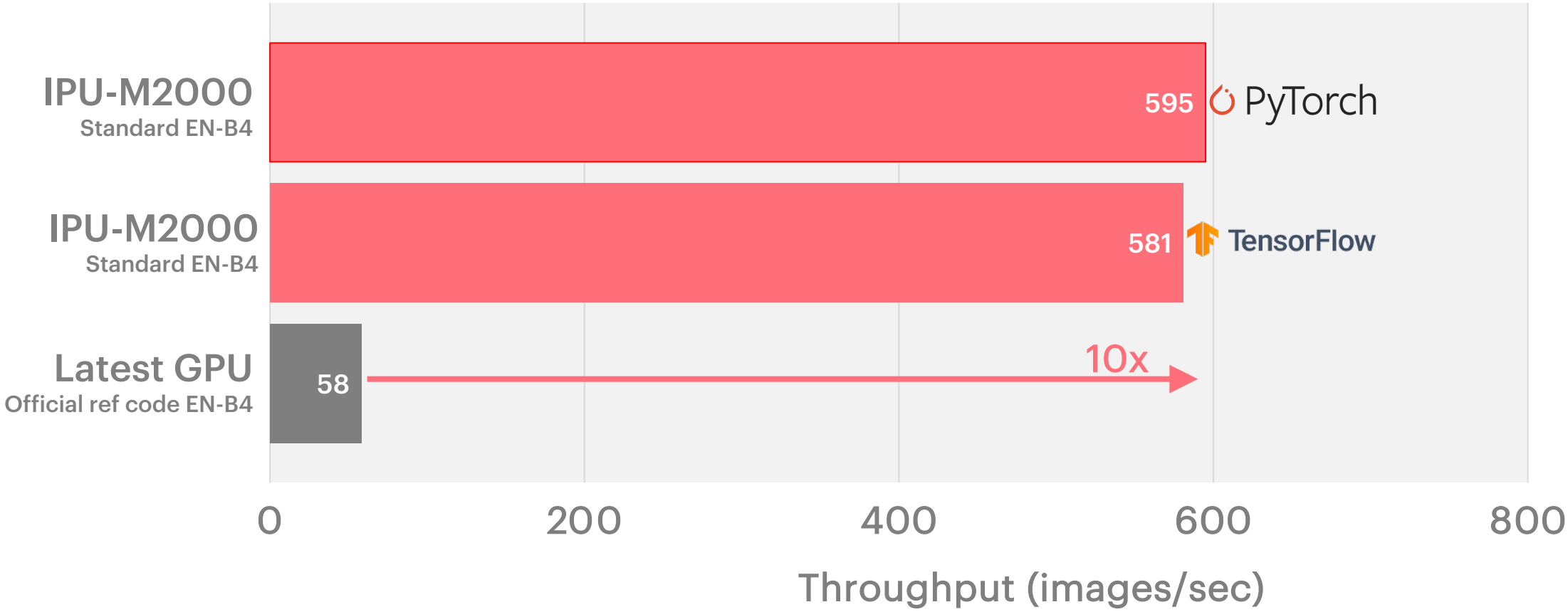
40x higher throughput | 10x lower latency



NOTES:
ResNext-101_32x4d | headline comparisons using lowest latency
1x IPU-M2000 using TensorFlow | Synthetic data | FP 16.16 | SDK 1.4.0 | Batch size 4 through 64 | Replicated scaling across IPU-M2000
1xA100 (40GB) GPU results for TensorFlow | Mixed Precision | Batch Size 1 through 256
No GPU results published on NV results website - using NV github results
(https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/Classification/ConvNets/resnext101-32x4d#advanced)

