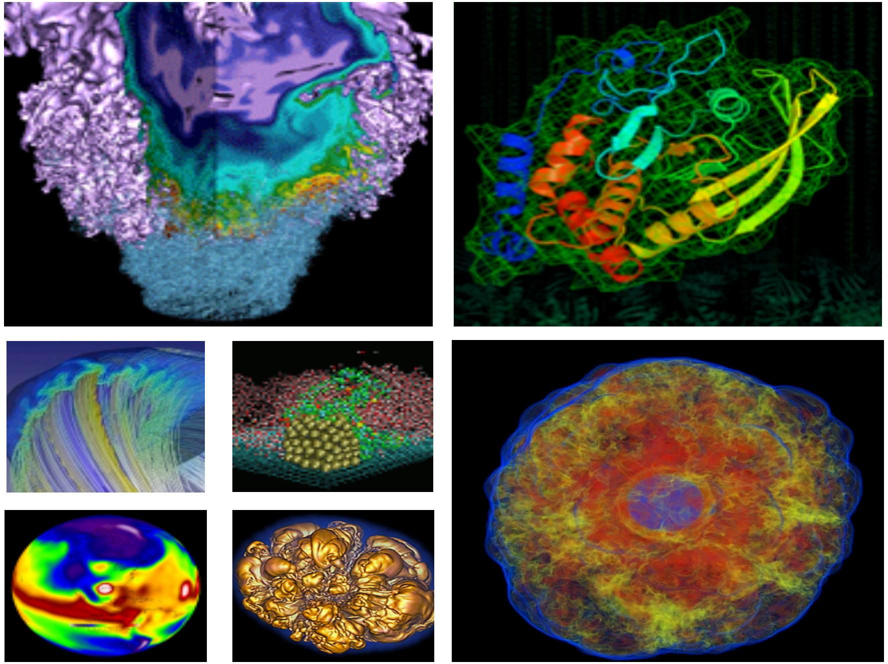
Deep learning for HEP/NP at NERSC

Jlab Machine Learning Workshop
November 6th 2018



Wahid Bhimji
(feat. Steve Farrell, Thorsten Kurth, Ben Nachman, Mustafa Mustafa, Michela Paganini, Prabhat, Evan Racah and others)
NERSC, Berkeley Lab (LBNL)

NERSC

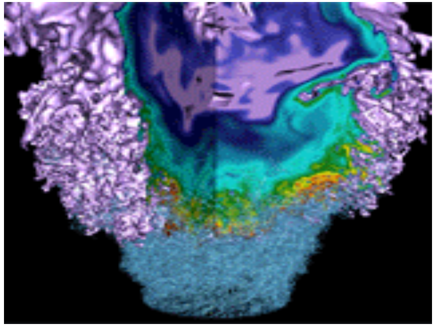
U.S. DEPARTMENT OF ENERGY | Office of Science

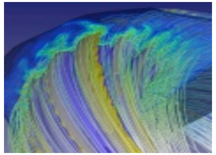
BERKELEY LAB
Lawrence Berkeley National Laboratory

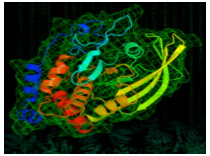
Outline

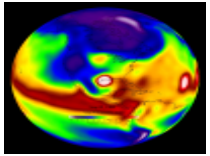
- **Introduction**
  - HEP / NP and Deep Learning
  - NERSC machines
- **Towards a Platform for Scientific Learning – Production DL stack at NERSC**
- **Examples of applications: HEP/NP Deep Learning projects at NERSC**
  - Supervised Learning: Classification with CNNs
  - Unsupervised Learning: Generation with GANs
  - Alternative representations: GraphCNN
  - Bayesian Inference with Probabilistic Programming
- **Productive DL at Scale**

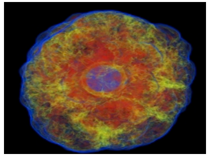
Introductions

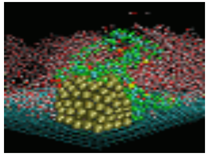




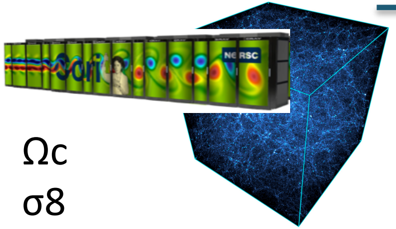
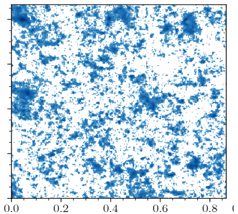






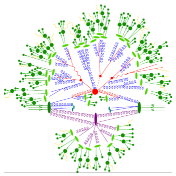
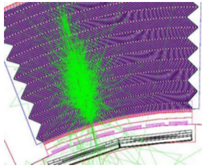


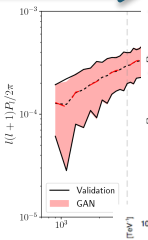
HEP/NP/Cosmology in practice

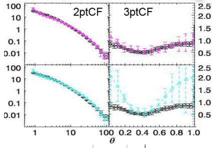



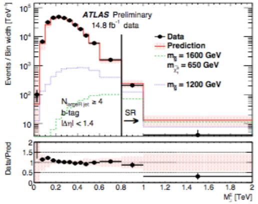
Ωc
$\sigma 8$

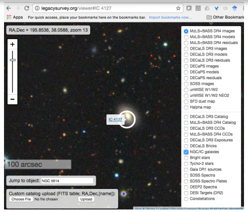
θ_{12}
m_H ...

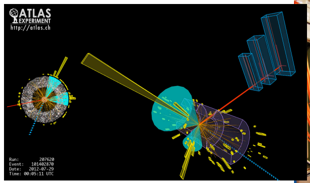



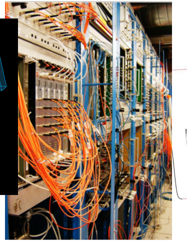
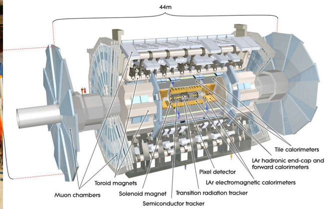










Theory into Simulations

- Cosmology: high-resolution; produce mass densities; populate with galaxies
- HEP/NP: detailed physics and detector simulation

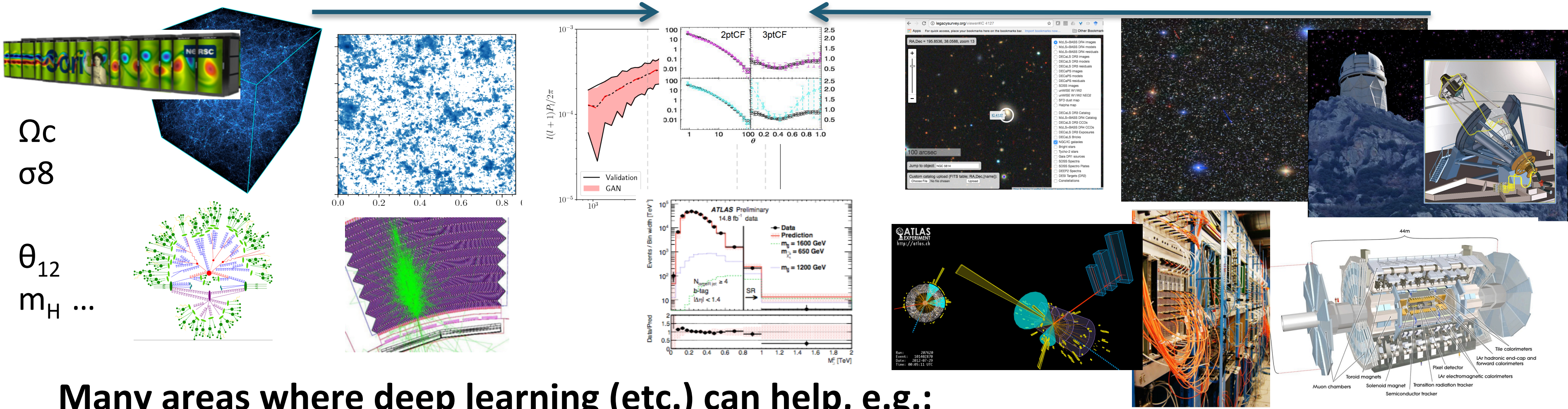
Summary statistics:

- E.g. 2pt /3pt correlation: spatial distribution
- E.g. Masses of reconstructed particles

Exp/Obs reconstruction

- Derive position of galaxies/ stars and properties for catalogs
- Reconstruct particle properties

HEP/NP/Cosmology in practice



Many areas where deep learning (etc.) can help, e.g.:

- **Classification** to find physics objects or new 'signal' events (on high dimensional data)
- **Regression** to aid reconstruction or of fundamental physics parameters
- **Clustering features** in high-dimension raw data for new physics or instrument issues
- **Generation** of data to replace simulation
- **Inference** directly of underlying physics from instrument data

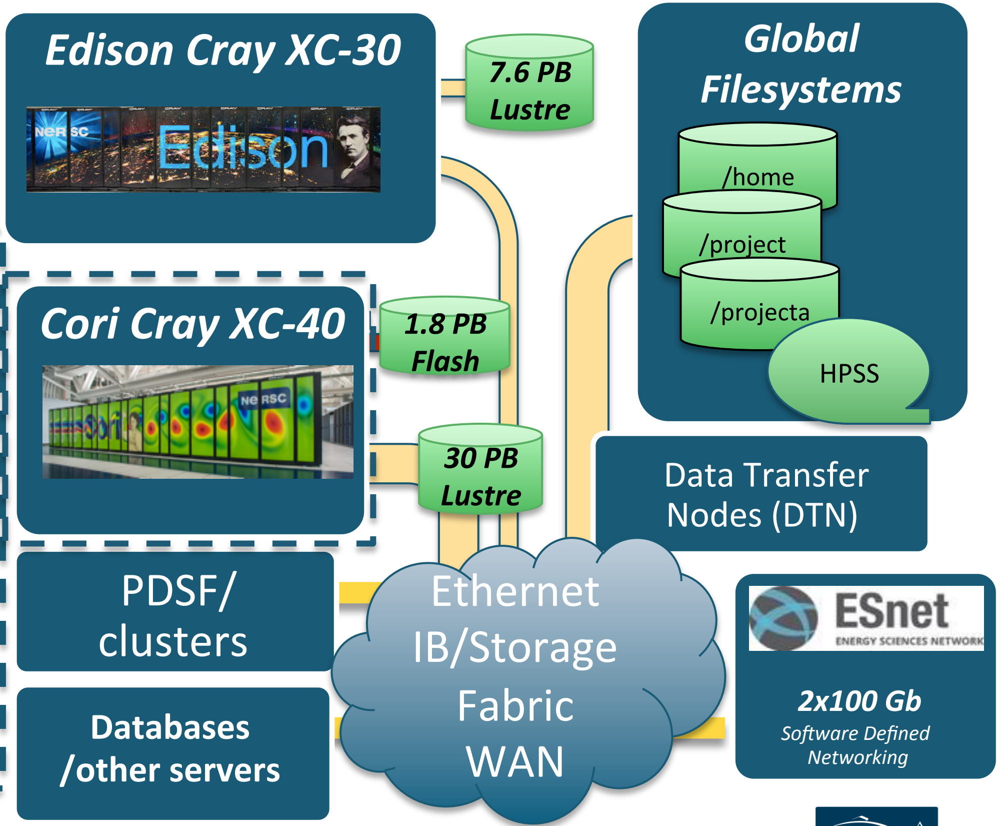
NERSC

Mission HPC center for US Dept. of Energy Office of Science:

>7000 users; 100s of projects; diverse sciences

Cori: 31.4 PF Peak –#10 in Top500

- 2388 Haswell 32-core 2.3 GHz; 128 GB
- 9668 KNL XeonPhi 68-core 1.4 GHz
  4 hardware threads; AVX-512;
  16 GB MCDRAM, 96 GB DDR4
- Cray Aries high-speed “dragonfly” interconnect
- 28 PB Lustre FS: 700 GB/s peak
- 1.8 PB Flash Burst Buffer: 1.7 TB/s

