



NVIDIA PAX-HPC Workshop

17th January 2024

Paul Graham | Senior Solutions Architect

pgraham@nvidia.com

NVIDIA DEVELOPER PROGRAM

JOIN THE COMMUNITY THAT'S CHANGING THE WORLD

TOOLS AND TECHNOLOGY

- Get exclusive access to an extensive library of NVIDIA software, spanning all of NVIDIA's technology platforms.
- Save time with ready-to-run, GPU-optimized software, model scripts, and containerized apps from the NVIDIA NGC catalog.
- Participate in early access programs where you can be one of the first to experience the latest NVIDIA technology.
- If you're an educator or researcher, access industry-leading hardware and software through the NVIDIA Hardware Grant Program.

TRAINING

- Choose from a broad catalog of training options through the NVIDIA Deep Learning Institute and NVIDIA Academy (together known as NVIDIA Training)
- Get unlimited access to NVIDIA On-Demand, the home for NVIDIA resources from GTC and other leading industry events.
- Take advantage of research papers, technical documentation, developer blogs, webinars, and other industry-specific resources.

COMMUNITY

- Network with like-minded developers, engage with GPU experts, and contribute to discussions in the developer forums.
- Apply and participate in open hackathons and bootcamps run in partnership with NVIDIA and OpenACC.
- Connect with NVIDIA experts through developer-focused webinars and instructor-led workshops.
- Attend exclusive meetups and events.

[Register now>>](#)

Agenda

- 10:15-10:45 Introduction to Heterogeneous Parallel Computing
- 10:45-11:00 Break
- 11:00-12:00 Key ways to accelerate applications
- 12:00-13:00 Programming for GPUs
- 13:00-13:45 Lunch
- 13:45-17:15 Hands-on practical

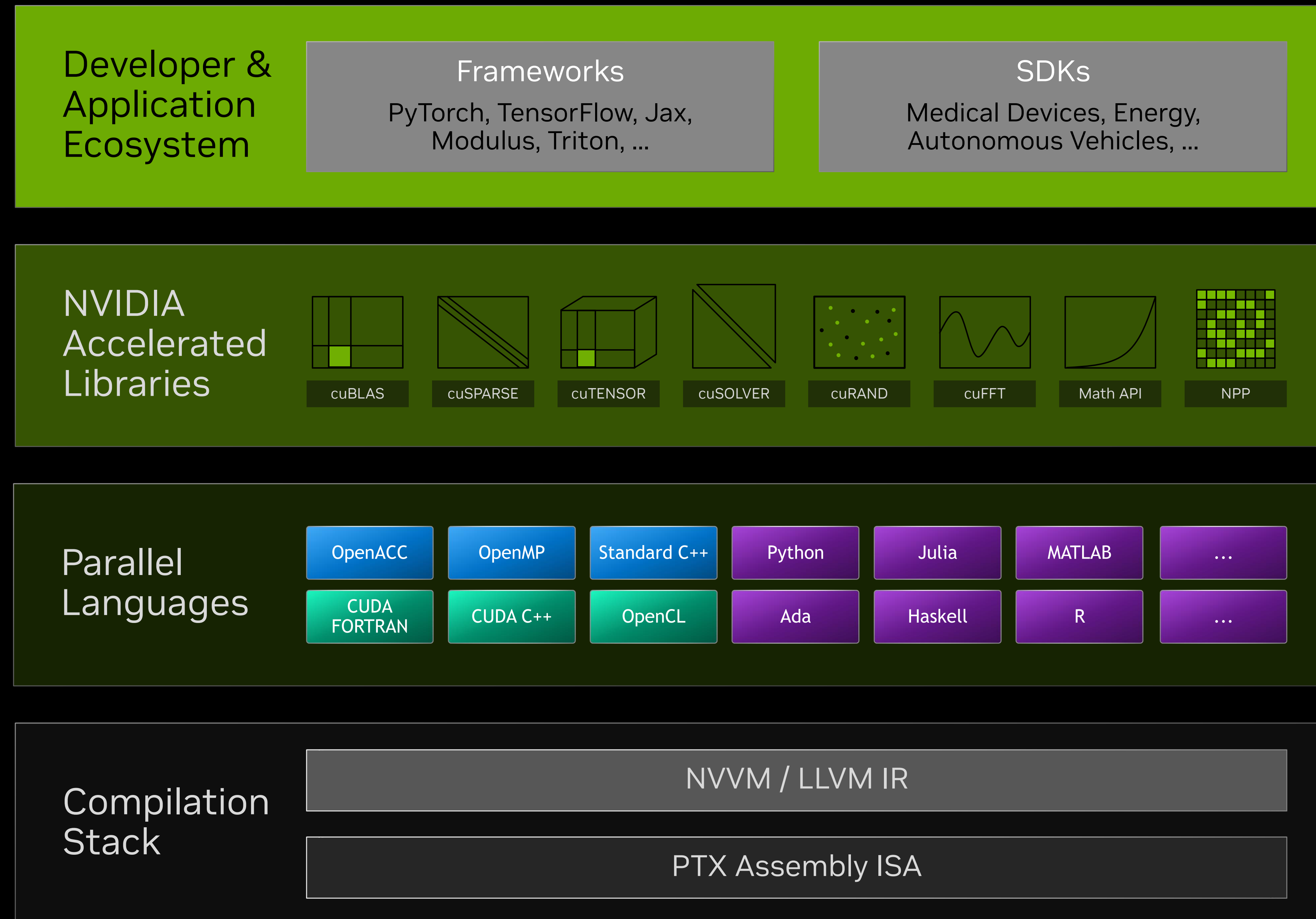
What is CUDA?

CUDA C++

It's Not Just C++



Target the Abstraction Layer That Works Best For Your Application



Ways to ACCELERATION

Applications and Frameworks

Libraries

“Drop-in”
Acceleration

Directives

Easily Accelerate
Applications

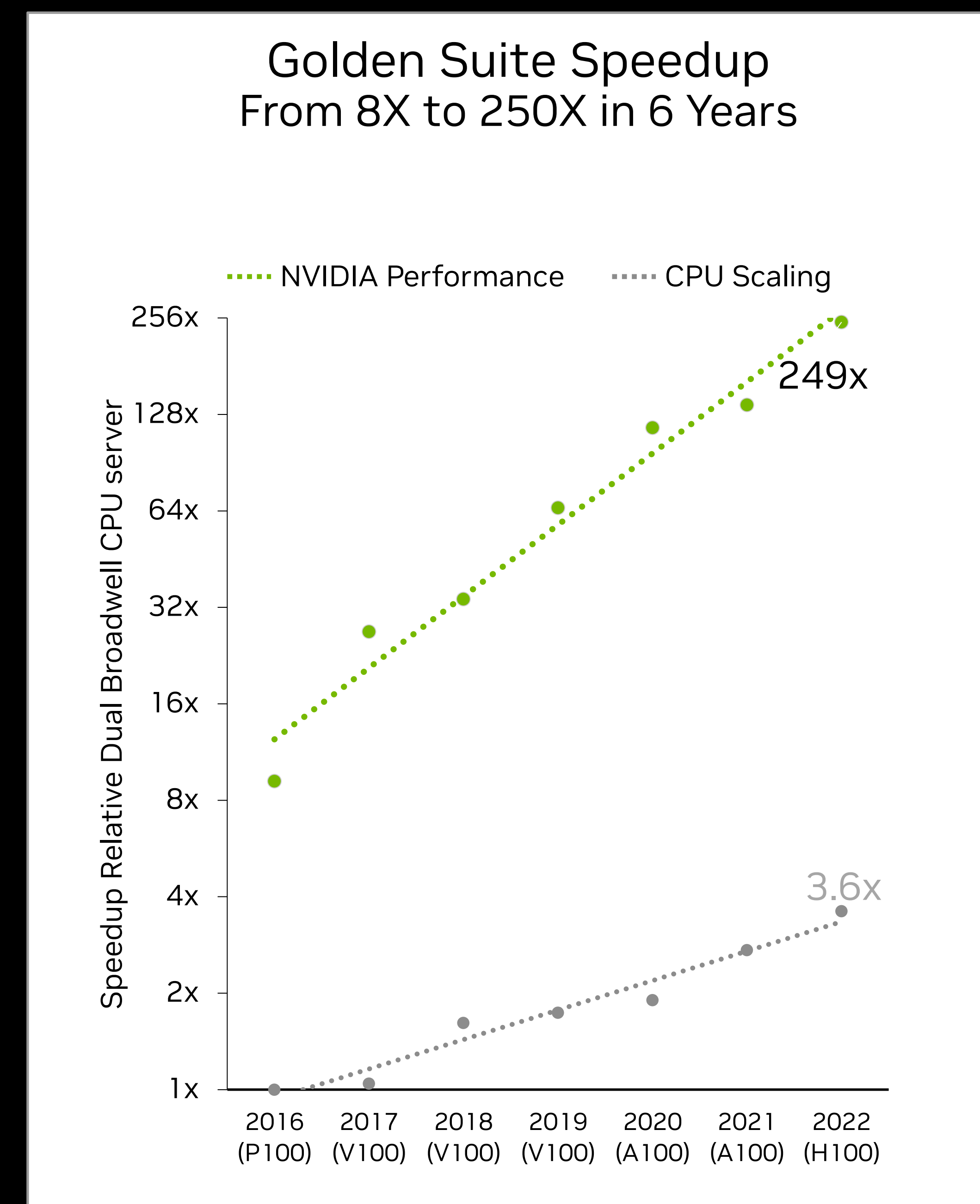
Programming
Languages

Maximum
Flexibility

GPU accelerated APPS and frameworks

- All major DL frameworks – PyTorch, TensorFlow etc
- Top 15 most used HPC apps globally
- Apps in a huge range of fields
 - Over 3000 apps in total - catalogue: [link](#)
- Domain-specific frameworks – robotics, vis, healthcare, smart cities etc
- [GROMACS](#) [VASP](#) LAMMPS, RELION, QE, NAMMD, SPECFEM3D ...

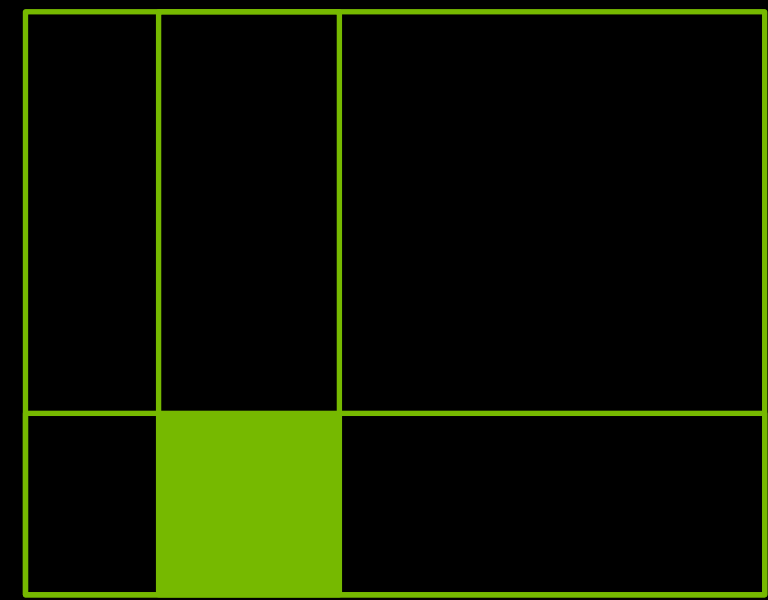
<https://developer.nvidia.com/hpc-application-performance>



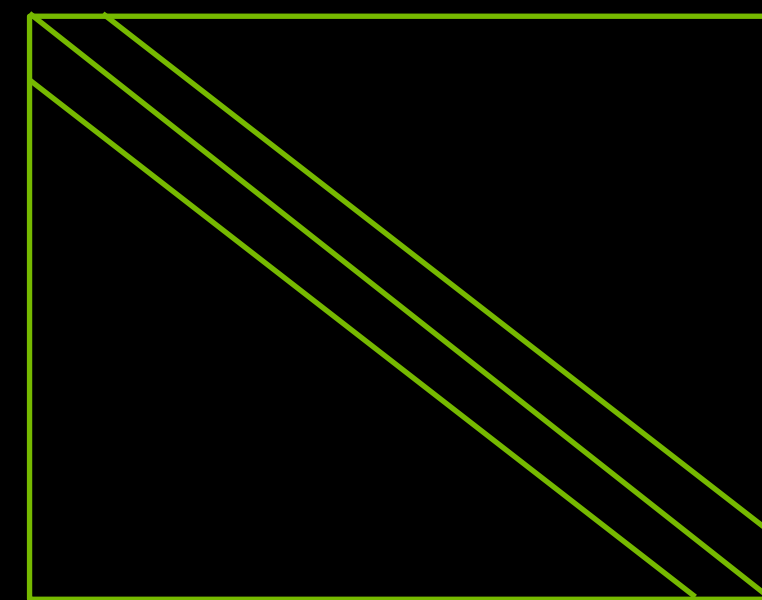
Geometric mean of application speedups relative to Dual Broadwell CPU
ESTIMATES ONLY] benchmark applications: Amber [PME-Cellulose_NVE], Chroma [HMC], GROMACS [ADH Dodec], MILC [Apex Medium], NAMMD [stmv_nve_cuda], PyTorch (BERT Large Fine Tuner), Quantum Espresso [AUSURF 112-jR]; TensorFlow [ResNet-50], VASP 6 [Si Huge]; Random Forest make_blobs (160000 x 64 : 10)

NVIDIA MATH LIBRARIES

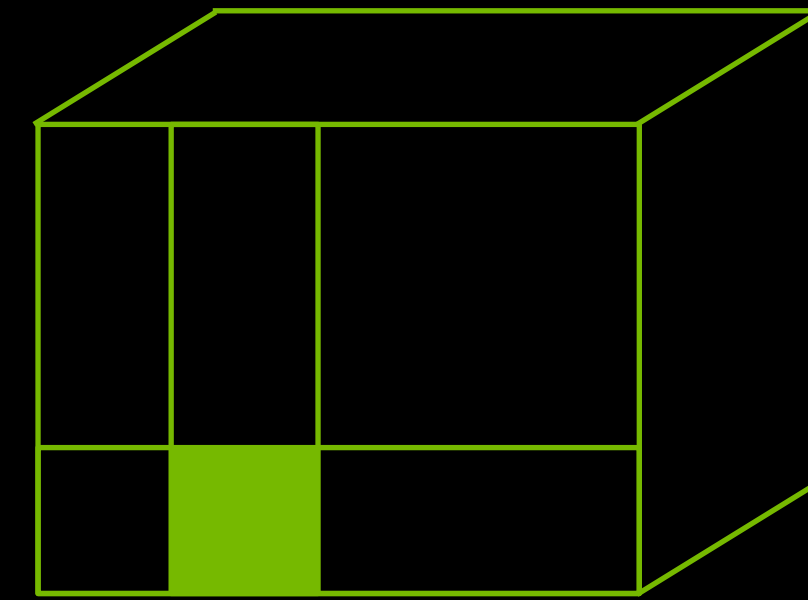
Linear Algebra, FFT, RNG and Basic Math



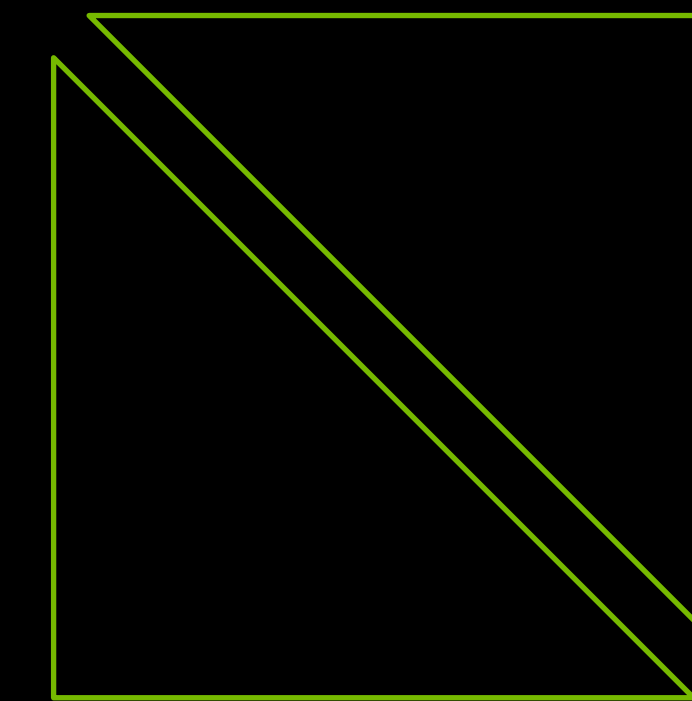
cuBLAS



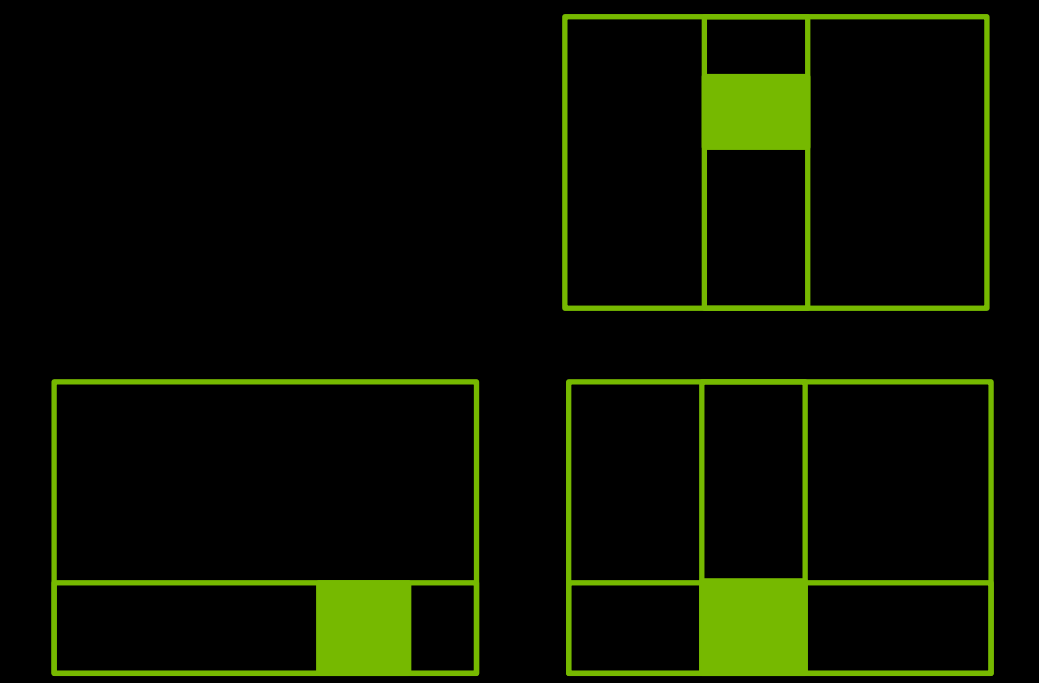
cuSPARSE



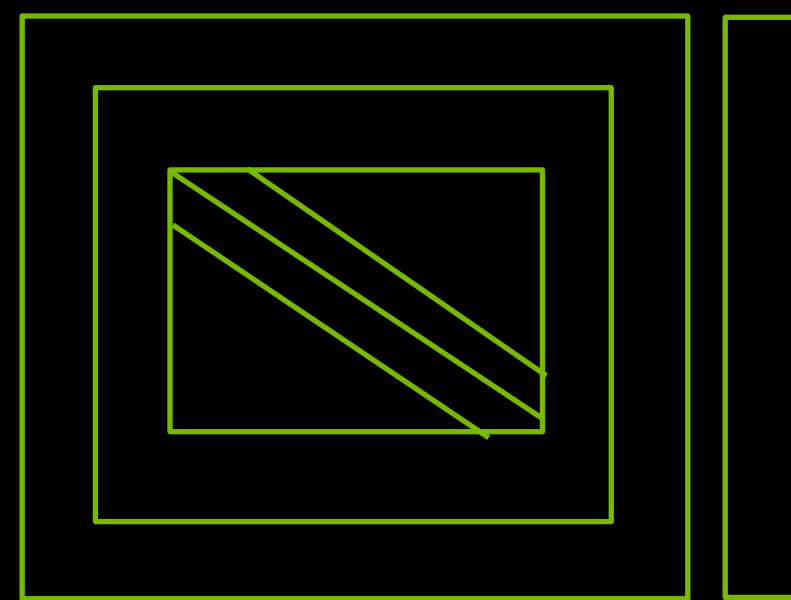
cuTENSOR



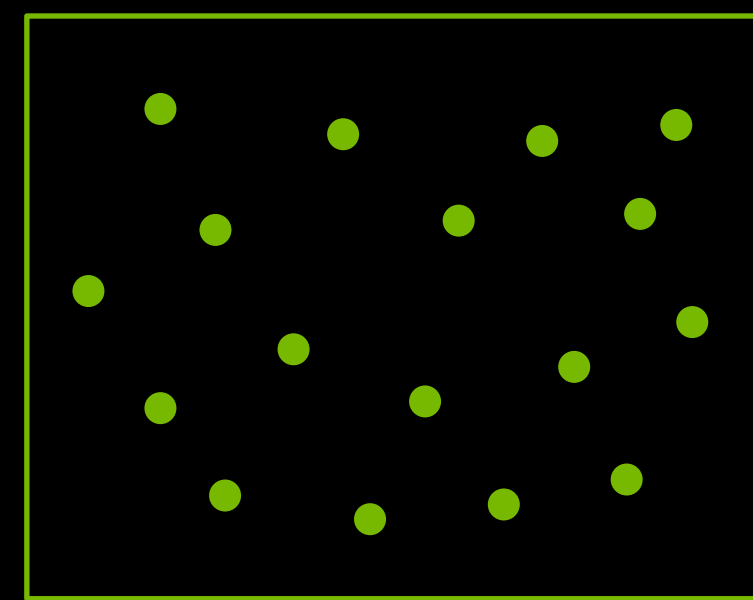
cuSOLVER



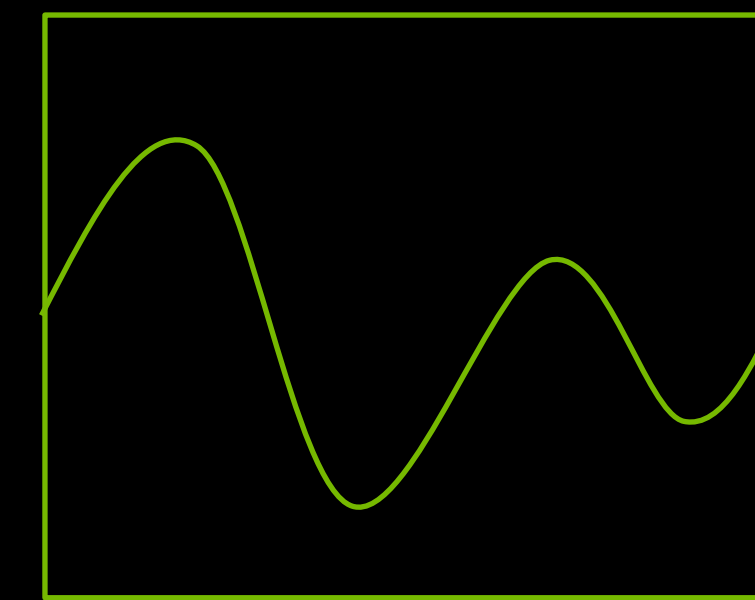
CUTLASS



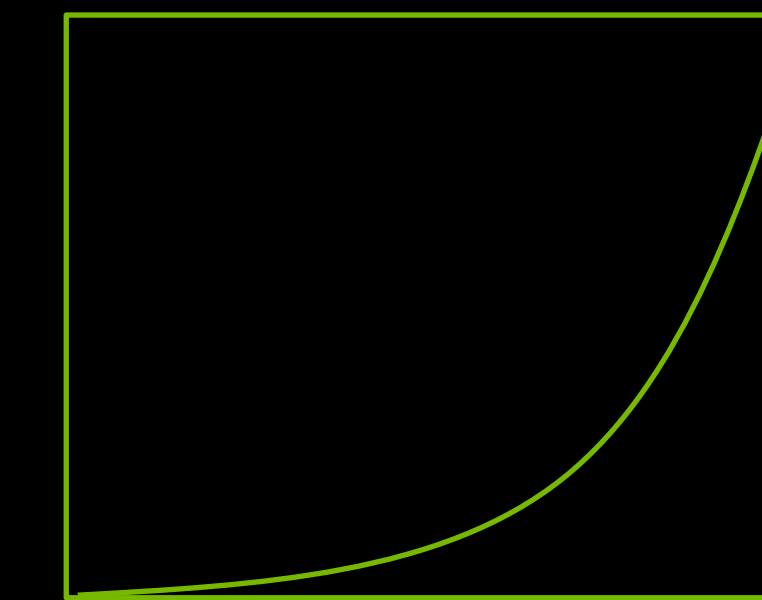
AMGX



cuRAND



cuFFT



CUDA Math API

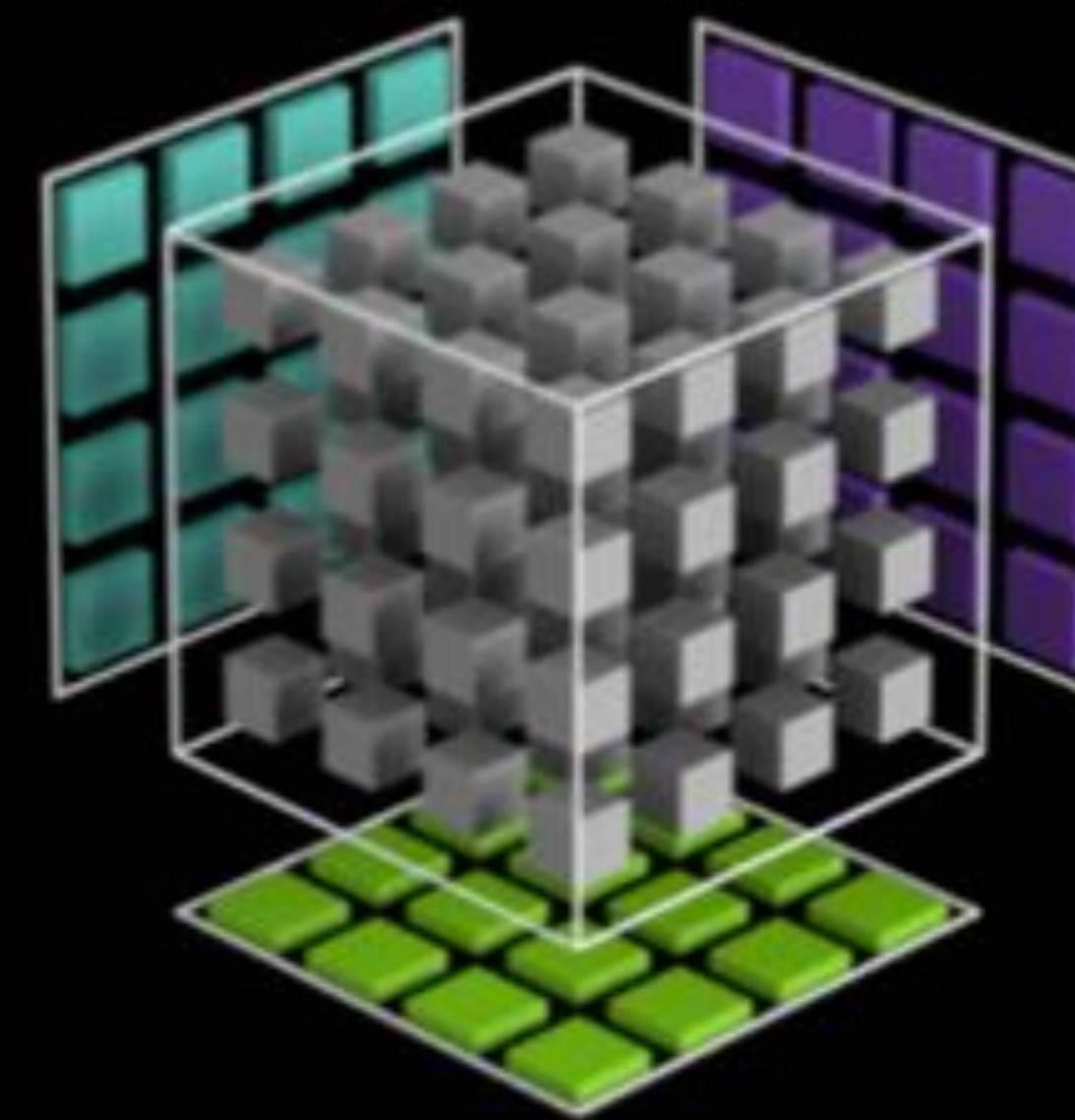
More libraries: <https://developer.nvidia.com/gpu-accelerated-libraries>

Tensor Cores

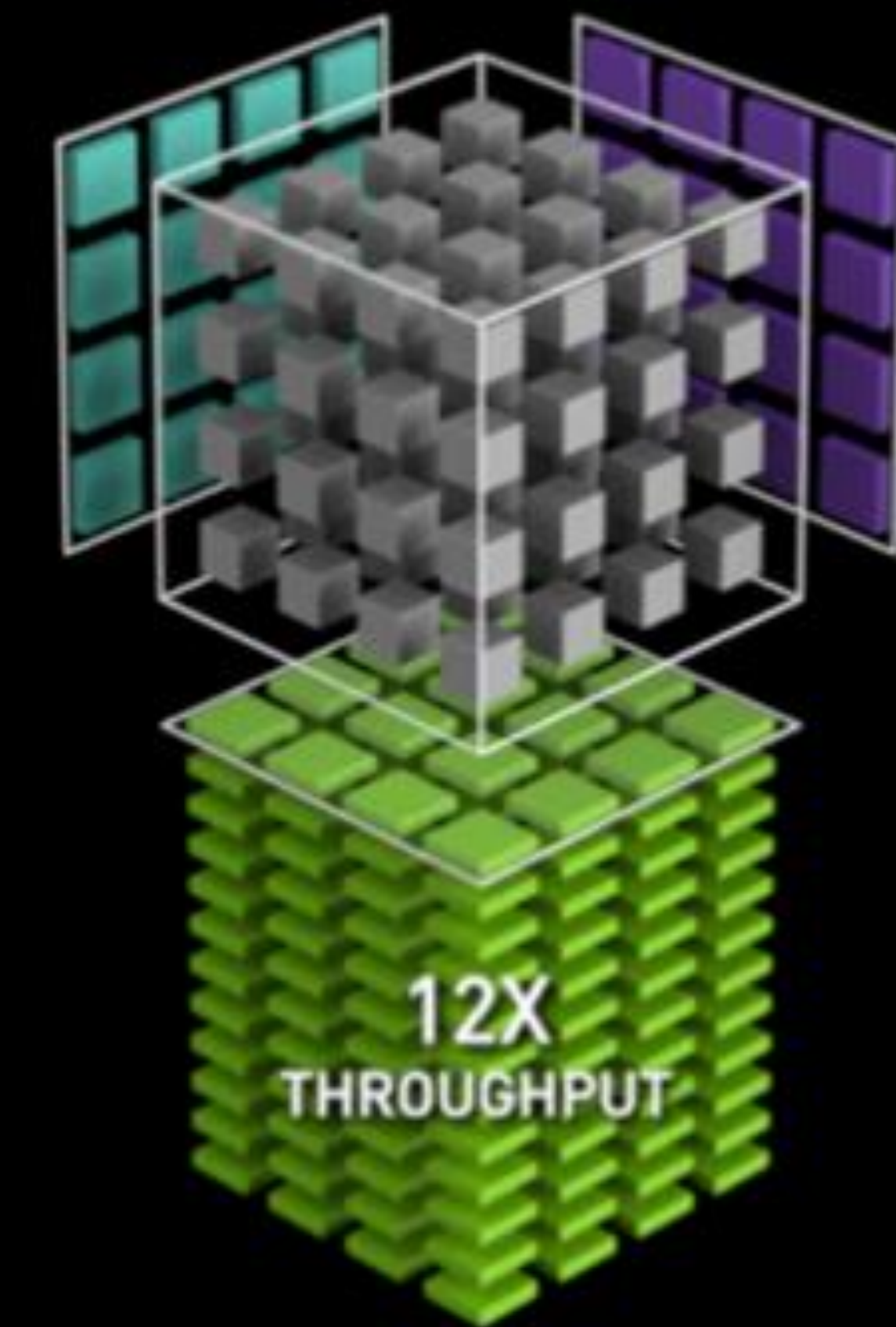
Hardware for Matrix Multiply and Accumulate operations

- Perform several MMA calcs per clock cycle
- Introduced in the V100
 - FP32 in, FP32 out (accumulate)
 - FP16 multiply
- Turing added int8, int4, int1 calculations
- Ampere (A100)
 - Full FP64 MMA
 - Bfloat16, Tensor Float 32
- Hopper (H100)
 - FP8
 - Transformer Engine