EFFICIENTNET-B4 : TRAINING

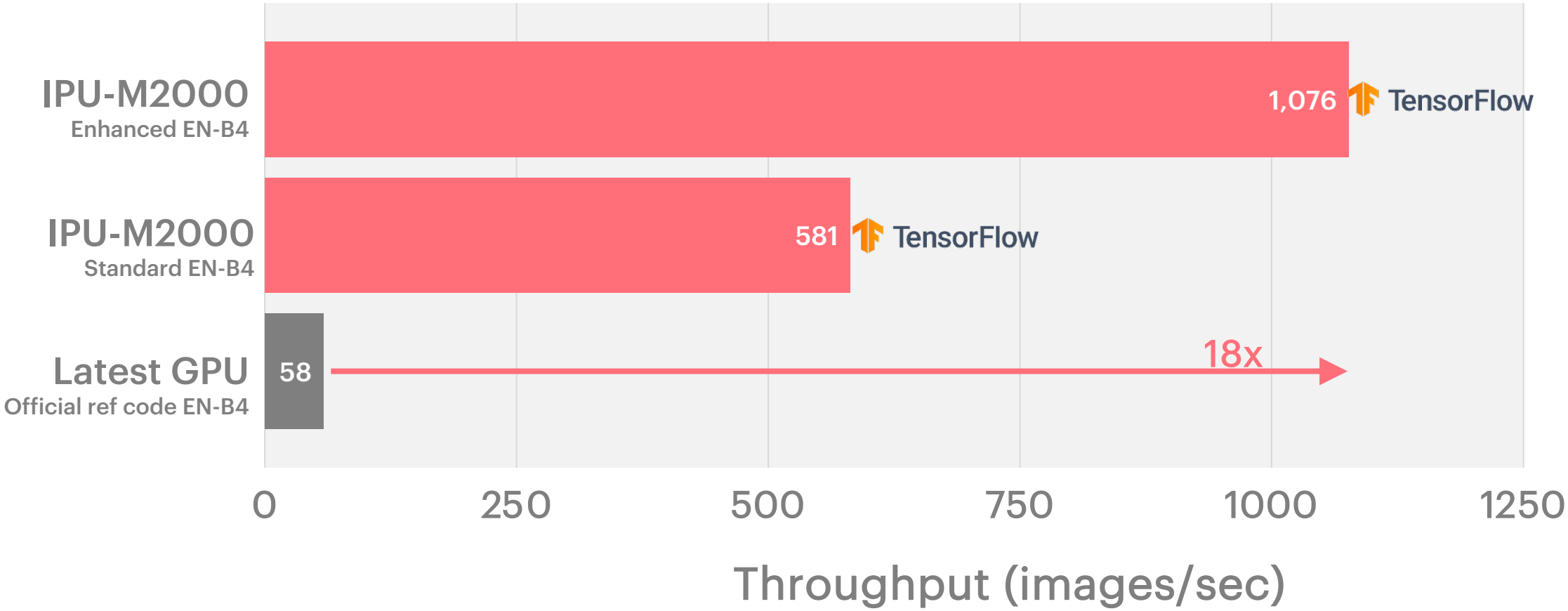
10x Higher Throughput with standard EN-B4



NOTES:
EfficientNet-B4 | ImageNet2012
1x IPU-M2000 | results for TensorFlow & PyTorch | FP 16.32 | SDK 1.4.0
No GPU results published on NV results website
1x Latest GPU (EN-B4 'best' throughput using Group Dim 1) using TensorFlow| FP32 | results measured using public Google repo (FP32 support only) (https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet/).

