Machine Learning, Datascience and Neutrino Physics at Argonne's Leadership Computing Facility

Corey Adams
Data Science Group
Argonne Leadership Computing Facility
corey.adams@anl.gov

* With content from the Datascience Team at ALCF, including Venkat Viswanath, Rick Zamora, Huihuo Zheng

Outline

- Argonne Leadership Computing Facility
- Datascience (And machine learning) at ALCF
- Scaling Machine Learning for High Performance Computing
- Deep Learning for Neutrino Physics
- How to get HPC resources at ALCF – we want you to use these resources!

ALCF

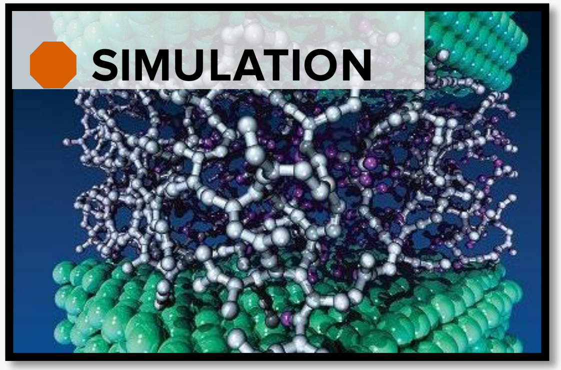
Argonne Leadership Computing Facility

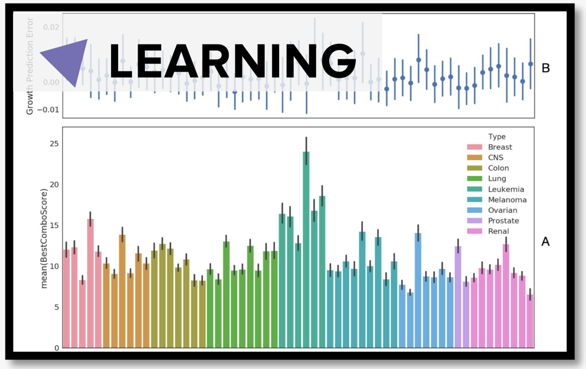
The Argonne Leadership Computing Facility provides world-class computing resources to the scientific community.

- Users pursue scientific challenges
- Resources fully dedicated to open science
- In-house experts to help maximize results

Theta Intel/Cray

4,392 nodes
281,088 cores
69 TiB MCDRAM
824 TiB DDR4
549 TB SSD
Peak flop rate: 11.69 PF





Mira IBM BG/Q
49,152 nodes
786,432 cores
786 TB RAM
Peak flop rate: 10 PF

ALCF offers different pipelines based on your computational readiness. Apply to the allocation program that fits your needs.

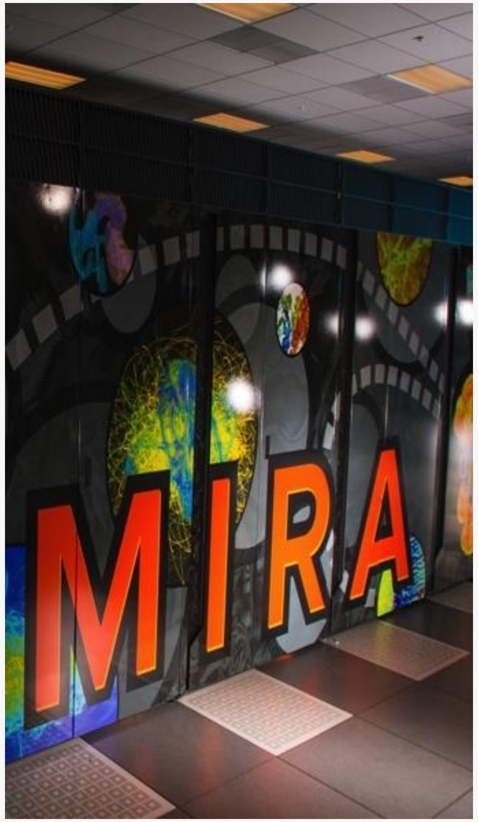
https://www.alcf.anl.gov

Argonne NATIONAL LABORATORY

Argonne's path to Exascale is critical to our nation's scientific leadership

2007

557 TeraFLOPS



2013

10 PetaFLOPS

2017

11 PetaFLOPS

2021

ALCF4

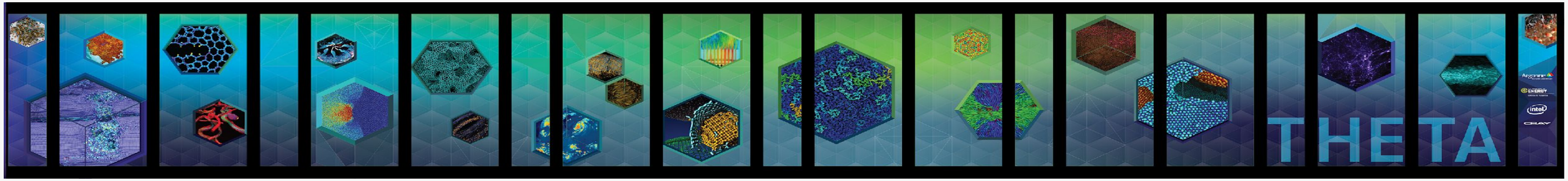
Argonne NATIONAL LABORATORY

Computing Resources
Focus on Theta

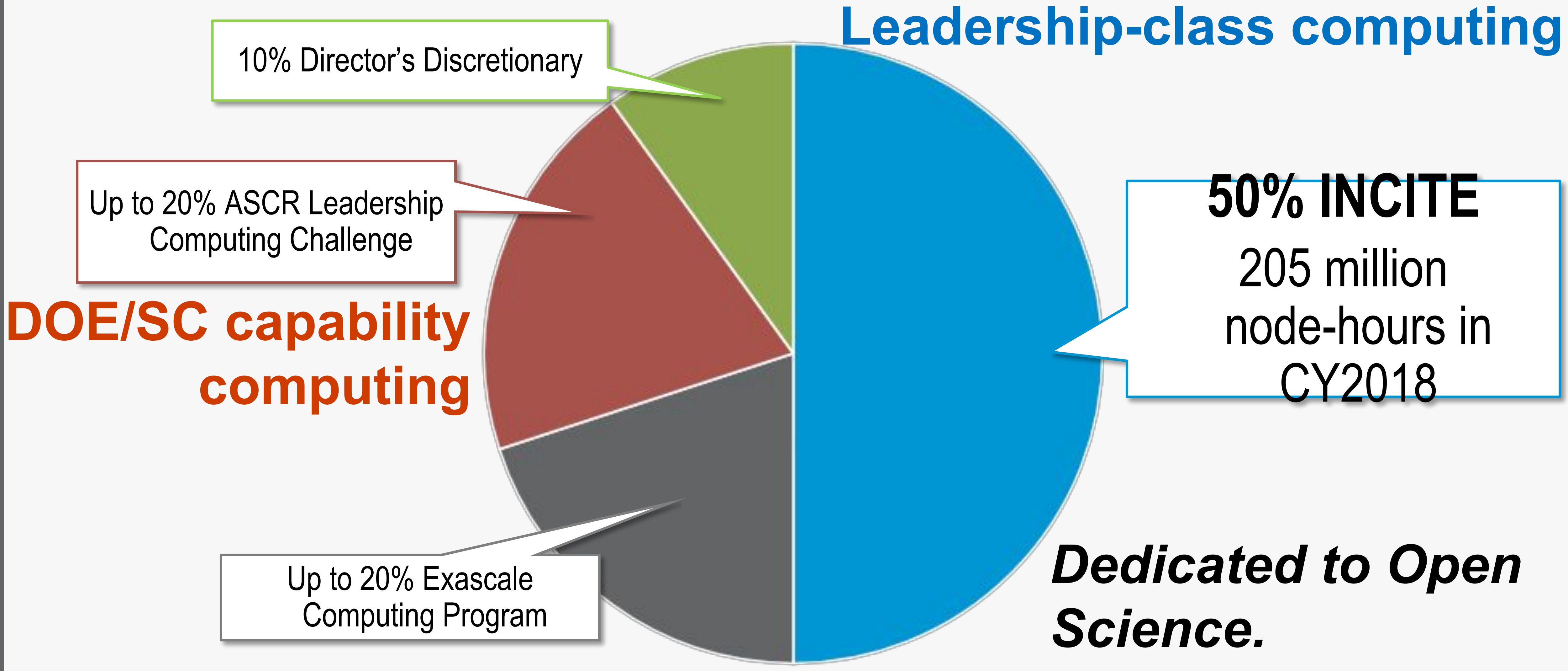
Theta (Cray XC40)

- **11.69 PF** Peak performance
- **4,392 nodes** (281,088 cores)
    - 2nd Generation Intel® Xeon Phi™ Processor (Knights Landing)
- **843.264 TB DDR4** and **70.272 TB MCDRAM** total memory
- **128 GB SSD** on each node
- Cray Aries high speed interconnect in dragonfly topology
- **10 PB** Lustre file system, **~200 GB/s** throughput

Also available: Mira, as well as other visualization and analytics clusters, and storage systems



Who Uses ALCF? Allocation Distributions



ALCF Data Science Program

ALCF Data Science Program (ADSP) Overview

- Big Data science problems that require the leadership scale and performance
- Span computational, experimental and observational sciences
- Focus on data science techniques including but not limited to statistics, machine learning, deep learning, UQ, image processing, graph analytics, complex and interactive workflows
- Two-year proposal period and will be renewed annually. Proposals will target science and software technology scaling for data science
- The program started in 2016 and now in the 3rd year
- Yearly call for proposal.
  Next deadline – ~June 2019
- https://www.alcf.anl.gov/alcf-data-science-program

Argonne NATIONAL LABORATORY

ALCF Data Science Program (ADSP)
Targets Data & Learning Pillars

Data

- Experimental/observational data
  - Image analysis
  - Multidimensional structure discovery
- Complex and interactive workflows
- On-demand HPC
- Persistent data techniques
  - Object store
  - Databases
- Streaming/real-time data
- Uncertainty quantification
- Statistical methods
- Graph analytics

Learning

- Deep learning
- Machine learning steering simulations
  - Parameter scans
  - Materials design
  - Observational signatures
- Data-driven models and refinement for science using ML/DL
- Hyperparameter optimization
- Pattern recognition
- Bridging gaps in theory

Big Data

datascience@alcf.anl.gov


Corey Adams


Prasanna Balaprakash


Taylor Childers


Murali Emani


Elise Jennings

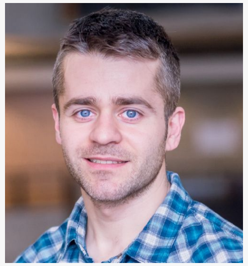

Xiao-Yong Jin


Murat Keceli


Bethany Lusch


Alvaro Vazquez

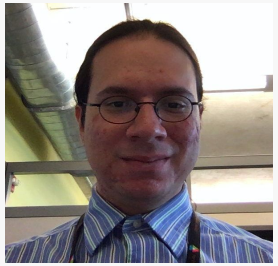

Adrian Pope


Misha Salim


William Scullin


Ganesh Sivaraman


Tom Uram

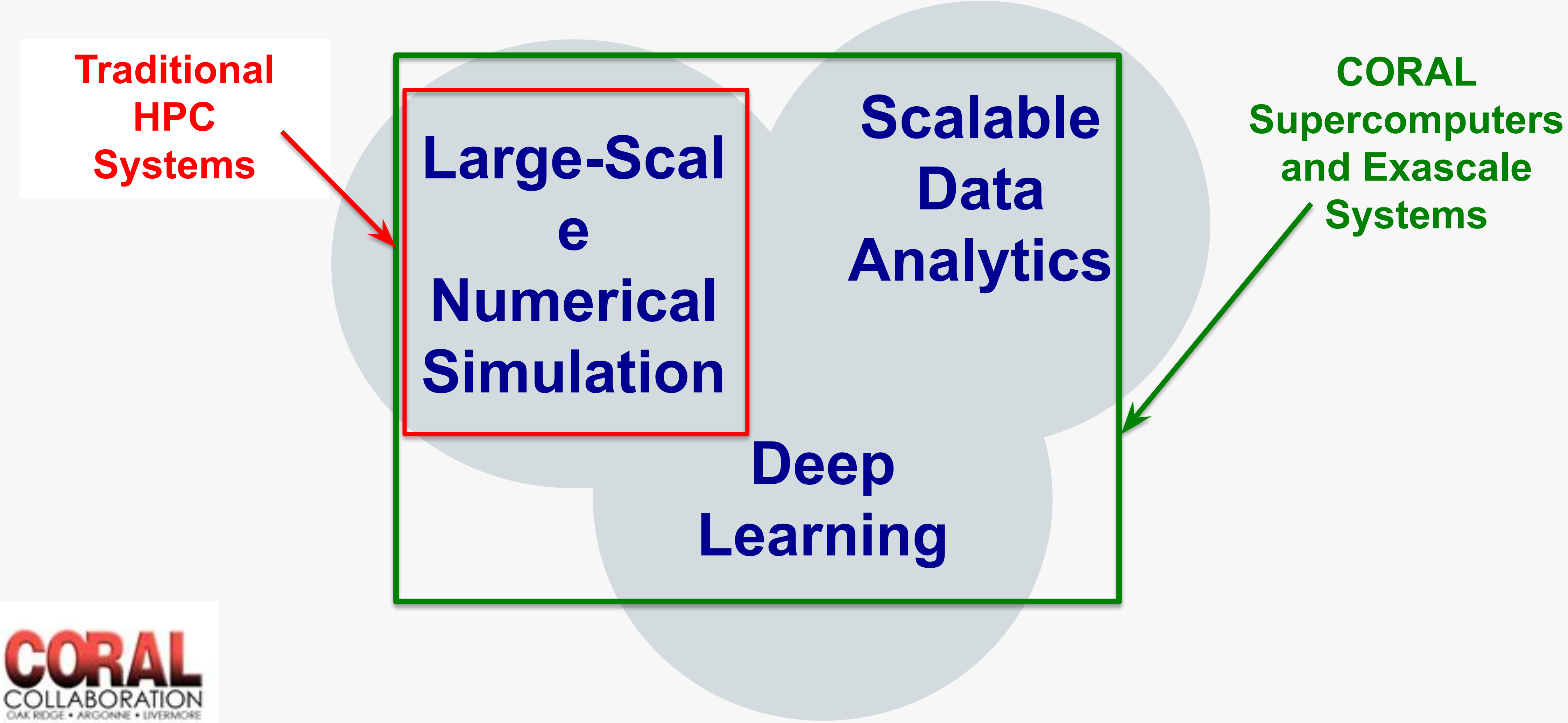

Antonio Villarreal


Venkat Vishwanath

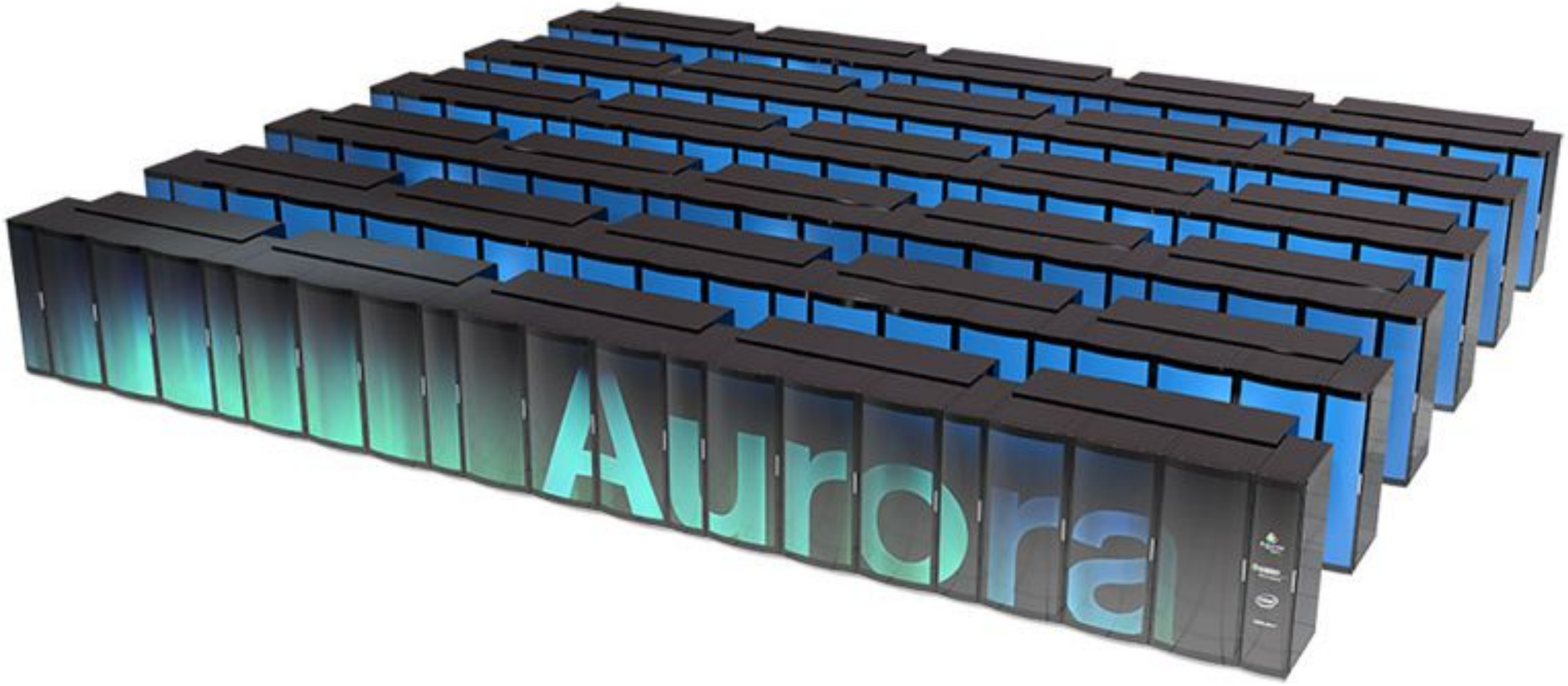

Richard Zamora


Huihuo Zheng

Integration of Simulation, Data Analytics and Machine Learning on supercomputers



Aurora 2021 - The first US Exascale System



Architecture supports three ways of computing

- **Large-scale Simulation (PDEs, traditional HPC)**
- **Data Intensive Applications (scalable science pipelines)**
- **Deep Learning and Emerging Science AI (training and inferencing)**

Machine Learning and High Performance Computing

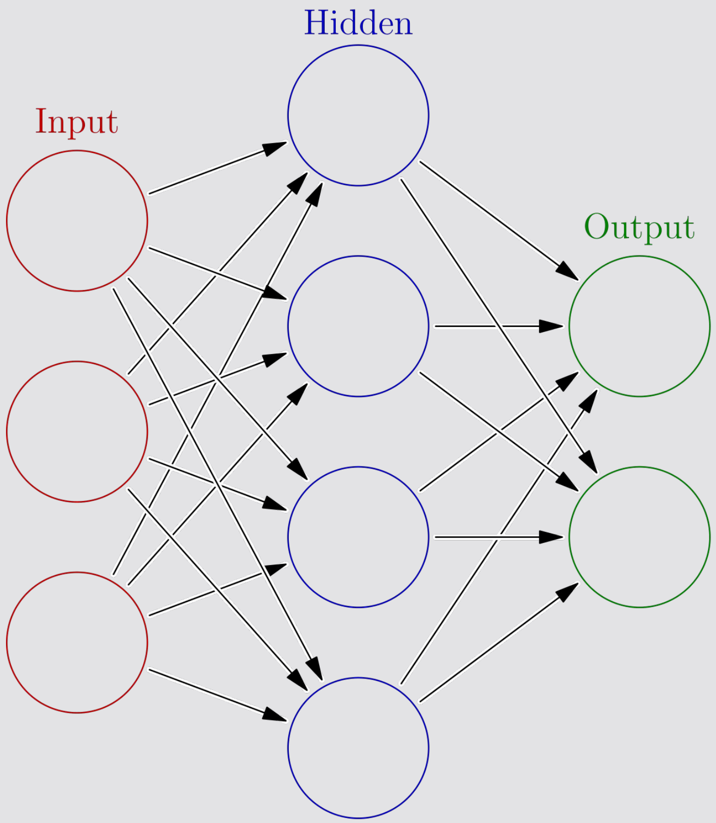
Machine Learning and HPC

Accelerate and improve an application's:

Time to Solution (Training) – with scalable learning techniques, you can process more images per second, reduce the time per epoch, and reach a trained network faster.

Quality of Solution – with more compute resources available, you can perform hyperparameter searches to optimize network designs and training schemes. With powerful accelerators, you can train bigger and more computationally intense networks.

Inference Throughput – with high bandwidth IO, it is easy to scale up the throughput of inference techniques for deep learning.



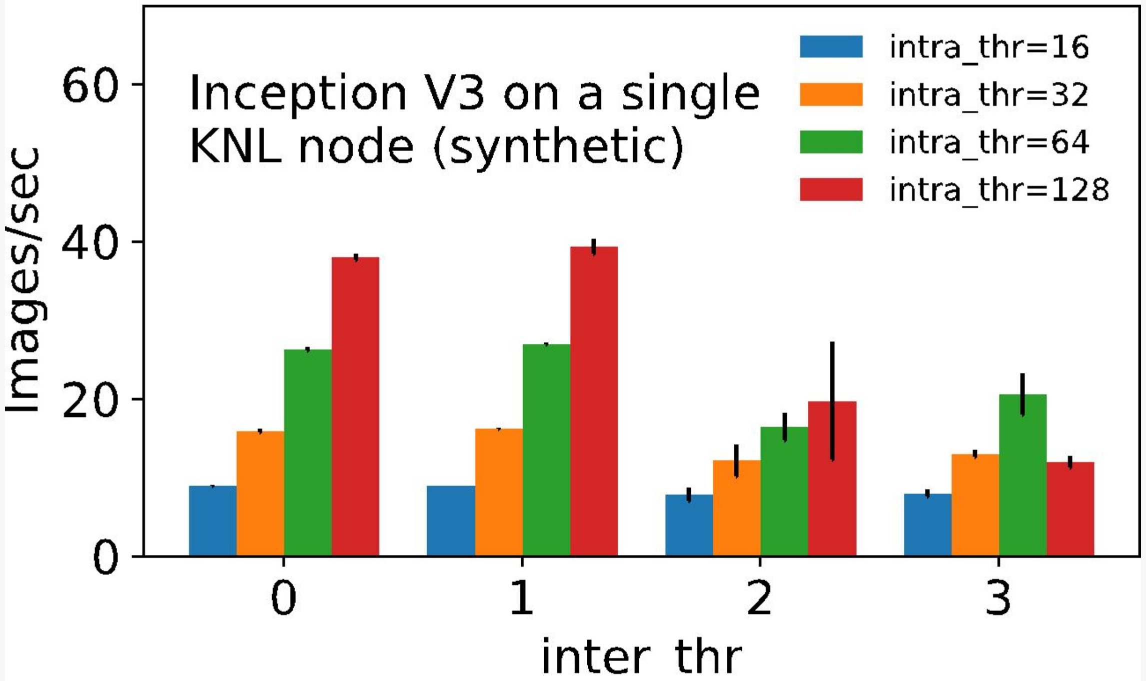
High Performance Computing can improve all aspects of training and inference in machine learning.

Machine Learning and HPC – KNL Nodes
Single Node Performance Matters!

Running a model on an HPC node is often not like a standard GPU – many configuration parameters matter, not all models have the same optimum parameters.

Intel KNL != Nvidia GPU, but KNL can be powerful.

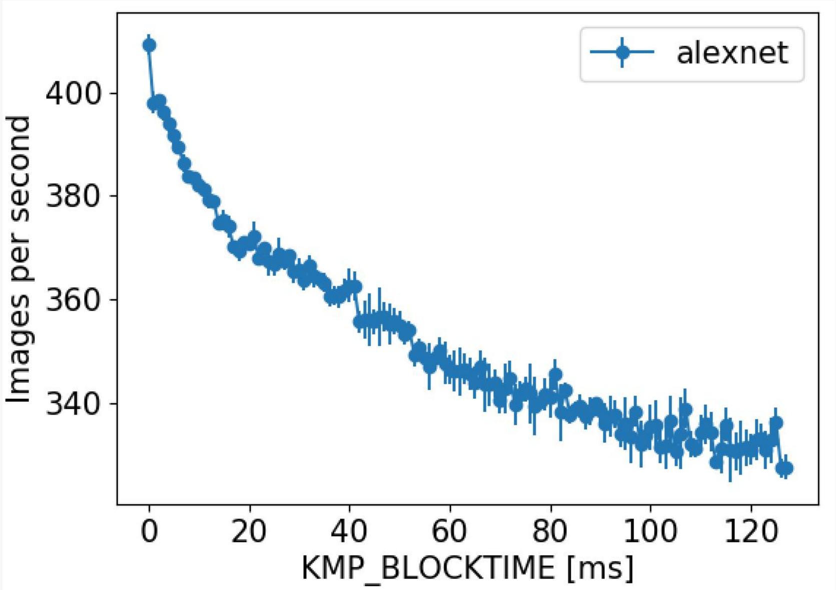
intra_op_parallelism_threads: Nodes that can use multiple threads to parallelize their execution will schedule the individual pieces into this pool.
inter_op_parallelism_threads: All ready nodes are scheduled in this pool.

