Predicting Protein Folding on Intel's Data Center GPU Max Series Architecture (PVC)

Dhani Ruhela, Aaditya Saxena,
Madhavan Prasanna

Table of Contents

Introduction Openfold and CASP Challenege 02

Alpha Fold and Open Fold Architecture 03

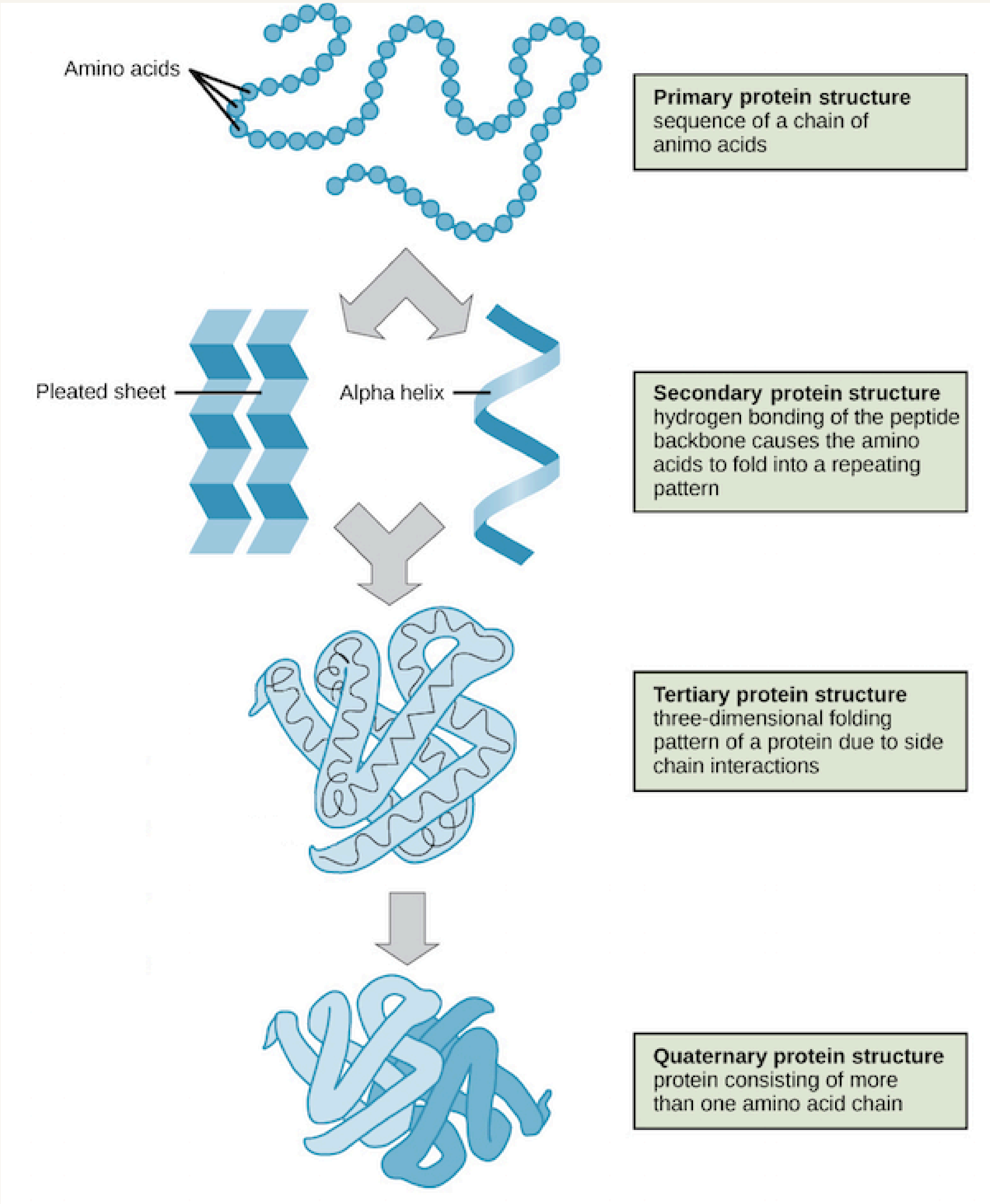
Alpha Fold and Open Fold Implementation 04

Experiments and Performance Results 05

Conclusions 06

Understanding Proteins

Proteins are fundamental to all living organisms, including cells and viruses. They consist of amino acids that fold into unique 3D structures, which directly determine their function in cells. Predicting protein folding accurately is crucial in biotechnology, medicine, and drug discovery.



Primary protein structure
sequence of a chain of animo acids

Secondary protein structure
hydrogen bonding of the peptide backbone causes the amino acids to fold into a repeating pattern

Tertiary protein structure
three-dimensional folding pattern of a protein due to side chain interactions

Quaternary protein structure
protein consisting of more than one amino acid chain

The Folding Puzzle

- First atomic structure of Protein revealed in 1960
- Dr. John Moult initiated the Critical Assessment of Techniques for Protein Structure Prediction(CASP) challenge in 1994.

Challenge:

Folding code

- ○ What balance of interatomic forces dictates the protein structure for a given amino acid sequence?