EFFICIENTNET-B4 : TRAINING

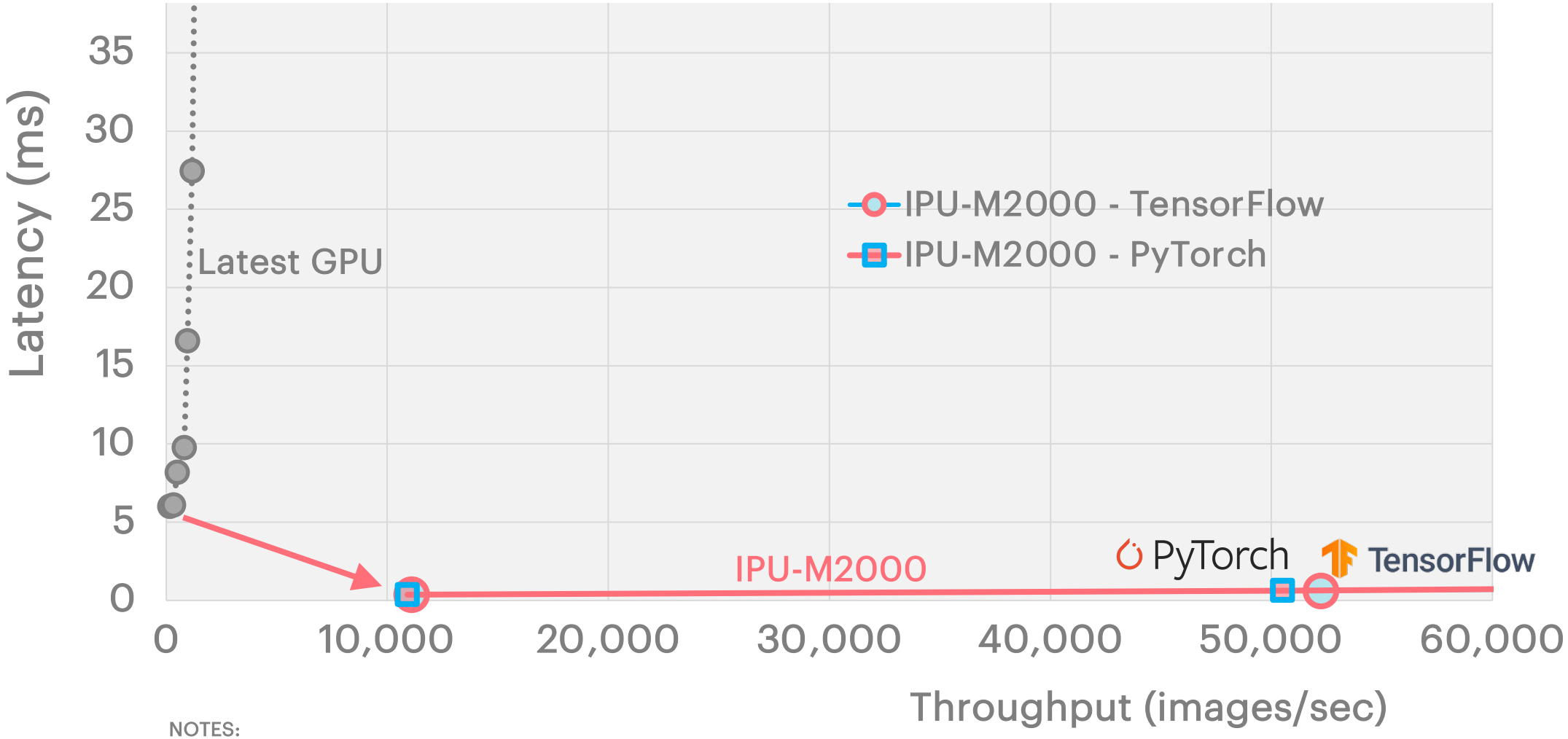
18x Higher Throughput with IPU optimised configuration



NOTES:
EfficientNet-B4 | Real Data (ImageNet)
1x IPU-M2000 | results for TensorFlow | FP 16.32 | SDK 1.4.0 | (EN-B4 Enhanced version using Group Dim 16, standard uses Group Dim 1)
No GPU results published on NV results website
1x Latest GPU (EN-B4 'best' throughput using Group Dim 1) using TensorFlow| FP32 | results measured using public Google repo (FP32 support only)
(https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet/).