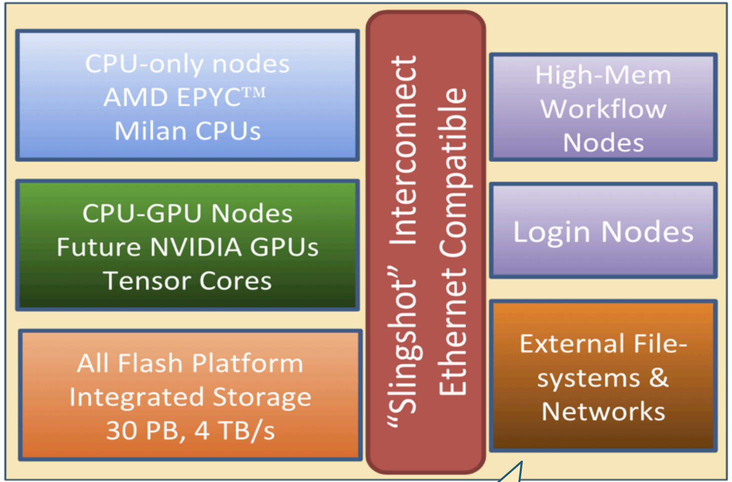


U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

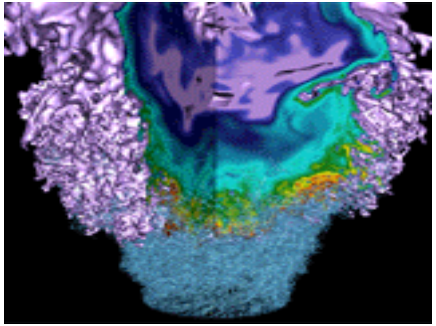
Perlmutter: A System for Science

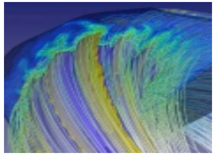
- **Cray Shasta System: 3-4x capability of Cori**
- **GPU-accelerated and CPU-only nodes meet the needs of large scale simulation and data analysis from experimental facilities**
  - >4,000 node CPU-only partition = all of Cori
  - Optimized stack for analytics/ ML at scale
  - GPU nodes: 4 NVIDIA GPUs: Tensor Cores; NVLink-3; 1 AMD "Milan" CPU
- **Cray "Slingshot": High-performance Ethernet- compatible network**
  - Capable of Terabit connections to outside
- **All-Flash Lustre based HPC file system**
  - 6x Cori's bandwidth

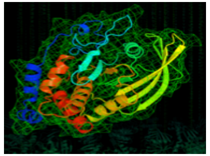


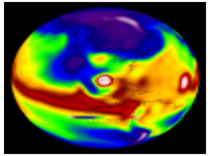
Delivery in late-2020

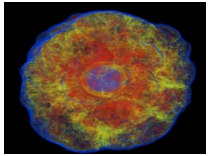
Deep Learning Production Stack at NERSC

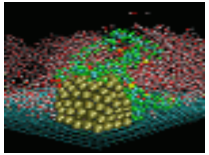








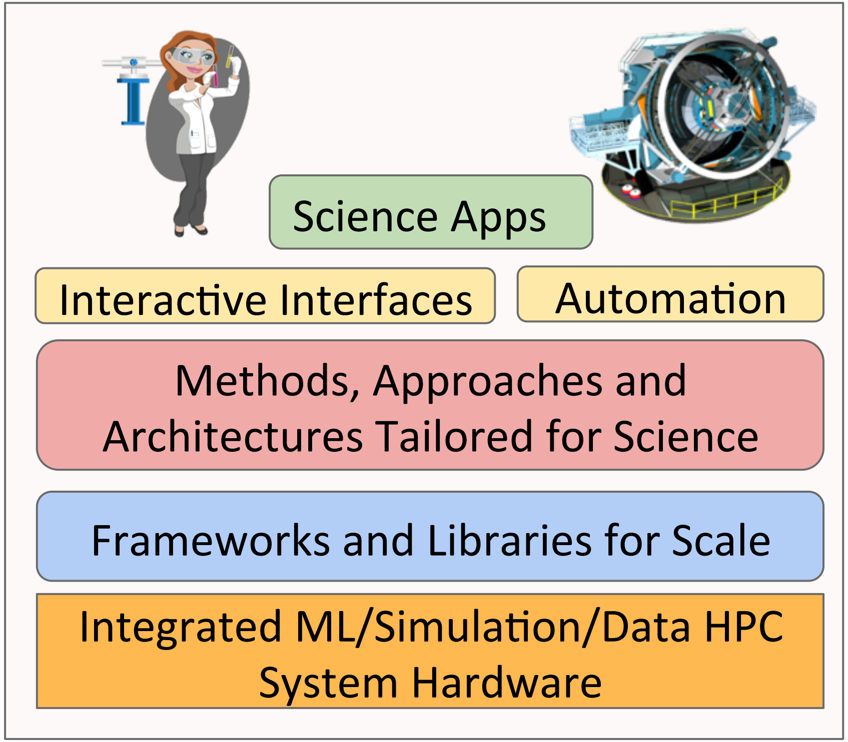




Provide a platform for scientific learning

NERSC Data and Analytics Group:

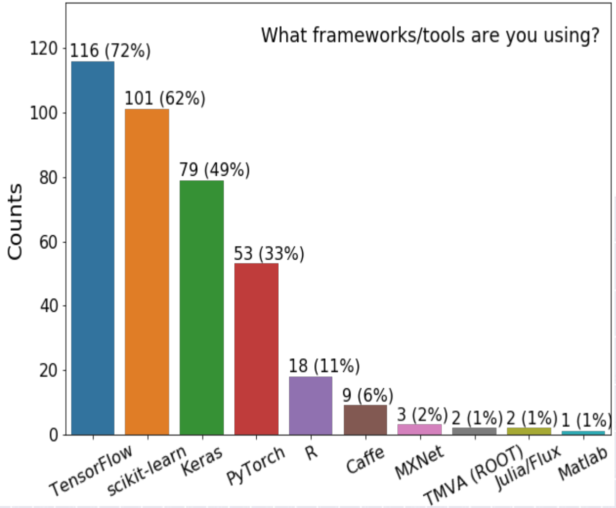
- **Provide training and tools for machine learning**
- **Optimize tools for hardware and for productivity and scale**
- **Encourage cutting-edge methods and new applications**
- **Collaborative Projects (with Scientists/ ML Researchers/ Industry)**



http://www.nersc.gov/users/data-analytics/data-analytics-2/deep-learning/

Tools

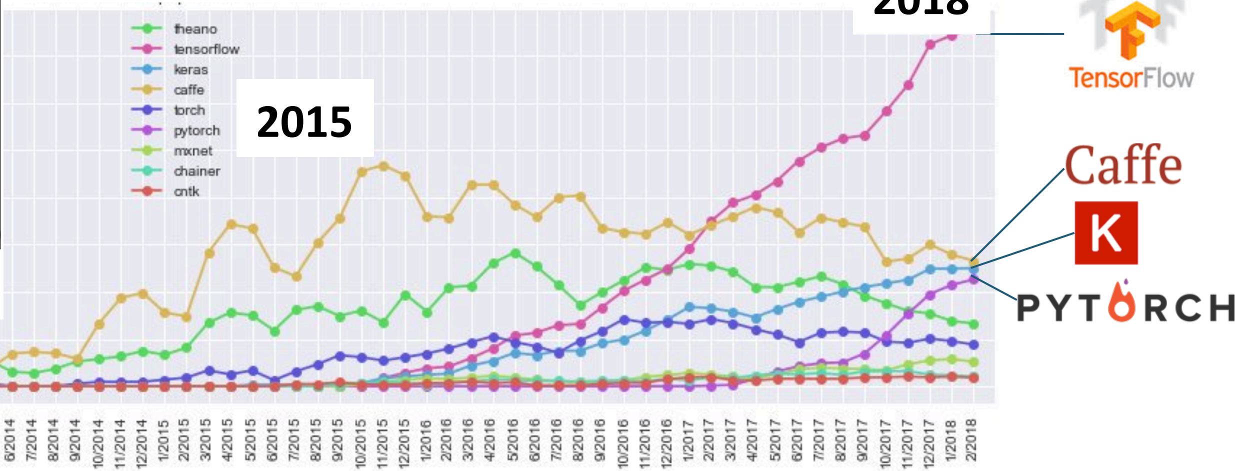
NERSC ML Survey (Now):



DL Frameworks evolving rapidly:

Caffe/Theano popular 3 years ago – now Tensorflow (TF) dominates (and Keras now in TF); Recent rise of PyTorch

Percent of ML Papers That Mention:



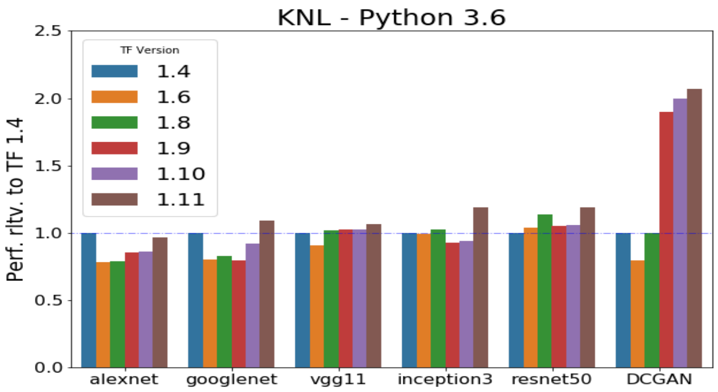
Source: https://twitter.com/karpathy/status/972295865187512320?lang=en

U.S. DEPARTMENT OF ENERGY | Office of Science

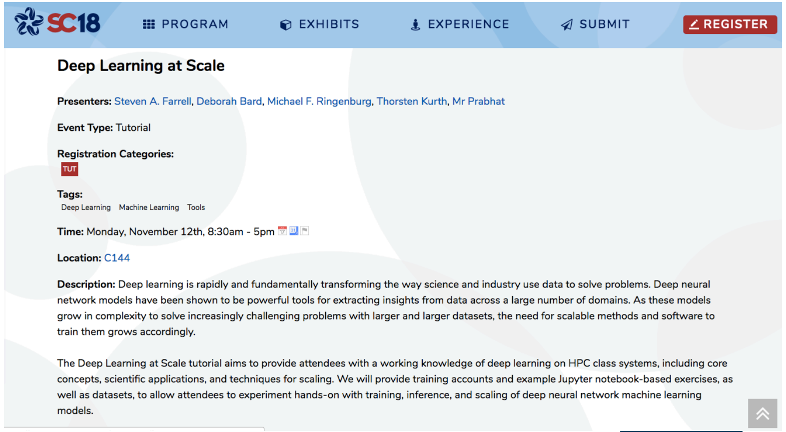
BERKELEY LAB

Tools and Training

- **Python DL frameworks rely on optimized backends to perform**
  - For CPU like Cori KNL this is Intel MKL
  - Working with Intel to improve performance for common networks (and science problems)



- **Training events for example**
  - Data day https://www.nersc.gov/users/training/data-day/data-day-2018/
  - Deep Learning At Scale at SC18 (next Monday) (with Cray Inc.)


SC18 PROGRAM EXHIBITS EXPERIENCE SUBMIT REGISTER

Deep Learning at Scale

Presenters: Steven A. Farrell, Deborah Bard, Michael F. Ringenburg, Thorsten Kurth, Mr Prabhat

Event Type: Tutorial

Registration Categories:
TUT

Tags:
Deep Learning Machine Learning Tools

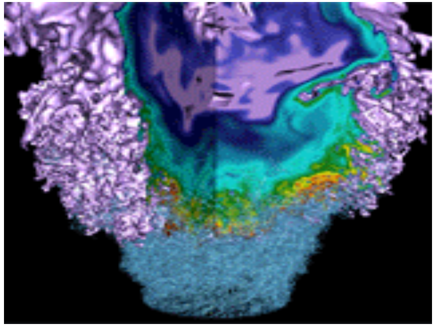
Time: Monday, November 12th, 8:30am - 5pm

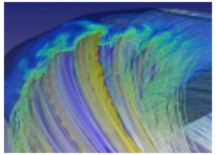
Location: C144

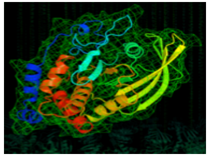
Description: Deep learning is rapidly and fundamentally transforming the way science and industry use data to solve problems. Deep neural network models have been shown to be powerful tools for extracting insights from data across a large number of domains. As these models grow in complexity to solve increasingly challenging problems with larger and larger datasets, the need for scalable methods and software to train them grows accordingly.

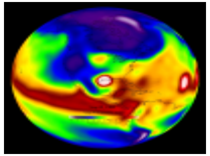
The Deep Learning at Scale tutorial aims to provide attendees with a working knowledge of deep learning on HPC class systems, including core concepts, scientific applications, and techniques for scaling. We will provide training accounts and example Jupyter notebook-based exercises, as well as datasets, to allow attendees to experiment hands-on with training, inference, and scaling of deep neural network machine learning models.

