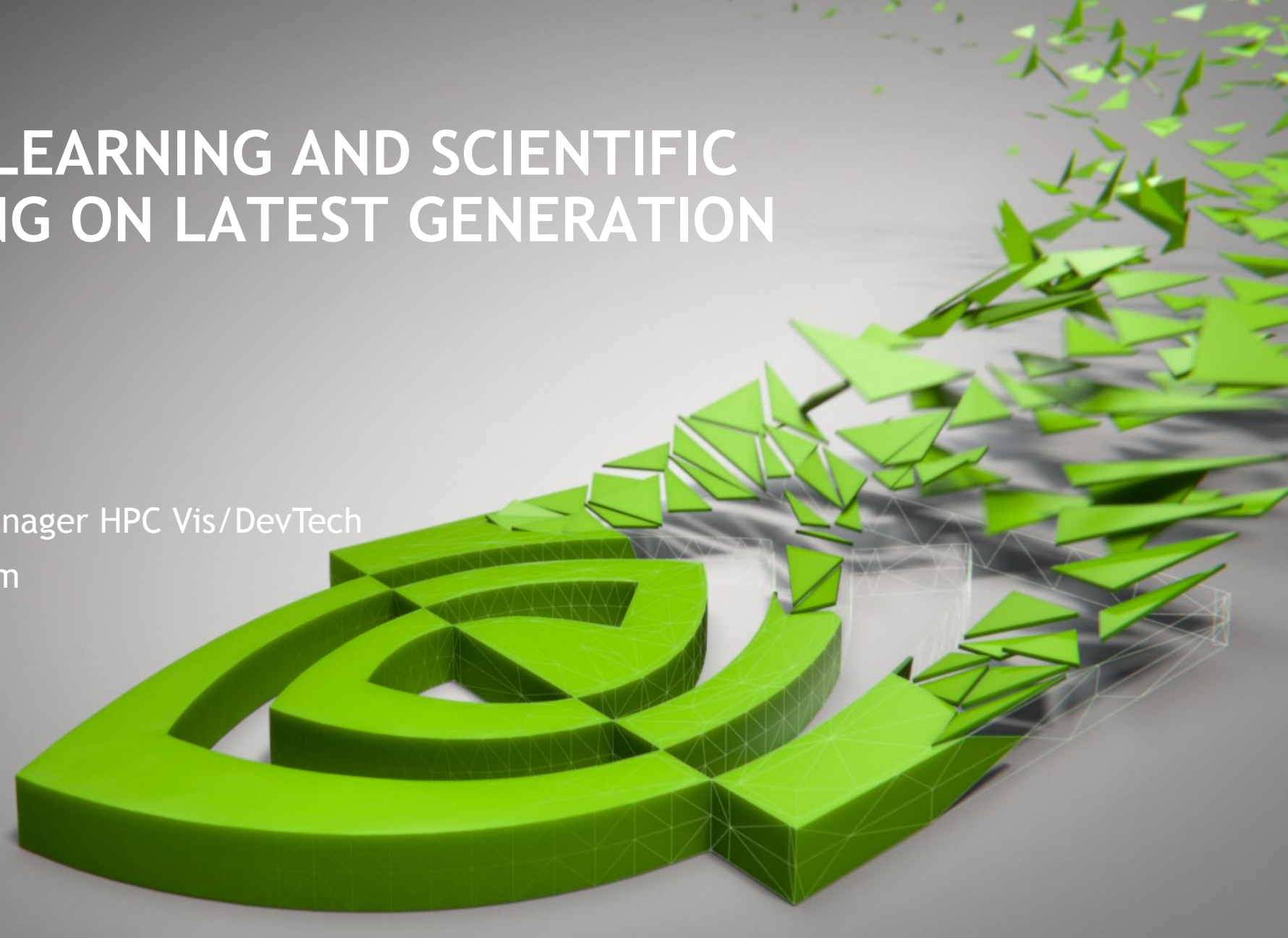

NVIDIA

MACHINE LEARNING AND SCIENTIFIC COMPUTING ON LATEST GENERATION GPUS

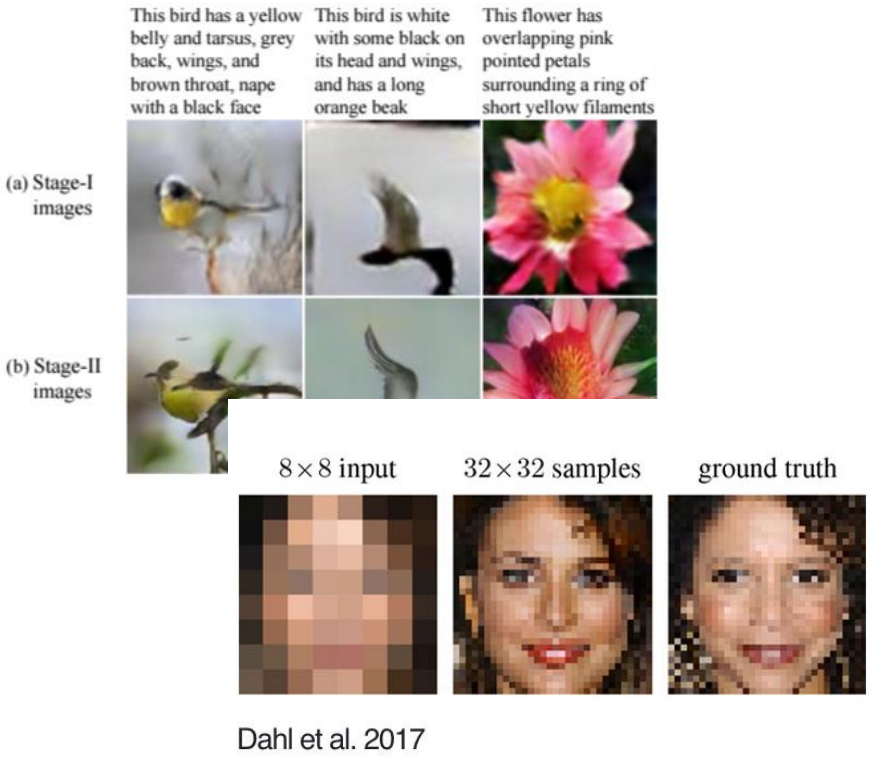
Peter Messmer, Sr. Manager HPC Vis/DevTech

pmessmer@nvidia.com



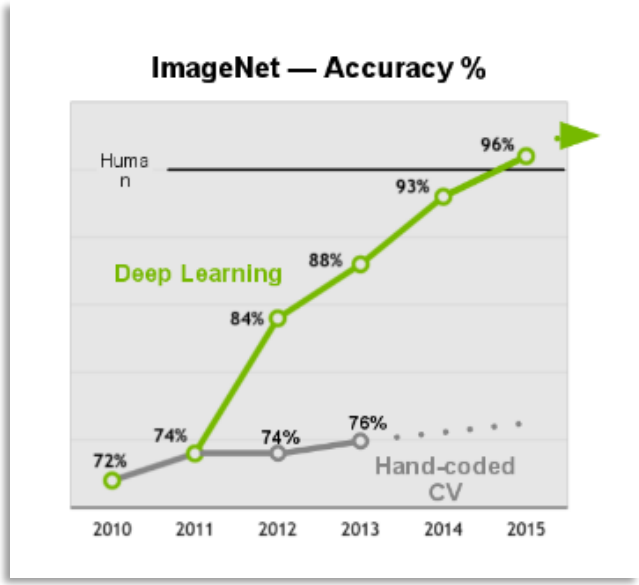
WHY THE EXCITEMENT?

GPUs as Enablers of Breakthrough Results

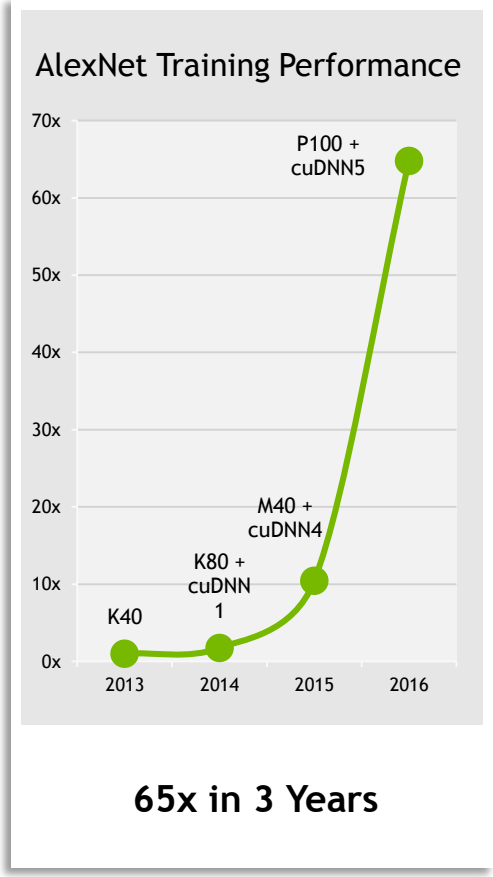


Dahl et al. 2017

We can generate photorealistic images from textual descriptions and super-enhance blurry photos!



Achieve super-human accuracy in classification

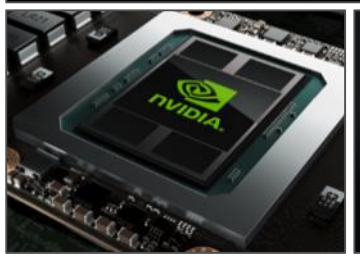


And we are getting faster fast

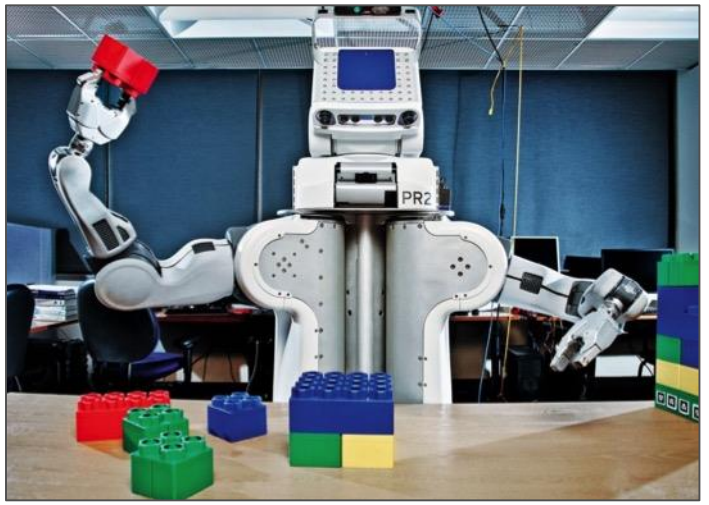
Paper: H.Zhang et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, arXiv:1612.03242

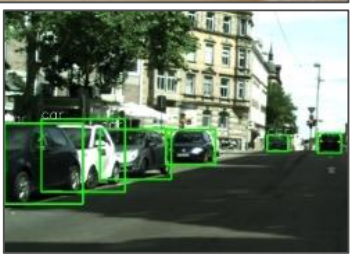
NVIDIA - AI COMPUTING COMPANY

Computer Graphics



GPU Computing





Artificial Intelligence

TESLA PLATFORM

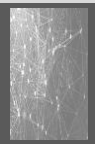
Leading Data Center Platform for Accelerating HPC and AI

APPLICATIONS

Alibaba amazon Baidu 百度 ebay facebook flickr Google Microsoft Pinterest skype twitter yelp

INTERNET SERVICES





Healthcare Manufacturing Finance

Automotive
Retail
Defense
...