EFFICIENTNET-B0: INFERENCE

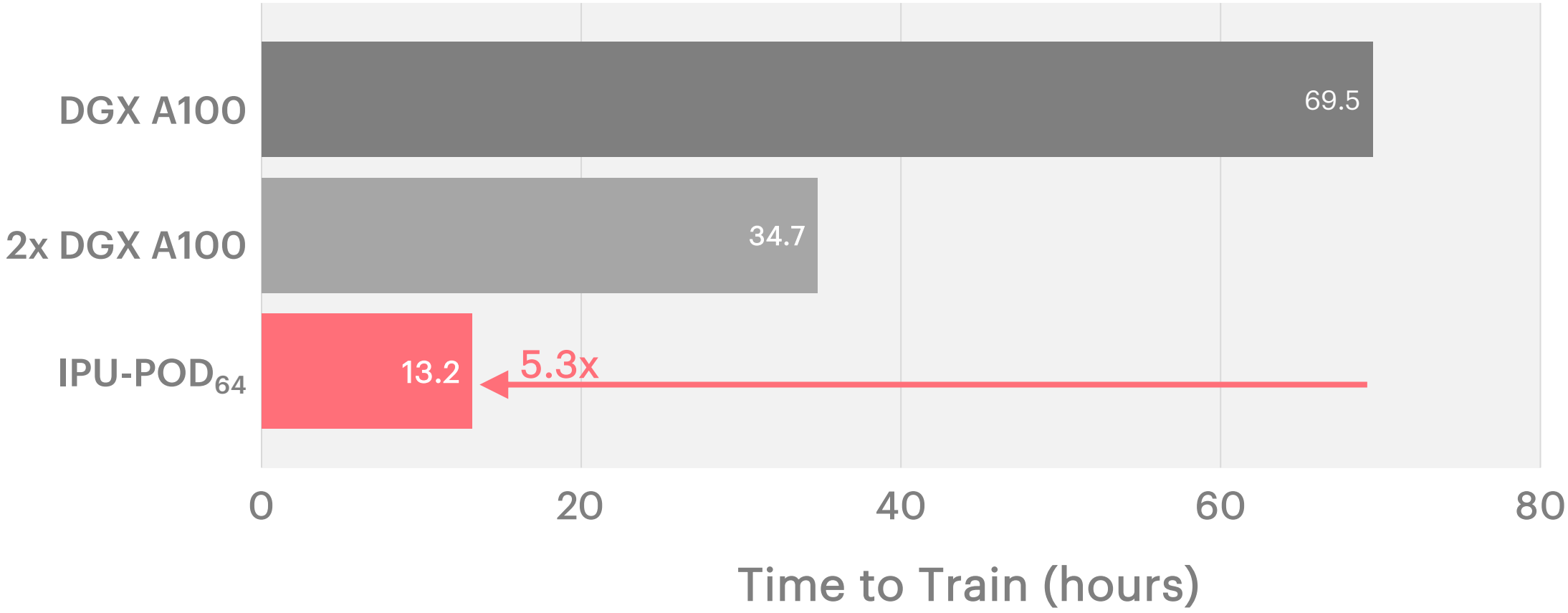
>60x higher throughput | >16x lower latency



NOTES:
EfficientNet-B0 | headline comparisons using lowest latency
1x IPU-M2000 using TensorFlow & PyTorch | FP 16.16 | Synthetic | SDK 1.4.0 | Batch size 4 through 160 | Replicated scaling across IPU-M2000
No GPU results published on NV results website
1x Latest GPU using TensorFlow | Batch Size 1 through 512 | results measured using public Google repo | FP32 support only | Synthetic | (https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet/).

BERT-LARGE: TRAINING

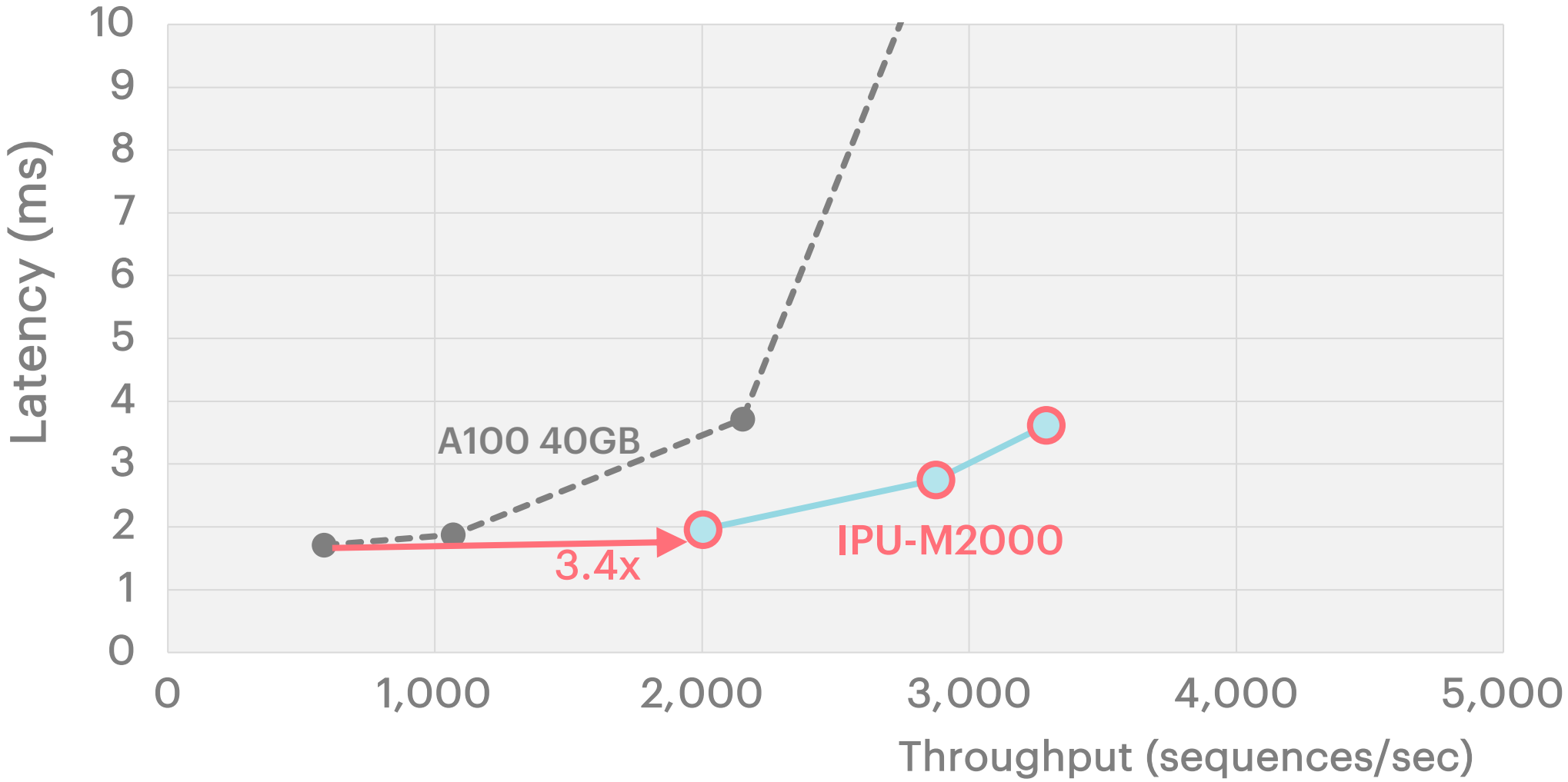
5.3x Faster Time To Train



NOTES:
BERT-Large using Wikipedia dataset | end to end pre-training
IPU-POD64 (16x IPU-M2000 Server) using PopART | SDK 1.4.0 | Mixed Precision Ph1 SL=128, Ph2 SL=384.
DGX A100 results calculated using NV published TensorFlow throughput & training scheme | Mixed Precision | Assume linear scaling from 1x to 2x DGX A100
https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/LanguageModeling/BERT#training-accuracy-results