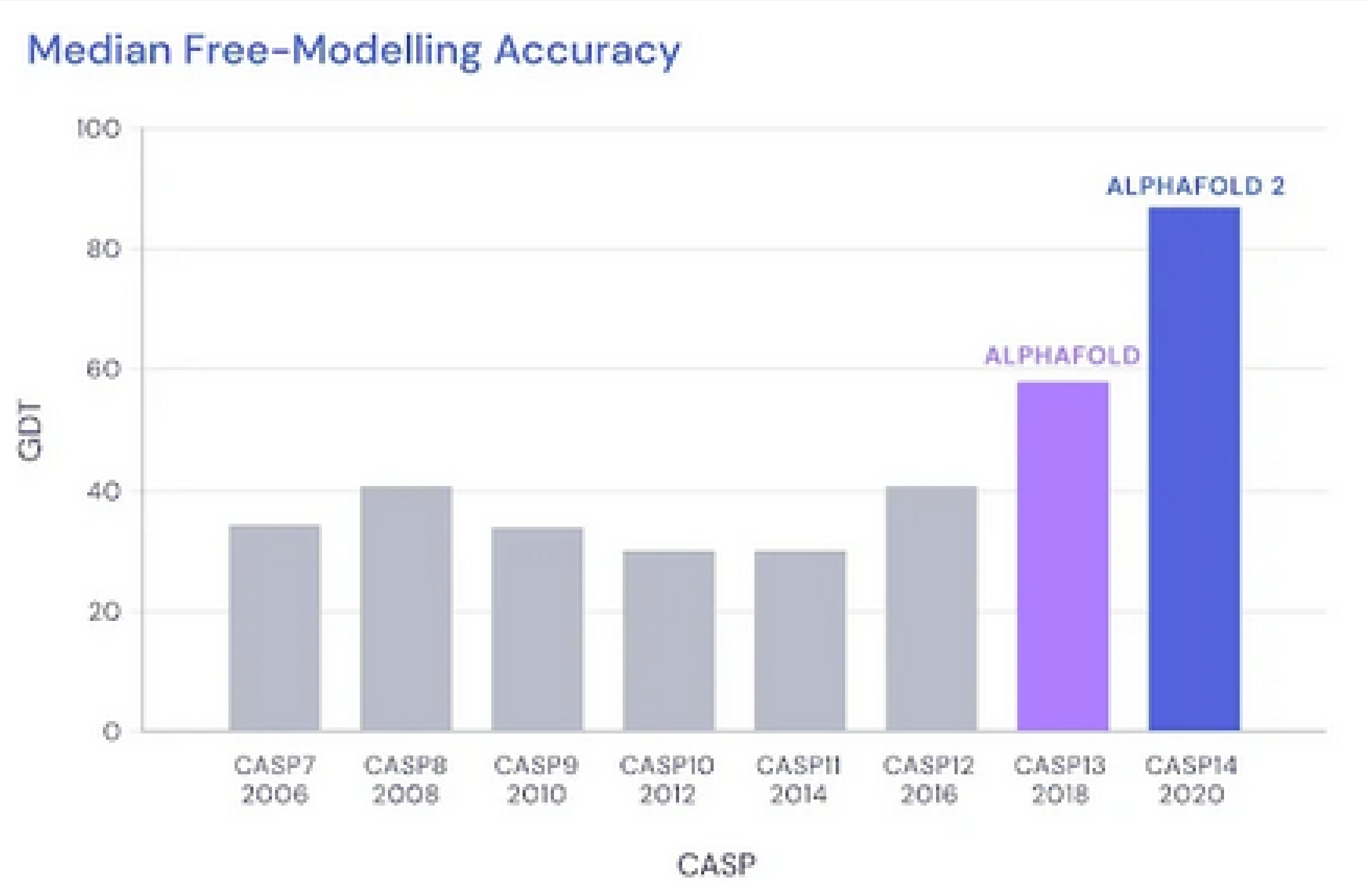
Structure prediction

- How to predict the 3-D structure of proteins from its 1-D amino acid sequence.

Folding process

- How do the amino acids fold quickly in an environment

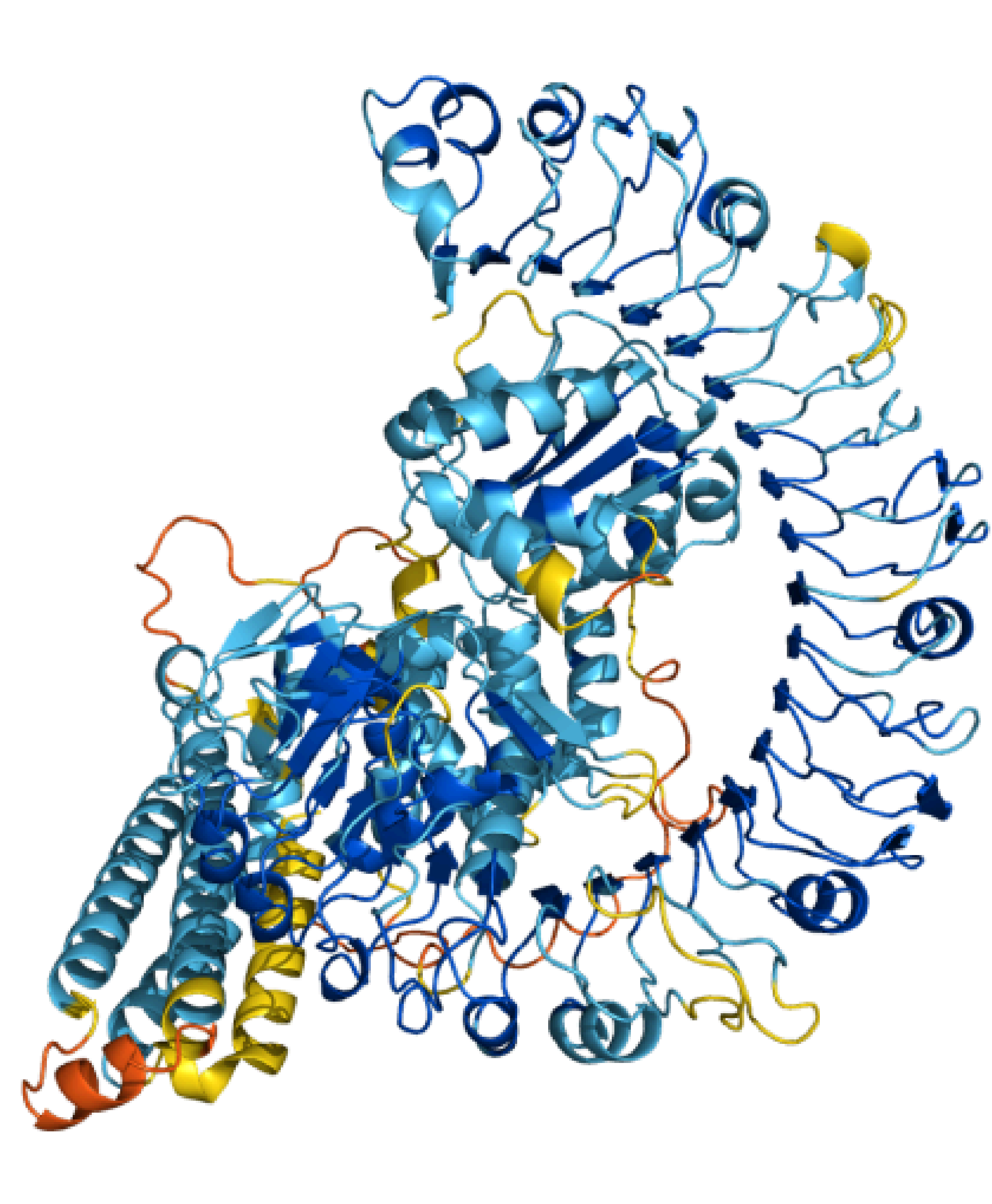
Progress of Protein Structure Detection



AlphaFold's Impact

Developed by Google's DeepMind team that predicted 88 out of 97 protein structures in CASP14 in 2020 with accuracy of 0.92 GDT.