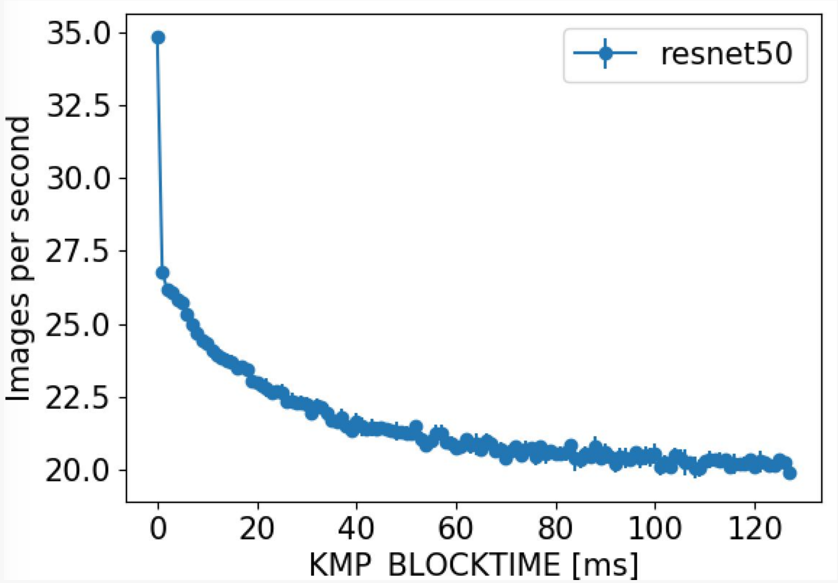


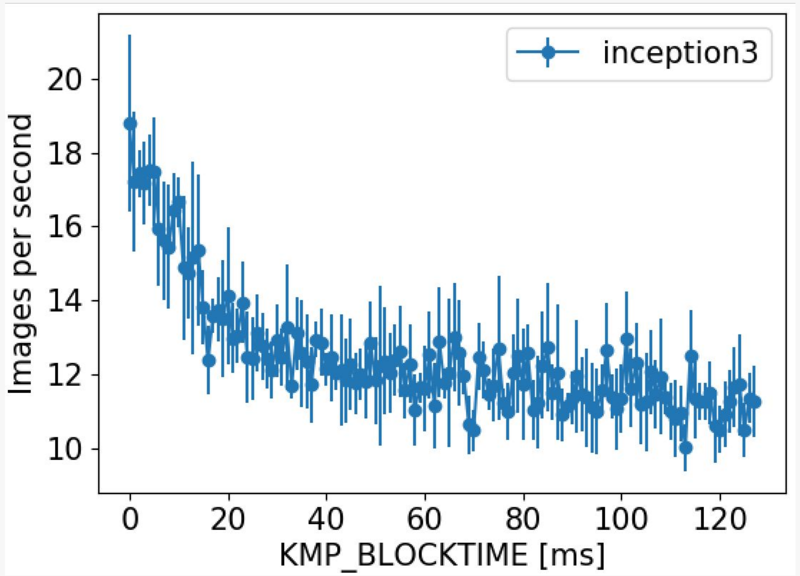

```
config = tf.ConfigProto()
config.intra_op_parallelism_threads = num_intra_threads
config.inter_op_parallelism_threads = num_inter_threads
tf.Session(config=config)
```

https://www.tensorflow.org/guide/performance/overview

Machine Learning and HPC – KNL Nodes
Affinity can play a large role







KMP_BLOCKTIME=0 is optimal for KNL Nodes

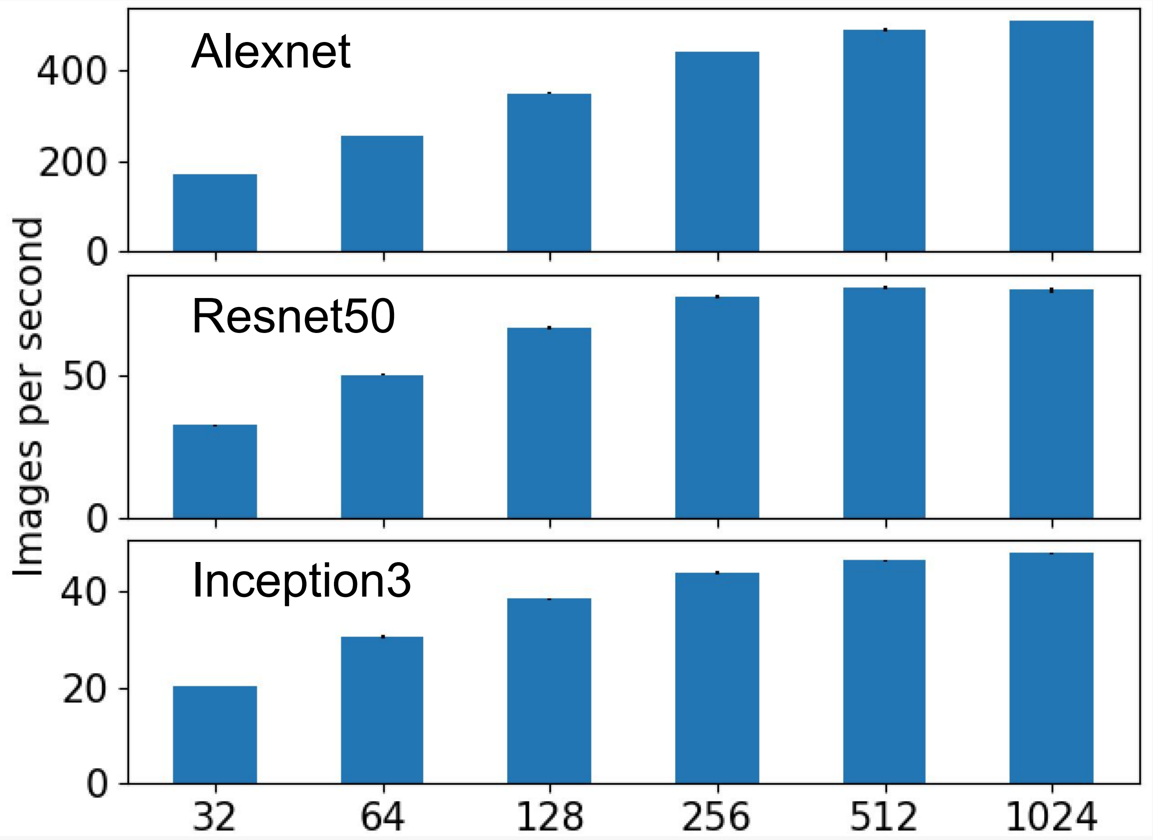
KMP_AFFINITY=granularity=fine,verbose,compact,1,0

Intel Affinity Guidelines

Machine Learning and HPC – KNL Nodes
Batch Size can be important

Running a model on an HPC node is often not like a standard GPU – many configuration parameters matter, not all models have the same optimum parameters.

Bigger batch size often yields more images/second throughput (though not always), but the downside is always more seconds/global step at a large batch size.



Batch Size Dependence

Distributed Learning

Machine learning is a very important workflow for current and future supercomputing systems.

How can you accelerate learning with more computing power?

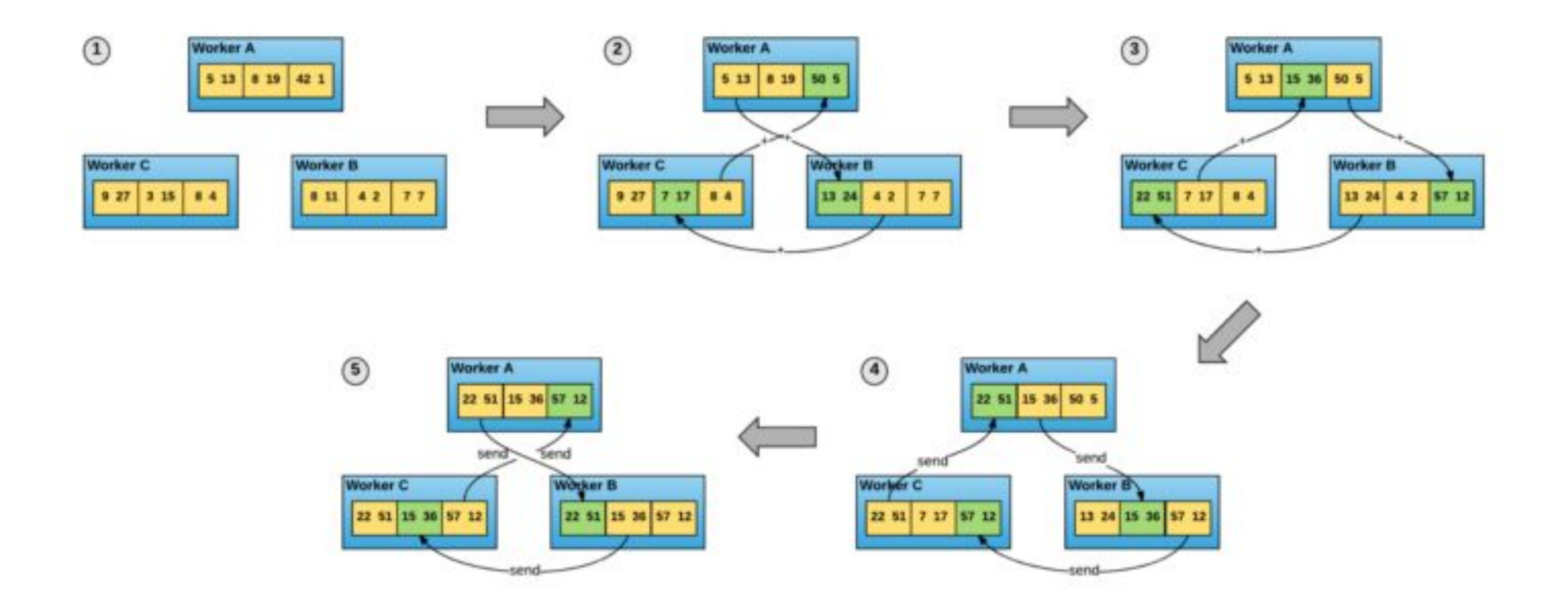


Image from Uber's Horovod: https://eng.uber.com/horovod/

What is Distributed Learning?
A technique to accelerate training

The backpropagation algorithm is unchanged at it's heart.

Data Parallel learning – with N nodes, replicate your model on each node. After the forward and backward computations, **average** the gradients across all nodes and use the averaged gradients to update the weights. Conceptually, **this multiplies the minibatch size by N.**

Model Parallel Learning – for models that don't fit on a single node, you can divide a single model across multiple locations. The design of distributing a model is not trivial, but tools are emerging.

Both ("Mesh" training) – Using n nodes for a single model, and N = k*n nodes for distributed training, you can achieve accelerated training of extremely large or expensive models.

Data Parallel Learning

All nodes communicate to average gradients.

Each Model gets unique input data and performs calculations independently.

Each Node gets it’s own copy of the model.

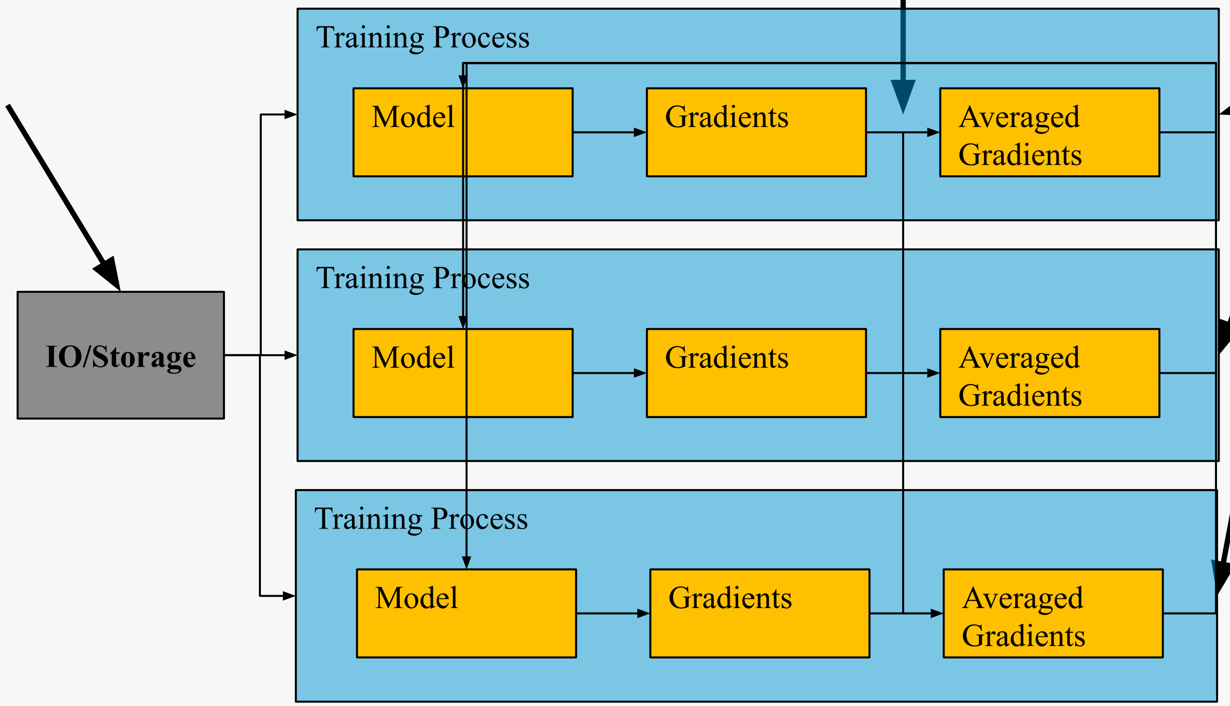


Image from Uber’s Horovod: https://eng.uber.com/horovod/

Data Parallel Learning

Scaling Challenges

All nodes communicate to average gradients.

Computation stalls during communication: keeping the communication to computation ratio small is important for scaling.

Each Model gets unique input data and performs calculations independently.

IO requires organization to ensure unique batches.

IO contention with many nodes requires parallel IO solutions

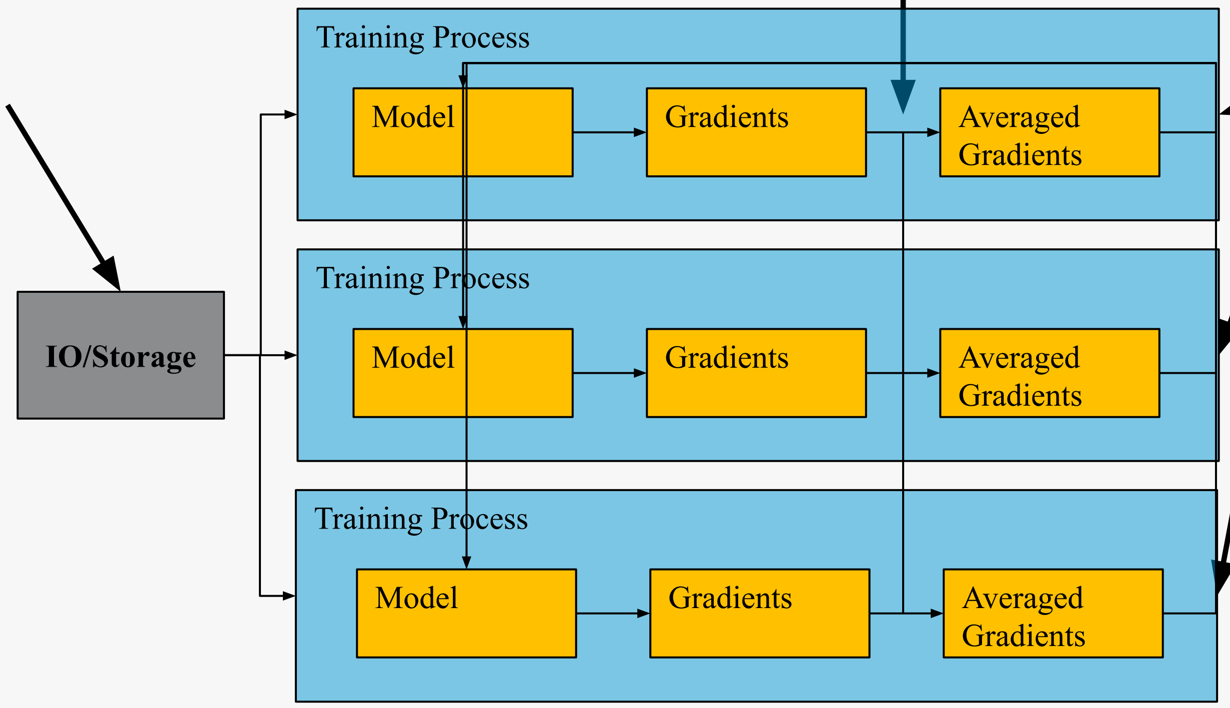


Image from Uber's Horovod: https://eng.uber.com/horovod/

Each Node gets it's own copy of the model.

Initialization must be identical or synchronized, and checkpointing/summary information must be managed with just one node.

Data Parallel Learning

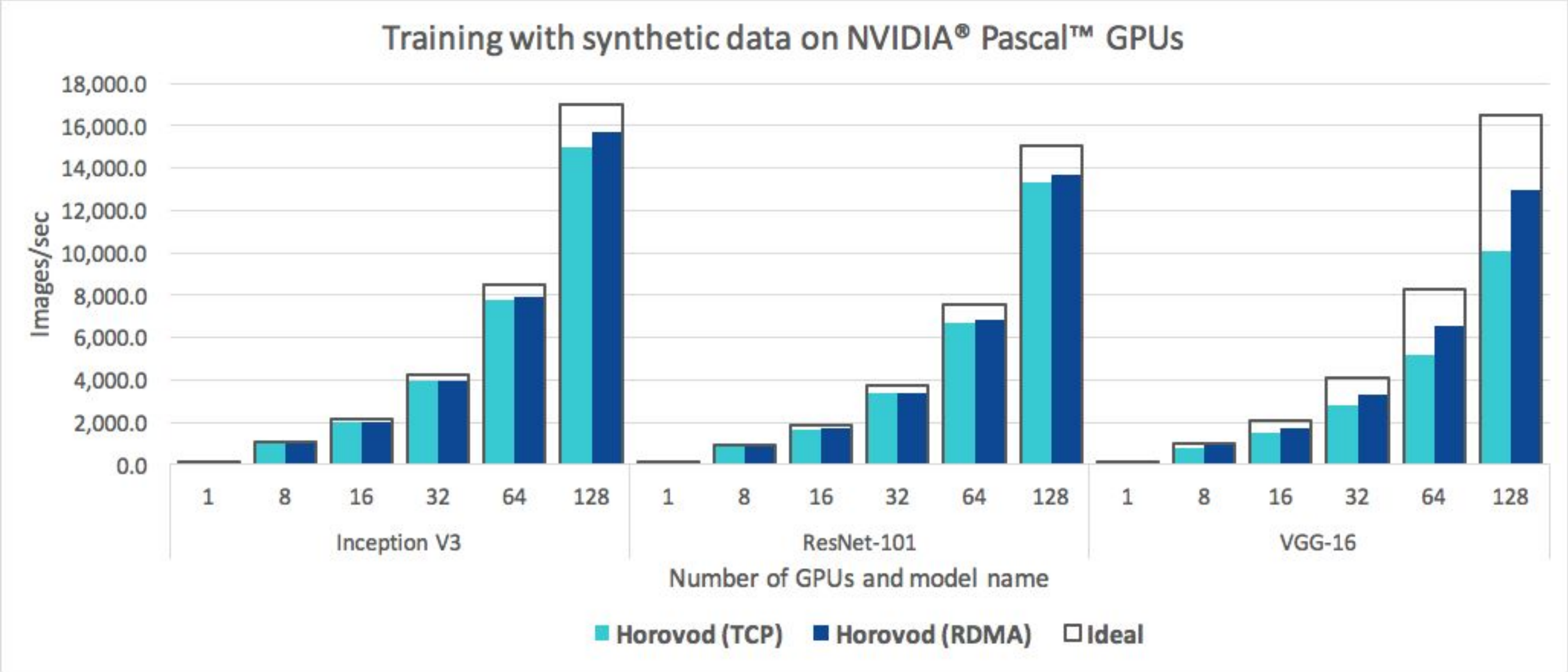


Image from Uber's Horovod: https://eng.uber.com/horovod/

Data Parallel Learning
Horovod

The simplest technique for data parallel learning

1. Initialize horovod (hvd.init()).
2. Wrap the optimizer in hvd.DistributedOptimizer.
    1. This uses the underlying optimizer for gradient calculations, and performs an averaging of all gradients before updating.
    2. Can adjust the learning rate to account for a bigger batch size.
3. Initialize the networks identically, or broadcast one network's weights to all others.
4. Ensure snapshots and summaries are only produced by one rank.

Horovod focuses on handling collective communication so you don't have to, but let's you use all of the tools of your favorite framework. Compatible with mpi4py.

HOROVOD

Horovod is an open source data parallel training software compatible with many common deep learning frameworks.

Meet Horovod
Github

Horovod Example Code

Tensorflow

```
import tensorflow as tf
import horovod.tensorflow as hvd
layers = tf.contrib.layers
learn = tf.contrib.learn
def main():
    # Horovod: initialize Horovod.
    hvd.init()
    # Download and load MNIST dataset.
    mnist = learn.datasets.mnist.read_data_sets('MNIST-data-%d' % hvd.rank())
    # Horovod: adjust learning rate based on number of GPUs.
    opt = tf.train.RMSPropOptimizer(0.001 * hvd.size())
    # Horovod: add Horovod Distributed Optimizer
    opt = hvd.DistributedOptimizer(opt)
    hooks = [
        hvd.BroadcastGlobalVariablesHook(0),
        tf.train.StopAtStepHook(last_step=20000 // hvd.size()),
        tf.train.LoggingTensorHook(tensors={'step': global_step, 'loss': loss},
                                   every_n_iter=10),
    ]
    checkpoint_dir = './checkpoints' if hvd.rank() == 0 else None
    with tf.train.MonitoredTrainingSession(checkpoint_dir=checkpoint_dir,
                                           hooks=hooks,
                                           config=config) as mon_sess
```

Horovod Example Code

Pytorch

```
import torch.nn as nn
import horovod.torch as hvd
hvd.init()
train_dataset = datasets.MNIST('data-%d' % hvd.rank(), train=True, download=True,
                    transform=transforms.Compose([
                        transforms.ToTensor(),
                        transforms.Normalize((0.1307,), (0.3081,))
                    ]))
train_sampler = torch.utils.data.distributed.DistributedSampler(
    train_dataset, num_replicas=hvd.size(), rank=hvd.rank())
train_loader = torch.utils.data.DataLoader(
    train_dataset, batch_size=args.batch_size, sampler=train_sampler, **kwargs)
# Horovod: broadcast parameters.
hvd.broadcast_parameters(model.state_dict(), root_rank=0)
# Horovod: scale learning rate by the number of GPUs.
optimizer = optim.SGD(model.parameters(), lr=args.lr * hvd.size(),
                      momentum=args.momentum)!
# Horovod: wrap optimizer with DistributedOptimizer.
optimizer = hvd.DistributedOptimizer(optimizer, named_parameters=model.named_parameters())
```

Horovod Example Code

Keras

```
import keras
import tensorflow as tf
import horovod.keras as hvd
# Horovod: initialize Horovod.
hvd.init()
# Horovod: adjust learning rate based on number of GPUs.
opt = keras.optimizers.Adadelta(1.0 * hvd.size())
# Horovod: add Horovod Distributed Optimizer.
opt = hvd.DistributedOptimizer(opt)
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=opt,
              metrics=['accuracy'])
callbacks = [
    # Horovod: broadcast initial variable states from rank 0 to all other processes.
    hvd.callbacks.BroadcastGlobalVariablesCallback(0),
]
# Horovod: save checkpoints only on worker 0 to prevent other workers from corrupting them.
if hvd.rank() == 0:
    callbacks.append(keras.callbacks.ModelCheckpoint('./checkpoint-{epoch}.h5’))
model.fit(x_train, y_train, batch_size=batch_size,
    callbacks=callbacks,
    epochs=epochs,
    verbose=1, validation_data=(x_test, y_test))
```

Effects of Distributed Learning

1. Increased Batch size means improved estimate of gradients.
   1. Scale by N nodes? Sqrt(N)?
   2. Scale in a layerwise way? Layerwise Adaptive Rate Scaling (LARS)
2. Increased learning rate can require warm up iterations.
   1. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour
3. Bigger minibatch means less iterations for the same number of epochs.
   1. May need to train for more epochs if another change is not made like boosting the learning rate.

Mesh Learning
Tensorflow Mesh

When data-parallel isn't enough…

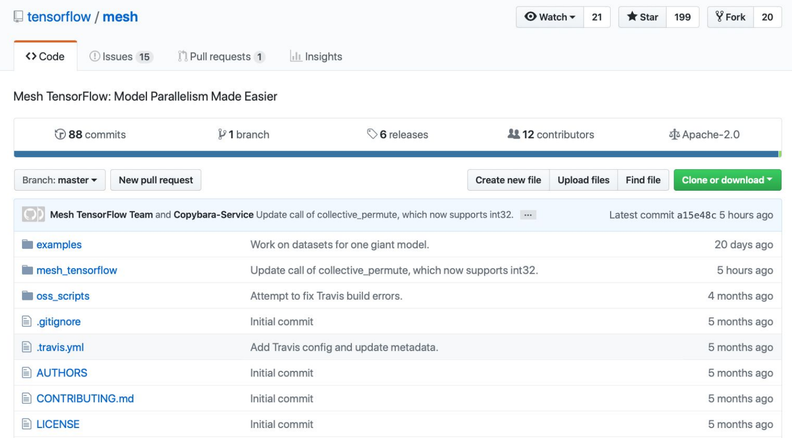
Why might you need a Mesh?

- Memory limitations due to network size (number of parameters)
- Memory limitations due to input size (massive images, etc)

Mesh Scaling is not trivial:

- Computations need to be distributed in an intelligent way to prevent idle nodes
- Communication needs to happen frequently during both the forward/backward pass
- Message passing organization details arise from forward/backward small-group communications and multi-group communications

Expect mesh scaling to get easier over the next few years (or wait for bigger, more powerful nodes!)



https://github.com/tensorflow/mesh

Neutrino Physics @ ALCF

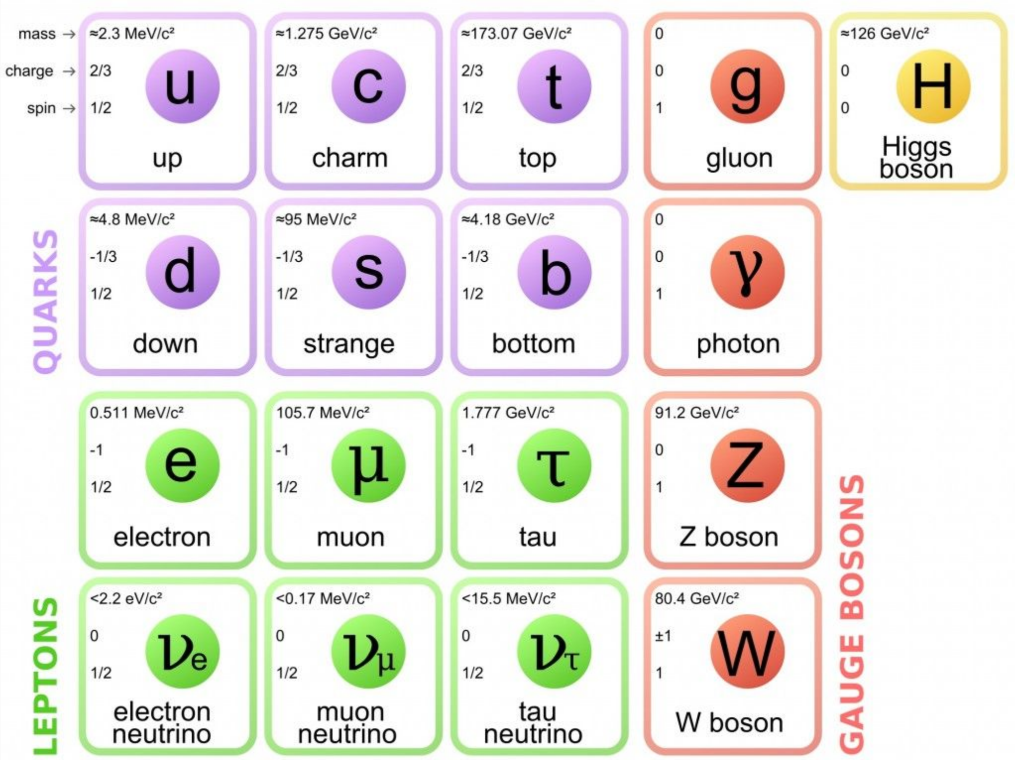
A Crash Course in Neutrino Physics

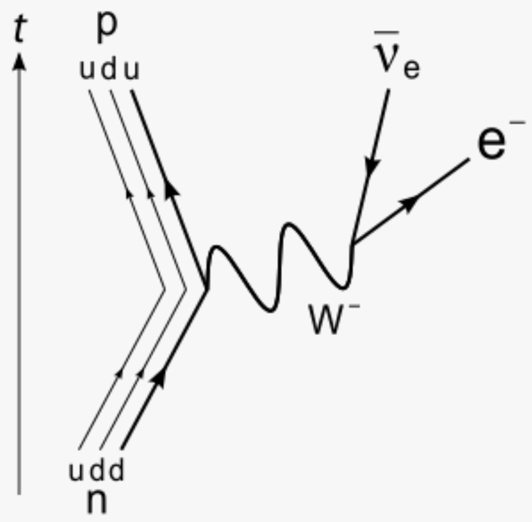
Particle Physics - Building Blocks

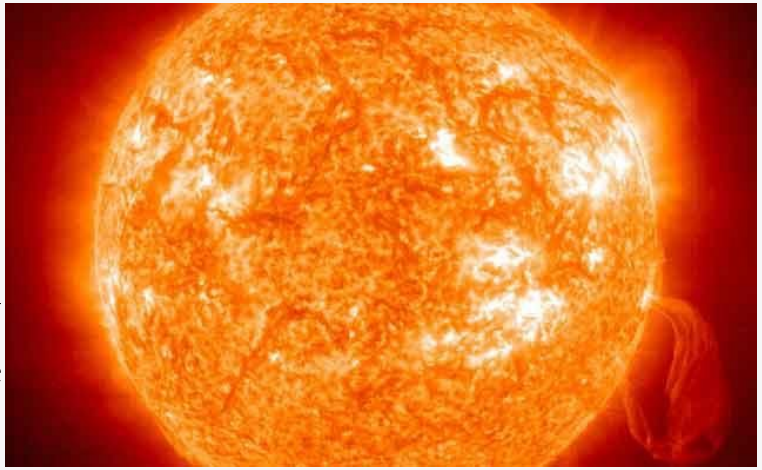
"Sewing the fabric of spacetime"

Everything we know about is made from these particles

Particle Interactions







Particle Interactions are really just
mediating particle

Argonne NATIONAL LABORATORY

Basic Rules

Fundamental Particles follow basic rules:

1. Anything not forbidden is allowed.
2. Things that are forbidden are forbidden by "conservation" laws.
    1. Traditional conservation laws: conservation of momentum, energy, charge, etc...
    2. Particle Physics conservation laws: conservation of "lepton" number, "baryon" number, "flavor", "color", ...

Neutrinos are one of the biggest rule breakers in the Universe.

Neutrino Oscillations

Neutrinos are super weird!

Over time, an electron stays as an electron.

An electron neutrino can turn into a muon neutrino

And back!

Measuring the frequency and strength of the "oscillations" tells a LOT about neutrinos.

Why Study Neutrinos?

Are neutrinos their own antiparticles?

Do neutrinos explain the matter/antimatter asymmetry in the Universe?

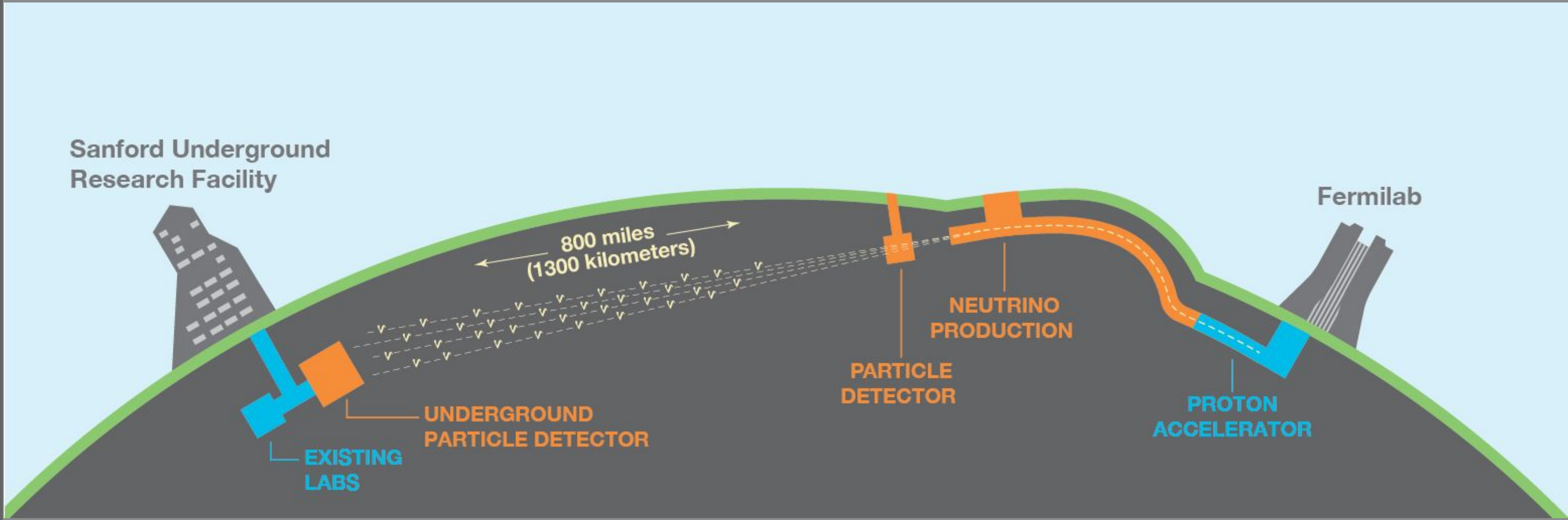
Are there more than 3 types of neutrinos?

Could new type of neutrinos explain dark matter?

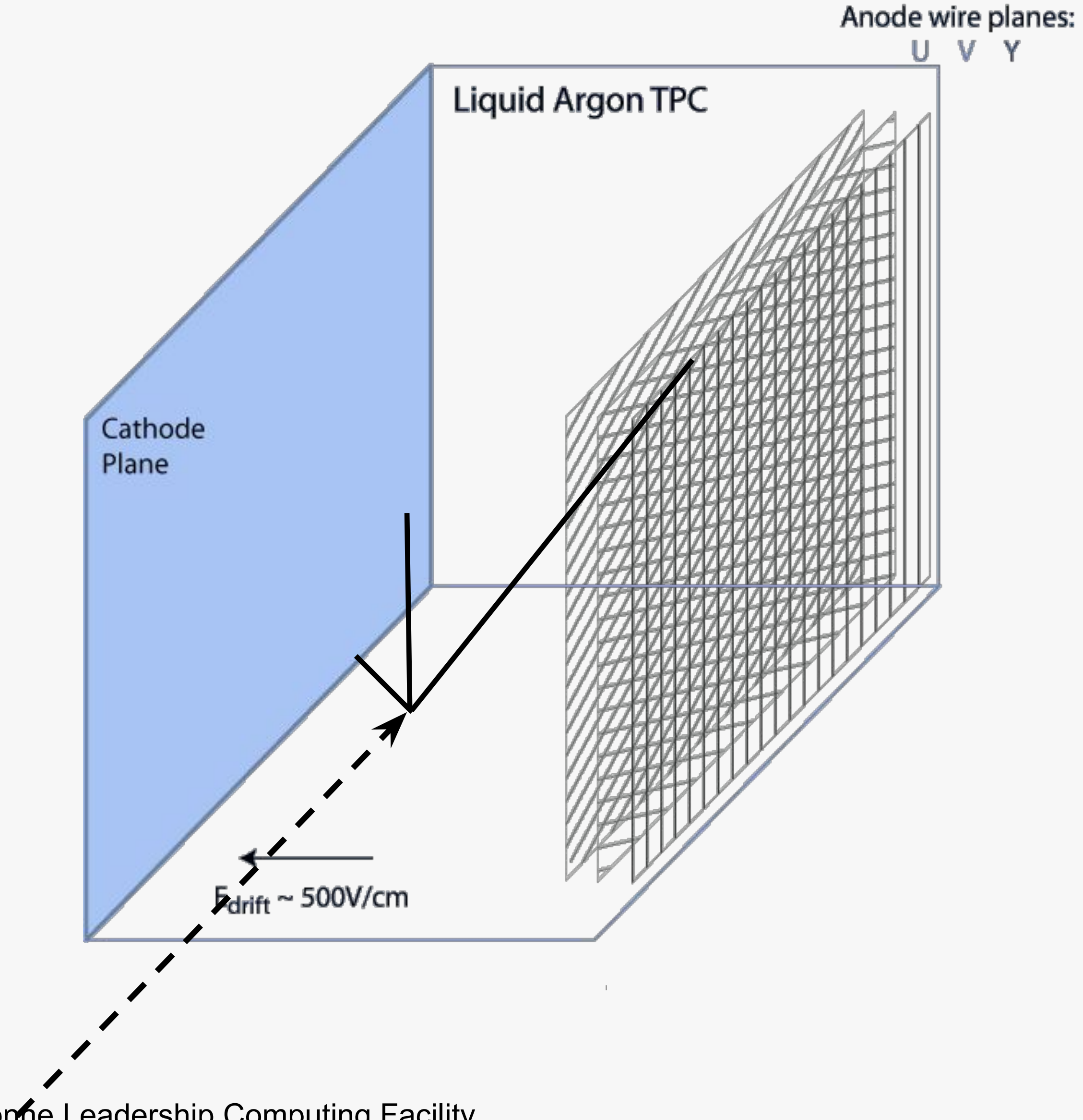
Why is the neutrino mass so much less than other particles?

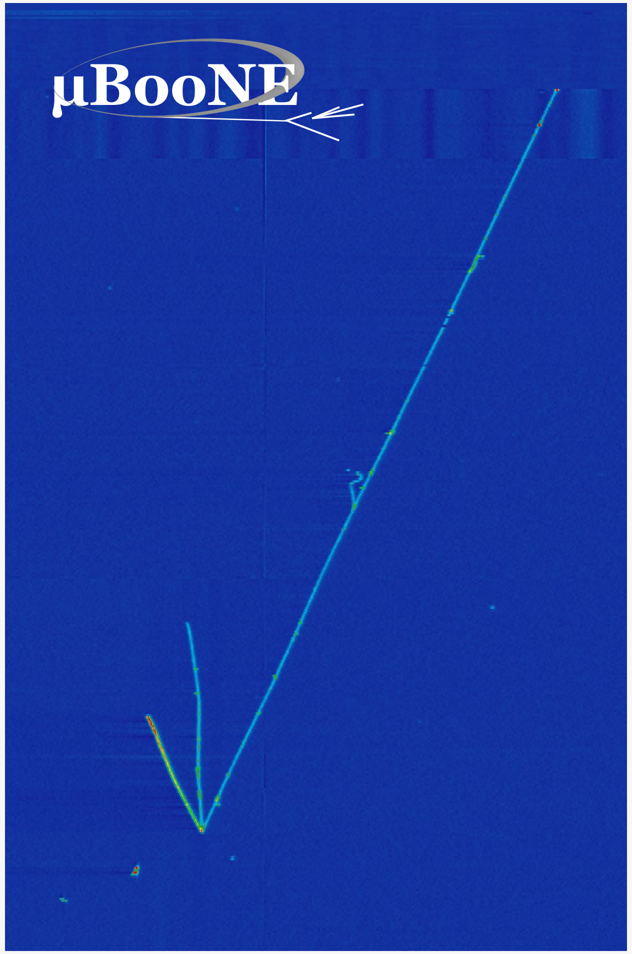
Liquid Argon Time Projection Chambers

The modern neutrino detector of choice for high energy physics is the **Liquid Argon Time Projection Chamber.**

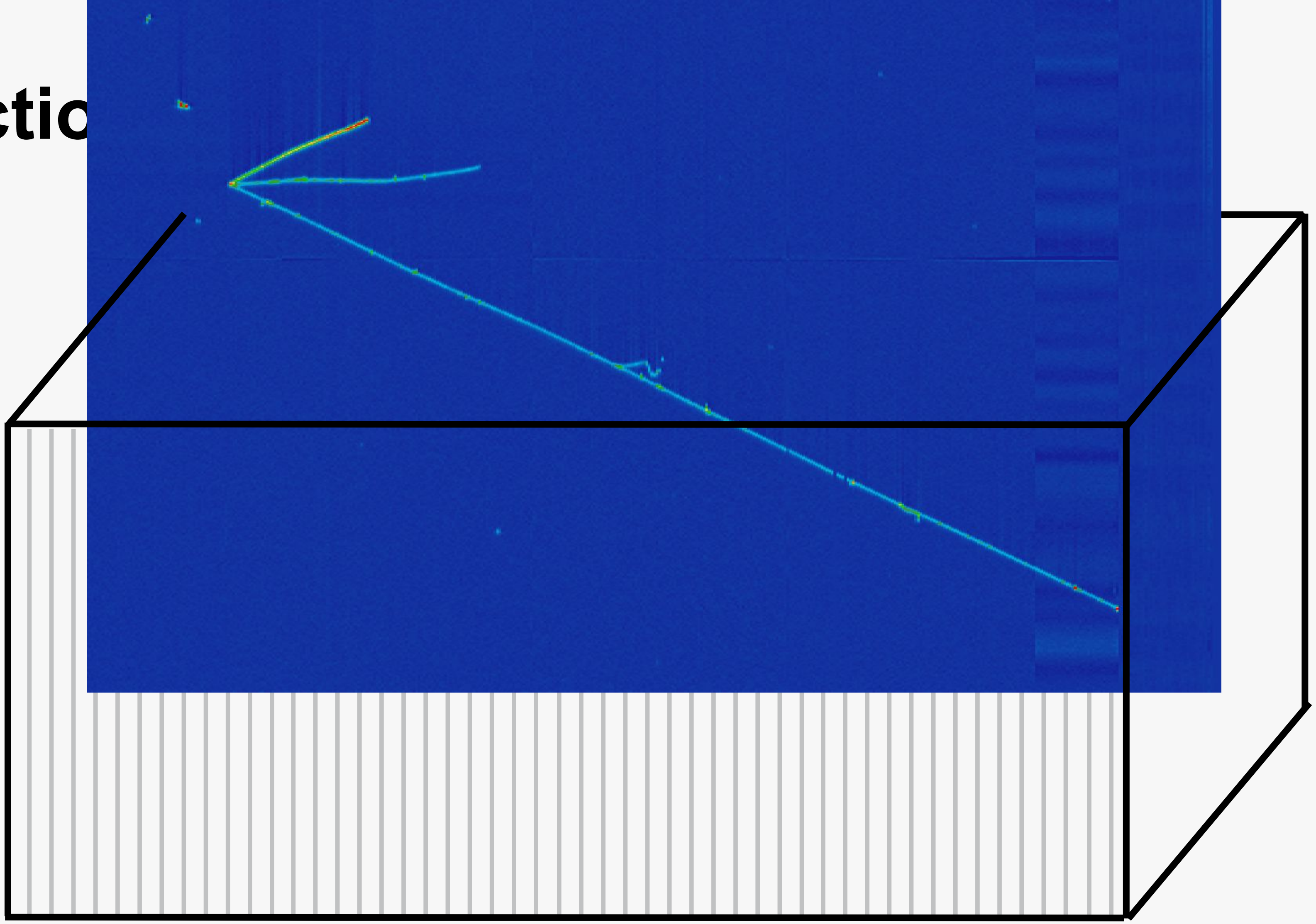


Liquid Argon Time Projection Chambers

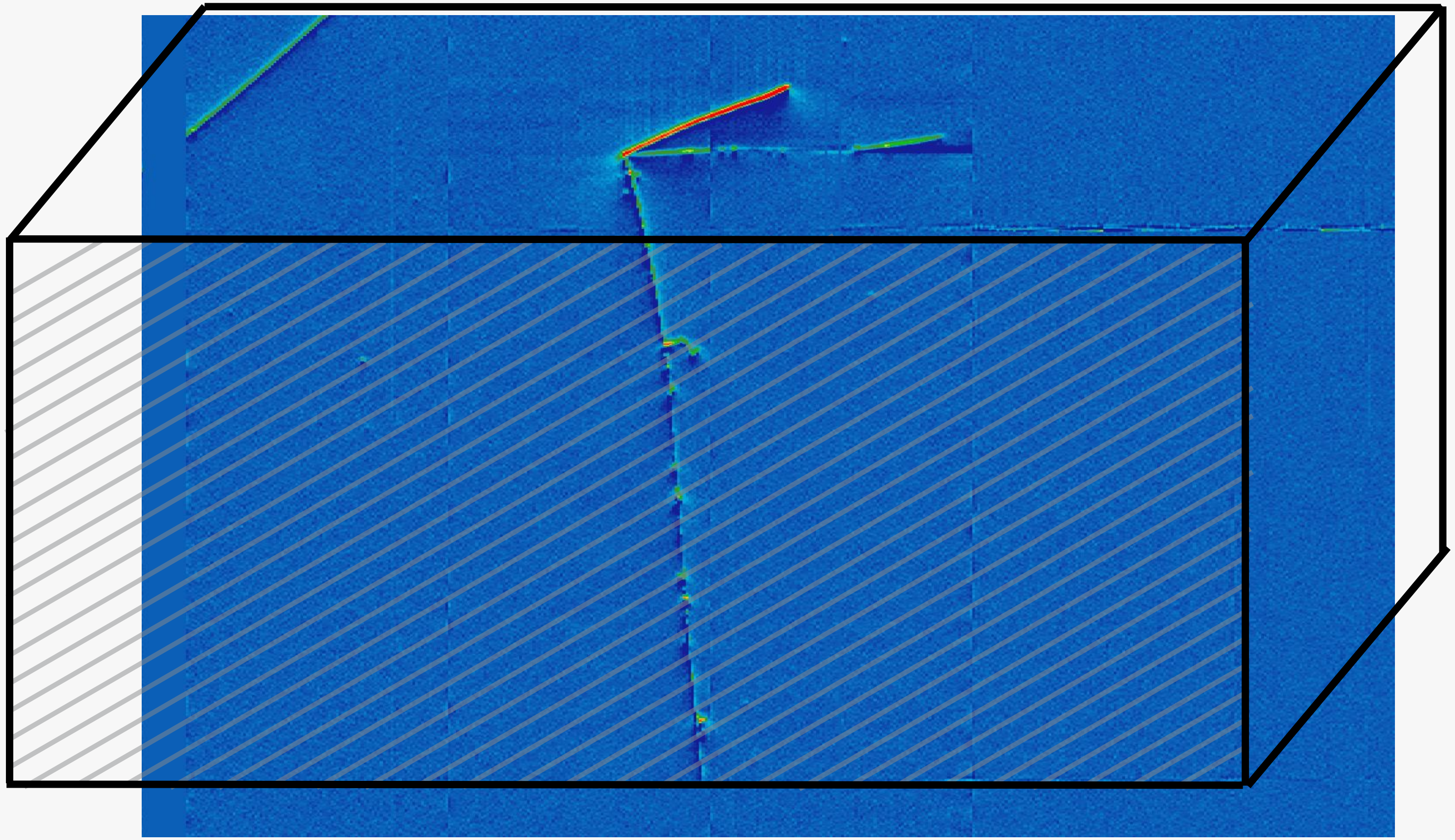




“Projectio

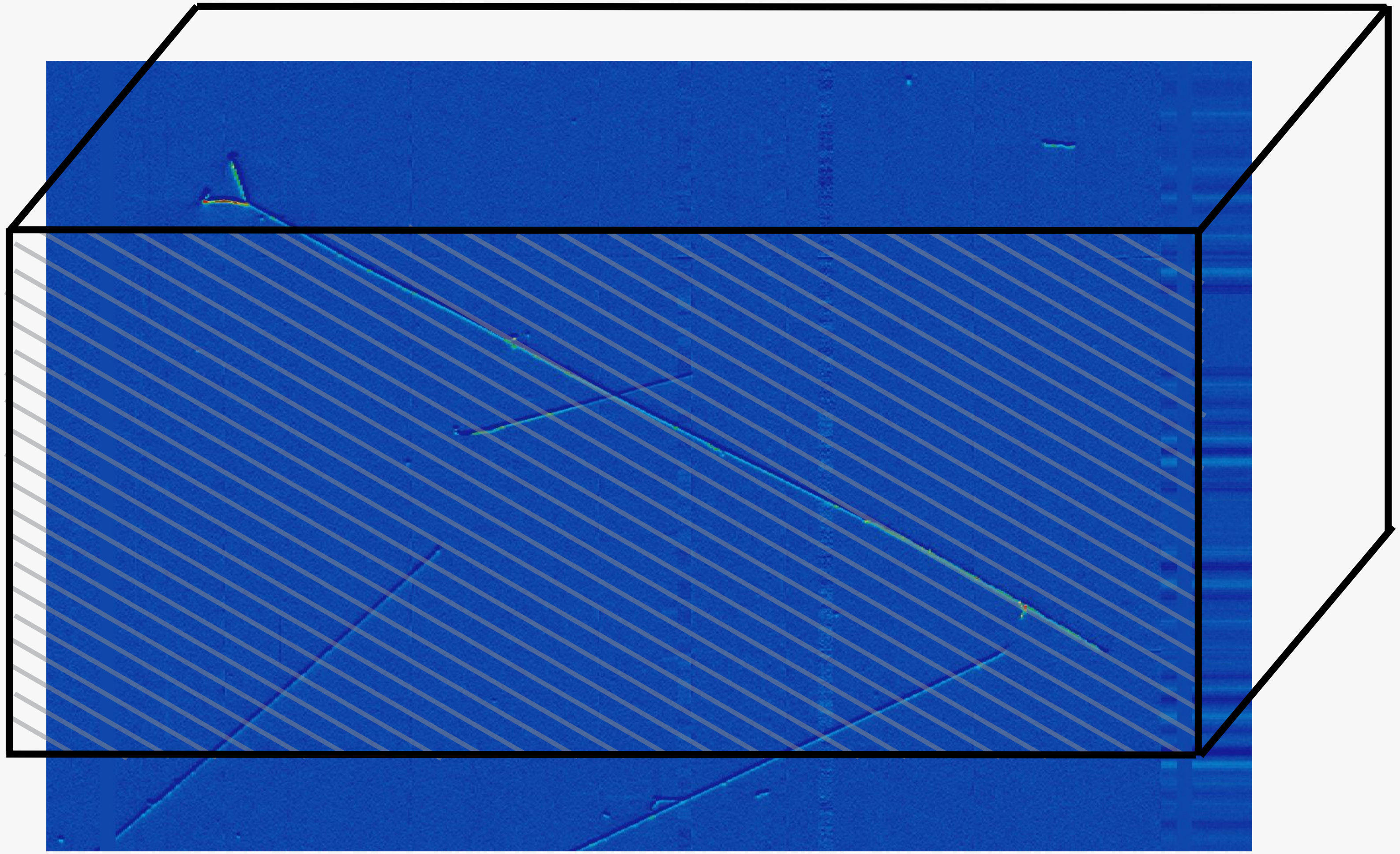


“Projection” Chamber – angle 1

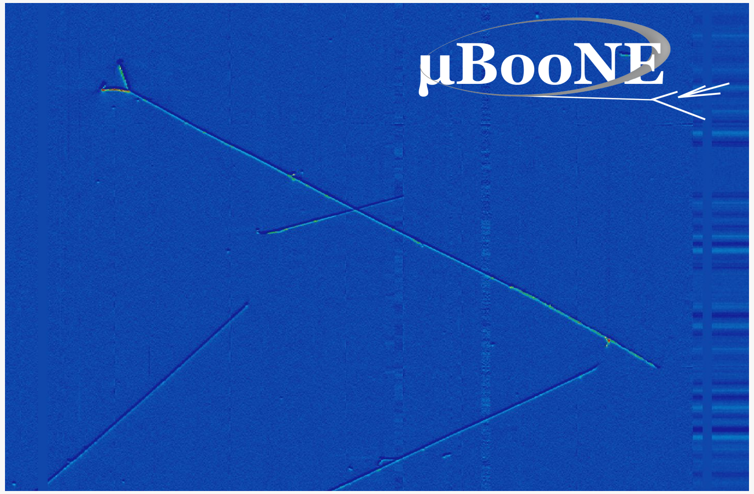


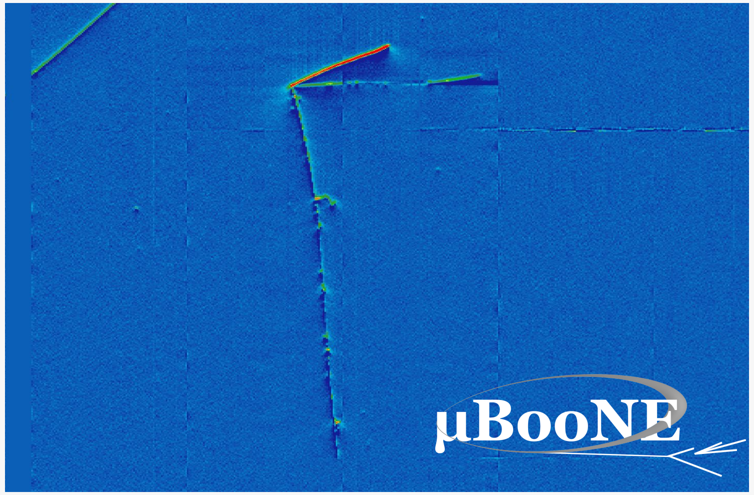
Argonne NATIONAL LABORATORY

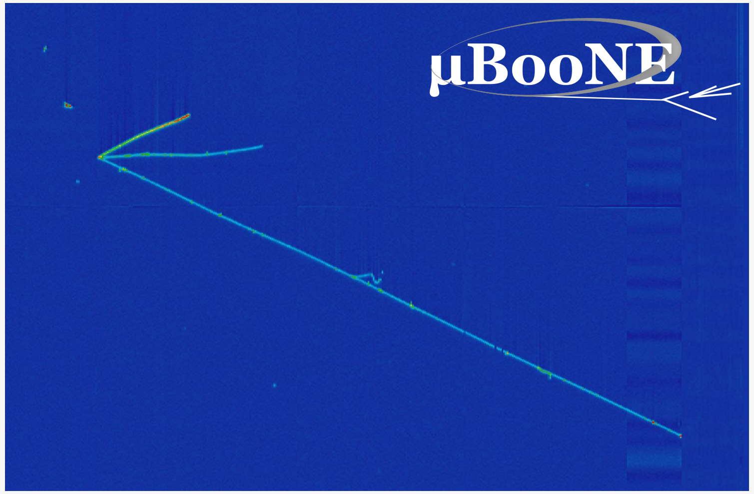
“Projection” Chamber – angle 2



3 Projections of Same Objects







Liquid argon time projection chambers are the detector-of-choice for high energy physics experiments

1. Argoneut (Fermilab, Decommissioned)
2. Lariat (Fermilab Testbed, Running)
3. MicroBooNE (Fermilab Running)
4. SBND (Fermilab, Construction)
5. ICARUS (Fermilab/INFN, Assembly)
6. ProtoDUNE SP (CERN, Running)
7. ProtoDUNE DP (CERN, Assemly)
8. DUNE ND (Fermilab, Design) – **pixel readout**
9. DUNE FD (Sanford Lab, SD, Design)

Many other physics experiments that need fine-grained tracking and detailed calorimetry use similar time projection chamber technology.

Novel designs will replace the sense wires with pixel planes for intrinsically 3D imaging.

Promising Technology for applications of Deep Learning and HPC

Large data size (O(10^7) pixels per image – for example, 1260 x 2048 pixels, 3 images = 1 Event)

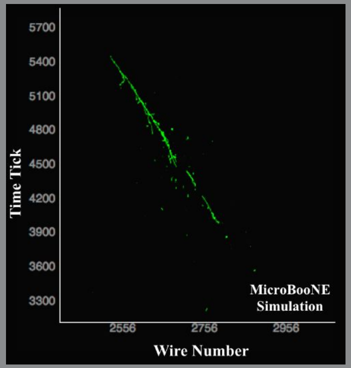
Huge datasets (millions of images of both simulated and real data, per experiment)

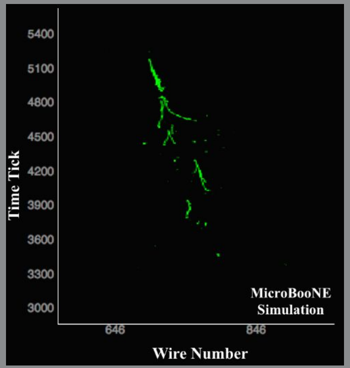
Multiple fundamental science applications from neutrino oscillations, proton decay searches, beyond-standard-model physics searches

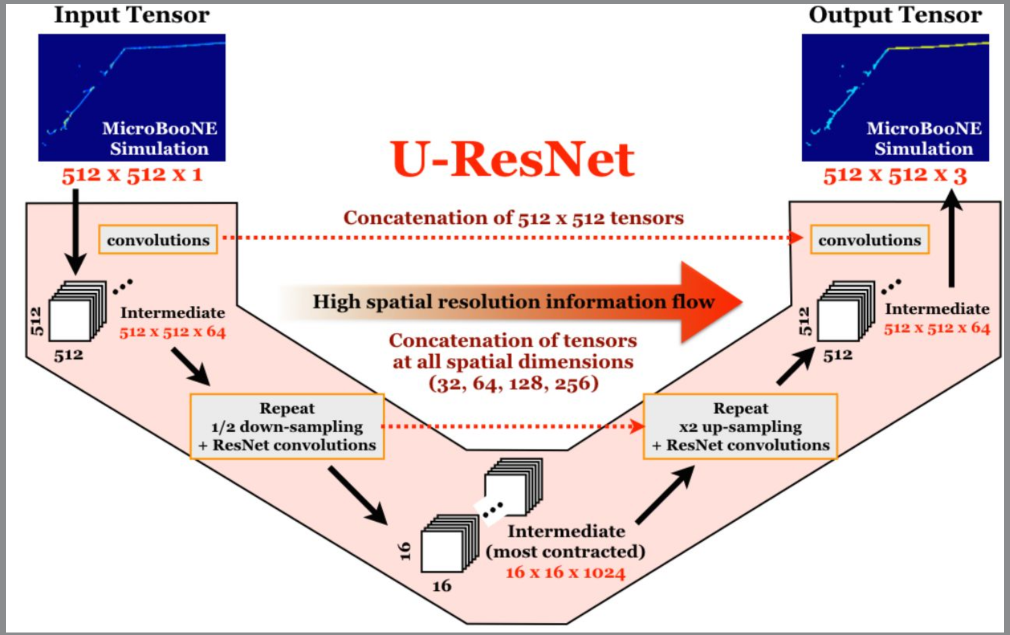
Multiple deep learning techniques directly applicable

Electron vs Photon (Classification)

JINST 12, P03011 (2017).







A Deep Neural Network for Pixel-Level Electromagnetic Particle Identification in the MicroBooNE Liquid Argon Time Projection Chamber

More Data, More Problems

Large scale, high resolution detectors don't fit into any successful paradigm of analysis.

For the MicroBooNE detector (Fermilab):
Each event is ~40MB of raw data for ~5ms of data. 8 GB/s of data (**almost 30 TB per hour!**)

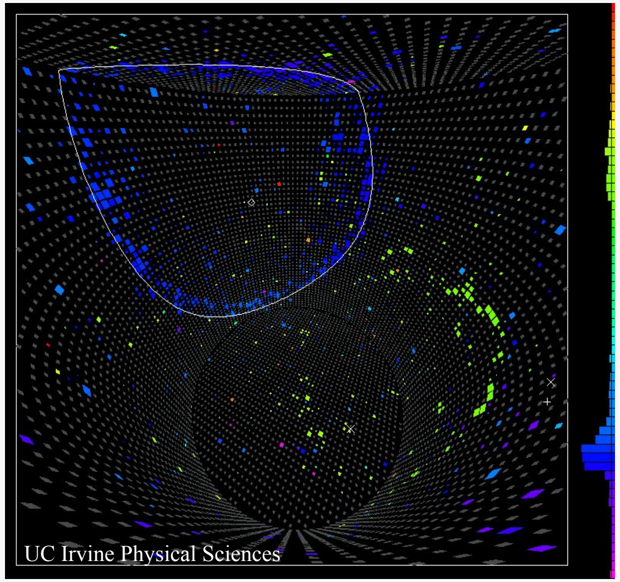
Not feasible!

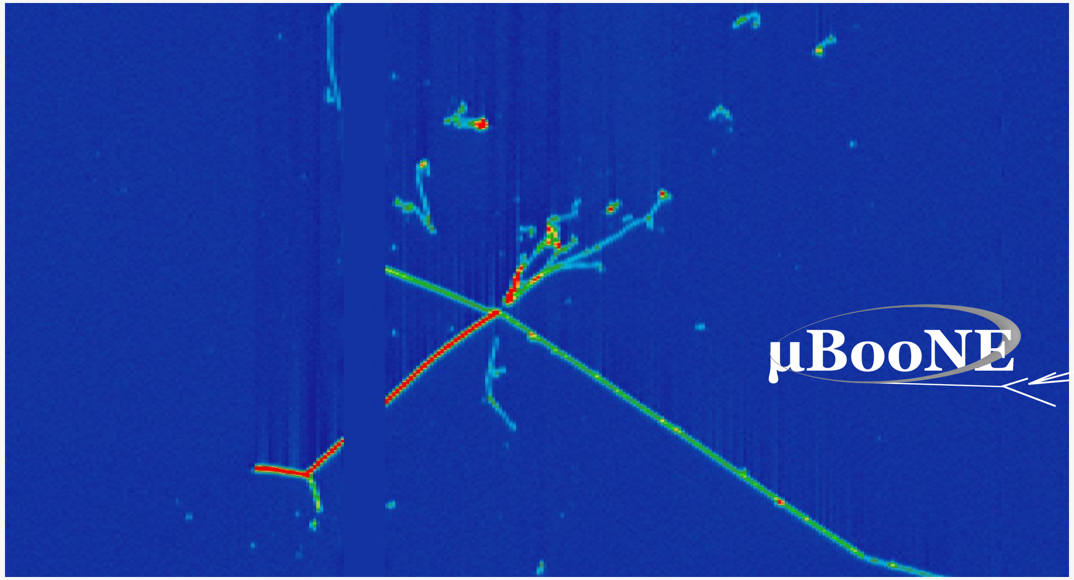
Need to develop a **fast, automated** pattern recognition.

Argonne NATIONAL LABORATORY

More Data, More Problems

Large scale, high resolution detectors don't fit into any successful paradigm of analysis





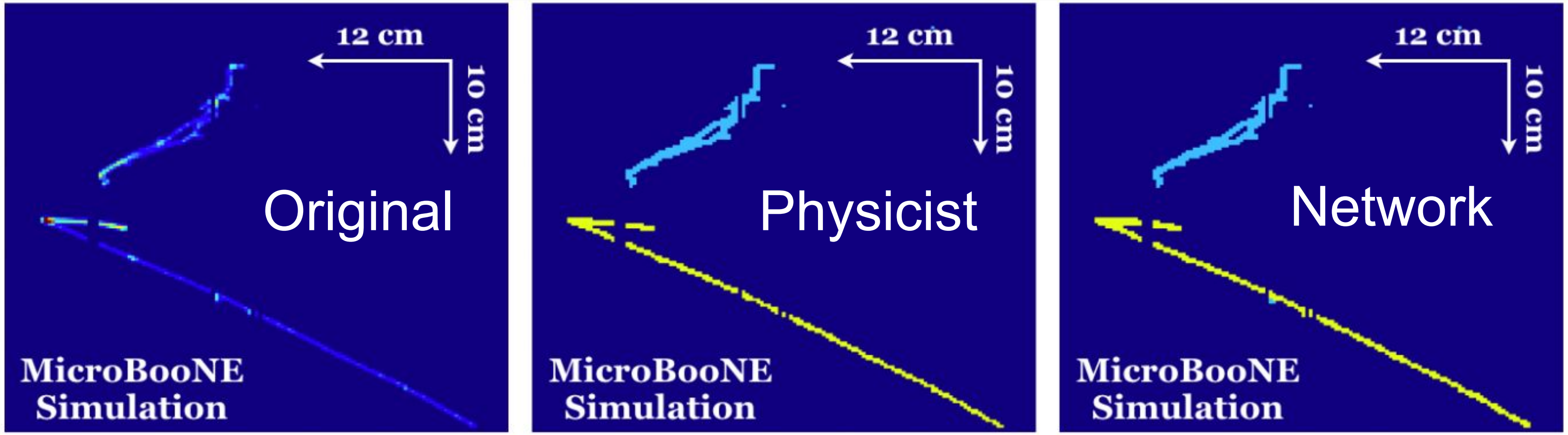
Complex data requires complex pattern recognition.

Physicist vs. The Machine

Sample	Data	Simulation	Simulation	Simulation
Label	Physicist	Physicist	Simulation	Simulation
Prediction	U-ResNet	U-ResNet	U-ResNet	Physicist
ICPF mean	3.4	2.5	1.8	2.0
ICPF 90%	9.0	5.7	4.6	4.8
Shower	4.8	3.4	3.0	2.6
Track	2.7	2.4	2.2	2.9

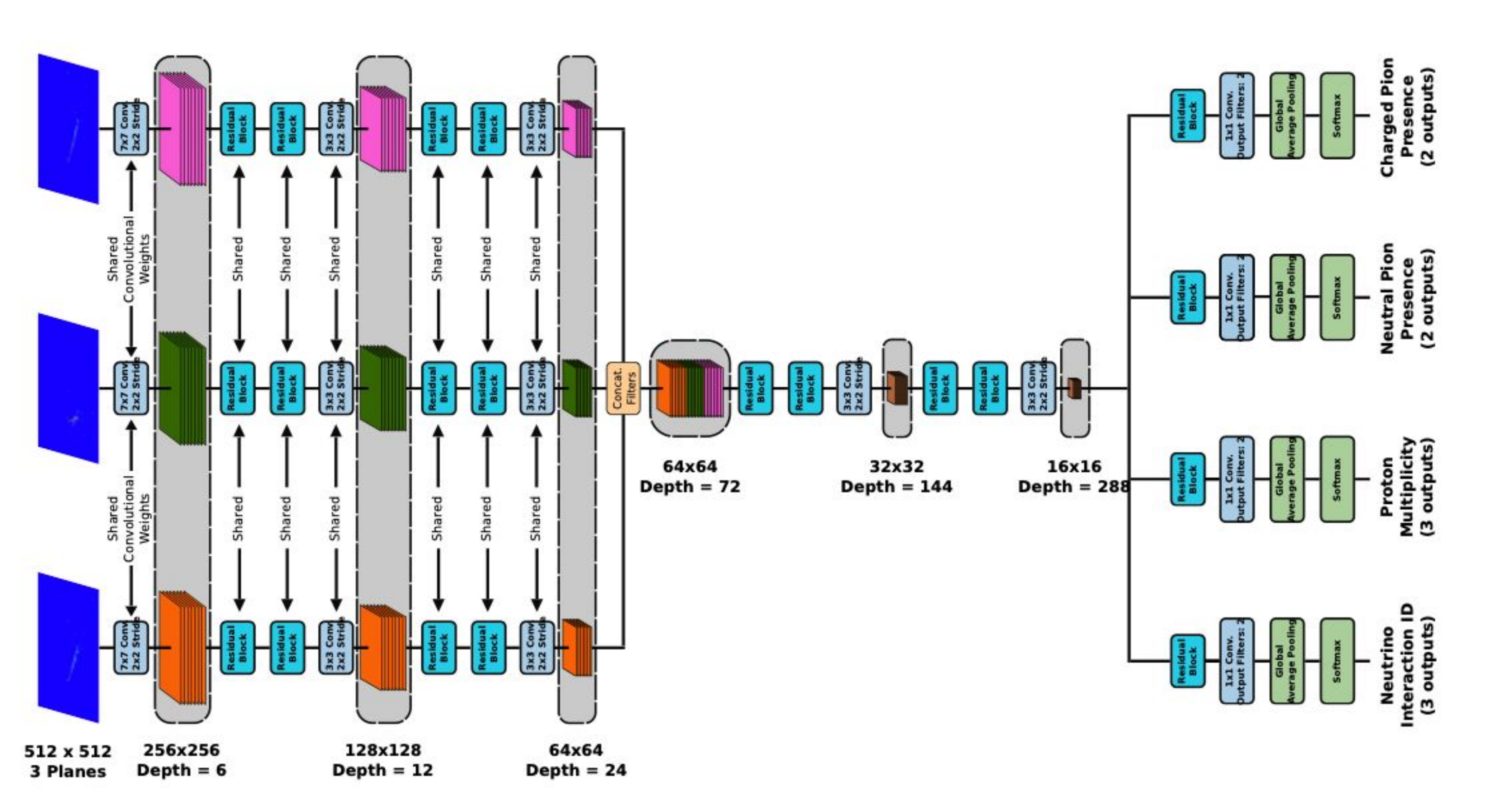
ICPF = Incorrectly Labeled Pixel Fraction.

No classical algorithm achieves this accuracy. Machine Learning out performs physicist hand labels.



Cross Plane Information - Classification

How to best utilize multiplane detectors?



Correlations across projections are correlated and learnable.

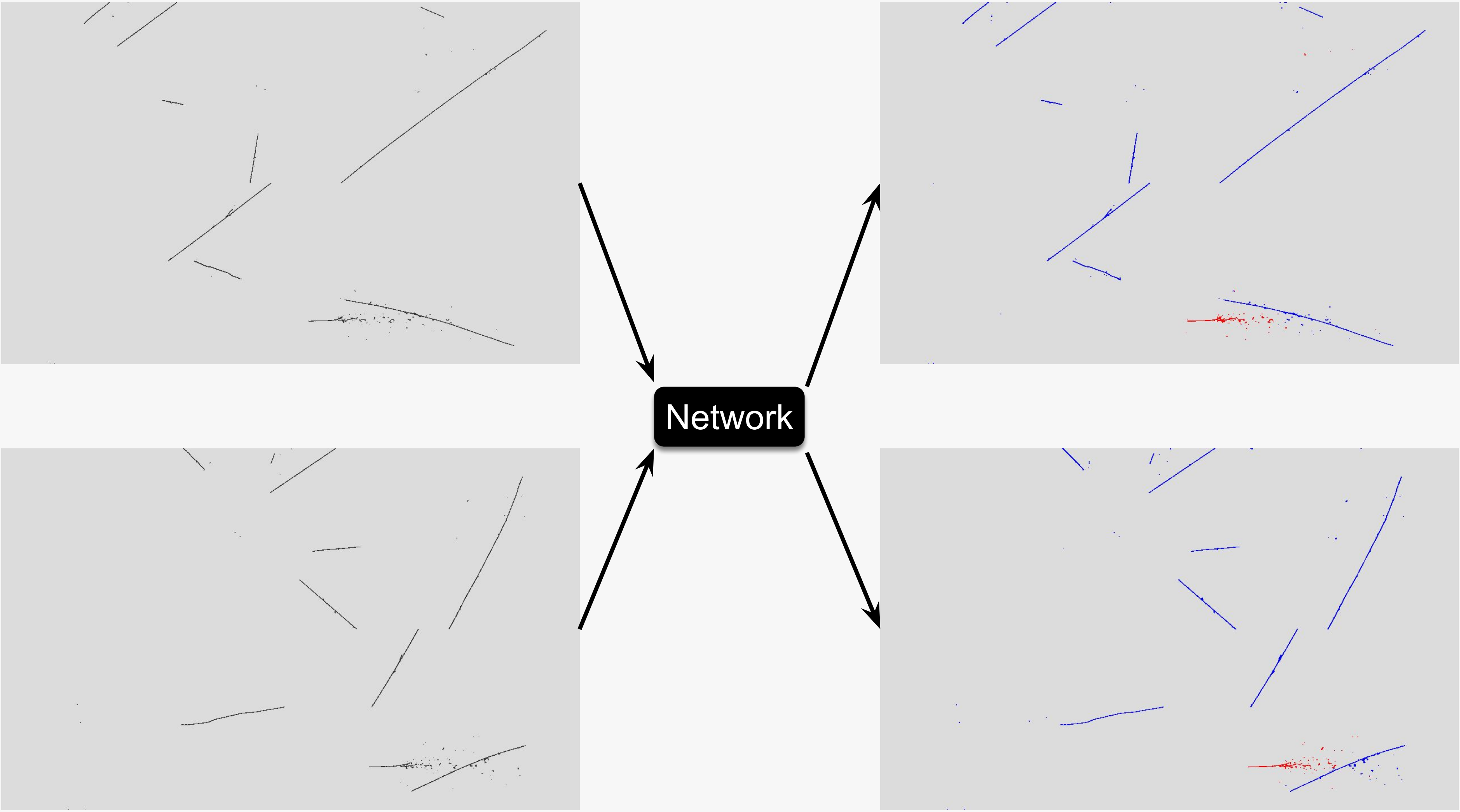
Argonne NATIONAL LABORATORY

Cosmic Tagging

For detectors on the surface of the earth, the particles that produce ionization from neutrino interactions (electrons, protons, muons, photons …) are the same ones that produce ioninzation from cosmic rays.

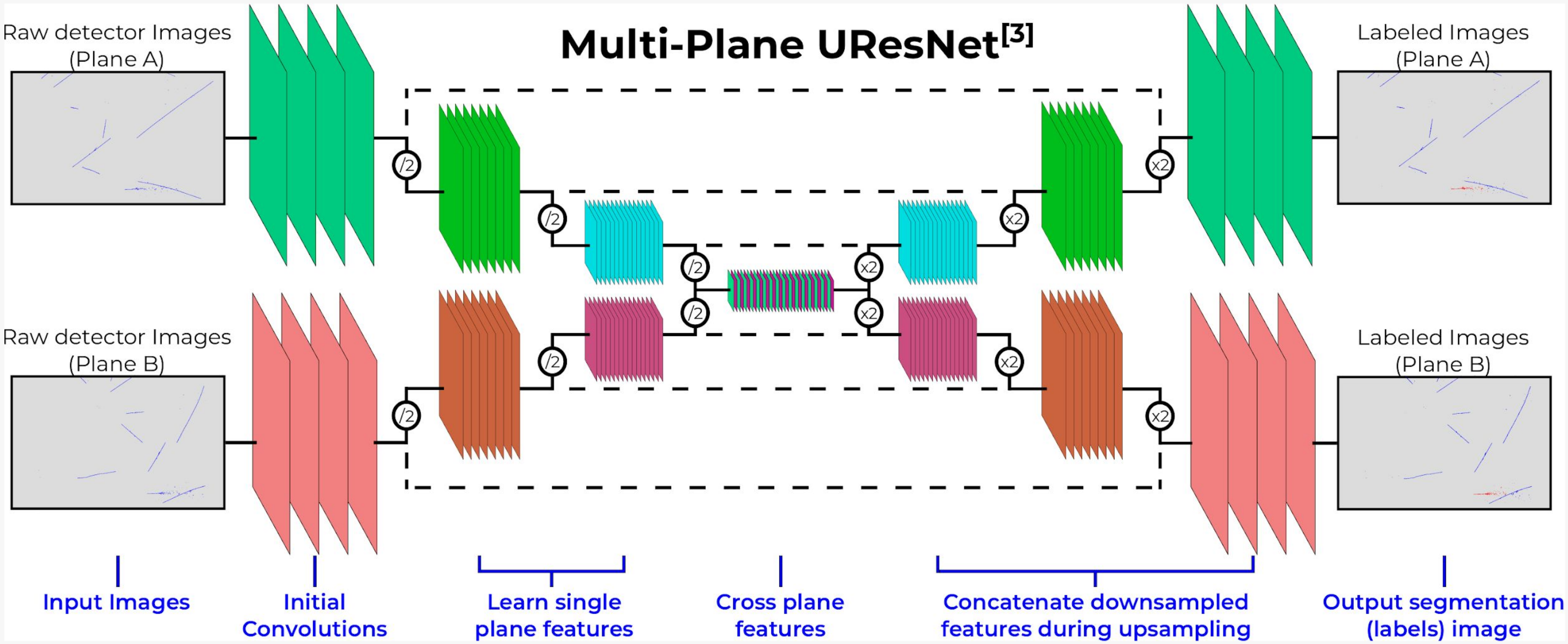
The signatures in the detector are indistinguishable without pattern recognition, but cosmic particles are **much** more common (many orders of magnitude)

Cross Plane Information



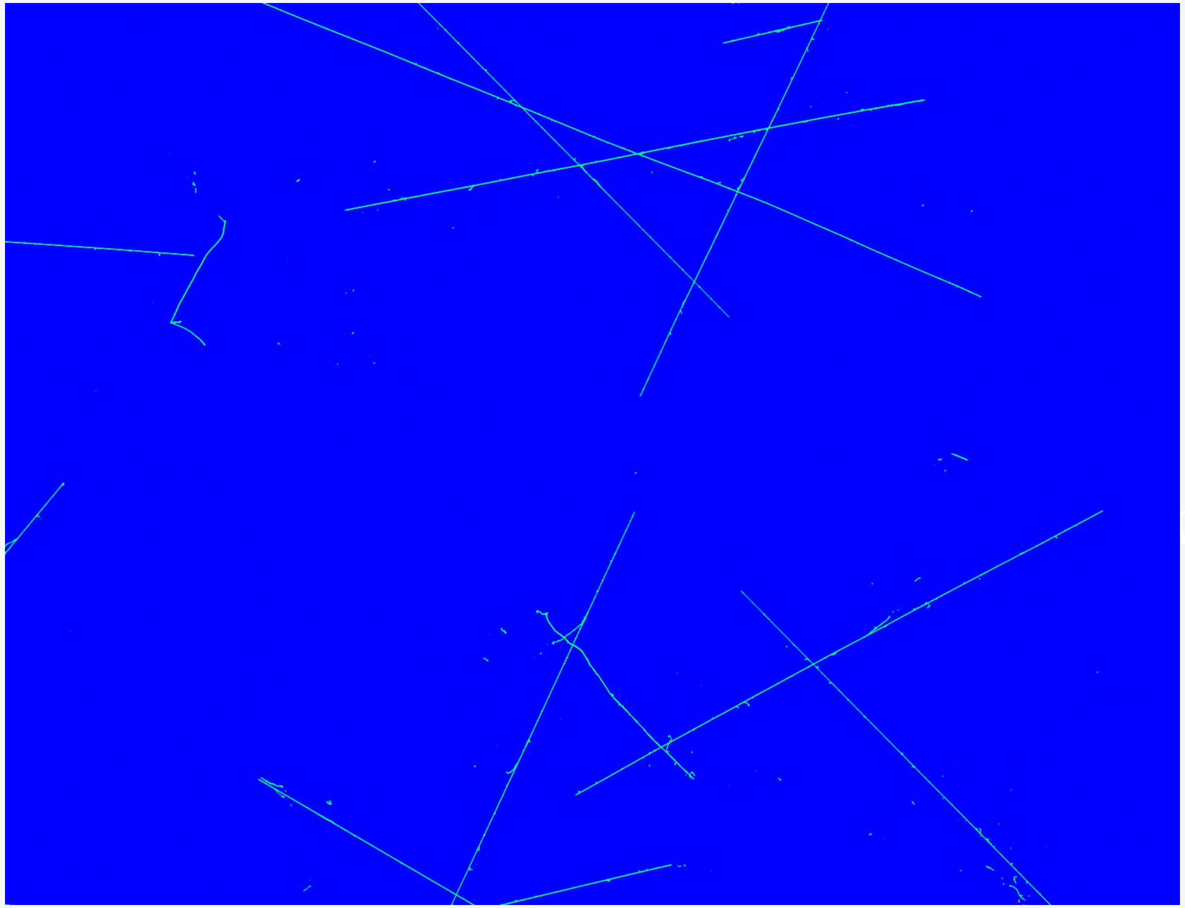
Cross Plane Information - Segmentation

How to best utilize multiplane detectors?

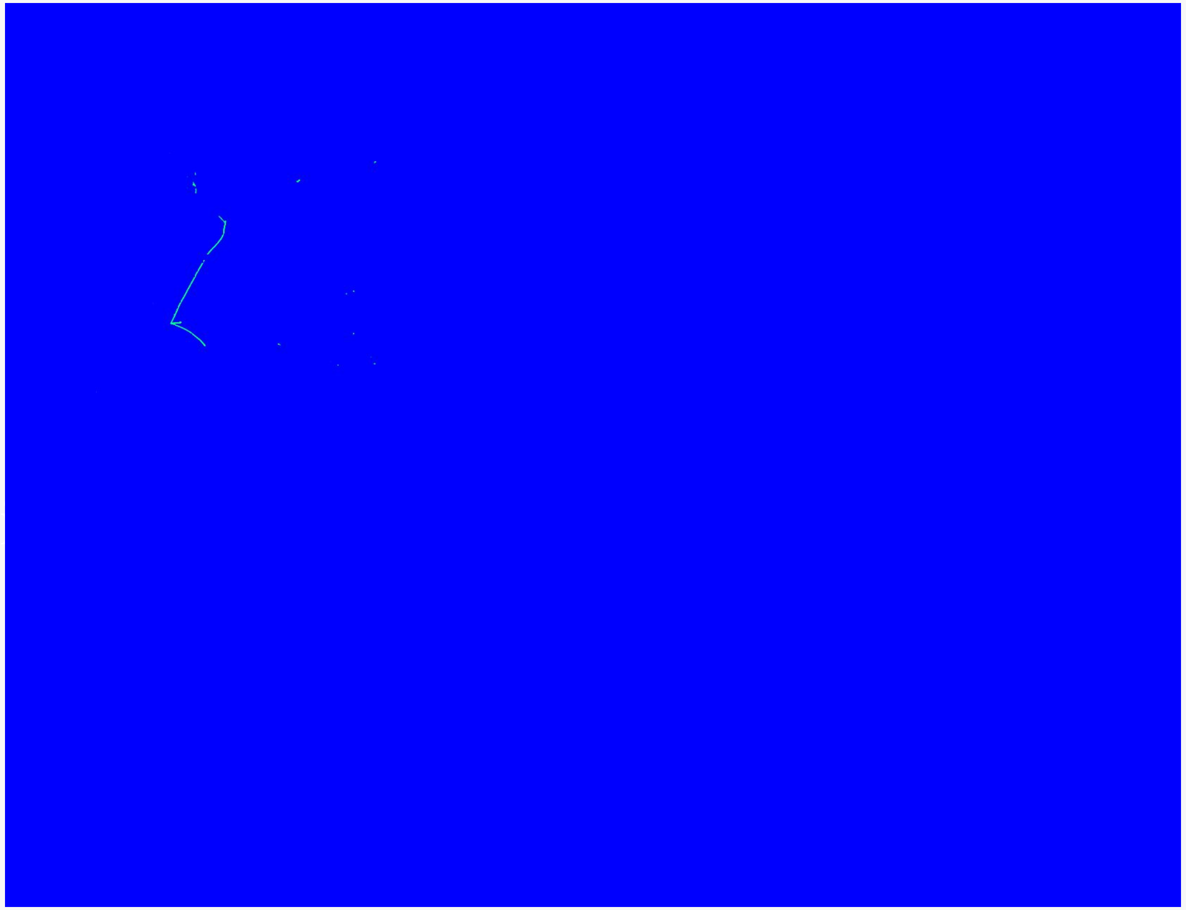


Correlations across projections are correlated and learnable.

Multiplane Segmentation

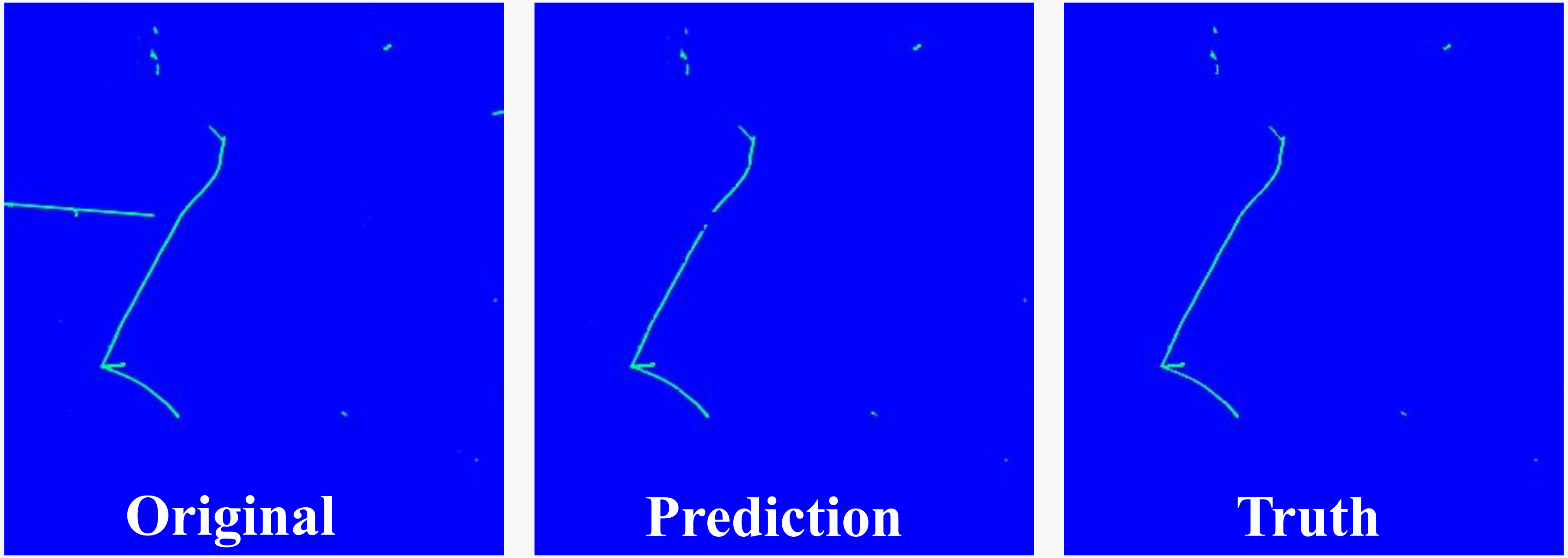


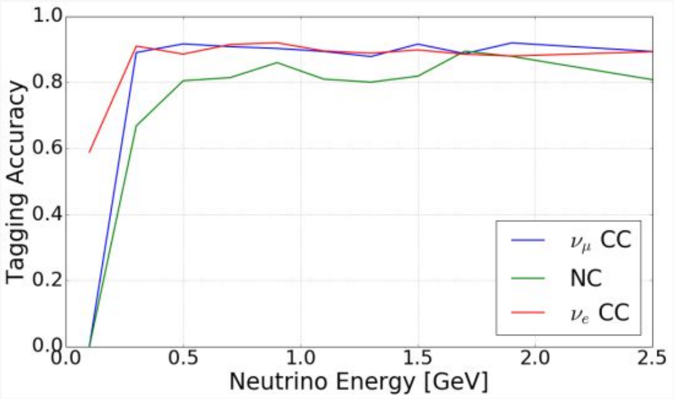
Network Input



Network Output
(No cosmics!)

Segmentation: Zoom on the Neutrino





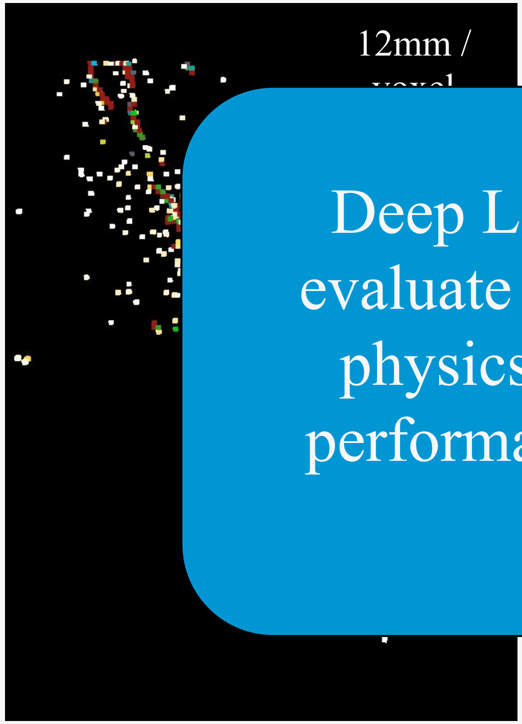
Neutrino passing rate: 80%
Background-only: 6%

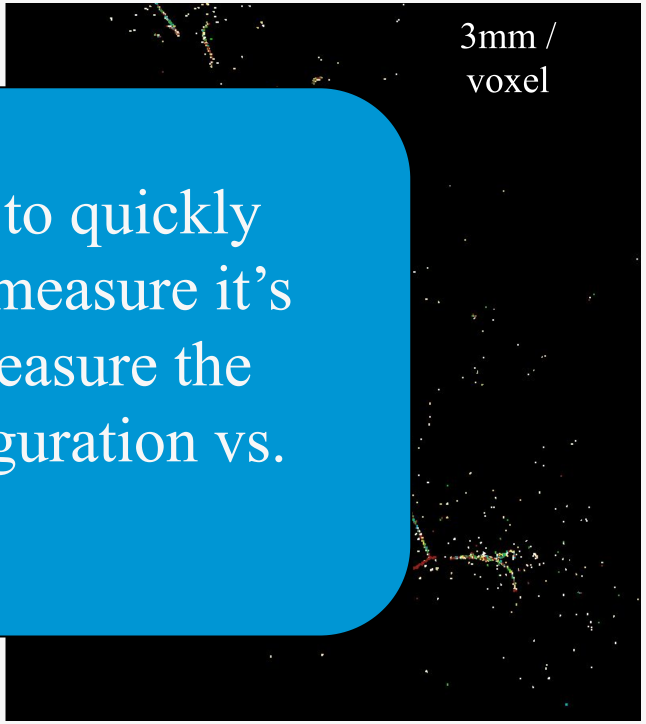
Neutrino Purity: 89%
There is no corresponding classical algorithm that achieves this result.

Natively 3D Detectors

Most LArTPCs at large scale are multi projection, 2D detectors - **but pixelated, 3D detectors are coming.**

An obvious question: are the performance gains worth the extra costs?



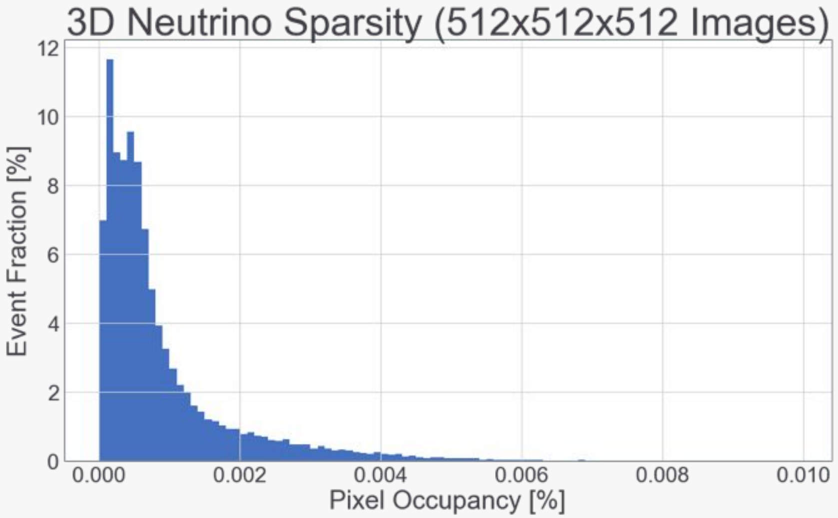


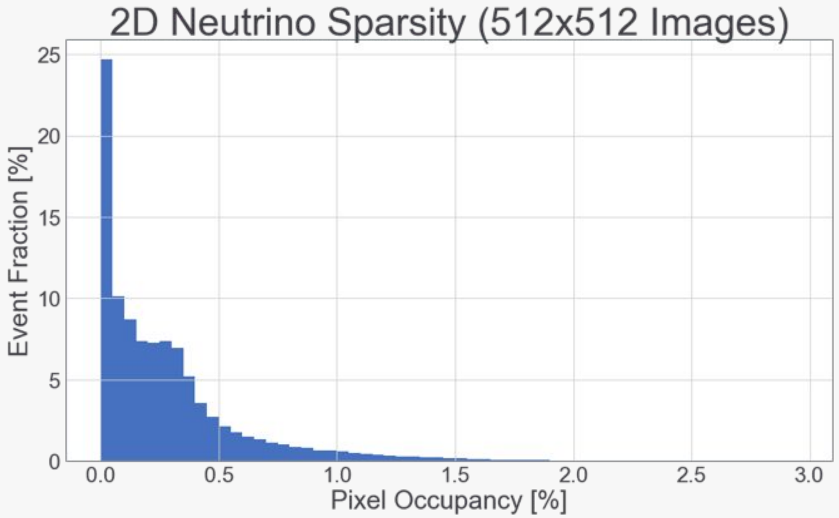
Deep Learning offers a way to quickly evaluate a configuration and measure it's physics reach, in order to measure the performance gains of a configuration vs. expected costs.

Sparse vs. Dense Convolutions

Deep learning in neutrino datasets could benefit dramatically from an intelligent application of sparse techniques to machine learning.

- Save **memory** by not storing activations on empty pixel sites
- Save **computations** by ignoring non-active sites.





Neutrino images from liquid argon time projections chambers are very sparse.

Occupied pixel fraction is:

< 0.01 in 2D at 512^2 pixels

< 5e-5 in 3D at 512^3 pixels

Sparse Machine Learning Implementations

Details vary, but there is some literature on convolutional neural networks on spatially sparse data.

1. SBNET (from Uber) - inference only gains in speed by only feeding forward active blocks of input.
2. OctNet - sparse convolutions with tree based organization for sparse convolutions
3. Sparse 3D Conv. Nets - hashmap based convolutions for managing sparse representations
4. Submanifold Sparse Convolutional Networks - follow up on 3, deals with dilation as well.
5. (PointNet - graph network that works OK on sparse data)
6. (Dynamic Graph CNN - graph network with convolution-like operation)

Submanifold Sparse Convolutional Networks

This technique brings three new convolution operators that differ from traditional convolutions:

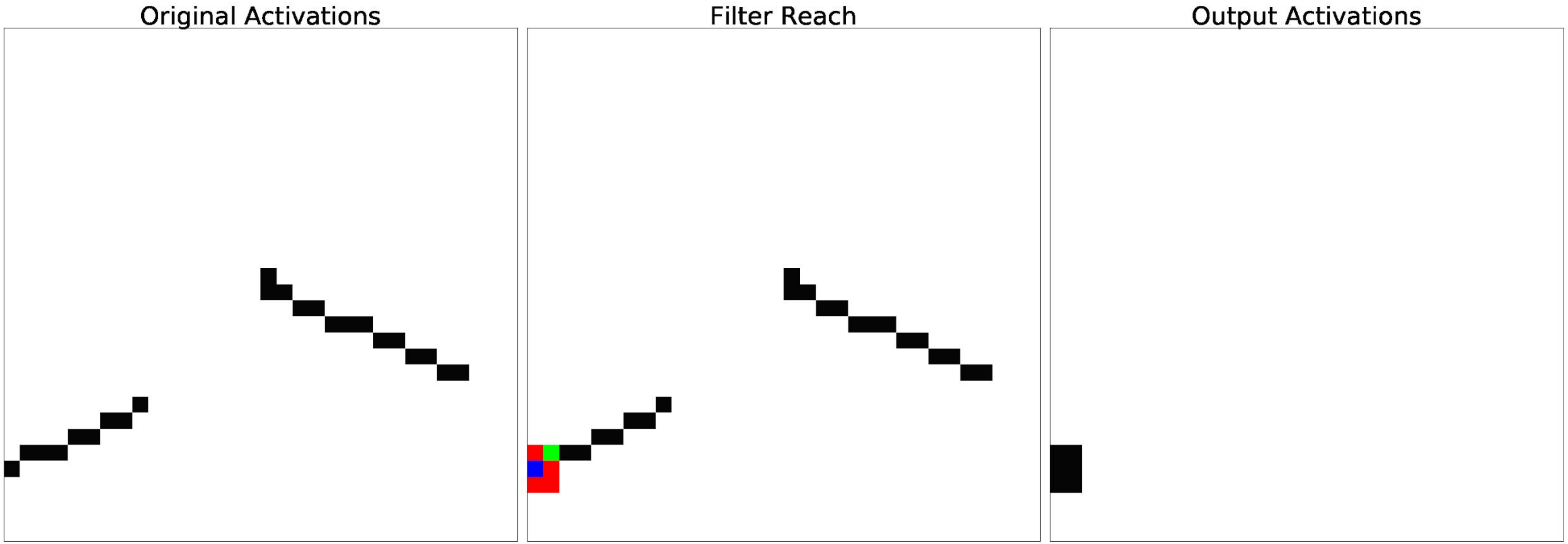
- **Sparse Convolutions** operate only on active input sites, but behave traditionally on the output layer.
- **Submanifold Sparse Convolutions** maintain the same set of active sites on the input and output layers (and are “valid” - no change to spatial size)
- **Sparse Deconvolutions** invert sparse convolutions by reversing the mapping from input to output of a sparse convolution, useful in UNet or FCN segmentation architectures.

Submanifold Sparse Convolutional Networks

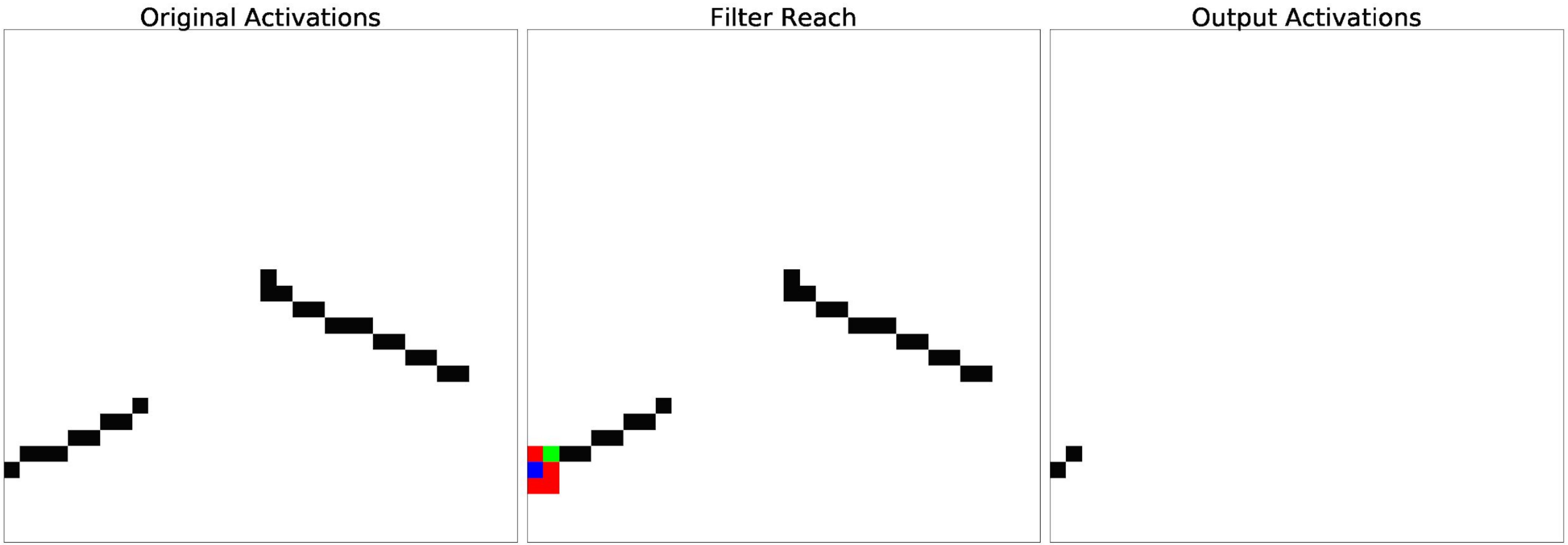
This technique brings three new convolution operators that differ from traditional convolutions:

- **Sparse Convolutions** operate only on active input sites, but behave traditionally on the output layer.
- **Submanifold Sparse Convolutions** maintain the same set of active sites on the input and output layers (and are "valid" - no change to spatial size)
- **Sparse Deconvolutions** invert sparse convolutions by reversing the mapping from input to output of a sparse convolution, useful in UNet or FCN segmentation architectures.

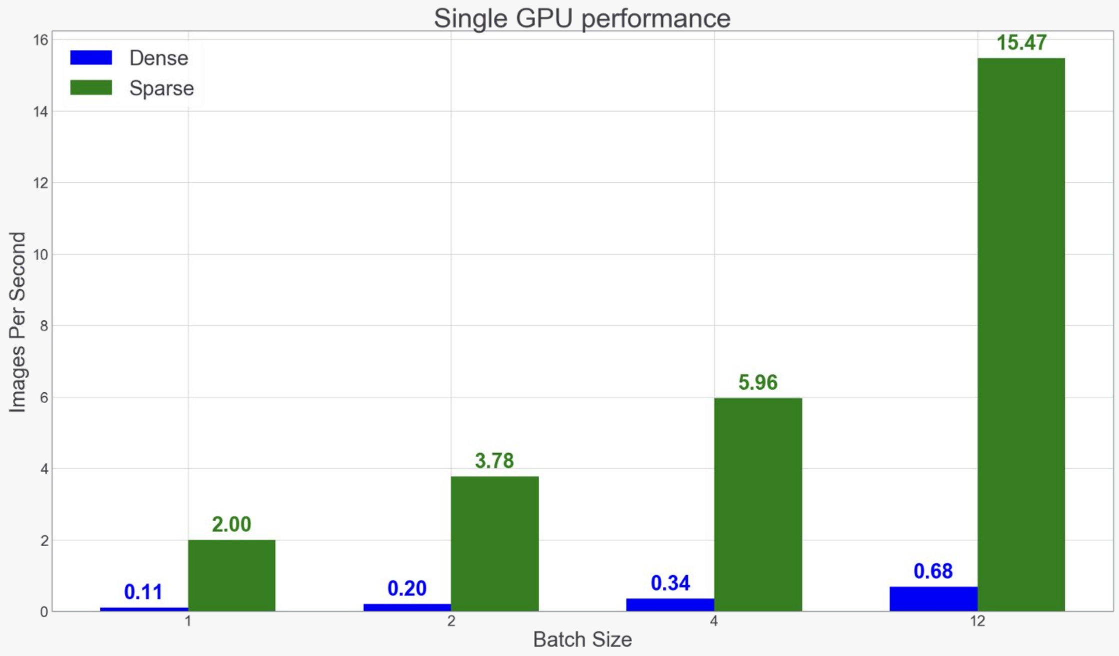
Sparse Convolutional

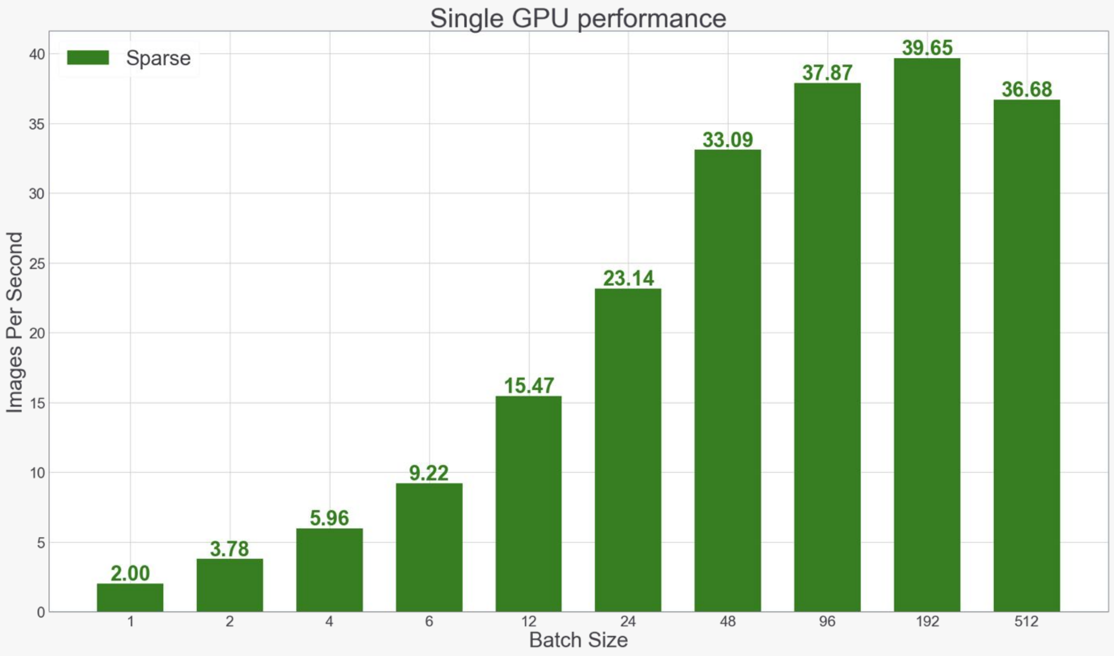


Submanifold Sparse Convolutional



Sparse/Dense Comparison

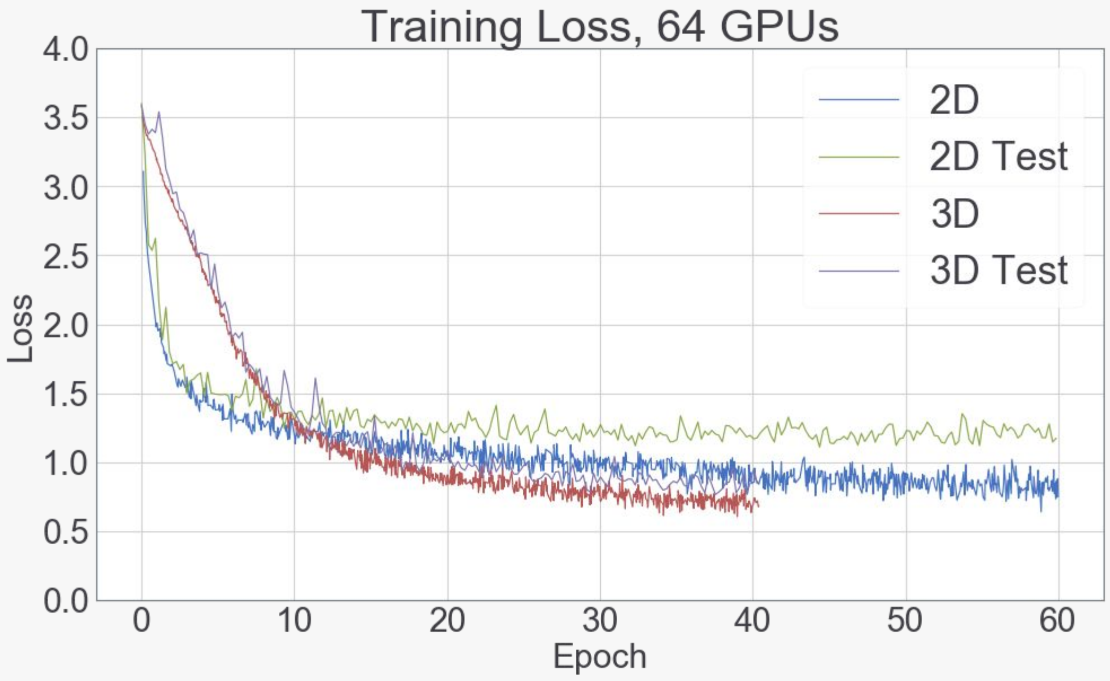


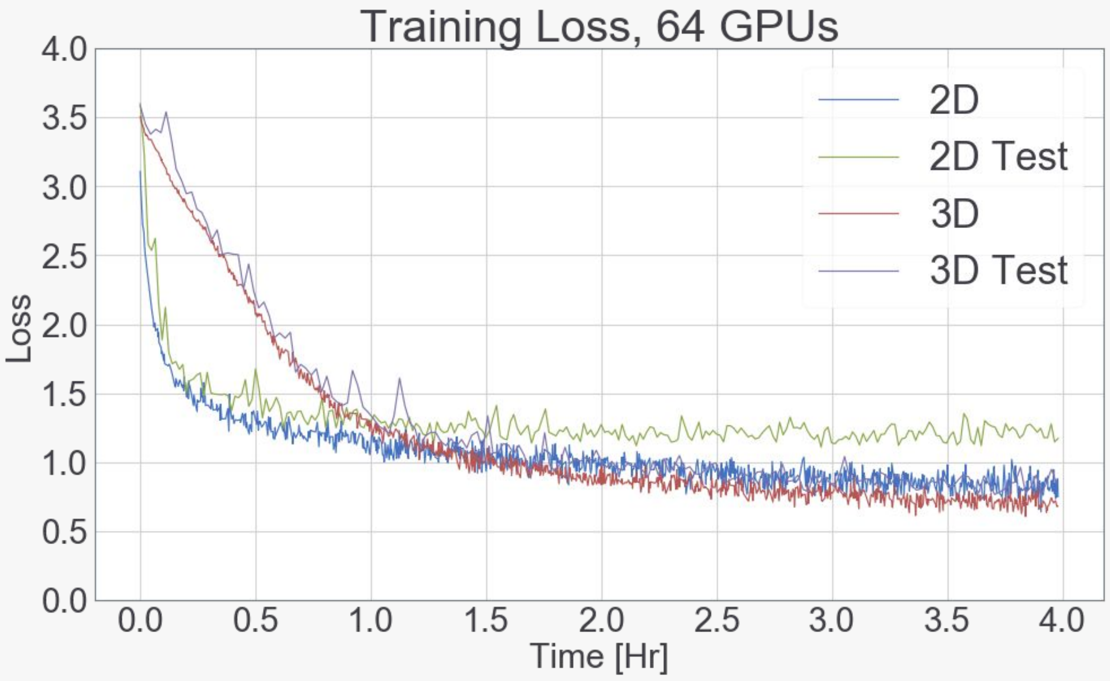


Roughly 18x speedup in training on identical batch sizes, 58x speedup in peak single-GPU throughput.

Argonne NATIONAL LABORATORY

Sparse/Dense Comparison





60 Epochs (40 in 3D) on 64 GPUs in just 4 hours! Approaches state-of-the-art performance, quantitative study starting to benchmark accuracy of dense vs. sparse implementations.

Sparse Convolutions

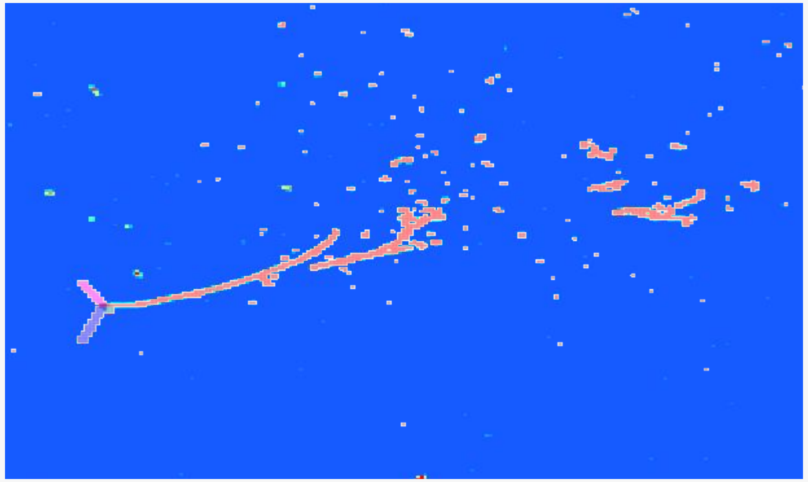
Relatively niche computer vision application could be very impactful in particle physics:

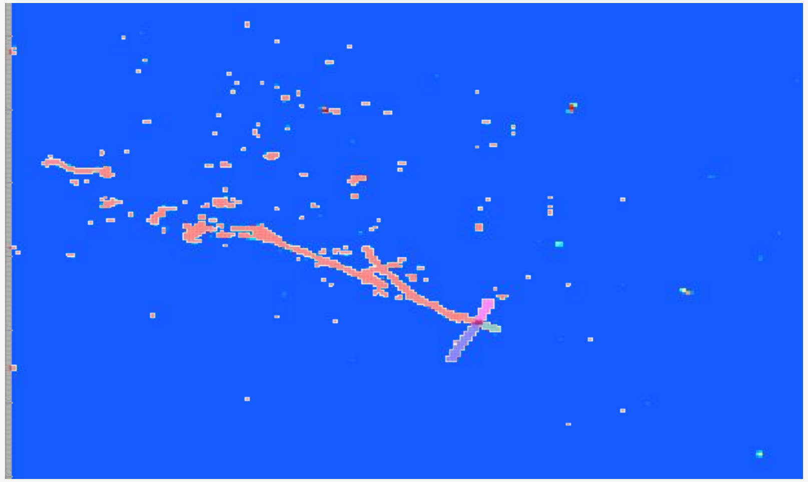
- Dramatic speed up of both training and inference times
- Substantial reduction of memory impact means less powerful machines can perform inference
  - Scalable to CPUs of open science grid
- Neutrino datasets can feed back to the computer vision community for development of new techniques (instance detection?) with sparse convolutions.

Where is the field going?

Full Event Pattern Recognition

Instance aware, cross-plane particle segmentation.



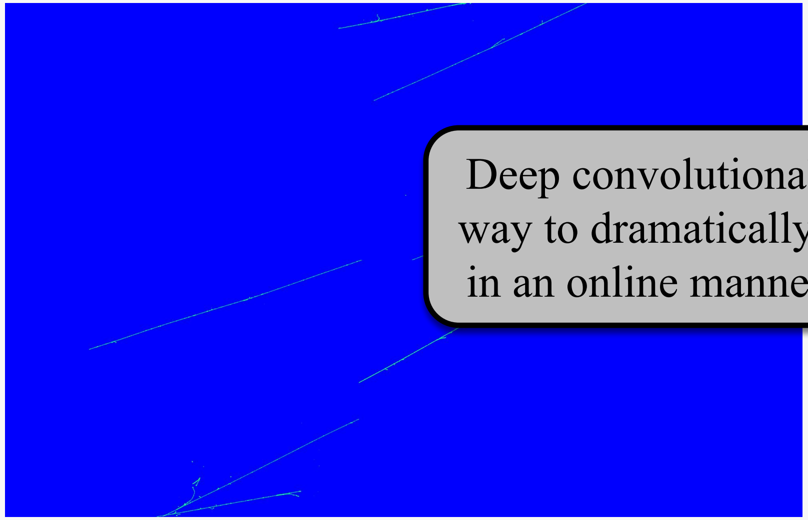


Already, successful demonstrations of segmentation, instance aware predictions, and multi-plane networks have succeeded.

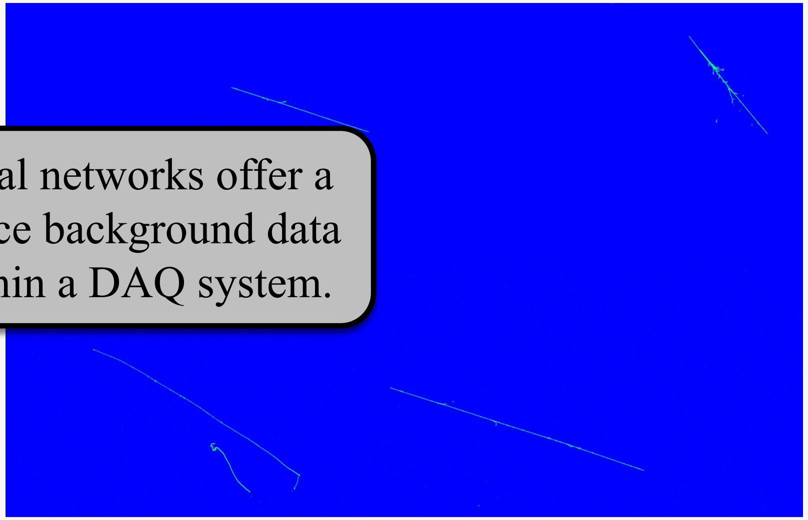
Online Triggers

MicroBooNE data rate is ~8 GB/s. New detectors will only be bigger and faster - how to keep up with the data stream?

Deep convolutional neural networks offer a way to dramatically reduce background data in an online manner, within a DAQ system.



Background Only



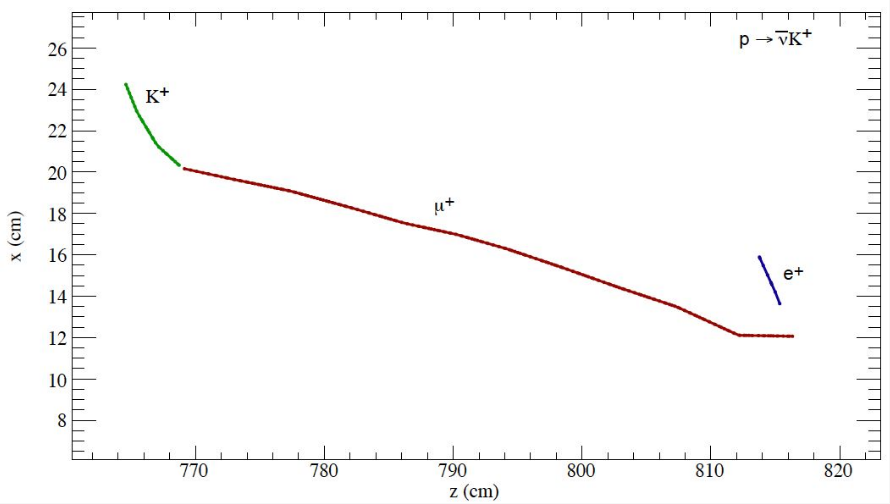
Neutrino Interaction

Requires a different data pipeline, but is feasible for future experiments.

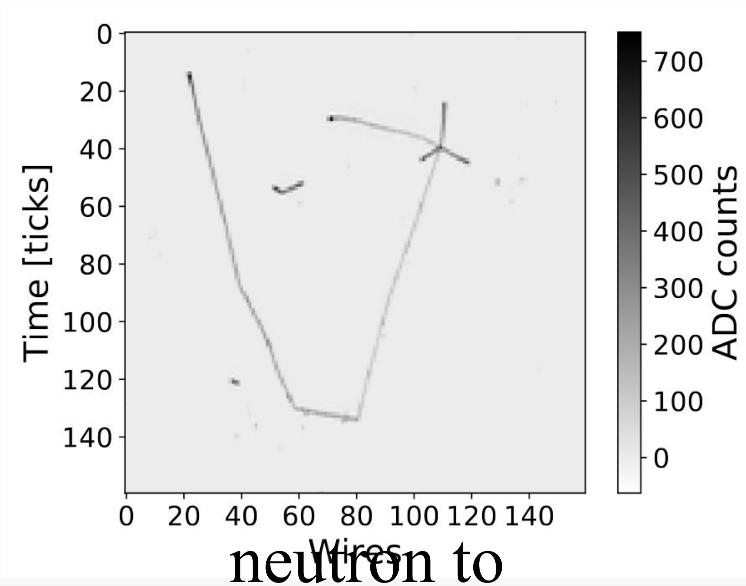
Rare Event Searches

In the Deep Underground Physics Experiment, there are exciting physics studies to do that require powerful analysis techinques.

J. Hewes Thesis, 2017 (http://inspirehep.net/record/1662648)



Proton Decay

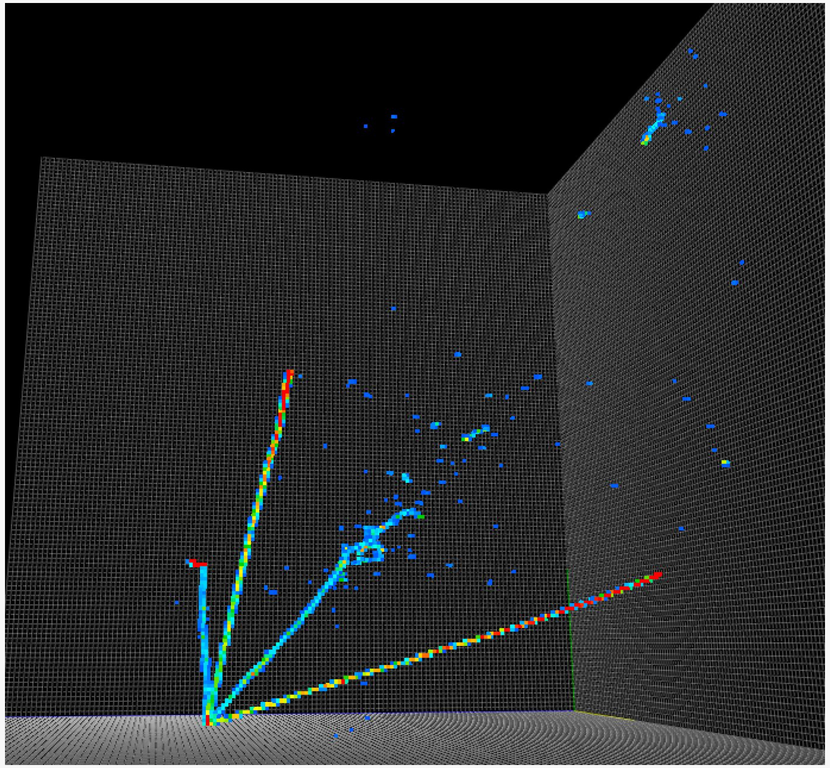


neutron to anti-neutron oscillation

Collaborative Development

Many experiments (and many students!) are interested in deep learning, but applications to neutrino physics are still emerging and common tools unavailable.

Response: **deeplearnphysics.org** to share open source development:



- Open source data format (and open data set)
- OpenGL Interactive Visualization Tools
- Open source network implementations
- Extensive Tutorials
- Slack channel for news, questions, etc.
- Cross-experimental meetings

DeepLearnPhysics

When Professor-of-Physics wants their new student to use deep learning in their thesis, where to turn?

Open Data Set

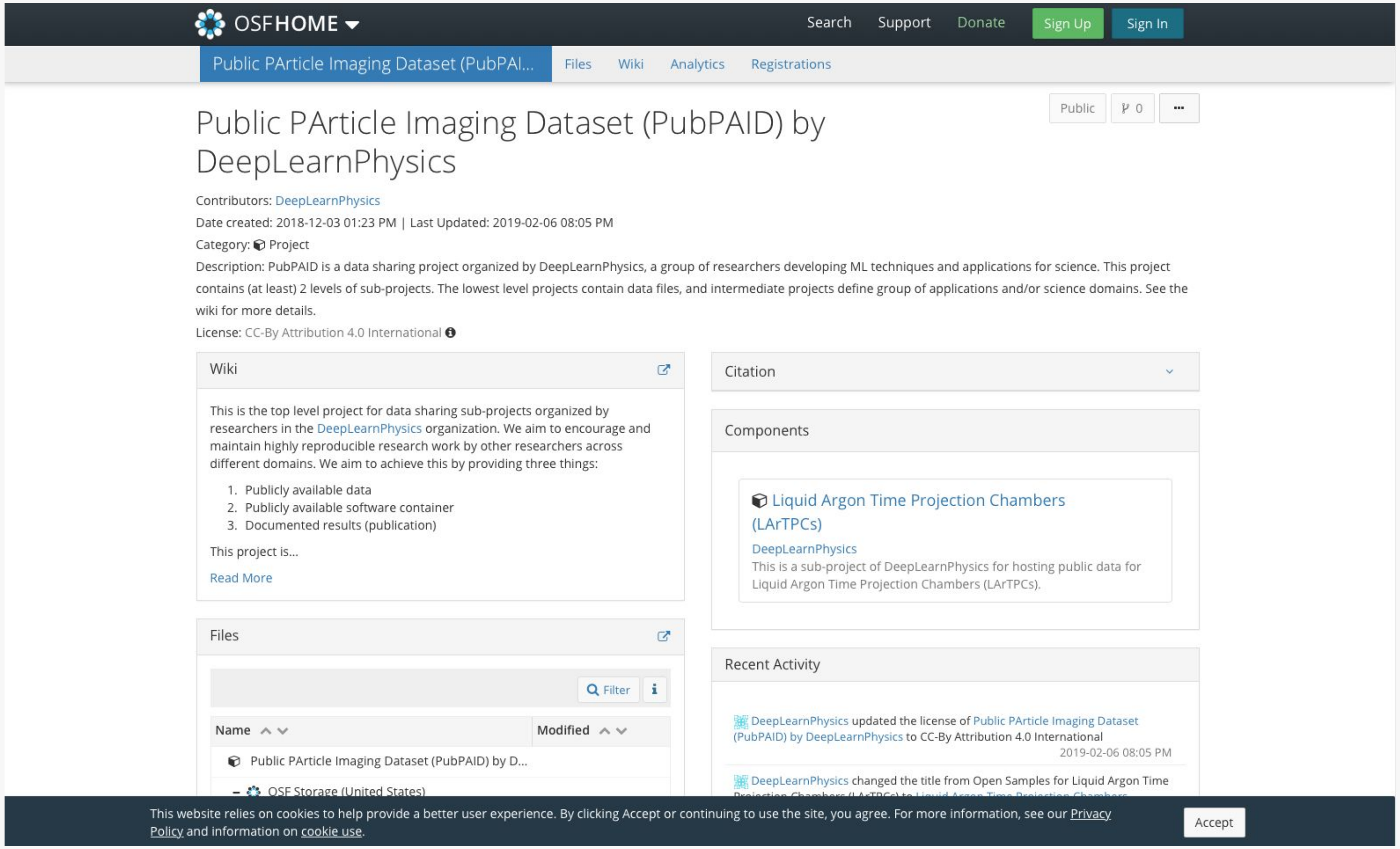
- Freely available data from authentic neutrino simulation
- Tutorials and walkthroughs on how to generate application-specific datasets

Model Zoo

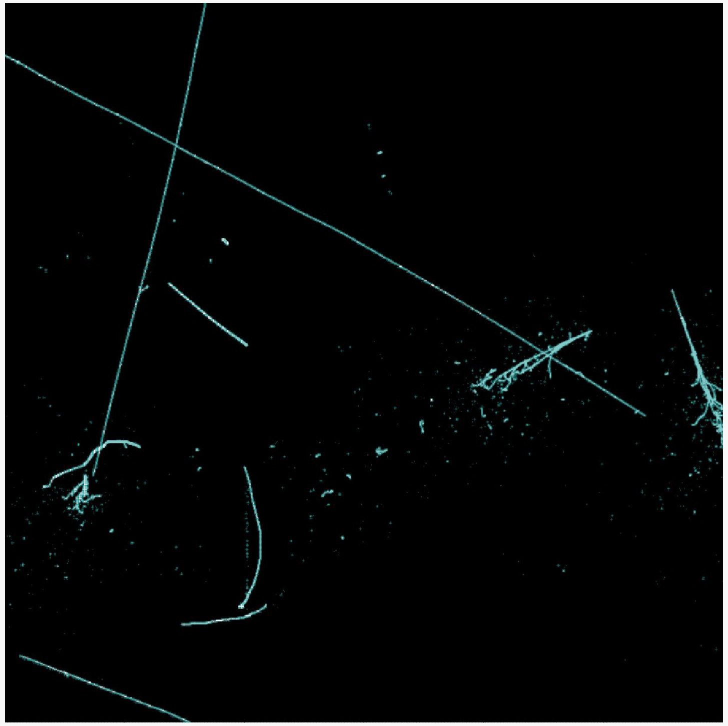
- Open source implementations of important networks for physics
- Evaluation and pre-trained weights for open data

Goal: Papers released early 2019

Open Data Set



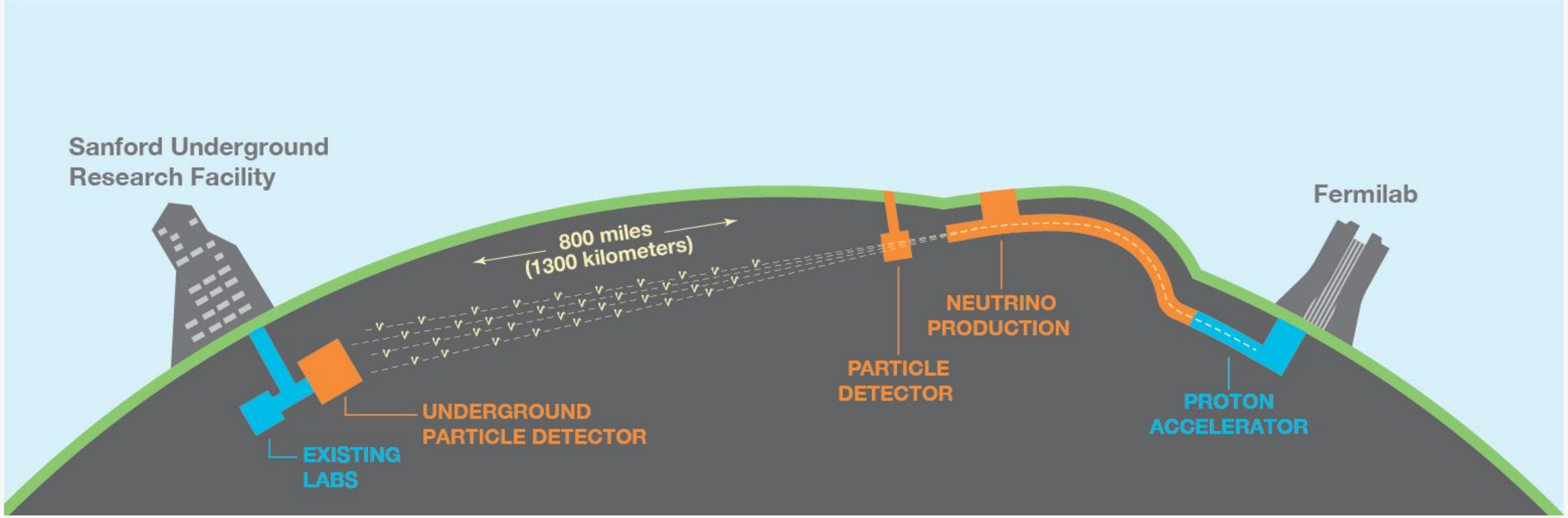
- Data set is public
- Descriptive Paper is in 2nd draft
- Model zoo is coming!



Argonne NATIONAL LABORATORY

DUNE

Neutrino Physics is investing into large, liquid argon imaging detectors.



Deep learning will be THE tool that meets the science needs of DUNE.

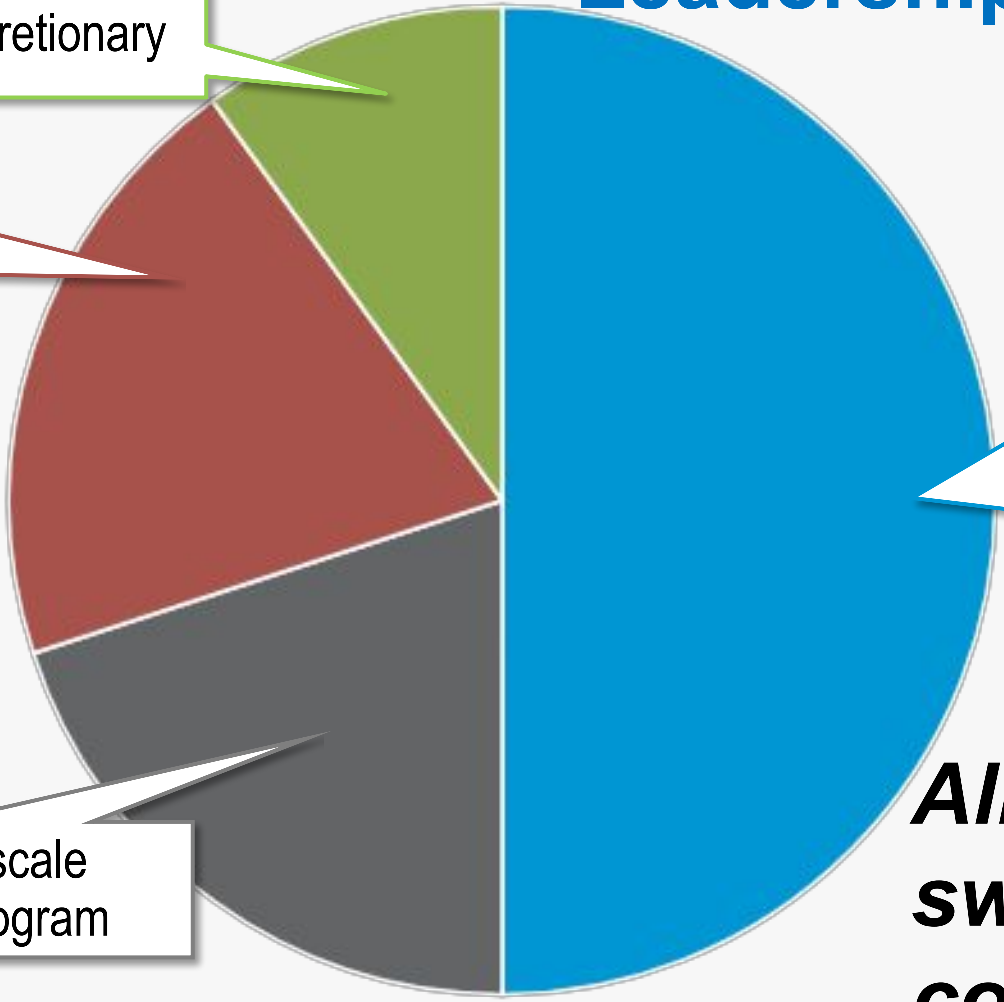
Your Science @ ALCF

Three primary ways for access to LCF

Distribution of allocable hours

Leadership-class computing

DOE/SC capability computing



Allocations switching from cores-hours to node-hours

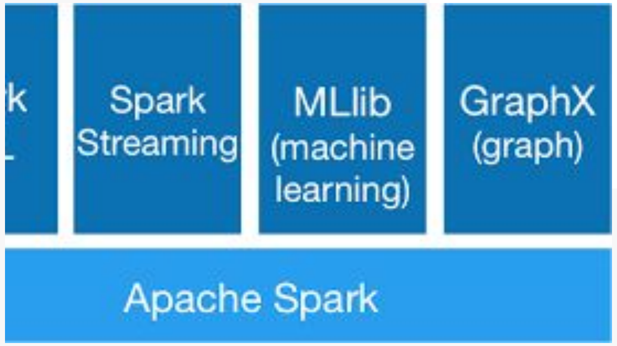
Software

- **ML/DL:**
  - **TensorFlow, Keras, Neon, MXNet, PyTorch, Sci-kit Learn, Graph Analytics (Cray Graph Engine), Horovod…**
  - **With performance libraries e.g. Intel MKL, MKL-DNN, LibXSMM etc enabled**
  - **Intel optimized Tensorflow**
    - **Conda package on Theta**
    - **Intel Distribution for Python's optimized numpy**

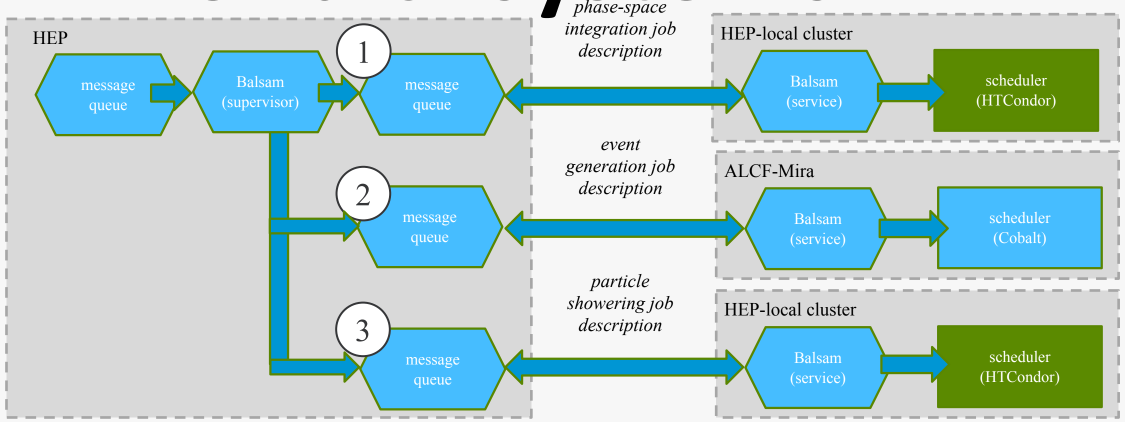
```
# Python 2.7
pip install https://anaconda.org/intel/tensorflow/1.4.0/download/tensorflow-
1.4.0-cp27-cp27mu-linux_x86_64.whl

# Python 3.5
pip install https://anaconda.org/intel/tensorflow/1.3.0/download/tensorflow-
1.3.0-cp35-cp35m-linux_x86_64.whl

# Python 3.6
pip install https://anaconda.org/intel/tensorflow/1.4.0/download/tensorflow-
1.4.0-cp36-cp36m-linux_x86_64.whl
```

Software

- **Workflow/Data analysis:**
  - Containers
    - Singularity container solution for application science workloads
    - Environment imported into container
    - mount additional directories into the container with the -B flag
    - aprun -n $RANKS -N 1 singularity exec my_image.img ./my_binary
  - Balsam
- Jupyter Hub, MongoDB, Apache Spark, R
  - Python
    - Intel and Cray modules on Theta
    - ALCF alcfpython/2.7.14-20180131
- **Visualization:** Paraview on Theta



BALSAM Workflow manager and edge serviceS for alcf systems



Schematic of ATLAS deployment of Balsam on multiple resources to execute a workflow with alternating serial and parallel stages

Balsam is a workflow manager that simplifies the task of running large-scale job campaigns on ALCF resources while minimizing user involvement and improving productivity. It interacts closely with job schedulers to optimize job throughput via individual jobs and ensemble jobs, staging data in and out as needed. A supervisor component manages execution across multiple compute resources.

https://www.alcf.anl.gov/balsam

The Impact

- Delivered **>150M Core hours for production** science on ALCF systems
- **The ATLAS experiment has used Balsam to run hundreds of millions of compute hours** of event generation jobs on ALCF systems. ***ALCF contribution to ATLAS computing*** ranks as the **6th largest country** in the world.
- The **DIII-D National Fusion Facility used Balsam** to trigger experiment-time analyses during Tokamak operation, running more complex analyses in less time, leading to higher experiment productivity
- We are investigating use of Balsam for several other projects, including real-time APS/ALCF, and in **ADSP and ECP** projects

DeepHyper: Scalable hyperparameter search for deep learning

P. Balaprakash, E. Jennings, M. Salim, T. Uram, S. Wild, V. Vishwanath

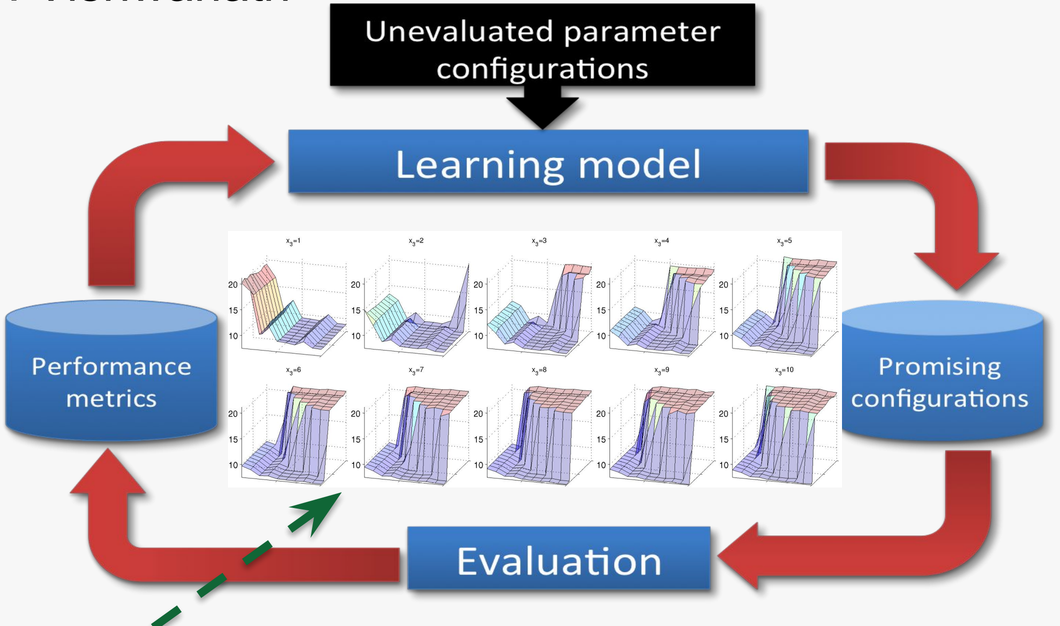
– Development of DL algorithms is time consuming and a significant portion is spend on tuning and optimizing the hyperparameters such as number of layers number of units learning rate optimizer epochs,

– Model-based search iteratively refines the model in promising input region by obtaining new outputs at unevaluated input configurations

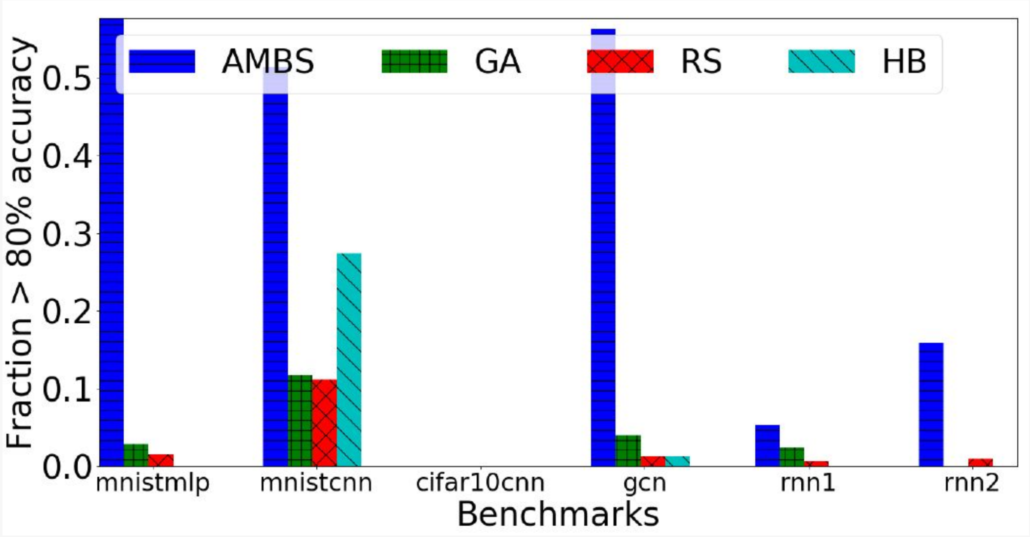
– Framework:

- Initialization phase
  - Random or Latin hypercube sampling
- Iterative phase
  - Fit model
  - Sample using the model

– Integration with the Balsam workflow enables for improved used productivity and performance. This facilitates job submission, data staging, model checkpointing, resource allocation, etc. **Scaled to 1024 Theta nodes** and 64 Cooley nodes



Example Surrogate Model Fitted to Sampled Performance (iterative refinement improves the learning model)



Map To Allocation Programs

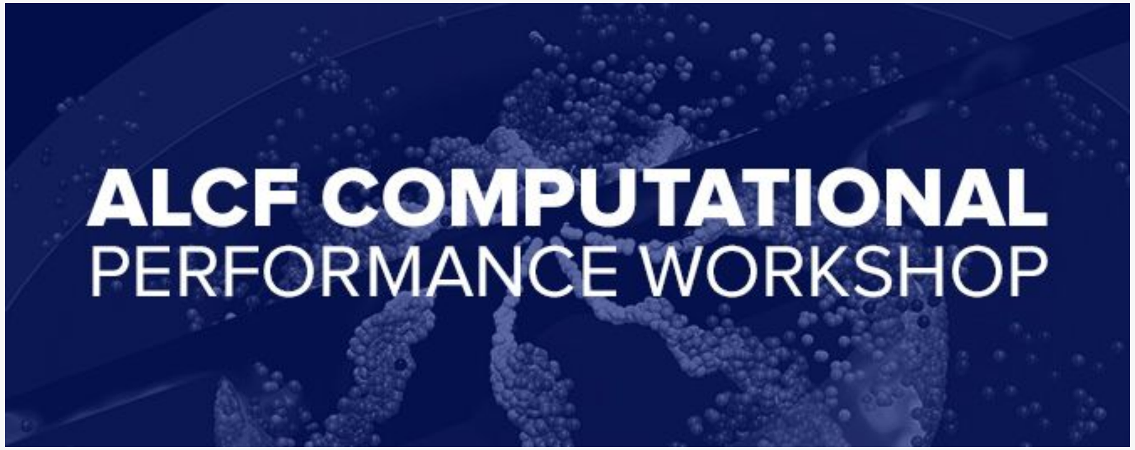
Program		
INCITE	**Production**	**Capability Computing**
ALCC	**Production**	**SC Capability Computing missions driven**
Discretionary	**Production**	**Development, testing, proposal preparation**
Data Science Program	**Production**	Developing technical and science capability for data and learning based workflows
Early Science Program	Next Generation	Developing technical and science capability for next generation systems
Exascale Computing Projects	Next Generation	The ECP mission-driven

https://www.alcf.anl.gov/user-guides/how-get-allocation

ALCF User Training Workshops and Opportunities

Most Recently: October 2-4, 2018

https://www.alcf.anl.gov/training



April 30 - May 2, 2019 Registration

Argonne Training Program on Extreme-Scale Computing (ATPESC)
https://extremecomputingtraining.anl.gov/application/

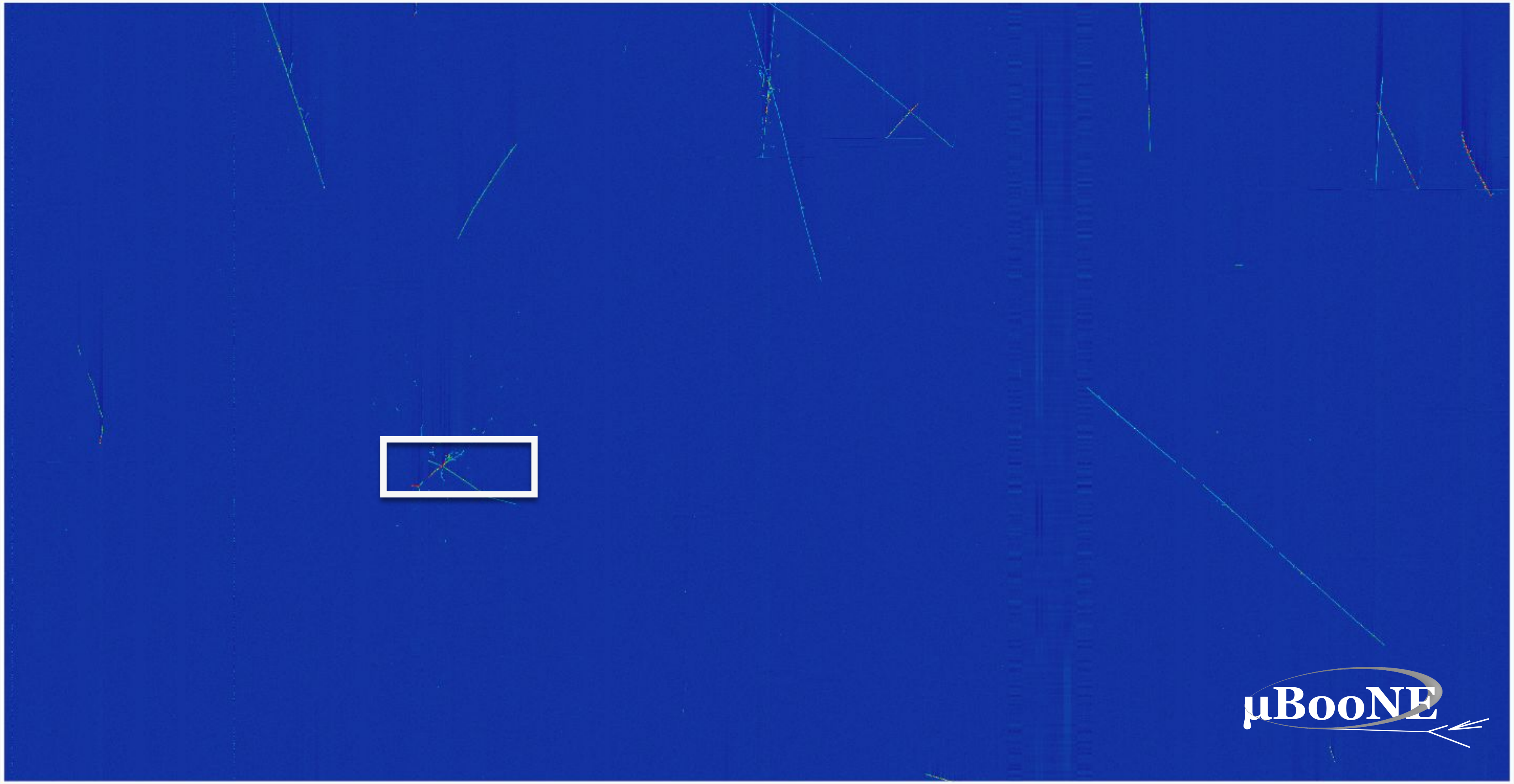
Thank You!!!

Feel free to contact us:
corey.adams@anl.gov
datascience@alcf.anl.gov

https://www.alcf.anl.gov/alcf-data-science-program

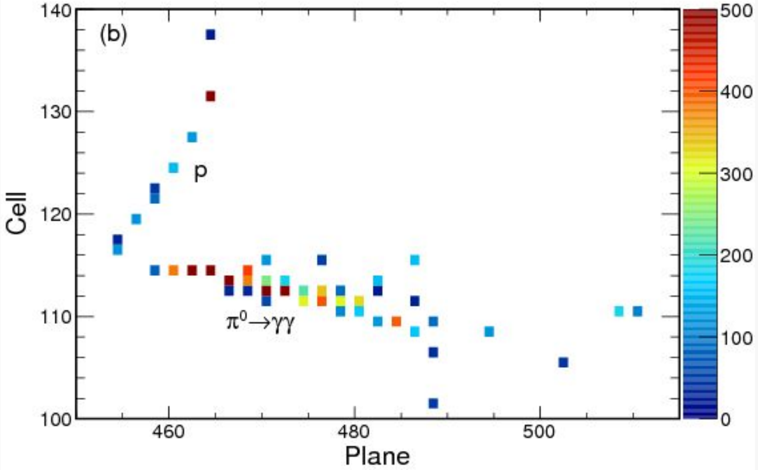
Extra Materials

Finding Neutrinos

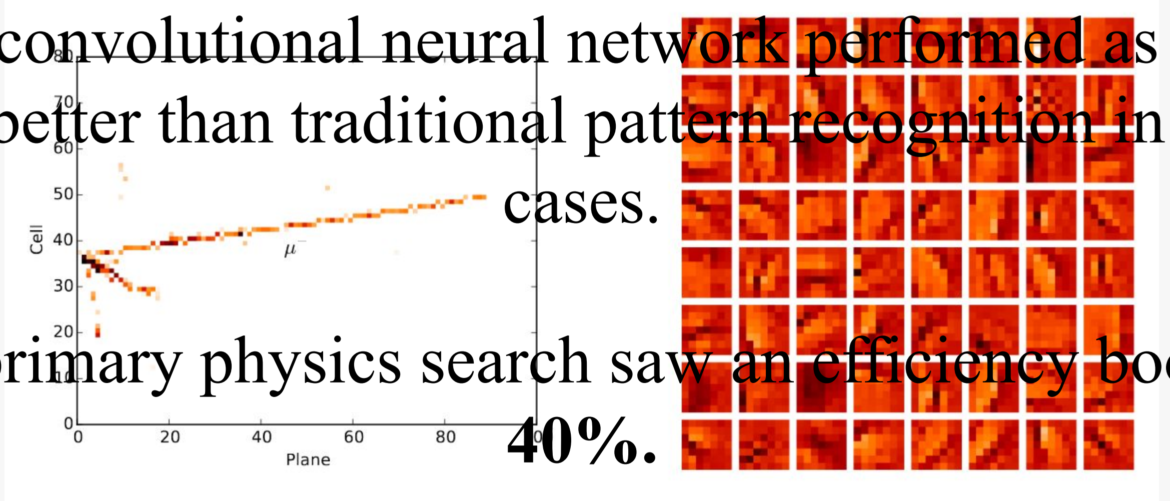


First Steps: Nova

"A Convolutional Neural Network Neutrino Event Classifier" (https://arxiv.org/abs/1604.01444)



The Nova experiment trained a modifed version of AlexNet to classify their neutrino interactions by type of neutrino.



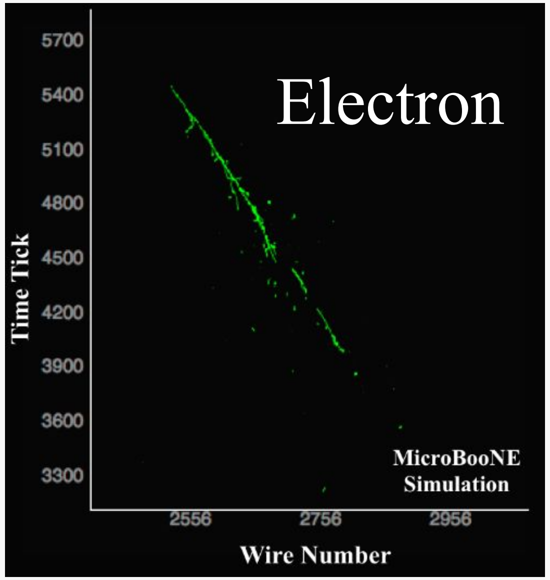
The convolutional neural network performed as well or better than traditional pattern recognition in all cases.

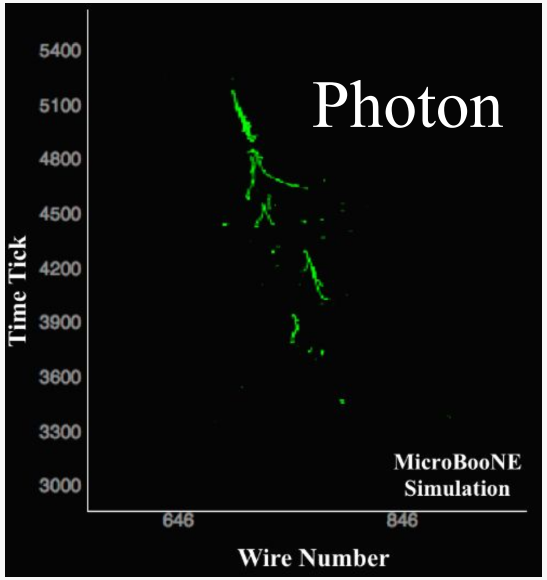
The primary physics search saw an efficiency boost of **40%.**

MicroBooNE: Particle ID

Sample	Electron	Photon	Muon	Pion	Proton
Detection Accuracy (%)	77.8 +/- 0.7	83.4 +/- 0.6	89.7 +/- 0.5	71.0 +/- 0.7	91.2 +/- 0.5
Most frequuent MisID (%)	γ (19.9)	e^- (15.0)	π^- (5.4)	μ^- (22.6)	μ^- (4.6)

Using AlexNet, can individual particles be distinguished? Yes... JINST 12, P03011 (2017).



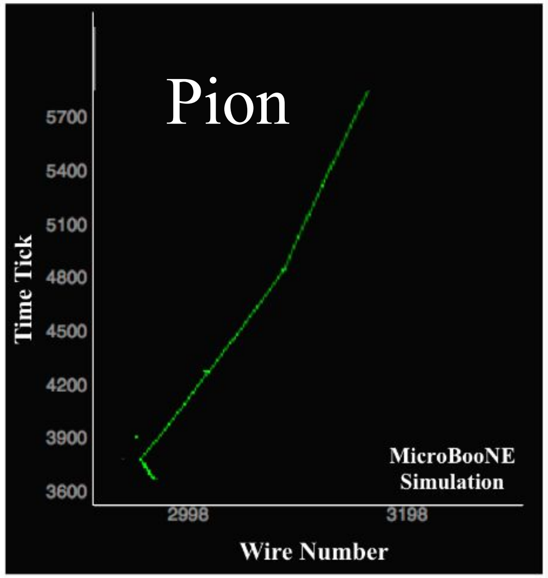


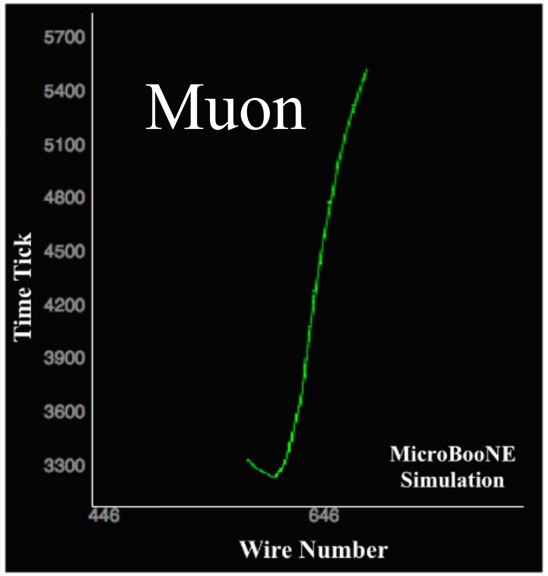
MicroBooNE: Particle ID

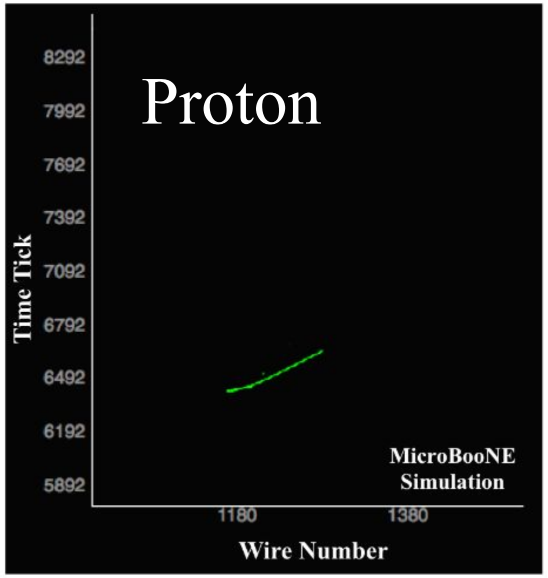
Sample	Electron	Photon	Muon	Pion	Proton
Detection Accuracy (%)	77.8 +/- 0.7	83.4 +/- 0.6	89.7 +/- 0.5	71.0 +/- 0.7	91.2 +/- 0.5
Most frequent MisID (%)	γ (19.9)	e^- (15.0)	π^- (5.4)	μ^- (22.6)	μ^- (4.6)

Using AlexNet, can individual particles be distinguished? Yes...

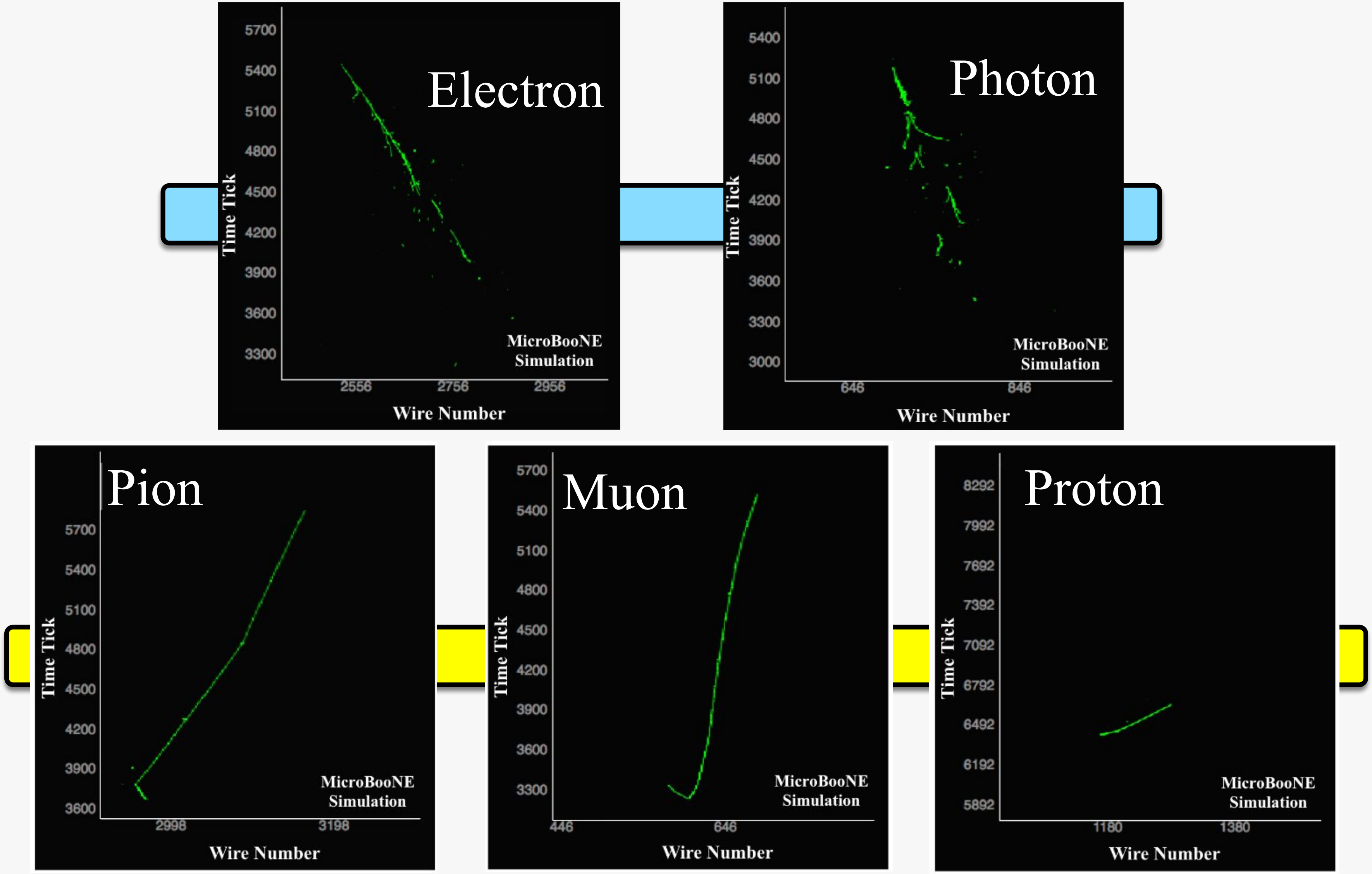
JINST 12, P03011 (2017).



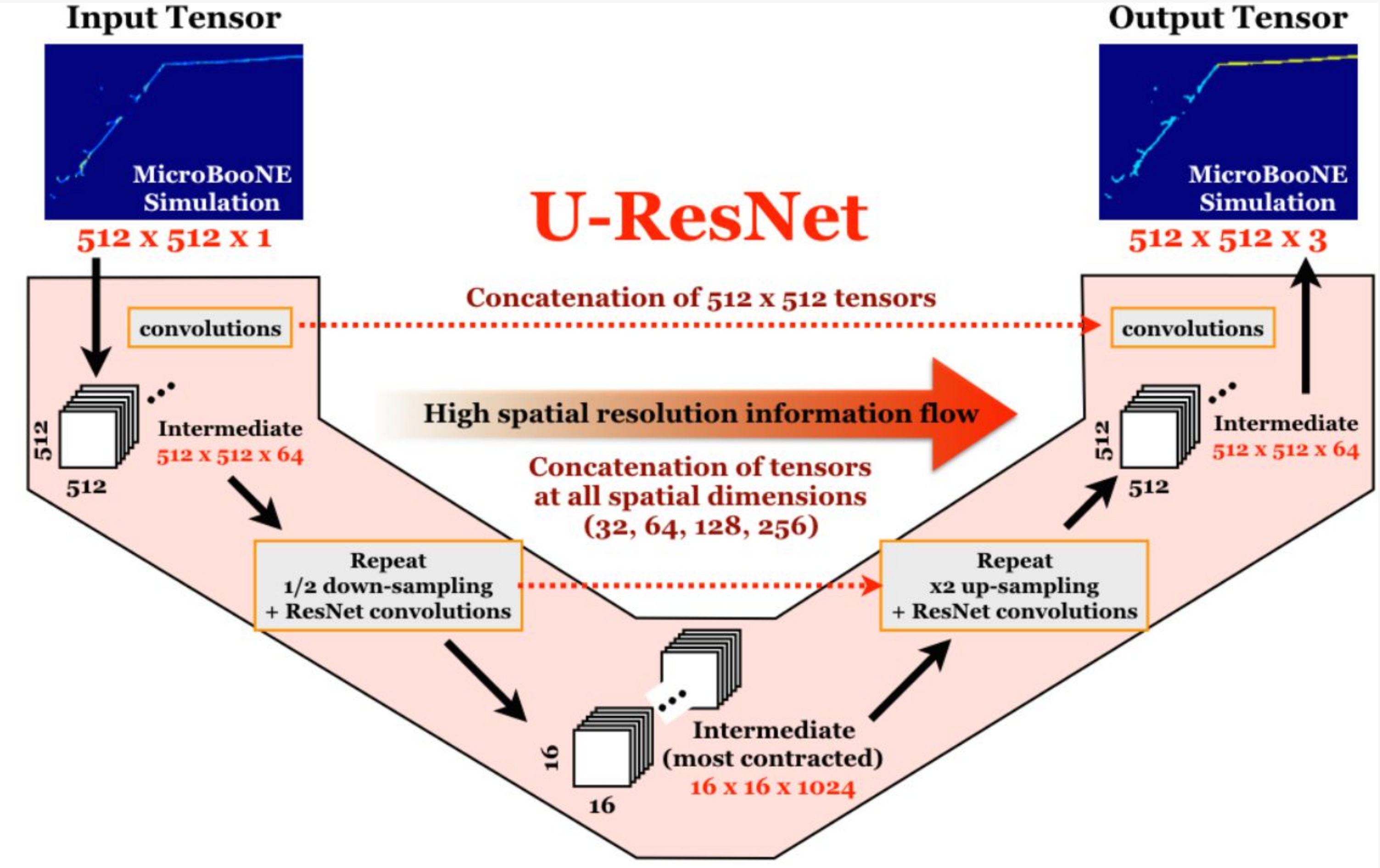




MicroBooNE: Particle ID

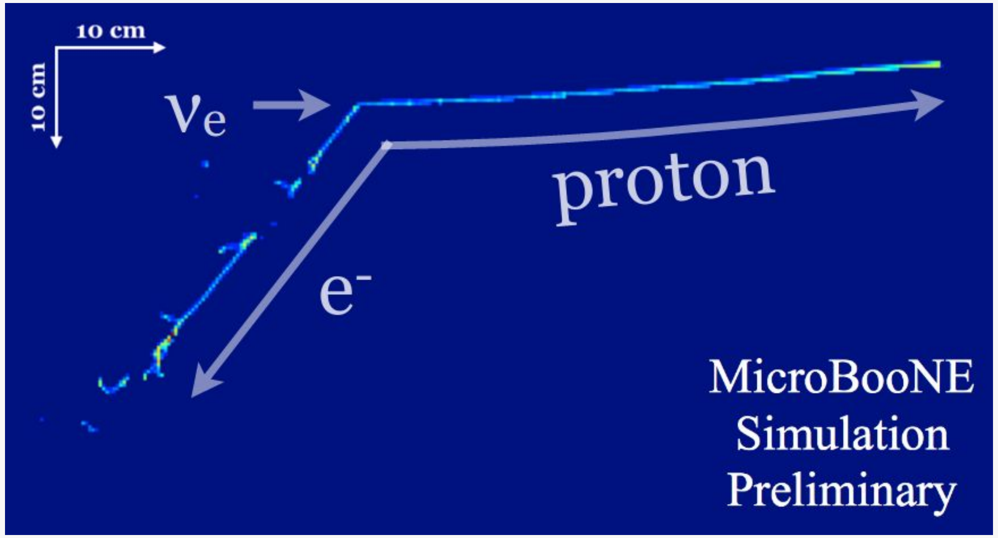


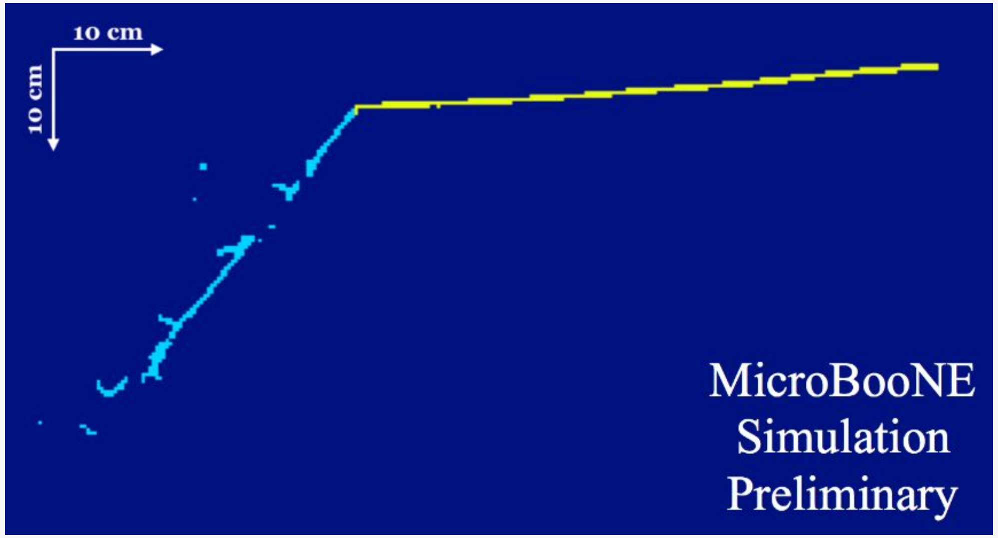
Semantic Segmentation



A Deep Neural Network for Pixel-Level Electromagnetic Particle Identification in the MicroBooNE Liquid Argon Time Projection Chamber
(https://arxiv.org/pdf/1808.07269.pdf)

Semantic Segmentation





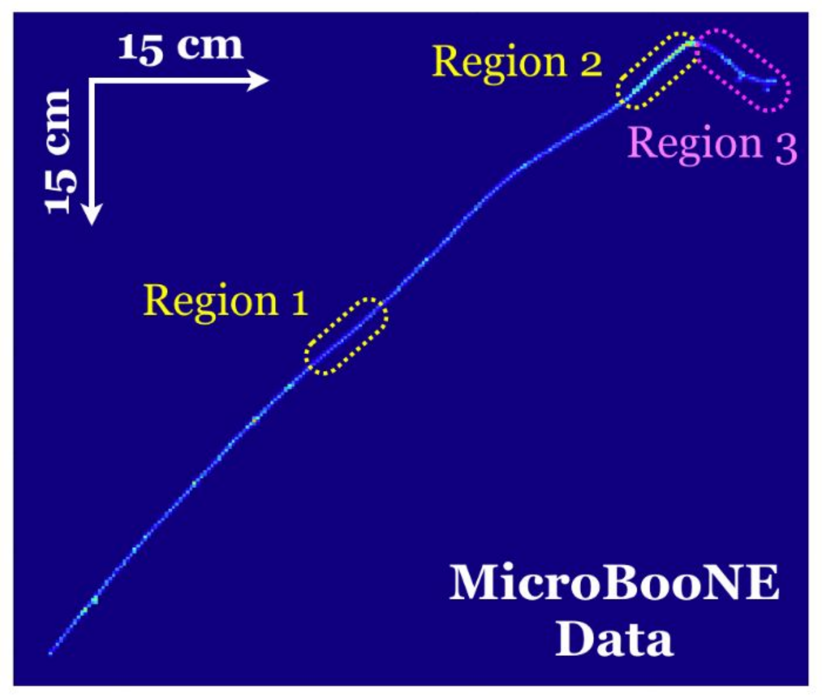
In uncrowded interactions, this is enough to augment traditional track-fitting algorithms.

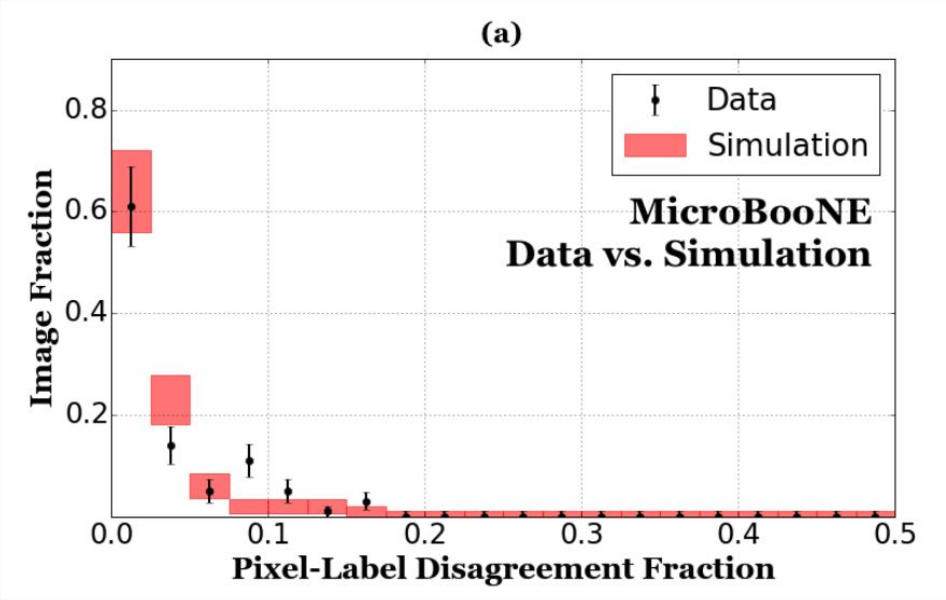
How do you evaluate performance on data?

Data Validation

Even with hand-labeling, how to find a benchmark data set?
Leverage existing physics analyses for network valdiation.

Cosmic Muon Decay Events



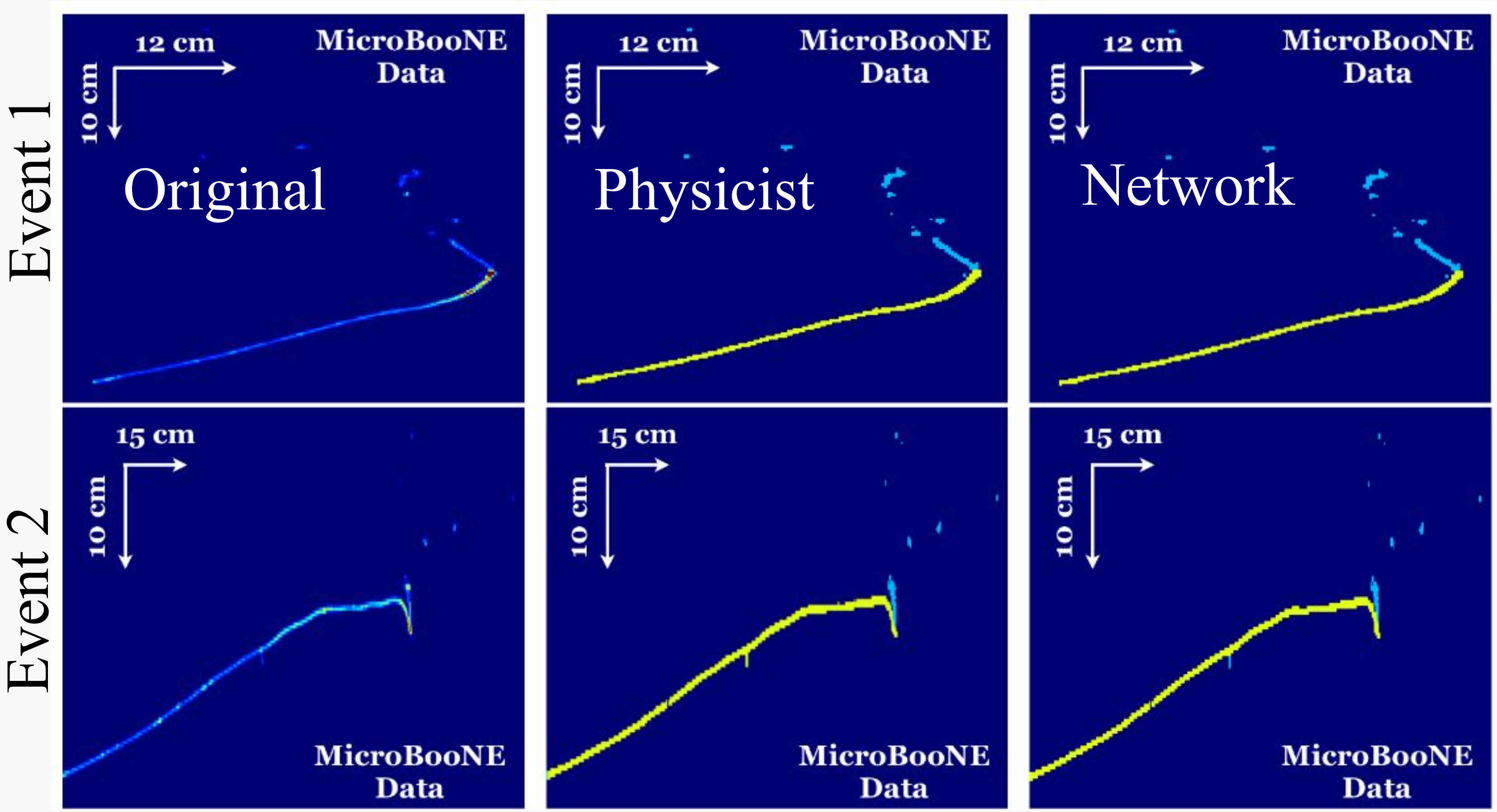


Comparison between network labels and physicist labels.

Data Validation

Even with hand-labeling, how to find a benchmark data set?

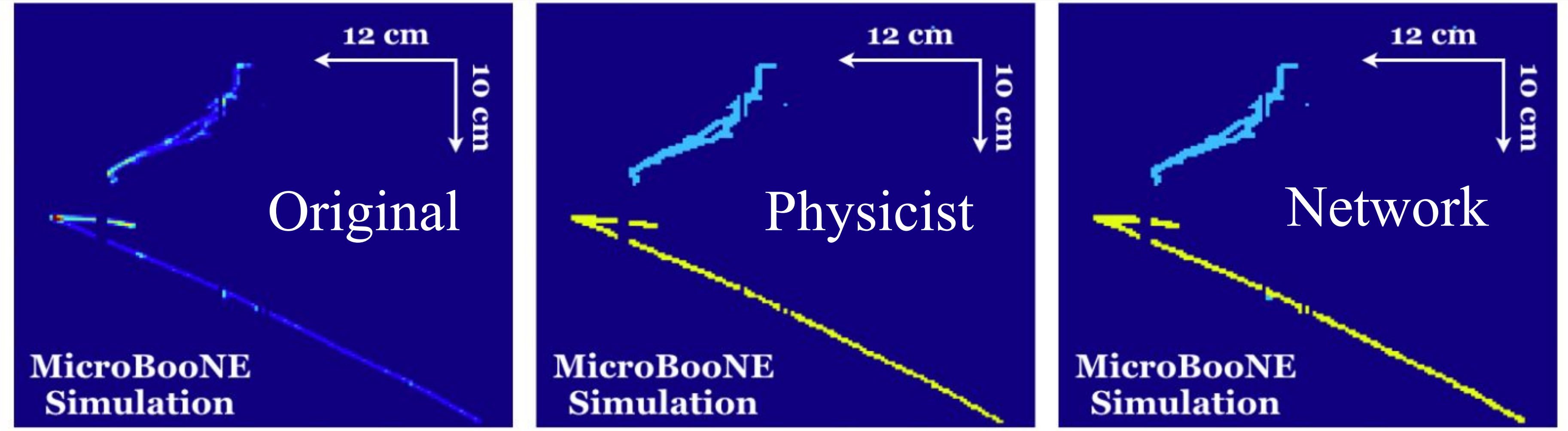
Leverage existing physics analyses for network valdiation.



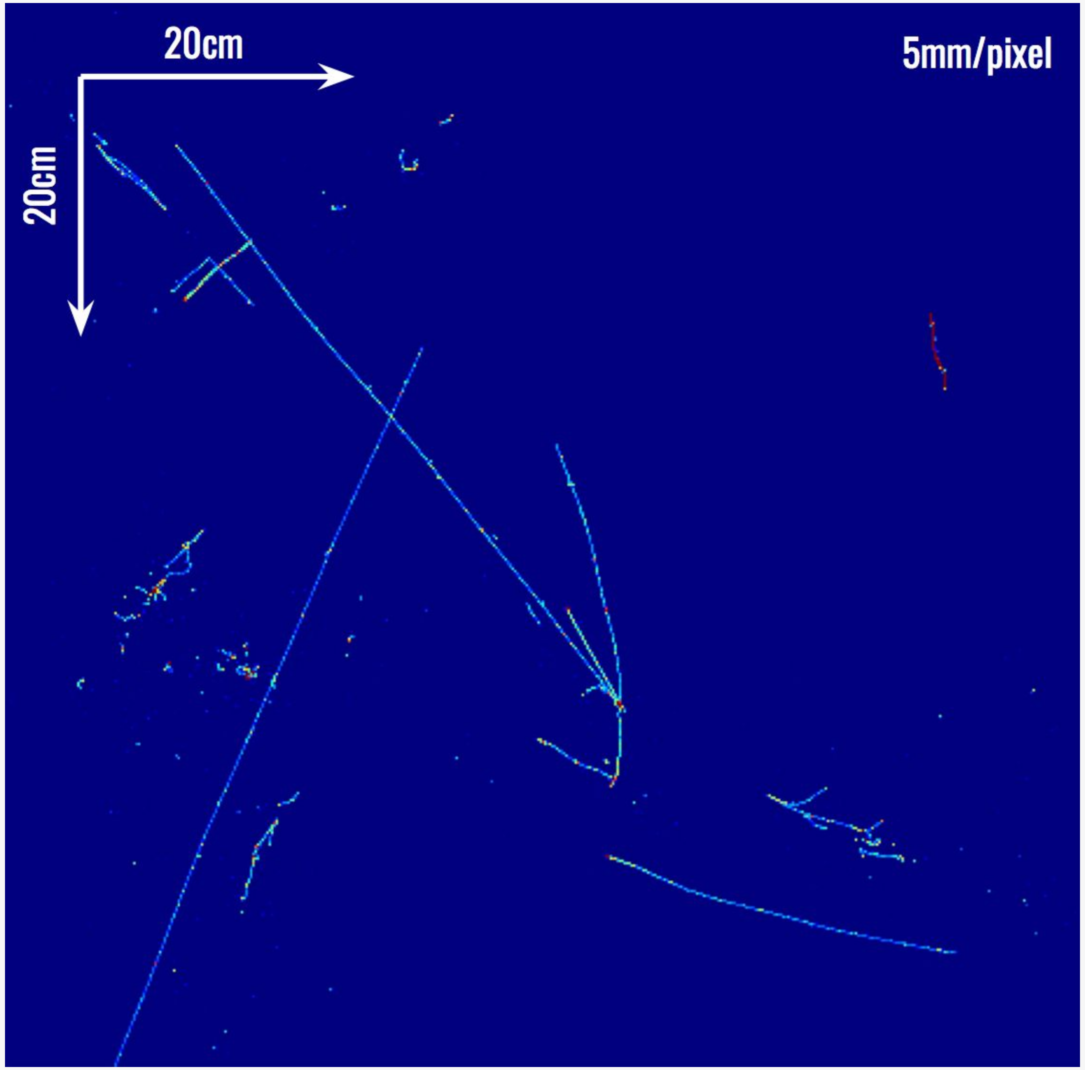
Argonne NATIONAL LABORATORY

Physicist vs. The Machine

Sample	Data	Simulation	Simulation	Simulation
Label	Physicist	Physicist	Simulation	Simulation
Prediction	U-ResNet	U-ResNet	U-ResNet	Physicist
ICPF mean	3.4	2.5	1.8	2.0
ICPF 90%	9.0	5.7	4.6	4.8
Shower	4.8	3.4	3.0	2.6
Track	2.7	2.4	2.2	2.9



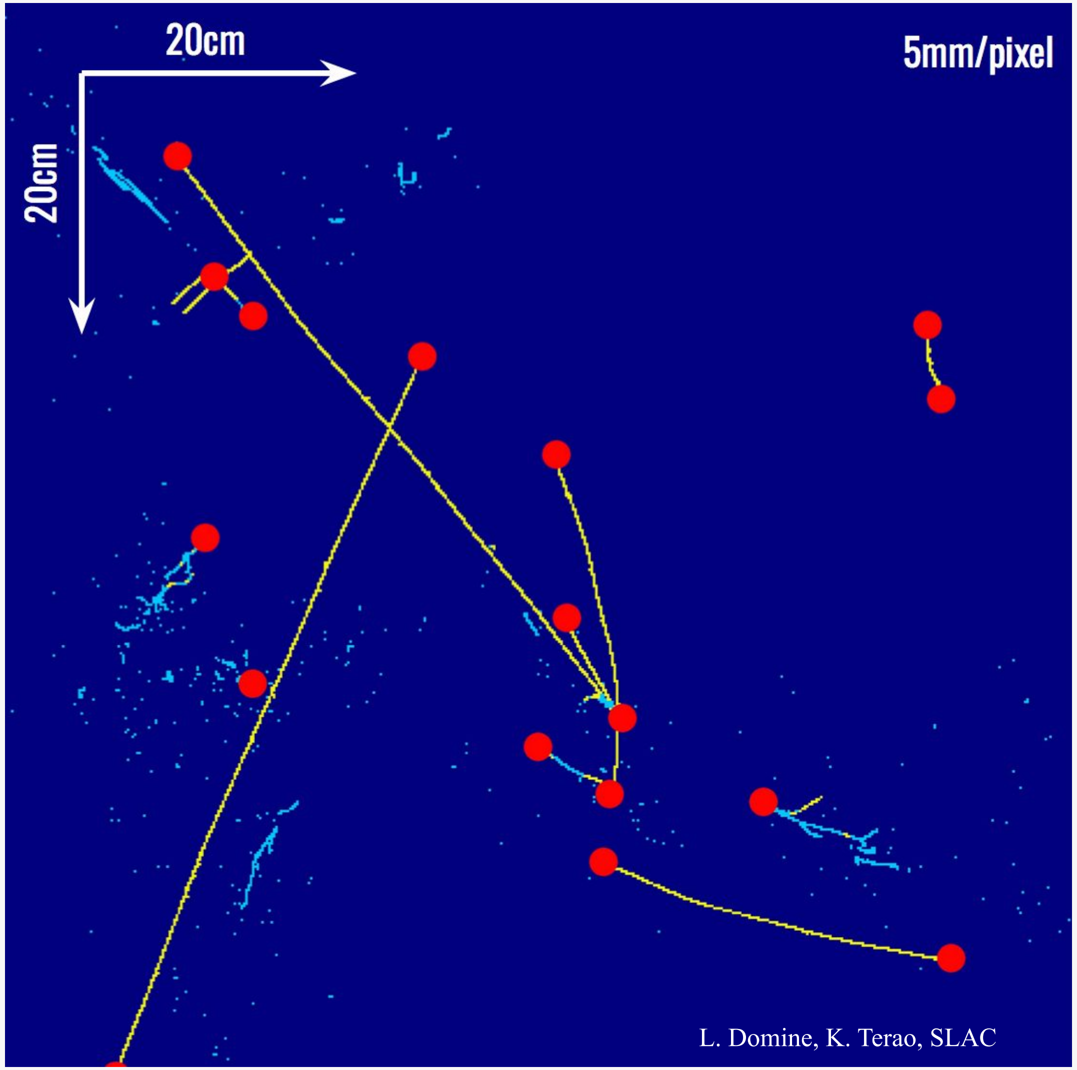
Point Prediction



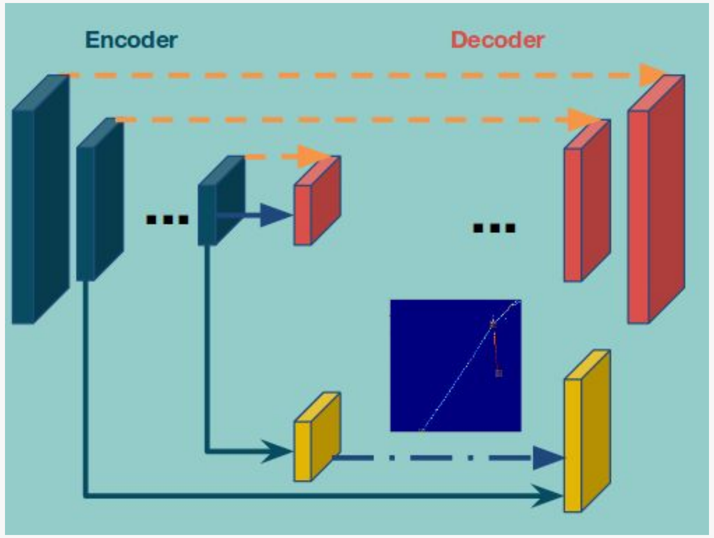
Network features for track/shower identification should also be useful for instance aware tasks.

Can a Region Proposal Network predict the start and end points of each particle?

Point Prediction



L. Domine, K. Terao, SLAC

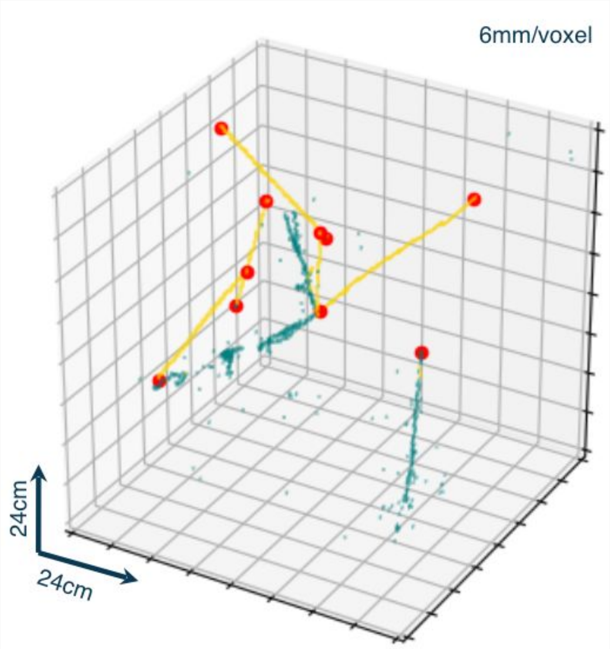


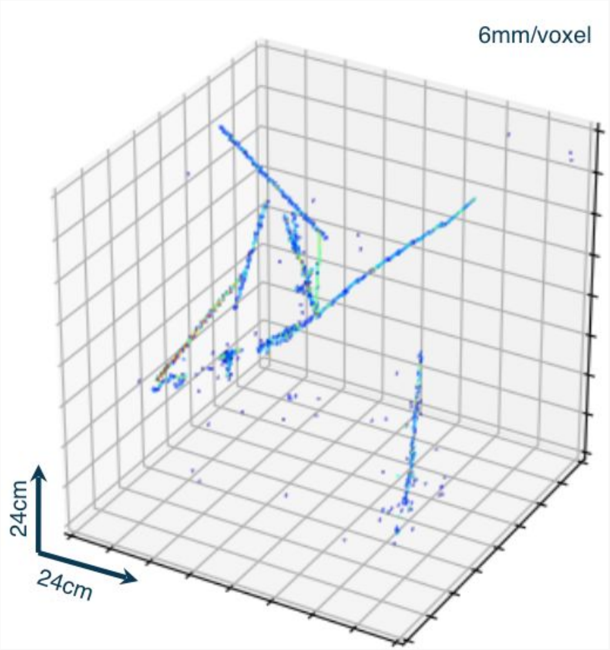
Can a Region Proposal Network predict the start and end points of each particle?

Yes

Segmentation in 3D

Meanwhile, extending segmentation networks to 3D has proven feasible and results are encouraging:





Better resolution gives better results, but GPU memory rapidly becomes an issue - difficult to train.

Convolution Implementation

Given a set of sparse input, in an N dimensional volume, the input data is represented as **m** features over **a** active locations and **a** coordinates in N dimensional space.

1. Build a hashmap to convert a spatial coordinate into an array index. Map each N dimensional active location to a single row index in an **a** x **m** matrix.
2. For each spatial location in a filter (3x3 = 9 locations, for example), create a list for each spatial (input, output) pair for that filter location.
   1. Submanifold convolutions enforce output locations to be in the input feature set, sparse convolutions allow all valid output locations
3. For input location i that maps to output location j, perform the matrix multiply of the (input row i)*(m x n) filter for that spatial location, and add it to output row j.