World's 1st Universal Processor for Servers / AI / HPC

Server / Supercomputer / AI Chip

- For hyperscale datacenters

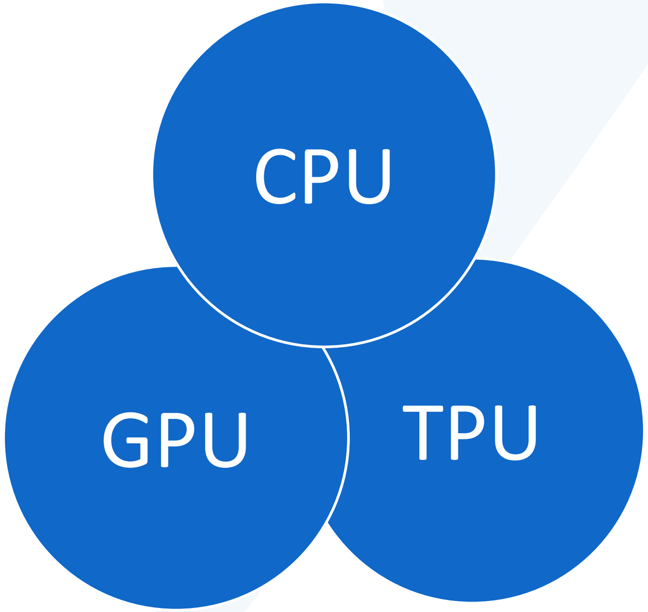
Humanity: 1st human brain sized AI

- Not only Focus on Deep Learning AI
- Also Explainable, Bio, Spiking and General AI

Prodigy is faster than Xeon/GPU/TPU

- Faster, 10x less power, 1/3 cost of Xeon
- Faster than NVIDIA A100 in HPC and AI

Tachyum Universal Processor is Best of



Long Vectors

Matrix Operations

AI The Most Important Driver of GDP Growth

Bloomberg 2018

AI adds $15T to the economy by 2030

Forbes 2017 AI & GDP

AI 40% productivity growth by 2035

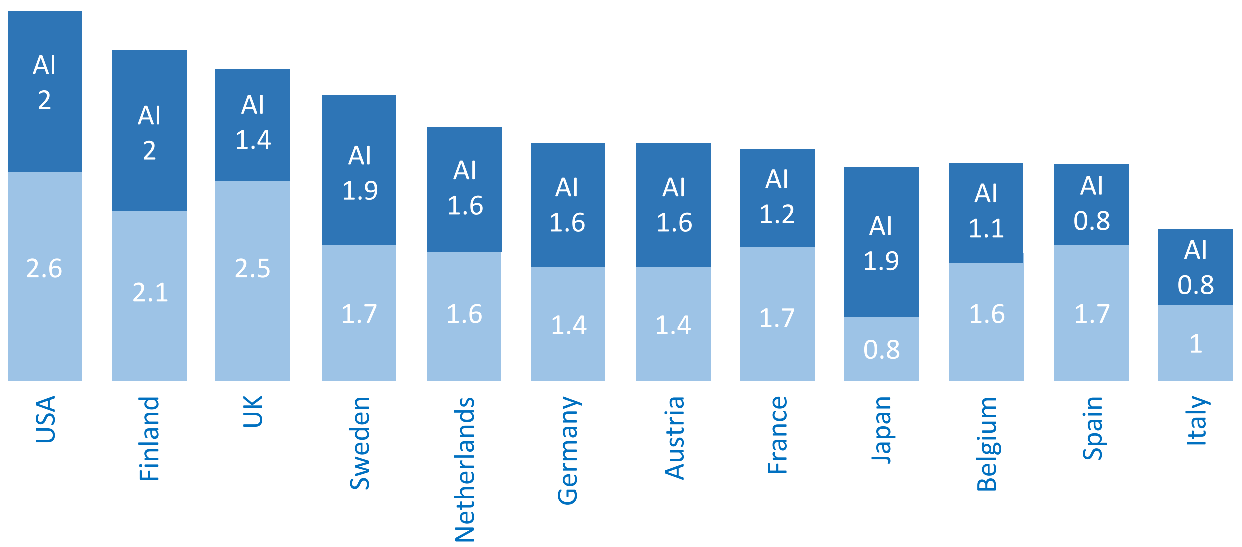
PwC 2017

AI adds 14% to GDP by 2030

Putin 2017

“the leader in AI will rule the world”

AI Almost Doubles GDP Growth (%)



Tachyum™

Tachyum is Critical for Datacenter Growth

3% of planet's electricity today

60% more than UK

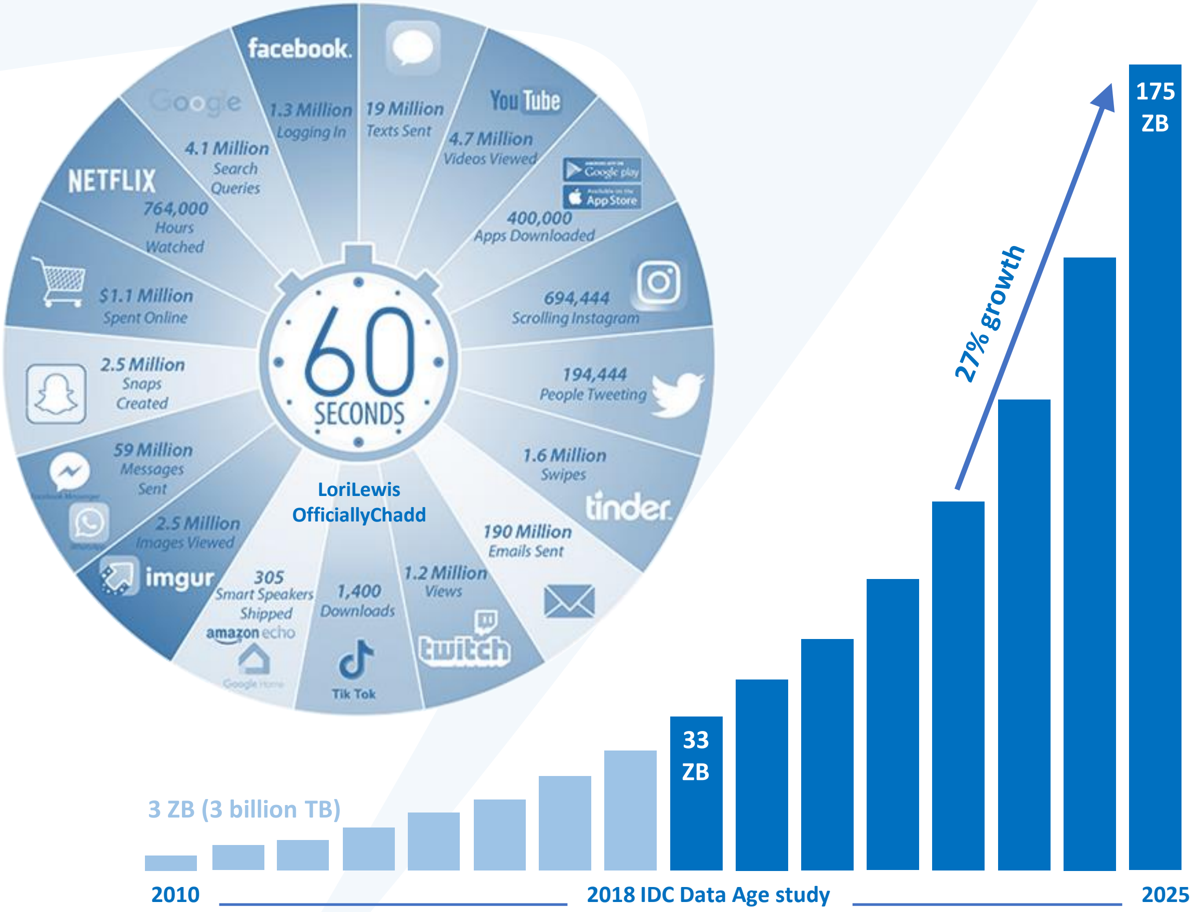
50% of the planet's energy by 2040

At 27% growth, it will be 33% by 2030

Largest CO_2 reduction impact

More than solar panels, windmills, …

10x lower power is needed to extend current datacenter growth



Prodigy Universal Processor Reduces Carbon Footprint

Lowers Greenhouse Gasses

High Performance
Low Power

- 3x higher performance
- 10x lower power

24/7 Server
“On” Time

- Unified CPU, GPU & TPU
- Homogenous & composable

Prodigy’s High Efficiency Helps to Keep Our Planet Green

AI Supercomputer: once-in-a-decades opportunity

EU AI is today in the hands of other countries, misaligned with EU interests

- Relying on other countries who are competitors and potential adversaries is not safe anymore
- Now, EU top priority is digital and technological sovereignty especially semiconductors