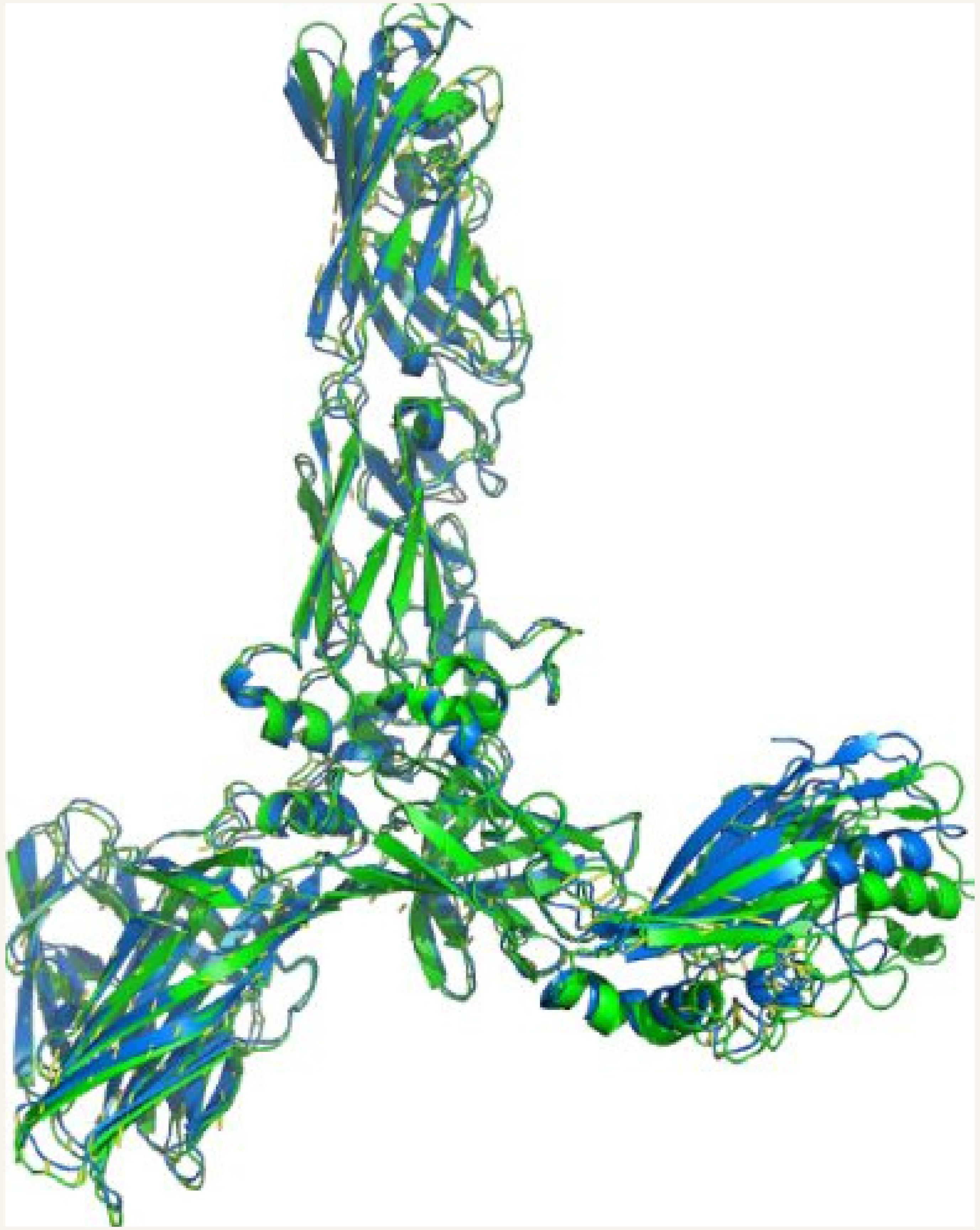
AlphaFold has predicted over 200 million protein structures – nearly all catalogued proteins known to science, saving millions of dollars of research time.



AlphaFold's Impact

Usecases:

- Medical:
  - Accelerating the fight against malaria
  - Paving the way for potential Parkinson's treatments
  - Racing against drug-resistant bacteria
- Environmental:
  - Breaking down plastic pollution
  - Increasing honeybees' chances of survival



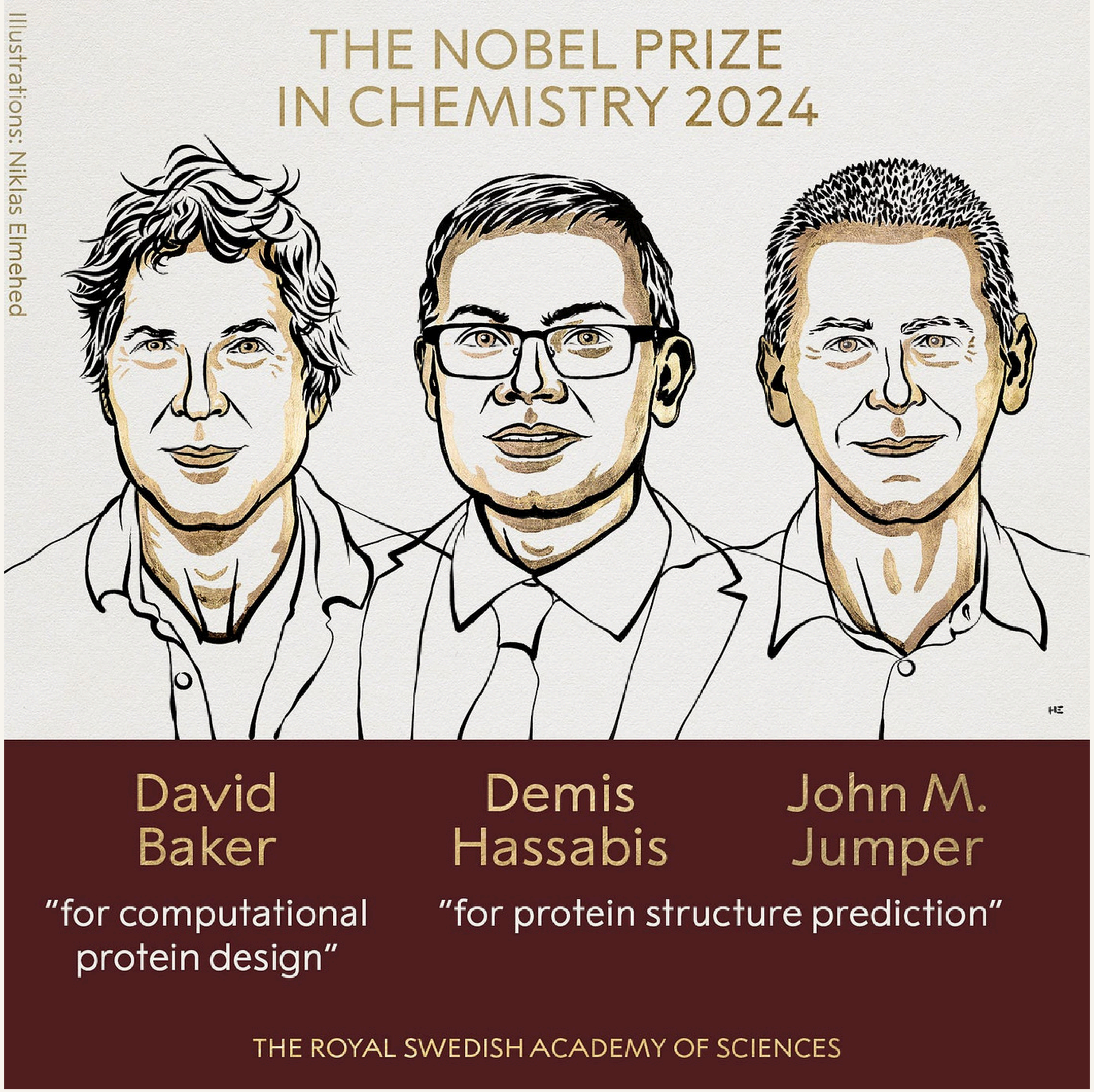
Celebrating Excellence in AI and Science 2024