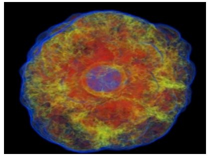
Methods and Applications

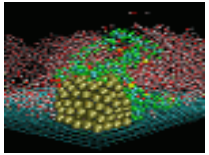






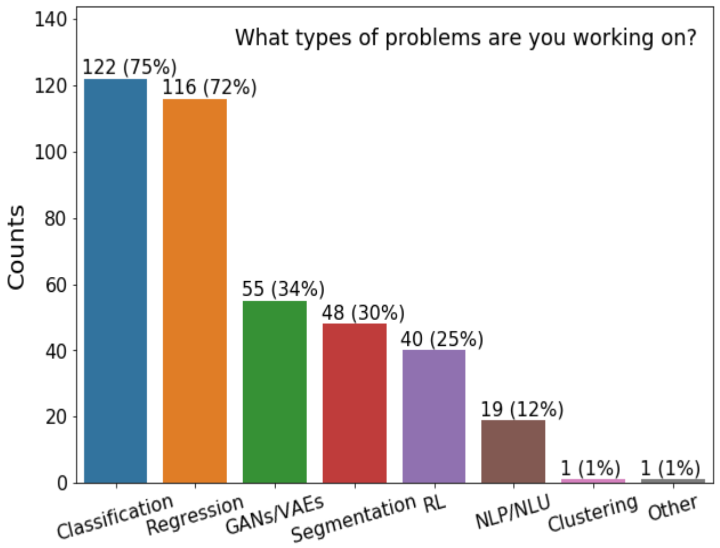




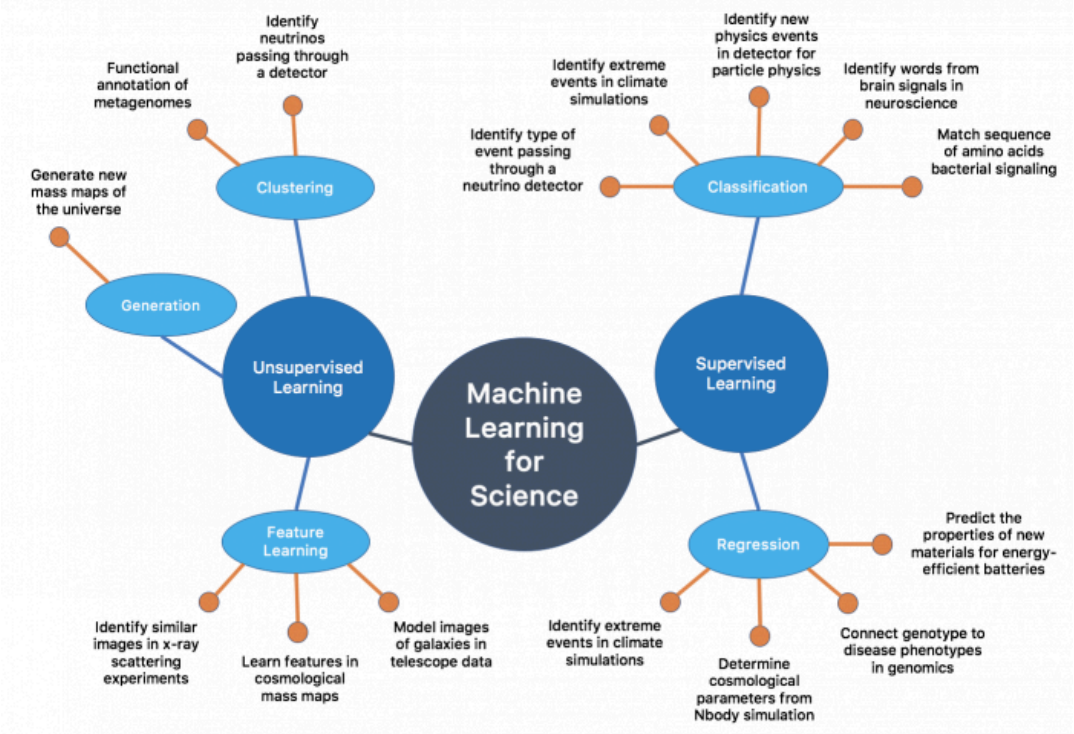


ML Applications in Science

NERSC ML Survey:



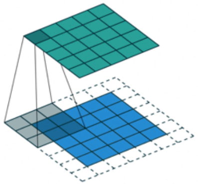
Example projects at LBL/NERSC:



U.S. DEPARTMENT OF ENERGY | Office of Science

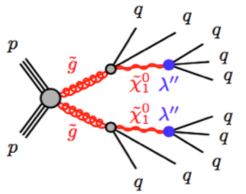
Classification with Convolutional Neural Networks

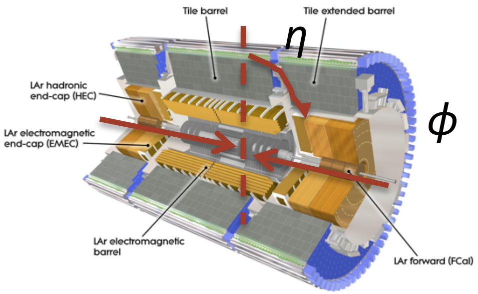
- **CNN – shared non-linear filters; reduce weights; exploit locality and symmetries: now popular in many science studies**

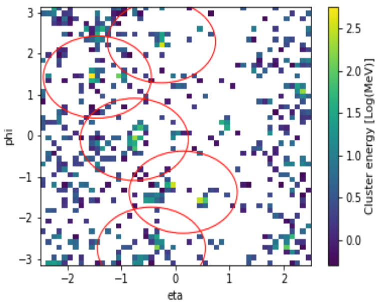


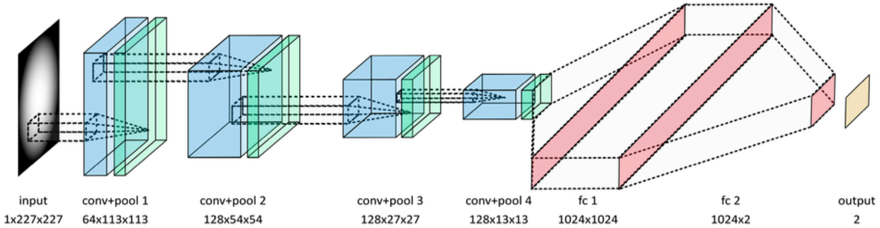
- **E.g. LHC-CNN: Unroll cylindrical detector data for image[1]; classify known (QCD) vs new physics (RPV supersymmetry)**
  - Use 3 channels for EM and HCal Calorimeters and number of tracks[2] and whole detector image 64x64 bins (~0.1 η/ϕ towers) or 224x224
  - Use our own large (Pythia+Delphes) simulated data samples
  - (3 or 4) alternating convolutional and pooling layers with batch norm.

From ATLAS-CONF-2016-057:









Bhimji, Farrell, Kurth, Paganini, Prabhat, Racah
https://arxiv.org/abs/1711.03573

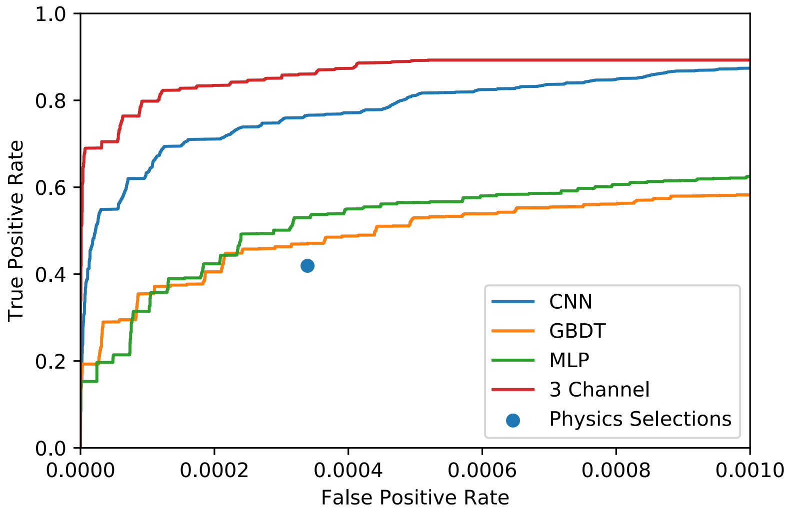
[1] As also in de Oliviera et. al. (*arXiv:*1511.05190) and others

[2] Similar to Komiske, Metodiev, and Schwartz arXiv:1612.01551

CNN performance

WB, Steve Farrell Thorsten Kurth, Michela Paganini, Prabhat, Evan Racah
https://arxiv.org/abs/1711.03573

- Use re-implementation of existing physics selections on jet variables from ATLAS-CONF-2016-057 as a benchmark
- Also compare to boosted decision tree (GBDT) and 1-layer NN (MLP)
  - Input to these jet variables used in the physics analysis (Sum of Jet Mass, Number of Jets, Eta between leading 2 jets) and four-momentum of first 5 jets

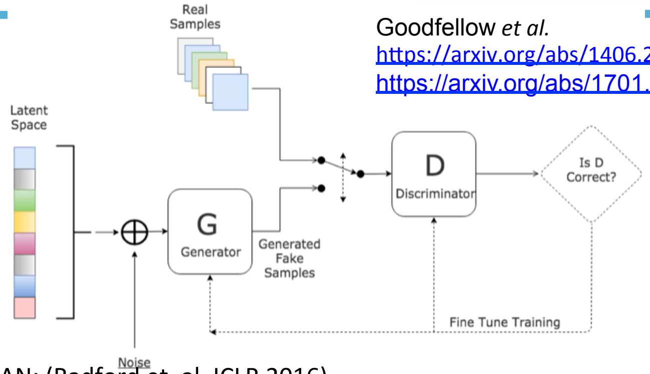


Potential to increase signal efficiency (from 0.41 to 0.77) at same background rejection as selections without using jet variables (approximate significance increase of 1.8x)

Further improvement from using 3-channels: Energy in E-Cal, H-Cal and No. tracks

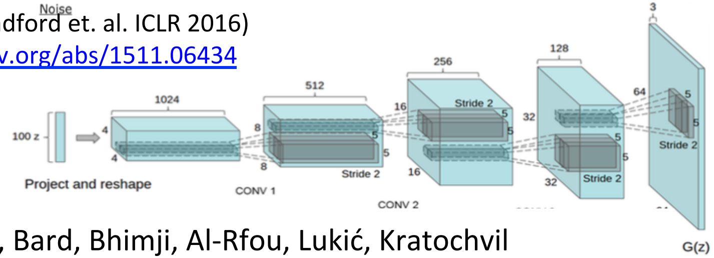
Generative Adversarial Networks GAN

- **Jointly optimize Discriminator (D) and Generator (G) NNs:** Loss for G/D in opposition
- **GANs can be unstable, science problems can have advantages:**
  - Underlying structure
  - Existing simulation samples and metrics
- **CosmoGAN: Cosmology simulations extremely computationally expensive:**
  - One product is weak lensing convergence maps, to compare to observations
- **Augment simulations with generative NN**
  - Train using existing simulation maps

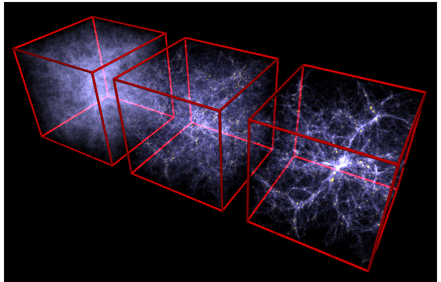


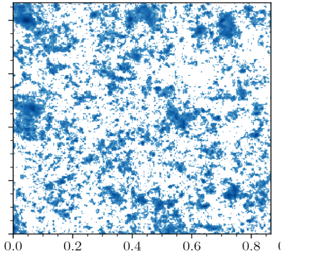
Goodfellow *et al.*
https://arxiv.org/abs/1406.2661
https://arxiv.org/abs/1701.00160

DCGAN: (Radford et. al. ICLR 2016)
https://arxiv.org/abs/1511.06434



Mustafa, Bard, Bhimji, Al-Rfou, Lukić, Kratochvil
https://arxiv.org/abs/1706.02390