Slovakia needs to transition from cars and assembly to a knowledge-based economy

- EU consumes 30% of world compute resources, but has only 5% of world's resources
- Tachyum offers unique once-in-a-decade opportunity for Slovakia, and to fulfill EU critical needs
- Replace "brain drain" with "brain gain" by creating world class job opportunities in Slovakia

The world's fastest and most-powerful AI Supercomputer is built in Slovakia

- Unifying Europe by bridging language divide
- Fostering new high-tech industry
- Scientists from around the world will come to Slovakia to conduct ground-breaking research

World's Fastest AI Supercomputer

SLOVAK ACADEMY OF SCIENCES

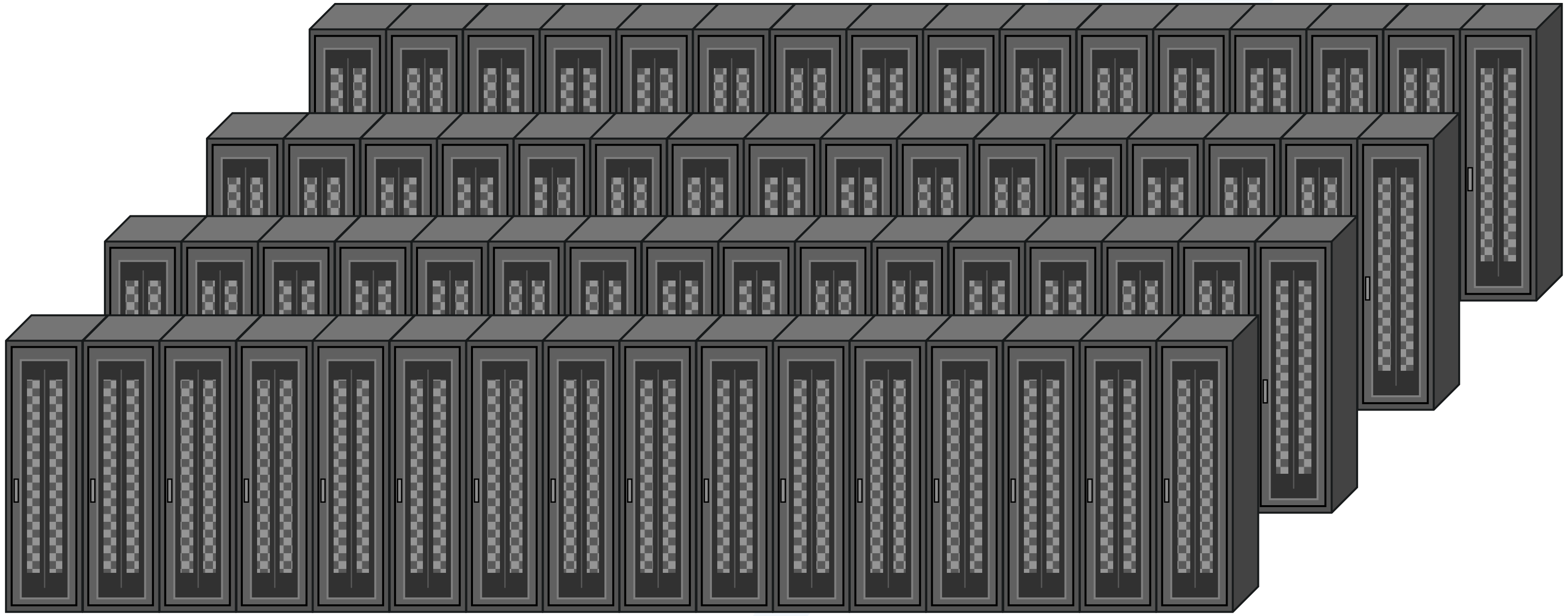
64 Compute Racks

64 AI ExaFLOPs

Operational in 2022

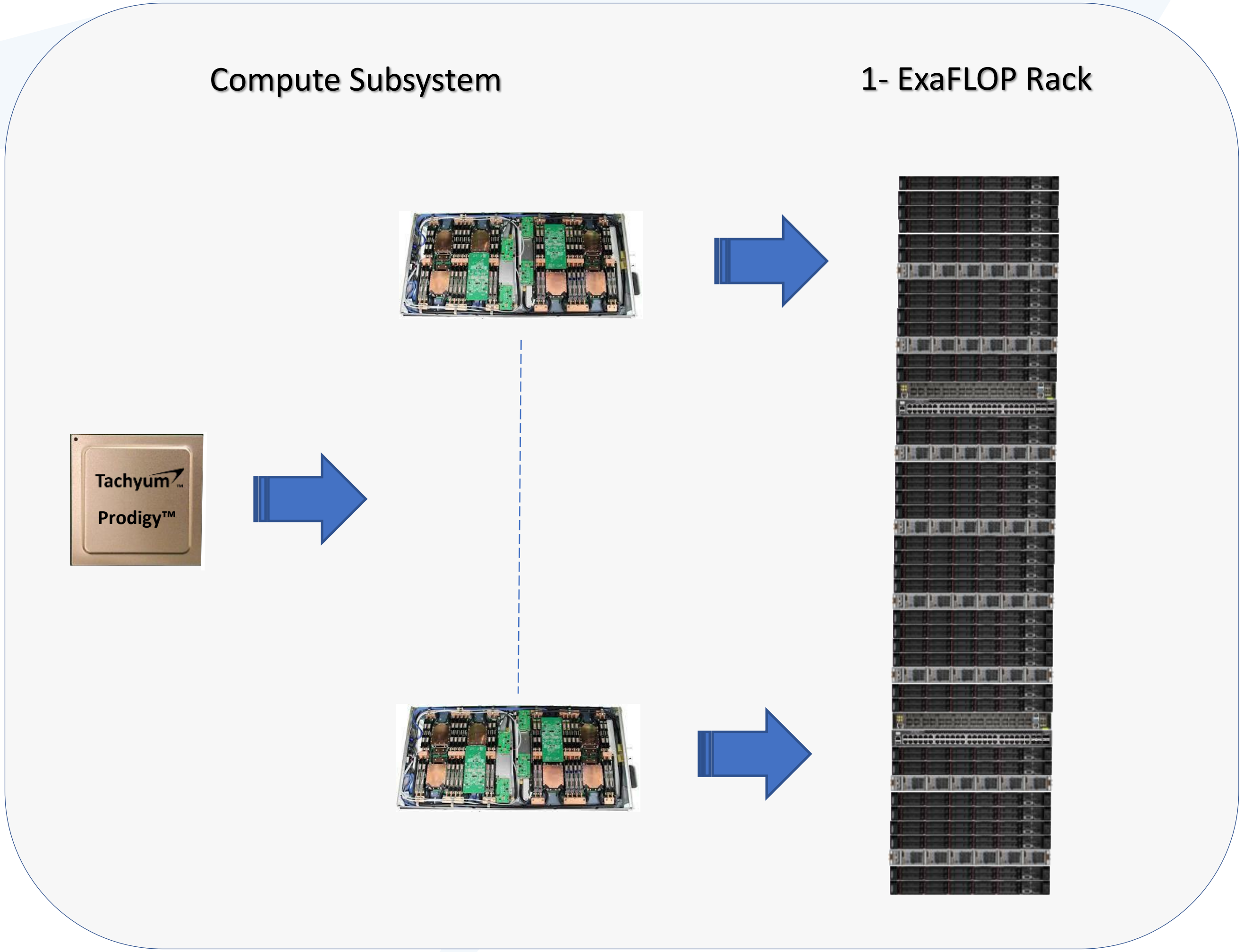
NSCC Slovakia Supercomputer

Prodigy-Powered



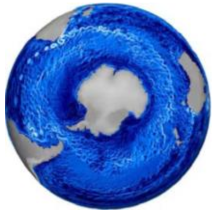
NSCC – SC Compute Rack

- High – Performance
  - 1 AI ExaFLOPs of Training and Inferencing per rack
- Prodigy T16128 Universal Processor
  - 128 64-bit cores
  - 2 vector units
  - Maximizes performance and efficiency
- Rack Configuration
  - 32 Prodigy 1U Compute Nodes
  - 8 sockets per compute node
  - 256 sockets per rack