Honoring the AlphaFold Team

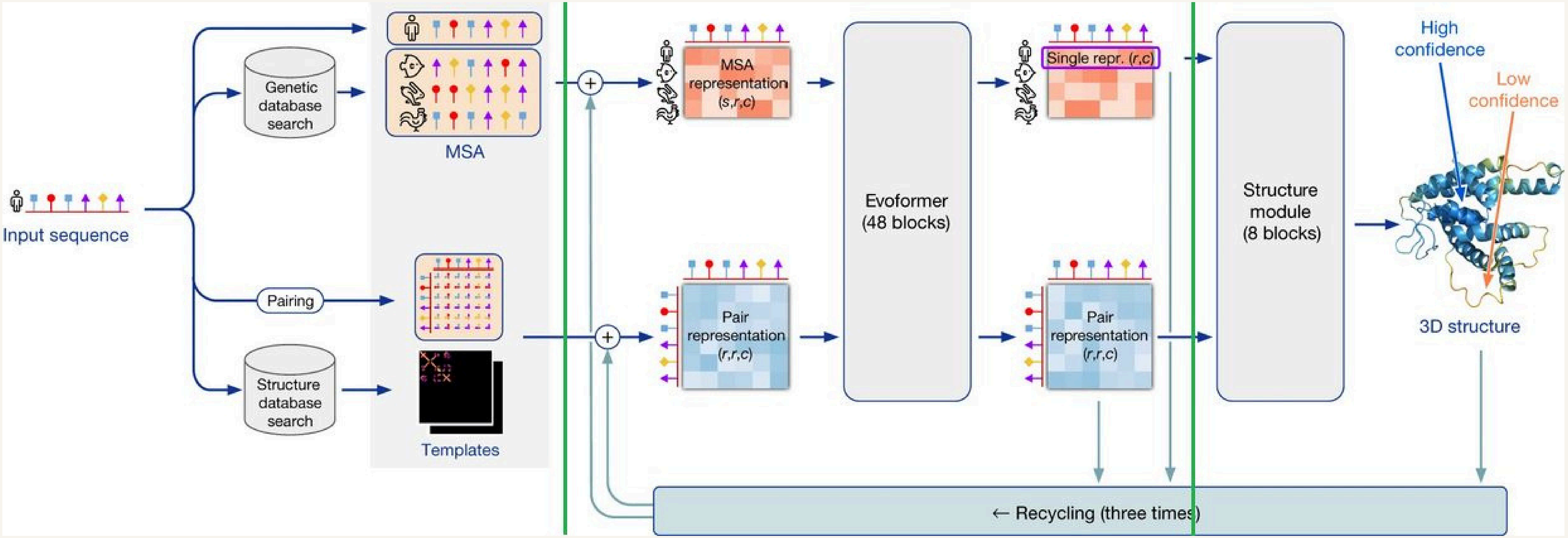
AlphaFold's revolutionary AI predicts protein structures with unmatched accuracy, unlocking new possibilities in biology and medicine.

This breakthrough accelerates research, offering immense potential for advancements in disease treatment and drug discovery.

A heartfelt congratulations to the AlphaFold team for their extraordinary contributions to science and AI!


Illustrations: Niklas Elmehed
THE NOBEL PRIZE
IN CHEMISTRY 2024
David Baker
"for computational protein design"
Demis Hassabis
John M. Jumper
"for protein structure prediction"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

AlphaFold Architecture


Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Openfold and MLCommons

Alphafold Limitations

- Lack of code and data for training new models
- Expansive computation for the voluminous dataset

OpenFold : Developed by Opensource community to overcome Alphafold limitations and provide a fast, memory efficient and trainable implementation.

Adopted by MLCommons HPC benchmarks in 2023
Received multiple submissions from worldwide organizations including NVIDIA.

OpenFold Implementation on Intel PVC GPUs

- First distributed implementations on Intel PVC GPUs
- Added
  - import intel_extension_for_pytorch as ipex
  - import oneccl_bindings_for_pytorch
- “cuda” Device changed to “xpu”
- All cuda references removed
- Added support for floating point precisions
- Added Torchrun launcher support

Experimental Setup

Hardware:

- Intel Xeon CPU MAX 9480 ("Sapphire Rapids HBM")
  - 96 cores on two sockets (2 x 48 cores)
  - 128 GB HBM 2e and 512 GB DDR5 Memory
  - 4x Intel Data Center GPU Max 1550s "Ponte Vecchio" (PVC)
  - 100GB/sec Omni-Path (OPA) network with a fat tree topology