BERT-LARGE: INFERENCE

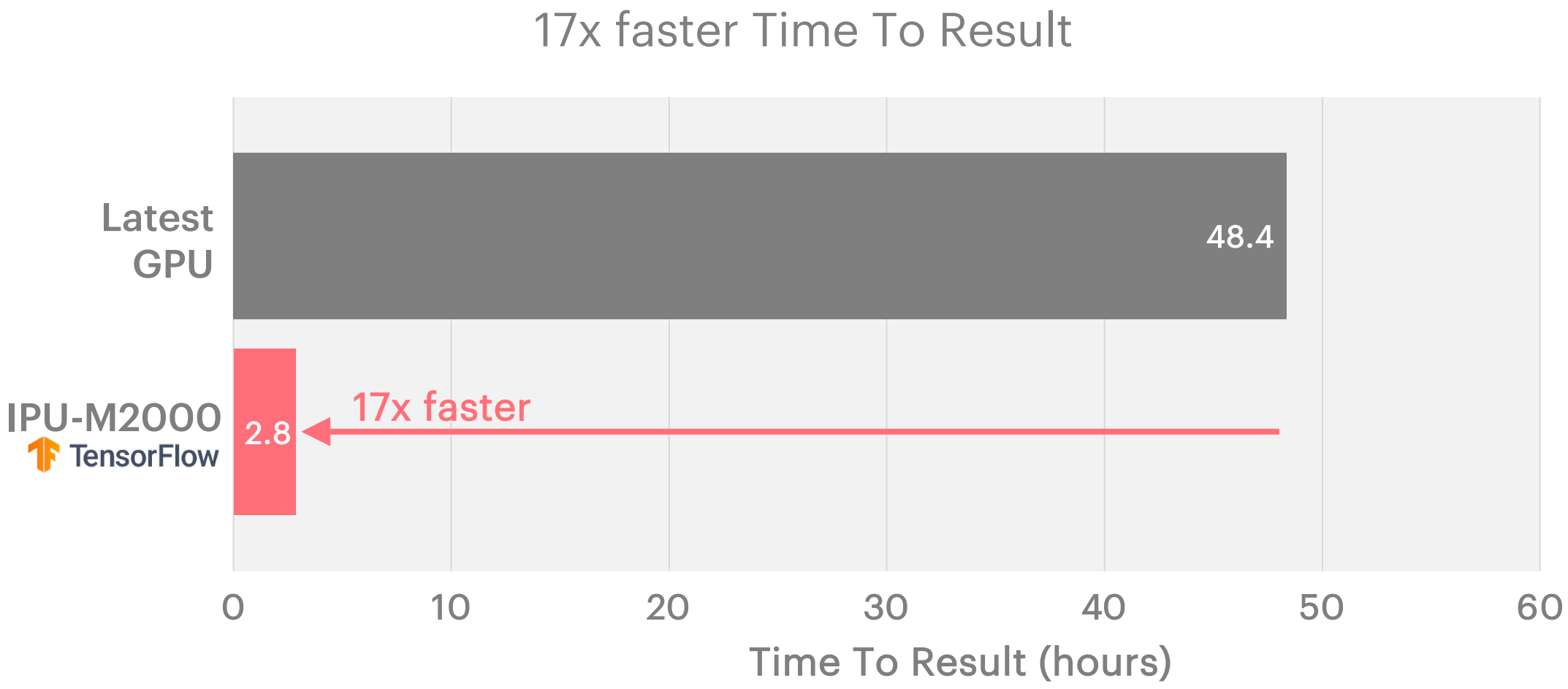
3.4x Higher Throughput at lowest latency



NOTES:
BERT-Large Inference | SQuAD | headline comparisons using lowest latency
1x IPU-M2000 using PopART | FP 16.16 | SDK 1.4.0 | Batch size 4,8,12 | Replicated scaling across IPU-M2000
1x A100 (A100-SXM4-40GB) results using TensorRT 7.2 | INT8 | Batch size 1,2,8+
A100 results published by NVIDIA (https://developer.nvidia.com/deep-learning-performance-training-inference)

MCMC PROBABILISTIC MODEL : TRAINING

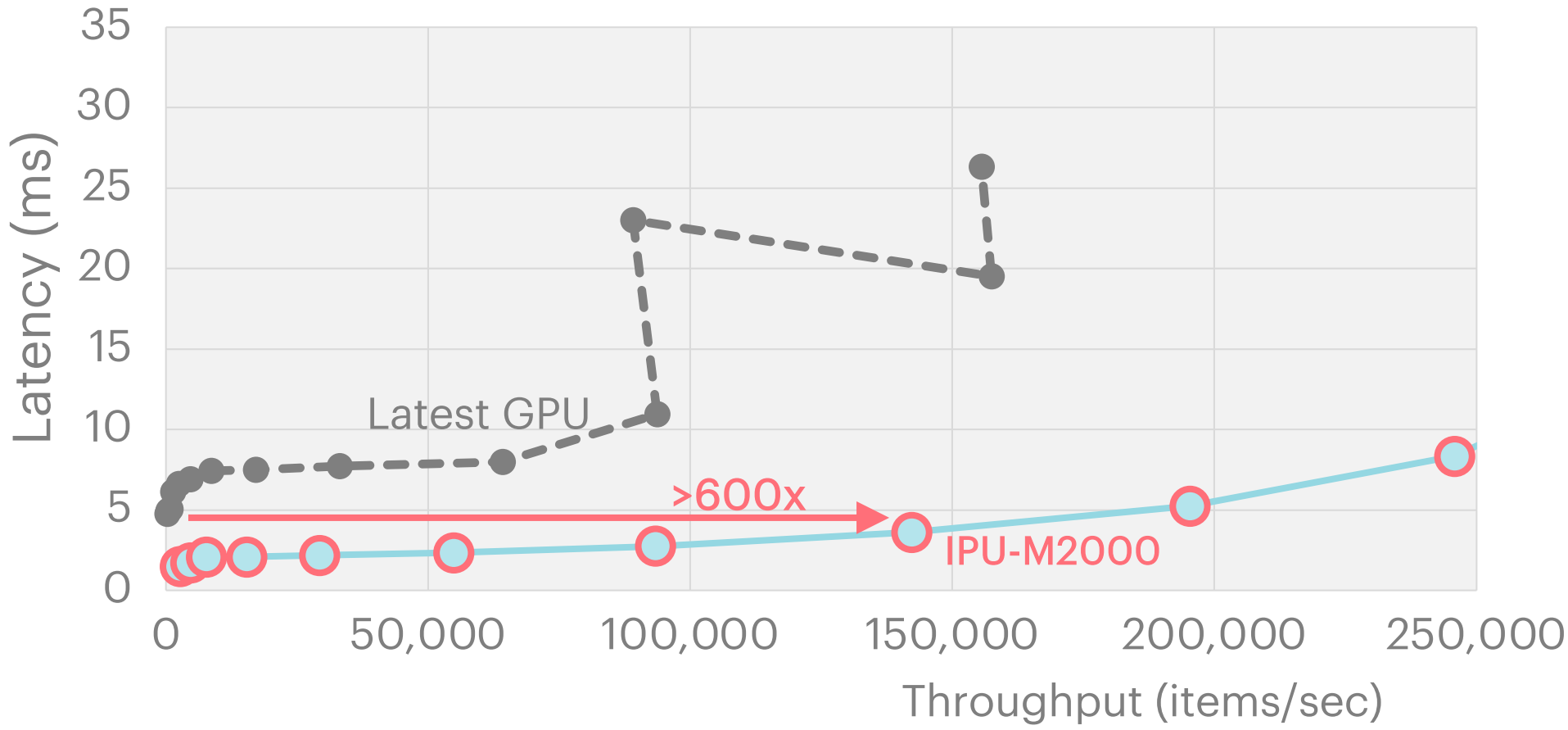
TensorFlow Probability model - representative finance workload for alpha estimation



NOTES:
Markov Chain Monte Carlo – Probabilistic model with TensorFlow Probability, representative of workload used by Carmot Capital
Neural network with 3 fully-connected layers (num units in 1st layer=40, #dimensions in training set =22, #leapfrog steps=1000, calcs in sliding window=200)
1x IPU-M2000 using TensorFlow | FP 32.32 | SDK 1.4.0 | 1600 samples
1x Latest GPU | FP 32.32 | 1600 samples