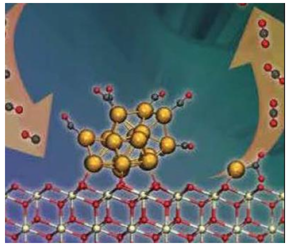


NSCC-SC and Prodigy Addressing the World’s Problems

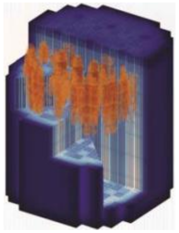
Climate change impact assessment



Biofuel catalyst design

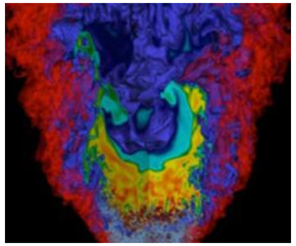


Next generation nuclear reactors

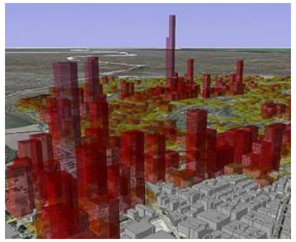


Improve efficiency and reduce cost

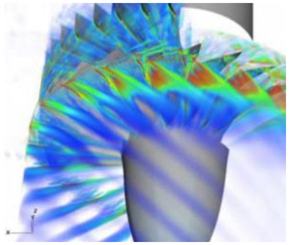
Design of low-emission engine



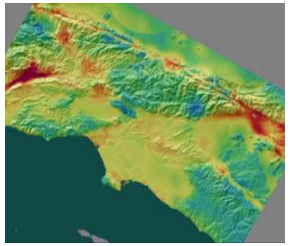
Energy and water nexus



Scaling carbon capture designs

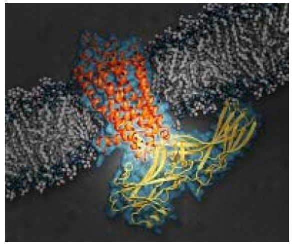


Modeling and risk assessment



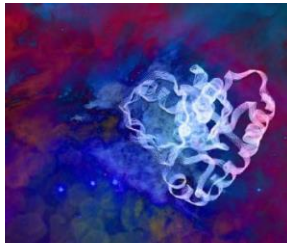
Renewable energy planning

Protein structure and dynamics



Process of additive manufacturing

Drugs and vaccines discovery



6x Faster Drugs and Vaccine Discovery

Discovery 1.5 years
Screen million molecules

Optimization 3 years