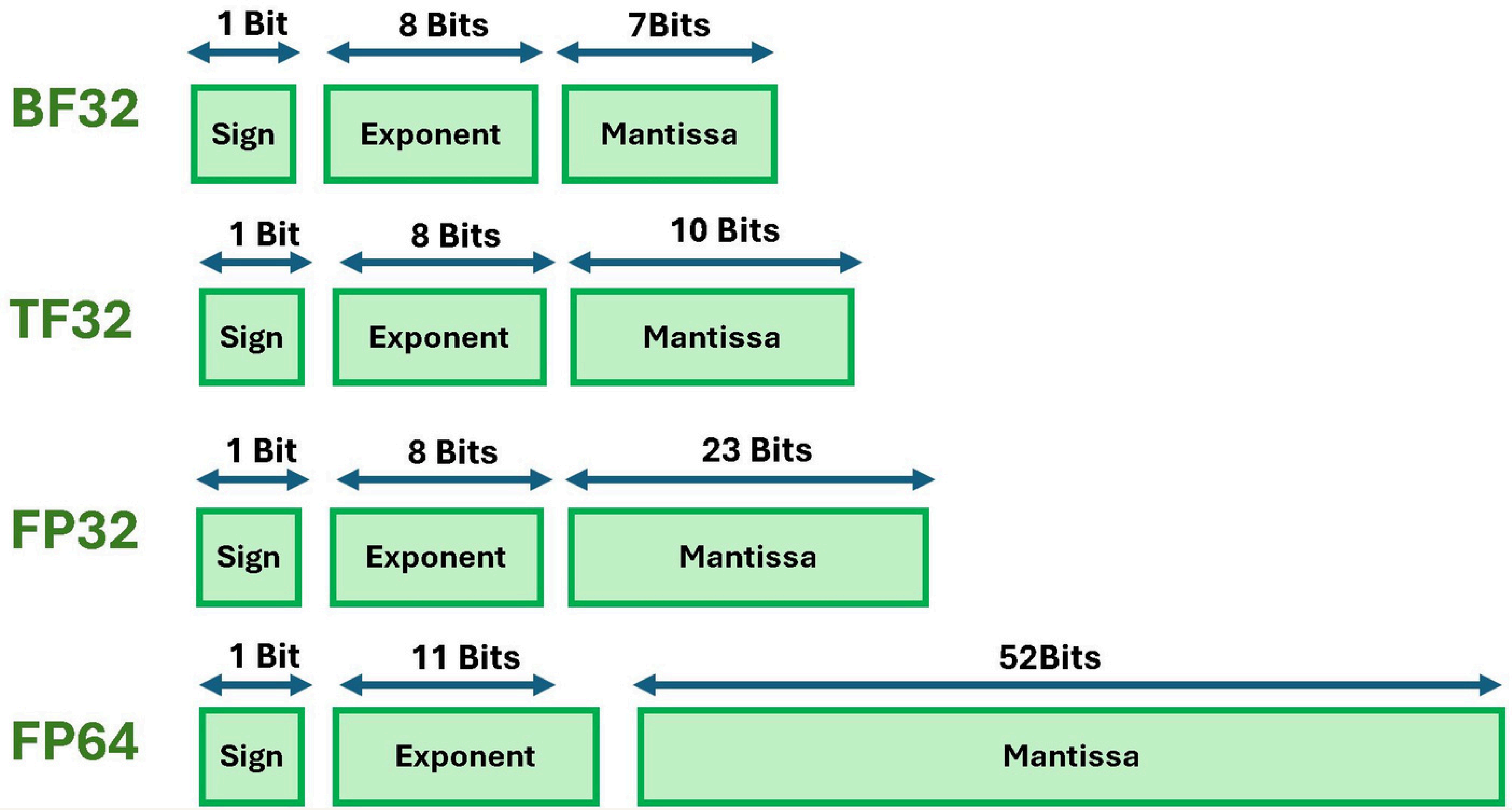
Software:

- Intel OneAPI 2024.1, PyTorch 3.9.18, IPEX 2.1.30.
- Modified OpenFold code from MLCommons GitHub.

Challenges

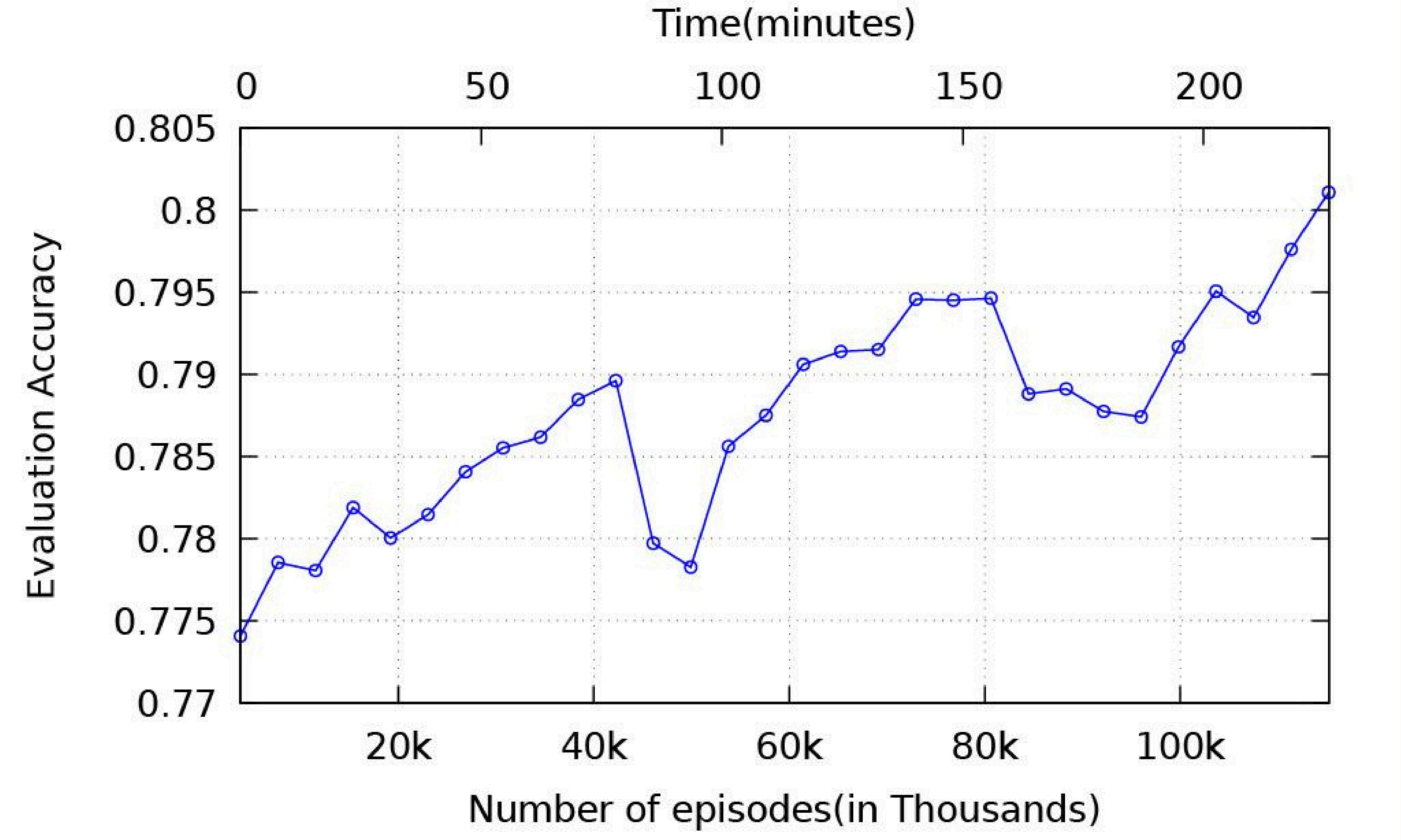
- **Setting up the software environment with Intel Conda / Pip channels**
- **Bugs in software libraries leading to poor performance**
- **Unstable APEX library leading to node Failures**