LSTM : INFERENCE

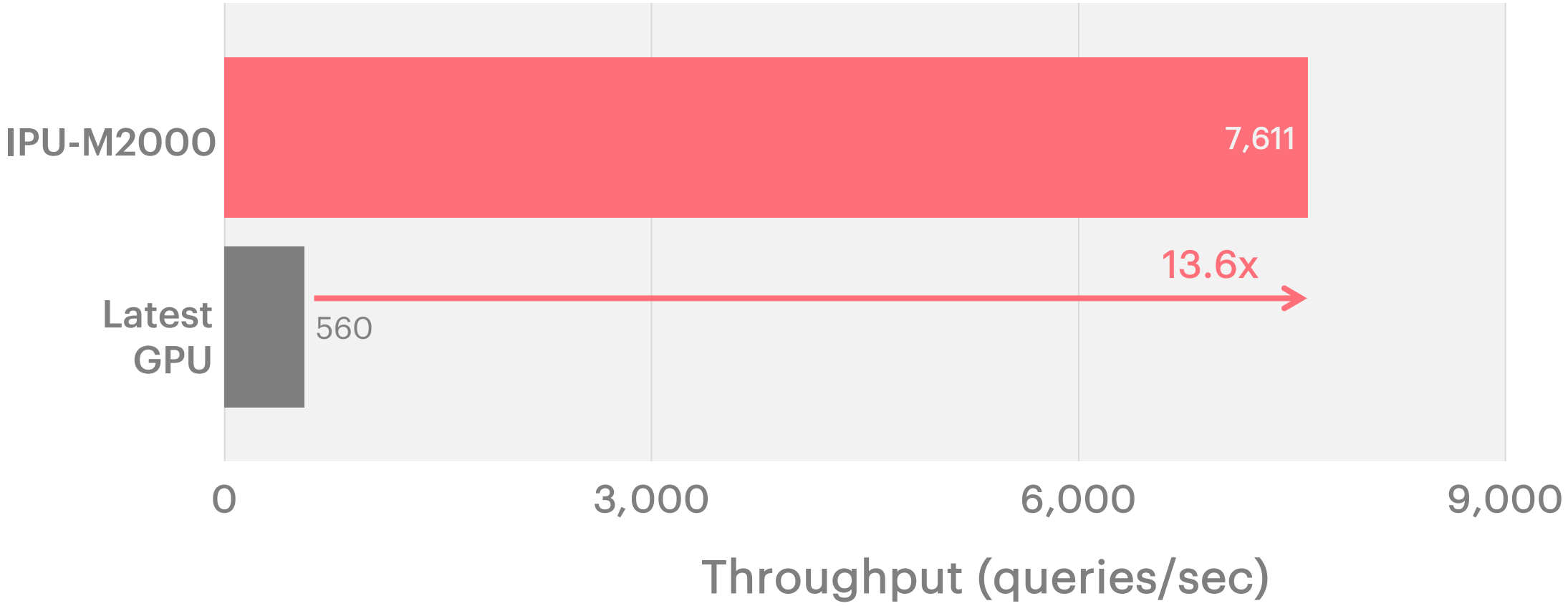
>600x higher throughput at lower latency



NOTES:
2 LSTM layers, each with 256 units, 200 time steps, 16 input dimensions, host generated data
1x IPU-M2000 using TensorFlow | FP 16.32 | SDK 1.4.0 | Batch size 4 through 2048 | Replicated scaling across IPU-M2000
1x Latest GPU | TensorFlow | Mixed Precision | Batch Size 1 through 4096

DeepVoice 3 TTS: TRAINING

13.6x higher throughput



NOTES:
DeepVoice3 TextToSpeech | VCTK Corpus | Regular SGD
1x IPU-M2000 using PopART| FP 32 | SDK 1.4.0 | Batch size 128
1x Latest GPU | Pytorch | FP32 | Batch Size 1024

THANK YOU