Design & test 1000 molecules

Trials 1.5 years
On animals

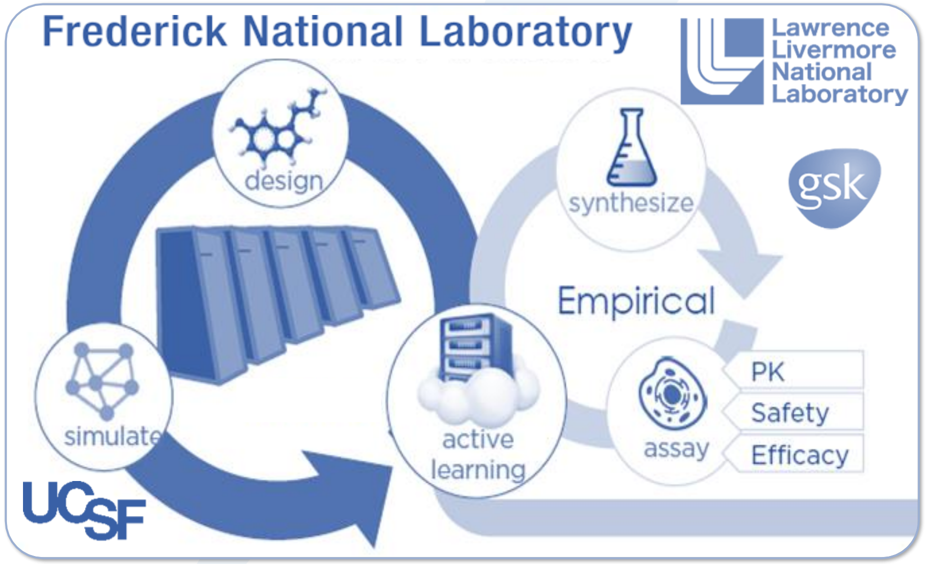
Clinical Trial

6 years → 1 year

Tachyum
Low Cost HPC
Available for ALL

Patient's data

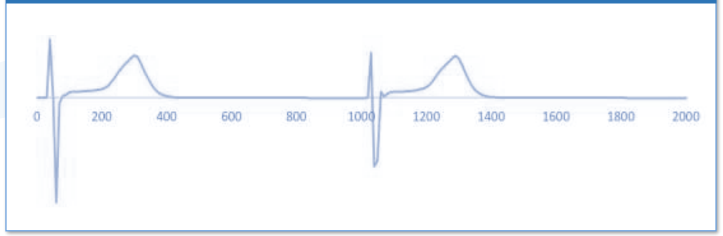
Personalized Medicine



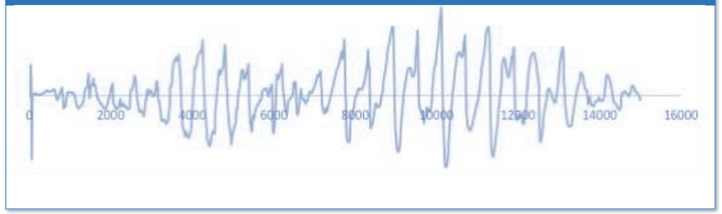
Tachyum™

25,000 Lives To Save Per Year

ECG simulation for healthy heart



Arrythmia simulation after drug Sotalol



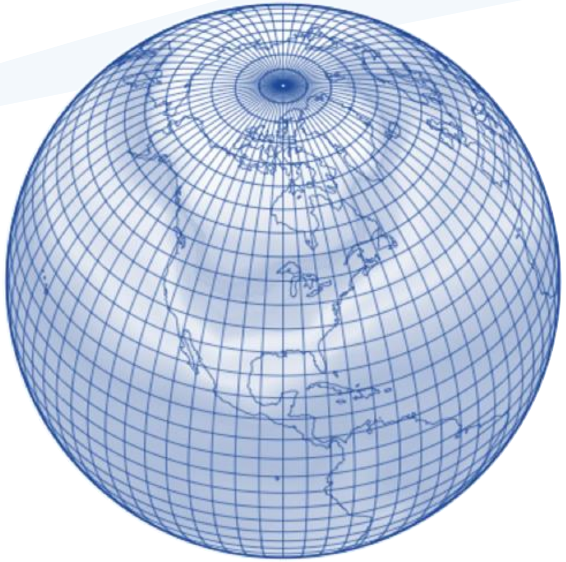
$100,000

Tachyum $10,000

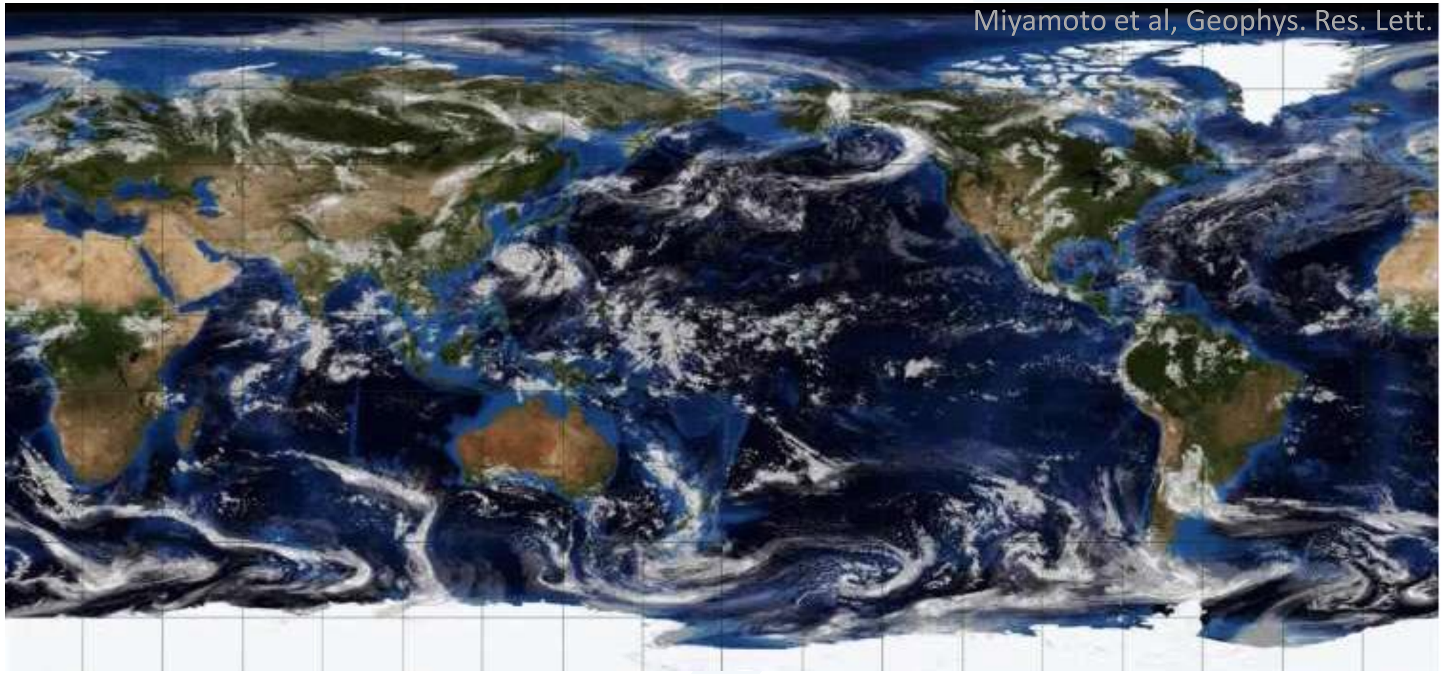
Tachyum Is Democratizing HPC

Key to Understand Climate Change

Existing models not accurate



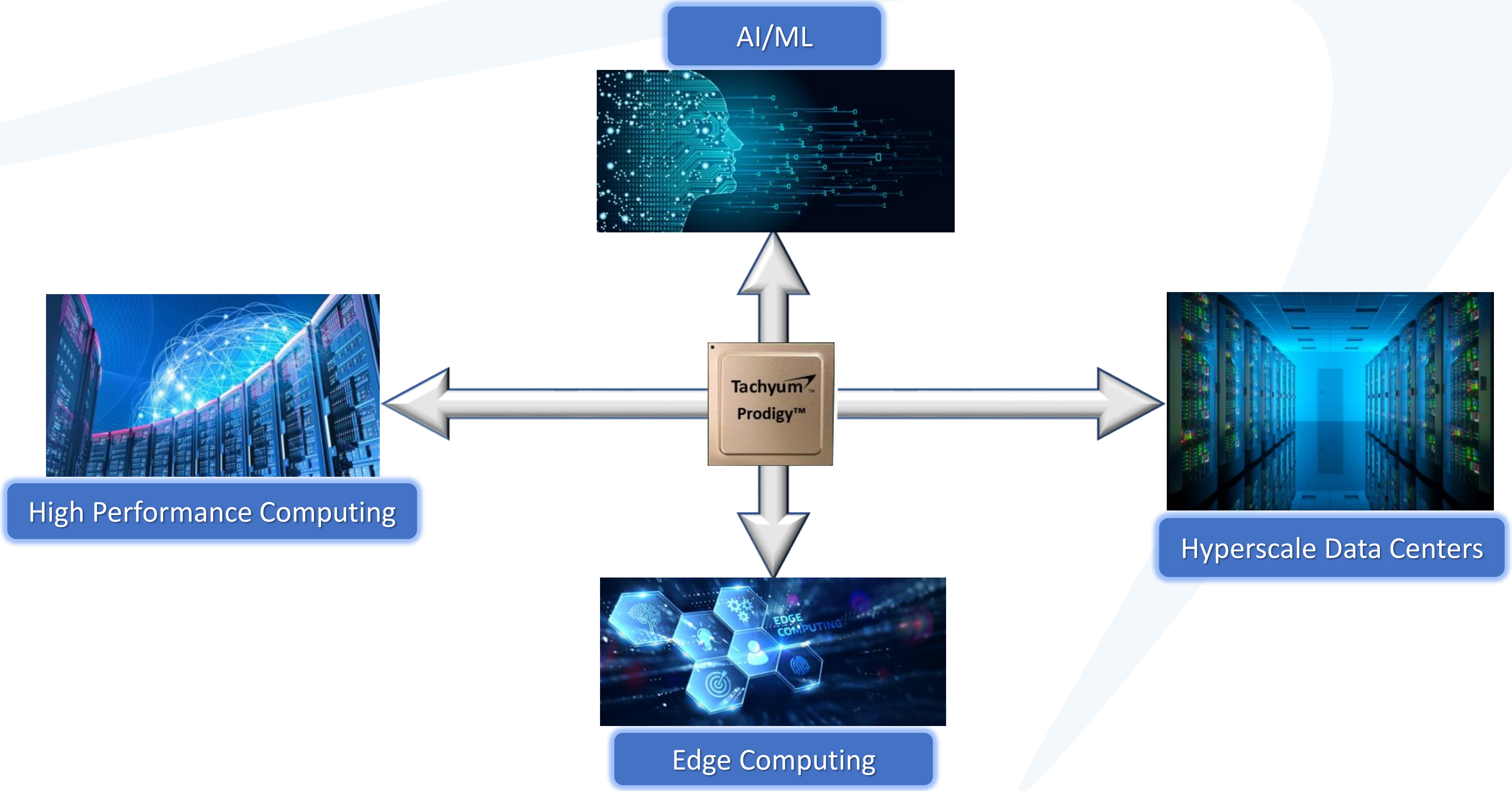
Tachyum enables <1 km resolution to accurately model clouds



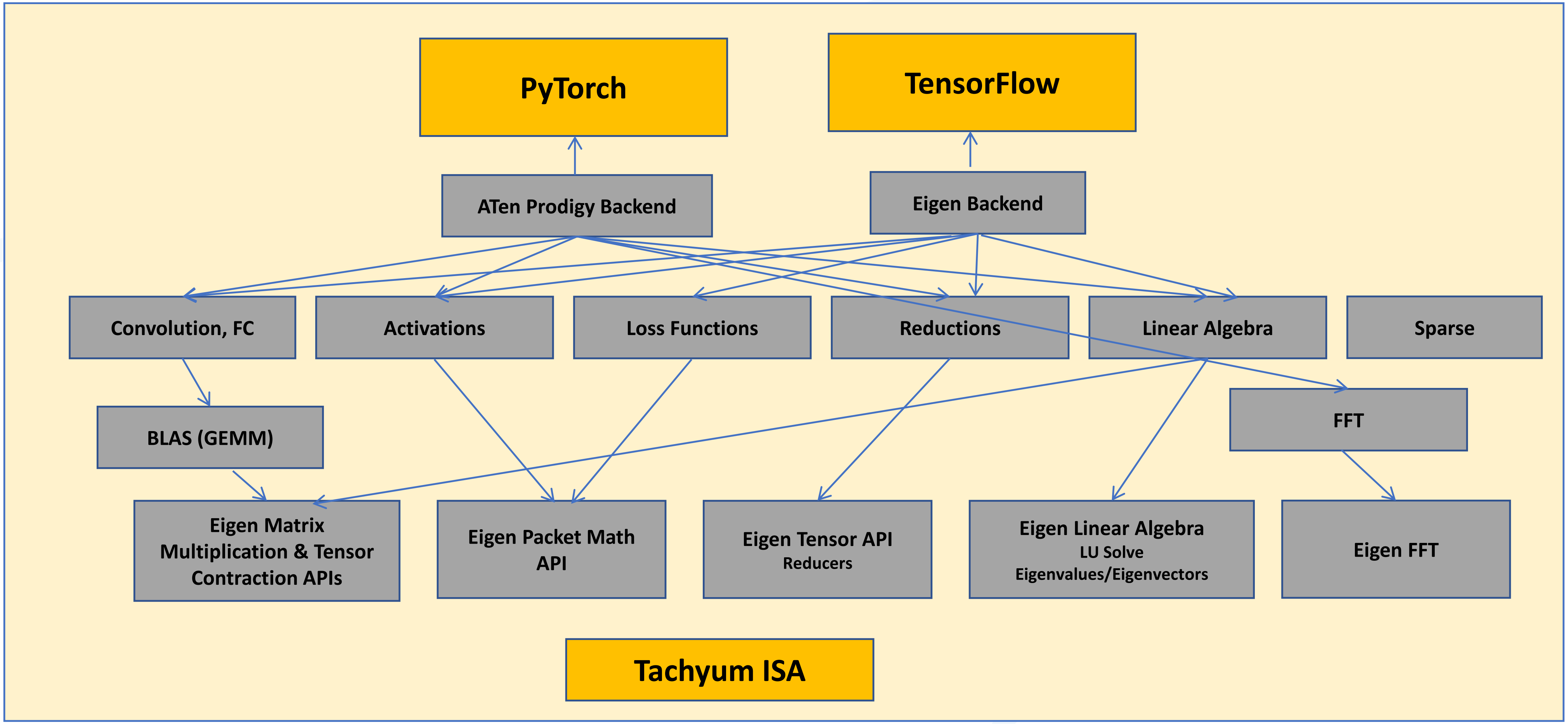
Tachyum™

Prodigy Target Platforms

Prodigy has multiple SKUs that align with a wide range of markets, applications, and workloads