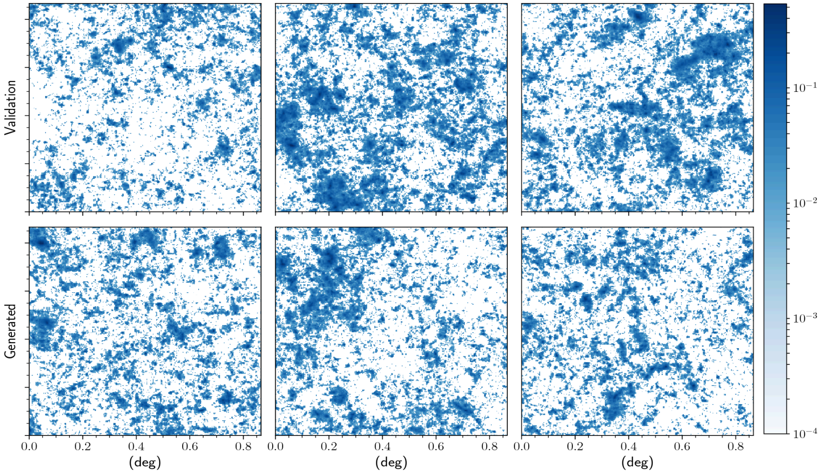


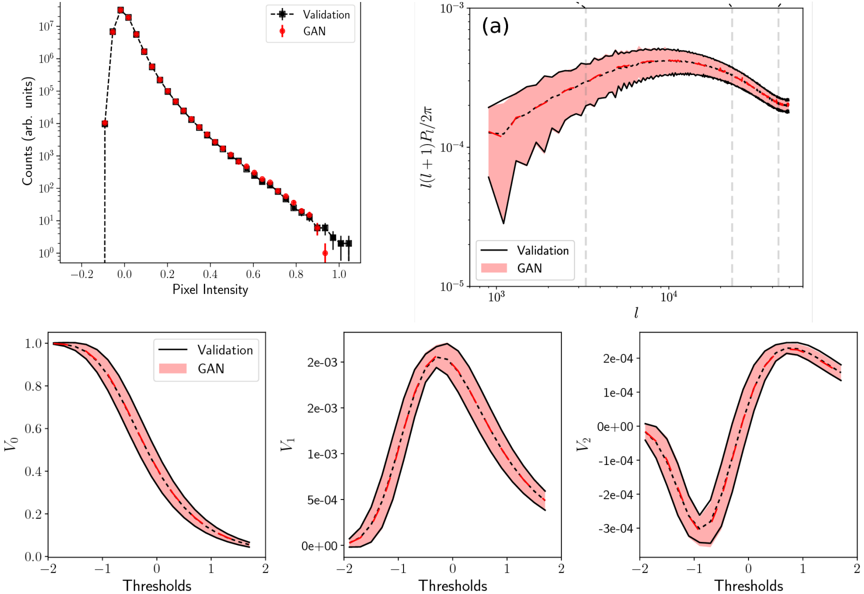


CosmoGAN

Mustafa et. al
https://arxiv.org/abs/1706.02390

- **GAN maps indistinguishable by eye**
- **Calculate power spectrum for generated images and validation sample**
  - Fourier transform of 2pt correlation
  - Excellent agreement (K-S p_value > 0.995 for 246/248 moments)
  - GAN not explicitly trained to reproduce these distributions
  - Also higher-order *Minkowski functionals* are reproduced

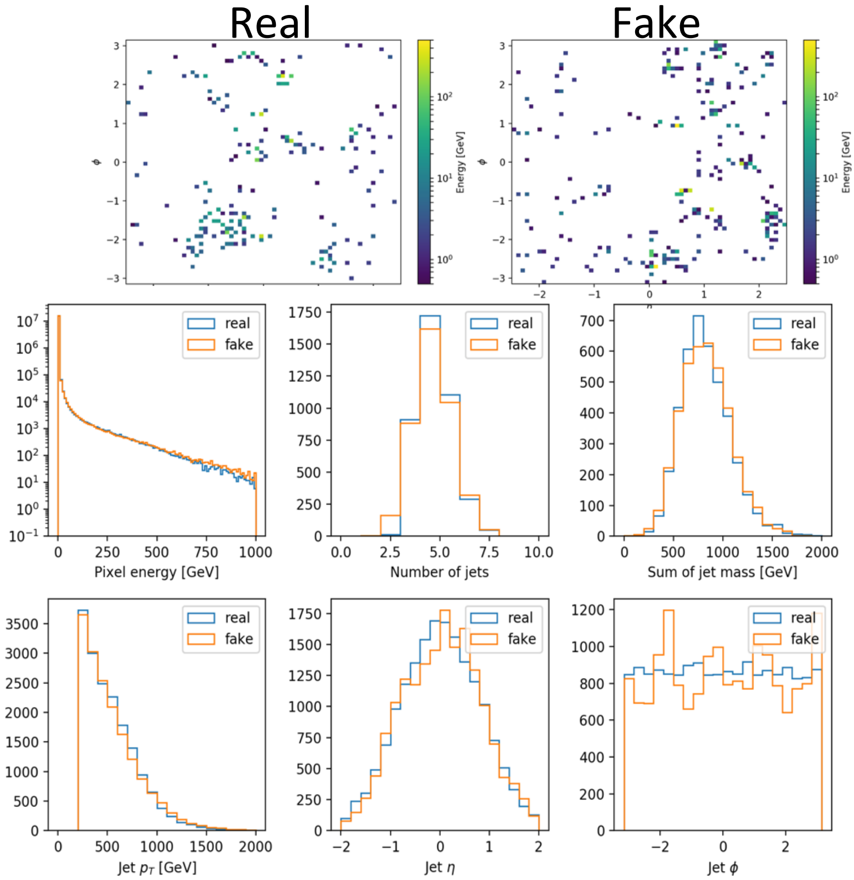




Full HEP detector GAN

Steve Farrell, Wahid Bhimji, Ben Nachman, Harley Patton and others: CHEP 2018
(See also CaloGAN Paganni , de Oliveira, Nachman https://arxiv.org/abs/1712.10321)

- **Train on full detector images (same data as LHC-CNN)**
- **Architecture: DCGAN with 4 conv + 1 dense layer in G and D networks**
  - Images reconstructed with FastJet (R=1, pt>200GeV); *Kolmogorov-Smirnoff* KS metric used to select best model and epoch in *random hyper-parameter search*
- **The generated samples produce realistic jet multiplicities and kinematics - without imposing physics knowledge**

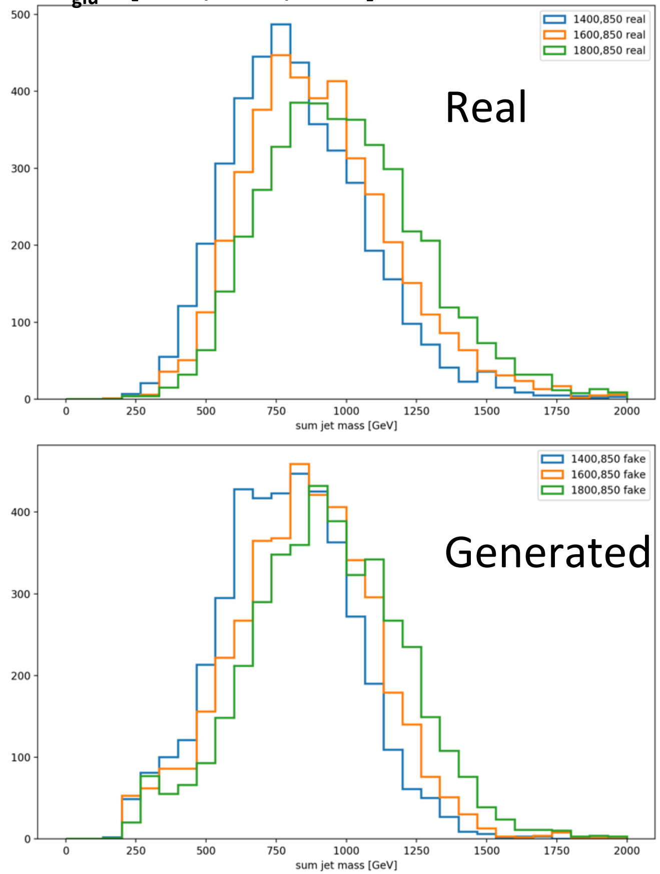


Conditional RPV GAN

Steve Farrell, Wahid Bhimji, Ben Nachman, Harley Patton and others
CHEP 2018

- **Can the GAN learn to produce images conditional on the SUSY theory parameters?**
  - We conditioned on M^{glu},M^{neu} by augmenting the discriminator (as a channel) and generator (as a input with latent vector)
- **The GAN is shown to learn the conditional distributions**
  - E.g., summed jet mass shifts as expected
- **Could ultimately use this to supplement full simulation in MC signal grids**
  - Coarse full-sim grid
  - Interpolate with GAN

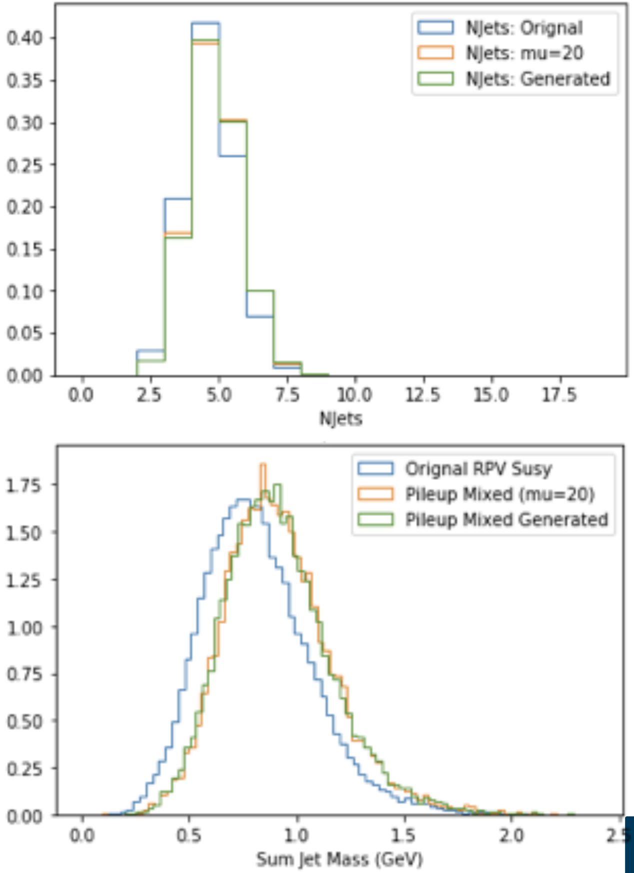
M_{glu} = [1400, 1600, 1800] GeV



Pileup GAN

Steve Farrell, Wahid Bhimji, Ben Nachman, Harley Patton and others
CHEP 2018

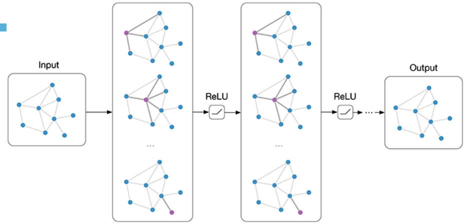
- **Pileup poses big challenges for HL-LHC computing workflows -** simulate and store a large volume of pileup events; read from disk during digitization
- **The distribution can be modeled with a whole-detector GAN: simulate samples and train model – use for fast, on-the-fly pileup sampling**
- **To test fidelity, evaluate the effects on reconstructed object kinematics:** overlay real pileup or generated pileup compare the shifts in the distributions

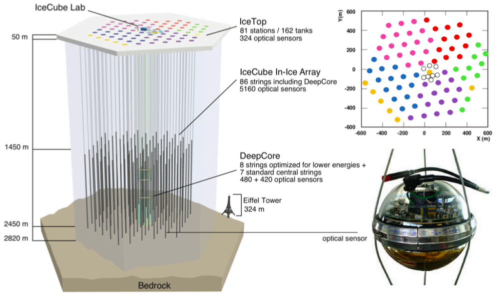


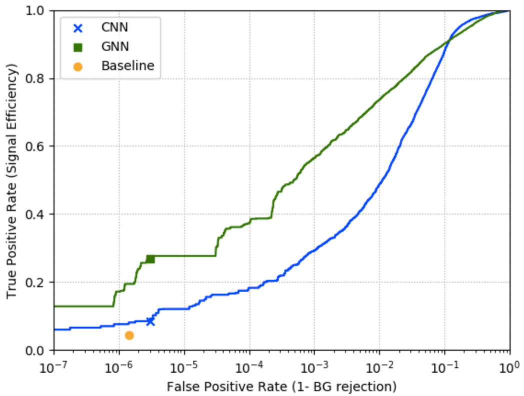
Graph CNNs

Nick Choma, Joan Bruna (NYU); Federico Monti, Michael Bronstein (ICS, U. Svizzera); Spencer Klein, Tomasz Palczewski (LBL Physics); Lisa Gerhardt, Wahid Bhimji (NERSC)
https://arxiv.org/abs/1809.06166

- **Use detector deposits rather than an image in a GraphCNN:** Represent signals as nodes of a graph with similarity as edge weights
- **E.g. IceCube: Classify neutrino signal vs cosmic rays**
  - Deal with non-uniform detector
  - Avoid sparse images
- **Graph vertices are active sensors (DOMs) in event and edges learned function of coords**
  - Adjacency matrix: gaussian kernel of DOM distance
  - Graph ‘Convolution’ and pooling analogous to CNN
- **Compared with ResNet-18 3D CNN with data on grid and physics baseline (tuned cuts on stochasticity)**