Floating point Precision



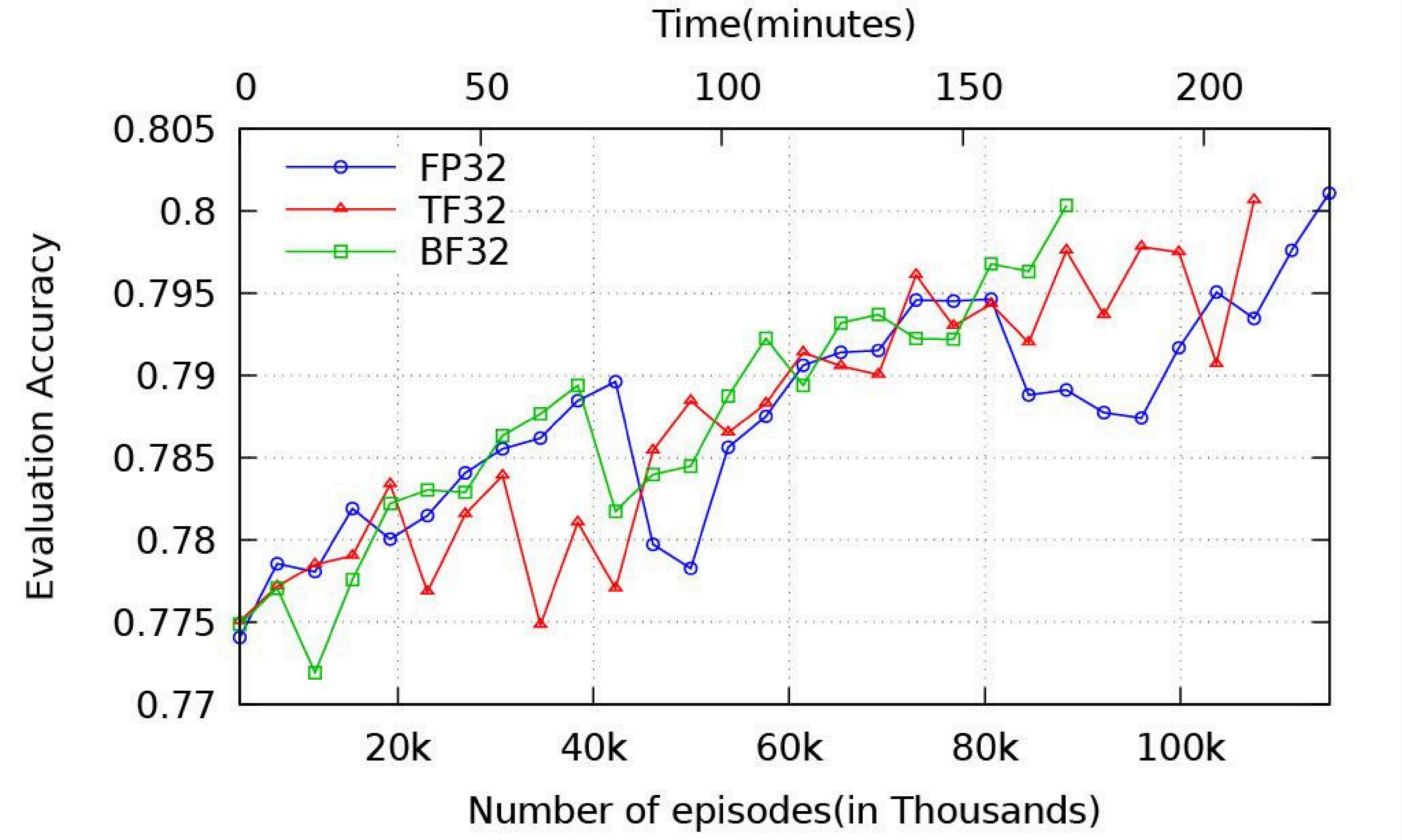
Baseline Performace

Observation: The benchmarks ran for 1200 iterations containing 115,200 epochs and took 229 minutes to 0.80 target accuracy.



12 PVC Nodes with 4 GPUs per node.
Each GPU supports two streaming units

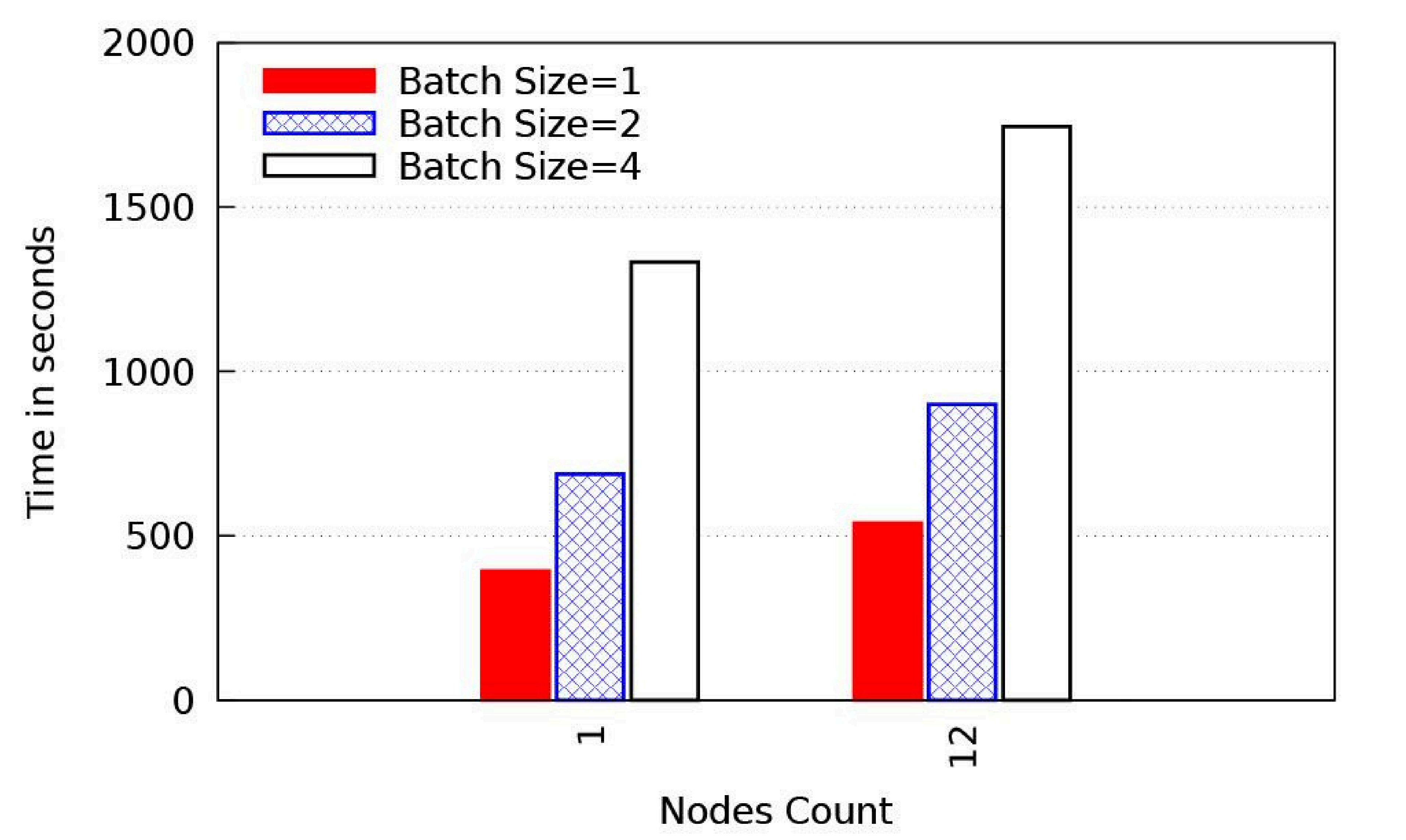
Floating Point Precision Benefits



Observations:

1. Reduced precision formats (BF16, TF32) speed up training while maintaining accuracy.
2. BF16 provided a 23% reduction in training time compared to FP32.

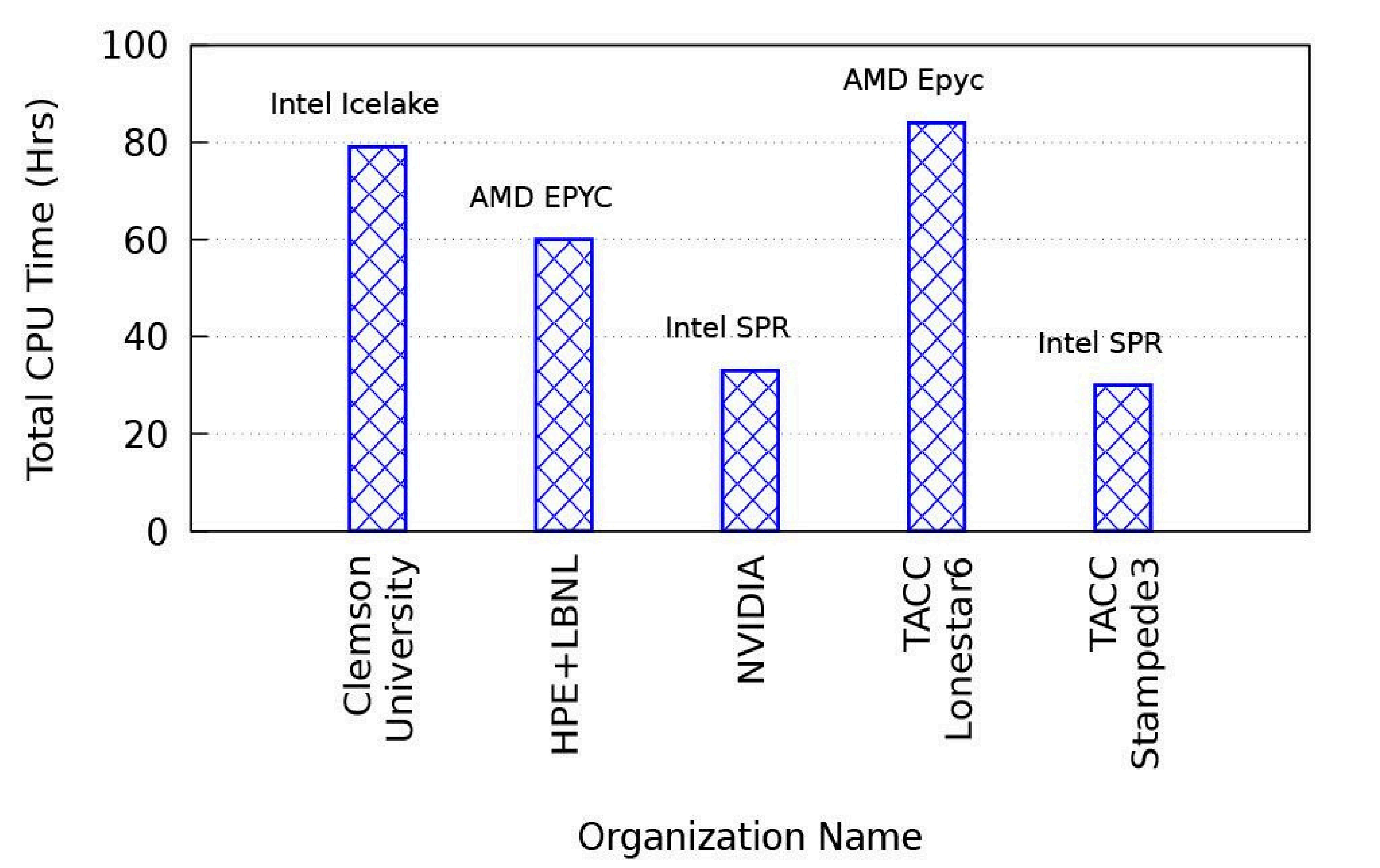
Optimal Batch Sizes for Efficiency

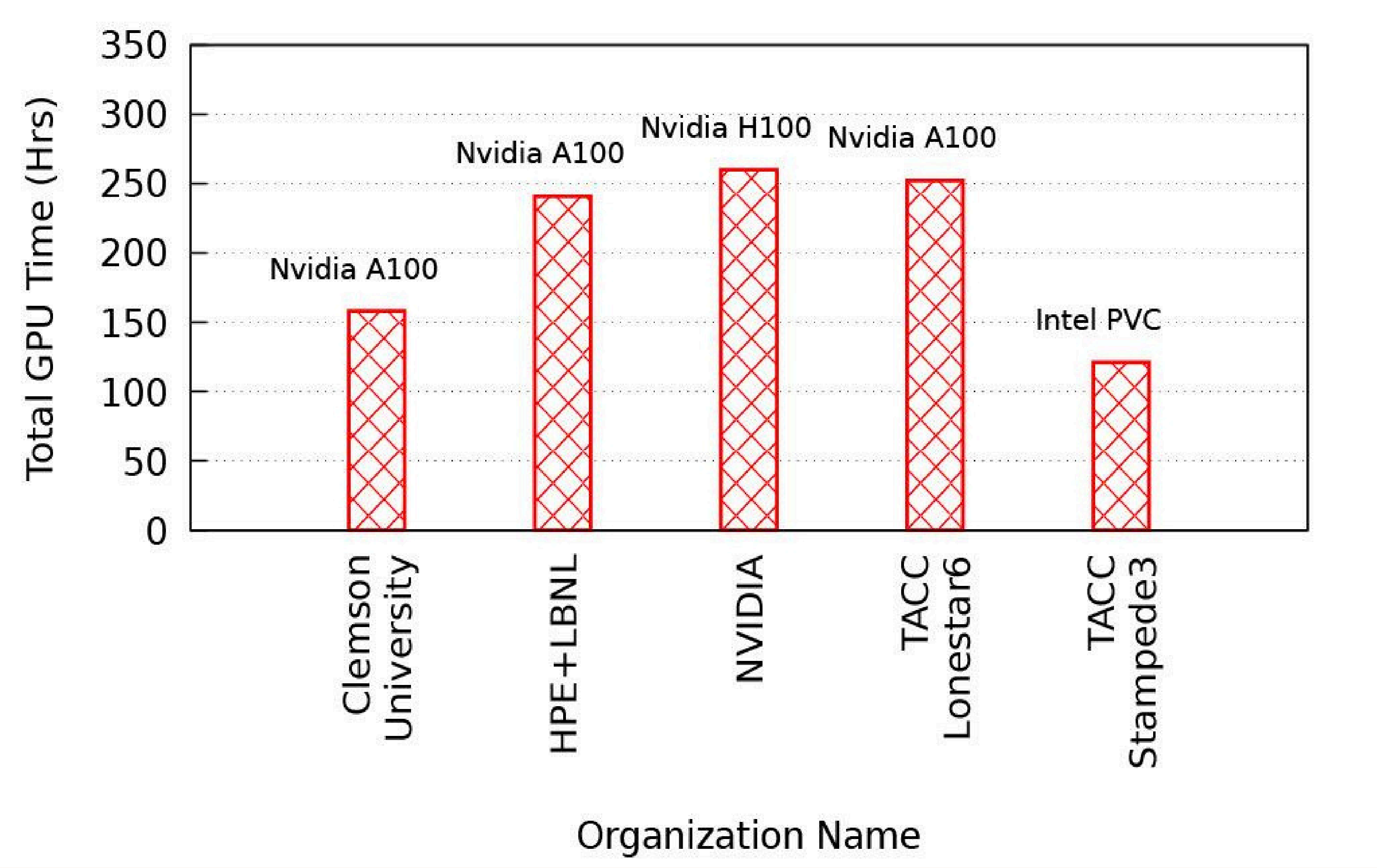


Observations:

- Larger batch sizes increased memory load but offered diminishing returns.
- Optimum performance observed with smaller batch sizes and balanced worker counts.
- Data loading time was insignificant compared to training time.

PVC Efficiency on Large-Scale ML

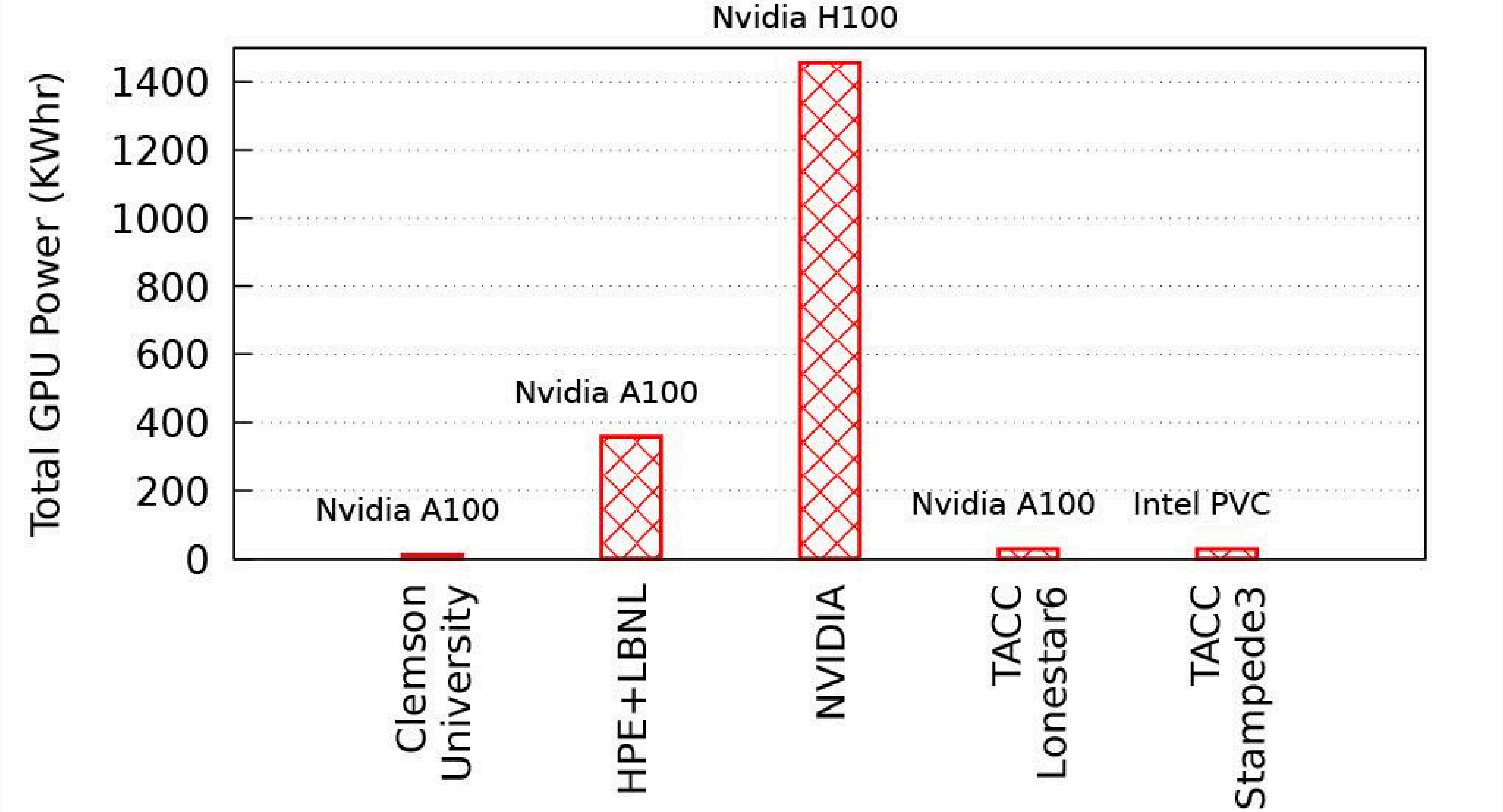




Observations:

1. Intel PVC used 4x fewer GPUs and 2x less total GPU hours compared to NVIDIA H100.
2. Demonstrated superior efficiency for large-scale ML workloads.

Energy efficiency and Cost Analysis



Observations:

1. Energy consumption of PVC was 30 kWhr compared to 1456 kWhr for NVIDIA H100 on Eos.
2. Significant cost savings (50x) achieved with lessers GPUs.

Key Findings and Conclusions

OpenFold successfully ported to Intel’s PVC architecture.

Performance optimizations led to faster training and significant energy savings.

PVC presents a viable alternative to NVIDIA GPUs for AI workloads.

Thank You

Acknowledgements

Texas Advanced Computing Center for providing resources.

Thanks to mentors and collaborators for support.

Thanks to Intel for technical support.