PASCAL



VOLTA TENSOR CORES



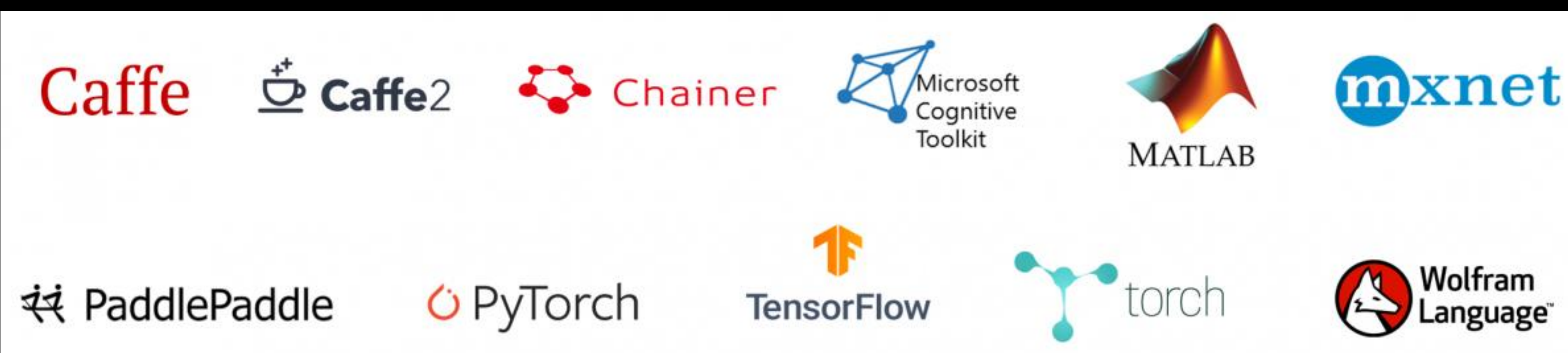
cuDNN, CUTLASS

Exploiting Tensor Cores

cuDNN - Accelerating deep learning primitives

Key Features

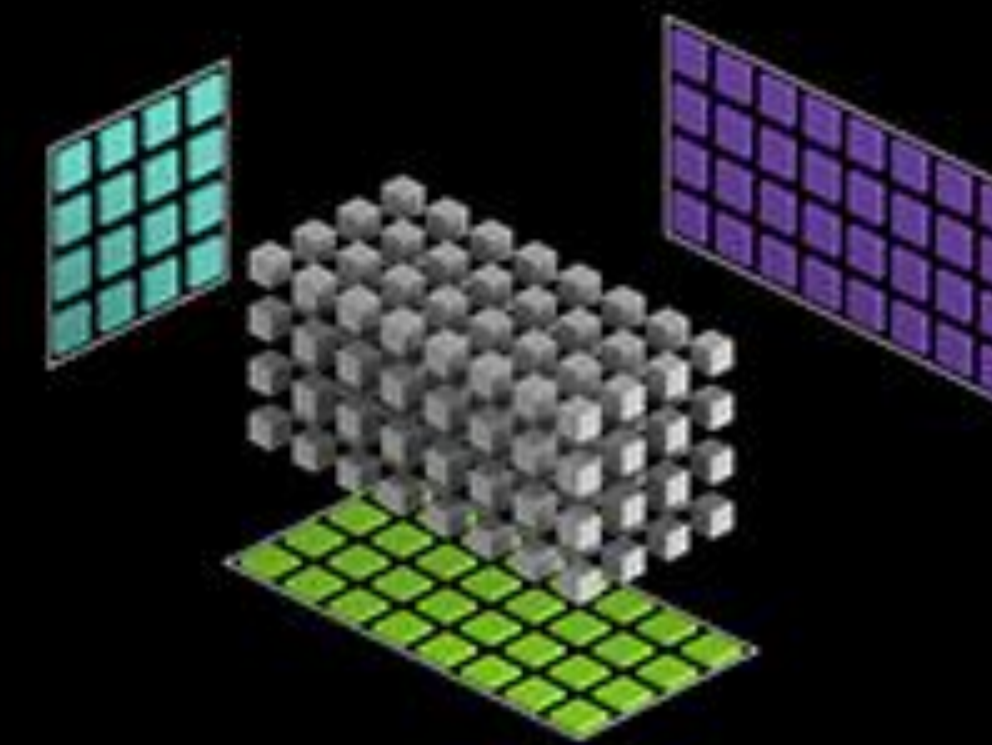
- Tensor Core acceleration for all popular convolutions
- Supports FP32, FP16, BF16 and TF32 floating point formats and INT8, and UINT8 integer formats
- Arbitrary dimension ordering, striding, and sub-regions for 4d tensors means easy integration into any neural net implementation



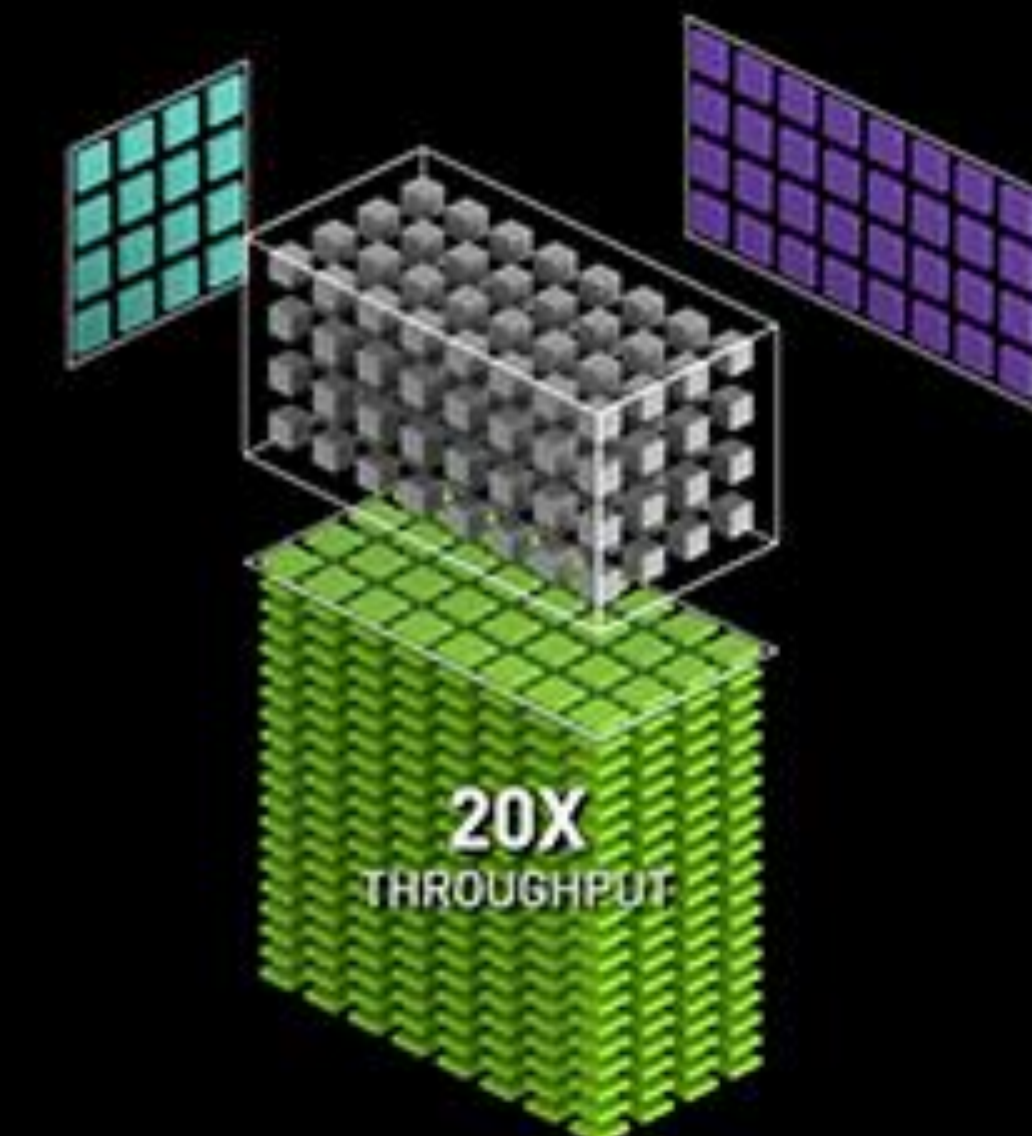
CUTLASS – Tensor Core Programming Model

- Warp-Level GEMM and Reusable Components for Linear Algebra Kernels in CUDA

NVIDIA V100 FP32



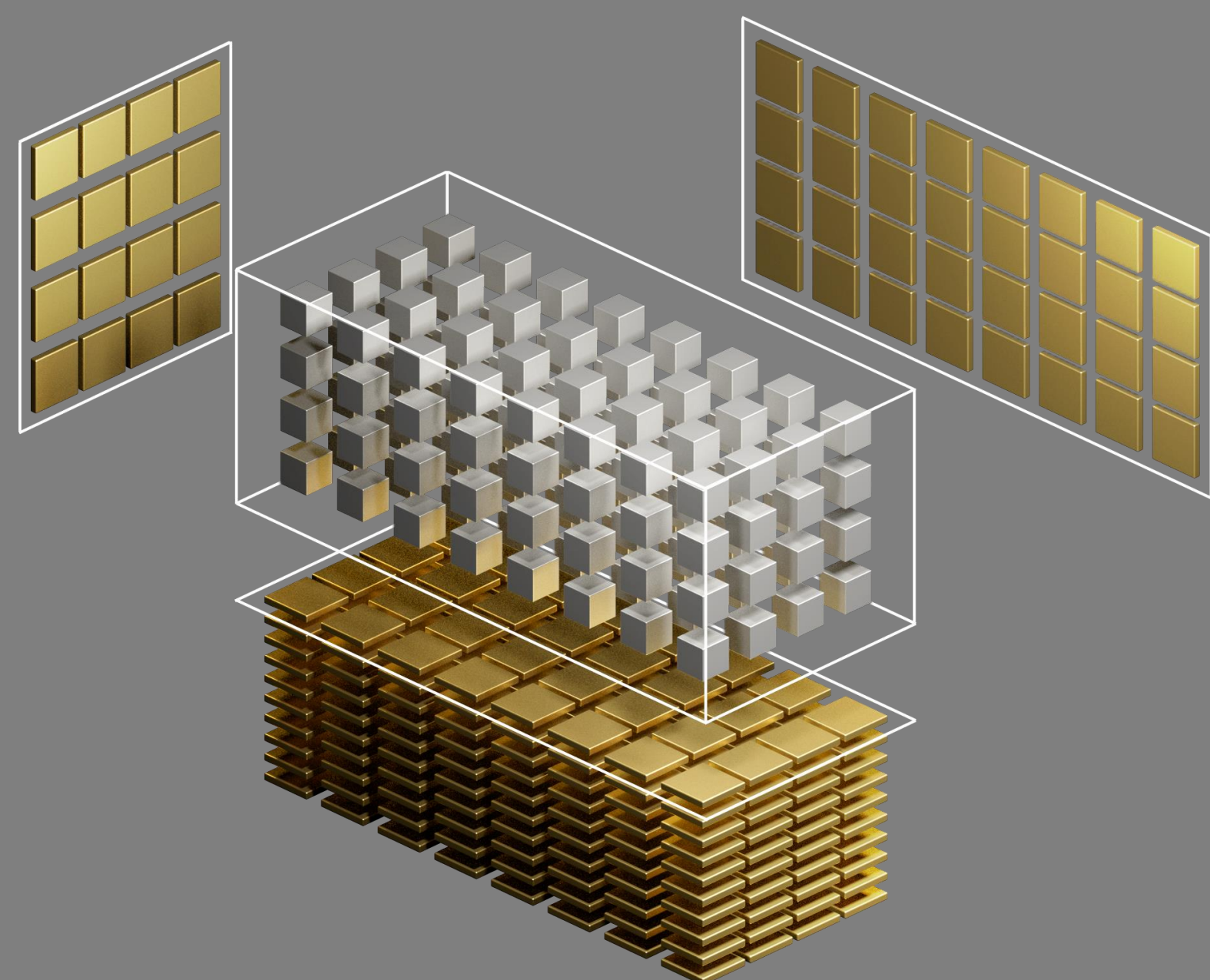
NVIDIA A100 Tensor Core TF32 with Sparsity



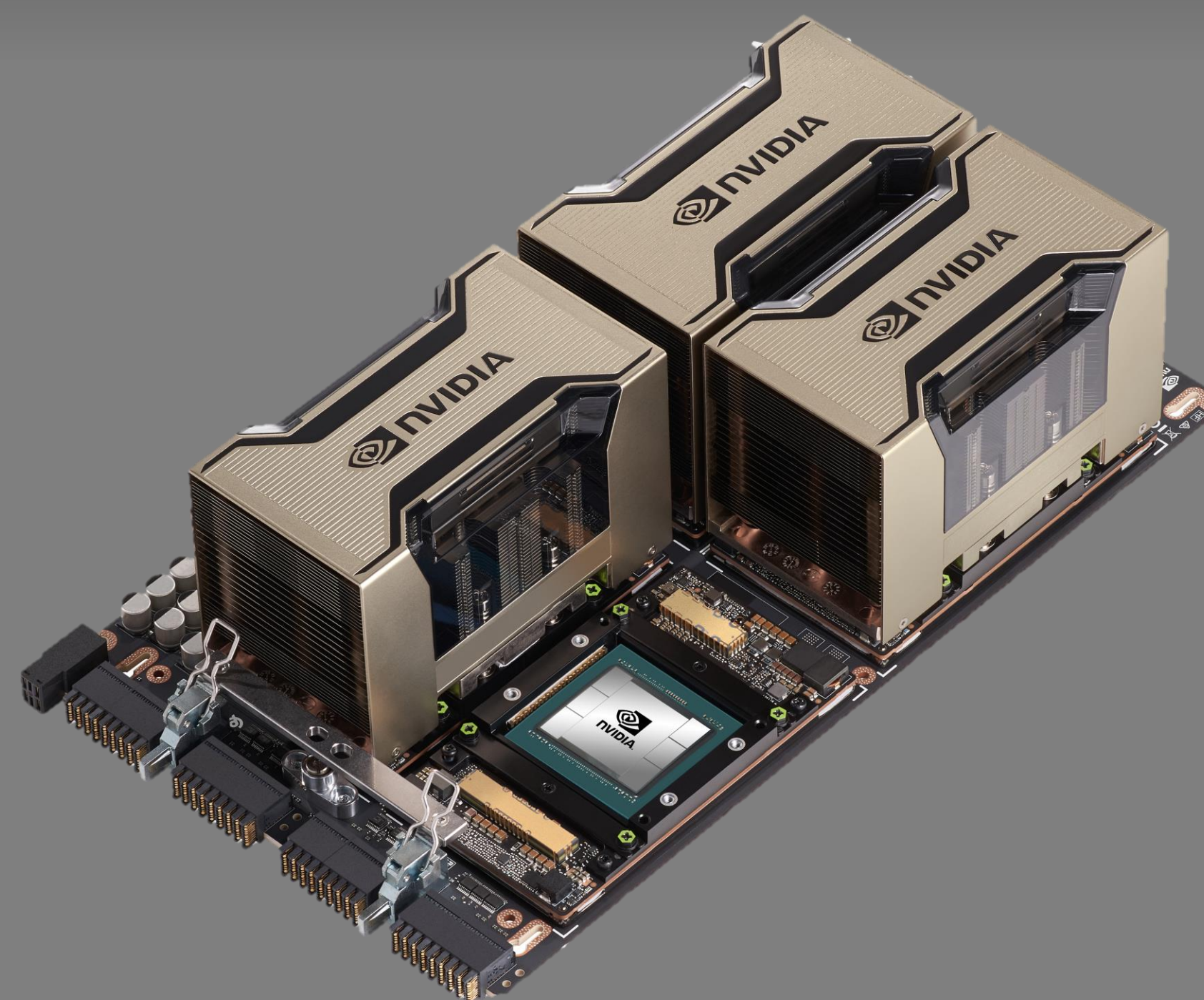
NVIDIA Performance Libraries

Direction of travel

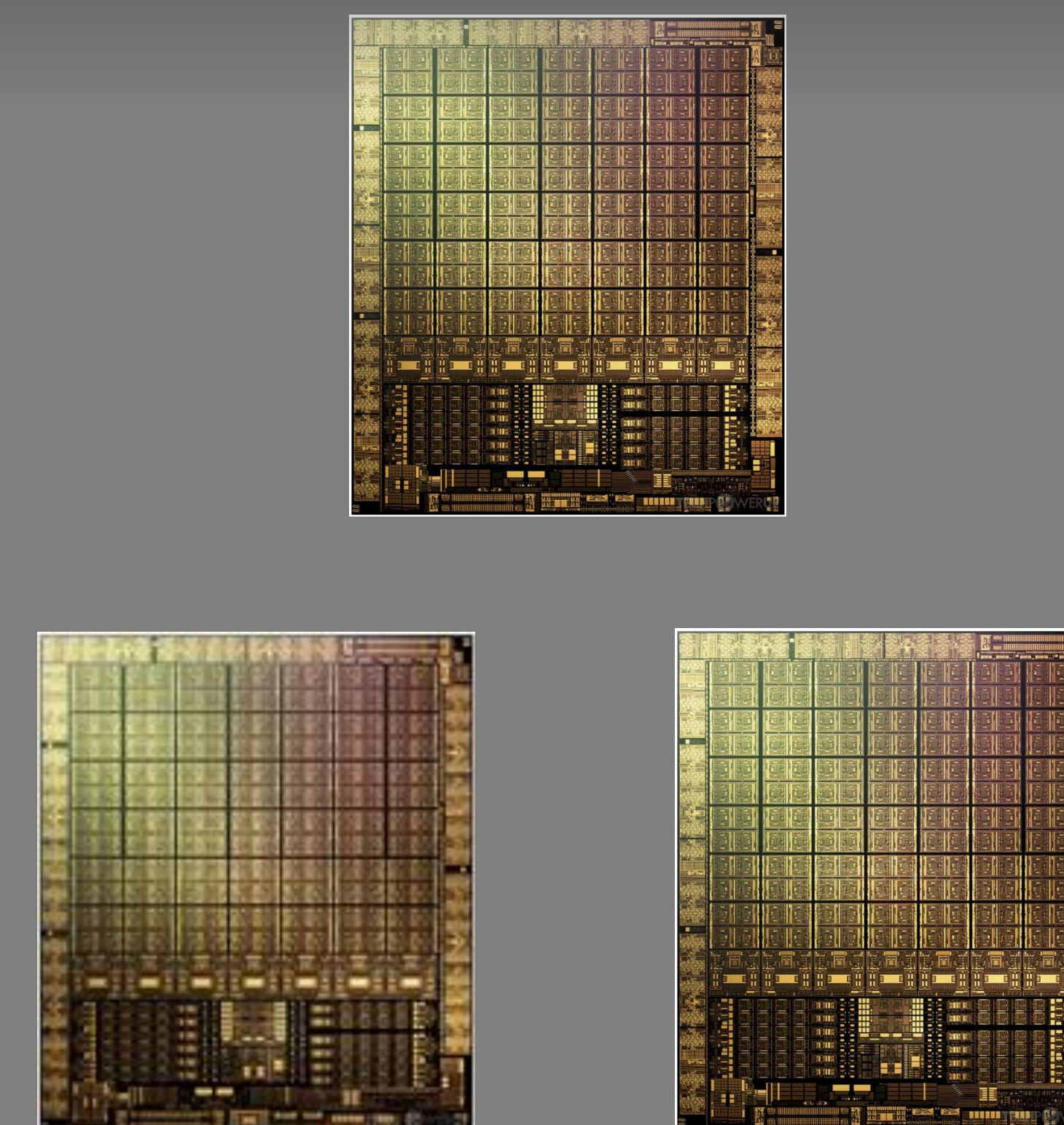
Seamless Acceleration
Tensor Cores, Enhanced L2\$ & SMEM



Scaling Up
Multi-GPU and Multi-Node Libraries



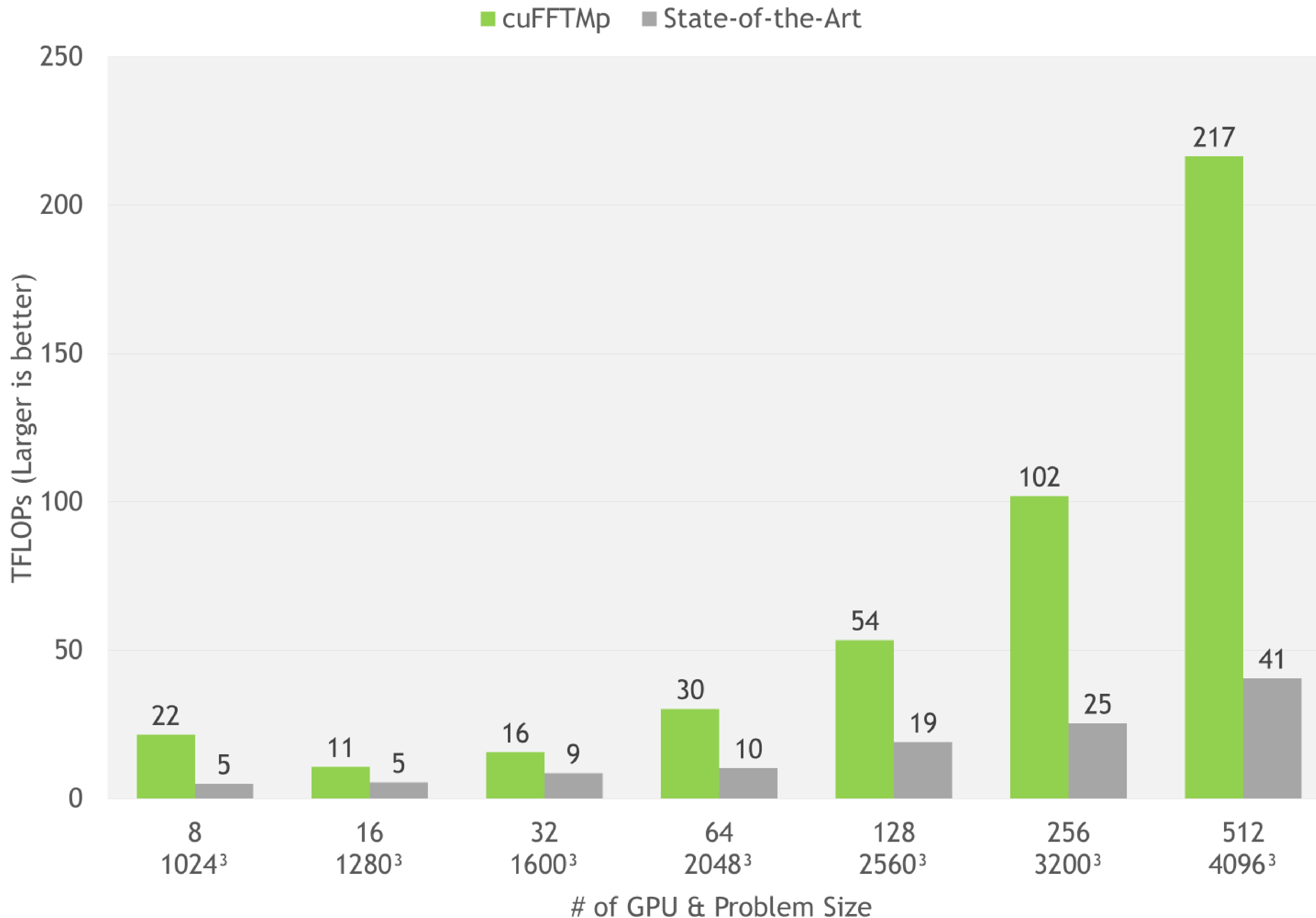
Composability
Device Functions



MULTI-NODE MATH LIBRARIES

cuFFTMp: Fast Fourier Transforms at scale

Performance: cuFFTMp vs. State-of-the-Art on Summit



cuFFTMp

A distributed-memory multi-node and multiGPU solution for solving FFTs at scale.

EA release available in Fall '21

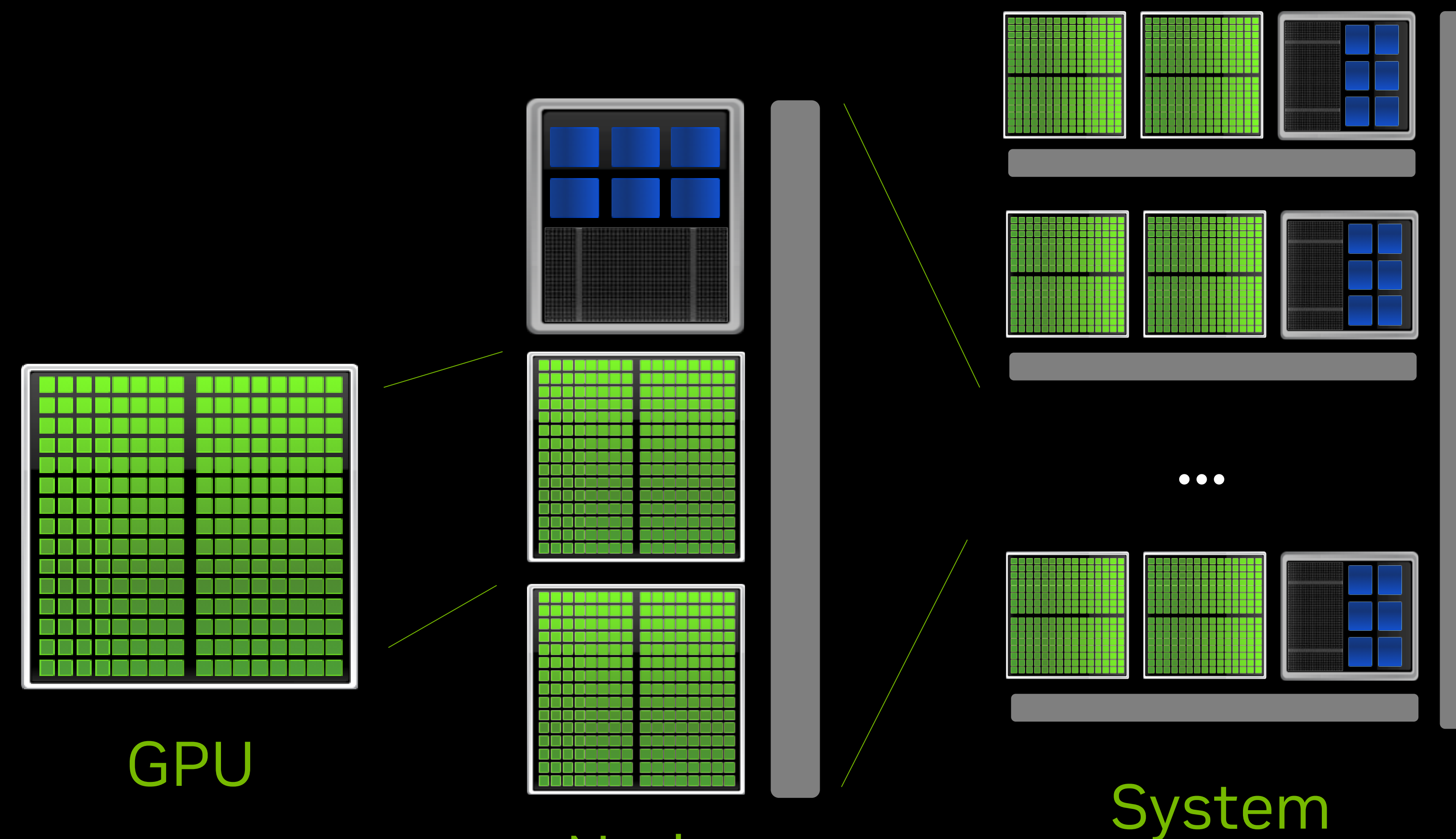
<https://developer.nvidia.com/cudamathlibraryea>

Initial release to 2D & 3D with Slab composition

Multiple GPUS

Accelerating at all scales

- [Unified Memory](#)
- [Multi-Process Service](#) – GROMACS [blog](#)
- [NVLink / NVSwitch](#) and new [NVLink Switch!](#)
- [CUDA-aware MPI](#)
- [NVSHMEM](#)
- [NCCL](#) – multi-GPU/node communication primitives
- [GPUDirect](#) – comms between GPUs
 - intra- and inter-node
 - now also to storage
- [Networking](#) - DPU
- Analysis: [Nsight](#) tools
- Note: DL Frameworks on NGC and many other HPC applications already have multiGPU and multi-node support built in



OpenACC is a directives-based programming approach to **parallel computing** designed for **performance** and **portability** on CPUs and GPUs for HPC.

Add Simple Compiler Directive

```
main()
{
  <serial code>
  #pragma acc kernels
  {
    <parallel code>
  }
}
```



Programming the NVIDIA Platform

CPU, GPU, and Network

ACCELERATED STANDARD LANGUAGES

ISO C++, ISO Fortran

```
std::transform(par, x, x+n, y, y,  
              [=](float x, float y){ return y +  
a*x; }  
);
```

```
do concurrent (i = 1:n)  
  y(i) = y(i) + a*x(i)  
enddo
```

```
import cunumeric as np  
...  
def saxpy(a, x, y):  
  y[:] += a*x
```

INCREMENTAL PORTABLE OPTIMIZATION

OpenACC, OpenMP

```
#pragma acc data copy(x,y) {  
  ...  
  std::transform(par, x, x+n, y, y,  
                [=](float x, float y){  
                  return y + a*x;  
                });  
  ...  
}  
  
#pragma omp target data map(x,y) {  
  ...  
  std::transform(par, x, x+n, y, y,  
                [=](float x, float y){  
                  return y + a*x;  
                });  
  ...  
}
```

PLATFORM SPECIALIZATION

CUDA

```
__global__  
void saxpy(int n, float a,  
           float *x, float *y) {  
  int i = blockIdx.x*blockDim.x +  
          threadIdx.x;  
  if (i < n) y[i] += a*x[i];  
}  
  
int main(void) {  
  ...  
  cudaMemcpy(d_x, x, ...);  
  cudaMemcpy(d_y, y, ...);  
  
  saxpy<<<(N+255)/256,256>>>(...);  
  
  cudaMemcpy(y, d_y, ...);  
}
```

ACCELERATION LIBRARIES

Core

Math

Communication

Data Analytics

AI

Quantum

Programming the NVIDIA Platform

CPU, GPU, and Network

ACCELERATED STANDARD LANGUAGES

ISO C++, ISO Fortran

```
std::transform(par, x, x+n, y, y,  
              [=](float x, float y){ return y +  
              a*x; }  
              );
```

```
do concurrent (i = 1:n)  
  y(i) = y(i) + a*x(i)  
enddo
```

```
import cunumeric as np  
...  
def saxpy(a, x, y):  
  y[:] += a*x
```

INCREMENTAL PORTABLE OPTIMIZATION

OpenACC, OpenMP

```
#pragma acc data copy(x,y) {  
  ...  
  std::transform(par, x, x+n, y, y,  
                [=](float x, float y){  
                  return y + a*x;  
                }  
                );  
  ...  
}  
#pragma omp target data map(x,y) {  
  ...  
  std::transform(par, x, x+n, y, y,  
                [=](float x, float y){  
                  return y + a*x;  
                }  
                );  
  ...  
}
```

PLATFORM SPECIALIZATION

CUDA

```
__global__  
void saxpy(int n, float a,  
           float *x, float *y) {  
  int i = blockIdx.x*blockDim.x +  
         threadIdx.x;  
  if (i < n) y[i] += a*x[i];  
}  
  
int main(void) {  
  ...  
  cudaMemcpy(d_x, x, ...);  
  cudaMemcpy(d_y, y, ...);  
  
  saxpy<<<(N+255)/256,256>>>(...);  
  
  cudaMemcpy(y, d_y, ...);  
}
```

ACCELERATION LIBRARIES

Core

Math

Communication

Data Analytics

AI

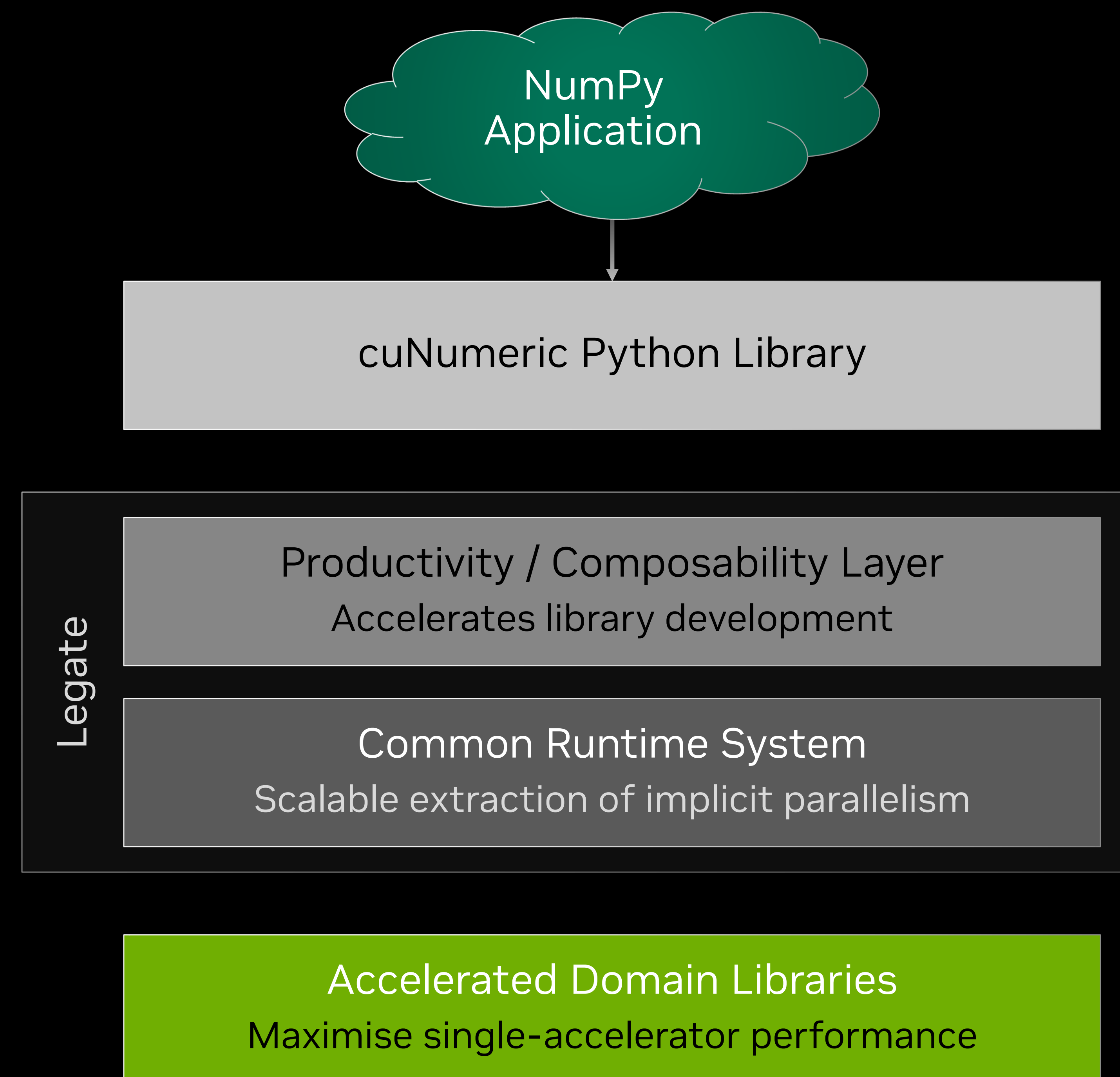
Quantum

cuNumeric – Implicitly Parallel Implementations of NumPy APIs

Stencil Benchmark

No modifications required to scale to a thousand GPUs

```
32
33 def run_stencil(N, I, warmup, timing): # noqa: E741
34     grid = initialize(N)
35
36     print("Running Jacobi stencil...")
37     center = grid[1:-1, 1:-1]
38     north = grid[0:-2, 1:-1]
39     east = grid[1:-1, 2:]
40     west = grid[1:-1, 0:-2]
41     south = grid[2:, 1:-1]
42
43     timer.start()
44     for i in range(I + warmup):
45         if i == warmup:
46             timer.start()
47             average = center + north + east + west + south
48             work = 0.2 * average
49             center[:] = work
50     total = timer.stop()
51
52     if timing:
53         print(f"Elapsed Time: {total} ms")
54     return total
```