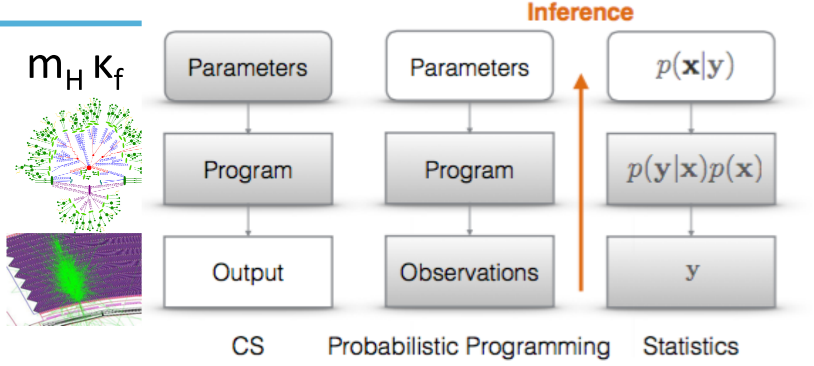


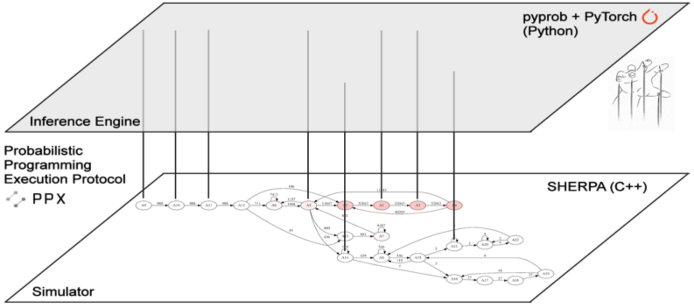
U.S. DEPARTMENT OF ENERGY | Office of Science

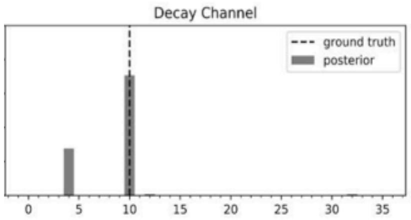
Probabilistic Programming:

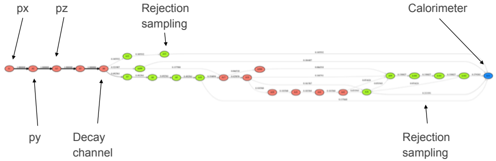
Atilim Gunes Baydin, Bradley Gram-Hansen (Oxford) Lukas Heinrich, , Kyle Cranmer (NYU) Wahid Bhimji, Prabhat (NERSC) Gilles Louppe (Liege), Lei Shao (Intel), Frank Wood (UBC) https://arxiv.org/abs/1807.07706

- **In HEP/NP often have detailed simulation (forward model) of physics and detector**
  – Ideally could 'invert' this to perform inference on real data – not easily done
- **'Invert' via probabilistic program (PPL) and embedding approach**
  – PPL: ***Sample*** from distribution (already in HEP sim. E.g. SHERPA) and ***Condition*** on observation (we add via *PyProb* without changing SHERPA)
  – Inference Compilation (IC): NN for inference
- **Initially applied to tau decay: predict particle decay channel; momentum etc. with full posterior and code traces**
  - Deep interpretability of particle decay chain and detector interactions

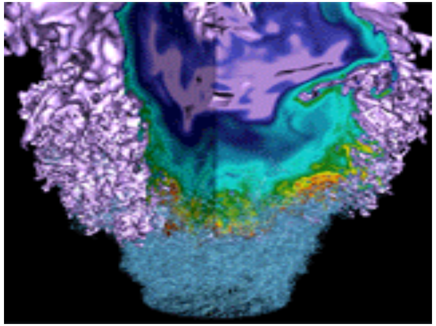


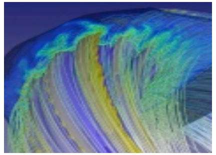


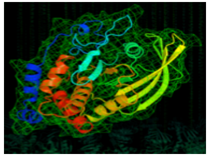


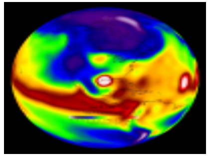


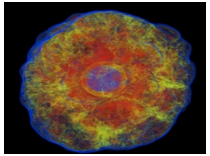
Productive deep learning at supercomputing scale

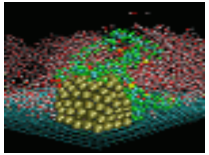








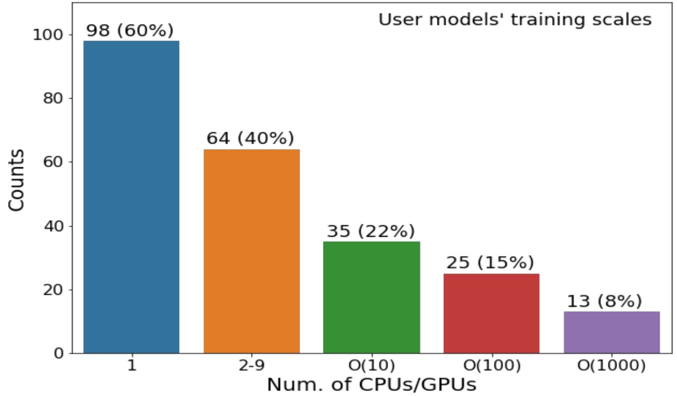


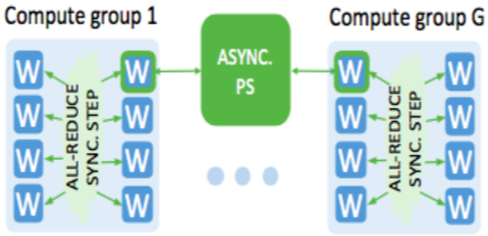


Multi-node training

- **Complex models benefit from training across multiple nodes**
- **Challenges to scaling include libraries to optimize communication and convergence at scale**
- **One way to distribute is via data parallel training for SGD**
  - Each node processes data independently then a global update
  - E.g. synchronous; asynchronous; hybrid approaches

NERSC ML Survey
Current and Future needs:





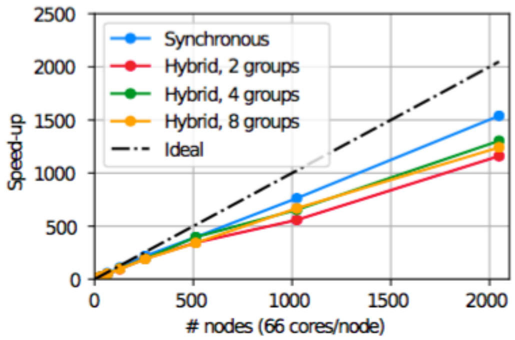
From Kurth et al. SC17 arXiv:1708.05256

Performant Multi-Node Libraries

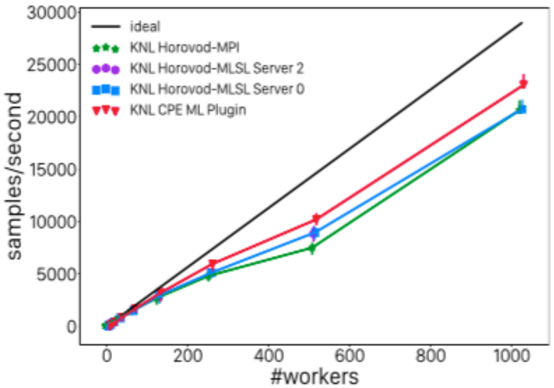
- **Initial scaling on NERSC involved a lot of work** (e.g. with Intel-Caffe and Intel-MLSL at SC17 **arXiv:1708.05256**)
- **Default TensorFlow uses gRPC for communication - non-ideal for Cori high-speed network:** (See e.g. Mathuriya et. al arXiv:1712.09388)
- **Now libraries available based on MPI with Horovod and Cray PE ML Plugin**

LHC-CNN (Intel-Caffe):

Kurth et al. SC17 **arXiv:1708.05256**

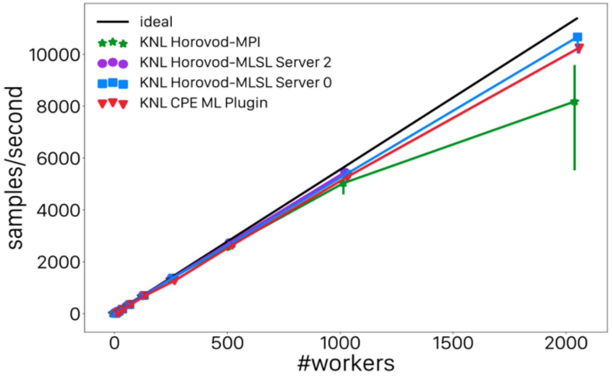


LHC-CNN (TF-Horvod)



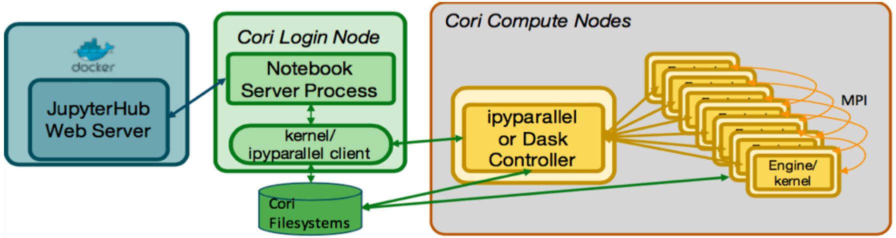
CosmoGAN:

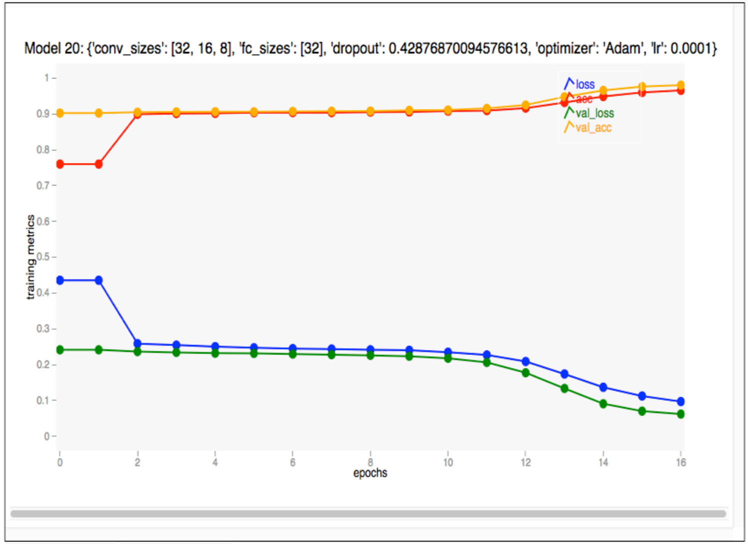
Kurth et al.
Concurrency Computat Pract Exper. 2018;e4989



Interactive HPC DL with Jupyter

Farrell, Vose, Evans, Henderson, Cholia, Pérez, Bhimji, Canon,Thomas , Prabhat,
[ISC 2018 Interactive HPC Workshop]





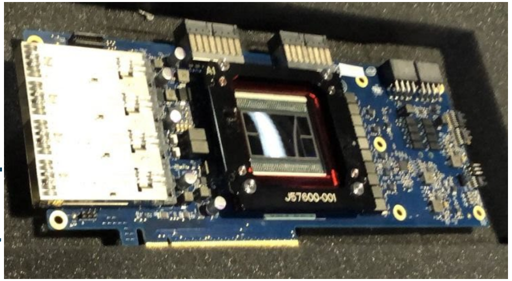
Stop selected | Restart selected

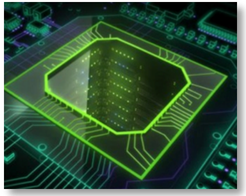
index	status	epoch	conv_sizes	fc_sizes	dropout	optimizer	lr	loss	val_loss	acc	val_acc
15	Ended Traini...	15	[8, 64, 32]	[256]	0.01919	Adam	0.001	0.0067703...	0.058146...	0.99775	0.98596875
19	Ended Traini...	15	[64, 16, 64]	[256]	0.61802	Adam	0.001	0.0497609...	0.039992...	0.982625	0.98496875
21	Ended Traini...	15	[4, 4, 8]	[64]	0.13547	Adam	0.0001	0.0735504...	0.051296...	0.9740625	0.982
20	Ended Traini...	15	[32, 16, 8]	[32]	0.42877	Adam	0.0001	0.0971096...	0.062174...	0.96575	0.98021875
18	Ended Traini...	15	[8, 8, 16]	[128]	0.29008	Adam	0.01	0.0700781...	0.073561...	0.975390...	0.969625
16	Ended Traini...	15	[8, 8, 8]	[256]	0.30157	Nadam	0.01	0.1863622...	0.162577...	0.927140...	0.93790625
25	Ended Traini...	15	[32, 8, 32]	[64]	0.57433	Nadam	0.01	0.2148241...	0.181064...	0.91615625	0.93253125
13	Ended Traini...	15	[8, 64, 128]	[256]	0.42386	Nadam	0.01	0.2105947...	0.191747...	0.9185	0.92828125
24	Ended Traini...	15	[16, 8, 128]	[128]	0.59087	Nadam	0.01	0.2333927...	0.201570...	0.9076875	0.92446875