ENTERPRISE APPLICATIONS

ANSYS SIMULIA GROMACS FAST. FLEXIBLE. FREE. VASP +450 Applications

HPC

INDUSTRY FRAMEWORKS & TOOLS

Caffe2 Microsoft Cognitive Toolkit mxnet PYTORCH TensorFlow theano

FRAMEWORKS

allinea ArrayFire CONTINUUM ANALYTICS WOLFRAM ROGUE WAVE MathWorks

ECOSYSTEM TOOLS

NVIDIA SDK

cuDNN TensorRT NCCL cuBLAS cuSPARSE DeepStream SDK

DEEP LEARNING SDK

CUDA C/C++ PGI OpenACC Directives for Accelerators

COMPUTEWORKS

TESLA GPU & SYSTEMS



TESLA GPU NVIDIA DGX-1 NVIDIA HGX-1

DELL Hewlett Packard Enterprise IBM

SYSTEM OEM

amazon web services Google Cloud Platform Microsoft Azure

CLOUD

AGENDA

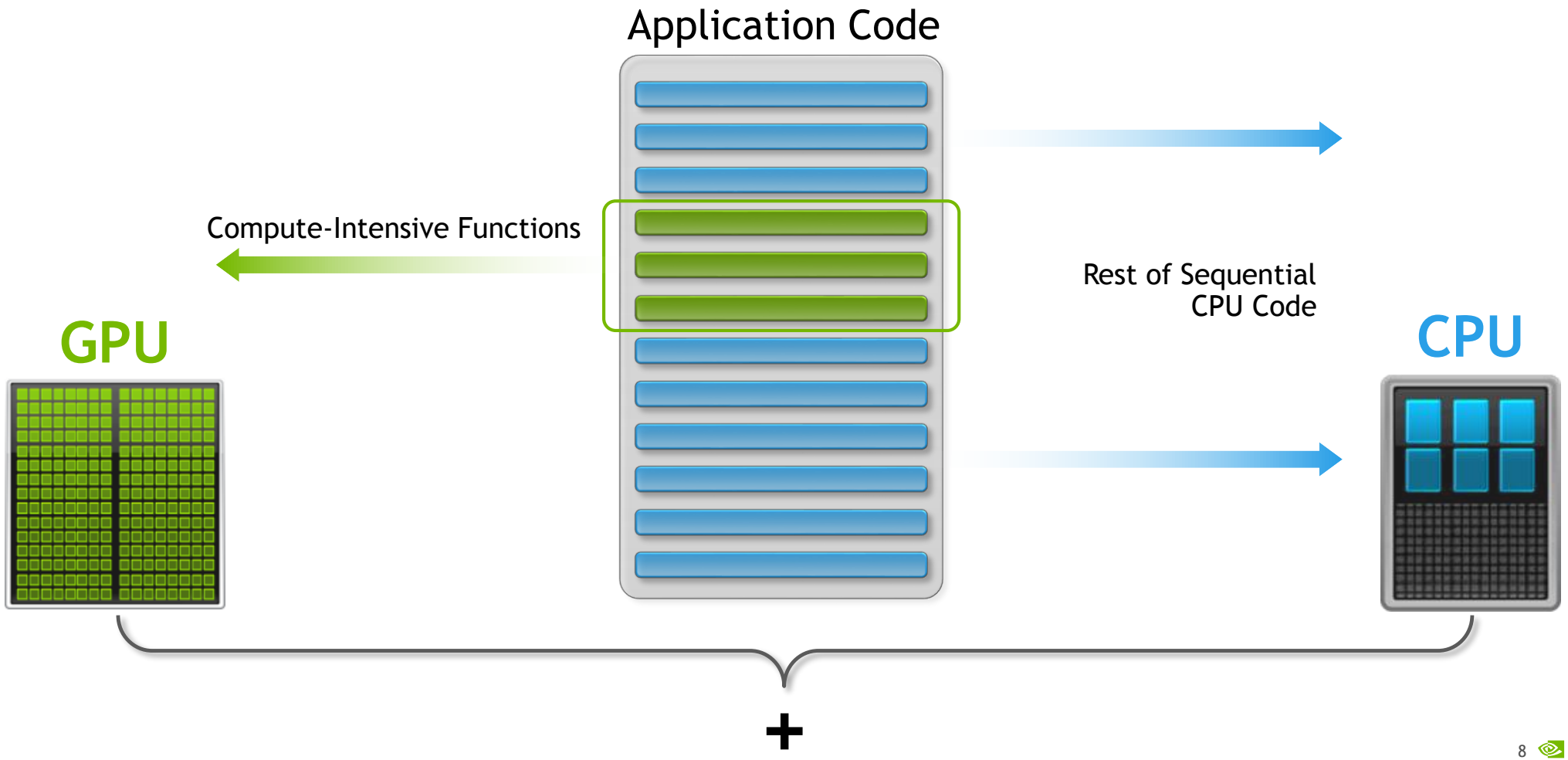
Latest Generation GPU

Quick intro to Neural Networks and Inference

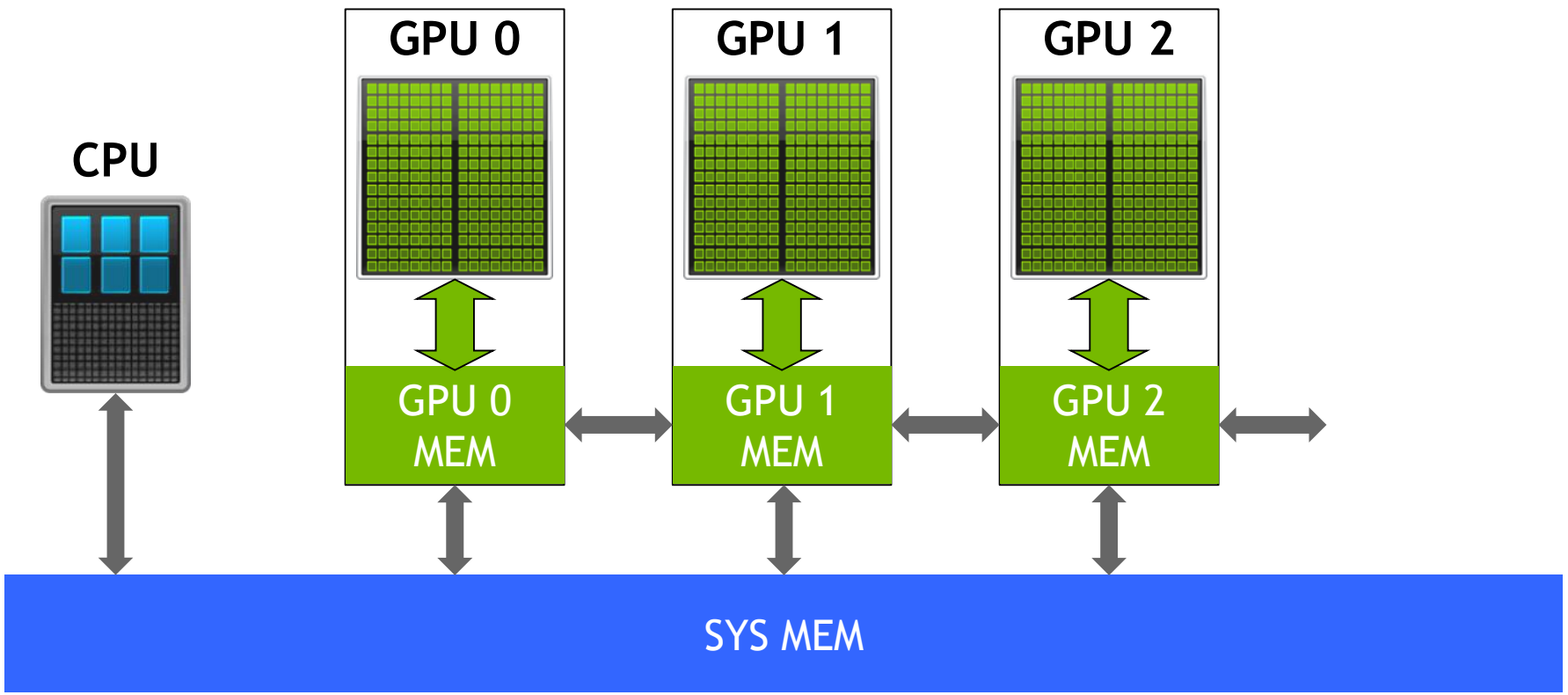
HPC + DL

LATEST GENERATION GPUS

HOW GPU ACCELERATION WORKS



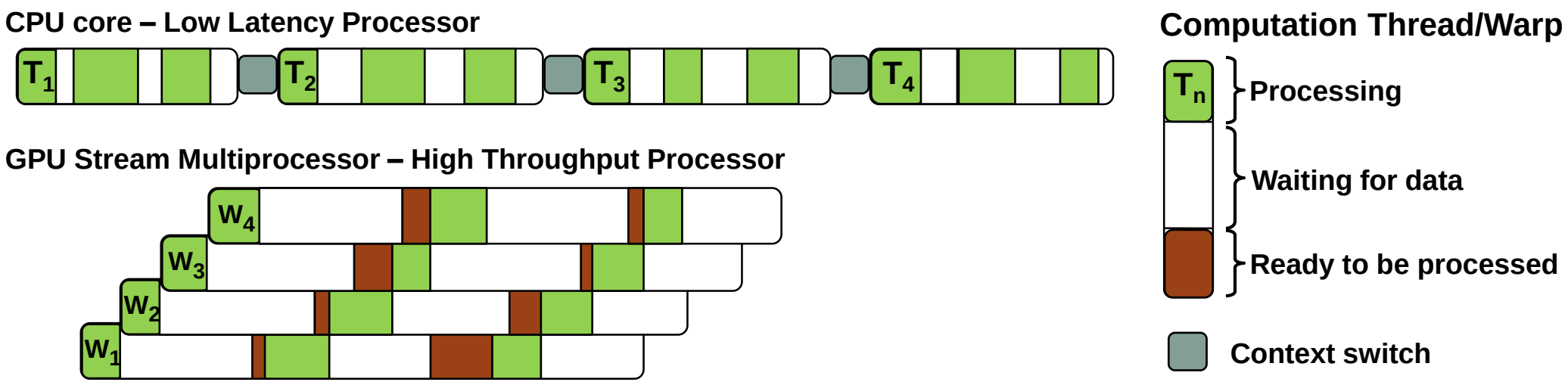
HETEROGENEOUS ARCHITECTURES



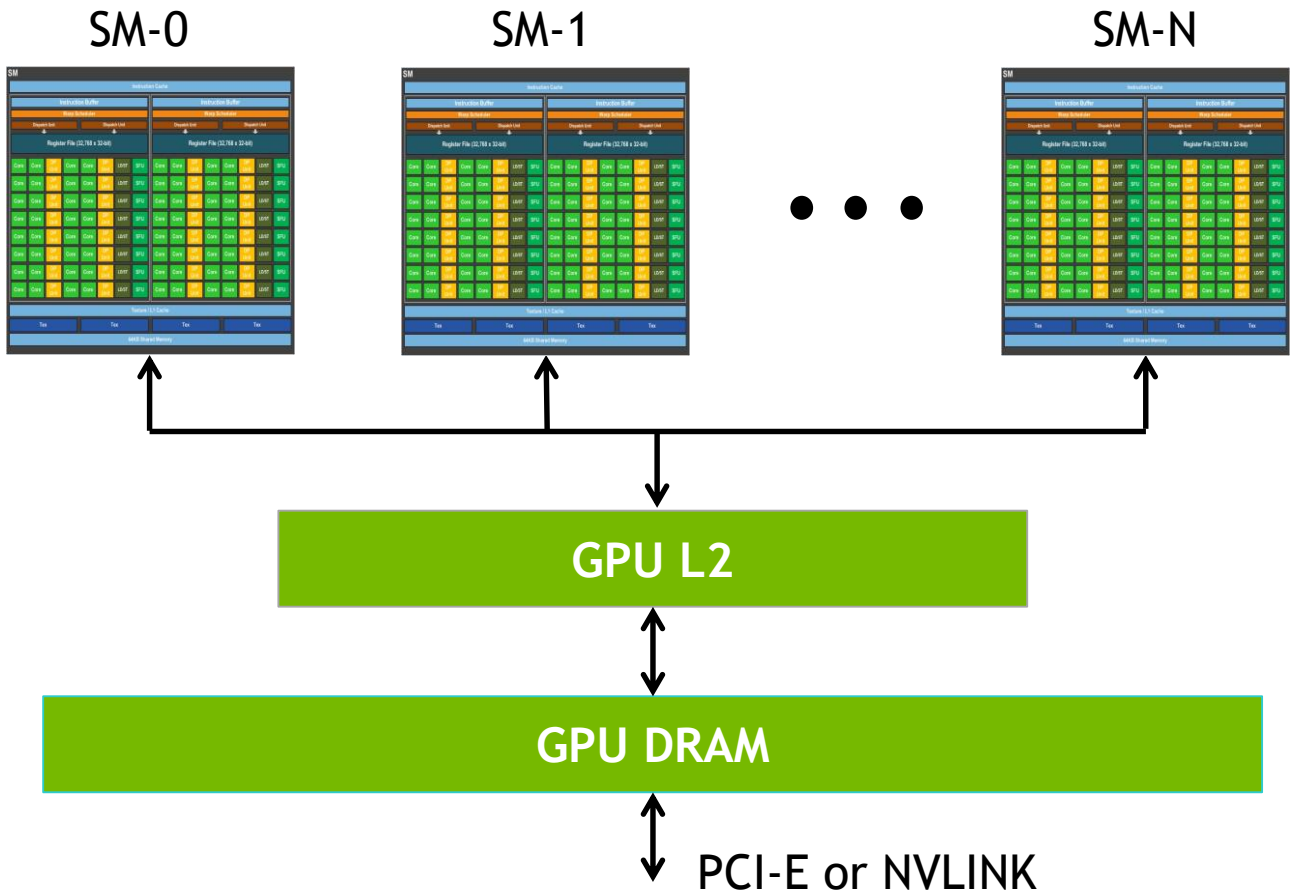
LOW LATENCY OF HIGH THROUGHPUT?

CPU architecture must **minimize latency** within each thread

GPU architecture **hides latency** with computation from other threads (warps)



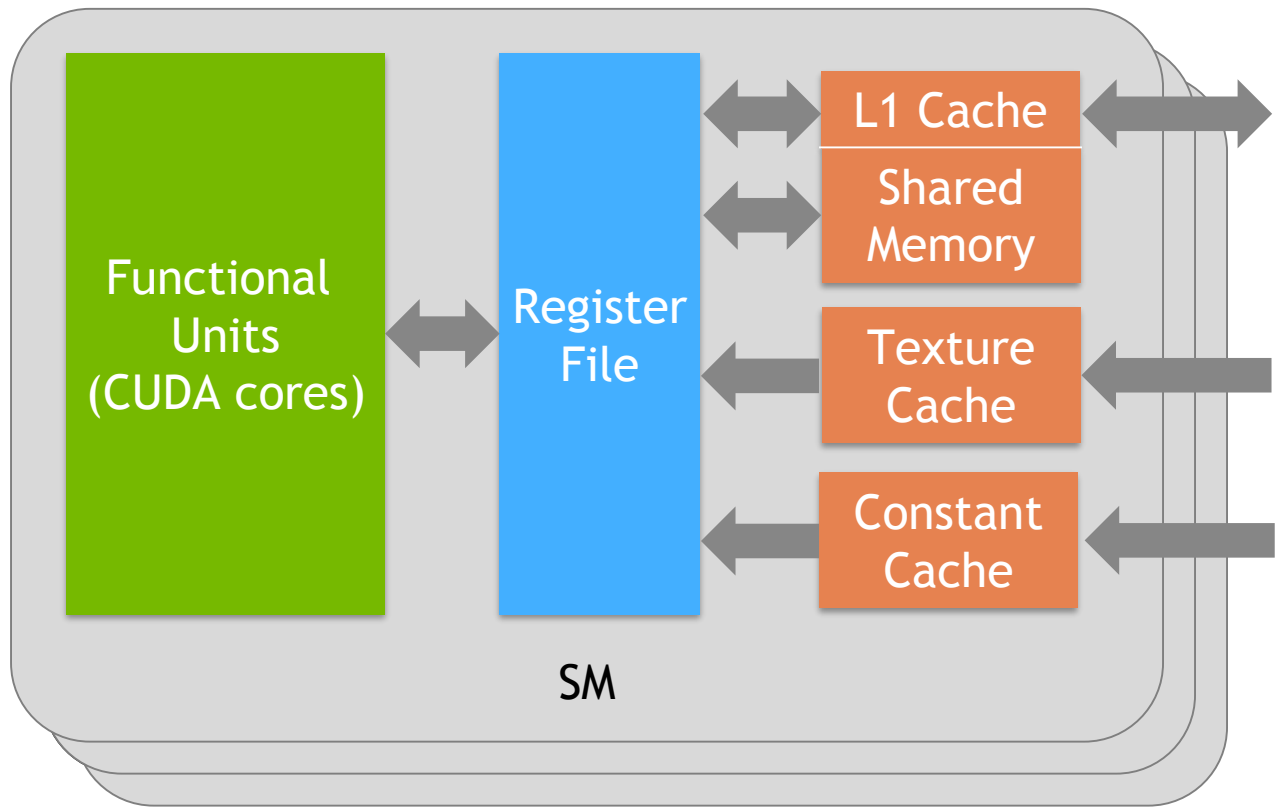
GPU ARCHITECTURE



GPU SM ARCHITECTURE

Kepler SM

	GK110
FP32 Cores	192
FP64 Cores	64
Register File	256 KB
Shared Memory	16/32/48 KB

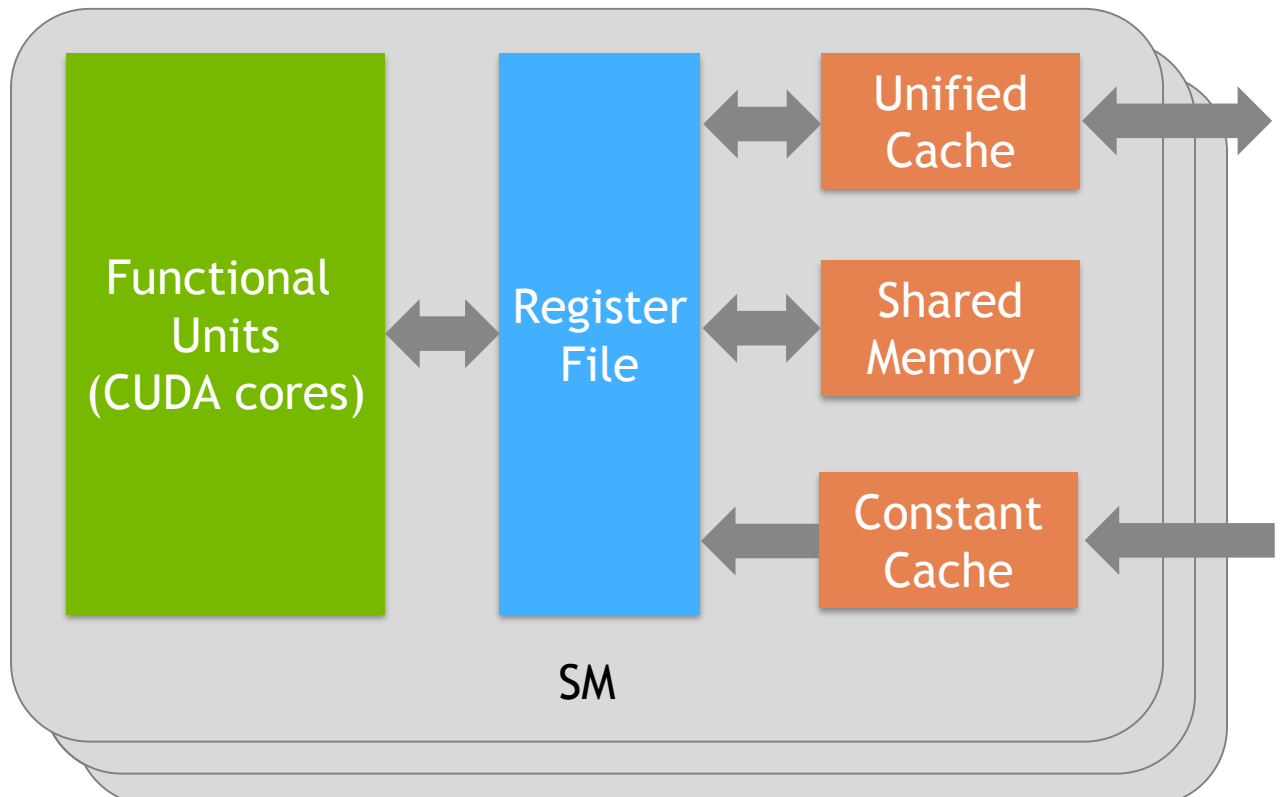


15 SMs on Tesla K40

GPU SM ARCHITECTURE

Pascal SM

	GP100
FP32 Cores	64
FP64 Cores	32
Register File	256 KB
Shared Memory	64 KB

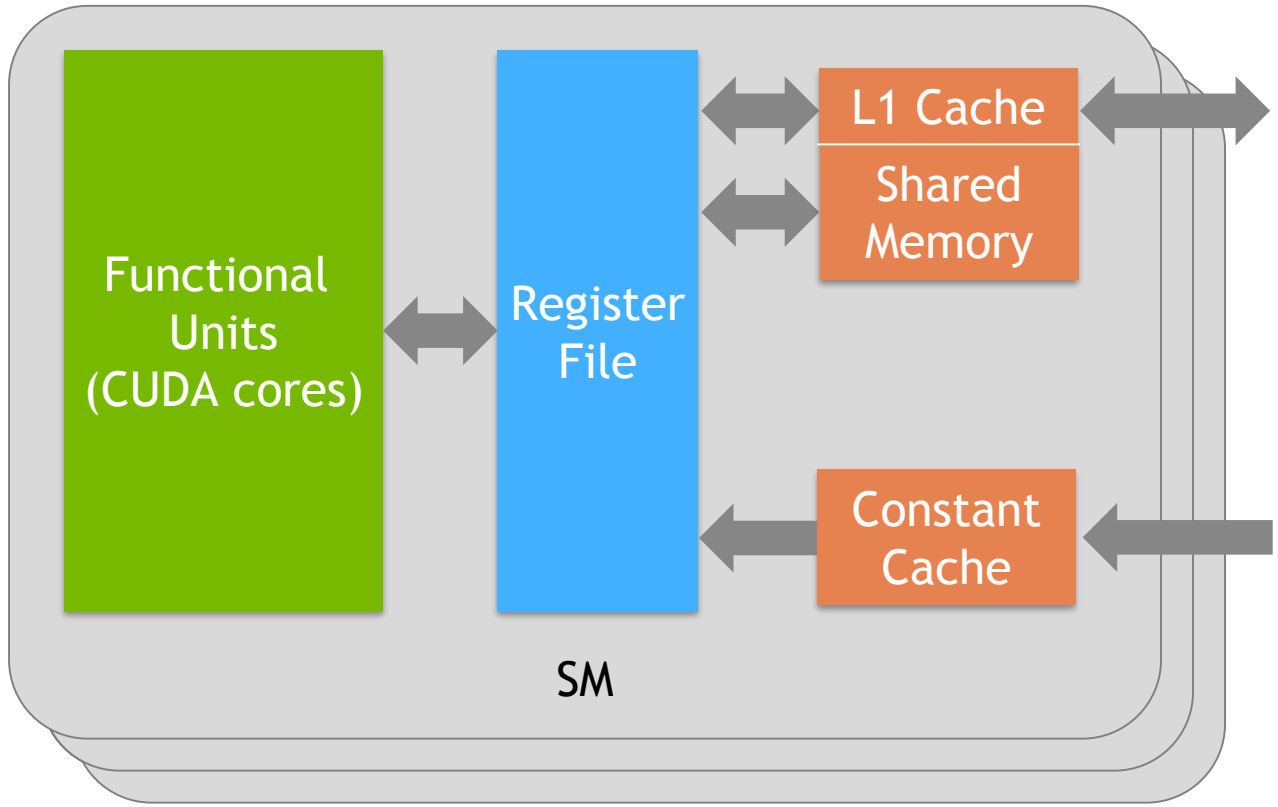


56 SMs on Tesla P100

GPU SM ARCHITECTURE

Volta SM

	GV100
FP32 Cores	64
FP64 Cores	32
Tensor Cores	8
Register File	256 KB
Shared Memory	up to 96 KB



80 SMs on Tesla V100

TESLA FAMILY

GPU comparison (boost clocks)

	Tesla K40	Tesla P100	Tesla V100
Peak FP32 (TFLOP/s)	5.04	10.6	15
Peak FP64 (TFLOP/s)	1.68	5.3	7.5
Peak Tensor Core (TFLOP/s)	N/A	N/A	120
Memory Size (GB)	12	16	16
Memory Bandwidth (GB/s)	288	732	900

NVIDIA TESLA V100

- 21B transistors
  815 mm^2, 12nm FFN
- 80 SM
  5120 CUDA Cores
  640 Tensor Cores
- 7.8 FP64 TFLOPS
- 15.6 FP32 TFLOPS
- 125 Tensor TFLOPS
- 16 GB HBM2
  900 GB/s memory bandwidth
- 300 GB/s NVLink bandwidth

*full GV100 chip contains 84 SMs

TENSOR CORE

Mixed Precision Matrix Math - 4x4 matrices

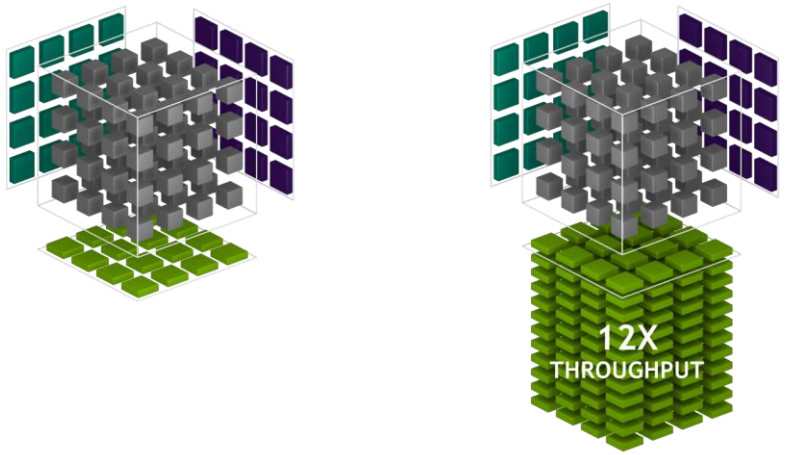
New CUDA TensorOp instructions
& data formats

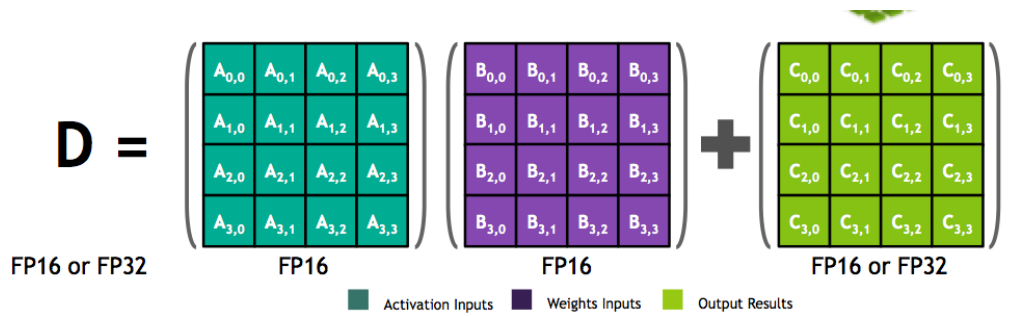
4x4 matrix processing array

D[FP32] = A[FP16] * B[FP16] + C[FP32]

Using Tensor cores via

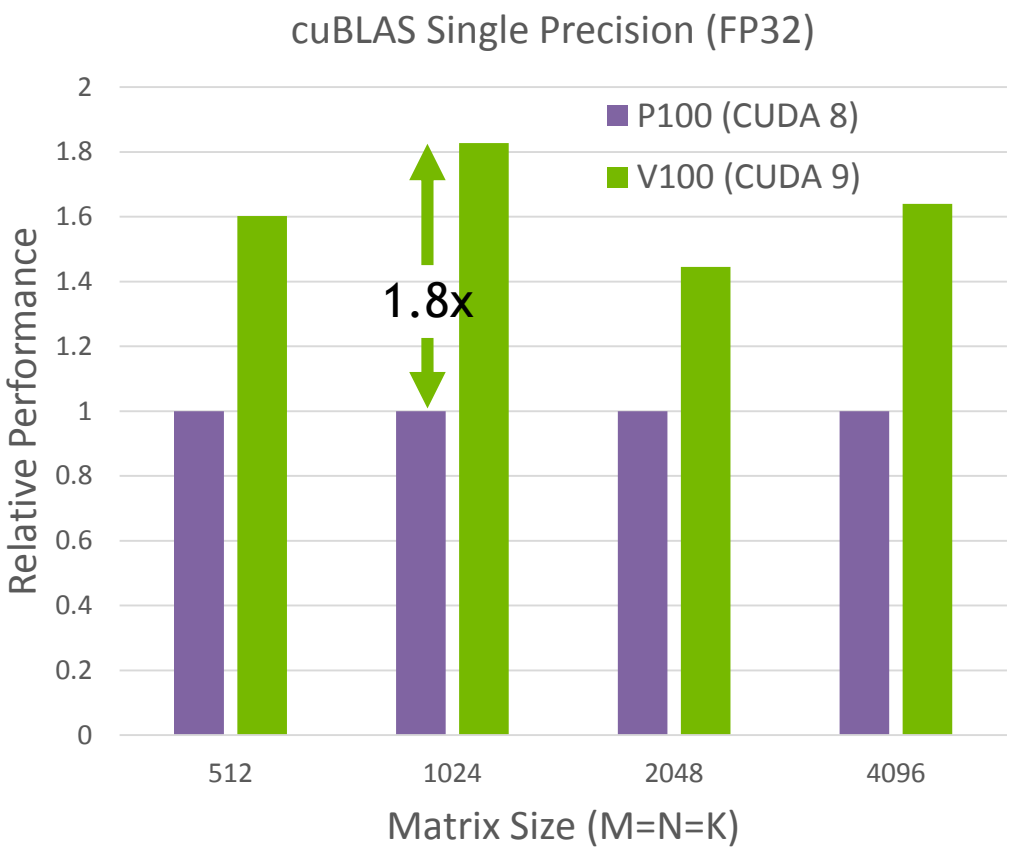
- Volta optimized frameworks and libraries (cuDNN, CuBLAS, TensorRT, ..)
- CUDA C++ Warp Level Matrix Operations

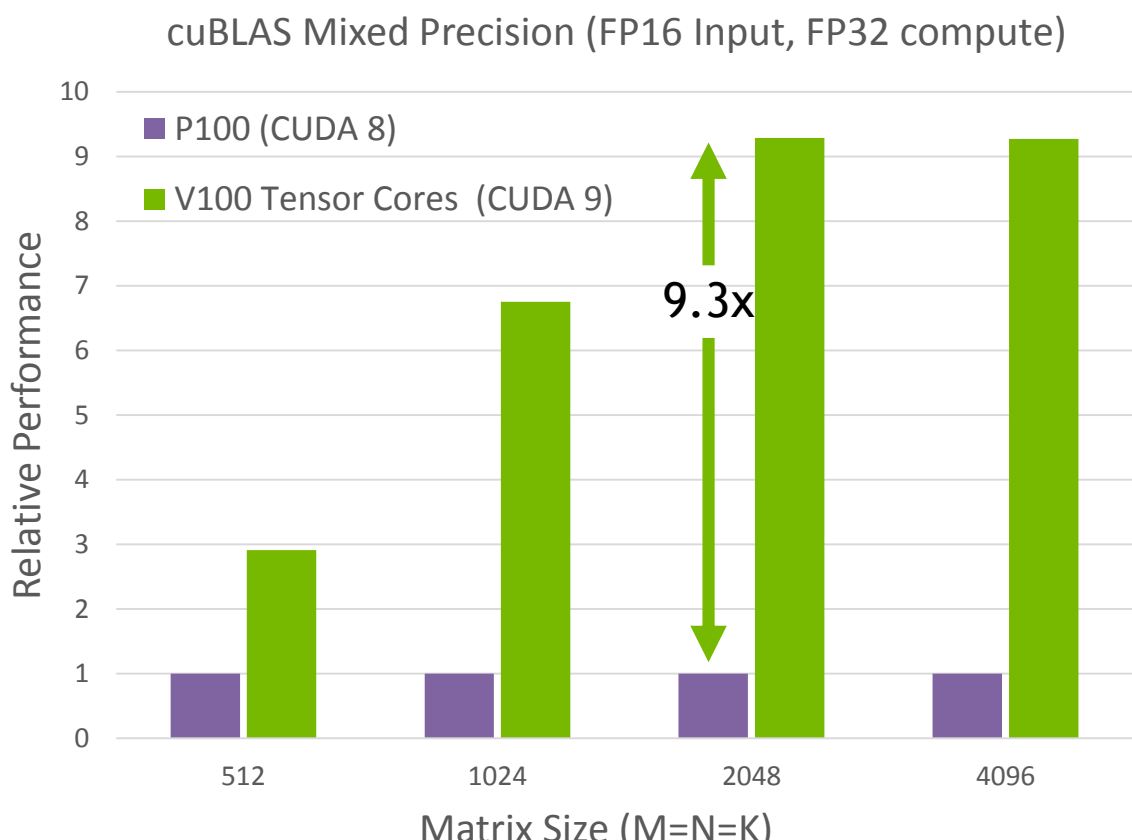




cuBLAS GEMMS FOR DEEP LEARNING

V100 Tensor Cores + CUDA 9: over 9x Faster Matrix-Matrix Multiply

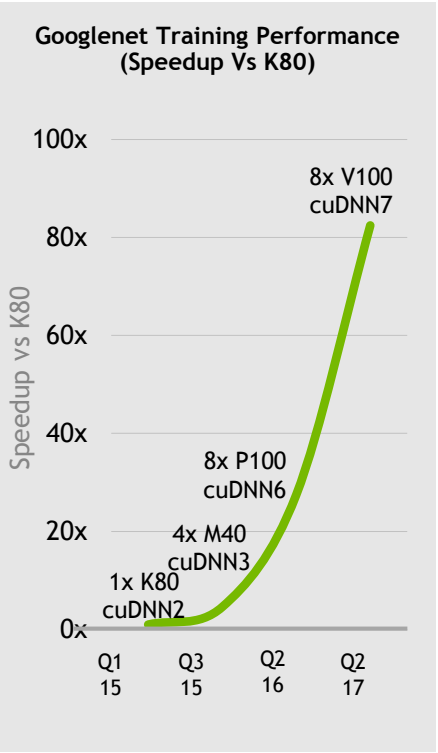




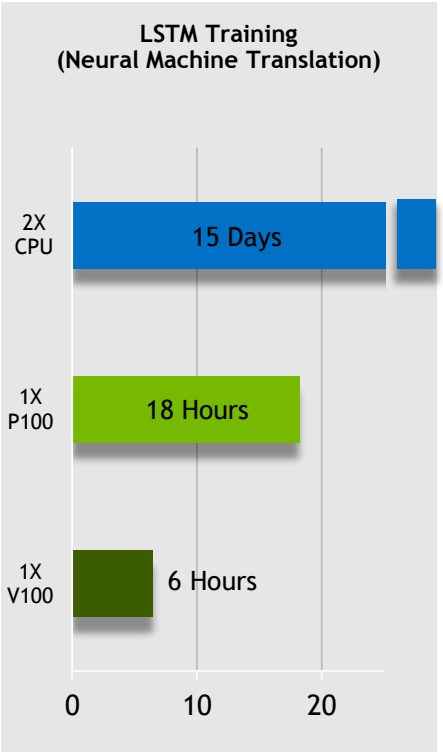
Note: pre-production Tesla V100 and pre-release CUDA 9. CUDA 8 GA release.

AI PERFORMANCE ON VOLTA

3X Faster DL Training Performance

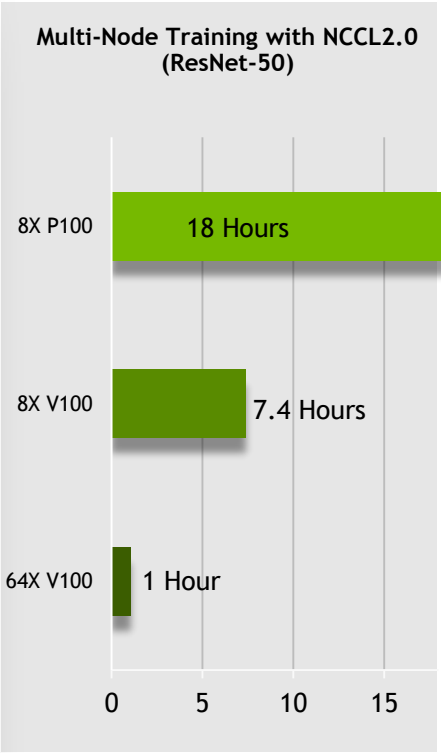


Over 80x DL Training Performance in 3 Years



3X Reduction in Time to Train Over P100

Neural Machine Translation Training for 13 Epochs |German ->English, WMT15 subset | CPU = 2x Xeon E5 2699 V4 | V100 performance measured on pre-production hardware.



85% Scale-Out Efficiency Scales to 64 GPUs with Microsoft Cognitive Toolkit

ResNet50 Training for 90 Epochs with 1.28M images dataset | Cognitive Toolkit with NCCL 2.0 | V100 performance measured on pre-production hardware.

NVIDIA cuDNN 7

Deep Learning Primitives

High performance building blocks for deep learning frameworks

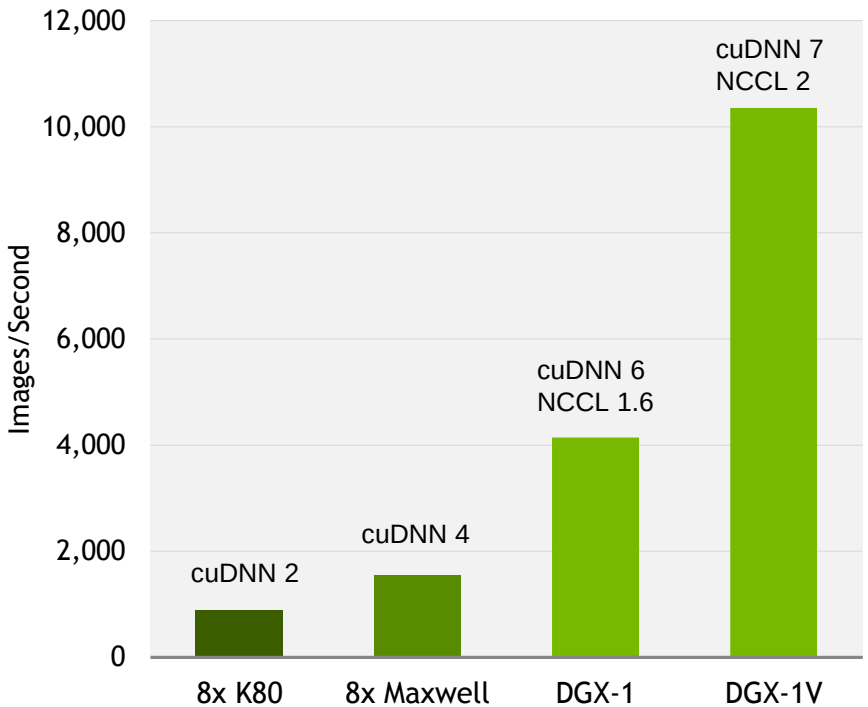
Drop-in acceleration for widely used deep learning frameworks such as Caffe2, Microsoft Cognitive Toolkit, PyTorch, Tensorflow, Theano and others

Accelerates industry vetted deep learning algorithms, such as convolutions, LSTM RNNs, fully connected, and pooling layers

Fast deep learning training performance tuned for NVIDIA GPUs

developer.nvidia.com/cudnn

Deep Learning Training Performance



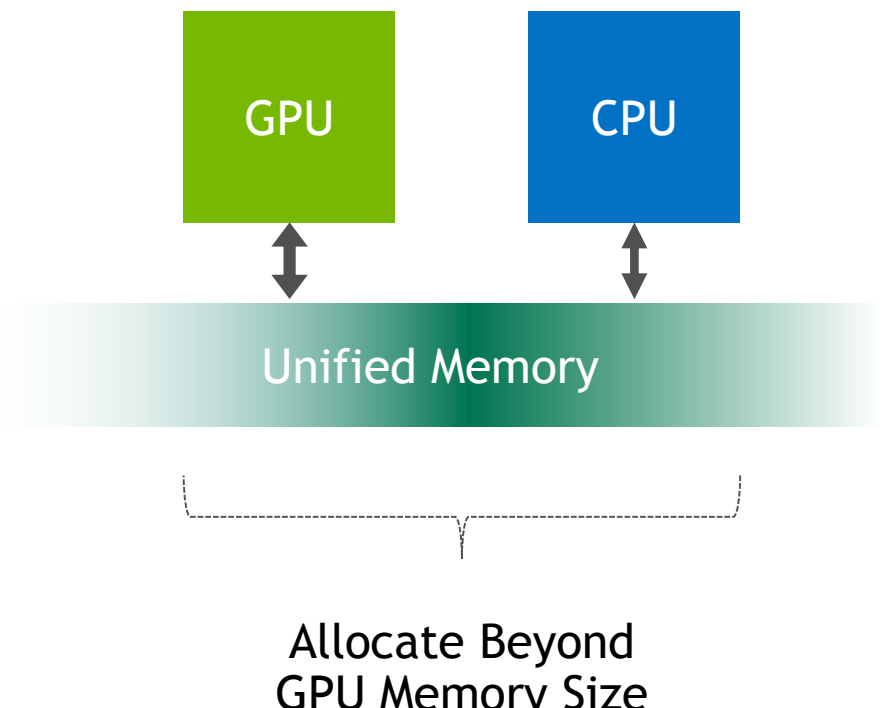
"NVIDIA has improved the speed of cuDNN with each release while extending the interface to more operations and devices at the same time."

— Evan Shelhamer, Lead Caffe Developer, UC Berkeley

UNIFIED MEMORY

Large datasets, simple programming, High Performance

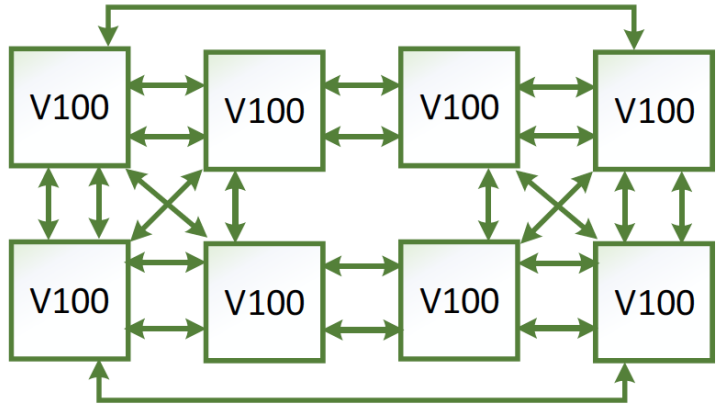
CUDA 8 and beyond



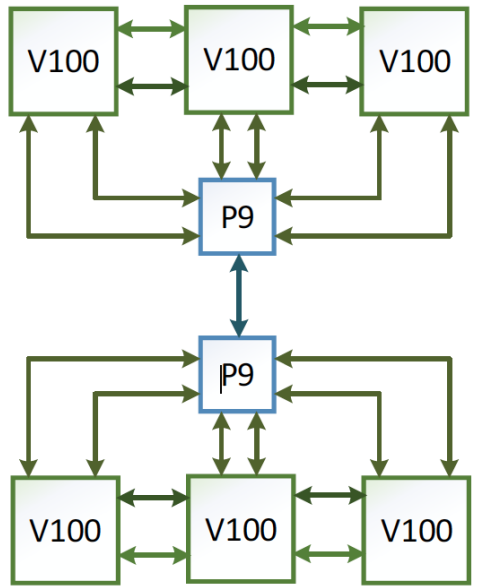
Enable Large Data Models	Oversubscribe GPU memory Allocate up to system memory size
Tune Unified Memory Performance	Usage hints via cudaMemAdvise API Explicit prefetching API
Simpler Data Access	CPU/GPU Data coherence Unified memory atomic operations

VOLTA NVLINK

- 6 NVLINKS @ 50 GB/s bidirectional
- Reduce number of lanes for lightly loaded link (Power savings)
- Coherence features for NVLINK enabled CPUs



Hybrid cube mesh
(eg. DGX1V)



POWER9 based node

NVIDIA DGX-1

AI supercomputer-appliance-in-a-box

8x Tesla V100 connected via NVLINK

(120 TFLOPS FP32, 960 Tensor TFLOPS)

Dual Xeon CPU, 512 GB Memory

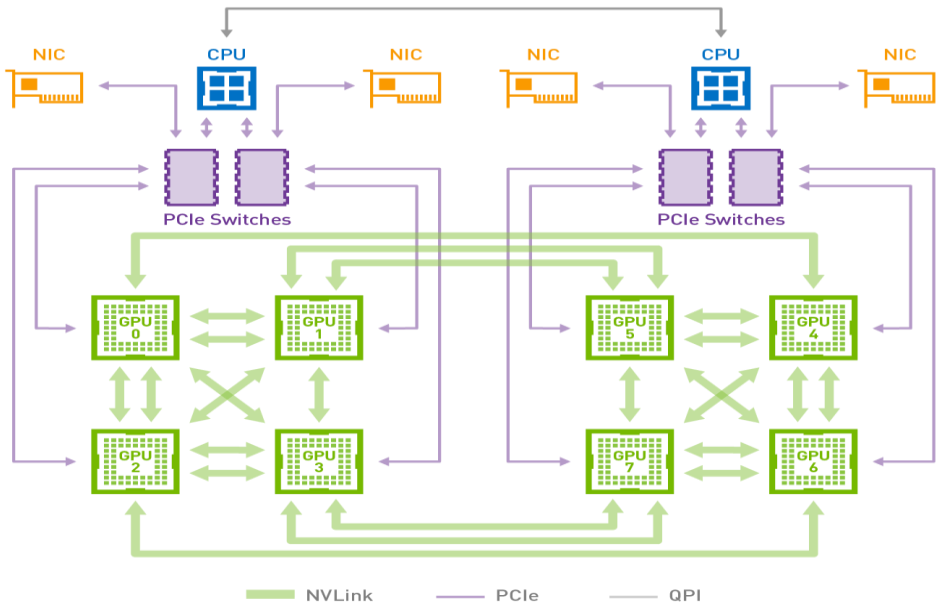
7 TB SSD Deep Learning Cache

Dual 10GbE, Quad IB 100Gb

3RU – 3200W

Optimized Deep Learning Software across the entire stack

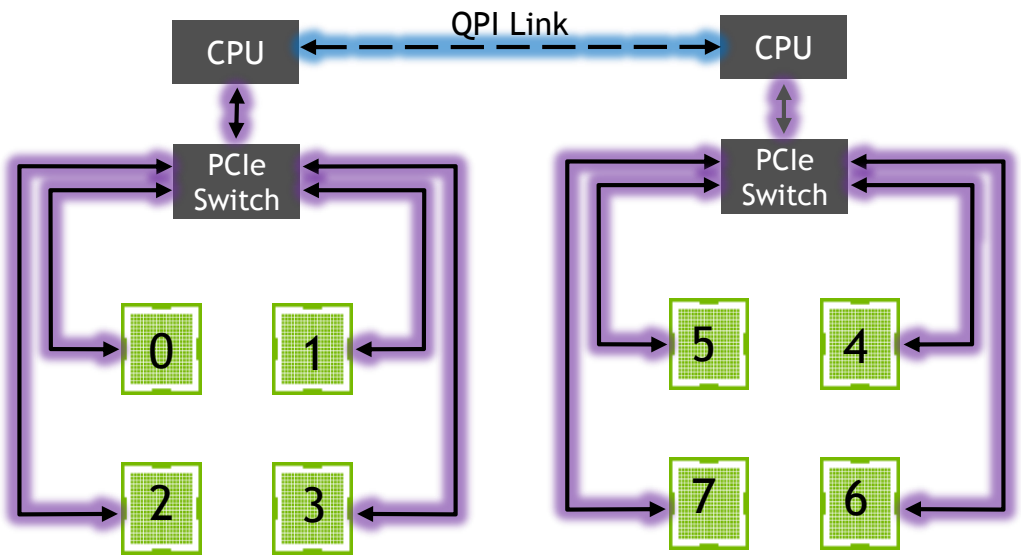
- Containerized frameworks
- Always up-to-date via the cloud



NVLINK AND MULTI-GPU SCALING

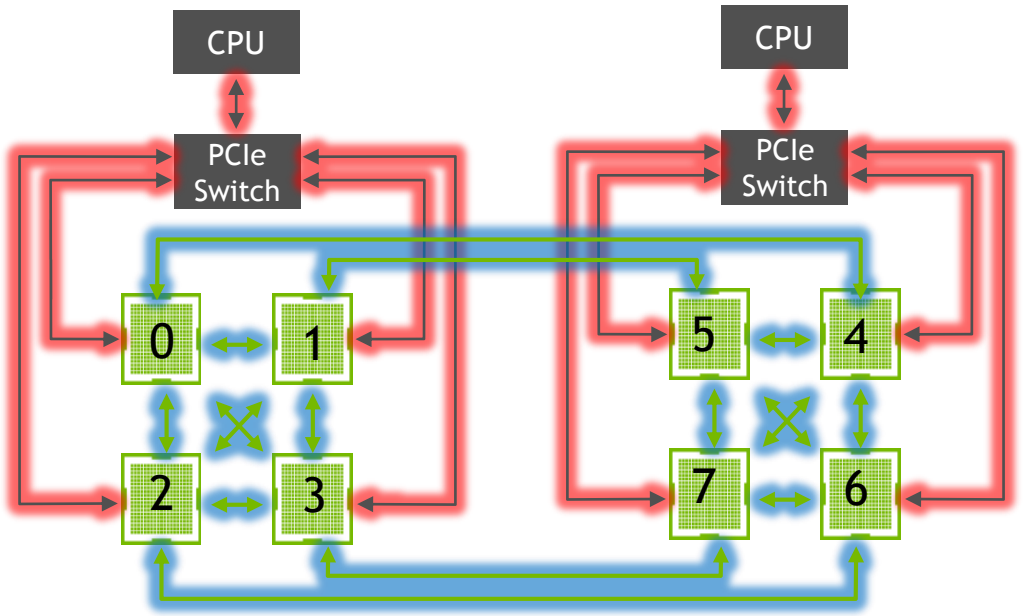
For Data Parallel Training

PCIe based system



- Data loading over PCIe
- Gradient averaging over PCIe and QPI
- Data loading and gradient averaging share communication resources: Congestion

NVLINK based system



- Data loading over PCIe (red)
- Gradient averaging over NVLink (blue)
- No sharing of communication resources: No congestion

NVIDIA Collective Communications Library (NCCL) 2

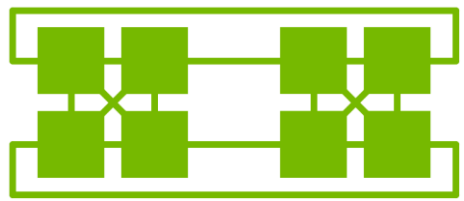
Multi-GPU and multi-node collective communication primitives

High-performance multi-GPU and multi-node collective communication primitives optimized for NVIDIA GPUs

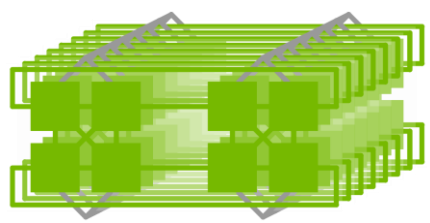
Fast routines for multi-GPU multi-node acceleration that maximizes inter-GPU bandwidth utilization

Easy to integrate and MPI compatible. Uses automatic topology detection to scale HPC and deep learning applications over PCIe and NVink

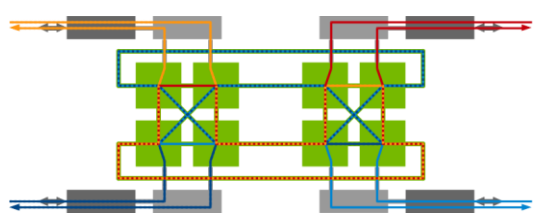
Accelerates leading deep learning frameworks such as Caffe2, Microsoft Cognitive Toolkit, MXNet, PyTorch and more



Multi-GPU:
NVLink
PCIe



Multi-Node:
InfiniBand verbs
IP Sockets



Automatic
Topology
Detection

developer.nvidia.com/nccl

WHAT'S NEW IN NCCL 2

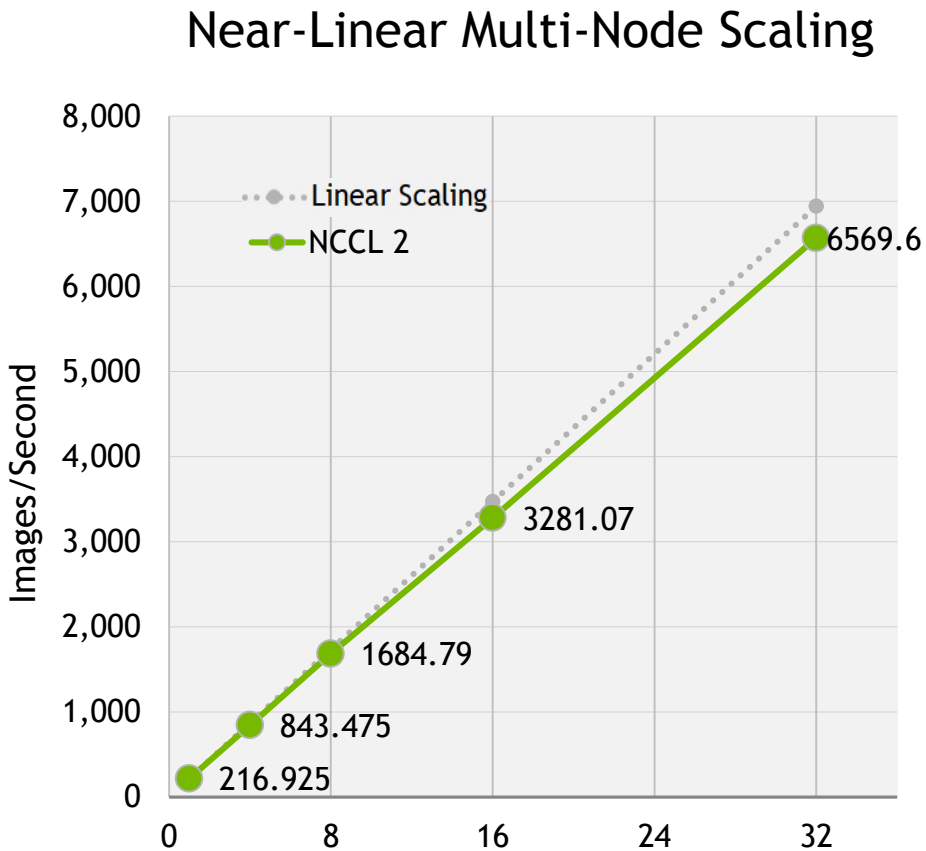
Performance

- Delivers over 90% multi-node scaling efficiency using up to eight GPU-accelerated servers

New Features

- Multi-node, multi-GPU communication collectives
- Automatic topology detection to determine optimal communication path
- Optimized to achieve high bandwidth over PCIe and NVink high-speed interconnect

Available now as a free download to members of NVIDIA Developer Program

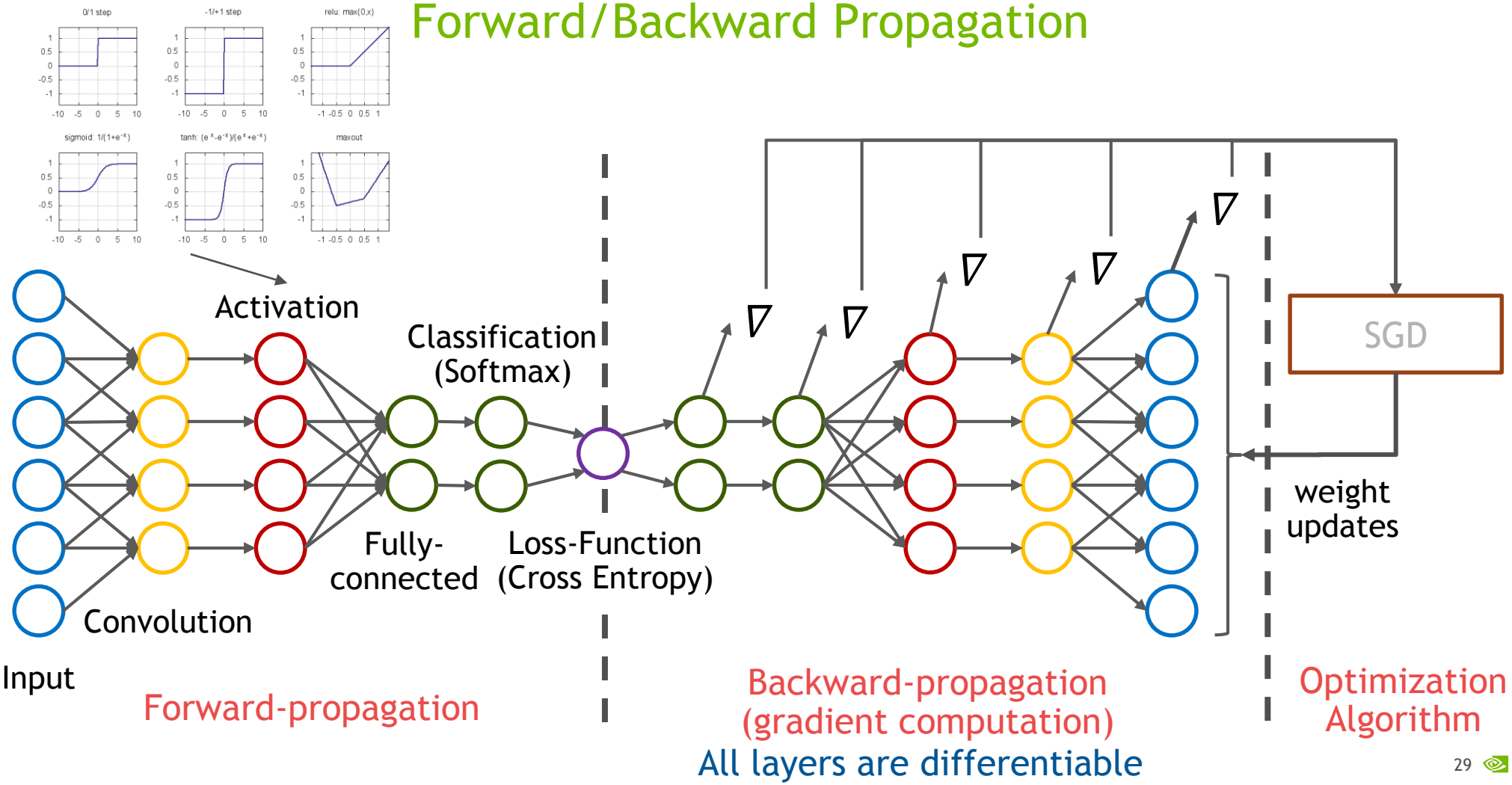


Microsoft Cognitive Toolkit multi-node scaling performance (images/sec), NVIDIA DGX-1 + cuDNN 6 (FP32), ResNet50, Batch size: 64

QUICK INTRO TO NEURAL NETWORKS

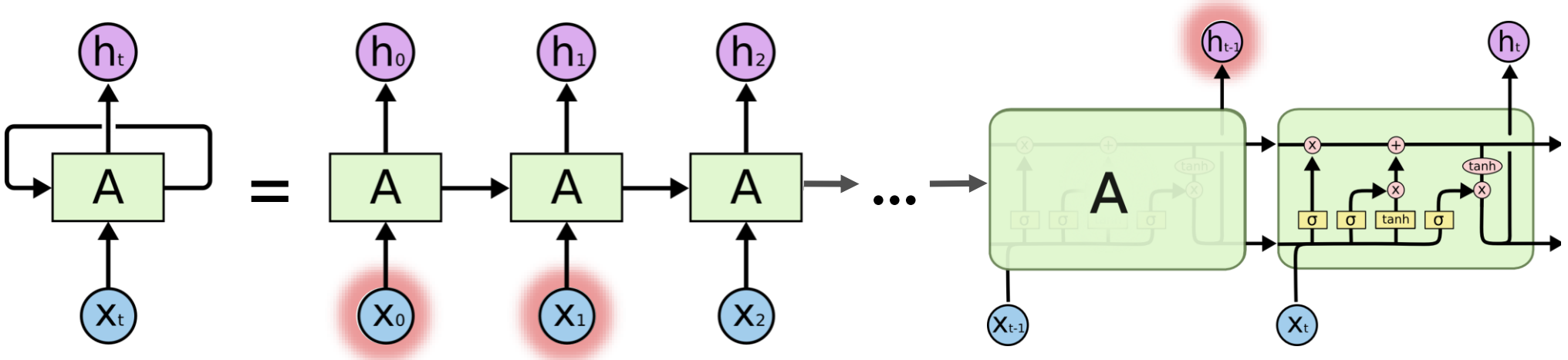
1-SLIDE INTRO TO CONVOLUTIONAL NEURAL NETS

Forward/Backward Propagation



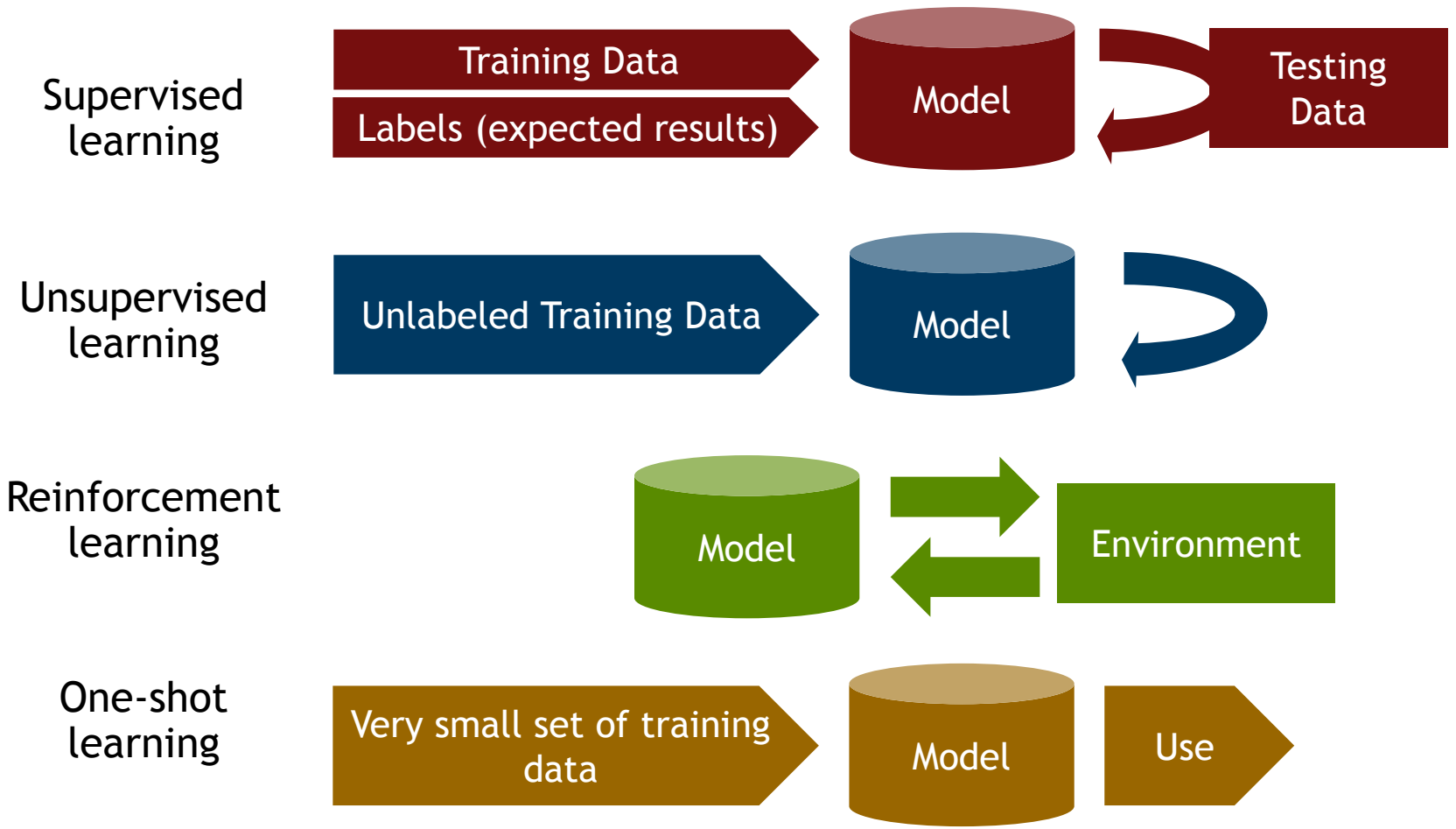
1-SLIDE INTRO TO RECURRENT NEURAL NETS

Network + Internal State => Dependencies Over Time



Diagrams from: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

CATEGORIZATION BY SIGNAL



CATEGORIZATION BY INPUT/OUTPUT

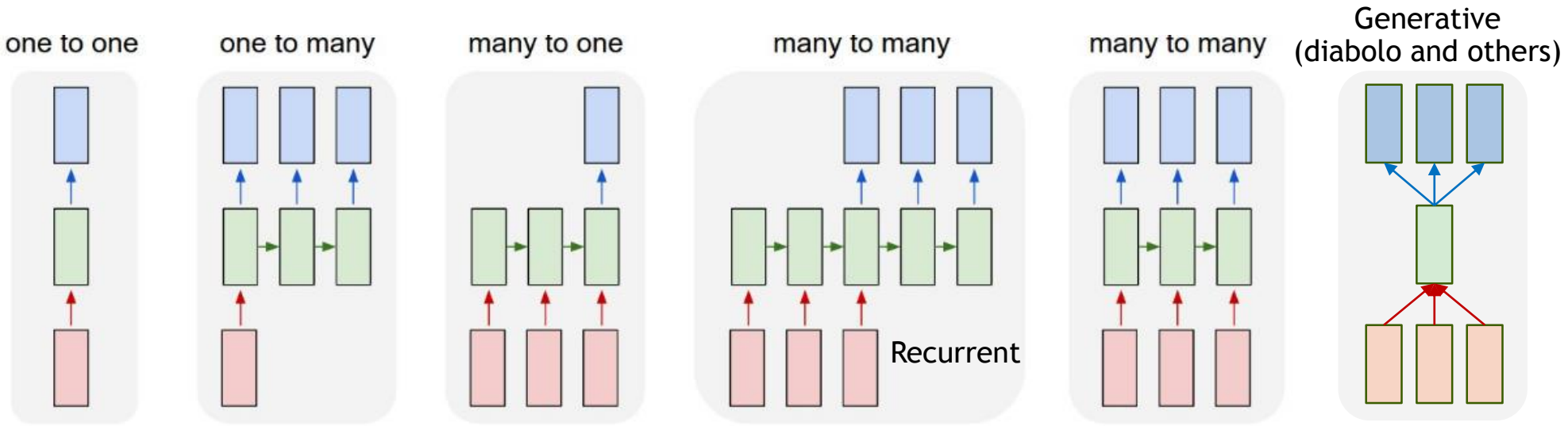
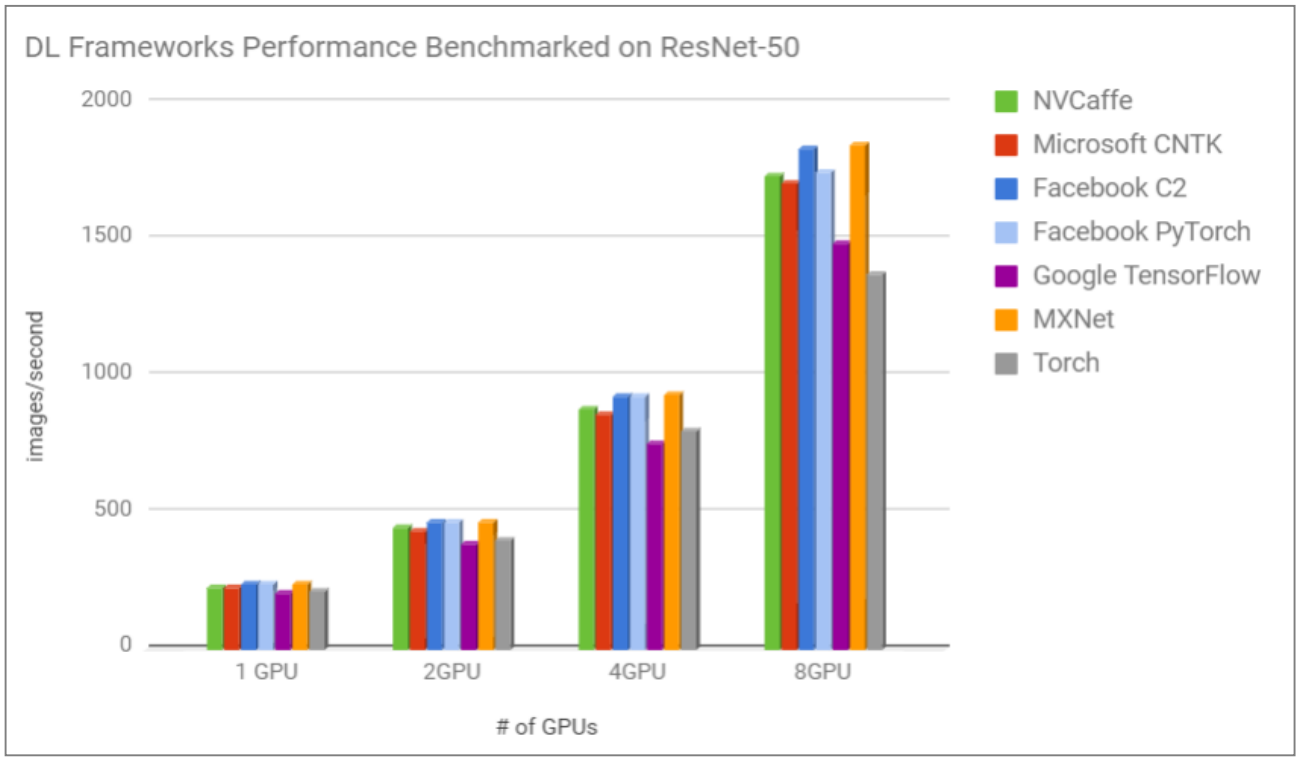


Diagram from: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

DEEP LEARNING OPTIMIZED FRAMEWORKS

Pascal DGX-1 Benchmarks



DGX-1 (Pascal) Images/s for ResNet-50; 17.07 (cuDNN 6.0.21, NCCL 2.0.3)

P100 to V100

Framework	CNN speedup
TensorFlow	2.8X
Caffe2	3.2X
PYTORCH	3X
mxnet	3.2X
Caffe	3.3X

Speed up is calculated for ResNet-50 using fp16 storage and Tensor Core Acceleration on 1 GPU (P100 to V100)

INFERENCE

AI INFERENCING IS EXPLODING

2 Trillion

Messages Per Day On LinkedIn

PERSONALIZATION

500M

Daily active users of iFlyTek

SPEECH

140 Billion

Words Per Day Translated by Google

TRANSLATION

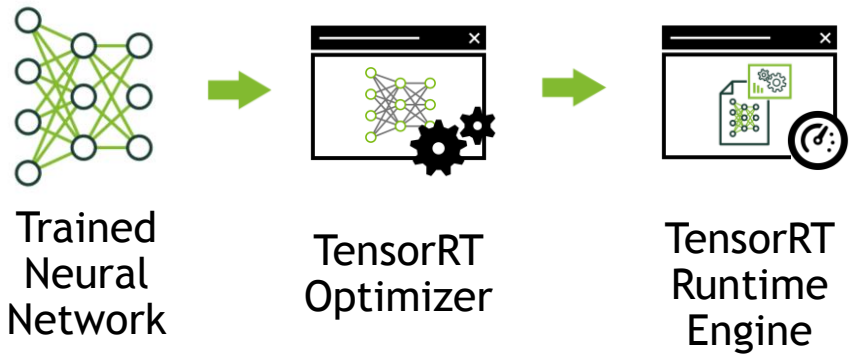
60 Billion

Video frames/day uploaded on Youtube

VIDEO

NVIDIA TensorRT 3

Deep Learning Inference Optimizer and Runtime

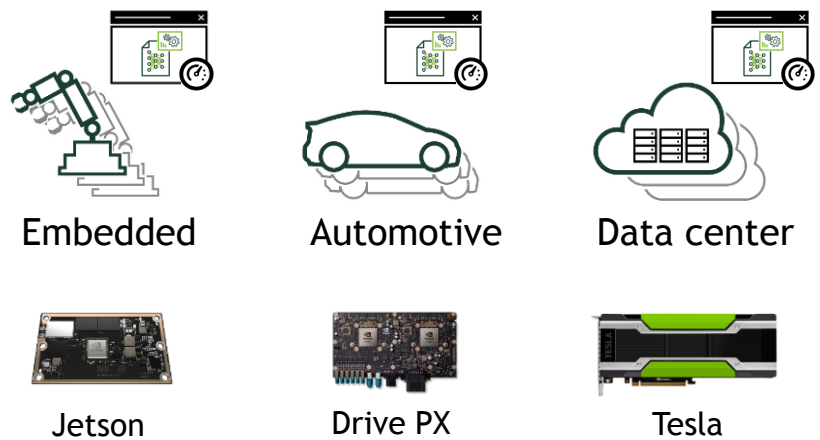


High performance neural network inference optimizer and runtime engine for production deployment

Maximize inference throughput for latency-critical services in hyperscale datacenters, embedded, and automotive production environments.

Optimize models trained in TensorFlow or Caffe to generate runtime engines that maximizes inference throughput

Deploy faster, more responsive and memory efficient deep learning applications with INT8 and FP16 optimized precision support

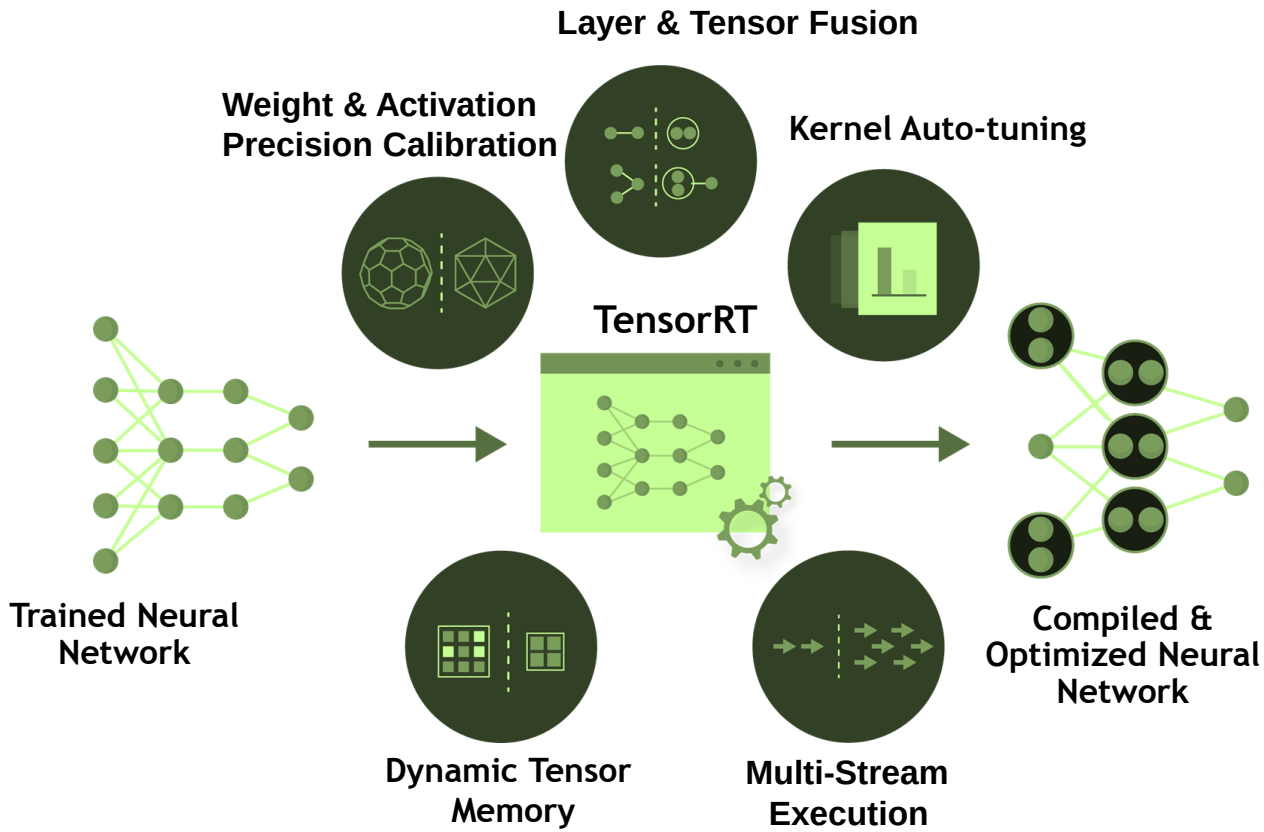


developer.nvidia.com/tensorrt

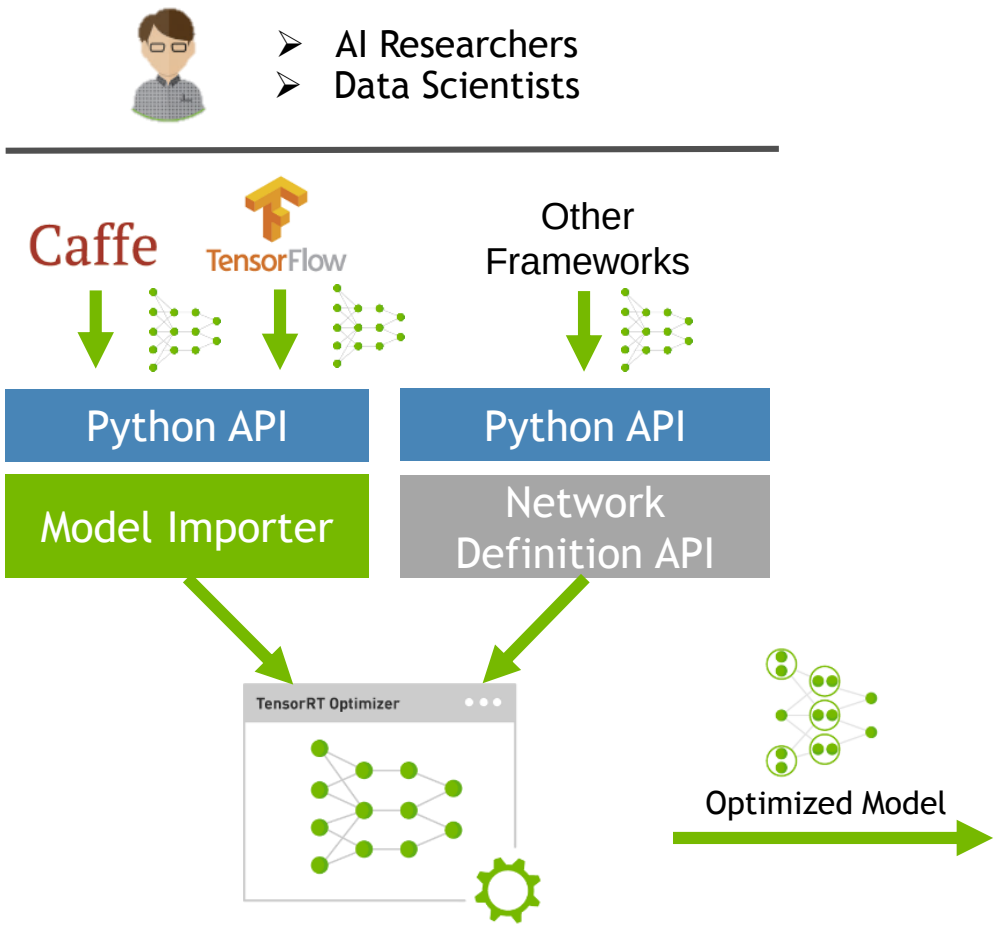
NVIDIA TensorRT 3

Programmable Inference Accelerator

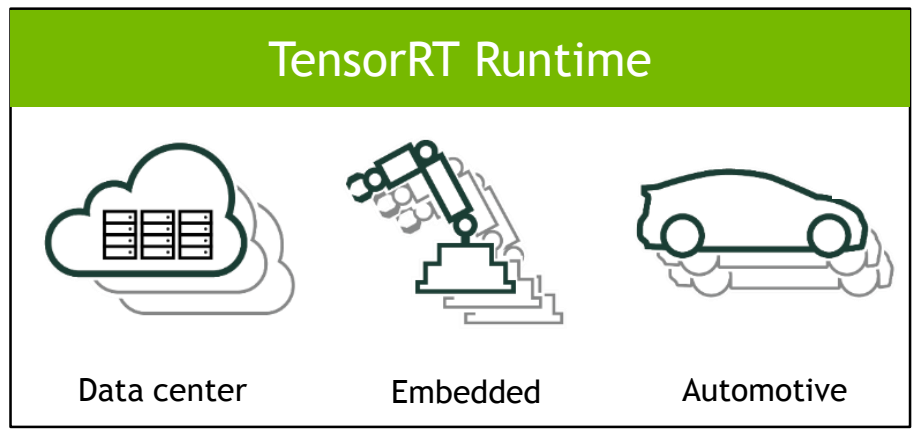
- Compiler for Optimized Neural Networks
- Weight & Activation Precision Calibration
- Layer & Tensor Fusion
- Kernel Auto-Tuning
- Multi-Stream Execution



TENSORRT 3: TENSORFLOW IMPORTER AND PYTHON API



- Optimize and deploy TensorFlow models that are up to **18x faster** vs. TensorFlow framework
- Improved productivity with **easy to use Python API for** Data Science workflows



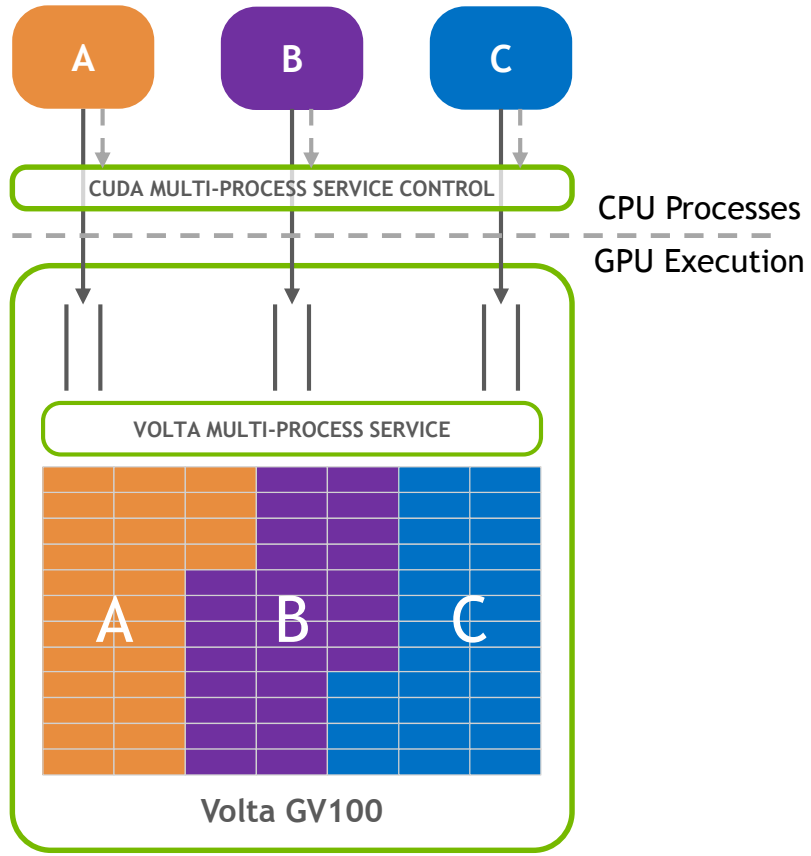
VOLTA MULTI-PROCESS SERVICE

Volta MPS Enhancements:

- MPS clients submit work directly to the work queues within the GPU
  - Reduced launch latency
  - Improved launch throughput
- Improved isolation amongst MPS clients
  - Address isolation with independent address spaces
  - Improved quality of service (QoS)
- 3x more clients than Pascal

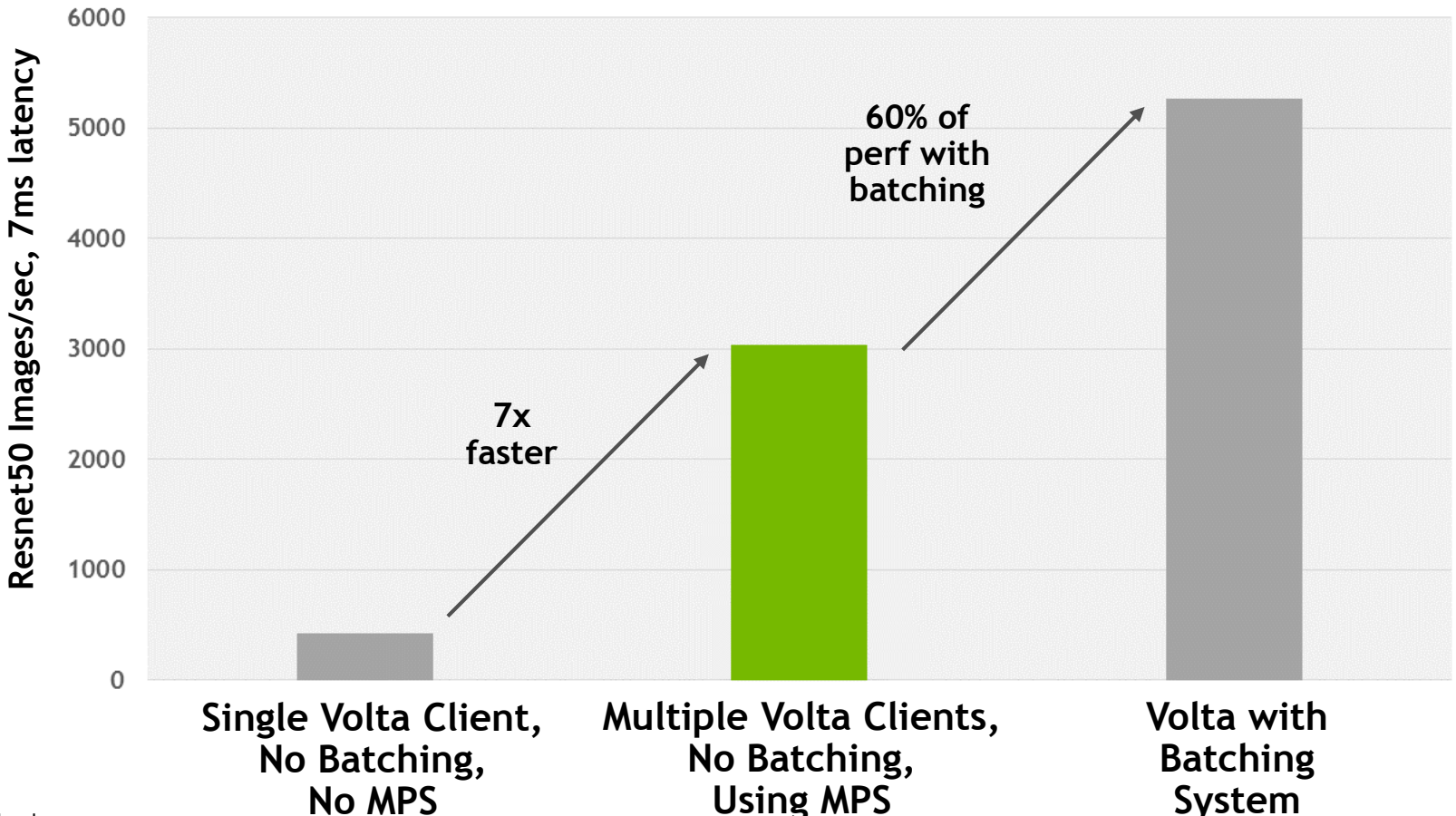
Hardware Accelerated Work Submission

Hardware Isolation



VOLTA MPS FOR INFERENCE

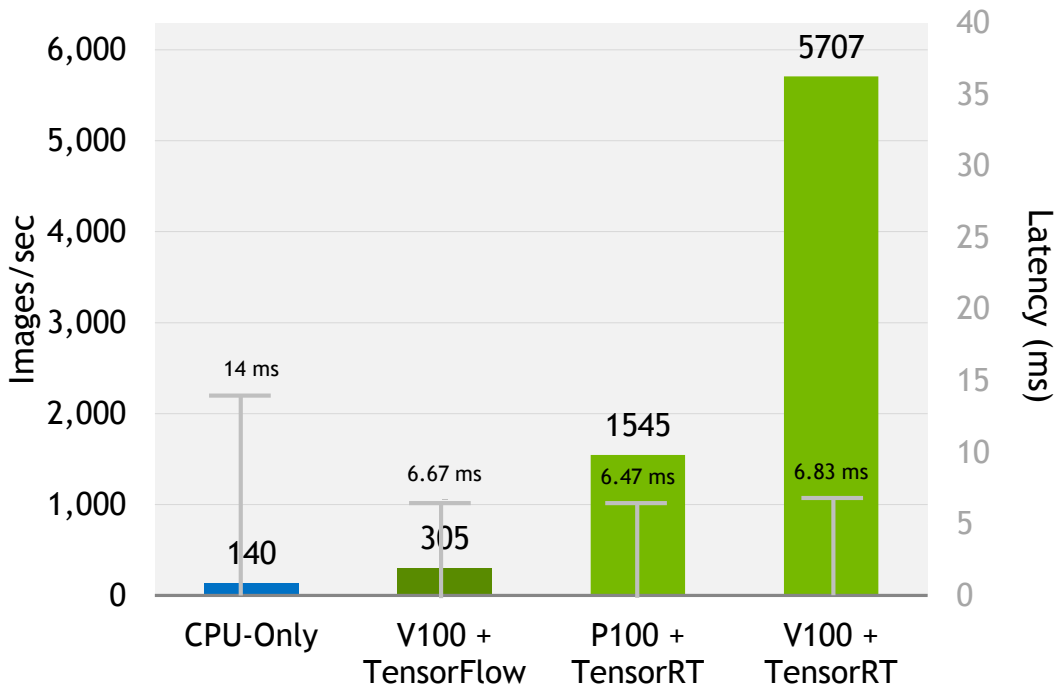
Efficient inference deployment without batching system



V100 measured on pre-production hardware.

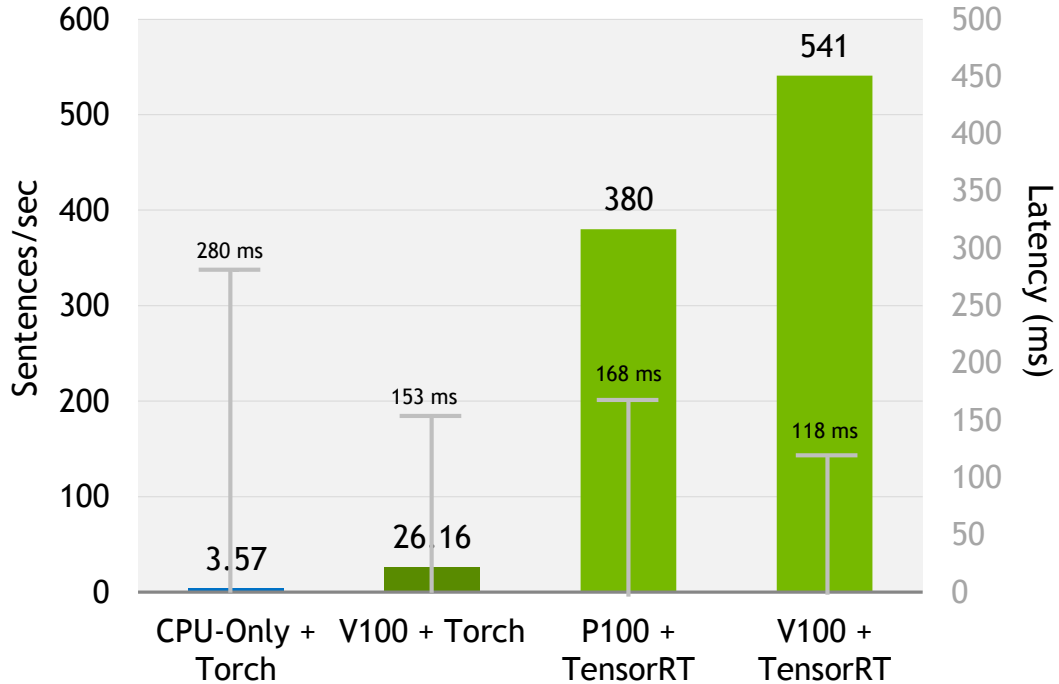
TENSORRT 3 PERFORMANCE

40x Faster CNNs on V100 vs. CPU-Only Under 7ms Latency (ResNet50)



Inference throughput (images/sec) on ResNet50. **V100 + TensorRT**: NVIDIA TensorRT (FP16), batch size 39, Tesla V100-SXM2-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On. **P100 + TensorRT**: NVIDIA TensorRT (FP16), batch size 10, Tesla P100-PCIE-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On **V100 + TensorFlow**: Preview of volta optimized TensorFlow (FP16), batch size 2, Tesla V100-PCIE-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On. **CPU-Only:** Intel Xeon-D 1587 Broadwell-E CPU and Intel DL SDK. Score doubled to comprehend Intel's stated claim of 2x performance improvement on Skylake with AVX512.

140x Faster Language Translation RNNs on V100 vs. CPU-Only Inference (OpenNMT)



Inference throughput (sentences/sec) on OpenNMT 692M. **V100 + TensorRT**: NVIDIA TensorRT (FP32), batch size 64, Tesla V100-PCIE-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On. **P100 + TensorRT**: NVIDIA TensorRT (FP32), batch size 64, Tesla P100-PCIE-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On **V100 + Torch**: Torch (FP32), batch size 4, Tesla V100-PCIE-16GB, E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On. **CPU-Only:** Torch (FP32), batch size 1, Intel E5-2690 v4@2.60GHz 3.5GHz Turbo (Broadwell) HT On

developer.nvidia.com/tensorrt

DL + HPC:
JOINTLY SOLVE NEW PROBLEMS, BETTER

VISION: DATA SCIENCE DRIVES ARCHITECTURE

Microsoft
Olympus

Facebook
OCP Big Basin

Data science/Deep learning needs heavily influence architecture

Extensible NVLink, CPU as orchestrator, ..

System level: Dense nodes, high-performance intra-node communication

DGX-1/SATURNV, Big Basin, Minsky, Olympus, …

GPU level: Instruction set influenced by needs of data science

Half/mixed precision, int8, ..

Develop HPC applications for high-density nodes (also e.g. CORAL)

Leverage the DL hardware features for scientific computing

VISION: DATA SCIENCE DRIVES SOFTWARE-STACK

NVIDIA is the AI computing company, thinking lots about software!

Data science mission-critical to non-traditional HPC organizations

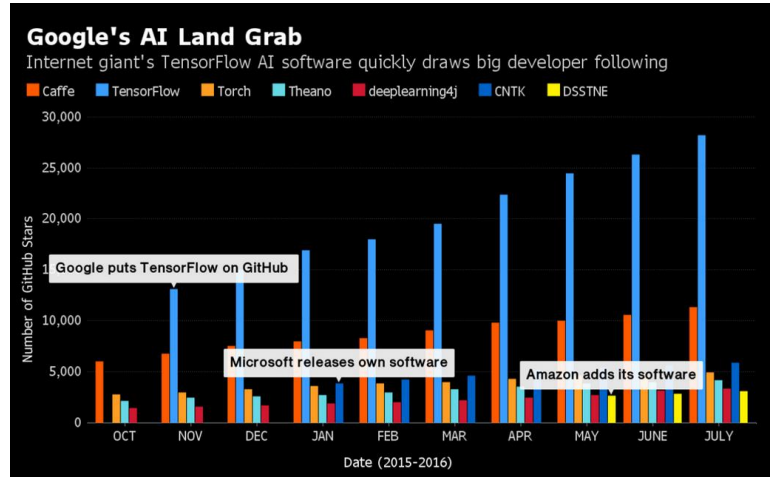
Deep learning, graph analytics, in-core databases,..

Sustainability and performance by scale

Frameworks supported by big corps, large communities

Big market, big support by all vendors

⇒ Economics drive performance portability and sustainability



Trilinos: 143 stars, Caffe: 16’679 stars

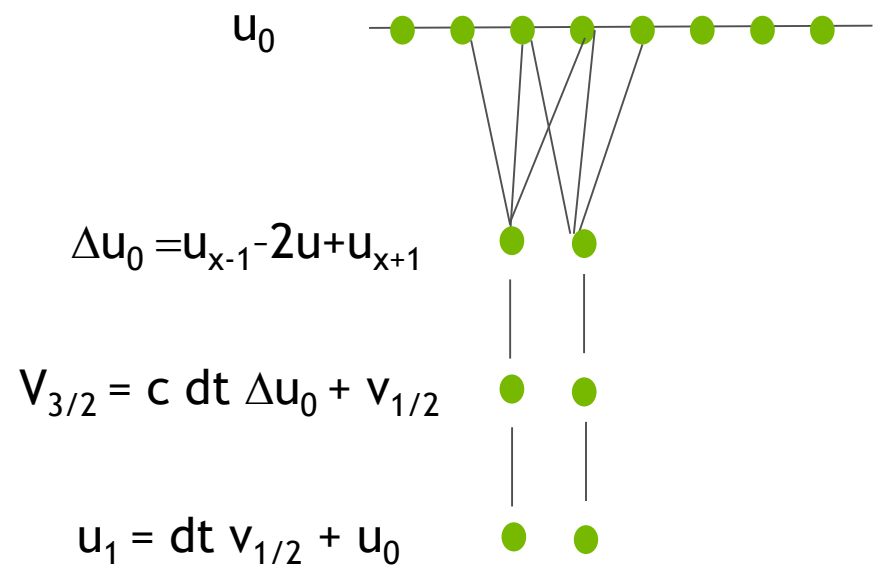
Cast HPC algorithms in DL terms

EXAMPLE: WAVE EQUATION VIA CONV NEURAL NETWORK

Applying stencils = Inference in Conv Network

$$\frac{dv}{dt} = c\,\Delta\,u$$

$$\frac{du}{dt} = v$$



Simplified 1D example cartoon
Approach works for higher dimensions

```
layer {
        name          : “laplace_u”
        type          : “convolution”
        bottom        : “u_0”
        top           : “laplace_u”
        convolution_param {
                num_output  : 1
                kernel_size : 3
                stride      : 1
                pad         : 1
..
}

layer {
        name          : “velocity_update”
        type          : “Eltwise”
        bottom        : “laplace_u”
        top           : “velocity_update”
}

layer  {
        name          : “position_update”
        type          : “Eltwise”
        bottom        : “velocity_update”
        top           : “position_update”
}
```

Other examples: Stencils, Spectral transforms, spectral elements, ..

VISION: HPC CAN CONTRIBUTE TO EMERGING DATA SCIENCE NEEDS

HPC solved lots of “new” problems in the past

DL will need distributed memory parallelism

New challenges for DL algorithms

HPC has probably hit those challenges in the past

Better implementation, better algorithms

https://www.hpcwire.com/2017/02/21/hpc-technique-benefits-deep-learning/

Collaborate with DL framework developers, contribute to DL frameworks

First step: speak a common language

VISION: COMBINED DL AND HPC

Jointly solve new problems, better

Many HPC models have “inaccurate” components, eg parameterized sub-model

Often complex control flow

A trained network might result in higher performance, better accuracy

Possible examples: collisional cross-sections, chemical reaction chains,

Simplified if rest of application is already in DL friendly fashion

Revisit parameterized models

COMBINING THE STRENGTHS OF HPC AND AI

HPC	AI
+40 years of Algorithms based on first principles theory Proven statistical models for accurate results in multiple science domains	New methods to improve predictive accuracy, insight into new phenomena and response time with previously unnavigable data sets
Develop training data sets using first principal models Apply Bayesian regression methods to expedite/ensure training accuracy Incorporate AI models in semi-empirical style applications to improve throughput Validate new findings from AI	Train inference models to improve accuracy and comprehend more of the physical parameter space Implement inference models with real time interactivity Analyze data sets that are simply intractable with classic statistical models Control and manage complex scientific experiments or apparatus

WHAT IF I DON'T HAVE ENOUGH TRAINING DATA?

HOW MUCH TRAINING DATA IS NEEDED?

A recursive answer

- No general answer, need to experiment
  - Test error >> training error: probably more data (overfitting?)
  - Test error ≈ training error: more data probably doesn't help
  - Look at learned filters: noisy filters generally want more training
- For N functions, need > log(N)+c training cases (see: A Theory of the Learnable, L.G. Valiant, 1984)
  - Example: N parameters of type float32 = max 2^{32N} distinct networks, wants 32N samples.
- Rough Guideline: some constant (e.g. 10) multiple of # parameters to avoid overfitting
  - Batch normalization, Regularization, etc can give improvement

HOW LARGE SHOULD MY NETWORK BE?

A recursive answer

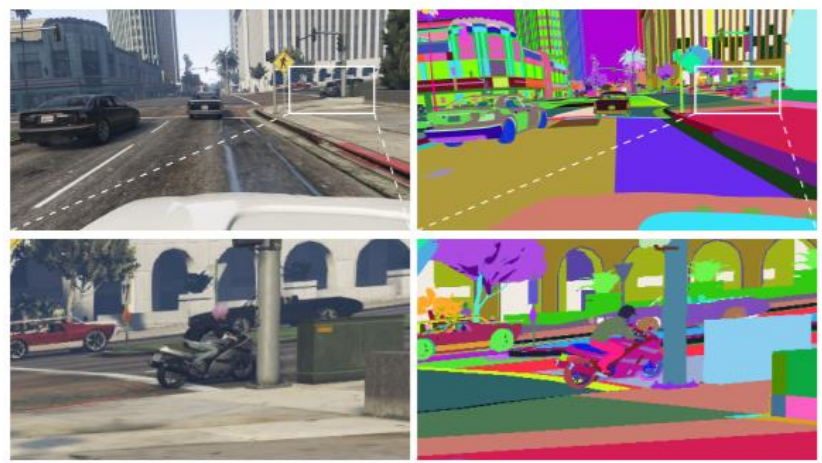
- Depends on the amount of training data available
  - Too small: bad generalization; Too large: overfitting
- And the complexity of the function to be learned[1]
  - 1-hidden layer (grows exponentially) vs. deep networks (may grow linearly)
- Rough Design Guideline:
  - First and last layer are given by model
  - Number of nodes of a hidden layer somewhere between the size of its input and output layer
  - Number of nodes in layer should be < 2 * #input nodes to avoid overfitting
- The rest is Art(?)

[1] Y.Bengio, Y.LeCun. Scaling learning algorithms towards AI. Large-scale Kernel Machines, 2007

HOW TO GET MORE TRAINING DATA?

And their Labels

- Data Augmentation and Data Synthesis
  - e.g., adding artificial background noise to speech samples (10x increase for Baidu)
  - e.g., adding shifts, rotations, distortions to images
- Training and Testing on Simulators
  - *Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, J. Tobin et al., Robotics 2017*
  - Self-Driving Vehicles *Playing for Data: Ground Truth from Computer Games, S.Richter et al., ECCV, 2016)*
- One-shot Learning, GANs (Apple uses GANs to improve generated training data), Autoencoders?



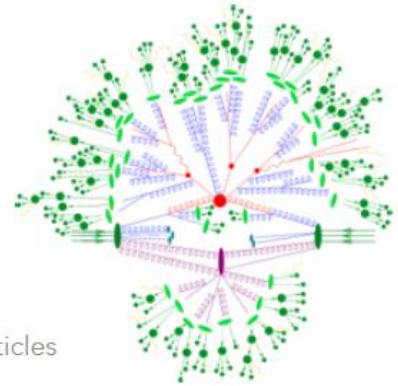
EXAMPLE: PARTICLE PHYSICS (CERN)

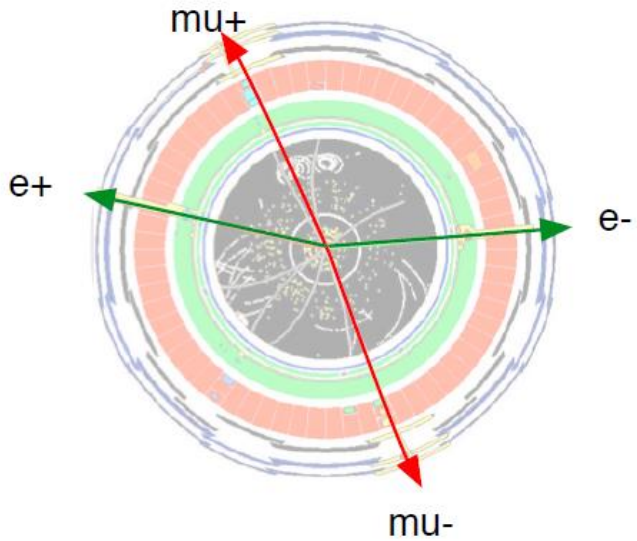
$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$
$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$
$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma\text{ and Higgs masses and couplings}}$$
$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: 2 → O(10) → O(100) particles





3) The interaction of outgoing particles with the detector is simulated.

>100 million sensors

4) Finally, we run particle identification and feature extraction algorithms on the simulated data as if they were from real collisions.

~10-30 features describe interesting part

From: K.Cranmer. Machine Learning & Likelihoos Free Inference in Particle Physics, NIPS2016

MY DATA IS SYMMETRIC OR INVARIANT IN XYZ?

INVARIANTS AND SYMMETRIES IN DATA

Pattern Recognition

- CNNs don’t understand Invariants and Symmetries out of the box
  - Pooling and downsampling helps with some transformations
- (Training and Test-time) Data augmentation may explode the training set
  - Scale/Rotate/Transform/Perturbate each training image many times?
- Approaches:
  - Teach networks about certain symmetries (e.g. rotation)
  - Normalize/preprocess data to ensure well-known layout
  - Find encoding of the data that is invariant to certain operations

EXPLOITING SYMMETRY IN CONV NETS

Teaching CNNs about Rotation

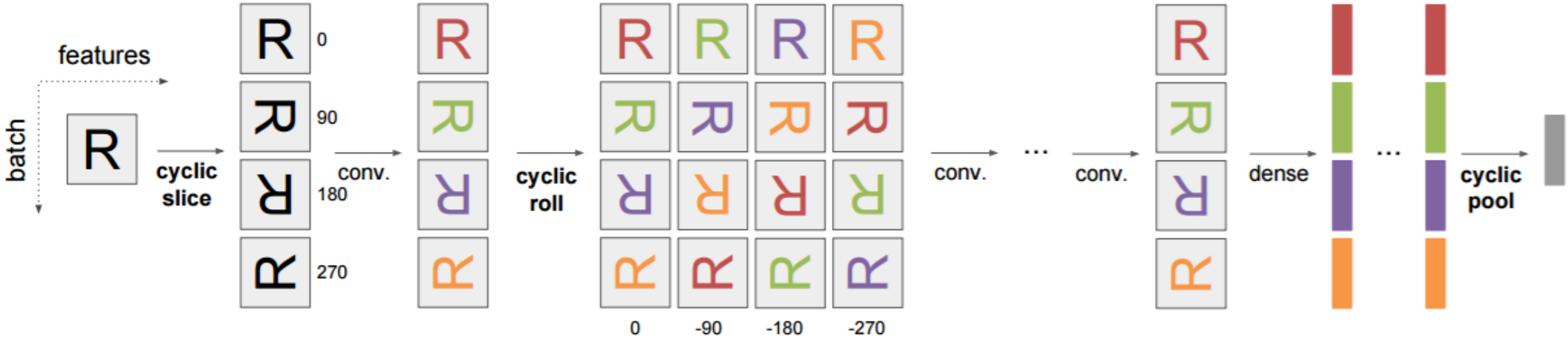


Figure 3. Schematic representation of the effect of the cyclic slice, roll and pool operations on the feature maps in a CNN. Arrows represent network layers. Each square represents a minibatch of feature maps. The letter 'R' is used to clearly distinguish orientations. Different colours are used to indicate that feature maps are qualitatively different, i.e. they are not rotations of each other. Feature maps in a column are stacked along the batch dimension in practice; feature maps in a row are stacked along the feature dimension.

From: S.Dieleman et al. Exploiting Cyclic Symmetry in Convolutional Neural Networks, CoRR, 2016

NORMALIZING AND PRE-PROCESSING

DL trained on jet images vs. physically-motivated feature driven approaches

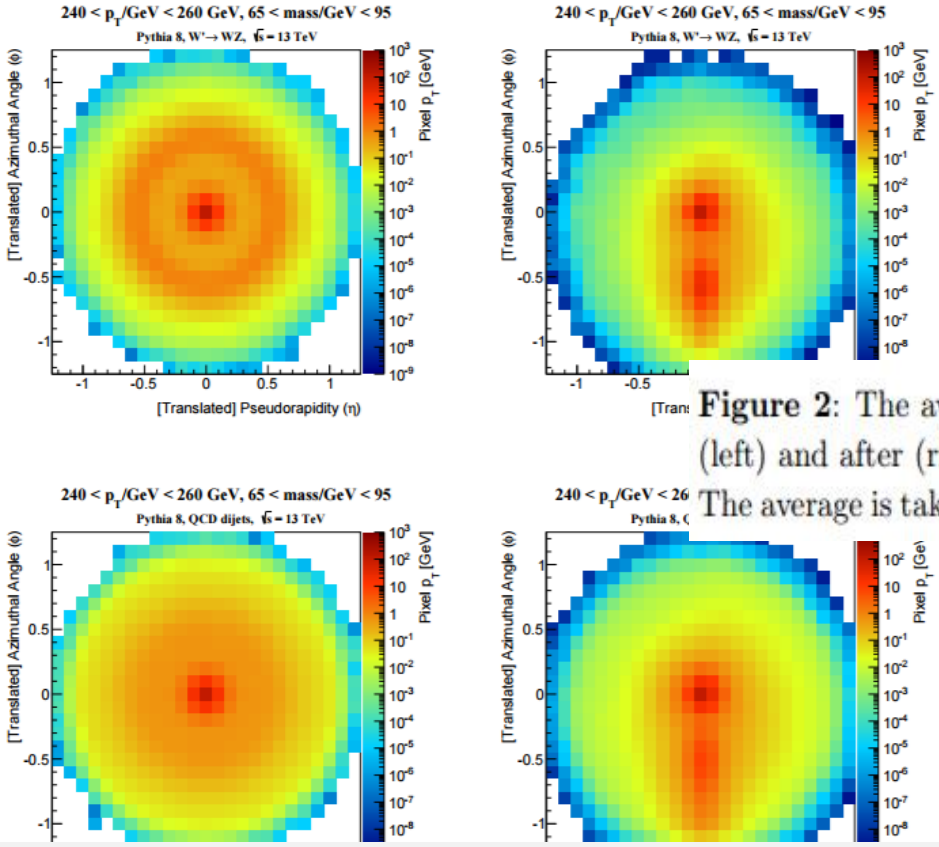


Figure 2: The average jet image for signal W jets (top) and background QCD jets (bottom) before (left) and after (right) applying the rotation, re-pixelation, and inversion steps of the pre-processing. The average is taken over images of jets with 240 GeV $< p_T <$ 260 GeV and 65 GeV $<$ mass $<$ 95 GeV.

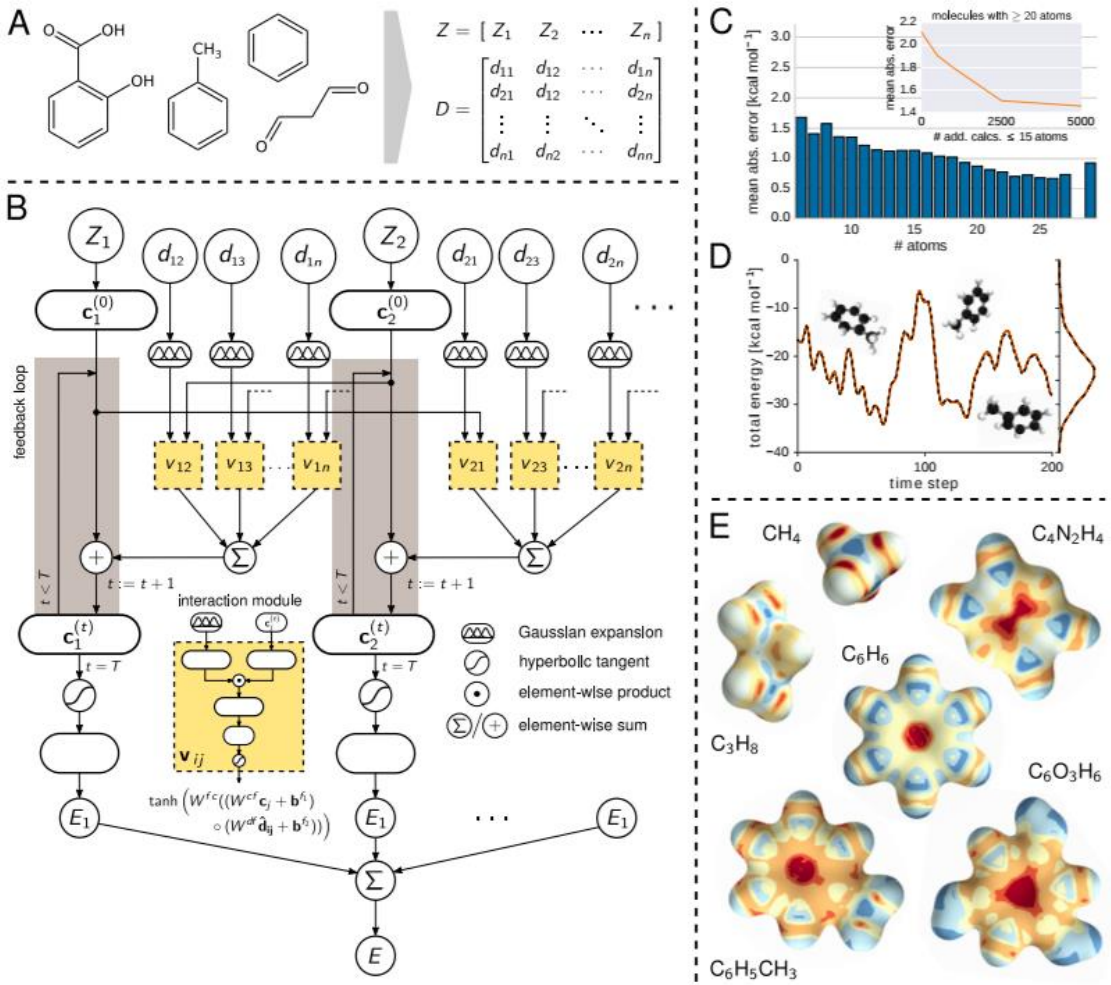
From: L. de Oliveira et al., Jet-Images -- Deep Learning Edition, JHEP07, 2016

From: L. de Oliveira et al., Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis, 2017

INVARIANT ENCODING

Molecules are encoded as Vectors of Nuclear Charges and Inter-atomic Distance Matrices

=> Translation and rotation Invariant Representation



From: K.Schütt et al., Quantum-Chemical Insights from Deep Tensor Neural Networks, arXiv:1609.08259

HOW DO I REPRESENT MY DATA IN NEURAL NETWORKS?

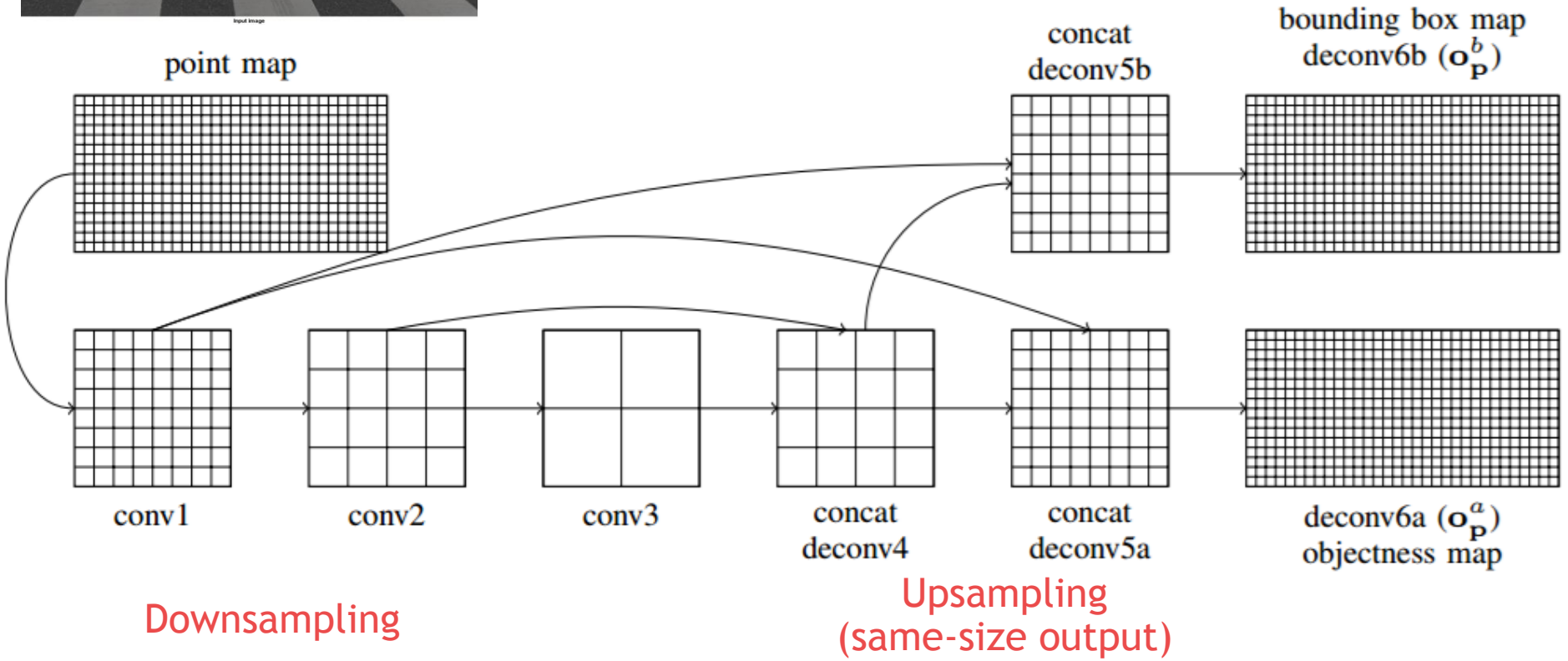
SIMPLE EXAMPLE: CLASSIFICATION

One-Hot Encoding

Training Data	Scalar Encoding	One-Hot Encoding
	[0.0]	[1,0,0,0,0,0,0,0,0,0]
	[1.0]	[0,1,0,0,0,0,0,0,0,0]
	[2.0]	[0,0,1,0,0,0,0,0,0,0]
	[3.0]	[0,0,0,1,0,0,0,0,0,0]
	[4.0]	[0,0,0,0,1,0,0,0,0,0]
	[5.0]	[0,0,0,0,0,1,0,0,0,0]
	[6.0]	[0,0,0,0,0,0,1,0,0,0]
	[7.0]	[0,0,0,0,0,0,0,1,0,0]
	[8.0]	[0,0,0,0,0,0,0,0,1,0]
	[9.0]	[0,0,0,0,0,0,0,0,0,1]

IMAGE SEGMENTATION & BOUNDING BOXES

Creative use of feature channels



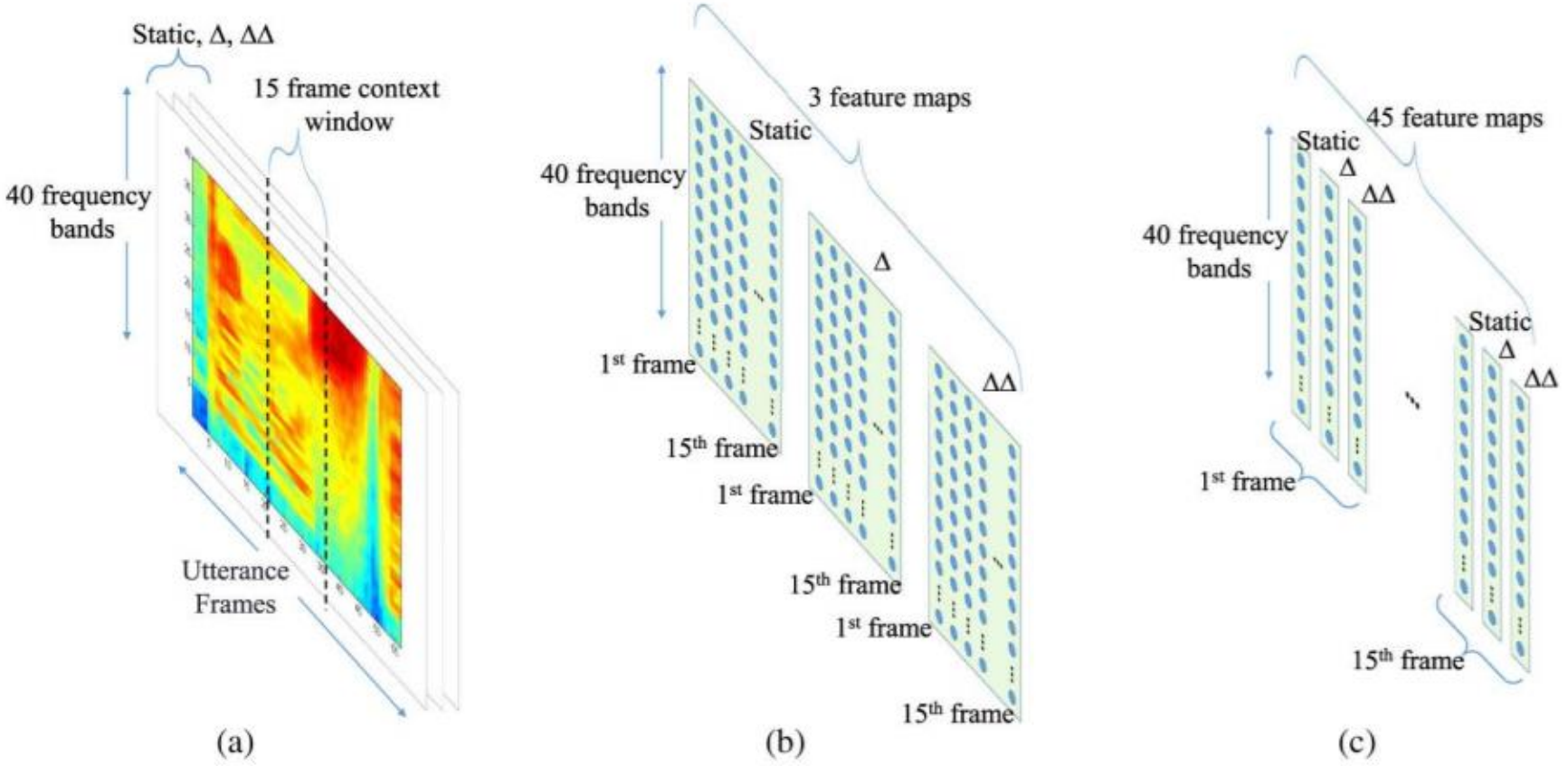
1 Channel per BB point coordinate

1 Channel per Object Class (incl. Background)

Diagram From: B.Li, T.Zhang, T.Xia. Vehicle Detection from 3D Lidar Using Fully Convolutional Network, CoRR, 2016
GIF from: https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5/

ORGANIZING SPEECH INTO FEATURE MAPS

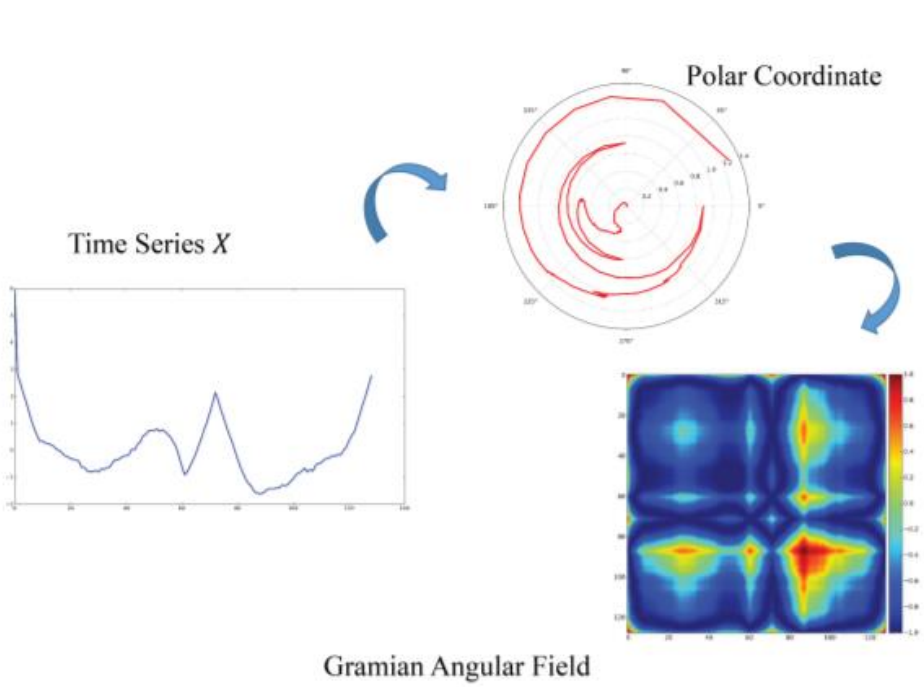
Reducing Problems to Image Recognition

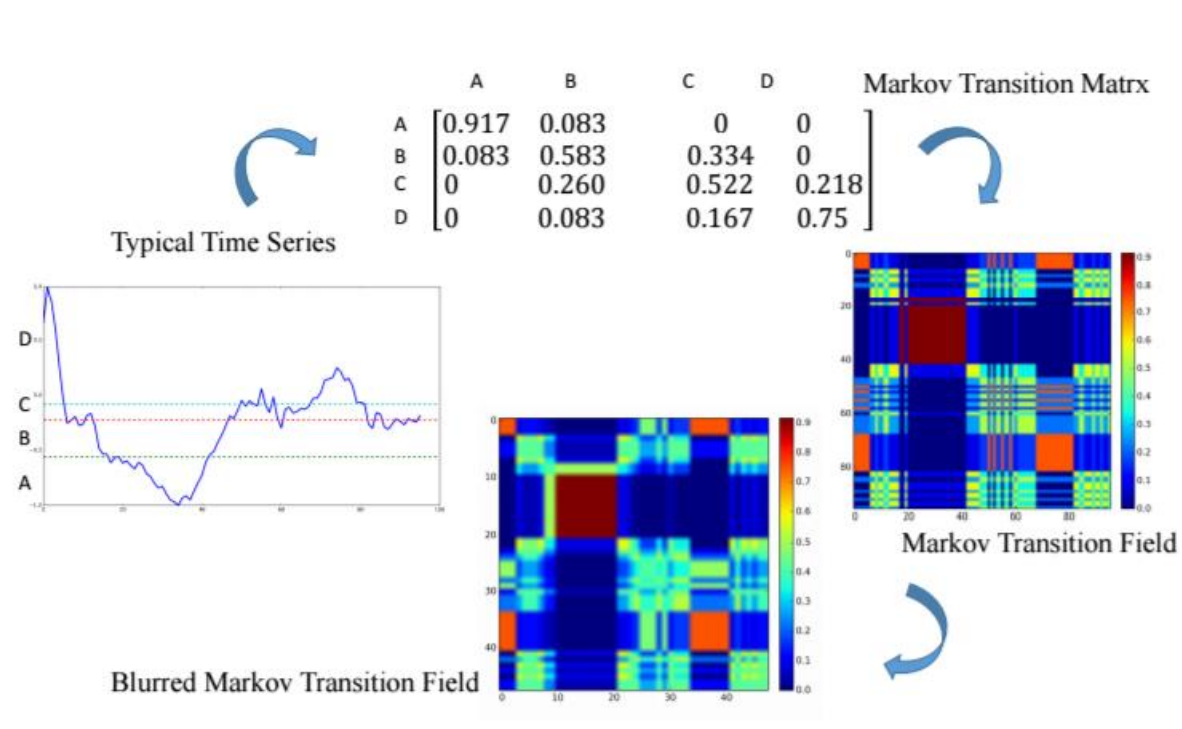


From: Ossama et al. Convolutional Neural Networks for Speech Recognition, IEEE/ACM Trans. Audio, Speech, and Lang. Proc, 2014

ENCODING TIME SERIES AS IMAGES

Gramian Angular Fields (GAF) and Markov Transition Fields (MTF)

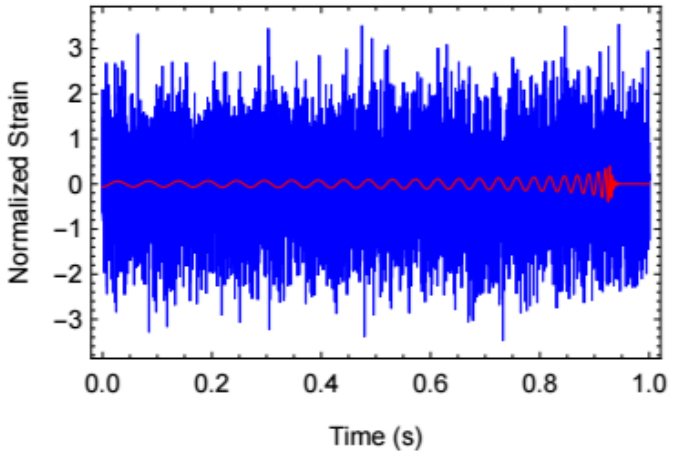




From: Z.Wang, T.Oates. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks, AAAI Workshop, 2015

DL FOR SIGNAL PROCESSING

Looking for Gravitational Waves



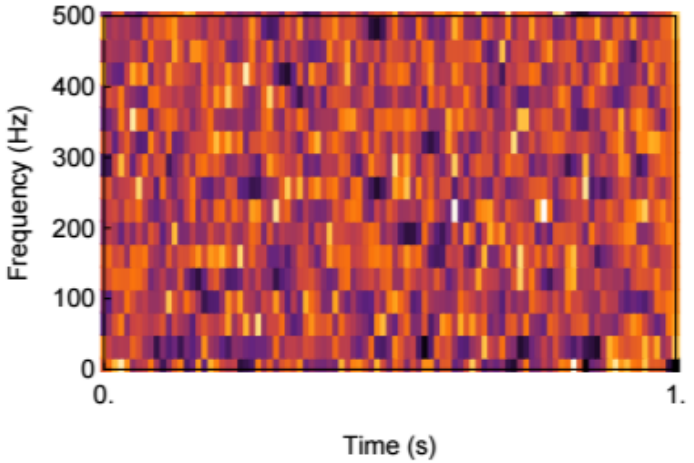
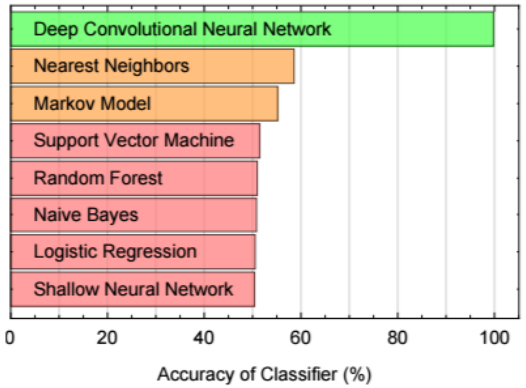


FIG. 2. Left panel: The blue curve is a sample of an input to our DNN algorithm. It contains a BBH GW signal (red) which was whitened with aLIGO's PSD design sensitivity (see Figure 3) and superimposed in noisy data with SNR = 0.5. Right panel: The corresponding spectrogram showing that the BBH GW signal on the left is not visible and thus cannot be detected by any algorithm trained for image recognition. Nevertheless, our DNN detects the presence of this signal from the time-series data, and reconstructs the source's parameters with excellent accuracy.



Classifier:
Detect Presence of GWs

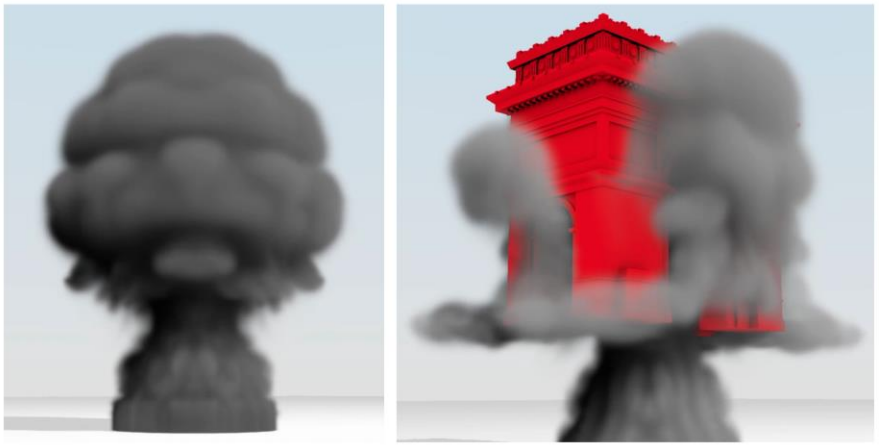
Regression:
Parameter Estimation
(i.e., masses of the two black holes)

From: D.George, E.A.Huerta. Deep Neural Networks to Enable Real-time Multimessenger Astrophysics, arXiv:1701.00008 [astro-ph.IM]

HOW CAN I TRUST THE NETWORK?

EULERIAN FLUID SIMULATION WITH CONVOLUTIONAL NETWORKS

Produces “visually similar results”, but is it “science”?



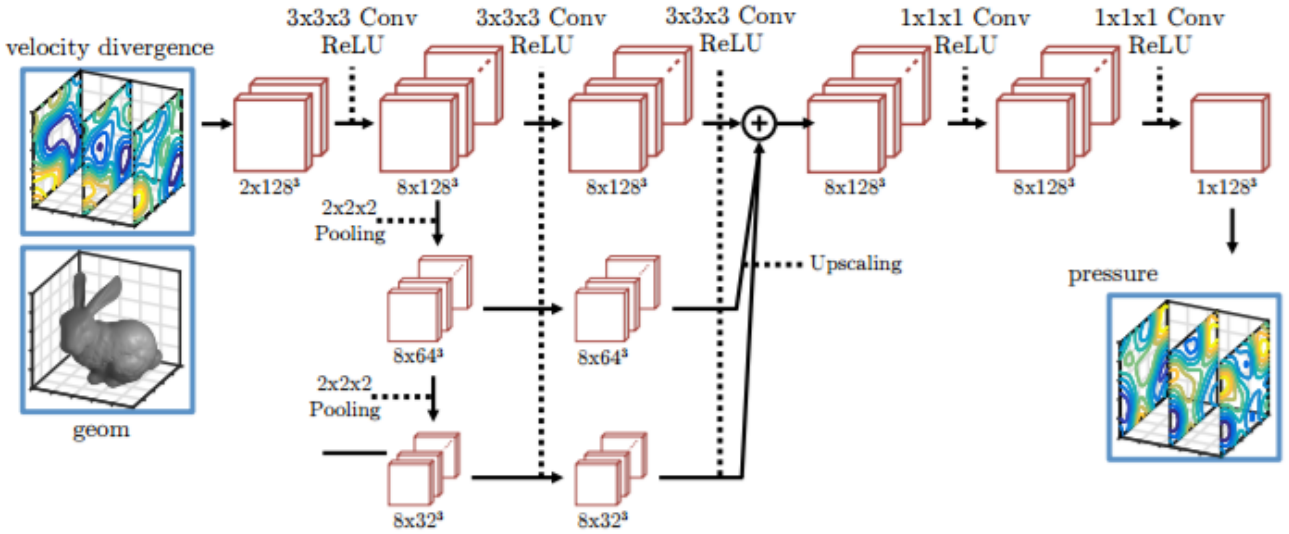


Fig. 4: Convolutional Network for Pressure Solve

From: J. Tompson et al. Accelerating Eulerian Fluid Simulation With Convolutional Networks, https://arxiv.org/abs/1607.03597

“DEEP NEURAL NETS ARE BLACK BOXES”

… even if you can look at the internals…

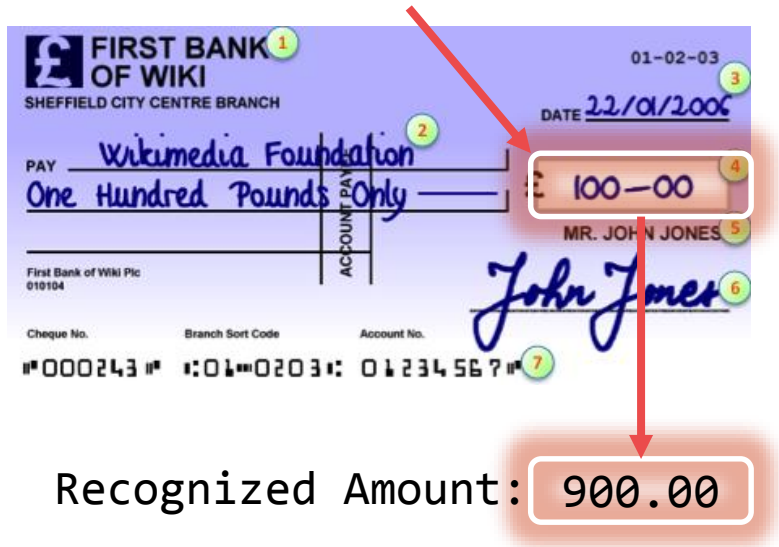
- If a network performs well on the test data and appears to work reasonably well on real data…
  - Can we trust it?
  - Are there formal error bounds on the recognition accuracy?
  - E.g., would you trust a trained NN to operate your nuclear power plant?
  - How to get out learned theory? (e.g. CERN and the Standard Model)
- Field of active research (DARPA, MIT, Capital One, many others)
  - Debugging and Understanding NN behavior
  - Rationales for network decisions

ATTACKING NEURAL NETWORKS

Spoofing and Malicious Misclassification

1. Run input x through the classifier model
2. Derive a perturbation tensor that maximizes chances of misclassification:
    1. Find blind spots in input space; or
    2. Linear perturbation in direction of neural network's cost function gradient; or
    3. Select only input dimensions with high saliency*
3. Apply scaled effective perturbation (δx) to x
    1. Larger perturbation == higher probability for misclassification
    2. Smaller perturbation == less likely for human detection

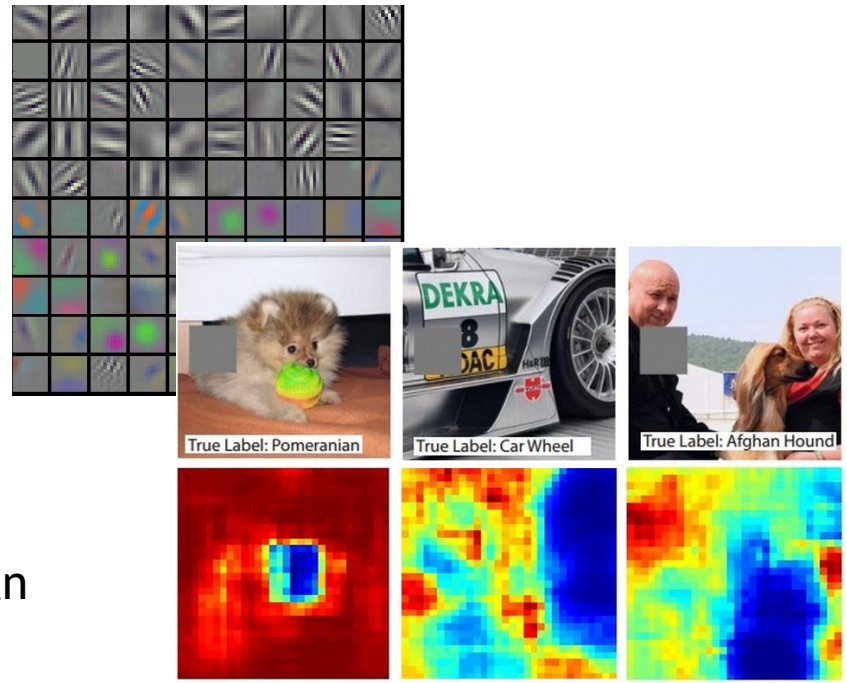
Small Pixel-level Pertubations



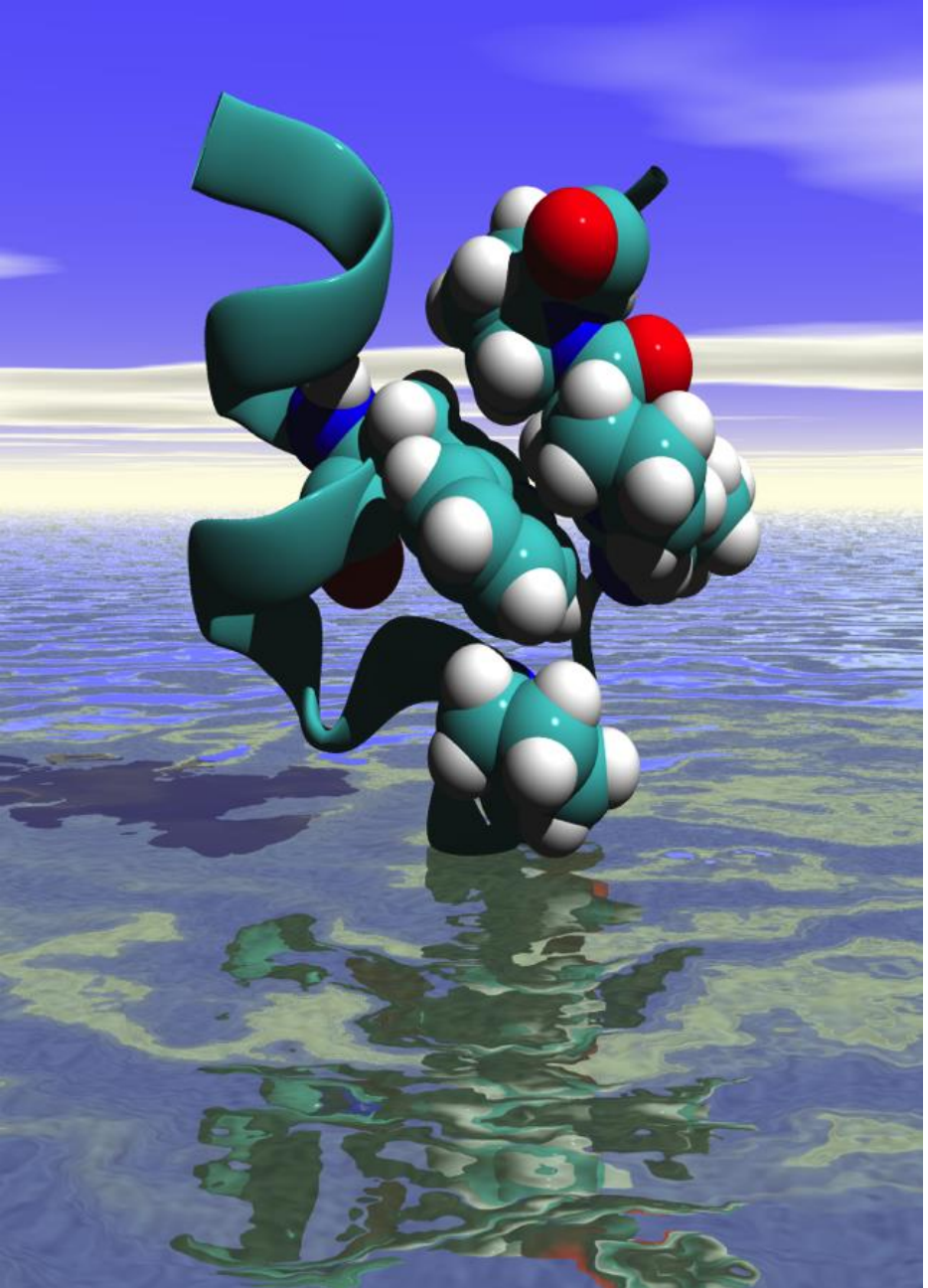
LOOKING INSIDE NEURAL NETS

Debugging, Understanding, Verifying

- Inspecting the NN
  - Visualize activations, filters, generate input that maximizes activation of a neuron
  - Occlude parts of the input and check expectations
  - (e.g., http://cs231n.github.io/understanding-cnn/)
- Capture Model Confidence, Estimate Uncertainty
  - Place Gaussian Distribution over Weights => Bayesian Neural Networks
  - G.Yarin. Uncertainty in Deep Learning, PhD Thesis, University of Cambridge, 2016
- How to gain scientific insight from a trained network?



SOME SUCCESS STORIES



ACCELERATING QUANTUM CHEMISTRY WITH DL FOR DRUG DISCOVERY

Background

Developing a new drug costs $2.5B and takes 10-15 years. Quantum chemistry (QC) simulations are important to accurately screen millions of potential drugs to a few most promising drug candidates.

Challenge

QC simulation is computationally expensive so researchers use approximations, compromising on accuracy. To screen 10M drug candidates, it takes 5 years to compute on CPUs.

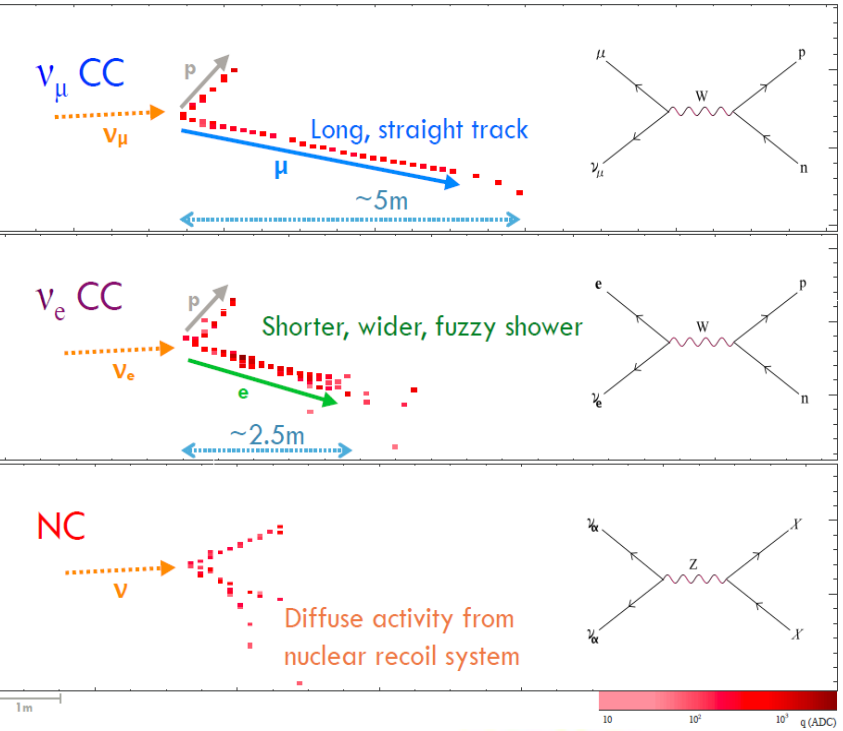
Solution

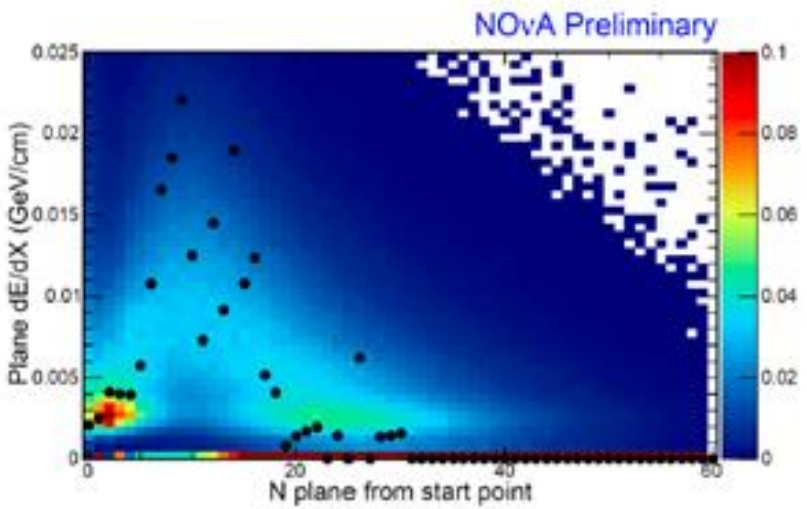
Researchers at the University of Florida and the University of North Carolina leveraged GPU deep learning to develop ANAKIN-ME, to reproduce molecular energy surfaces with 100,000x speedup, extremely high (DFT) accuracy, and at 1-10/millionths of the cost of current computational methods. The new algo can screen 10M drug candidates in 8 minutes.

Impact

Faster, more accurate screening of new drugs to save tons of money.

UF UNIVERSITY of FLORIDA





FINDING THE "GHOST PARTICLE" WITH AI

Background

The NoVA experiment managed by Fermi lab comprises 200 scientists at 40 institutions in 7 countries. The goal is to track neutrino's, which are often referred to as the "Ghost Particle", and detect oscillation which is used to better understand how this super abundant, and elusive particle interacts with matter.

Challenge

The experiment is built underground and is comprised of a main injector beam and two large detector apparatus located 50 miles apart. The near detector is 215 Tons and the Far detector is 15,000 Tons. The experiment can be thought of as a 30 Mn pound detector that takes 2 Mn pictures per second.

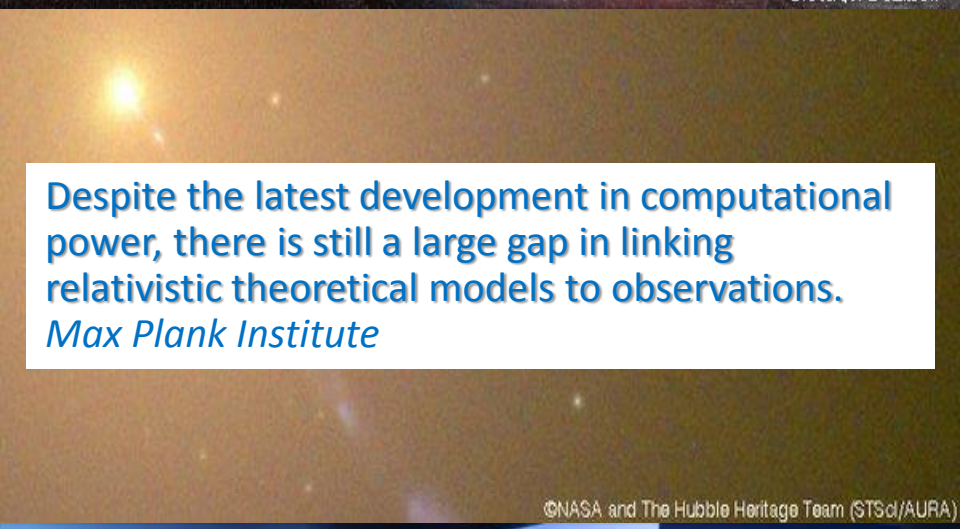
The detectability of the current experiment is proportional to the size of the detectors, so increasing the "visibility" is complex and costly.

Solution

A DNN was developed and trained using a data set derived from multiple HPC simulations including GENIE and GEANT using 2 K40 GPU's. the CVN was basd on convolutional neural networks used for image processing

Impact

The result was an overall improvement of 33%, where the optimized CVN signal-detection-optimized efficiency of 49% is a significant gain over the efficiency of 35% quoted in prior art. This would net to a 10Mn pound increase the physical detector



Despite the latest development in computational power, there is still a large gap in linking relativistic theoretical models to observations.
Max Plank Institute

GRAVITATIONAL ASTROPHYSICS

Background

The LIGO (Advanced Laser Interferometer Gravitational Wave Observatory) experiment successfully discovered signals proving Einstein's theory of General Relativity and the existence of cosmic Gravitational Waves. While this discovery was by itself extraordinary it is seen to be highly desirable to combine multiple observational data sources to obtain a richer understanding of the phenomena.

Challenge

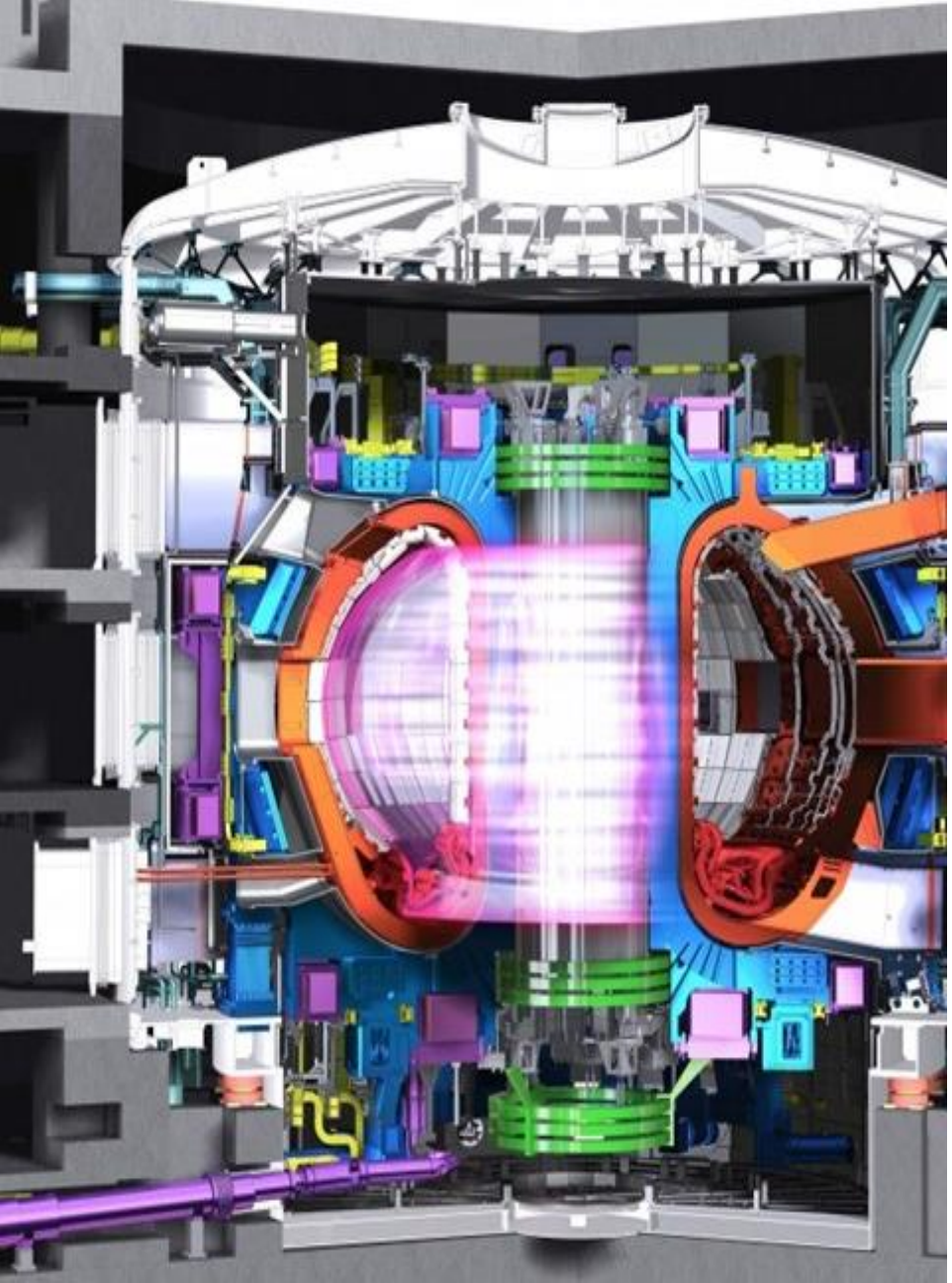
The initial LIGO discoveries were successfully completed using classic data analytics. The processing pipeline used hundreds of CPU's where the bulk of the detection processing was done offline. Here the latency is far outside the range needed to activate resources, such as the Large Synaptic Space survey Telescope (LSST) which observe phenomena in the electromagnetic spectrum in time to "see" what LIGO can "hear".

Solution

A DNN was developed and trained using a data set derived from the CACTUS simulation using the Einstein Toolkit. The DNN was shown to produce better accuracy with latencies 1000x better than the original CPU based waveform detection.

Impact

Faster and more accurate detection of gravitational waves with the potential to steer other observational data sources.



PREDICTING DISRUPTIONS IN FUSION REACTOR USING DL

Background

Grand challenge of fusion energy offers mankind changing opportunity to provide clean, safe energy for millions of years. ITER is a $25B international investment in a fusion reactor.

Challenge

Fusion is highly sensitive, any disruption to conditions can cause reaction to stop suddenly. Challenge is to predict when a disruption will occur to prevent damage to ITER and to steer the reaction to continue to produce power. Traditional simulation and ML approaches don't deliver accurate enough results.

Solution

DL network called FRNN using Theano exceeds today's best accuracy results. It scales to 200 Tesla K20s, and with more GPUs, can deliver higher accuracy. Goal is to reach 95% accuracy.

Impact

Vision is to operate ITER with FRNN, operating and steering experiments in real-time to minimize damage and down-time.

PRINCETON UNIVERSITY

Axel Koehler (akoehler@nvidia.com)