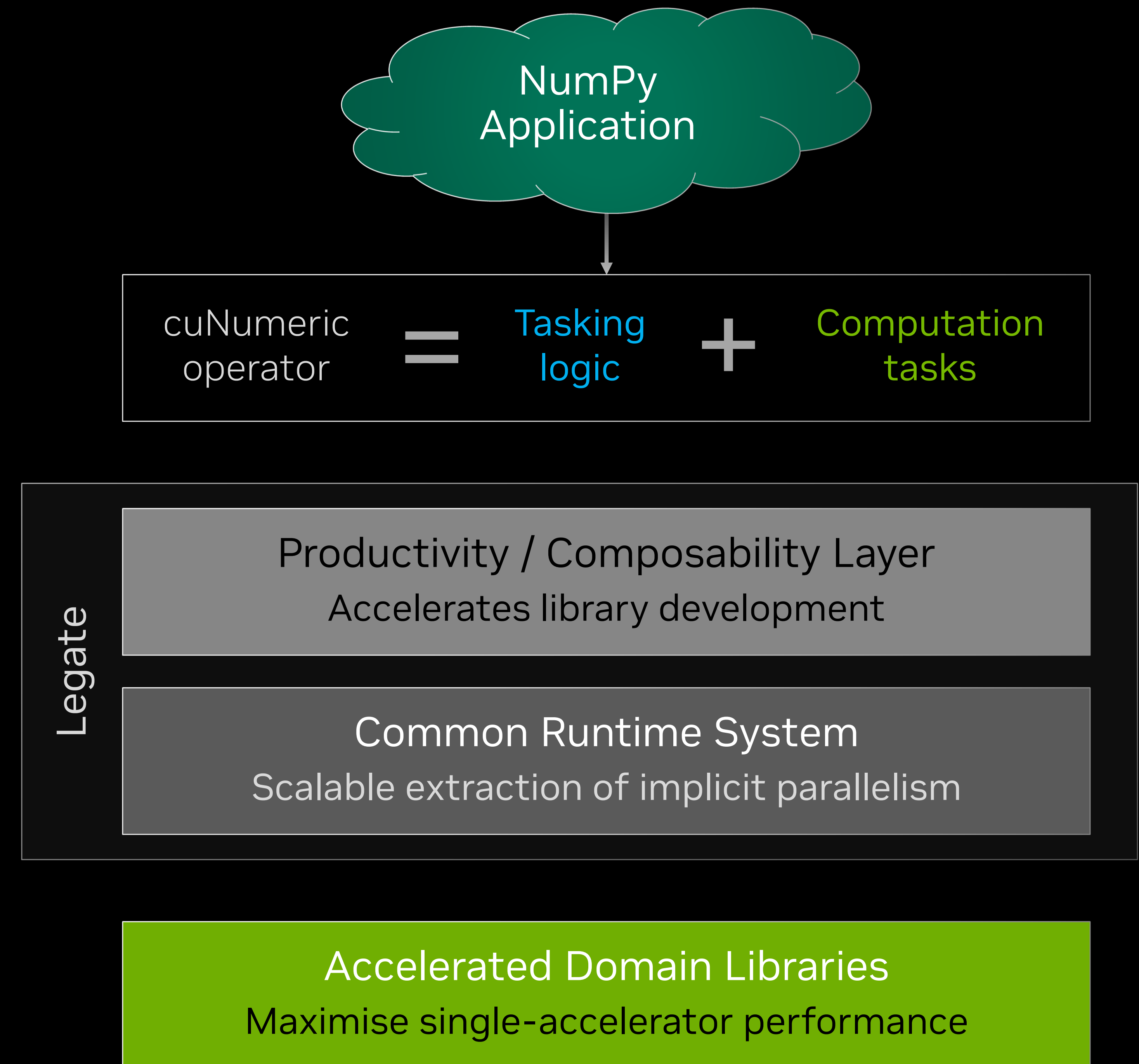


cuNumeric – Implicitly Parallel Implementations of NumPy APIs

Stencil Benchmark

No modifications required to scale to a thousand GPUs

```
32
33 def run_stencil(N, I, warmup, timing): # noqa: E741
34     grid = initialize(N)
35
36     print("Running Jacobi stencil...")
37     center = grid[1:-1, 1:-1]
38     north = grid[0:-2, 1:-1]
39     east = grid[1:-1, 2:]
40     west = grid[1:-1, 0:-2]
41     south = grid[2:, 1:-1]
42
43     timer.start()
44     for i in range(I + warmup):
45         if i == warmup:
46             timer.start()
47             average = center + north + east + west + south
48             work = 0.2 * average
49             center[:] = work
50     total = timer.stop()
51
52     if timing:
53         print(f"Elapsed Time: {total} ms")
54     return total
```

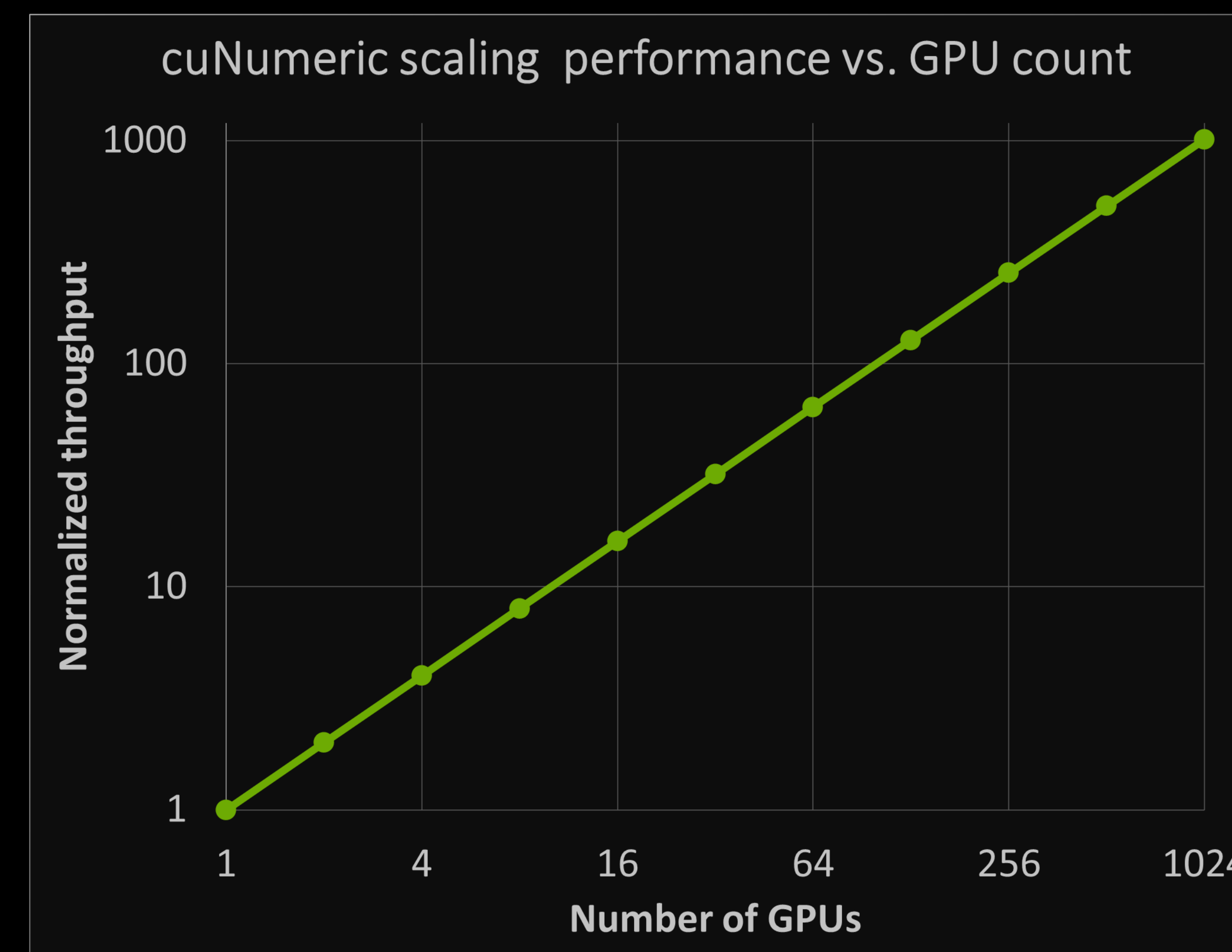


cuNumeric – Implicitly Parallel Implementations of NumPy APIs

Stencil Benchmark

No modifications required to scale to a thousand GPUs

```
32
33 def run_stencil(N, I, warmup, timing): # noqa: E741
34     grid = initialize(N)
35
36     print("Running Jacobi stencil...")
37     center = grid[1:-1, 1:-1]
38     north = grid[0:-2, 1:-1]
39     east = grid[1:-1, 2:]
40     west = grid[1:-1, 0:-2]
41     south = grid[2:, 1:-1]
42
43     timer.start()
44     for i in range(I + warmup):
45         if i == warmup:
46             timer.start()
47             average = center + north + east + west + south
48             work = 0.2 * average
49             center[:] = work
50     total = timer.stop()
51
52     if timing:
53         print(f"Elapsed Time: {total} ms")
54     return total
```





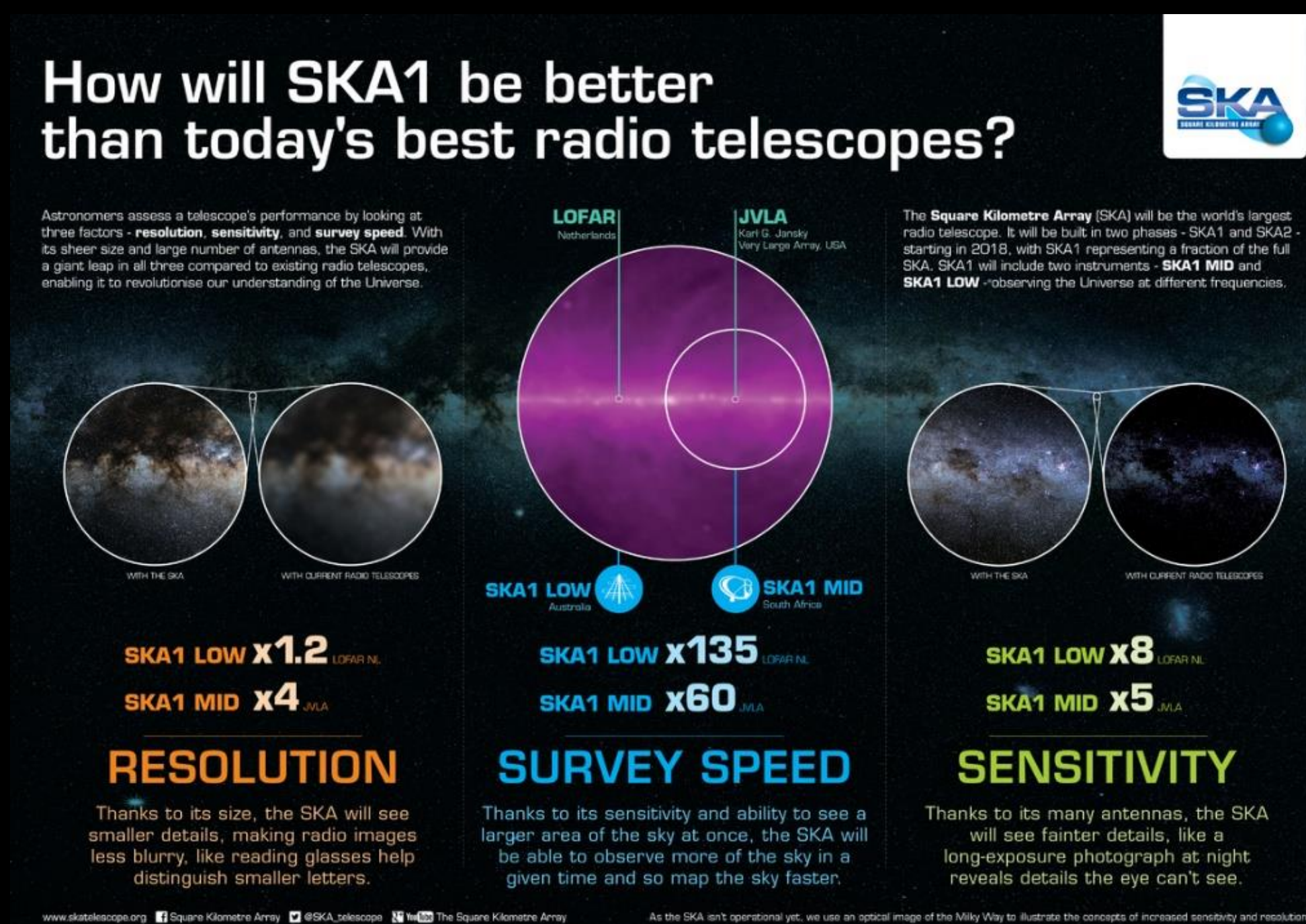
Frameworks



HPC & AI

HPC CHALLENGE 1

Fully Integrate New Experiments into the HPC Workflow



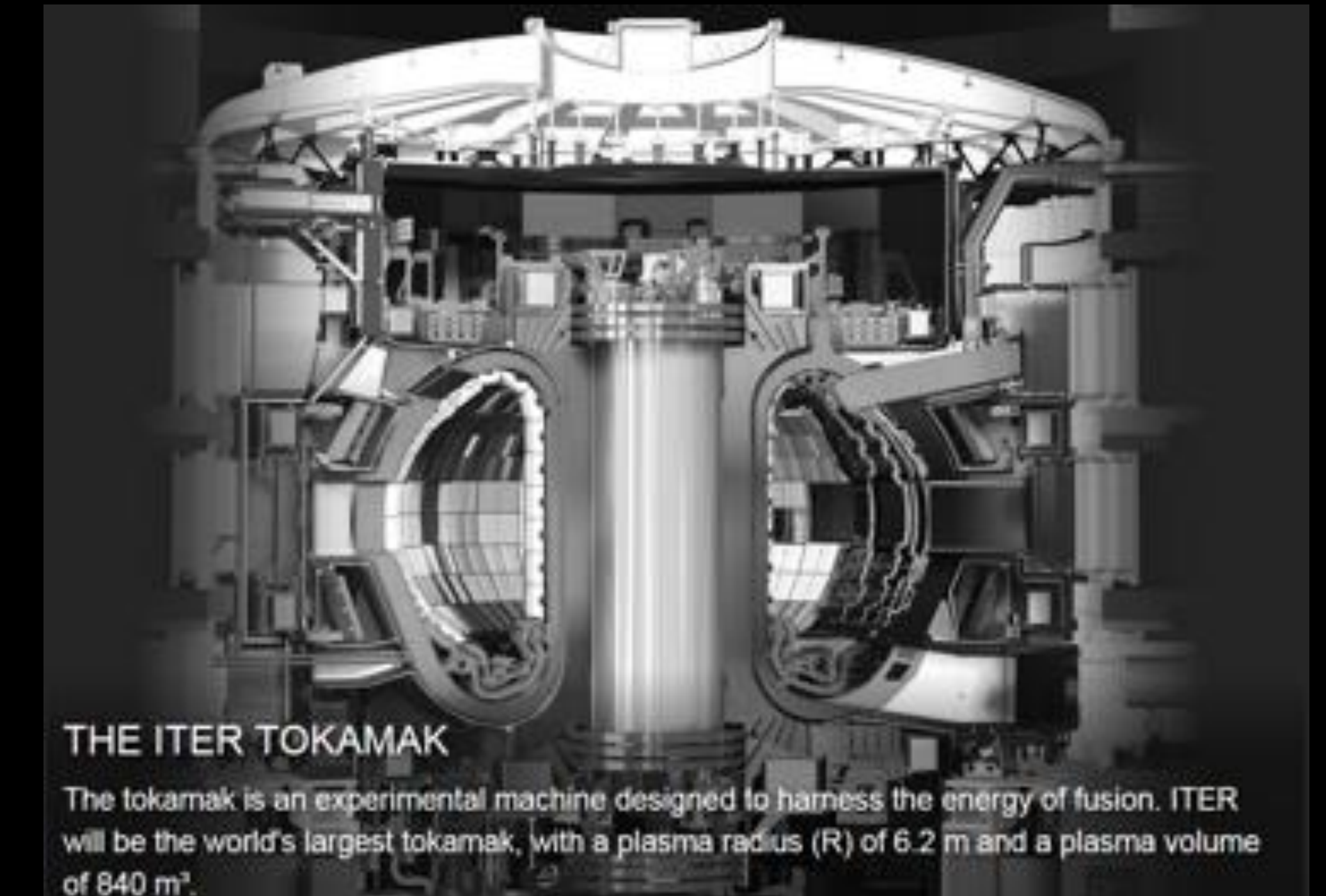
1EB/Day

SKA1 Square Kilometre Array radio telescope will generate more than an Exabyte of data every day



10X

The CERN large Hadron collider's High Luminosity upgrade will result in a 10X increase in data volume

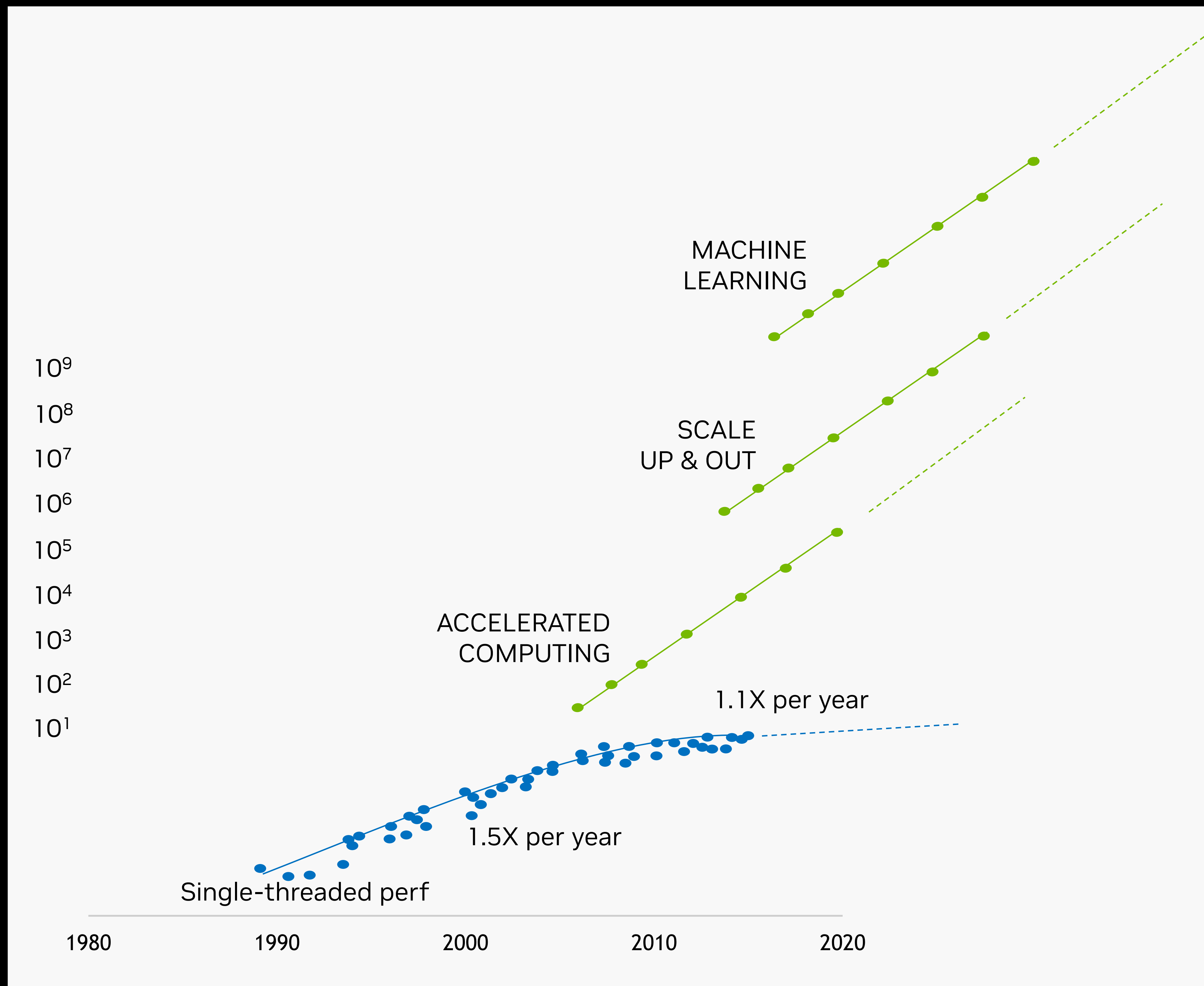


30X

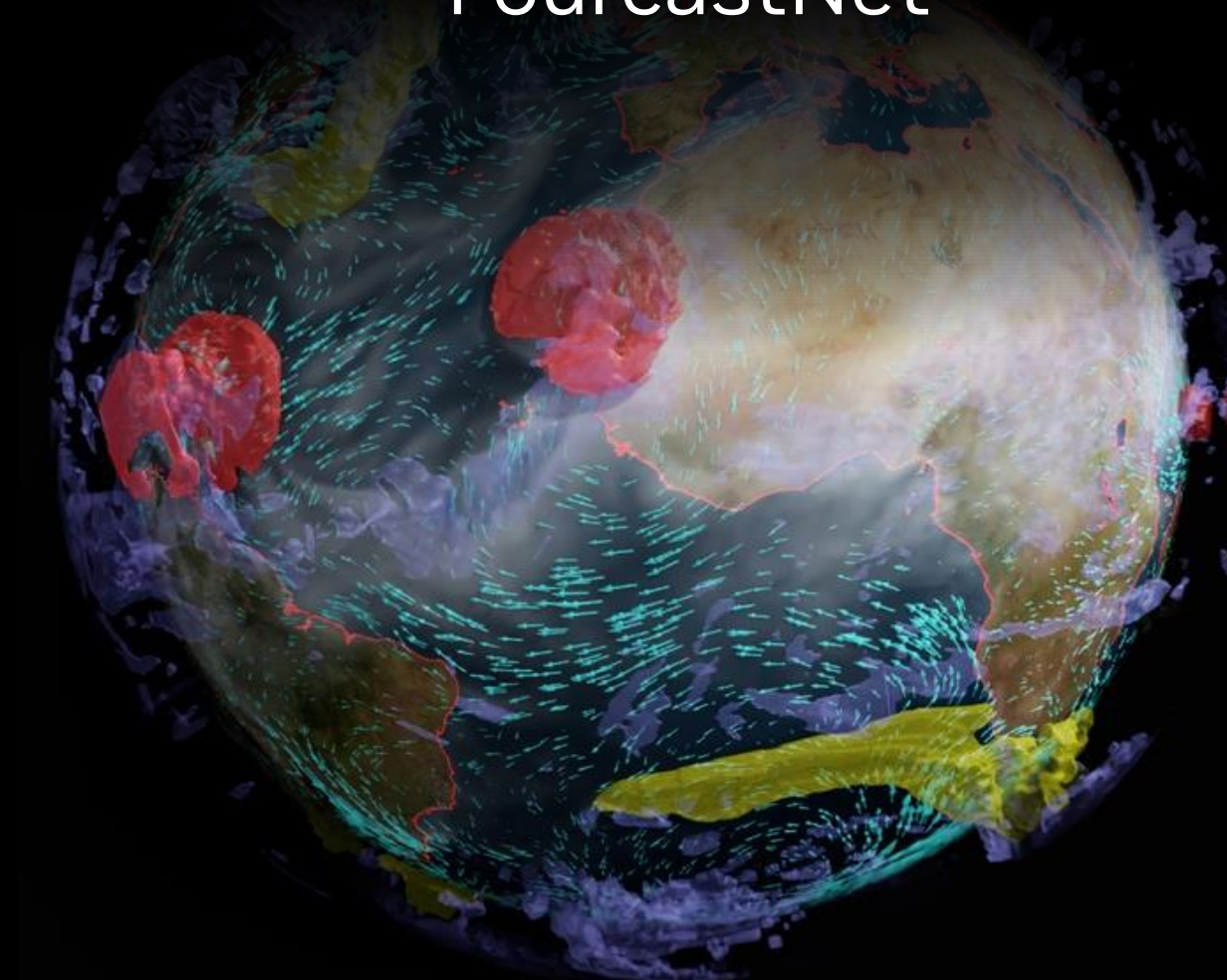
The 500 MW ITER fusion experiment will provide a 30X increase in output power over the largest previous experiment

HPC Challenge 2

HPC beyond Moore's Law: Simulation + AI



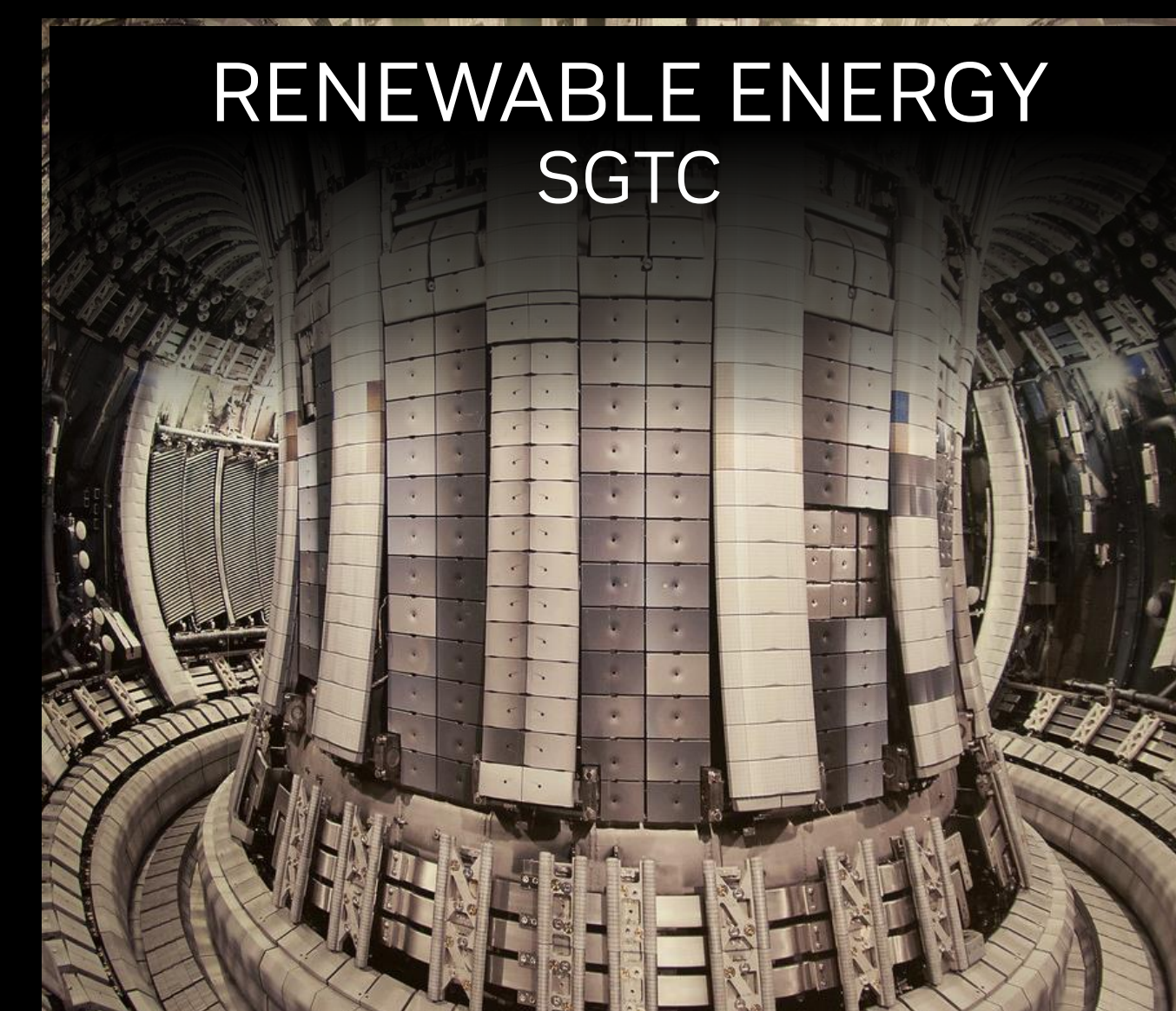
CLIMATE CHANGE
FourcastNet



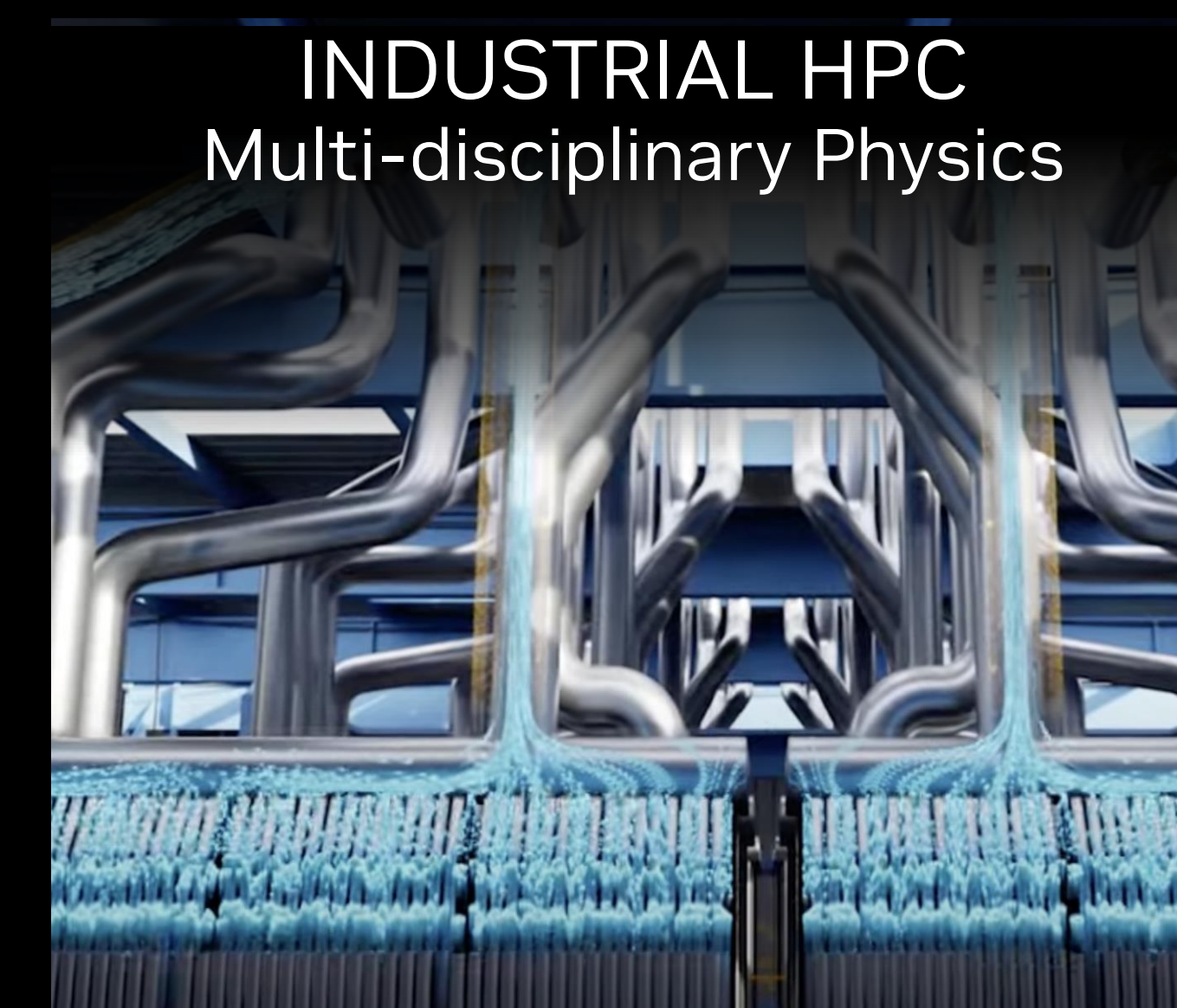
DIGITAL BIOLOGY
Orbnet



RENEWABLE ENERGY
SGTC



INDUSTRIAL HPC
Multi-disciplinary Physics



CONVERGED HPC*AI TAXONOMY

How AI Algorithms are Being Applied in the HPC Workflow

Modelling and Simulation

Experiment Data Processing

Ab Initio Algorithm
Enhancement

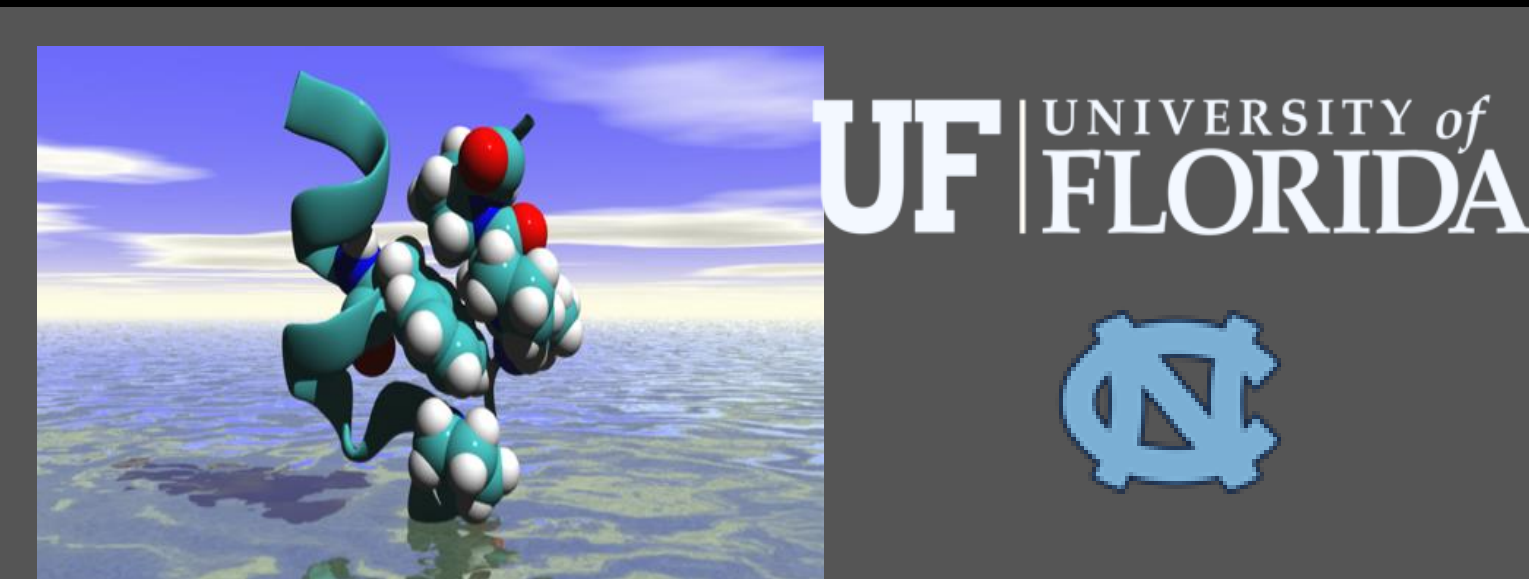
Reduced Order Model
Replacement

Detection
Pipeline

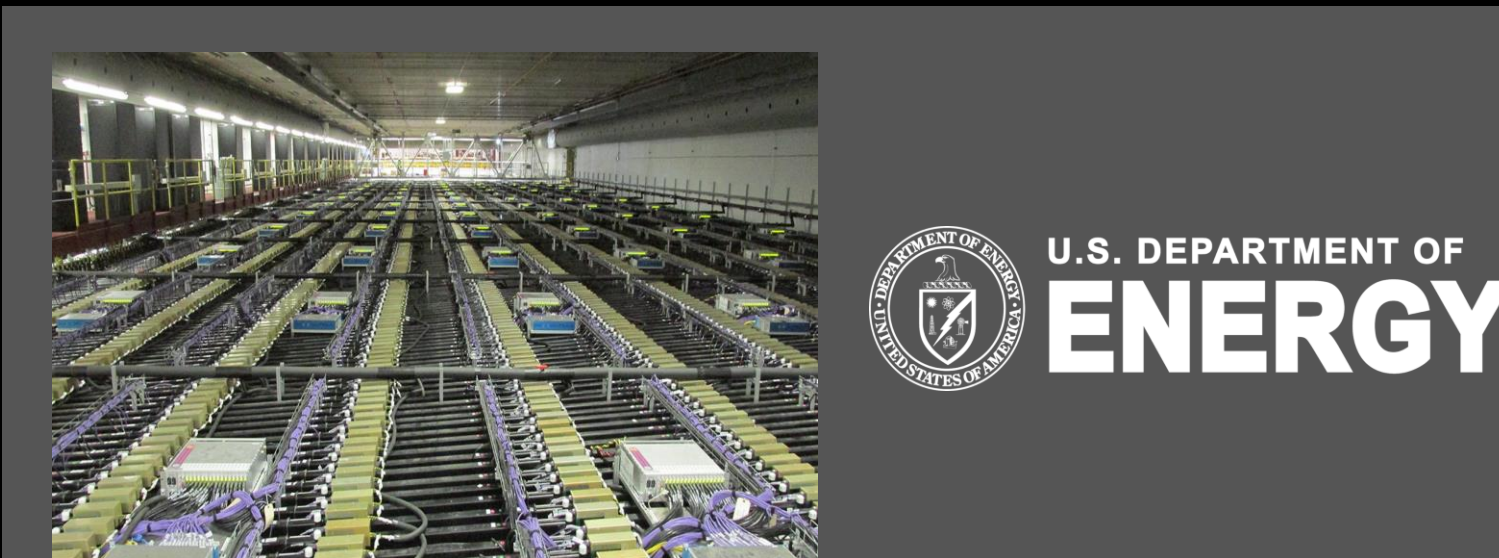
Real-Time
Control



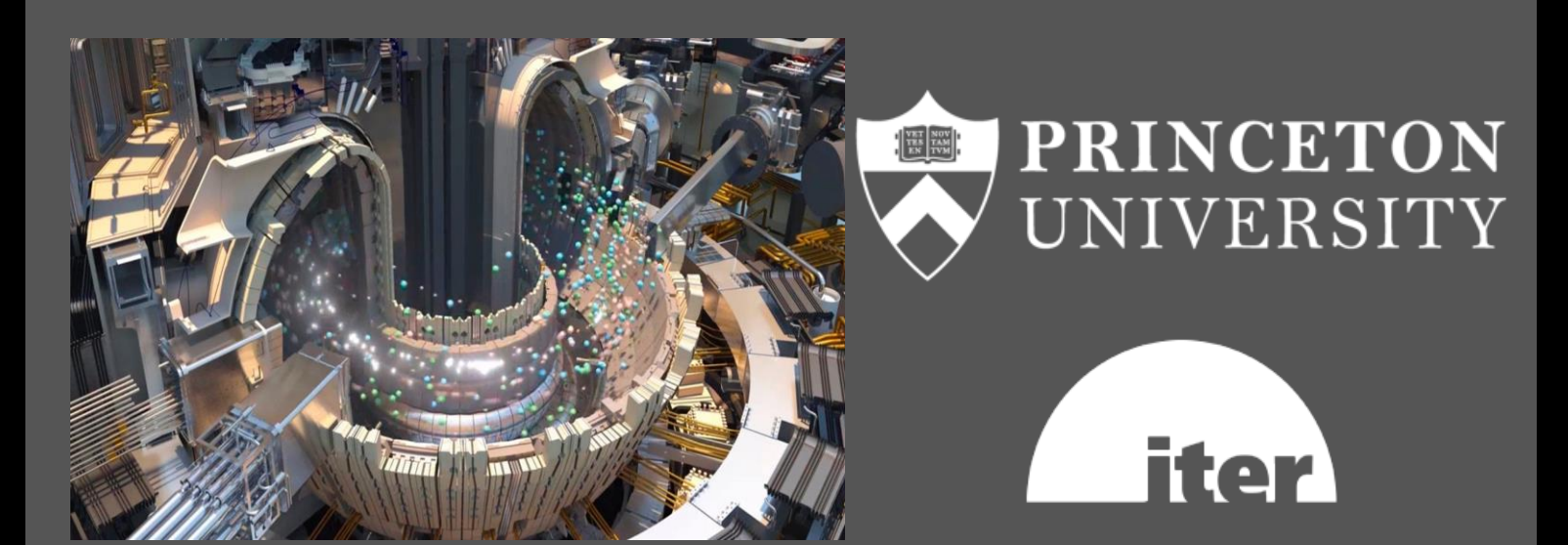
20X Faster Time to Solution



300,000X Faster
Time to solution



49% More Accurate
Detection

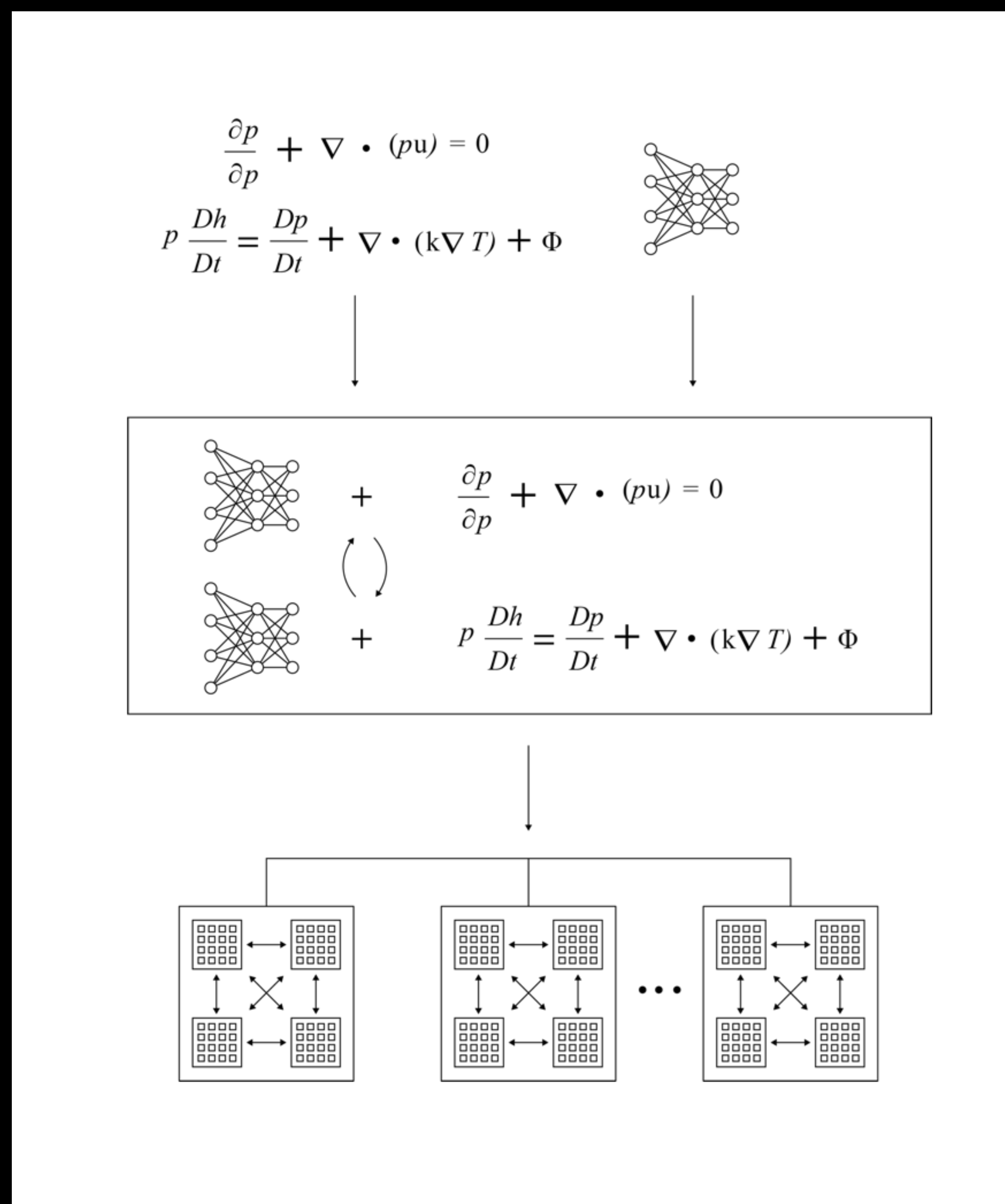


95% prediction accuracy
with 5% False positives

NVIDIA Modulus

Framework for Developing Physics Machine Learning Neural Network Models

TRAINING NEURAL NETWORKS USING BOTH DATA AND THE GOVERNING EQUATIONS



ADVANCING SCIENTIFIC DISCOVERY WITH MODULUS

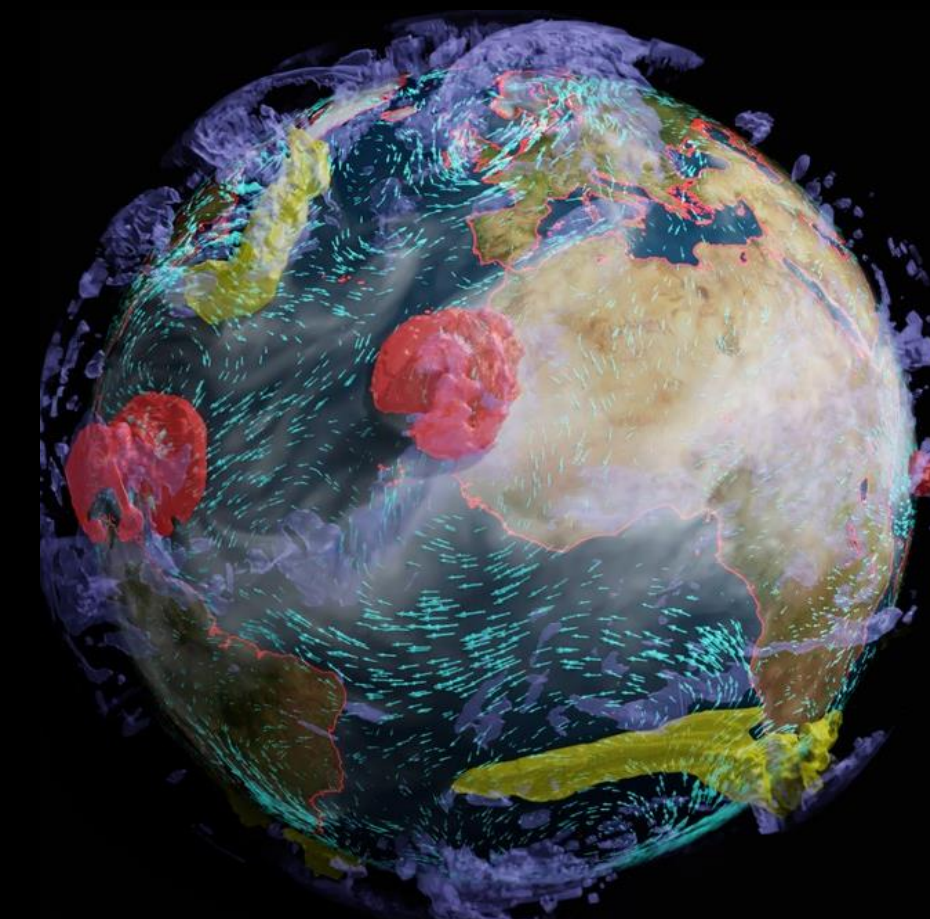
RENEWABLE ENERGY

Siemens Gamesa: Up to 4000X Speedup of Wind Turbine Wake Optimization



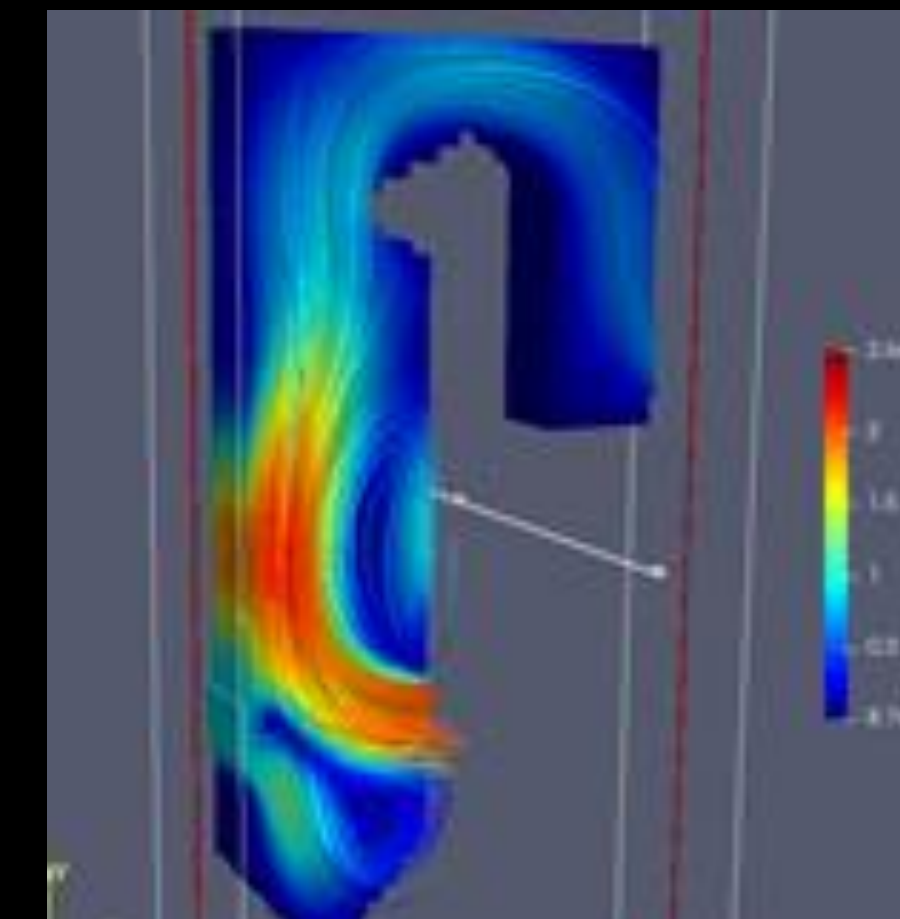
CLIMATE CHANGE

45,000X Speedup of Extreme weather Prediction with FourCastNet



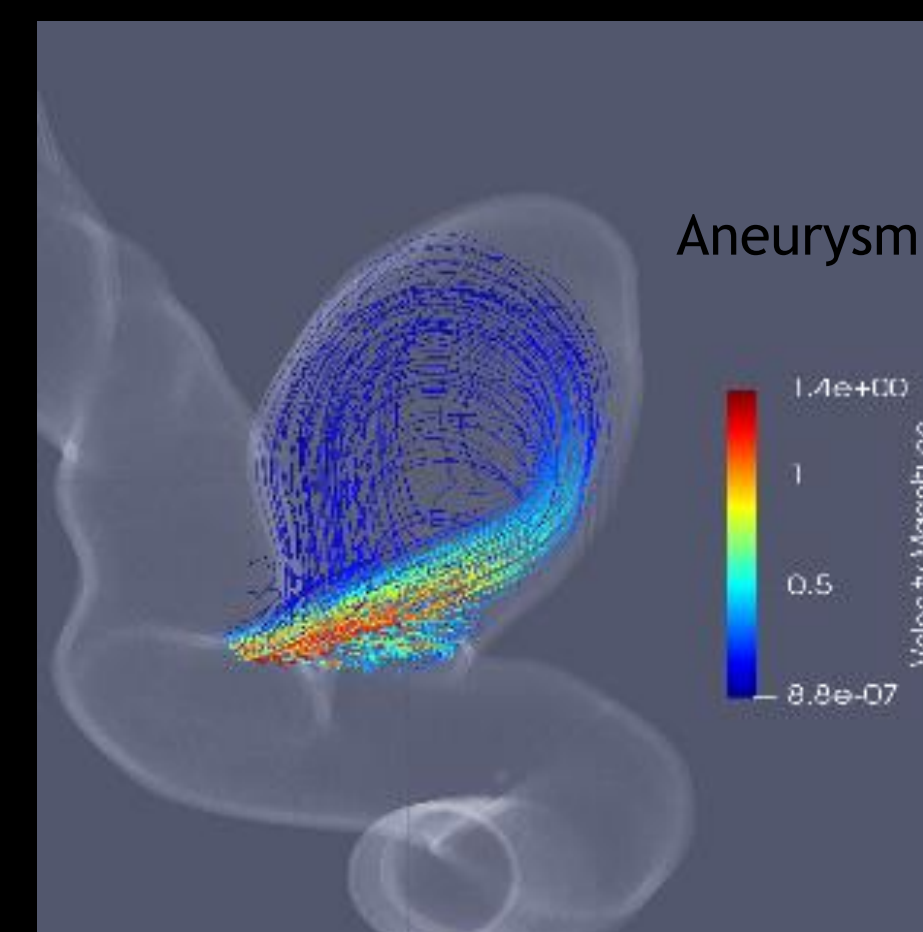
INDUSTRIAL HPC

NETL: 10,000X Faster Build Of high-fidelity surrogate models



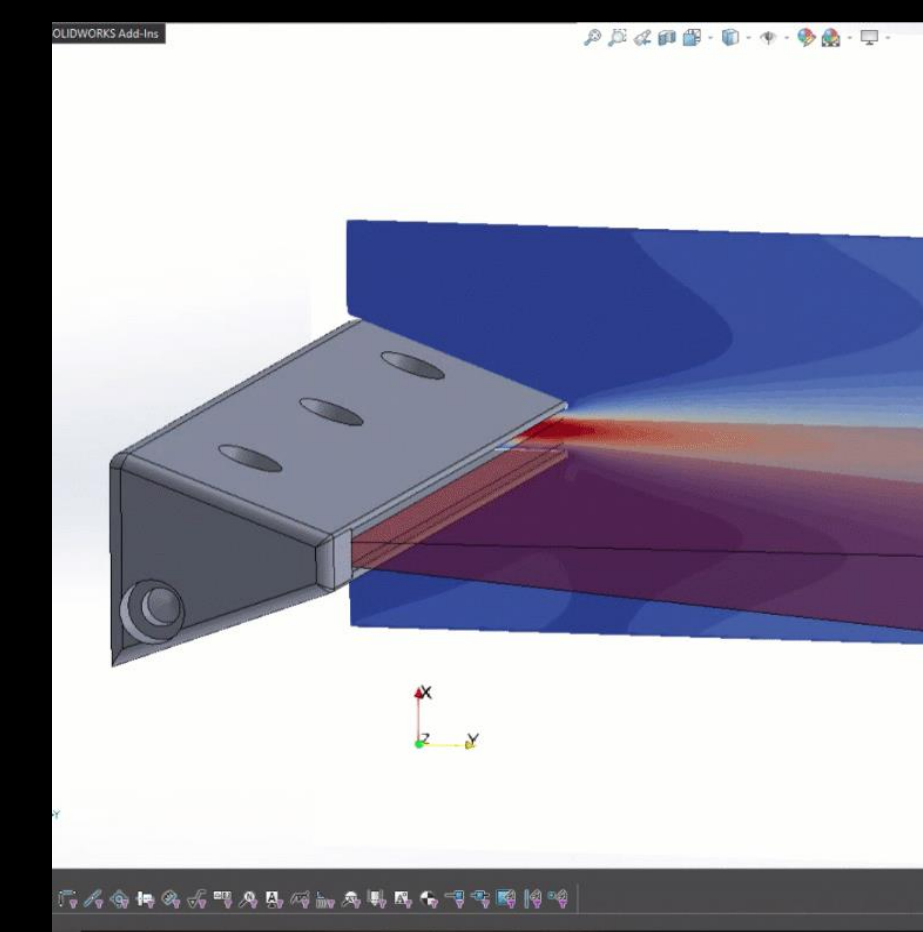
HEALTHCARE

Achieve high-fidelity results faster for blood flow in inter-cranial aneurysm



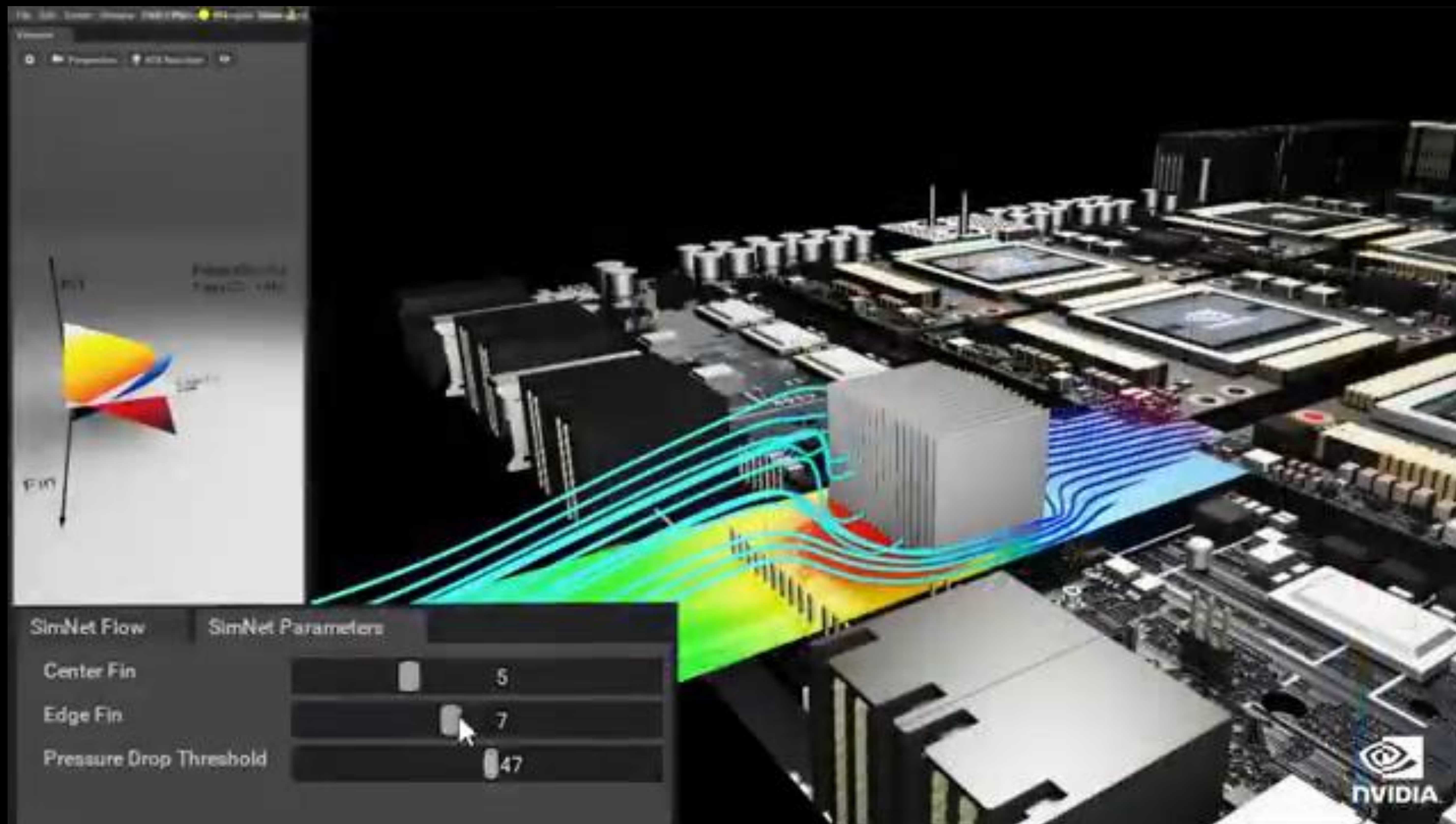
DIGITAL TWINS

Kinetic Vision: Design Optimization Using parameterized models



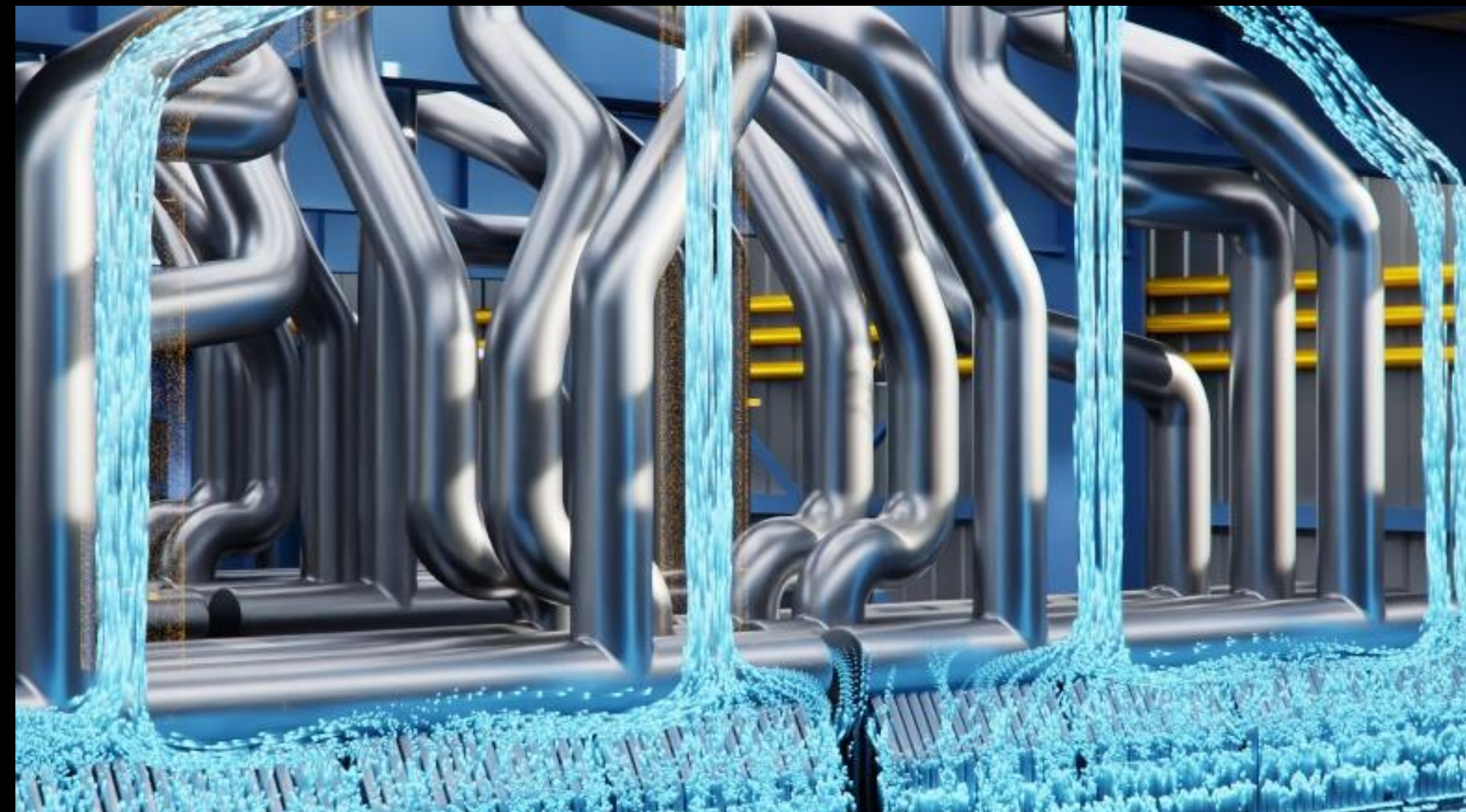
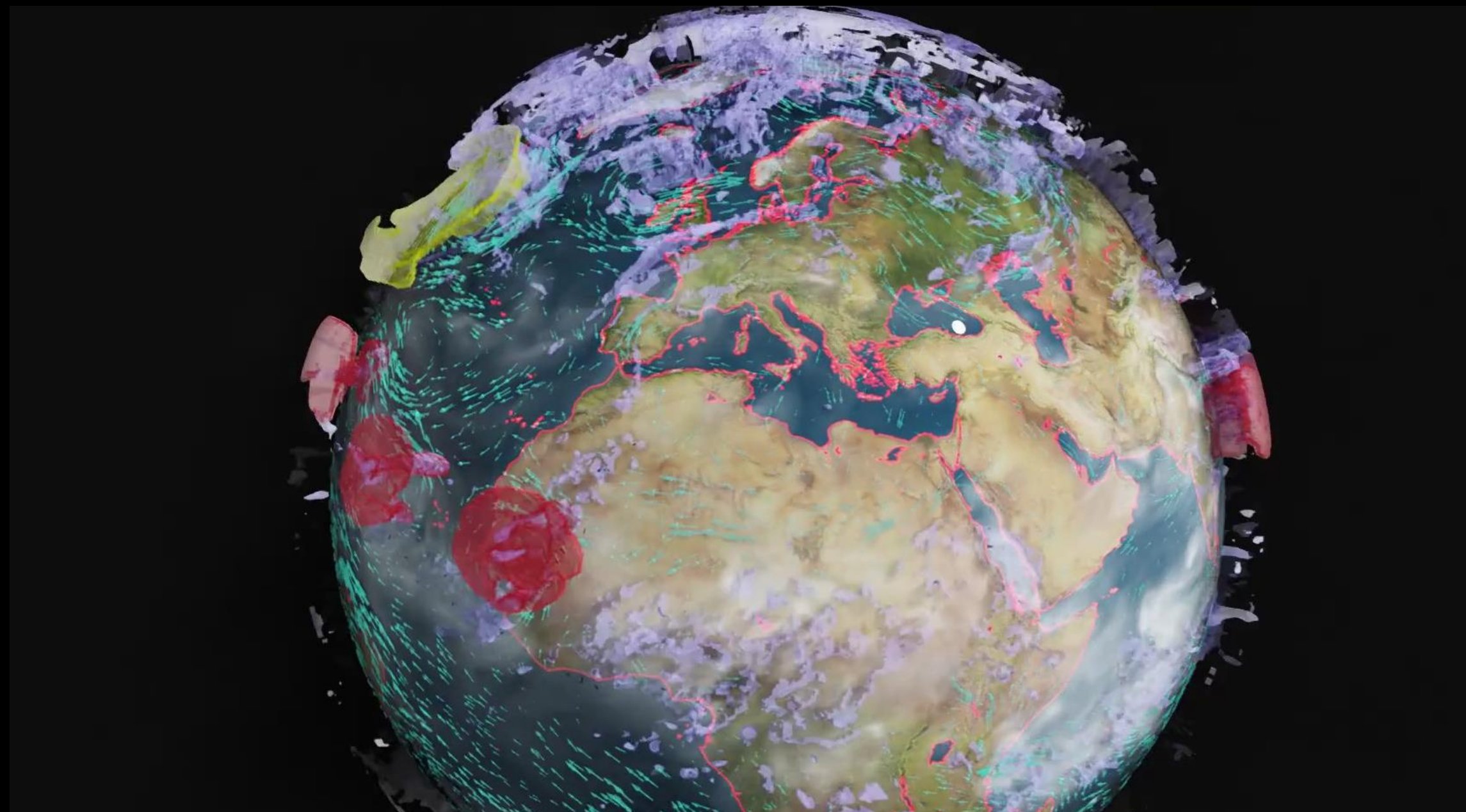
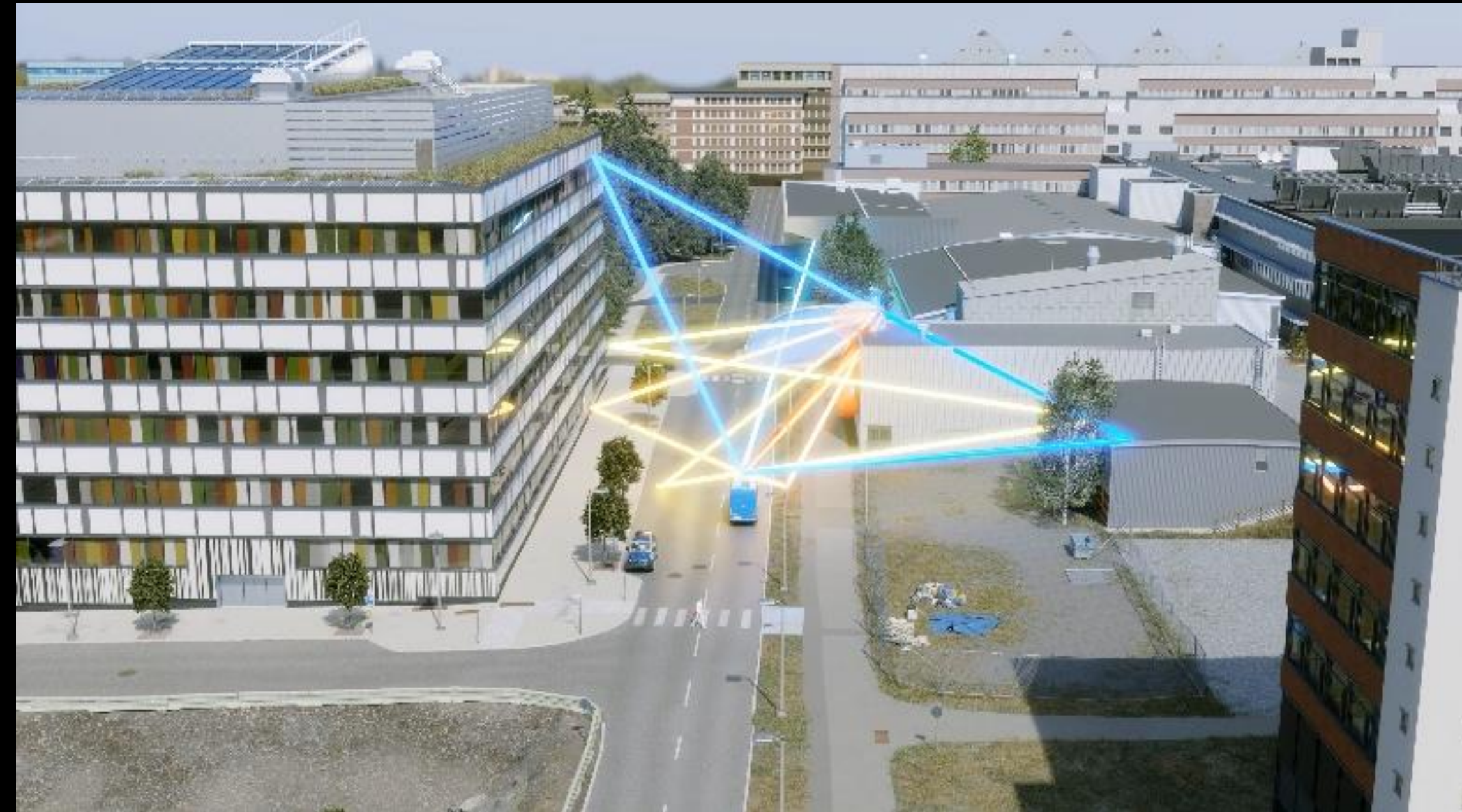
Optimising heat sink design