- **Jupyter notebooks very popular development environment at NERSC**
- **Demonstrate interfacing jupyter to Cori interactive queue via ipyparallel or Dask cluster**
- **Distributed training communication is via MPI Horovod – no overhead in notebook**
- **Load-balanced Hyperparameter optimization tasks**
- **Live plots and task loss/accuracies in notebook; start/stop and monitor node resources with '[kale]'**

Rise of AI focused computing hardware

Intel adding mixed precision to Xeon Phi (Knights Mill), Also bought custom-chip startup Nervana and FPGA company Altera





NVIDIA builds DGX deep learning appliance with P100 and now V100 Voltas

Microsoft and Baidu deploy FPGA's

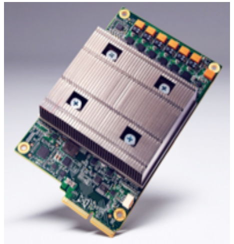
Many startups designing custom chips

cerebras

GRAPHCORE

Extra motivation to use (pure-)ML/DL based algorithms/frameworks for HEP/NP?

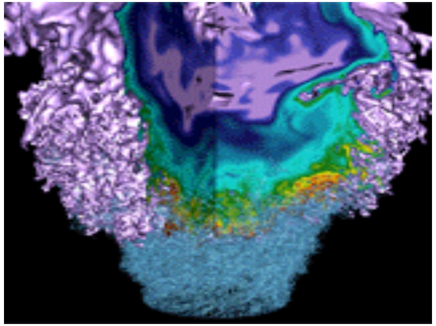
Google designs its own TPU

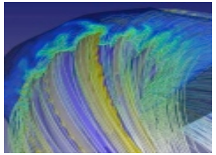


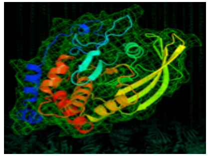
Conclusions

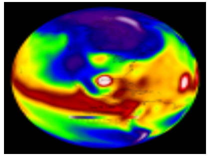
- **Deep learning – fast moving field with many techniques and tools applicable to HEP/NP problems**
  - Some HEP applications in ‘production’ many more coming.
  - Allows higher-dimensional ‘raw’ data, unsupervised learning, etc.
  - Potential gains from new approaches e.g. in inference and simulation
- **Build on industry tools and hardware to enable researchers to develop new ML/DL algorithms and easily run at scale**
  - Challenges are computational; methodological and practical
  - Welcome new collaborations; new ideas; joint events etc.

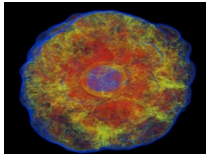
Backups

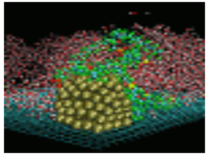






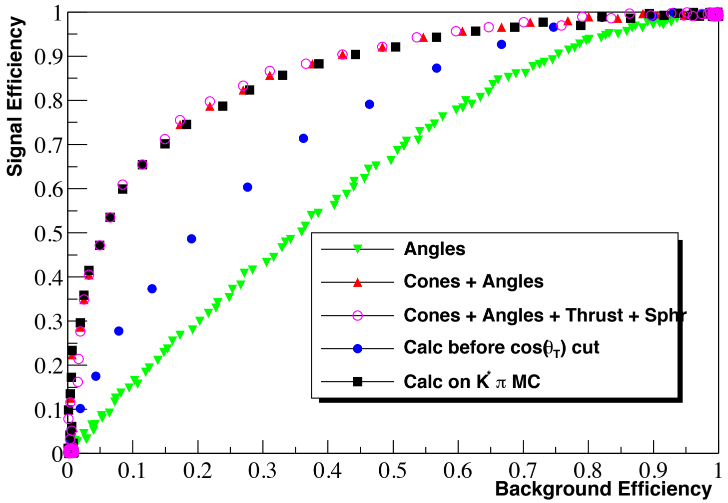






Machine Learning in High Energy Physics

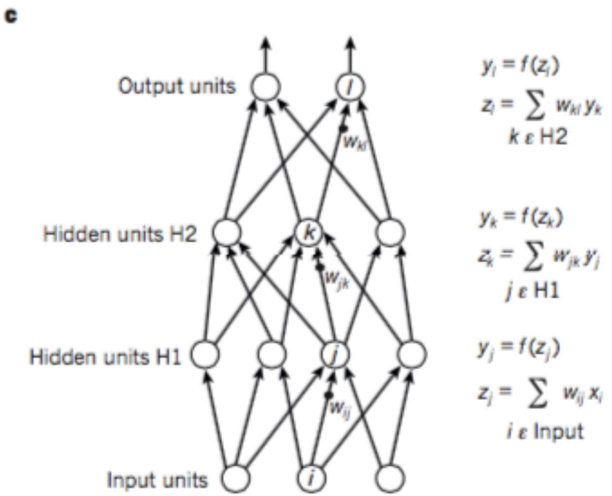
- **Machine learning including Neural Nets (NN) have a long history in experimental high-energy physics**
  - [E.g. Peterson (1988) “Track finding with Neural Networks”]()
- **Heavy use for last ~20 years – largely linear discriminants, decision trees and NNs with one hidden layer**
  - To form or combine high-level domain-specific variables
  - E.g. ‘Fisher linear’ discriminant in my 2002 PhD thesis on Babar experiment

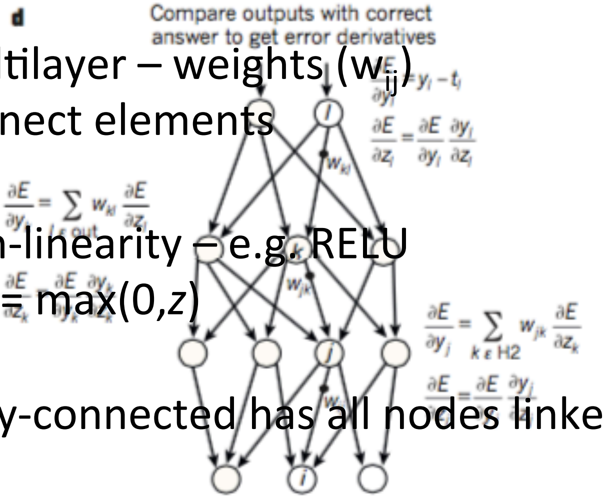


ROC curve – compares efficiency of searched for physics ‘signal’ (true positive rate) to boring known-physics background (false positive rate)

Deep learning: Rise of the Machine

- Neutral Networks with multiple hidden layers
  - Highly non-linear; millions of weights – huge capacity
  - Clever architectures (e.g. convolutional neural networks (CNNs)) to share weights – computationally tractable
  - *Not explained in detail here – but a few important elements:*





Multilayer – weights (w_{ij}) connect elements

Non-linearity – e.g. RELU $f(z) = \max(0,z)$

Fully-connected has all nodes linked

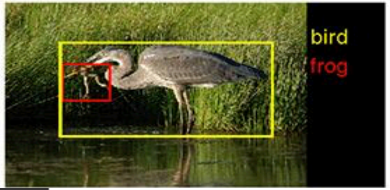
Cost/loss function applied at output

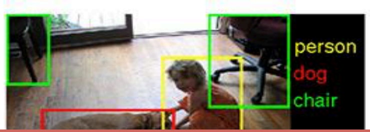
Backpropagation to update weights

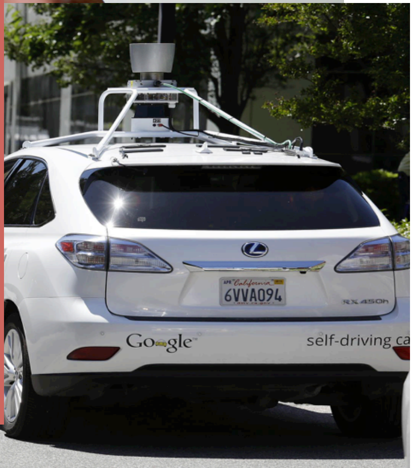
Can be trained by stochastic gradient descent (SGD)

Figures from LeCun,Bengio,Hinton **doi:10.1038/nature14539**

Industry investment and results







Resulting improvements in

- Methods – e.g. for image processing
- Software frameworks - e.g. Google's Tensorflow, Facebooks pyTorch etc..
- Hardware – e.g. GPU / Google's TPU..

Leverage these for HEP/Cosmology

- Improved new physics sensitivity
- High-dimensional 'raw' full-detector data
- Model-agnostic discovery
- Fast simulation at high-resolution

Tools: Deep Learning Stack on HPC

	Technologies
Deep Learning Frameworks	K, TensorFlow, PYTORCH, Caffe, Neon, CNTK, MXNet, …
Multi Node libraries	Cray ML PE Plugin, Horovod, MLSL
Single Node libraries	MPI, GRPC, MKL-DNN, CuDNN
Hardware	CPUs (KNL), GPUs, FPGAs, Accelerators

Deep networks with high-level variables

- First step was to replace existing HEP ML approaches with deep fully connected networks combining high-level physics variables
- Such approaches now used in production in HEP experiments
- Software tools and methods have moved on considerably since then allowing move to more raw information

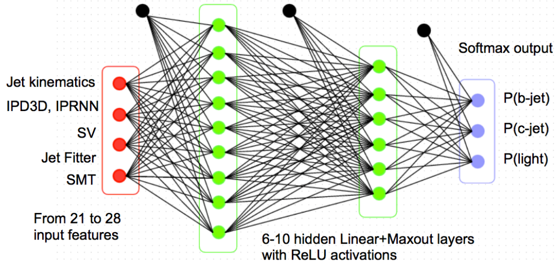
ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014 DOI: 10.1038/ncomms5308

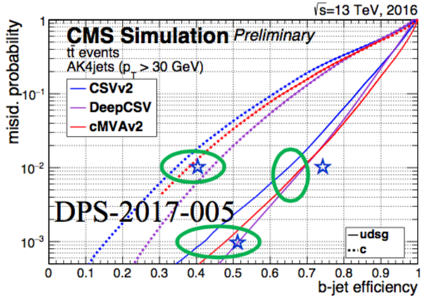
Searching for exotic particles in high-energy physics with deep learning

P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

ATLAS 'DL1': ATL-PHYS-PUB-2017-013



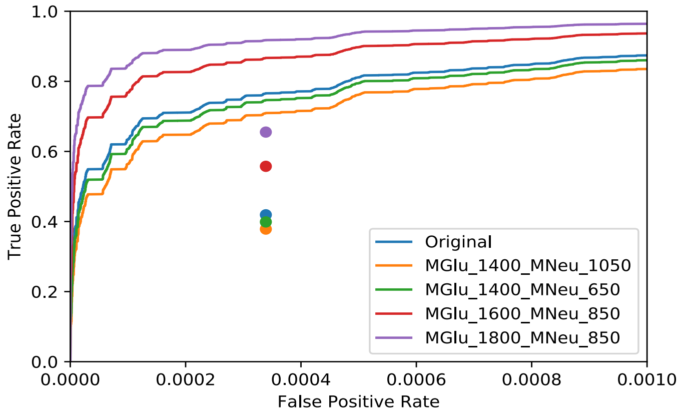
CMS DeepCSV:

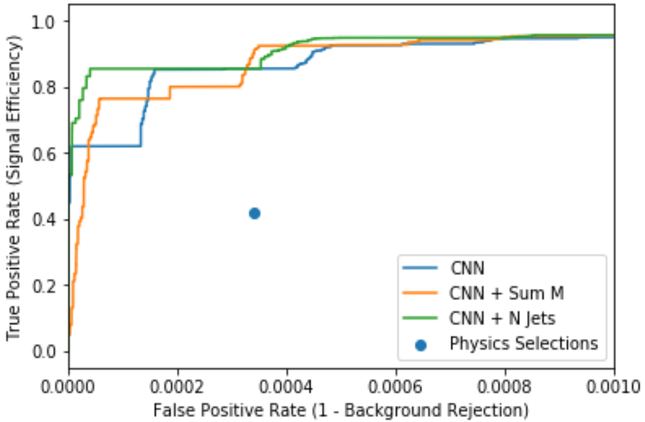


LHC – CNN Robustness and interpreting

- Can apply without retraining to other signal models (particle masses)