Native support for AI frameworks



PYTORCH TensorFlow

- Activation & Loss Function – optimized utilizing Tachyum vector instructions in standard and low precision modes
- Dense GEMM library implemented utilizing Tachyum matrix instructions in standard and low precision modes, stochastic rounding, single and multithreaded
- Custom Sparse GEMM library implemented utilizing Tachyum vector and matrix instructions
- Convolutional and Dense operators implemented utilizing Tachyum matrix instructions in standard and low precision modes, including depthwise separable and pointwise convolutions
- Circulant and Butterfly Convolutional and Dense operators implemented utilizing custom FFT for matrix multiplication

Native support for AI frameworks


```
root@tachy:~/tachy_pytorch# exit

tachy 0.1 tachy ttyS0

tachy login:
```

Revolutionizing AI training with high performance Prodigy Matrix Instructions with reduced precision

- Prodigy CPU addresses continuing trends in AI models, explosion in complexity as demanded by more complex NLP models and more accurate conversational AI.
- NLP transformer models are hundreds of times larger and more complex than image classification models like ResNet-50. Training these massive models in FP32 precision can take days or even weeks.
- Matrix multiplication in Prodigy CPU provide an order-of-magnitude higher performance with reduced precisions substantially reducing training-to-convergence times while maintaining accuracy.

Vector and Matrix Execution

Floating-Point/ Integer Units

- IEEE Double, Single, and Half-Precision FPU
- AI 8-bit Floating-Point Data Type
- 2 x 1024-bit Multiply-Add Vector/Matrix Units
- 2 x 1024-bit ALUs Supporting 8, 16, and 32-bit Integers with No/Signed/Unsigned Saturation

Vector and Matrix Operations

- Matrix Operations: 4x Less Power than competition
- 8-bit Int/FP: 16 x 16
- 16-bit Int/FP: 8 x 8
- FP64, FP32: 4 x 4
- 8 x 8 Matrix Multiply-Add = 1024 Flops
    - Uses 6 Source and 2 Destination Registers
- Ability to Increase Performance 2x in the Future

Maximum Issue Rate per Clock

- 2 x 1024-bit Multiply-Add
- 2 x 1024-bit Integer Instructions
- 1 Load, 1 Load/Store, 1 Store

P16128 Total FLOPS by Data Type

Data Type	FLOPS/ Core	Total FLOPS – P16128 (128 cores x 4 GHz x FLOPS/Core)
Double Precision	2 x 32 FLOPS = 64	32 TeraFLOPS
Single Precision	2 x 128 FLOPS = 128	128 TeraFLOPS
Half Precision	2 x 512 FLOPS = 1024	512 TeraFLOPS
FP8	2 x 2048 = 4096	4 PetaFLOPS

Prodigy Supports 16x16, 8x8, and 4x4 Matrix Operations

$$\begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} & d_{0,4} & d_{0,5} & d_{0,6} & d_{0,7} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} & d_{1,7} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} & d_{2,7} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} & d_{3,7} \\ d_{4,0} & d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} & d_{4,7} \\ d_{5,0} & d_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & d_{5,6} & d_{5,7} \\ d_{6,0} & d_{6,1} & d_{6,2} & d_{6,3} & d_{6,4} & d_{6,5} & d_{6,6} & d_{6,7} \\ d_{7,0} & d_{7,1} & d_{7,2} & d_{7,3} & d_{7,4} & d_{7,5} & d_{7,6} & d_{7,7} \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} & a_{0,5} & a_{0,6} & a_{0,7} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & a_{1,7} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & a_{2,7} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & a_{3,7} \\ a_{4,0} & a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} & a_{4,7} \\ a_{5,0} & a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & a_{5,6} & a_{5,7} \\ a_{6,0} & a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} & a_{6,7} \\ a_{7,0} & a_{7,1} & a_{7,2} & a_{7,3} & a_{7,4} & a_{7,5} & a_{7,6} & a_{7,7} \end{bmatrix} \times \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} & b_{0,6} & b_{0,7} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} & b_{2,6} & b_{2,7} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & b_{3,6} & b_{3,7} \\ b_{4,0} & b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} & b_{4,7} \\ b_{5,0} & b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} & b_{5,7} \\ b_{6,0} & b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} & b_{6,7} \\ b_{7,0} & b_{7,1} & b_{7,2} & b_{7,3} & b_{7,4} & b_{7,5} & b_{7,6} & b_{7,7} \end{bmatrix} + \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} & c_{0,5} & c_{0,6} & c_{0,7} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & c_{1,6} & c_{1,7} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} & c_{2,6} & c_{2,7} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} & c_{3,6} & c_{3,7} \\ c_{4,0} & c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} & c_{4,6} & c_{4,7} \\ c_{5,0} & c_{5,1} & c_{5,2} & c_{5,3} & c_{5,4} & c_{5,5} & c_{5,6} & c_{5,7} \\ c_{6,0} & c_{6,1} & c_{6,2} & c_{6,3} & c_{6,4} & c_{6,5} & c_{6,6} & c_{6,7} \\ c_{7,0} & c_{7,1} & c_{7,2} & c_{7,3} & c_{7,4} & c_{7,5} & c_{7,6} & c_{7,7} \end{bmatrix}$$

Quantization

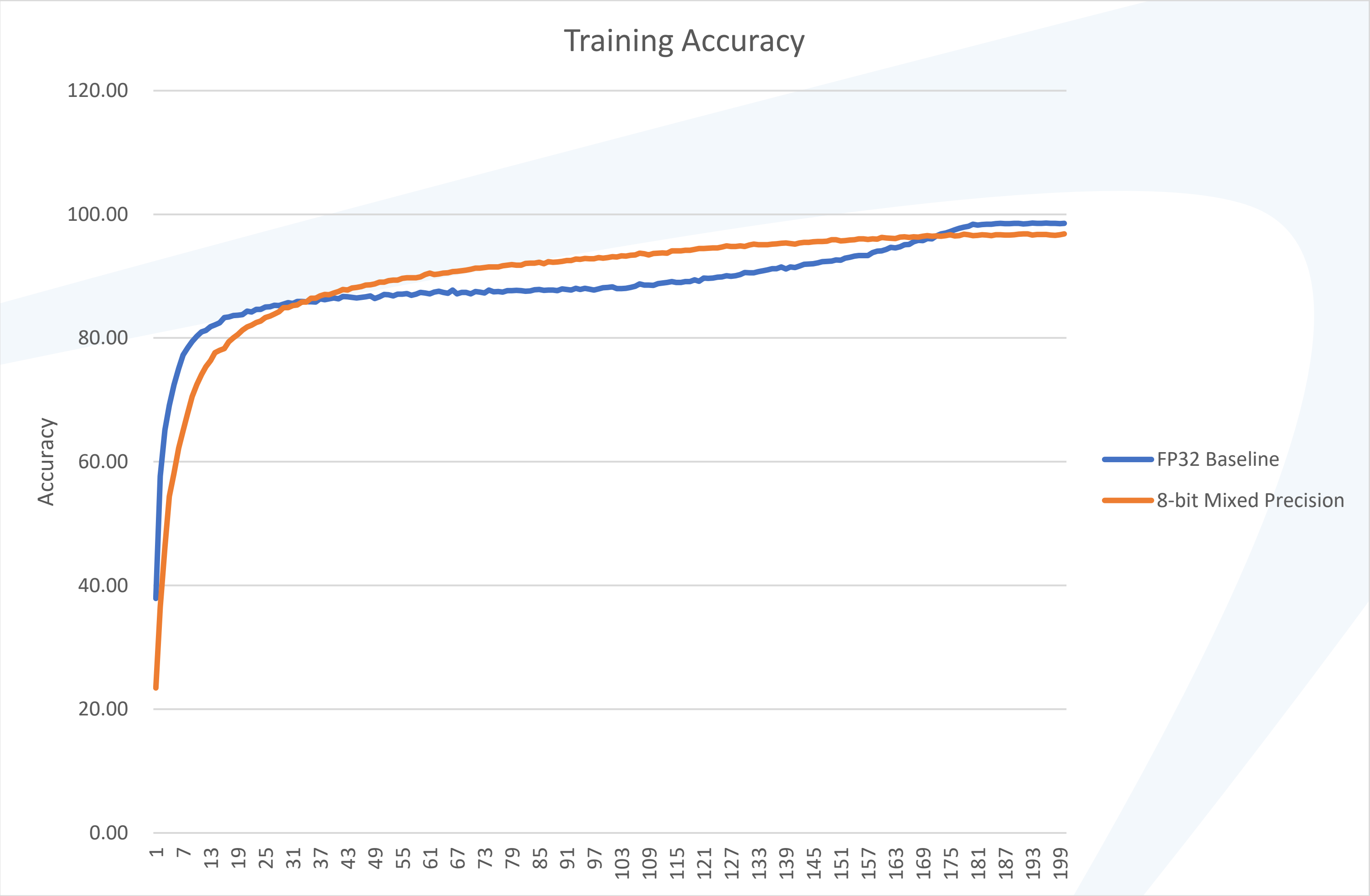
- Quantization is an effective method for reducing memory footprint and inference time of Neural Networks.

- **Quantization Aware Training**
  - **Mixed Precision Training**
    - master copy of weights and gradient momentum in BF16
    - Loss and per-layer gradient scaling
  - Supported Lows Precision Data Types: BF16, Float8, Float4

- **Post Training Quantization Inference**
  - Supported low precision data types: INT8, Float8, Float4

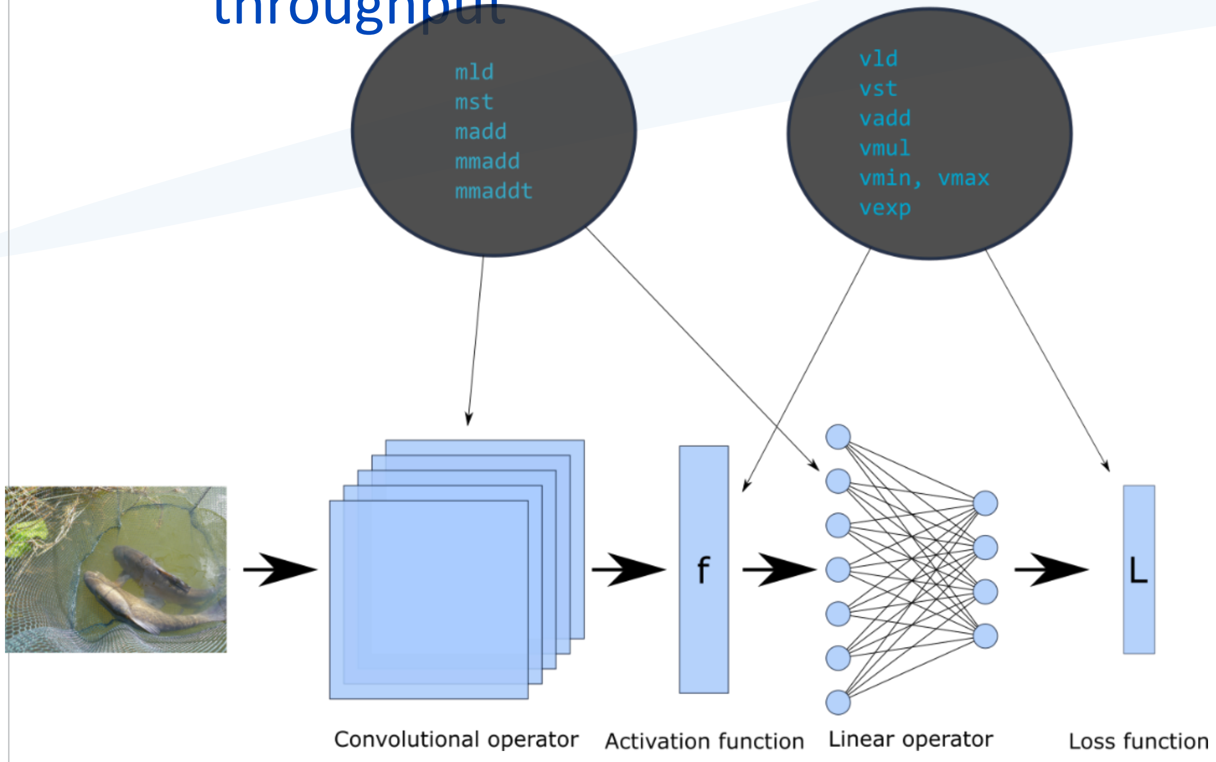
- Ultra-low precision quantization could lead to significant degradation in model accuracy. A promising method to address this is to perform mixed-precision quantization, where more sensitive layers are kept at higher precision. However, the search space for a mixed-precision quantization is exponential in the number of layers.

- Hessian based framework, with the aim of reducing this exponential search space by using second-order information. Hessian based framework provides a method for automatic bit precision selection of different layers without any manual intervention by analyzing sensitivity of loss surface ith respect to bit precision of different layers to bit precision

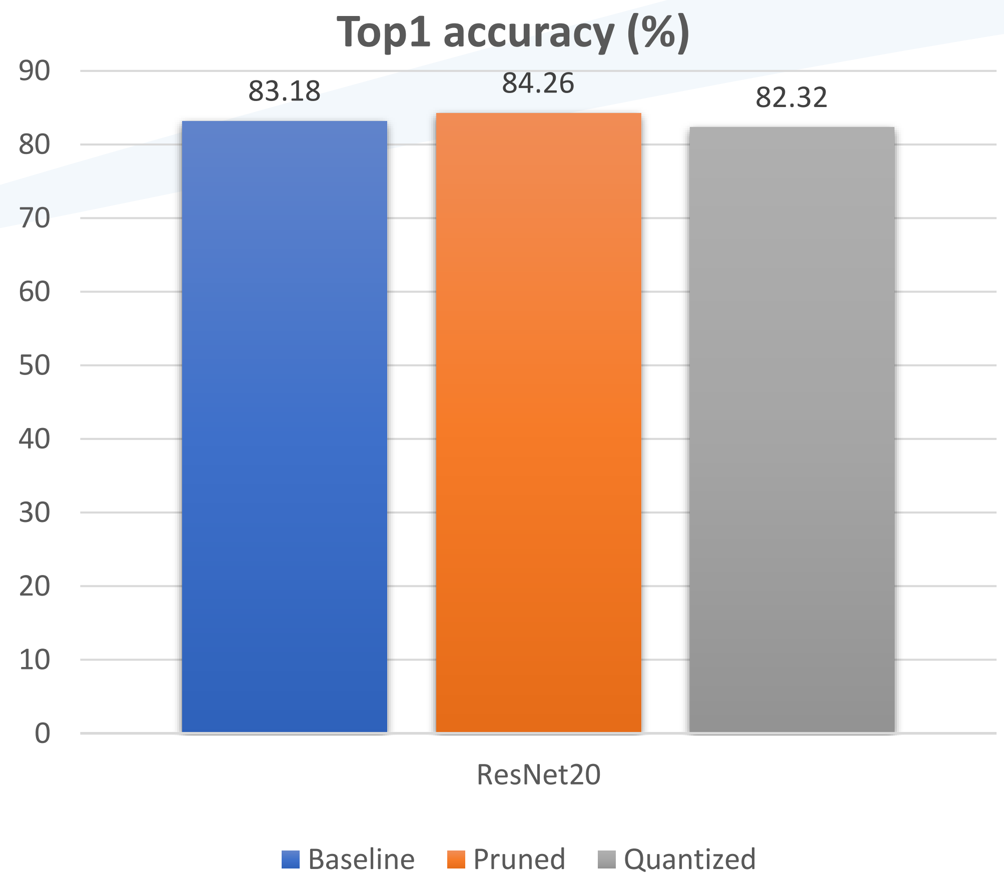

Training Accuracy
Accuracy
120.00
100.00
80.00
60.00
40.00
20.00
0.00
1 7 13 19 25 31 37 43 49 55 61 67 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157 163 169 175 181 187 193 199
FP32 Baseline
8-bit Mixed Precision

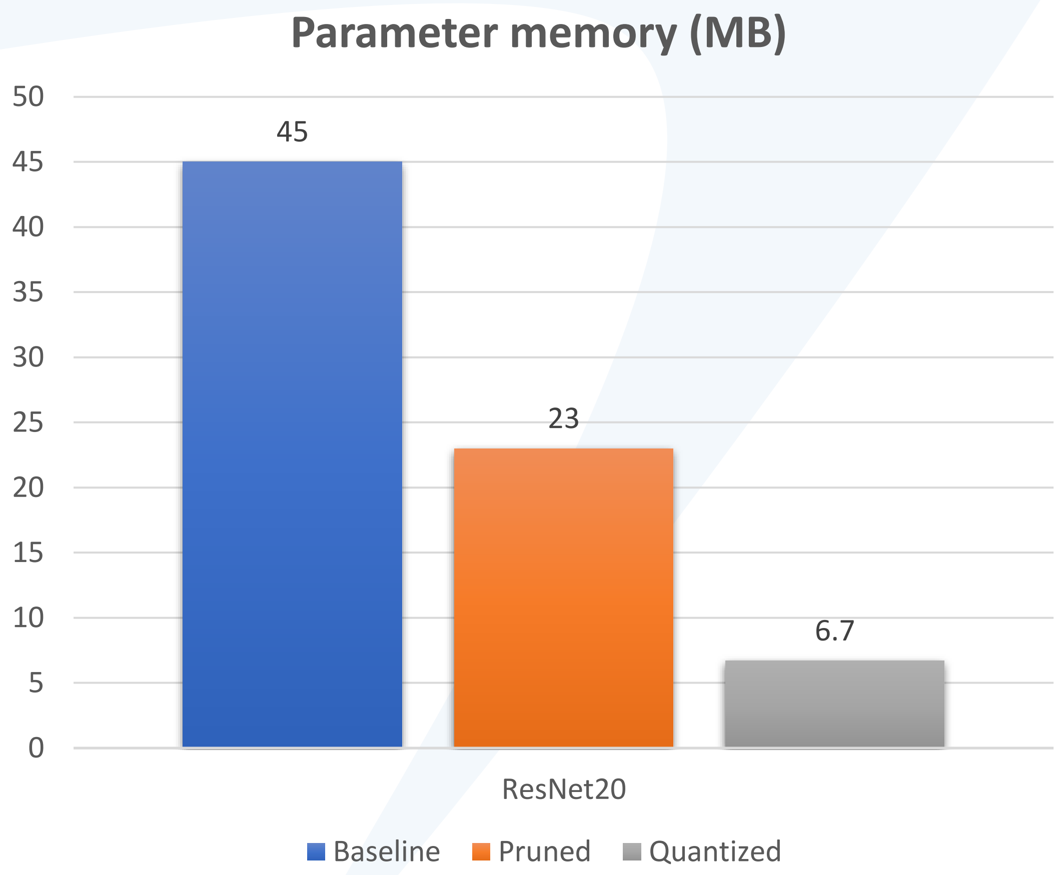
Efficient AI inference

- Compressed and quantized models exploiting the Prodigy low precision data types for vector instructions and matrix multiplication and compressed matrix multipliers while still maintaining high accuracy, low latency and high throughput