Modulus



OMNIVERSE

Virtual worlds are essential for the next era of AI



Reinforcement Learning – Isaac Gym



Digital Twin – BMW Factory



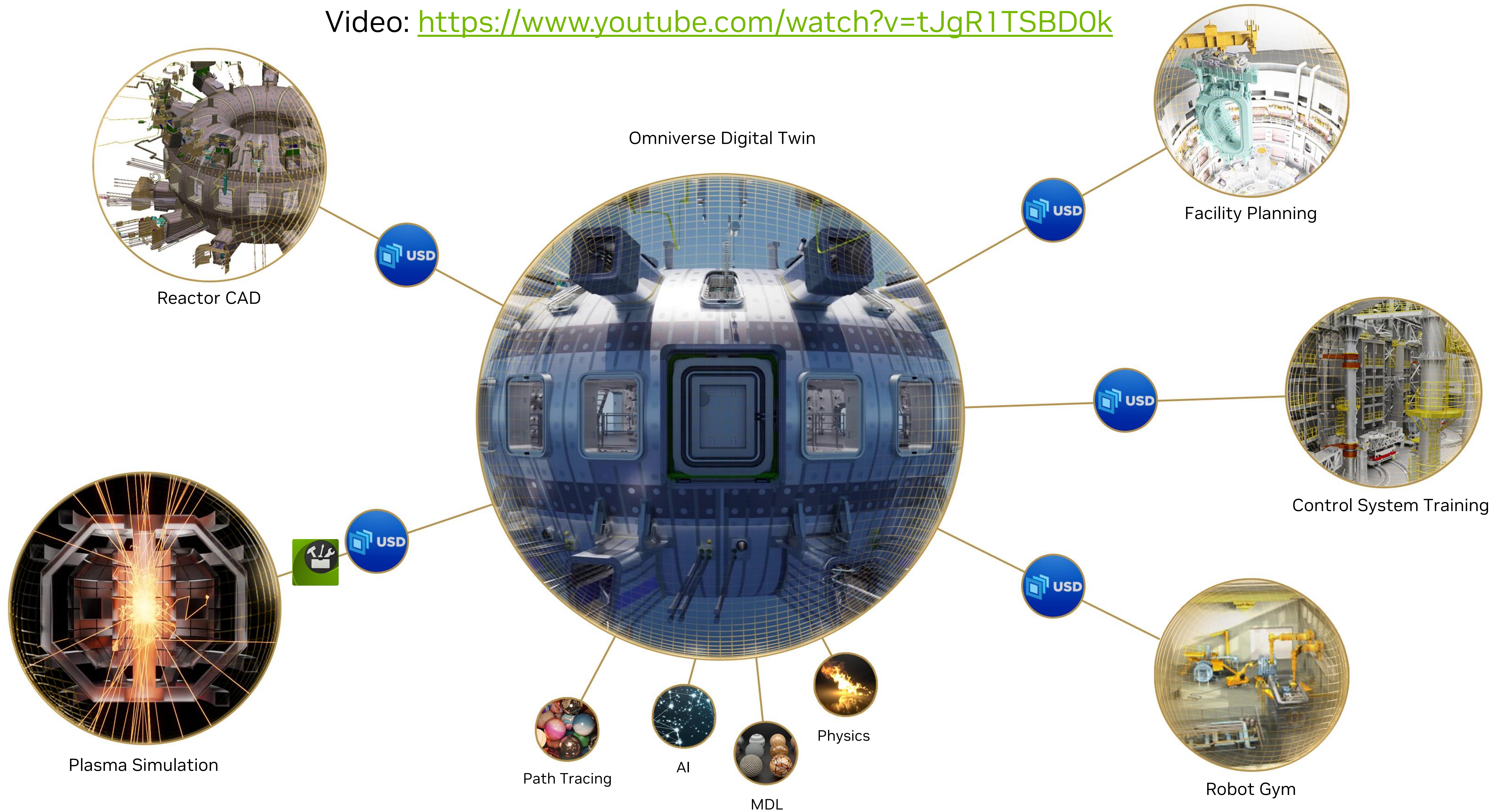
Maximizing Wind Energy Production Using Wake Optimization

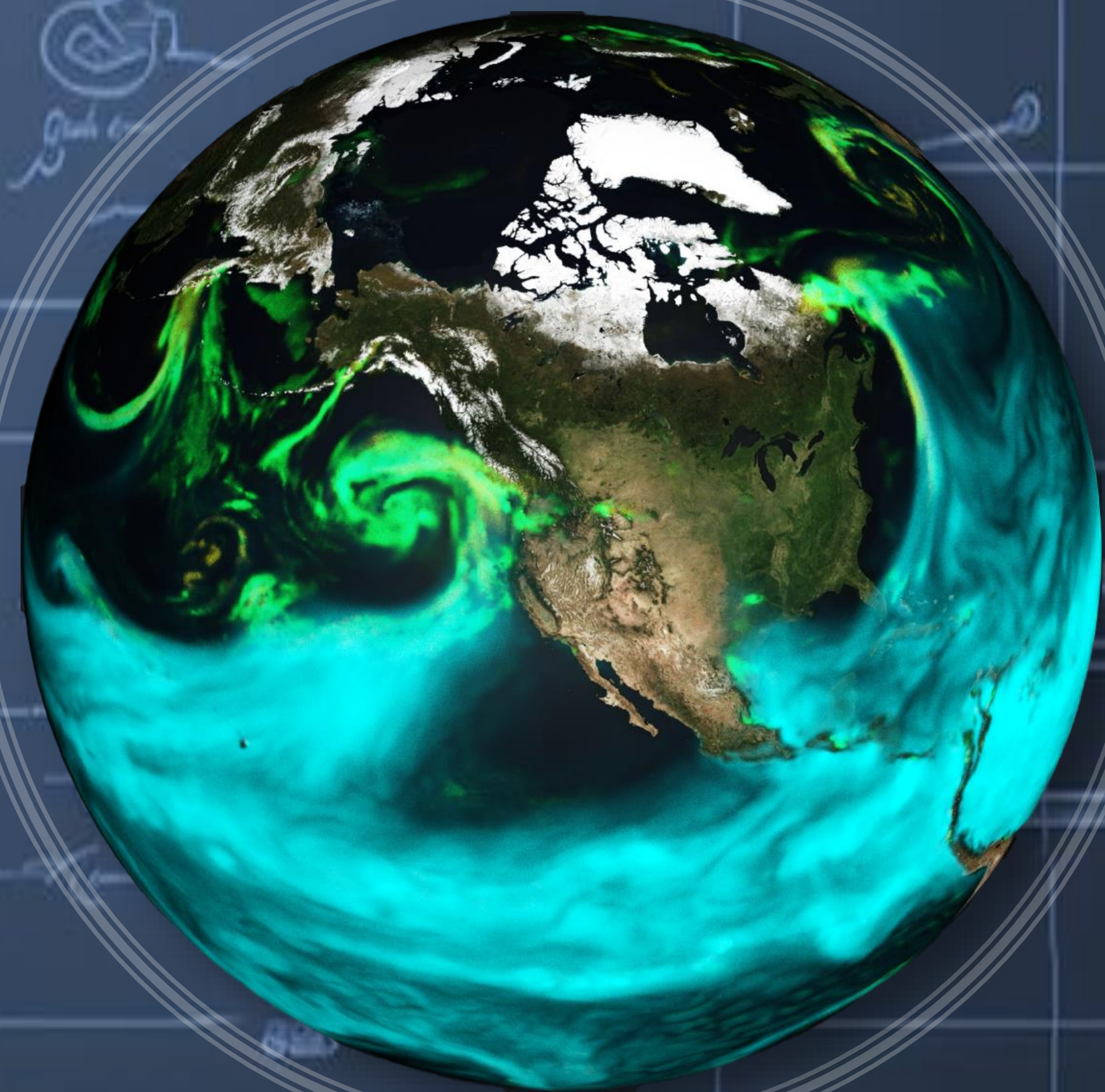
NVIDIA Modulus and Omniverse



UKAEA using Omniverse To Develop Fusion Reactor Digital Twin

Video: <https://www.youtube.com/watch?v=tJgR1TSBD0k>





FourCastNet

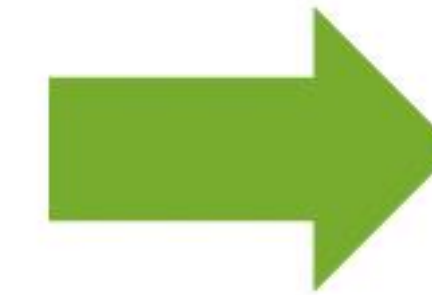
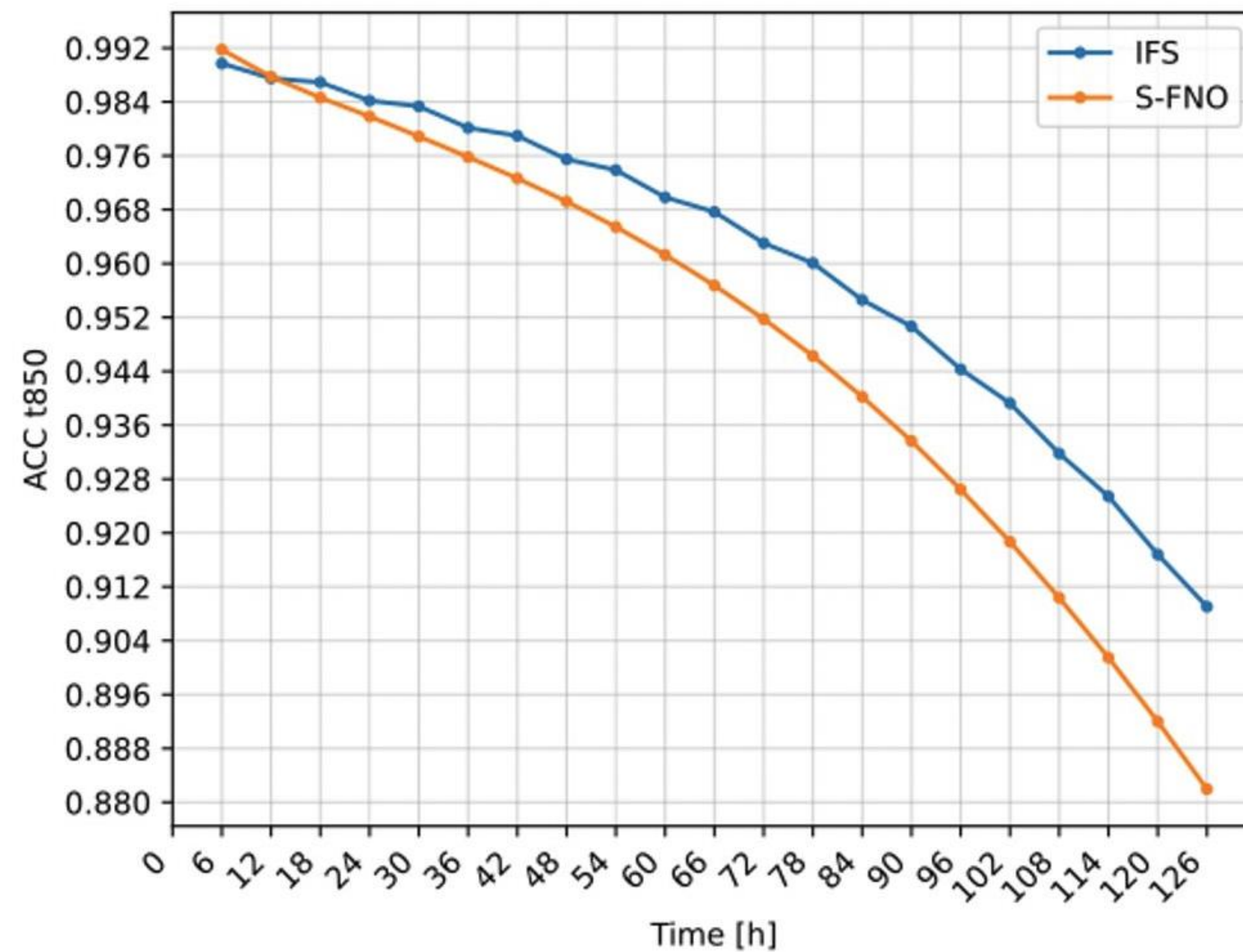
Full model emulation of ERA5

Scope	Global, Medium-Range Weather
Model Type	Full-Atmosphere AI Surrogate
Architecture	Fourier Neural Operator + Transformer
Resolution	25km
Training Data	ERA5 Reanalysis (1979 – Present)
Initial Condition	ERA5 / GFS / UFS
Training Time	$O(1000)$ GPU-hours (NVIDIA A100)
Inference Time	0.5 sec (2-week forecast)
Calibration	IC + Bayesian model uncertainty
Speedup vs NWP	$O(10,000 - 100,000)$
Power Savings	$O(10,000)$
Max Stable Rollout	250+ days
Project Type	Open-source

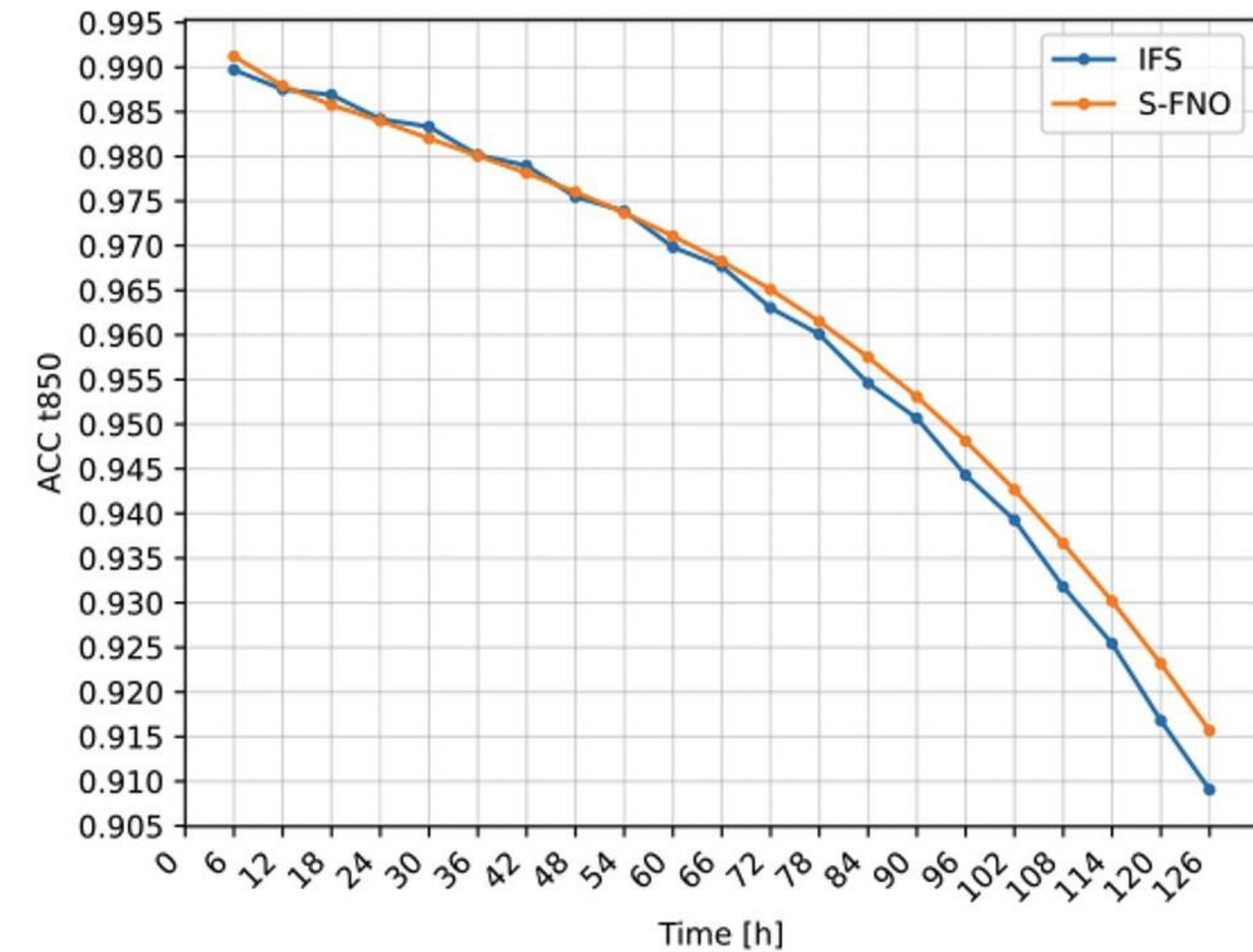
FCN is fully data-driven.
Atmospheric dynamics were learned from data alone.

- **Skill** now matching European Center for individual, 14-day forecasts.
 - Beating IFS on 60% of tracked metrics, 5-10% less error than IFS on 3-7 day timescale.

Before: IFS is more accurate



Now: SFNO becoming more accurate, a milestone.



FCN skill now matches or exceeds that of IFS (as measured by ACC and RMSE)

TORNADO VISUALIZATION

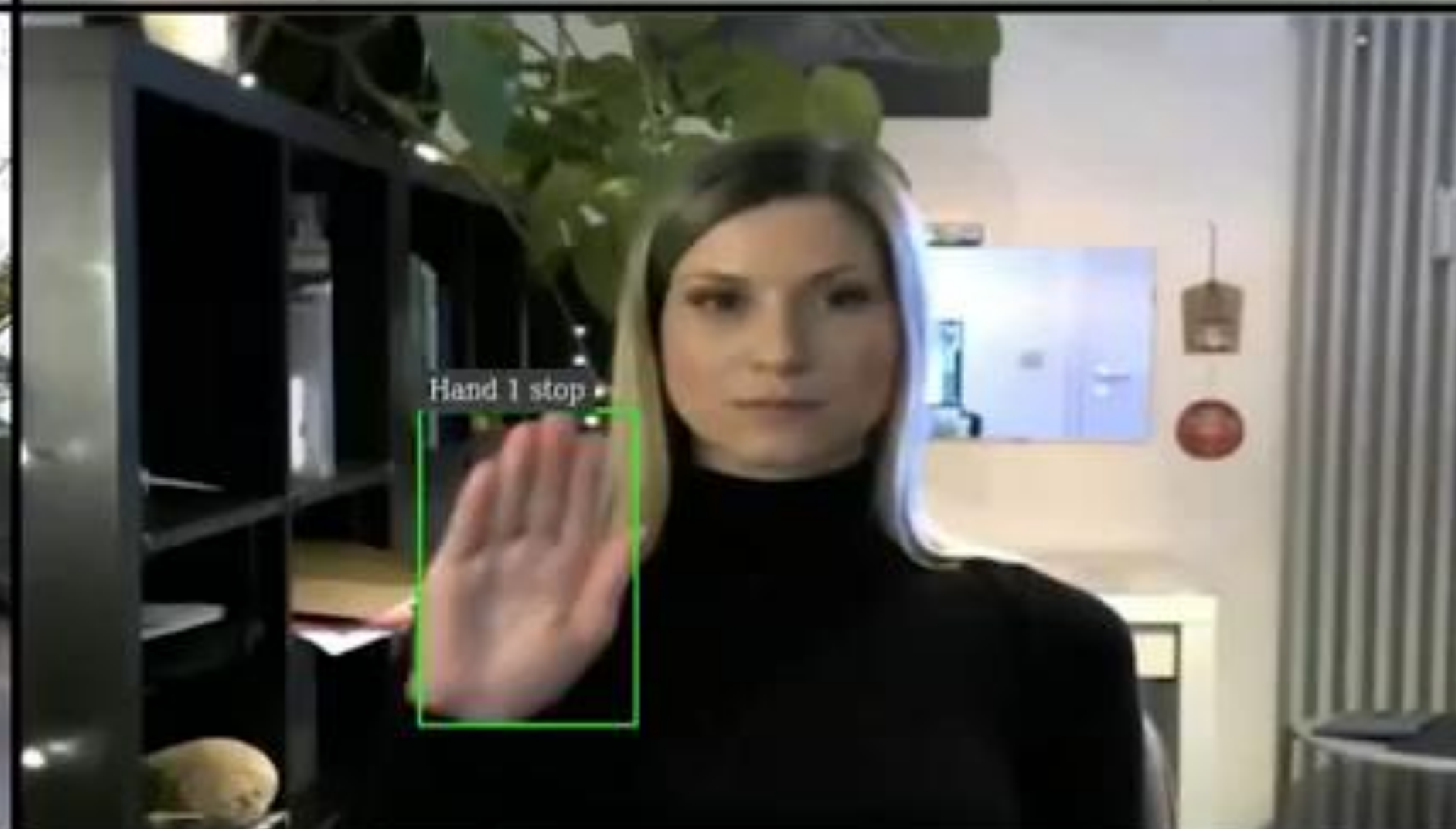
NanoVDB, IndeX, GPUDirect Storage, Iray, OmniVerse



The background of the slide features a dark green to black gradient with a series of bright green, curved, overlapping bands that create a sense of depth and movement. The text 'AI Frameworks' is positioned on the left side of the slide.

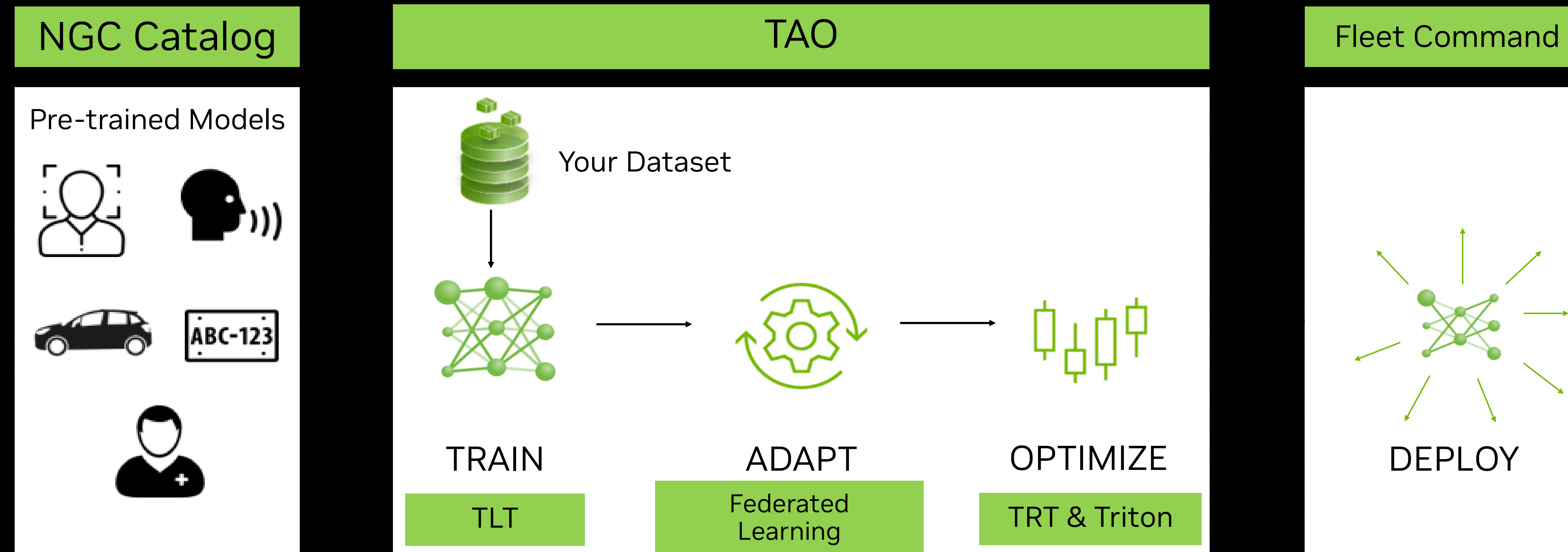
AI Frameworks

Fast track your AI with NVIDIA Pretrained models



NVIDIA TAO framework

Train | Adapt | Optimize



TRAIN

UI based framework simplifies AI development

Domain specific models in hours vs months

ADAPT & OPTIMIZE

Increase model accuracy with federated learning

Optimize with TensorRT

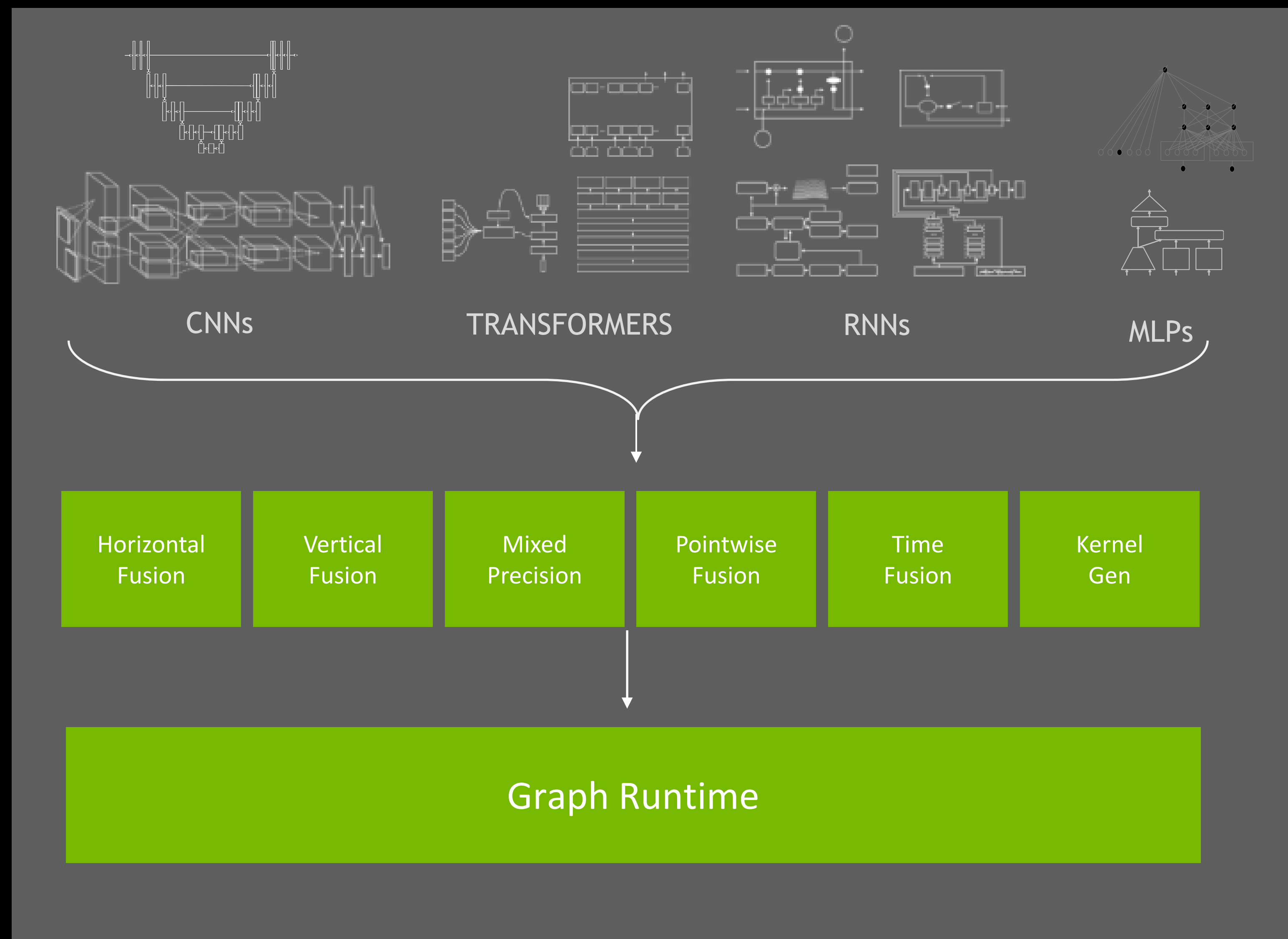
DEPLOY

Deploy from anywhere to everywhere

Effortless management and secure deployments

TensorRT

SDK for High-Performance Deep Learning Inference



Optimize and Deploy neural networks in production

- Maximize throughput for latency-critical apps with compiler & runtime
- Deploy responsive and memory efficient apps
- FP32, TF32, FP16 & INT8
- Optimize every network including CNNs, RNNs and Transformers
- Accelerate every framework - ONNX support, TensorFlow integration
- Run multiple models on a node with containerized inference server

DeepStream SDK

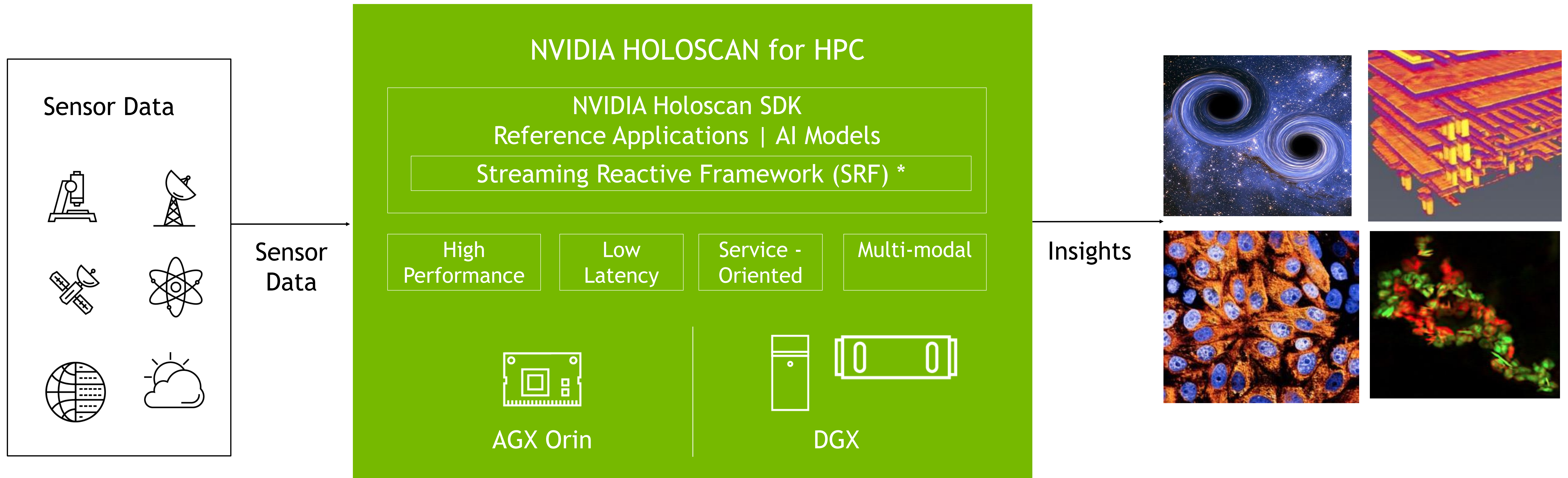
Transform pixel and sensor data to actionable insights

- **Streaming analytics toolkit** for AI-based multi-sensor processing, video and image understanding with **TLS security**
- Deploy on the edge and connect to any cloud
- **C/C++ and python** choice of development
- Extensive AI model support: **SSD, YOLO, FasterRCNN, and MaskRCNN** and more
- **Flexibility** for rapid prototyping to full production
- **Speed up overall development efforts** by training with TLT and deploying with DS
- **Turnkey integration** with AWS IoT and Azure IoT
- Select from 15+ existing DS custom plugins or create your own

Deploy Your Vision AI Application Anywhere



HOLOSCAN SDK FOR HPC AT THE EDGE

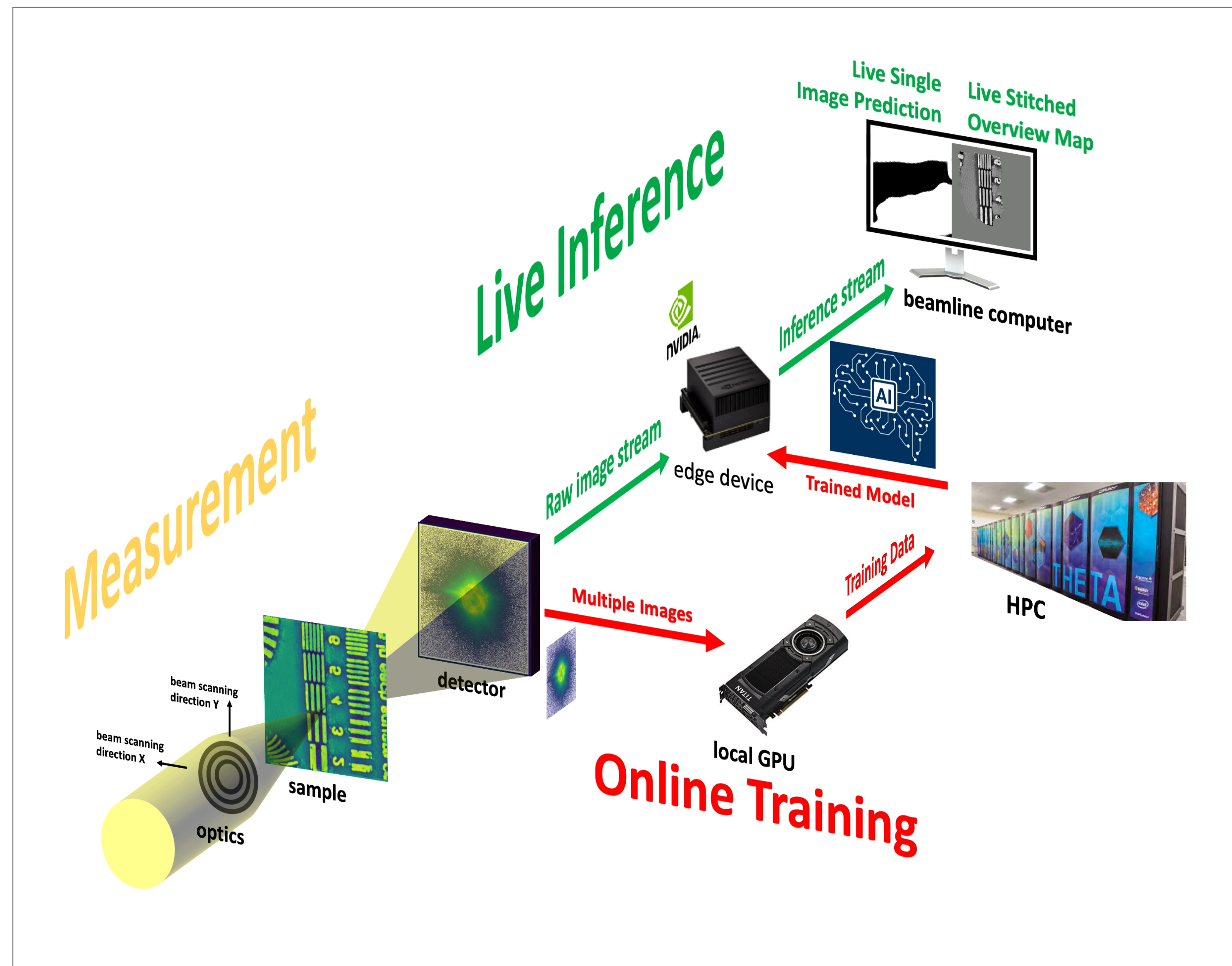


- Streaming, Flexible, Scalable
- Modular, Reusable, Multi-Modal
- Authenticate, Monitor, Orchestrate