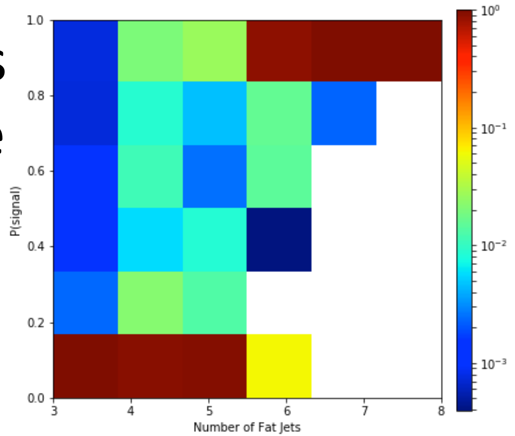
- Captures power of jet variables – e.g. add selections to CNN output in a 1 layer NN
  - Little/no increase in performance

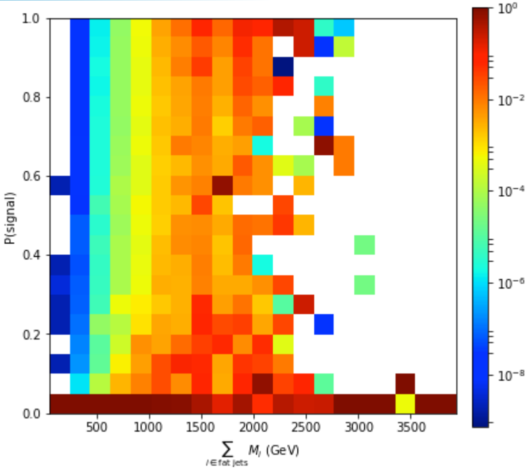


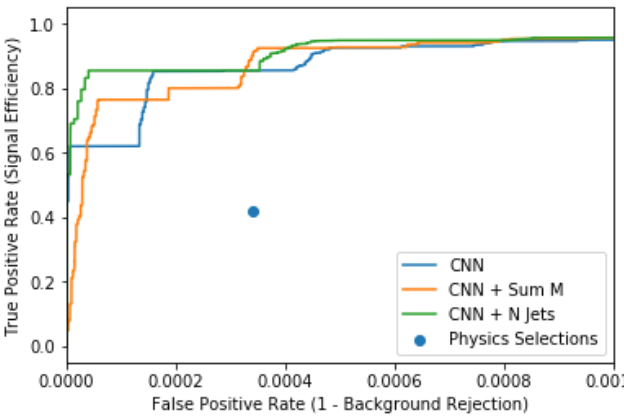


Comparing CNN to jet variables

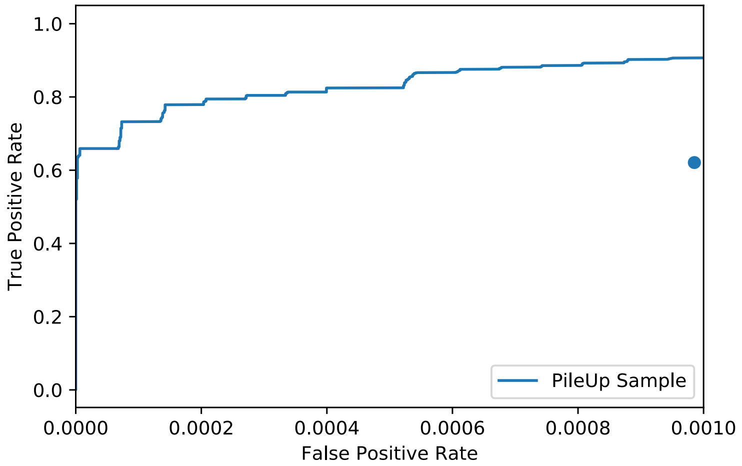
- Plot NN output (P(signal)) vs benchmark analysis variable
- Clear correlation
  - (Signal cuts:
    NJets >= 4 /5
    MJet >= 800/600 GeV)
- Add jet variable to CNN output in a 1 layer NN
  - Little/no increase in performance

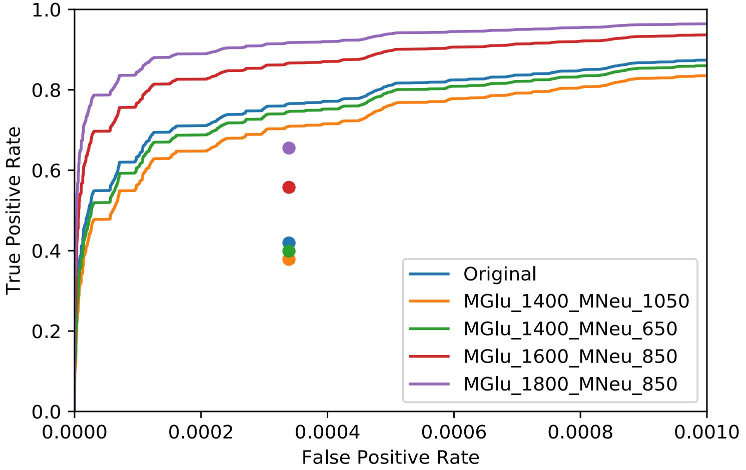






Different signals and pileup

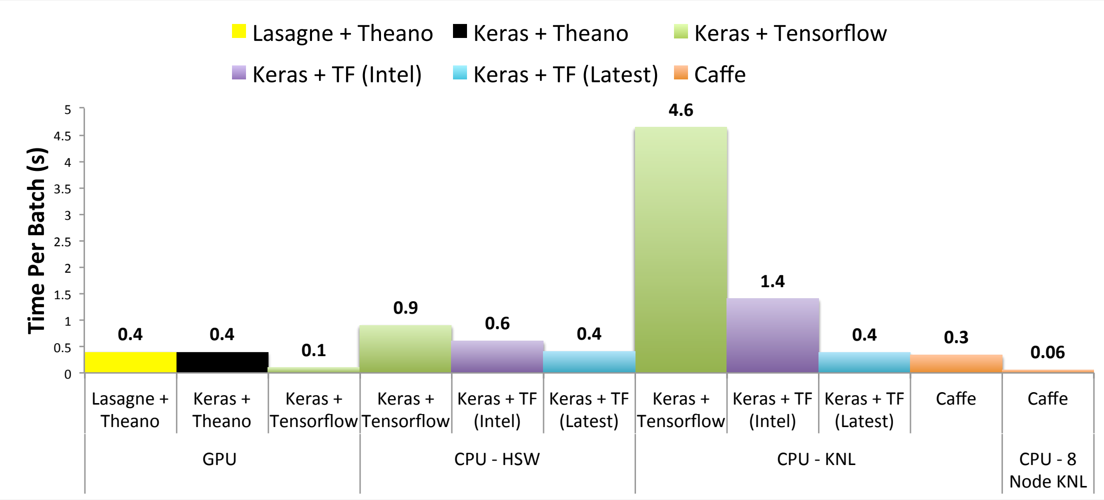




- Repeat analysis with Delphes pileup card (mu=20)
- Apply to nearby signal points without retraining

Timings and Tensorflow

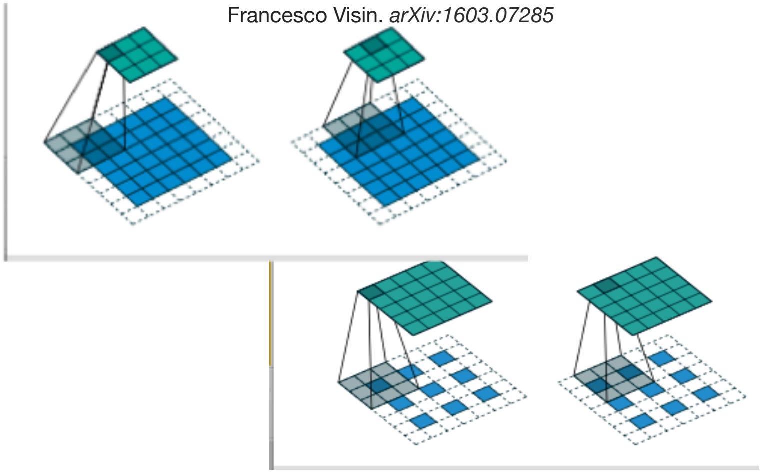
- CPU performance of default TF 1.2 was poor
- Intel optimisations with Intel Math Kernel Library (MKL) e.g. Conv layers multi-threaded,vectorize channels/ filters and cache blocking
  - Released in main TF-repo
- Further optimisations (now also upstreamed): e.g. MKL element-wise operations (avoid MKL->Eigen conversions)



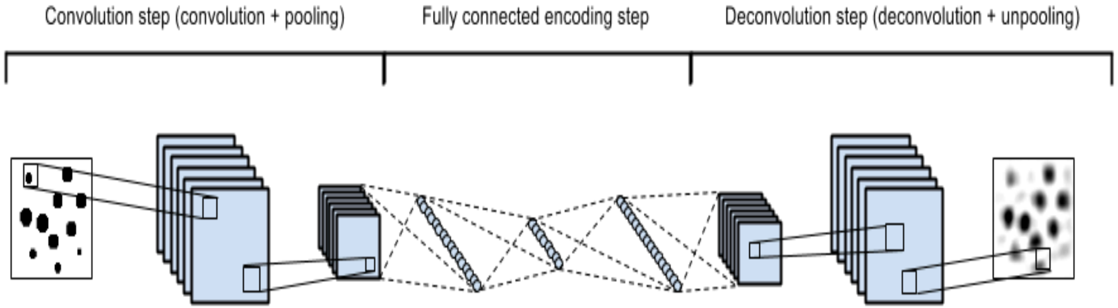
- (Intel)Caffe similar optimizations and Multi-node with MLSL library e.g. scale to 8 nodes time 6x faster for this 64x64 network

Convolutional Autoencoder (ConvAE)

Figure from Dumoulin, Vincent, and Francesco Visin. *arXiv:1603.07285*



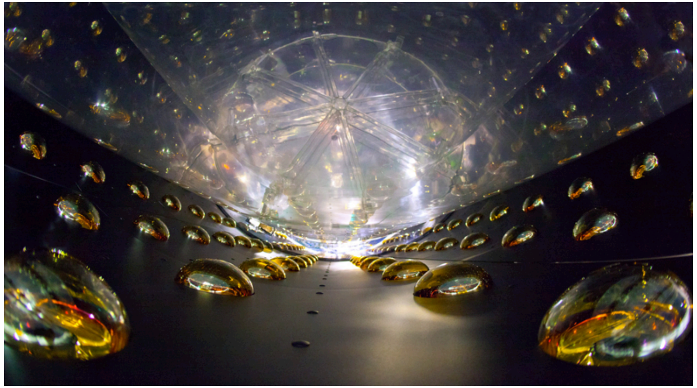
- Learn a lower dimensional, latent representation of data
  - Training target output is the input
- Encoder: (Convolutional) layers transform input into feature vector at bottleneck layer
- Decoder: transposed conv (deconvolutional) layers to reconstruct input

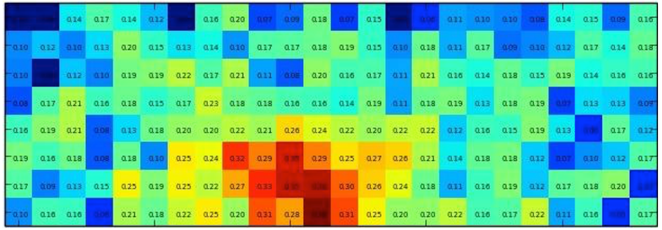


U.S. DEPARTMENT OF ENERGY | Office of Science

HEP use-case: Daya Bay Neutrino Experiment

- Detector has 192 Photomultiplier tubes in 8x24 cylinder
  - Use calibrated charge in 'image'
- Unsupervised training on experiment data. But use existing analysis for labeling after. Event types include:
  - **Inverse Beta Decay (IBD)** prompt and delayed signals
  - Cosmic **muon**
  - **Flasher** – instrument effect