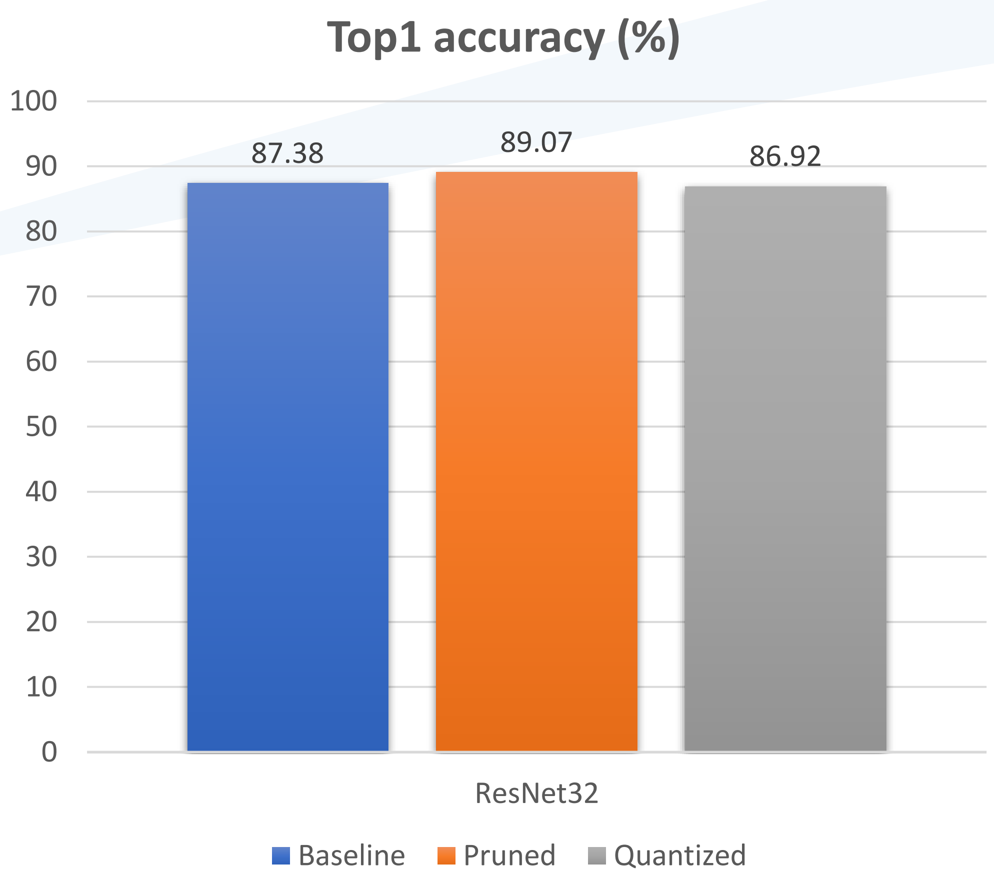


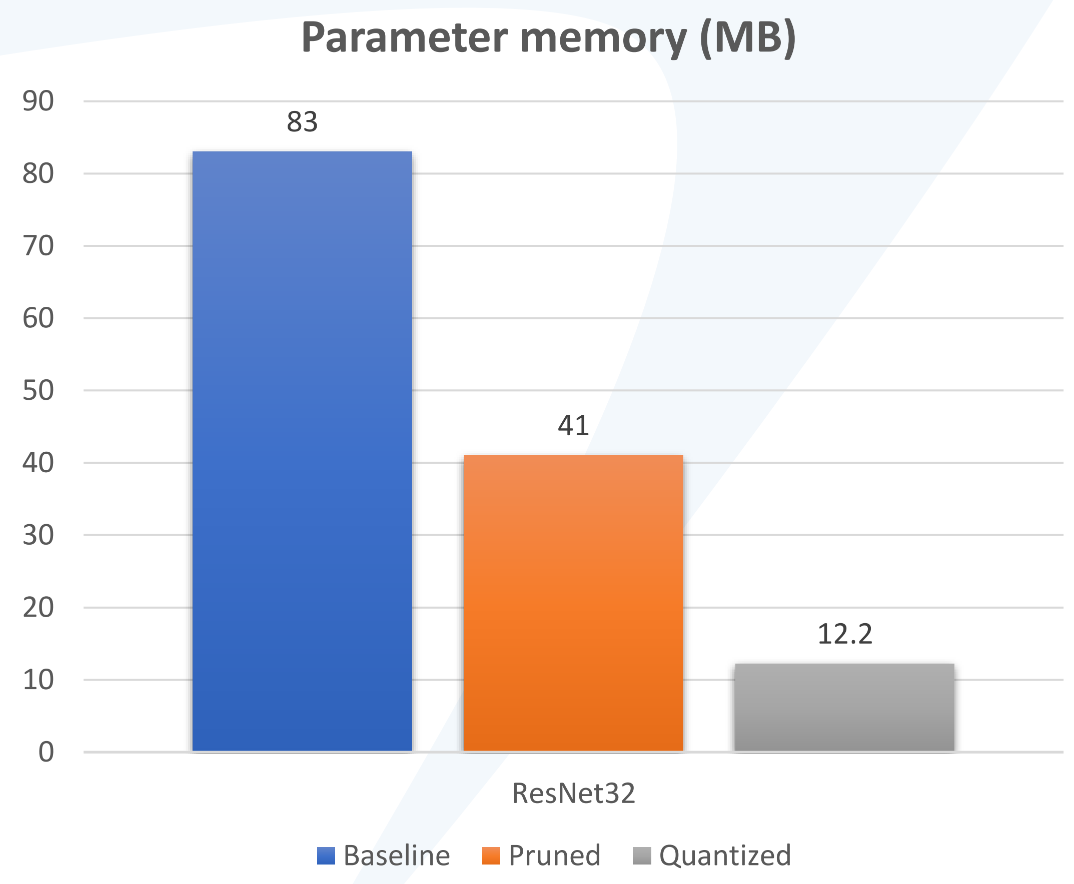
ResNet20 INT4W/INT8A quantization

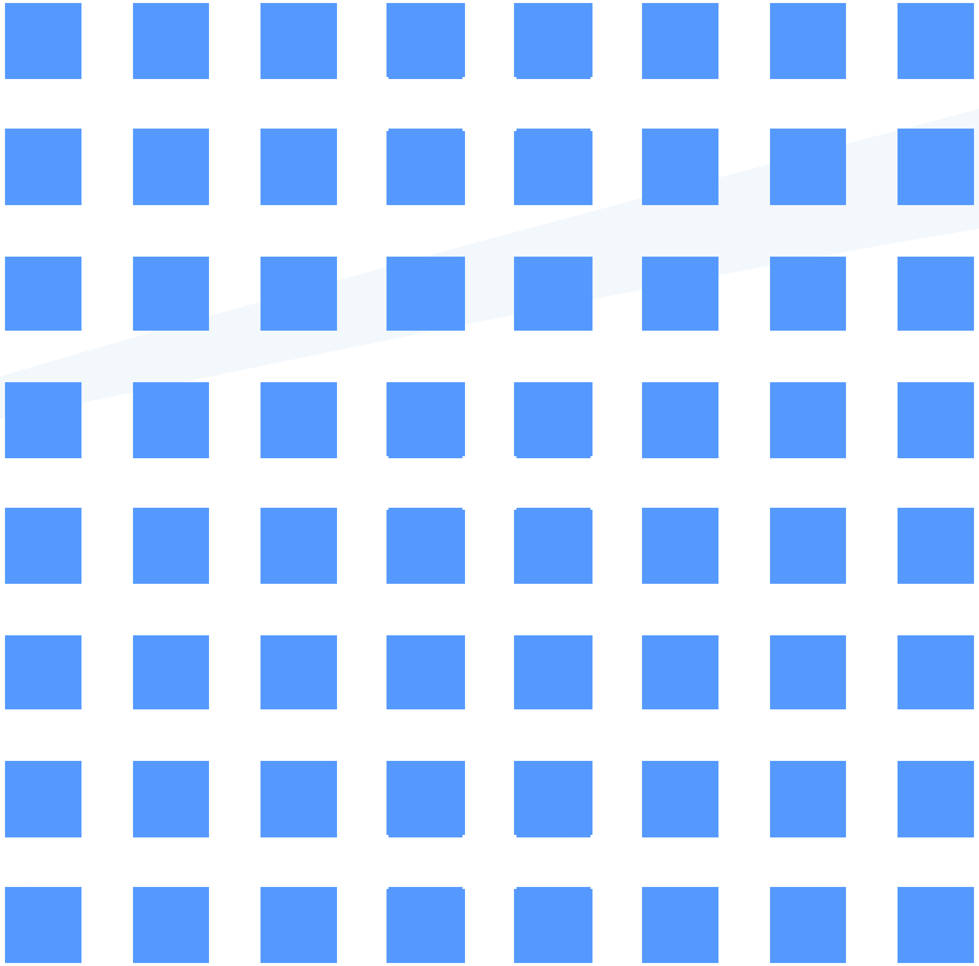




ResNet32 INT4W/INT8W quantization



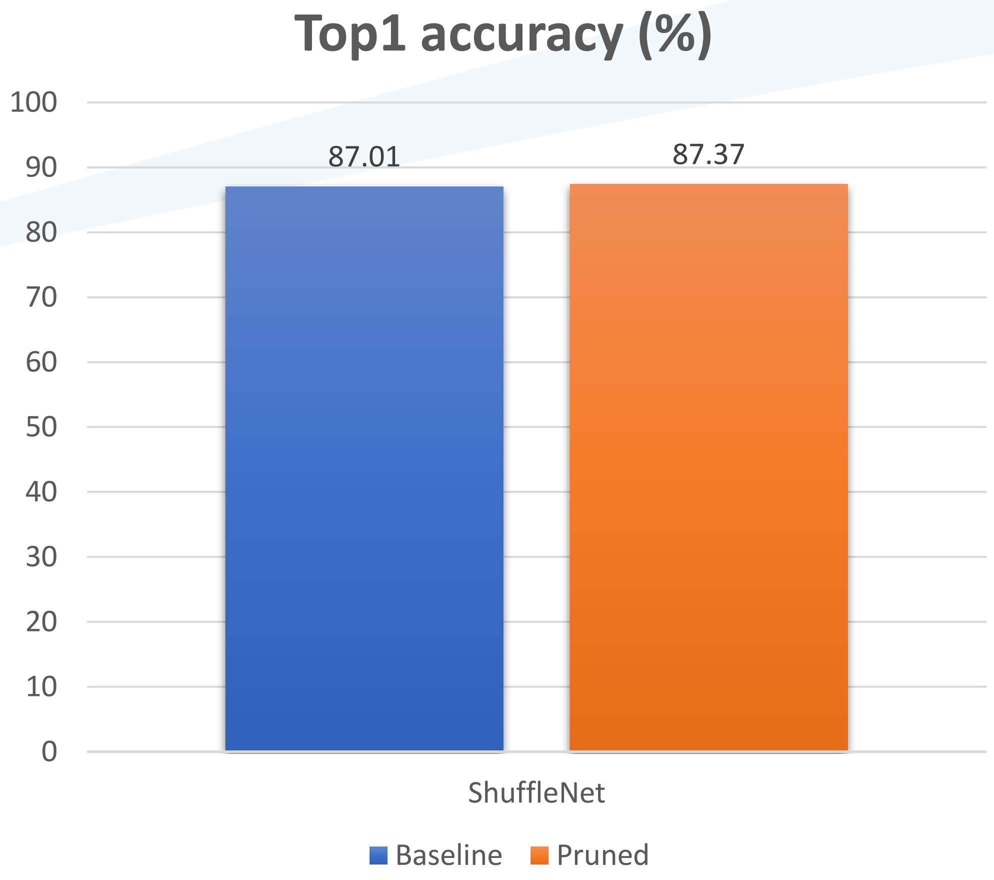


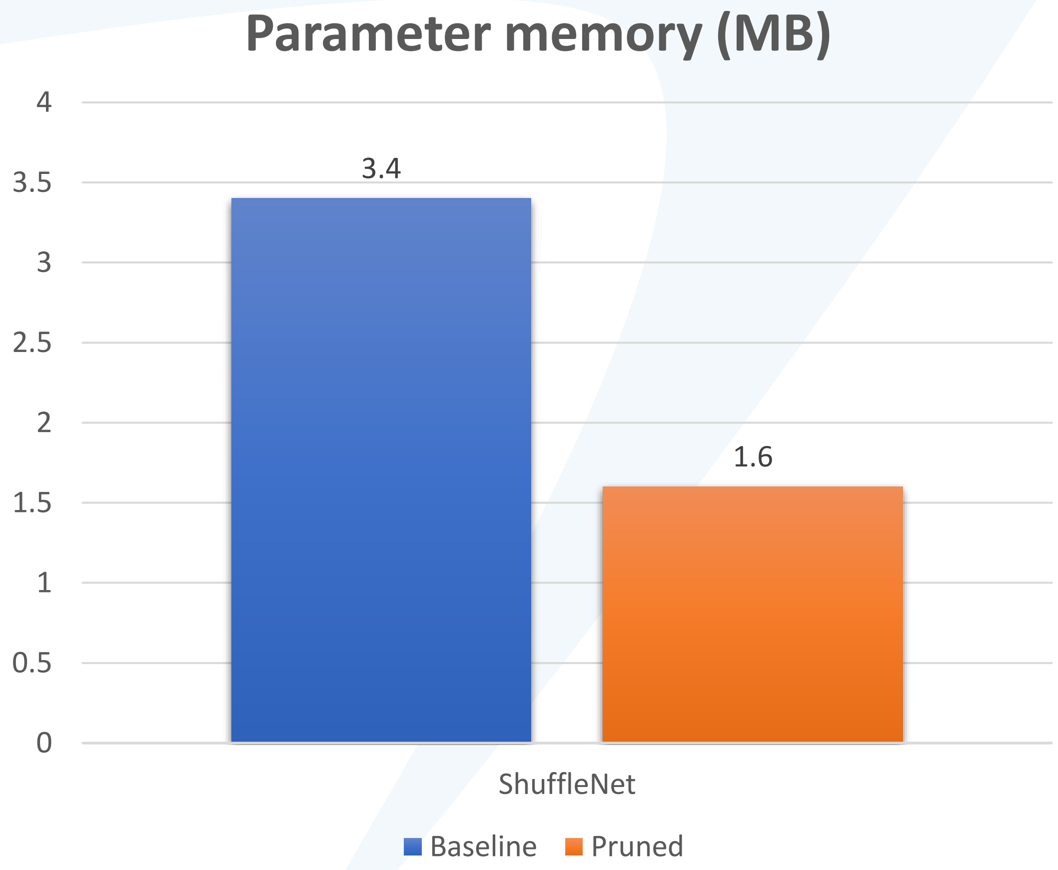


Compression, Pruning

- Magnitude based weight pruning – N:M block pruning
- Lottery Tickets – pruning weights and retrain
- Support for sparse matrix operations (block sparsity) optimized for compressed networks/models thus reducing memory and computation requirements
- Specific instructions for efficient storing and loading sparse matrices and for sparse structured matrix multiplication

ShuffleNet pruning test





Tachyum™