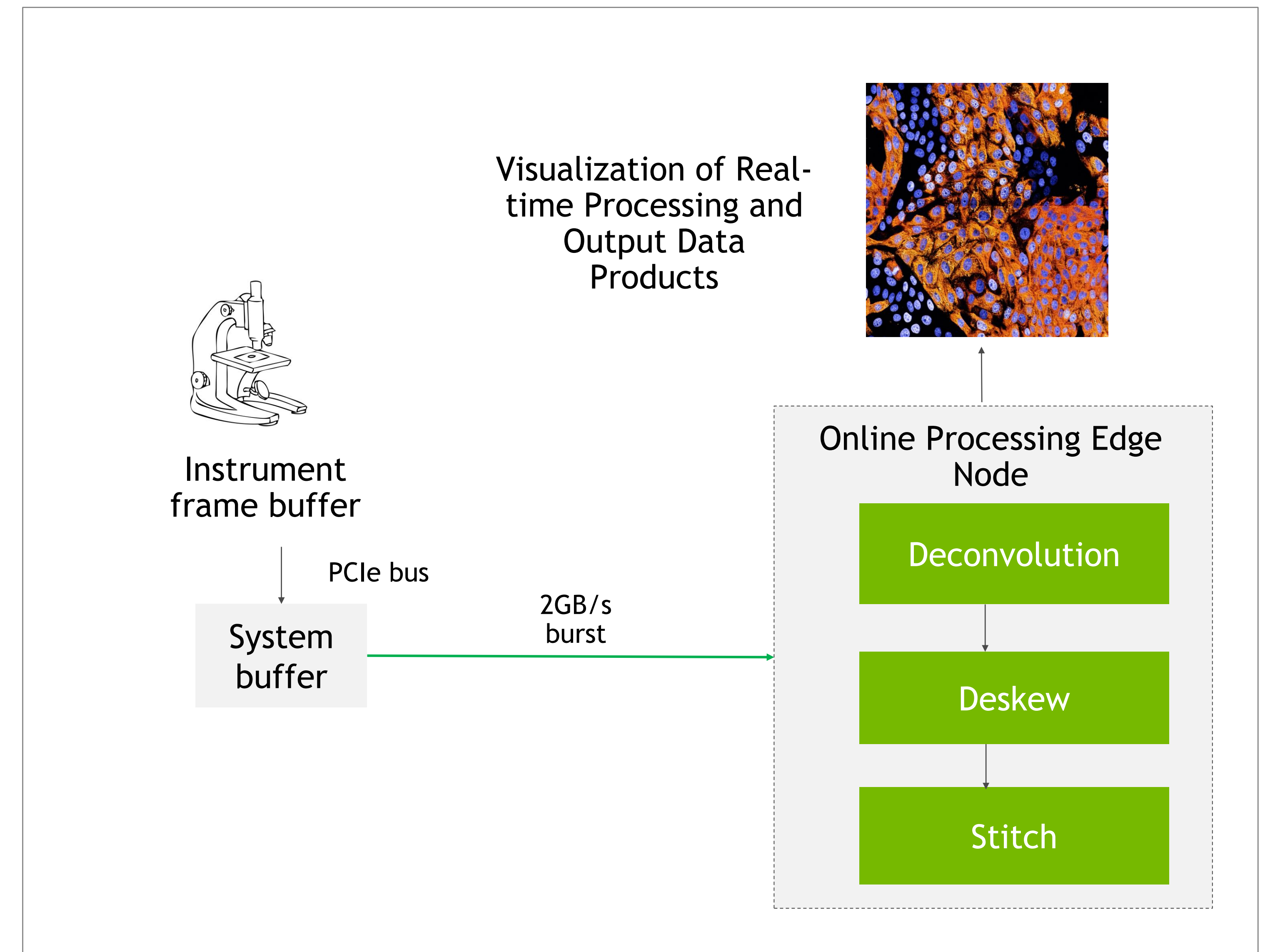
* (SRF) Available on [GitHub](#) mid-June

SUPERCHARGING SCIENCE EXPERIMENTS AND INSTRUMENTS

ANL/ APS ACCELERATES X-RAY PTYCHOGRAPHY 300X WITH PTYCHONN



ADVANCED BIOIMAGING CENTER @UC-BERKELEY REAL TIME LIVE CELL IMAGING LIGHT SHEET MICROSCOPY



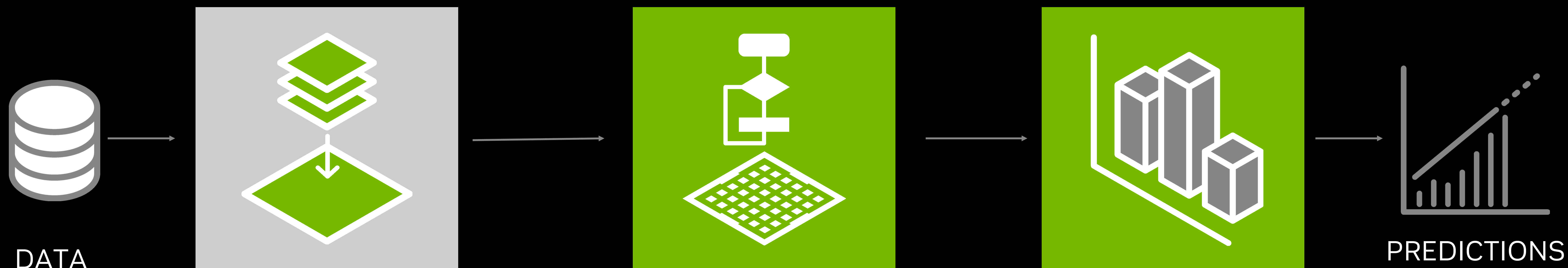
PtychoNN paper: [AI-enabled high-resolution scanning coherent diffraction imaging](#)

Link to keynote video - <https://youtu.be/rXG27G3bWzY>

RAPIDS

GPU-ACCELERATED DATA SCIENCE WORKFLOW

RAPIDS.ai



DATA PREPARATION - ETL

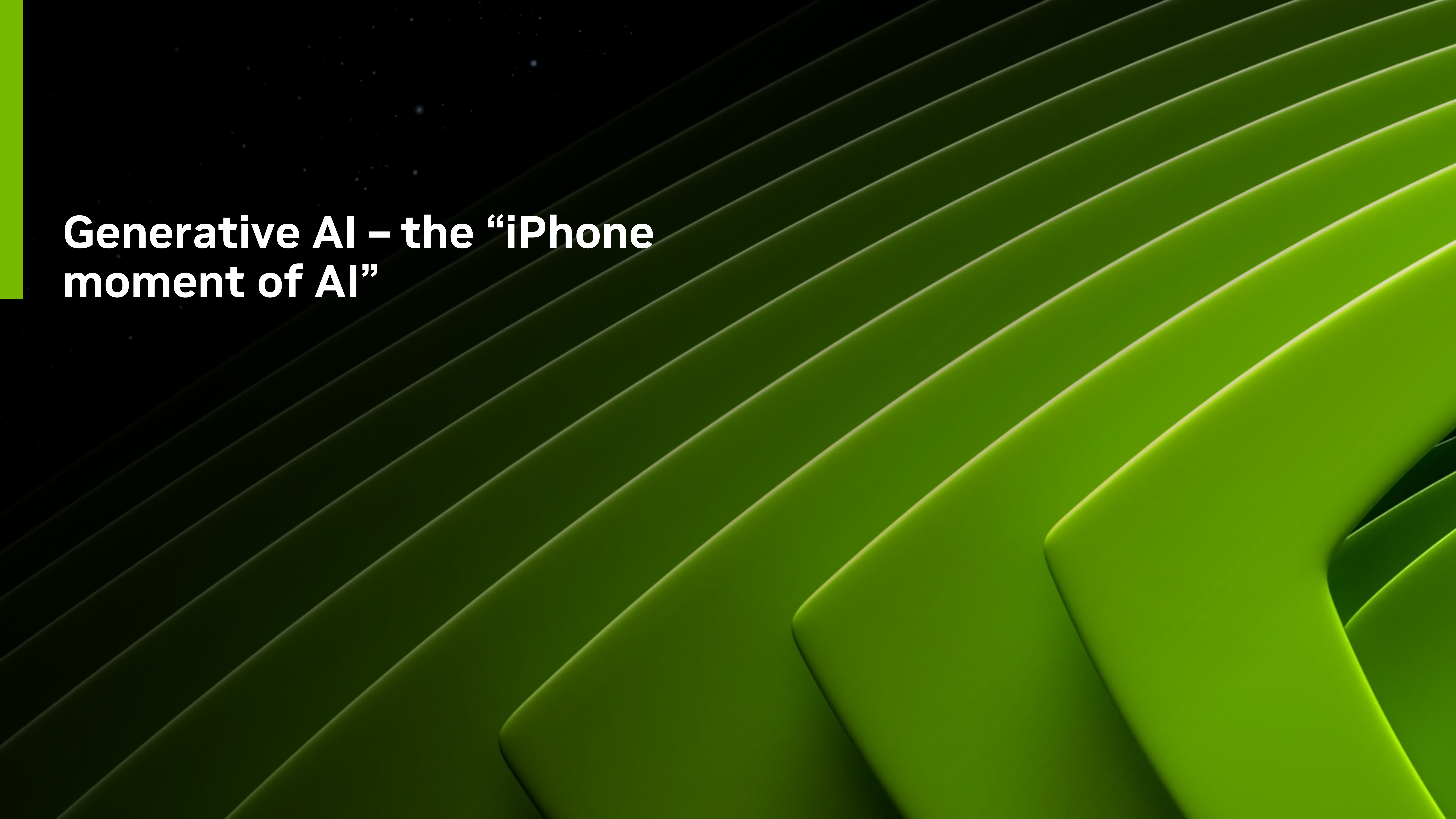
cuDF: Python drop-in **pandas** replacement built on CUDA.
GPU-accelerated Spark

MODEL TRAINING

cuML: GPU-acceleration of popular ML algorithms e.g. **XGBoost**
Easy-to-adopt, **scikit-learn** like interface

VISUALIZATION

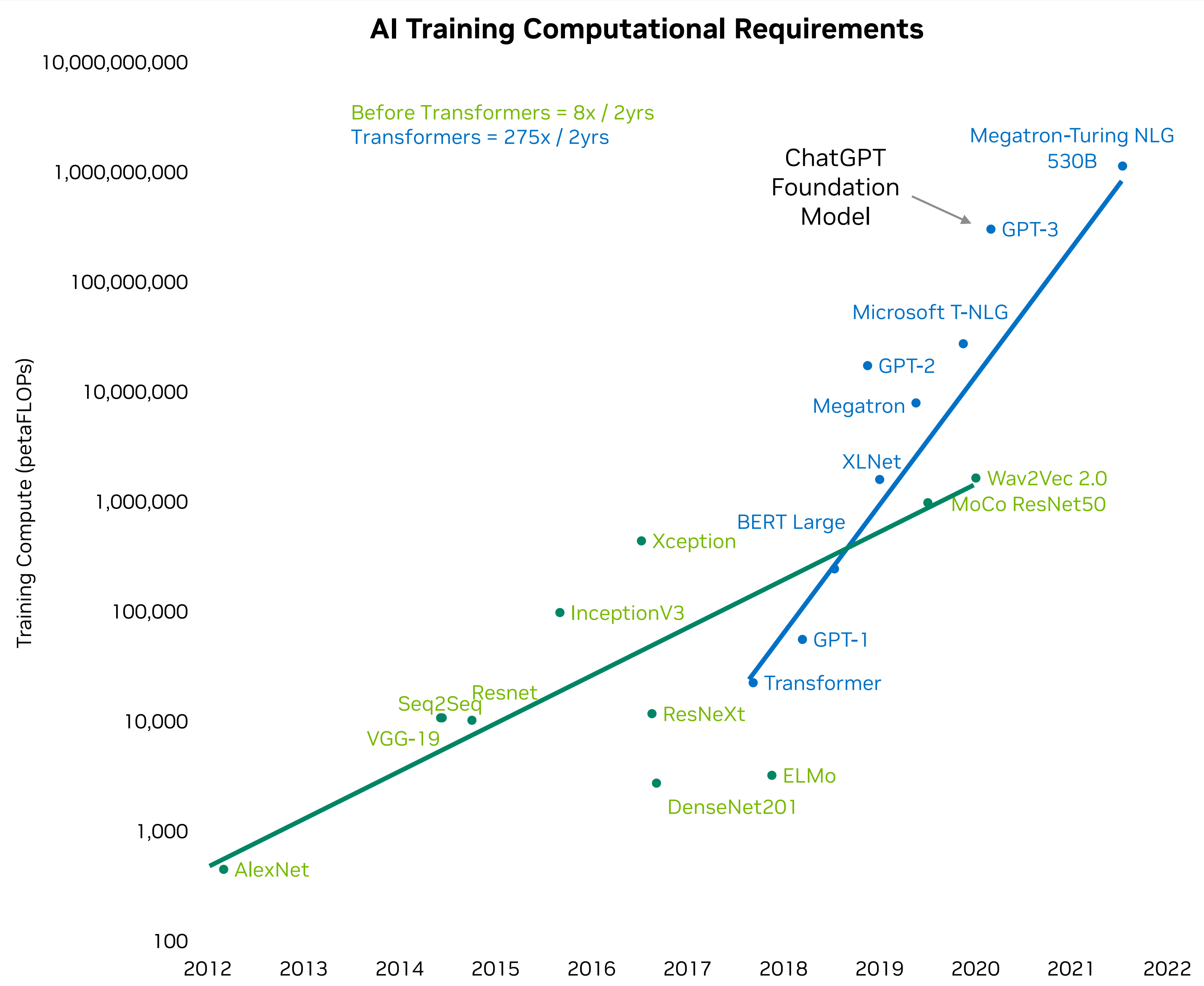
Effortless exploration of datasets, billions of records in milliseconds
Dynamic interaction with data = faster ML model development



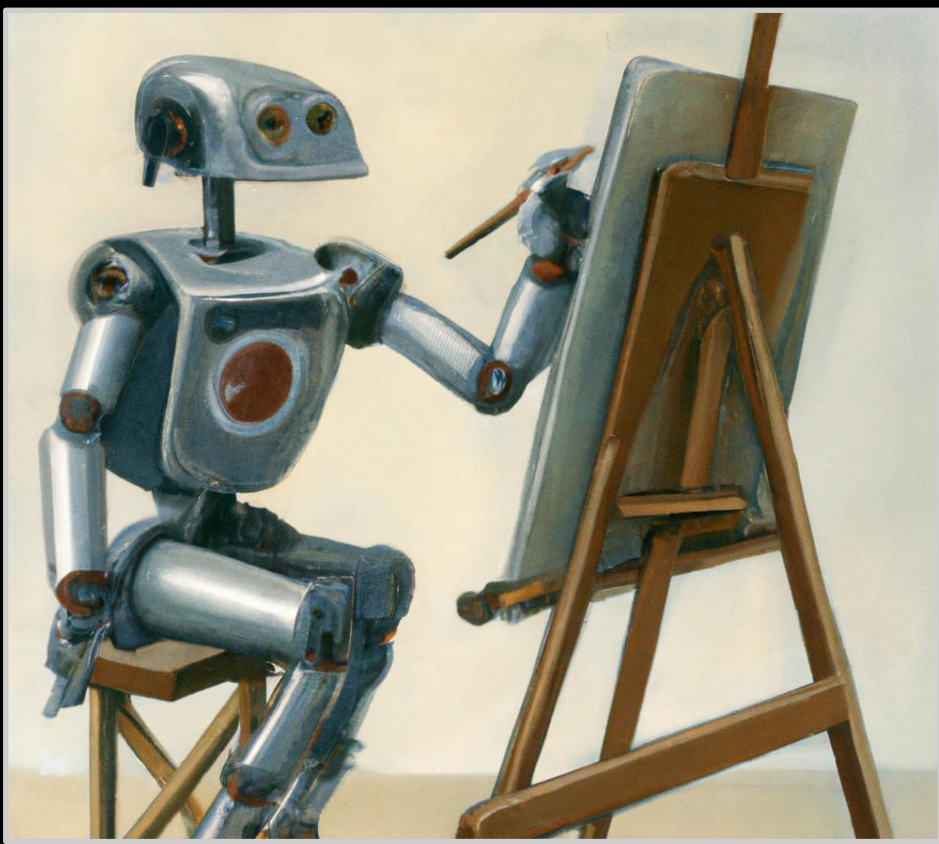
**Generative AI – the “iPhone
moment of AI”**

Large Language Models Codifying Intelligence

LLMs Require Exponential Compute Growth



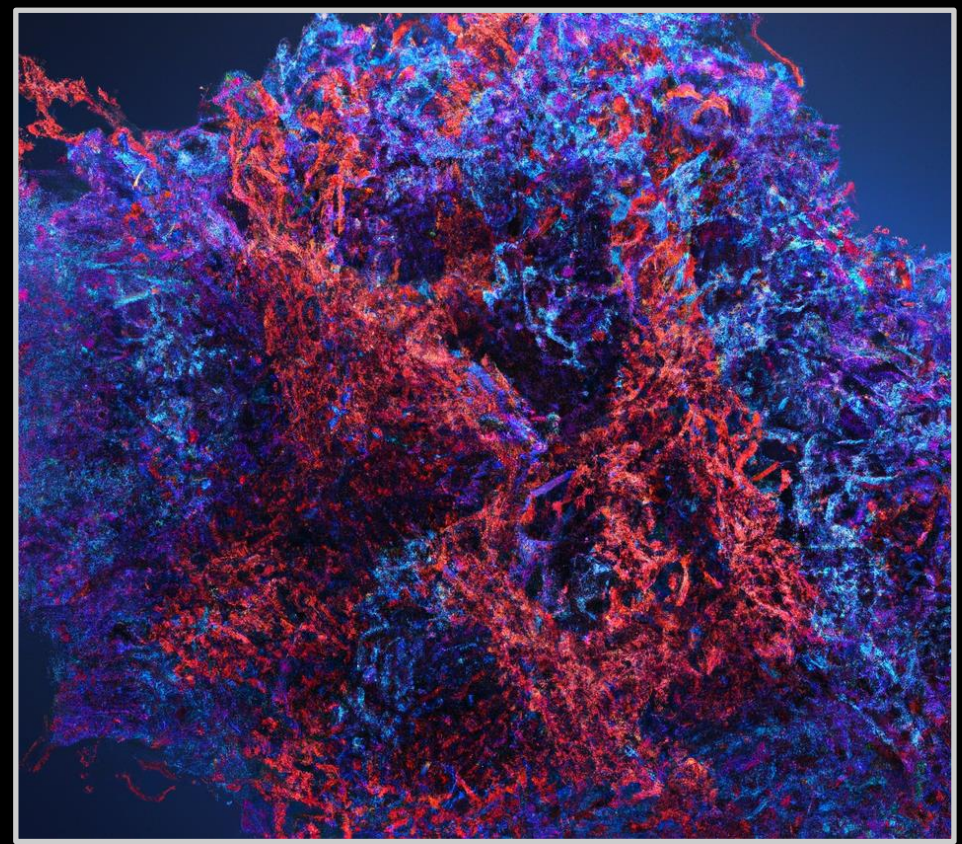
Brand Creation
Gaming Characters



eCommerce
Personalized Content



Molecular Representations
Drug Discovery



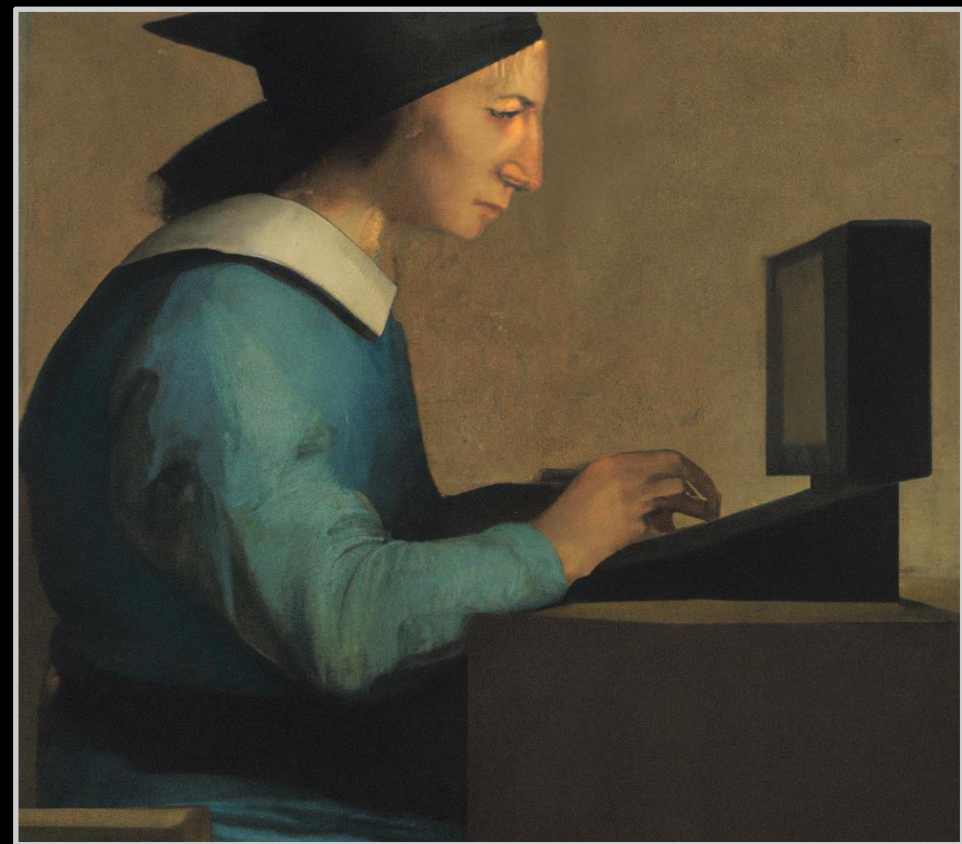
Translating Wikipedia
Real-Time Metaverse Translation



Summarization
Marketing Copy



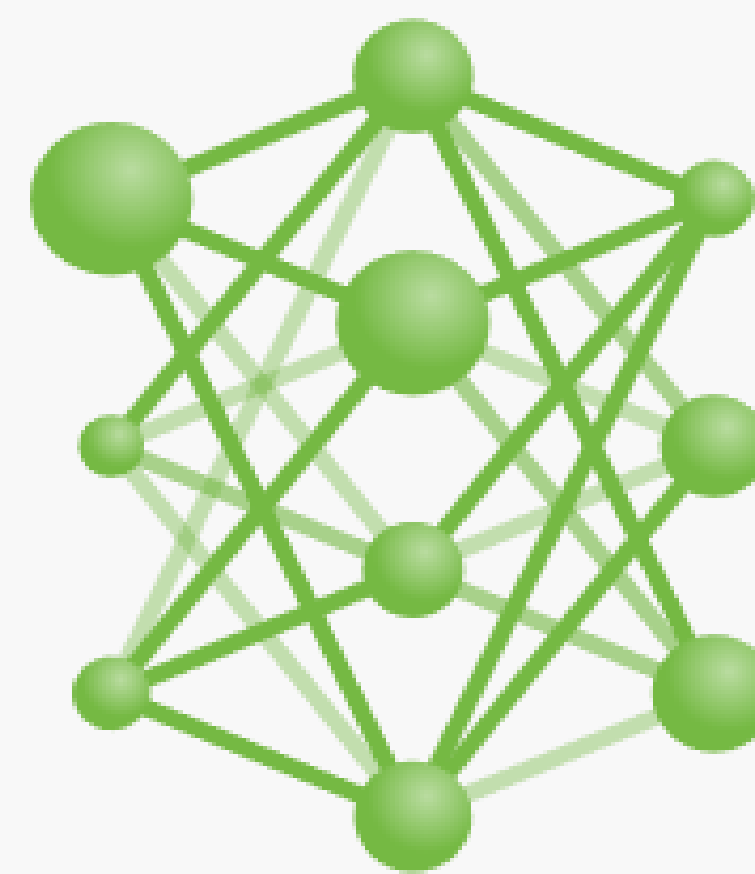
Dynamic Code Comments
Function Generation



Large language model

Generative Pre-trained Transformer - GPT

The cat is playing in the



Next Token Prediction



- garden
- kitchen
- ground
- grass
- ...

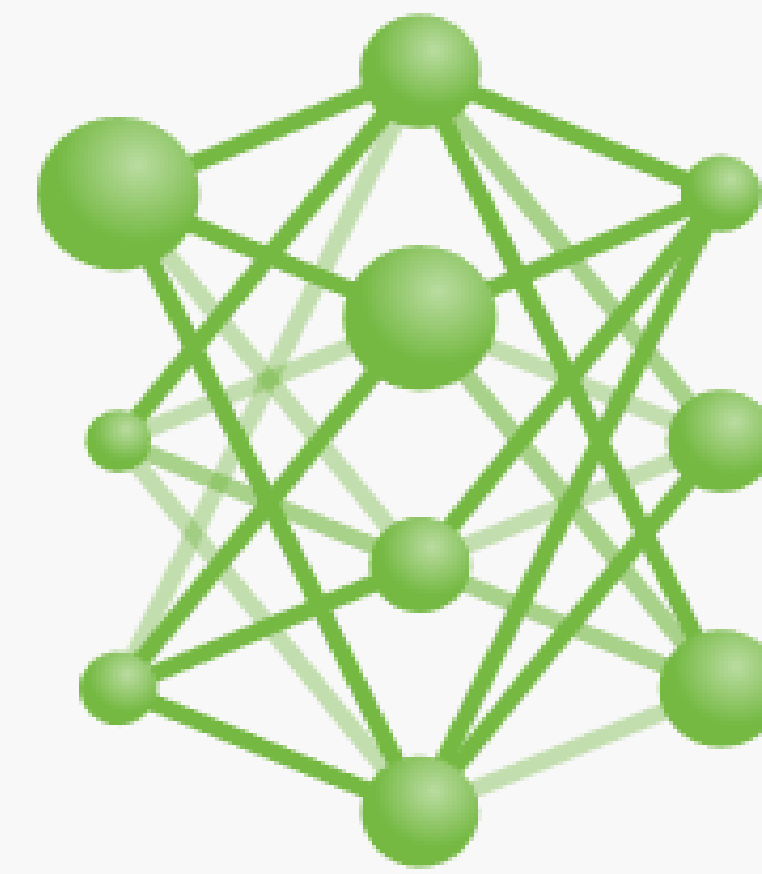
Large language model

Generative Pre-trained Transformer - GPT

Predicted Next token over the vocab

The cat is playing in the

Prompt



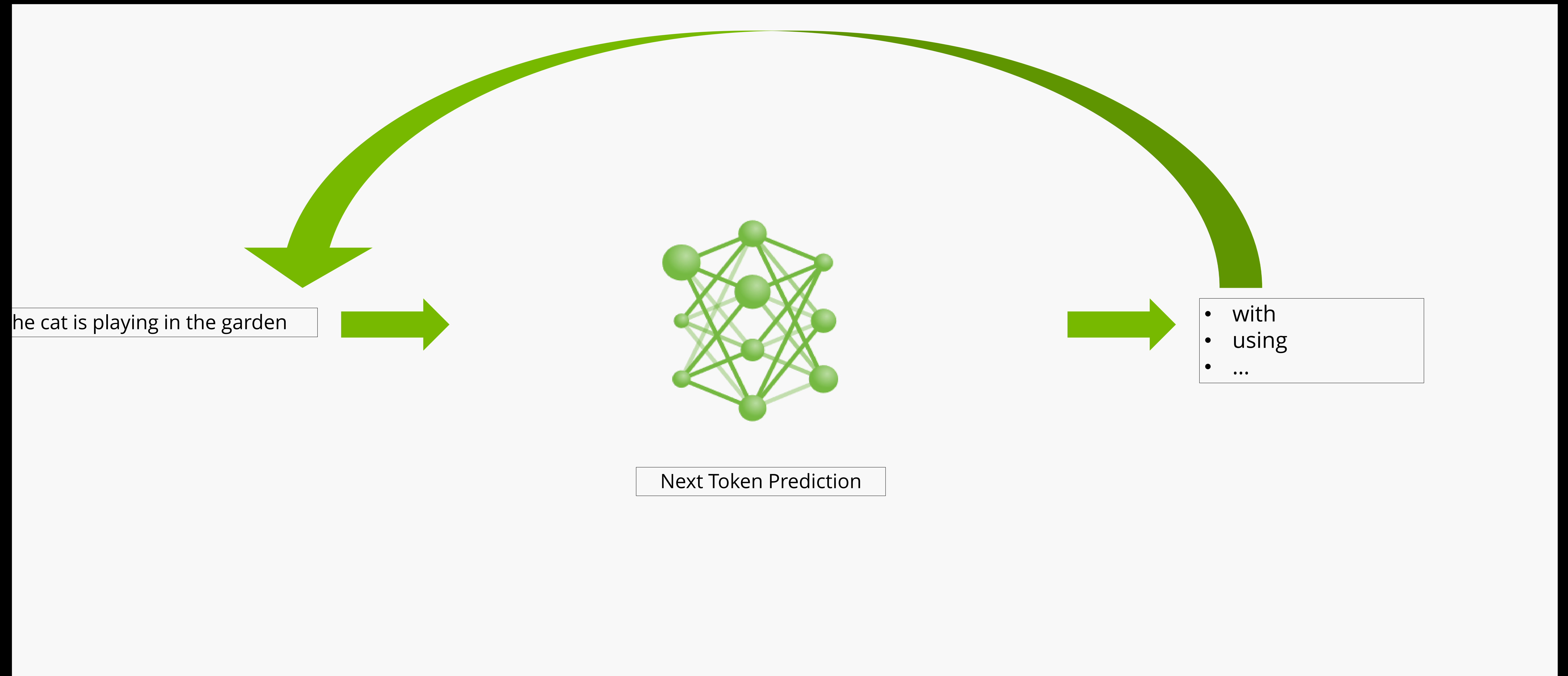
Next Token Prediction



- garden
- kitchen
- ground
- grass
- ...

Large language model

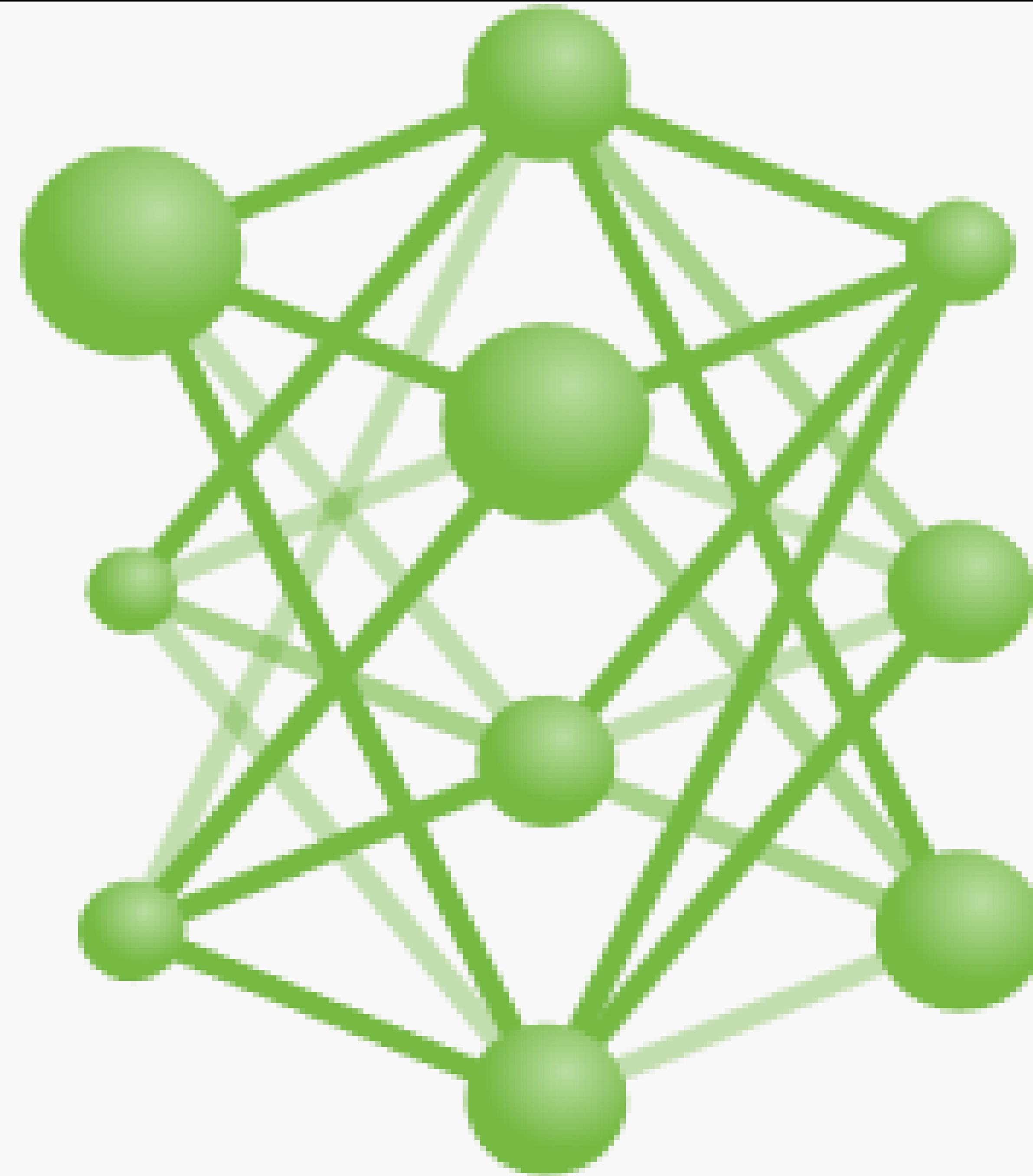
Auto-Regressive Loop



Large language model

GPT | GPT2 | GPT3

The cat is playing in the garden



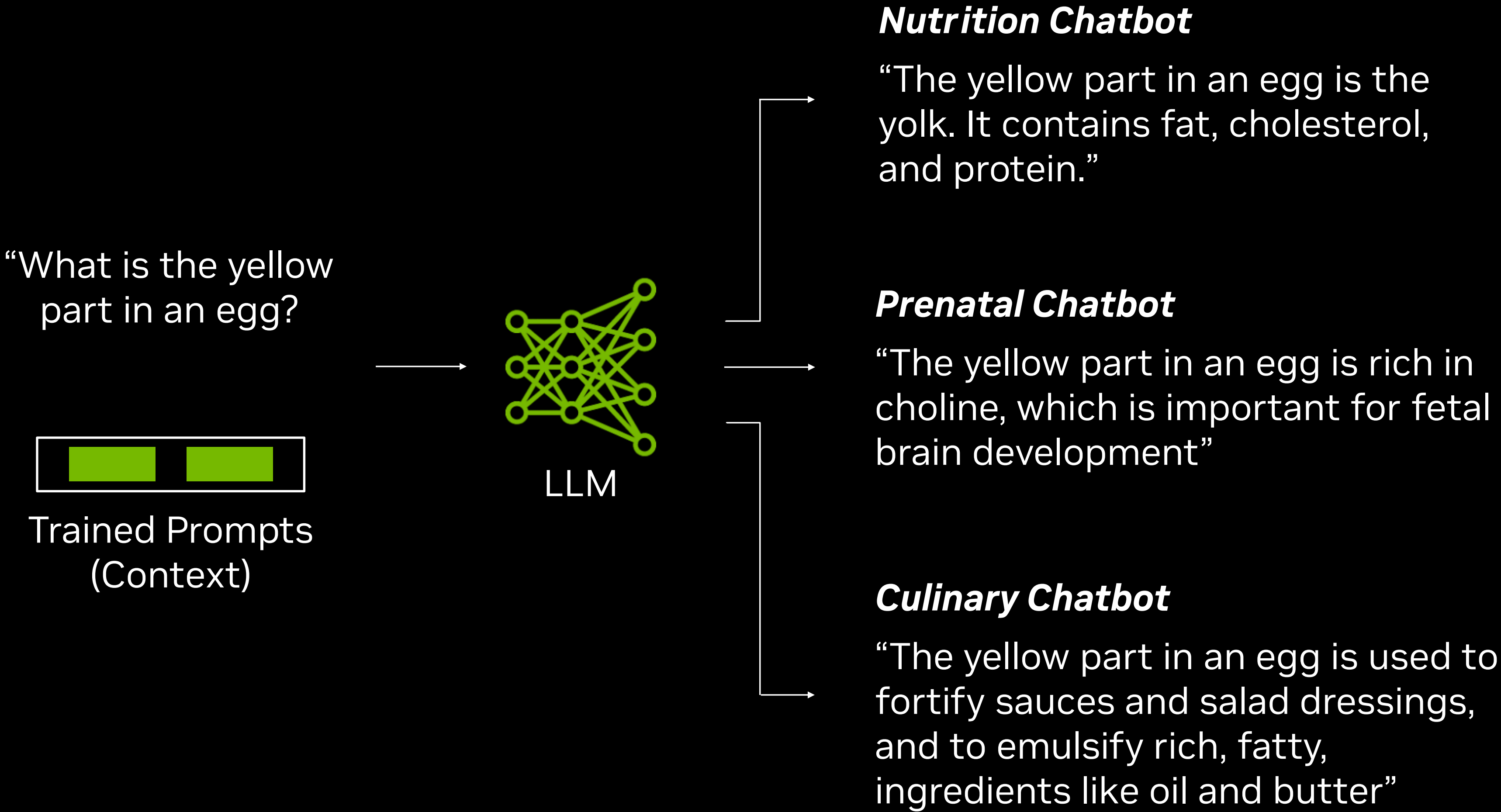
Next Token Prediction



- with
- using
- ...

Foundation Models Customization

P-Tuned Response



Retrieval Augmented Generation (RAG)



From Language Model to powerful Problem Solver

GPT-4: tool warmup

The computer is answering questions. If the computer needs any current information to answer the question, it searches the web by saying `SEARCH("query")`, reads the snippets in the result, and then answers the question. If it needs to run any calculations, it says `CALC(expression)`, and then answers the question. If it needs to get a specific character from a string, it calls `CHARACTER(string, index)`.
</endofprompt>

Q: Who is the current president of the United States?

Computer: `SEARCH("current US president")`

> SEARCH SNIPPETS

[1] Joe Biden: The President | The White House

President Joe Biden Vice President Kamala Harris First Lady Dr. Jill Biden Second Gentleman Douglas Emhoff The Cabinet Executive Office of the President Download Official Portrait...

[2] Who is the current US president? | The US Sun

Who is the current US president? Katrina Schollenberger Jennifer Roback NY Published: 8:30 ET, Jul 2 2022 Updated: 8:39 ET, Jul 2 2022 DONALD Trump controversially defeated Hillary Clinton in the 2016 Presidential Election, making him the 45th president of the United States.

[3] President of the United States - Wikipedia

Joe Biden is the 46th and current president of the United States, having assumed office on January 20, 2021. Contents 1 History and development 1.1 Origins 1.2 1789-1933 1.3 Imperial Presidency 1.4 Critics of presidency's evolution 2 Legislative powers 2.1 Signing and vetoing bills 2.2 Setting the agenda 2.3 Promulgating regulations

Computer: Joe Biden

Q: What is `sqrt(34324*2432)`?

Computer: `CALC(sqrt(34324*2432))`

> 9136.518374085394

Computer: 9136.518374085394

Q: What is the 13th letter of the word 'supralapsarian'?

Computer: `CHARACTER('supralapsarian', 13)`

> a

Computer: The 13th letter of the word 'supralapsarian' is a.

GPT-4

Prompt: Can you write a proof that there are infinitely many primes, with every line that rhymes?

GPT-4:

```
Yes, I think I can,  
Though it might take a clever plan.  
I'll start by noting Euclid's proof,  
Which shows that primes aren't just aloof.  
Assume we have a finite list,  
Of primes, and that none have been missed.  
Multiply them all together,  
And add one, just to be clever.  
The result will be a number,  
That has no prime factors, oh wonder!  
But every number has a prime,  
That divides it, it's just a matter of time.  
So we've found a contradiction,  
And our finite list needs eviction.  
There must be infinitely many primes,  
And that's the end of my rhyming lines.s
```

Prompt: Draw a unicorn in TikZ.

GPT-4: [Produces `LaTeX` compiling to following picture.]

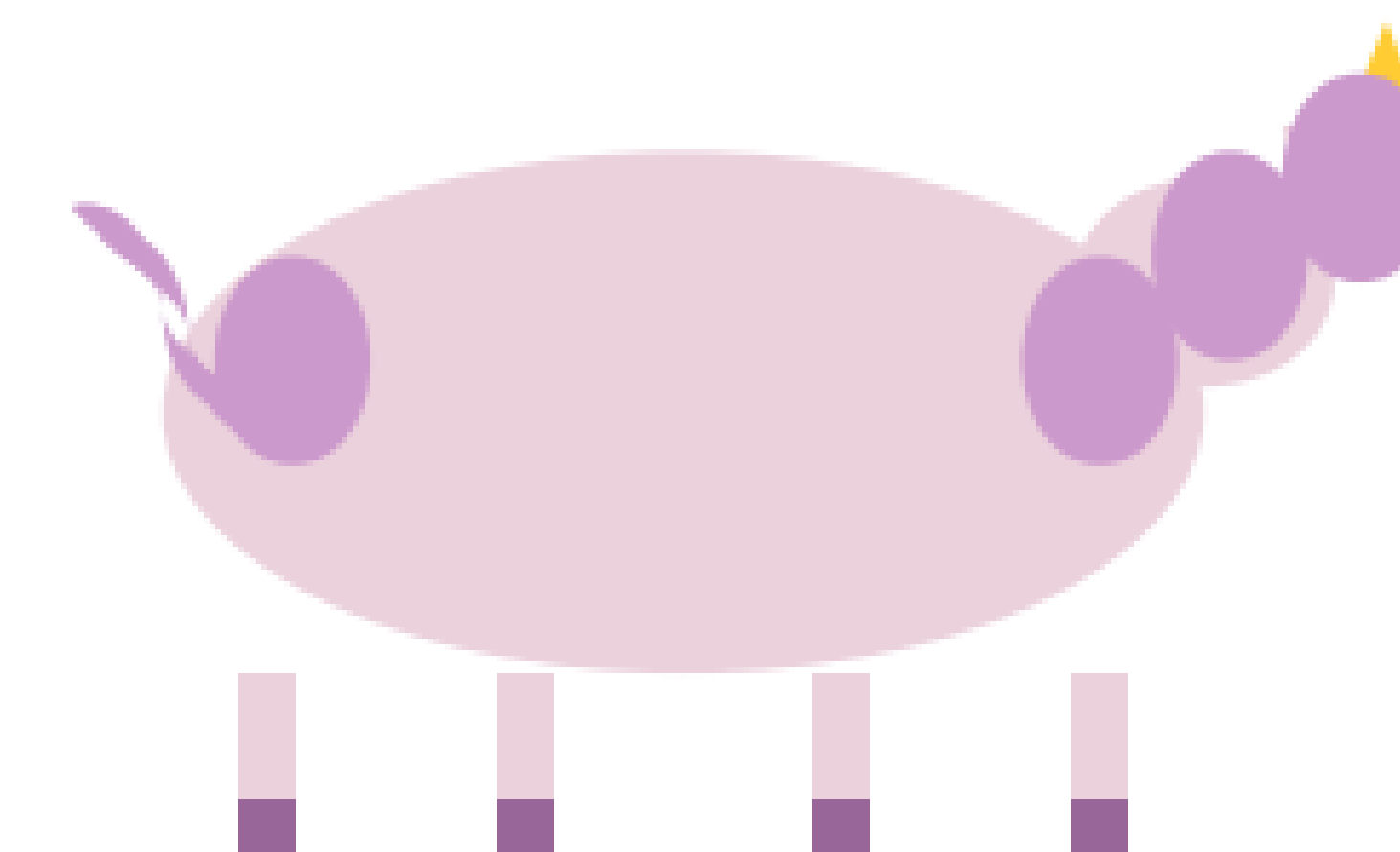
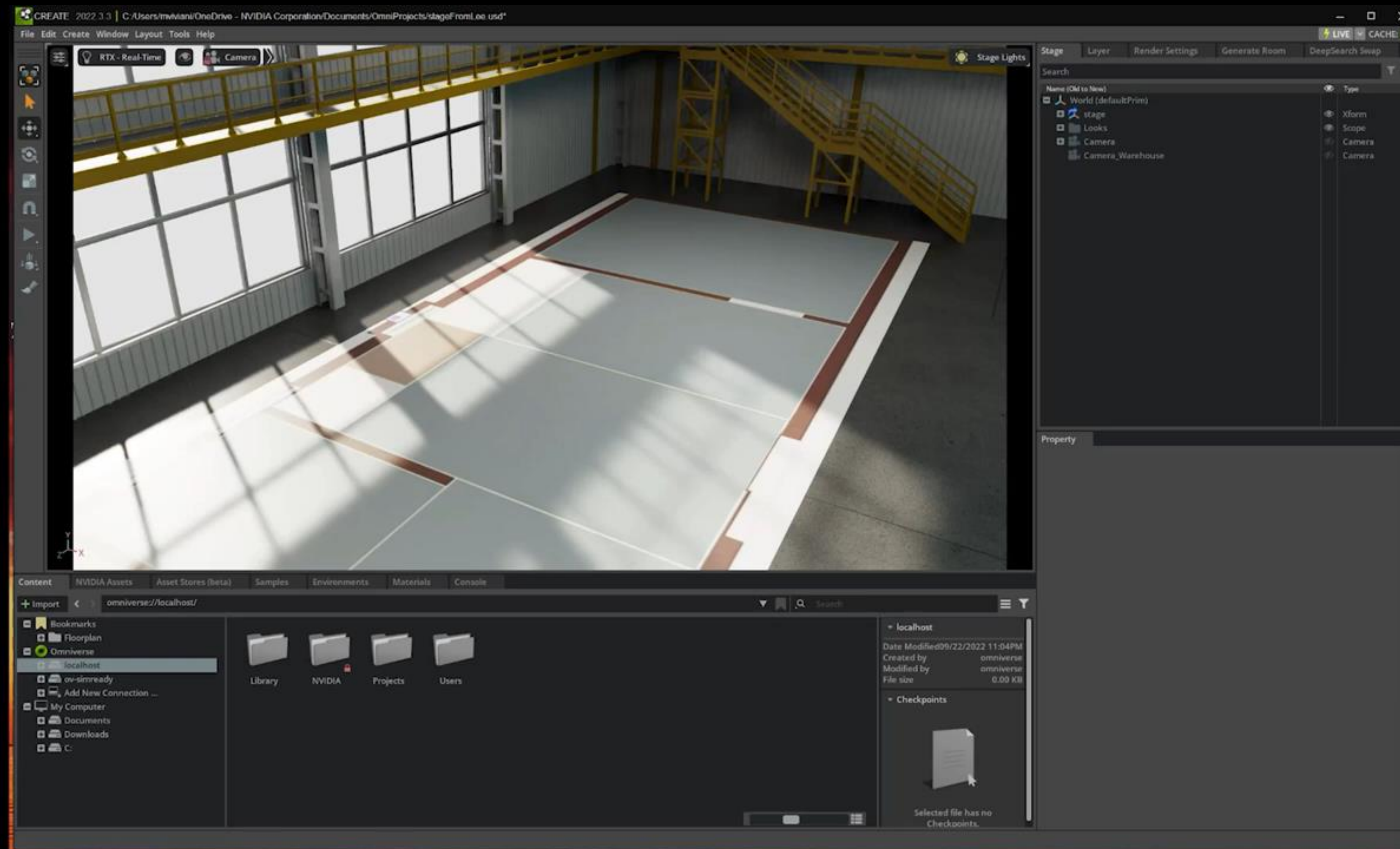


Figure 5.2: Examples of GPT-4 using various tools to answer questions.

Example: UI automation

Using ChatGPT and GPT-4 to Generate 3D Content in Omniverse



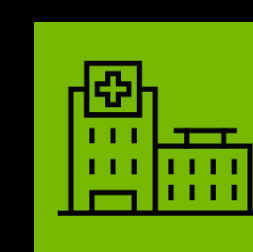
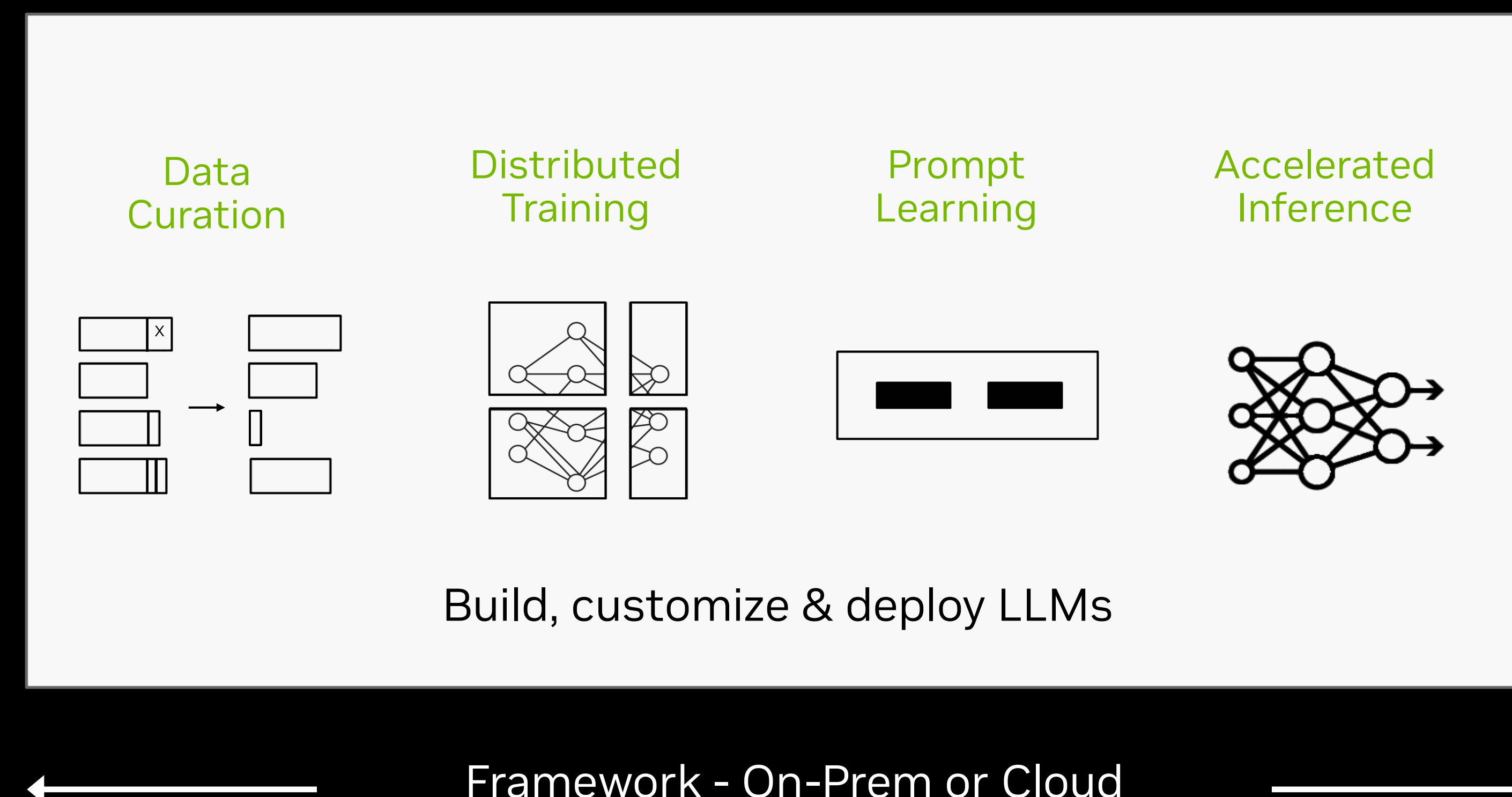
NVIDIA BioNeMo Framework

AI-powered drug discovery

AI Framework for BioMolecular Large Language Models

BioNeMo Framework

ESM-1, ProtT5, MMB



Domain Specific

Drug discovery researchers need an LLM framework that speaks the language of biology and chemistry.

Optimized For Scale

LLMs are massive requiring model parallelism and supercomputing resources to train.

Pre-Trained Models

Training LLMs takes enormous compute resources and is very time consuming. Pre-trained models are ready for production.

Cloud Native

Use pre-trained models, create powerful customized workflows and deploy in the cloud.

Expert, Natural Q&A with NVIDIA Omniverse Avatar for Project Tokkio

NVIDIA Tokkio Showcase and Omniverse Avatar Cloud Engine (ACE)

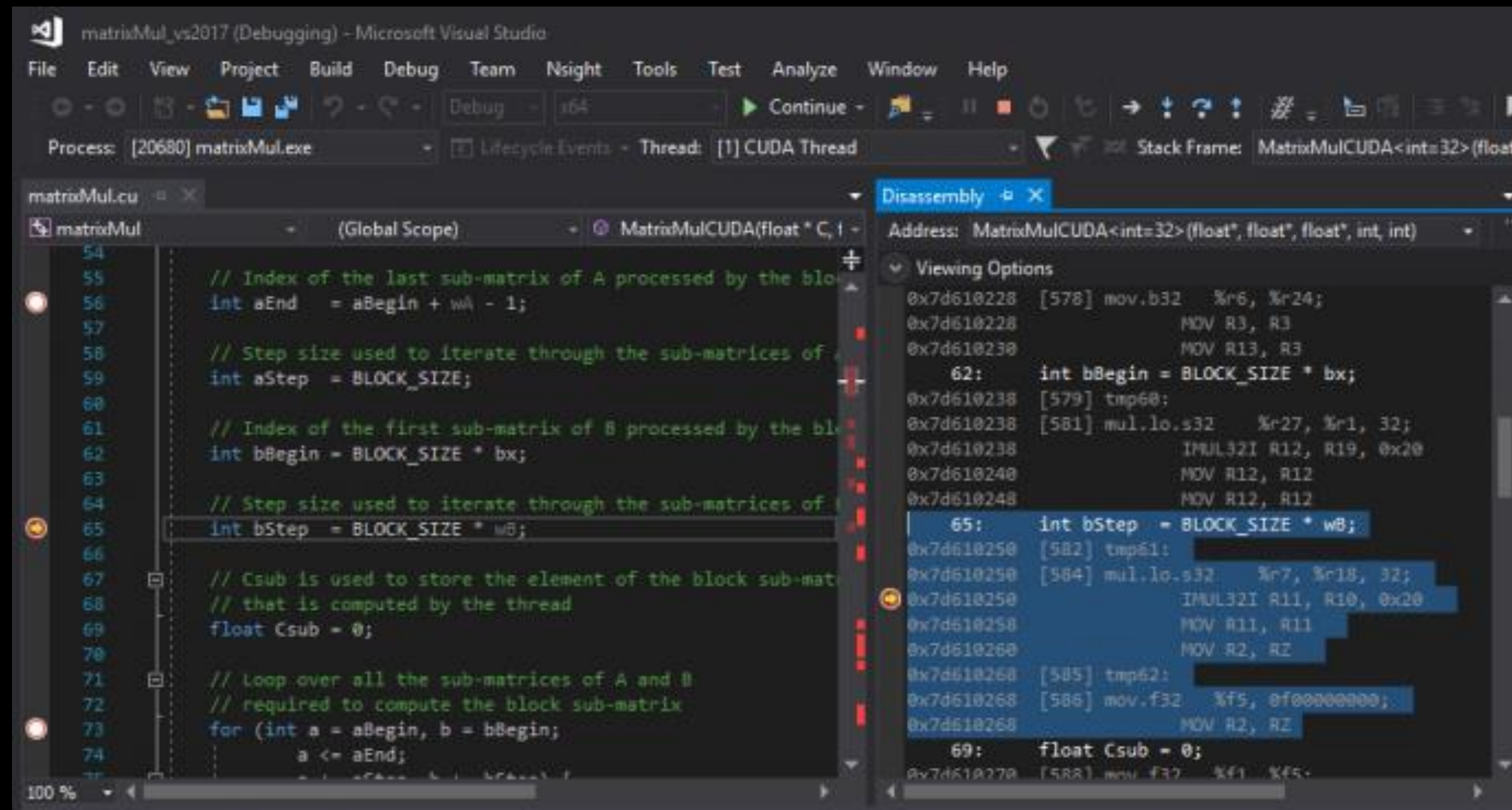




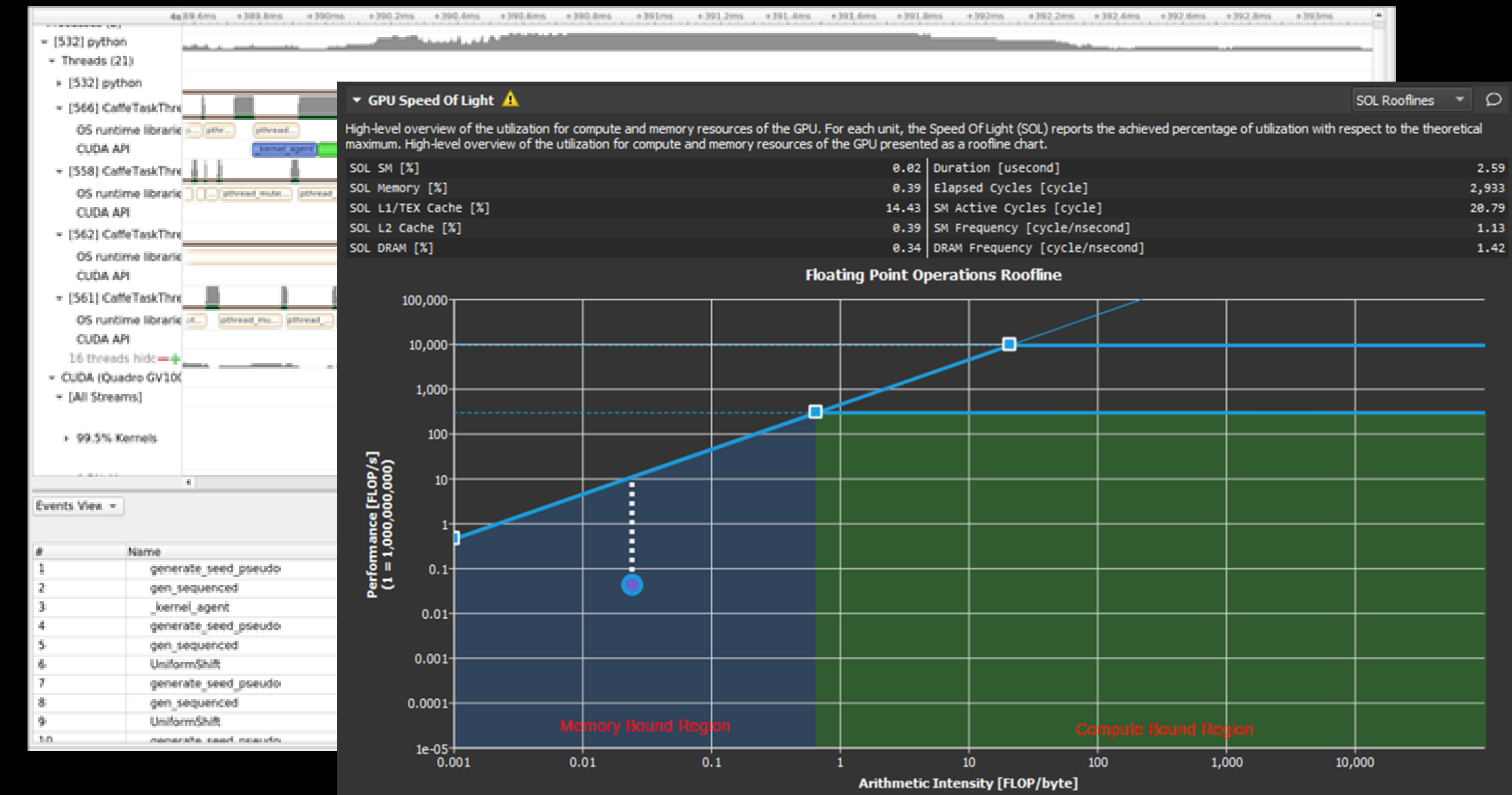
Resources

Developer tools

Debuggers: cuda-gdb, Nsight Visual Studio Edition



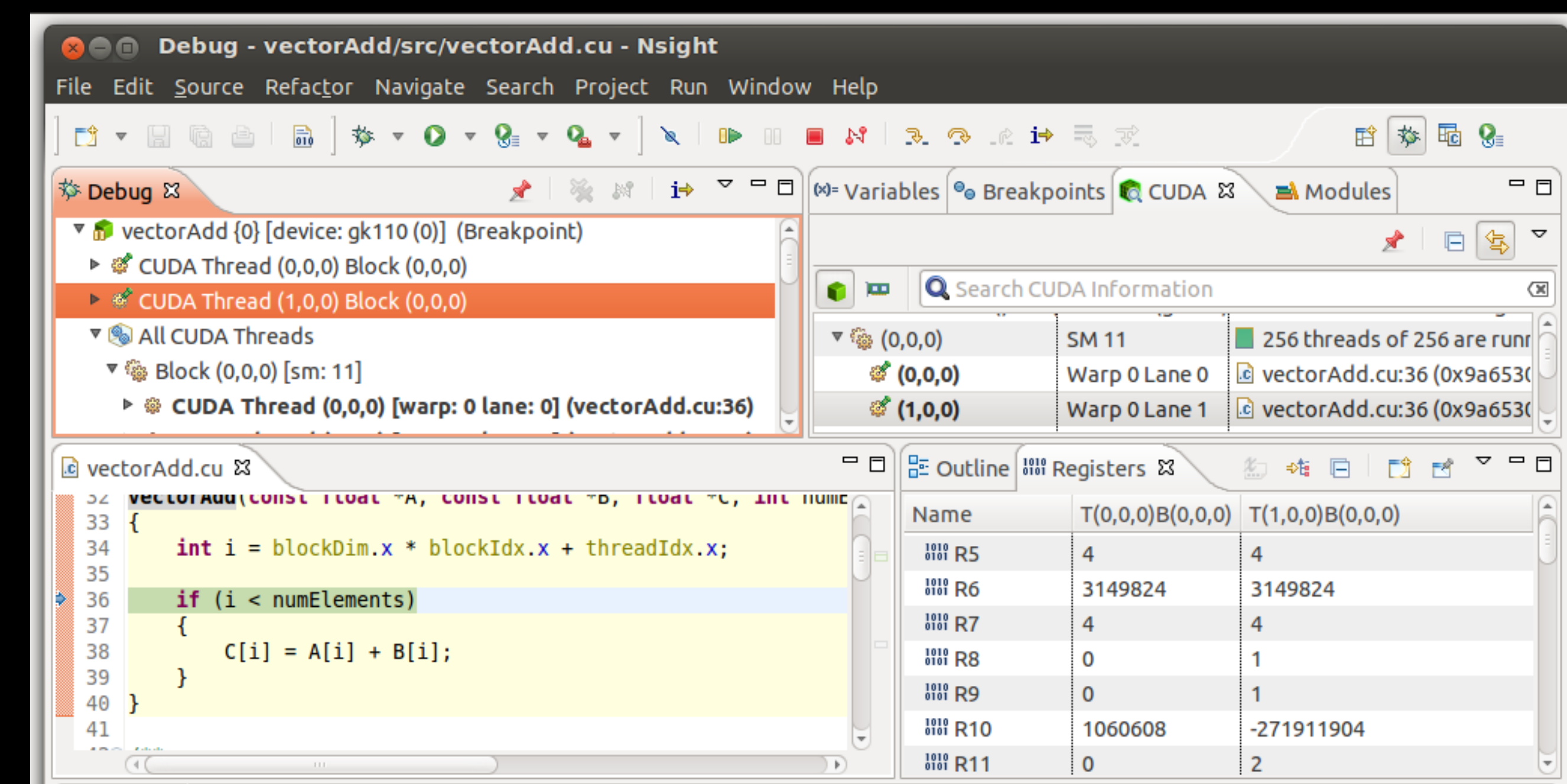
Profilers



Correctness Checker: Compute Sanitizer

```
$ compute-sanitizer --leak-check full memcheck_demo
===== COMPUTE-SANITIZER
Mallocing memory
Running unaligned_kernel
Ran unaligned_kernel: no error
Sync: no error
Running out_of_bounds_kernel
Ran out_of_bounds_kernel: no error
Sync: no error
===== Invalid __global__ write of size 4 bytes
===== at 0x60 in memcheck_demo.cu:6:unaligned_kernel(void)
===== by thread (0,0,0) in block (0,0,0)
===== Address 0x400100001 is misaligned
```

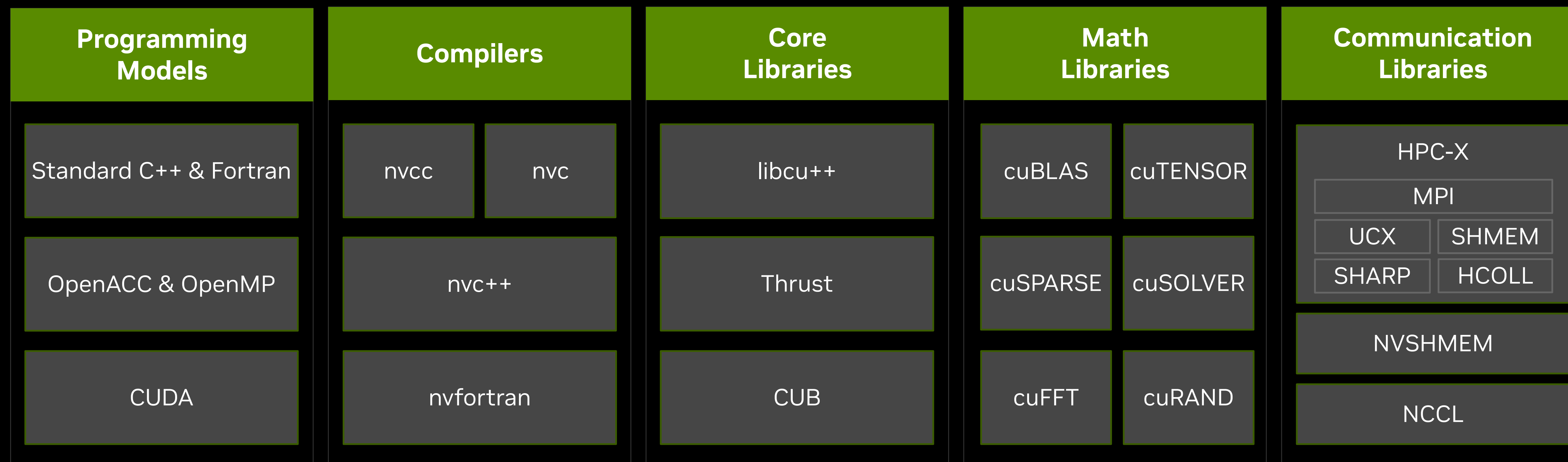
IDE integrations: Nsight Eclipse Edition
Nsight Visual Studio Edition
Nsight Visual Studio Code Edition



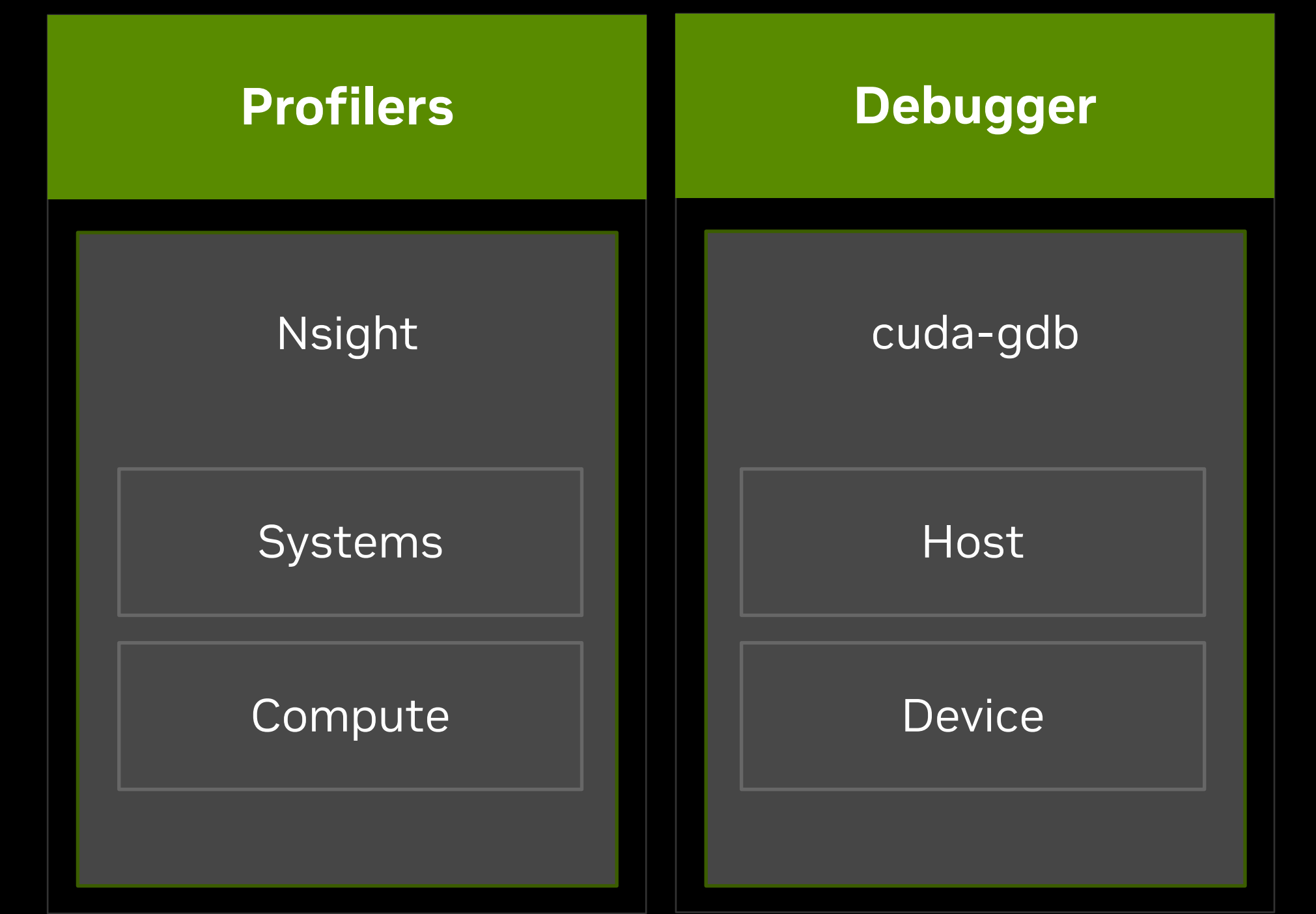
NVIDIA HPC SDK

Available at developer.nvidia.com/hpc-sdk on NGC, via Spack, and in the Cloud

DEVELOPMENT



ANALYSIS



Develop for the NVIDIA Platform: GPU, CPU and Interconnect
Libraries | Accelerated C++ and Fortran | Directives | CUDA
7-8 Releases Per Year | Freely Available

NGC: GPU-OPTIMIZED SOFTWARE HUB

Simplifying DL, ML and HPC Workflows

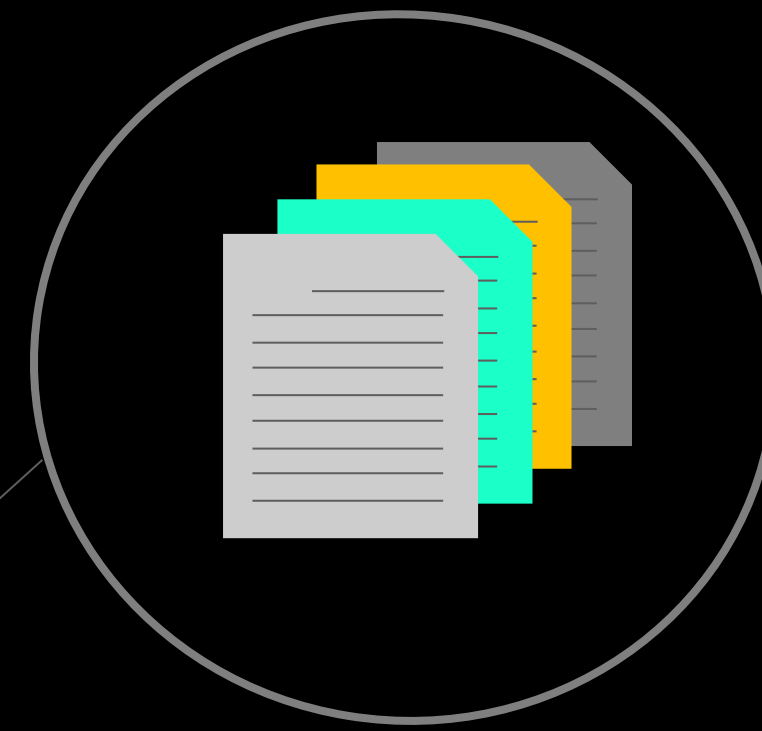
50+ Containers

DL, ML, HPC

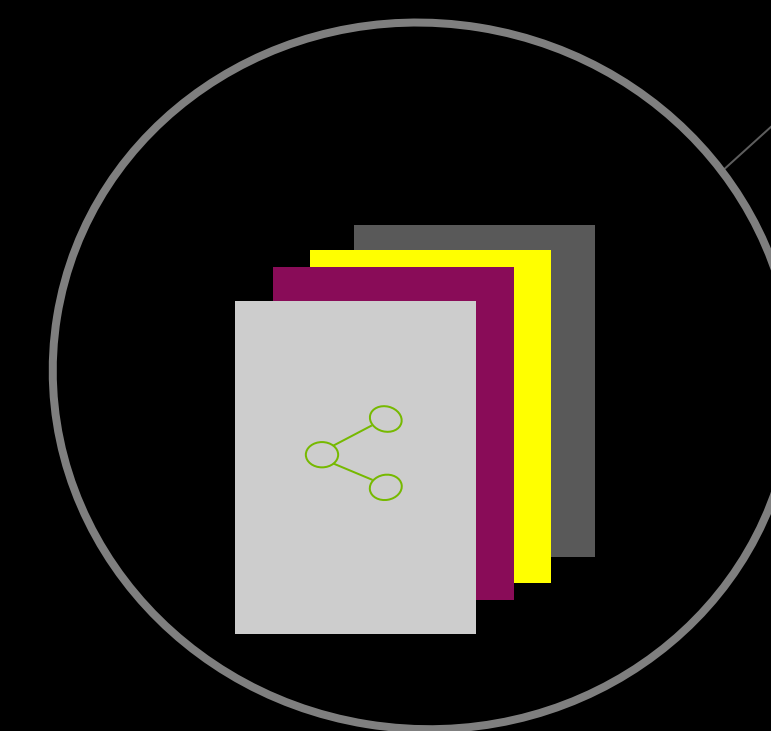


Model Training Scripts

NLP, Image Classification, Object Detection and more



NGC



60 Pre-trained Models

NLP, Image Classification, Object Detection and more



Workflows

Medical Imaging, Intelligent Video Analytics



DEEP LEARNING

TensorFlow | PyTorch | more



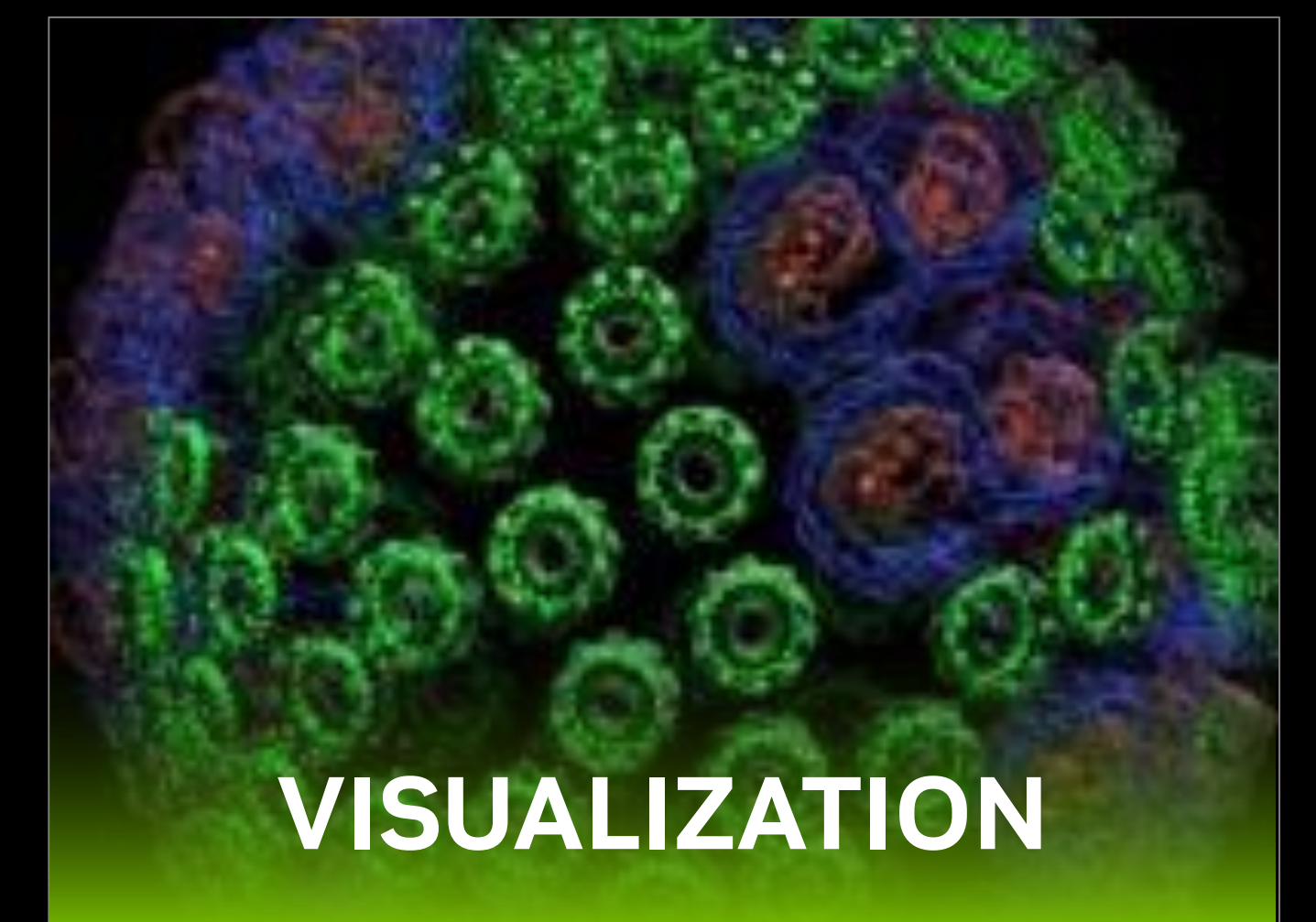
MACHINE LEARNING

RAPIDS | H2O | more



HPC

NAMD | GROMACS | more

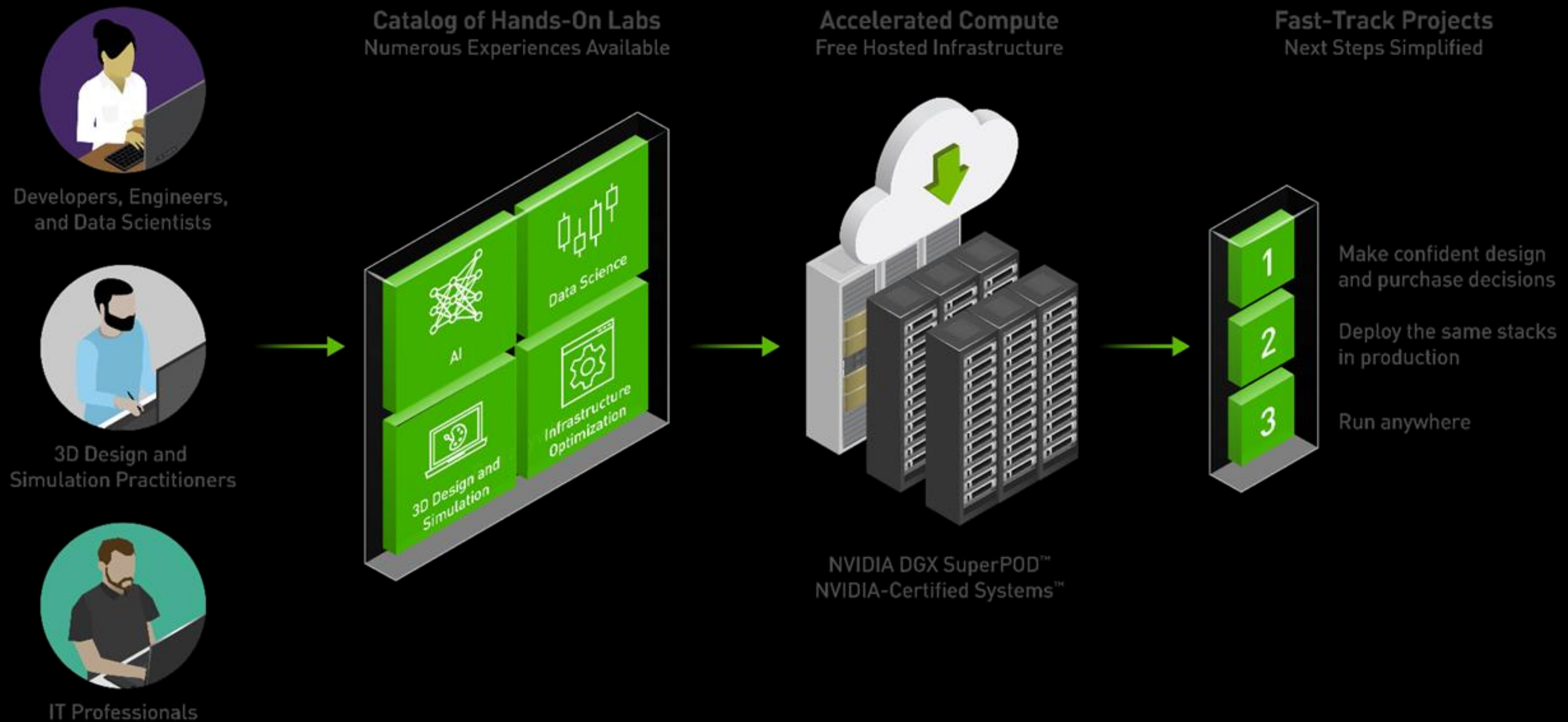


VISUALIZATION

ParaView | IndeX | more

NVIDIA LAUNCHPAD

Instantly experience end-to-end workflows for AI, data science, 3D design collaboration, and more



Get Started at nvidia.com/launchpad

DEVELOPER ENGAGEMENT PLATFORMS

Information, tutorials, news, downloads, special programs, code samples, and bug submission	developer.nvidia.com
Forums: Insights & help from other developers and NVIDIA technical staff	forums.developer.nvidia.com
Technical documentation	docs.nvidia.com
In depth technical how-to blogs	devblogs.nvidia.com
Webinars	nvidia.com/webinar-portal
GTC on-demand content	nvidia.com/on-demand

Bootcamps

~2 day tailored training events or workshops, typically for a target group e.g. N-ways to GPU, AI for Science

Hackathons:

Bring-your-code events, in-depth porting/optimization with access to expert mentors

DEEP LEARNING INSTITUTE (DLI)

Hands-on, self-paced and instructor-led training in deep learning and accelerated computing

Request onsite instructor-led workshops at your organization:

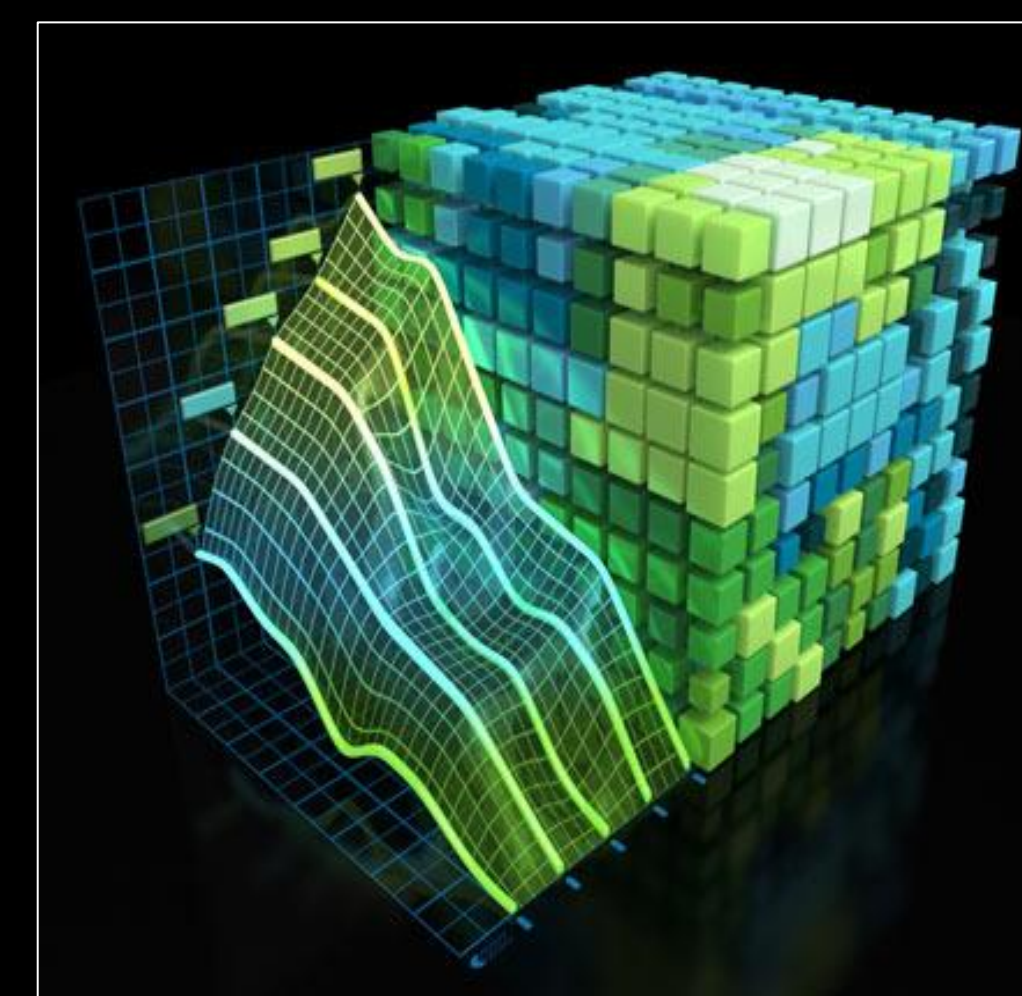
www.nvidia.com/requestdli

Take self-paced courses online:

www.nvidia.com/dlilabs

Download the course catalog, view upcoming workshops, and learn about the University Ambassador Program:

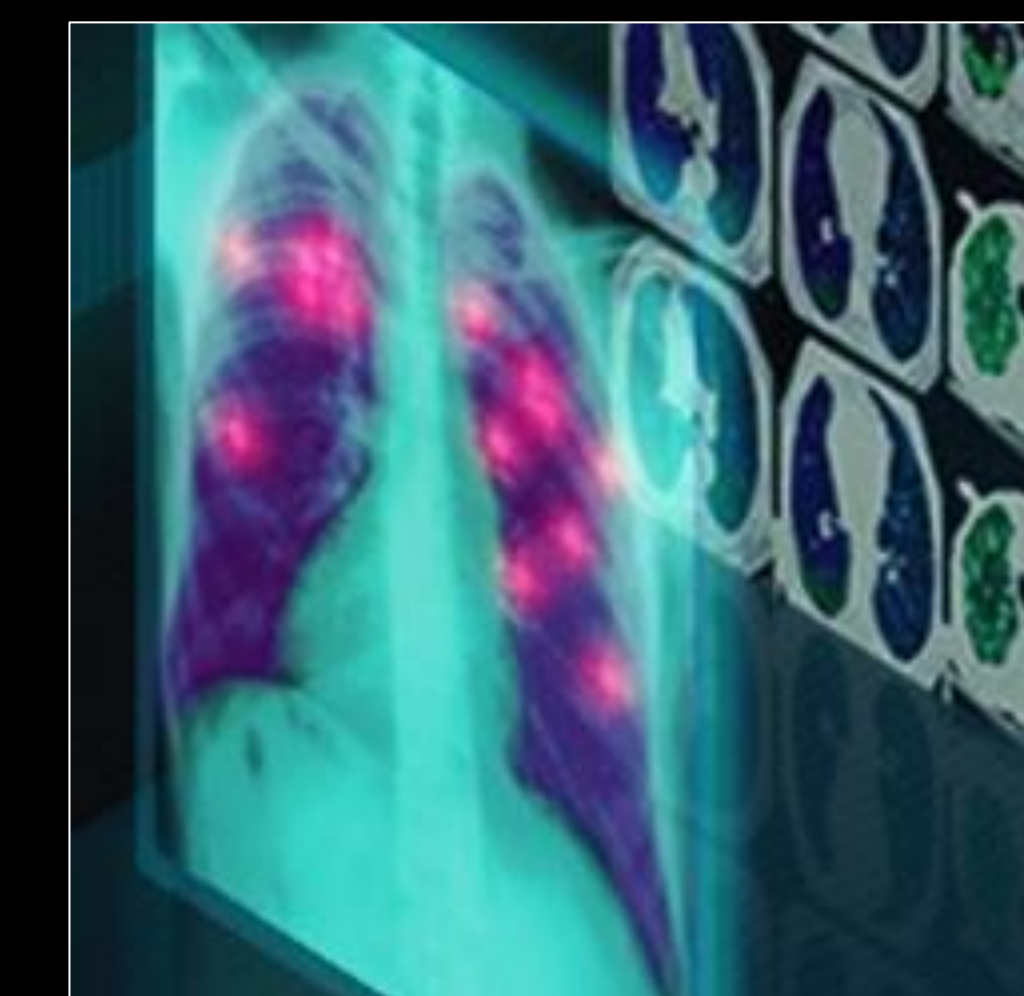
www.nvidia.com/dli



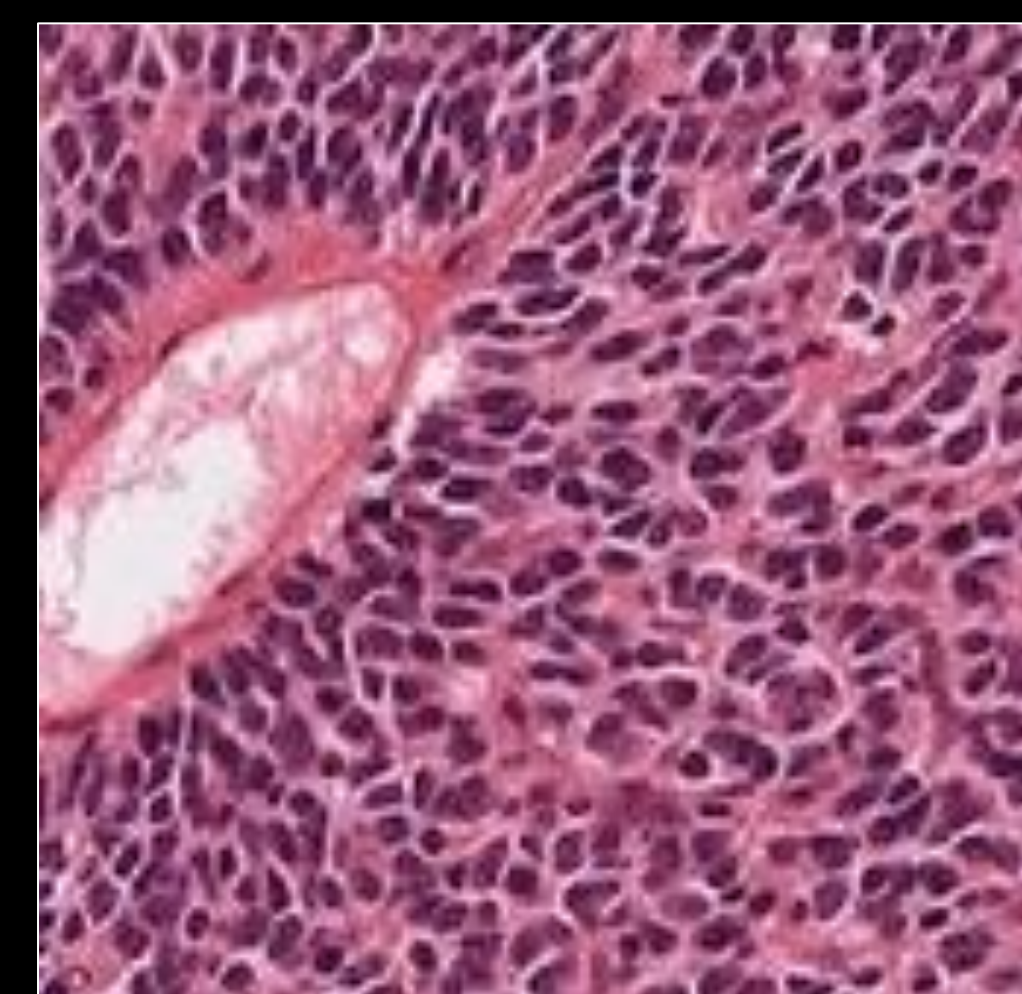
Accelerated Computing Fundamentals



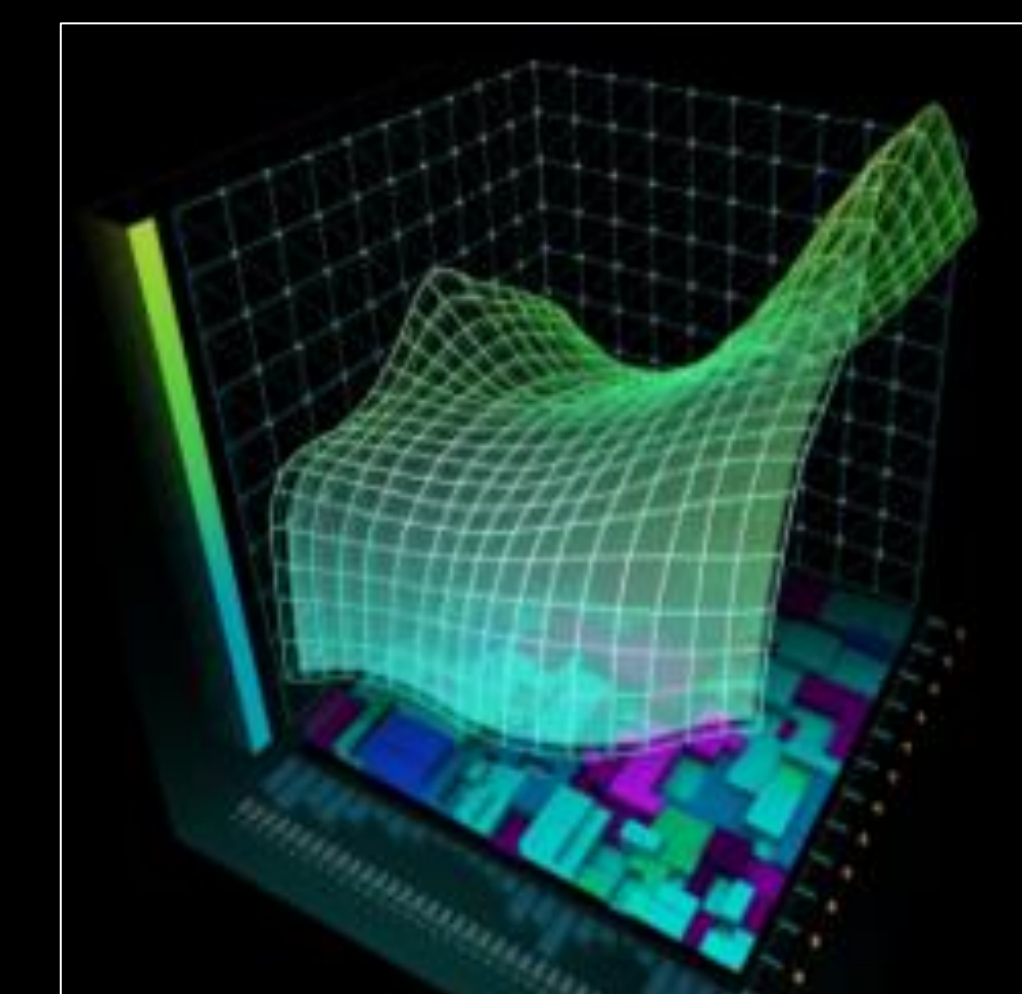
Autonomous Vehicles



Medical Image Analysis



Genomics



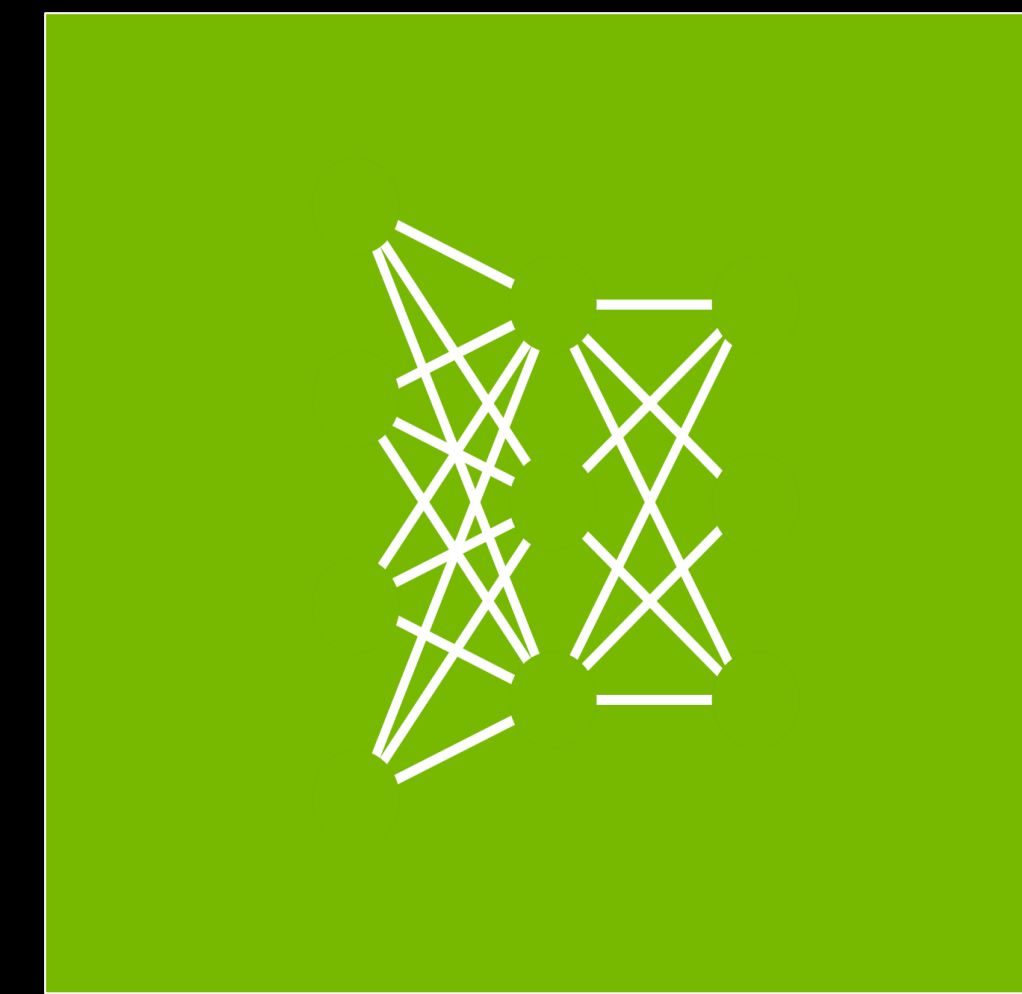
Finance



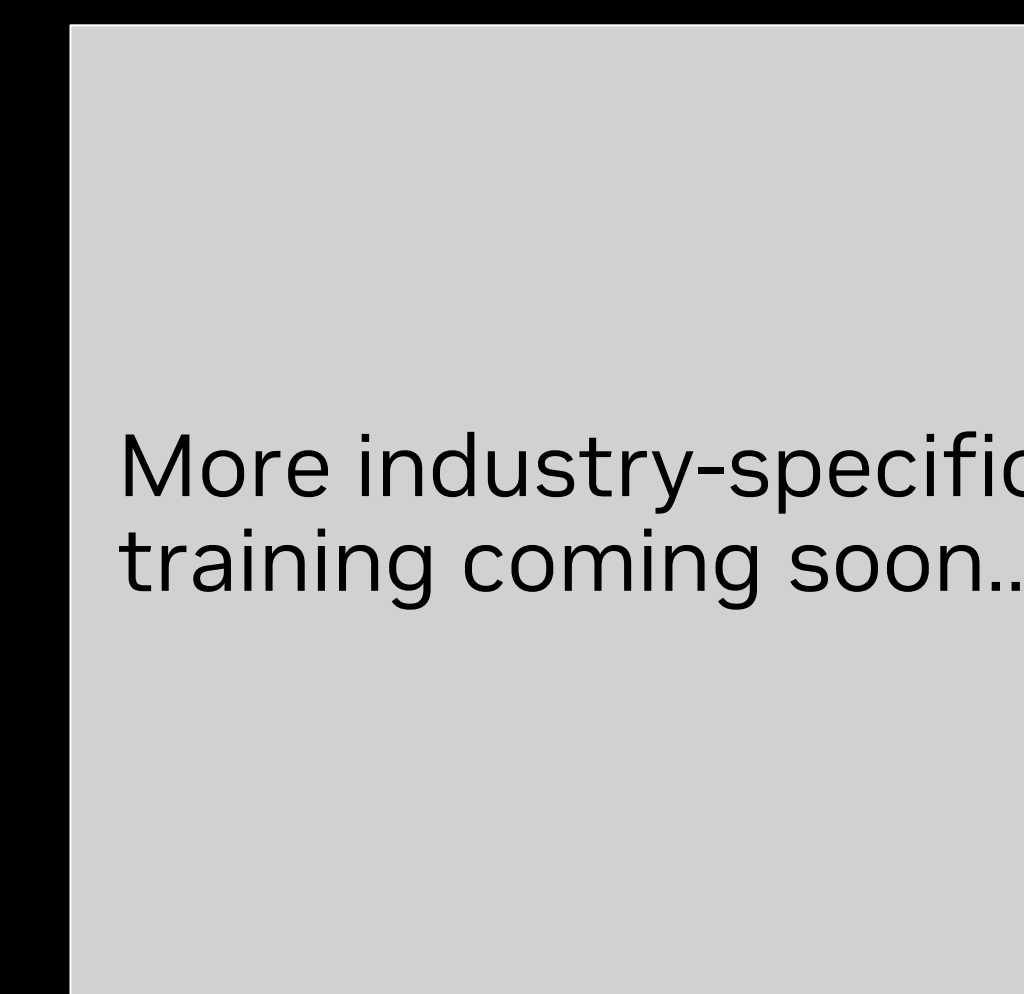
Content



Game Development



Deep Learning Fundamentals



More industry-specific training coming soon...

Agenda

- 10:15-10:45 Introduction to Heterogeneous Parallel Computing
- 10:45-11:00 Break
- 11:00-12:00 Key ways to accelerate applications
- 12:00-13:00 Programming for GPUs
- 13:00-13:45 Lunch
- 13:45-17:15 Hands-on practical