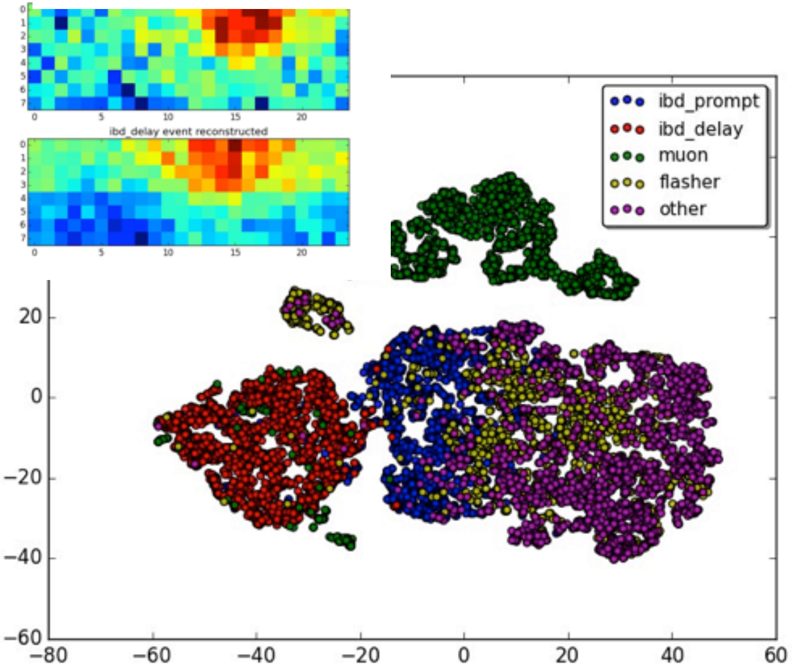




Clustering results

Evan Racah, Seyoon Ko, Peter Sadowski, WB, Craig Tull, Sang-Yun Oh, Pierre Baldi, Prabhat
https://arxiv.org/abs/1601.07621

- **Use 2 conv-layer ConvAE with a 10-d bottleneck feature vector**
  - Project onto 2 arbirary axes with t-SNE
- **Deep autoencoder reconstructs patterns of physics interest**
  - Forms clusters for different physics types (not trained with those labels)
- **Potential for data quality monitoring for new instrument effects or generic new physics filters where simulated training data doesn't exist**



Generative Adversarial Networks – Loss function

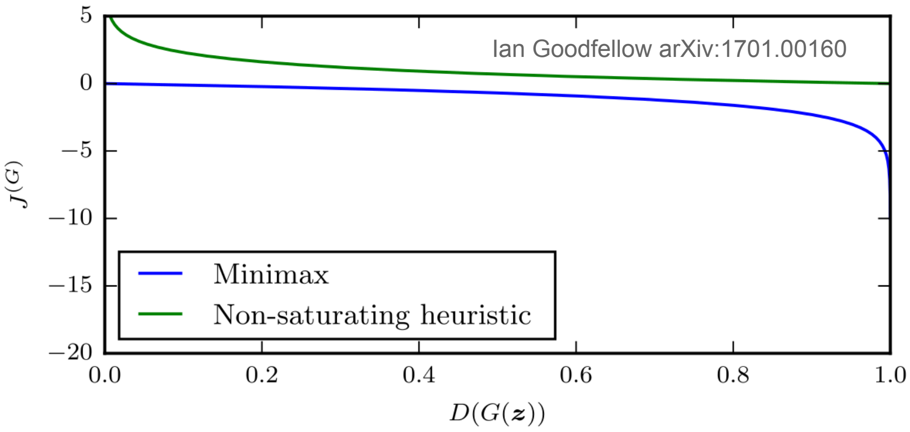
Minimax game formulation (saturating):

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim\mathbb{P}_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_{z\sim p_z} \log(1 - D((G(z)))$$

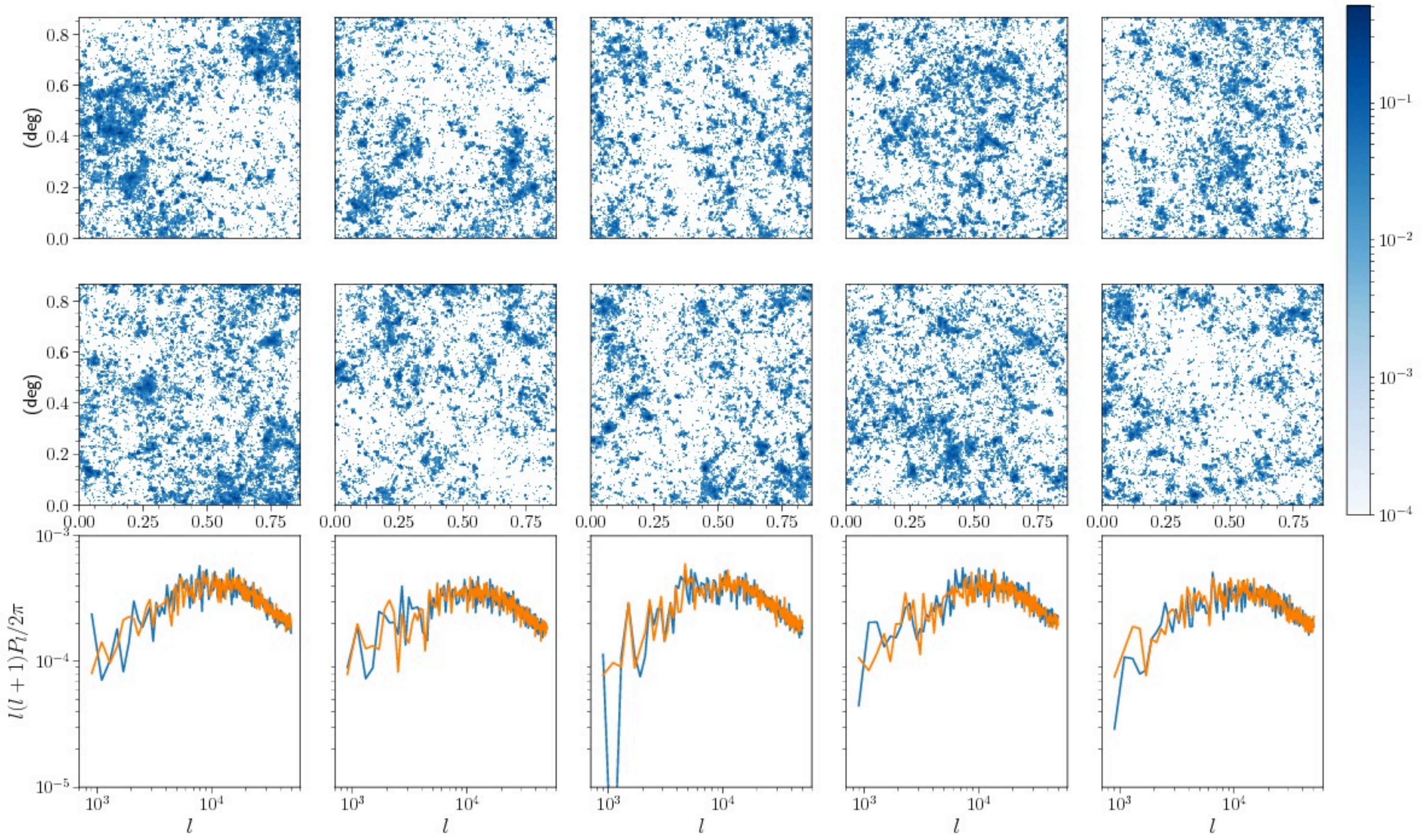
$$J^{(G)} = -J^{(D)}$$

Heuristic loss function (non-saturating):

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{z\sim p_z} \log D(G(z))$$

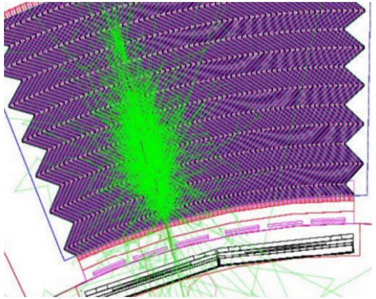


GAN not memorizing training images

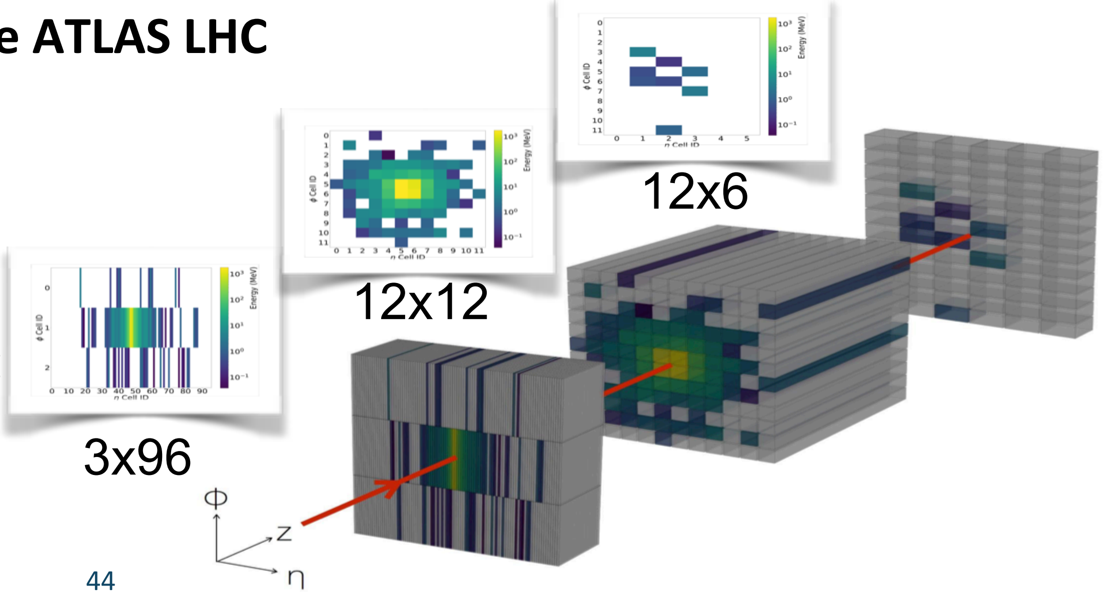


CaloGAN

Michela Paganini, Luke de Oliveira,
Benjamin Nachmann
https://arxiv.org/abs/1705.02355



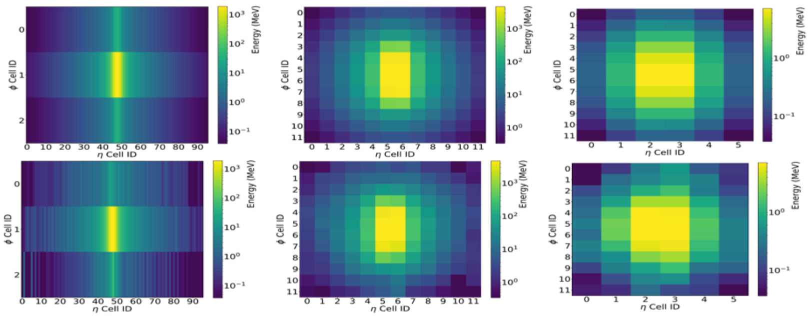
- Particle physics uses detailed micro-physics detector simulations (e.g. with Geant4)
  - >~50% LHC computing budget (10^{9} CPU hours)
  - Much of this compute time in calorimeter 'shower'
- **CaloGAN models a 3-layer calorimeter detector inspired by that of the ATLAS LHC experiment**
- **Custom NN design**
  - sparsity
  - high dynamic range
  - highly location-dependent features



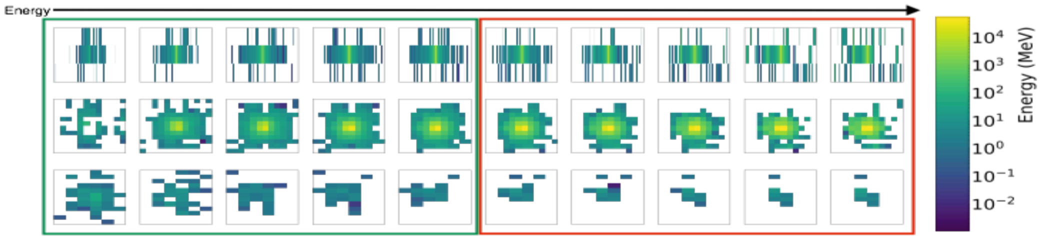
CaloGAN - results

Michela Paganini, Luke de Oliveira,
Benjamin Nachmann
https://arxiv.org/abs/1705.02355

- **Realistic average and individual images**
- **Conditional generation based on physical attributes**
    - Allowing parameter interpolation and extrapolation



Average energy deposition per calorimeter layer in the GEANT4 training dataset (top) and in the GAN generated dataset (bottom)



Requested (GeV)	1.0	23.1	45.2	67.3	89.4	111.6	133.7	155.8	177.9	200.0
Generated (GeV)	1.4	23.1	46.8	69.3	91.6	113.6	135.2	159.3	184.4	211.2

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB