机器学习在粒子物理中的应用

刘北江
中国科学院高能物理研究所

第一届高能物理计算暑期学校（IHEP School of Computing 2020）2020.8.24-25

Reference

Lots of tutorials/info on the web...

Online book by Nielsen ("Neural Networks and Deep Learning") at https://neuralnetworksanddeeplearning.com

Much more detailed book: “Deep Learning” by Goodfellow, Bengio, Courville; MIT press; see also http://www.deeplearningbook.org

Andrew Ng https://www.deeplearning.ai

Lectures by F. Marquardt https://machine-learning-for-physicists.org

HEPML https://github.com/iml-wg

…

, where some of the slides come from

Outline

- Introduction
  - What’s ML
  - Data flow in HEP
- ML applications in HEP-ex
  - A guided tour
- Outlook and challenges

What’s Machine Learning: Working Definitions

Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

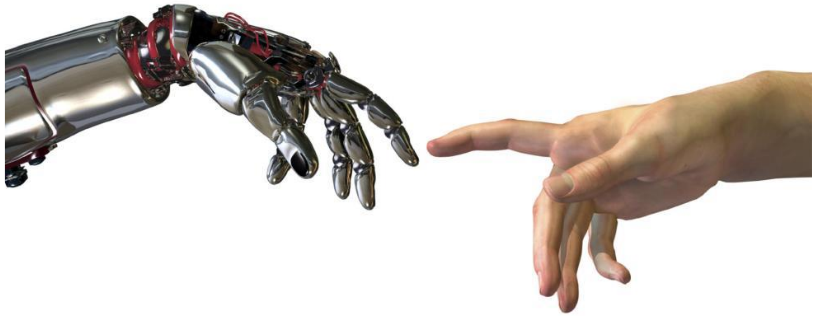
- Arthur Samuel, 1959

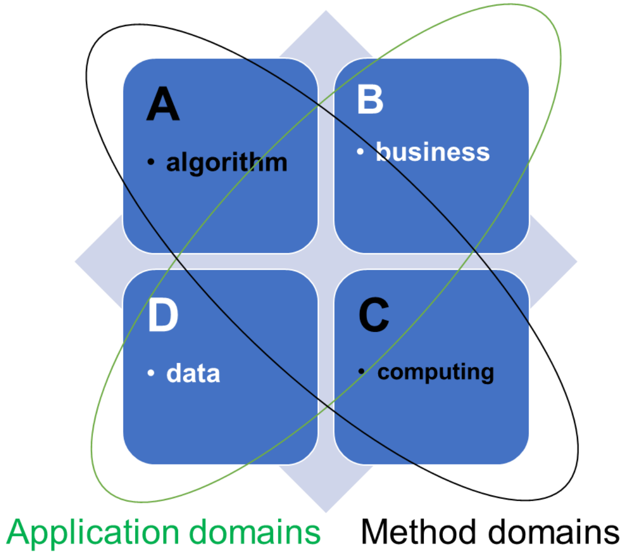
Machine Learning: A set of rules that allows systems to learn directly from examples, data and experience.

- Royal Society, 2017

“Learning” is the process of transforming information into expertise or knowledge; “Machine learning” is automated learning.

- Paraphrased from Jordan et al., 2015





Learning

- Supervised Learning
  - Data: (x, y) x is data, y is label
  - Goal: Learn a function to map x -> y
  - Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

- Unsupervised Learning
  - Data: x Just data, no labels!
  - Goal: Learn some underlying hidden structure of the data
  - Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

We have: $$y^{\text{out}} = F_w(y^{\text{in}})$$

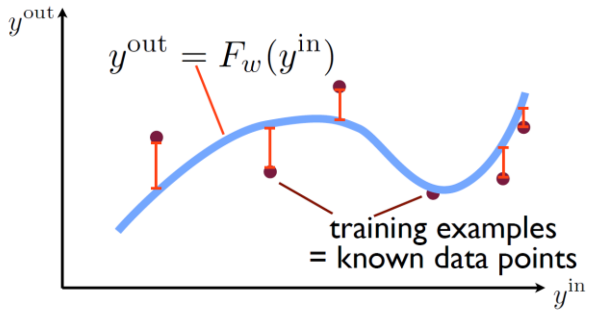
neural network
(w here also stands for the biases)

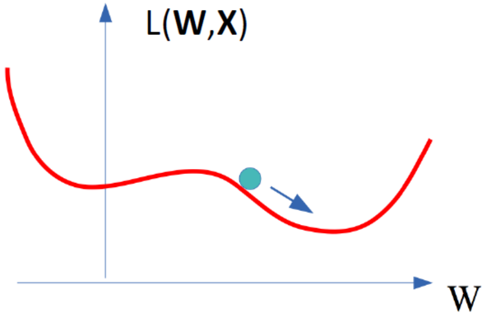
We would like: $$y^{\text{out}} \approx F(y^{\text{in}})$$

desired "target" function

$$C(w) \approx \frac{1}{2}\frac{1}{N}\sum_{s=1}^{N} \| F_w(y^{(s)}) - F(y^{(s)}) \|^2$$

s=index of sample



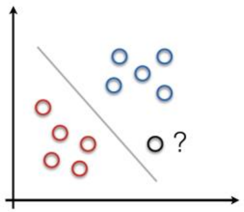


Machine Learning

- **Model**: reflects our knowledge of the system
- **Learning**: From "data" to "model", cast as an optimization problem
- **Inference:** From "model" to "answers"

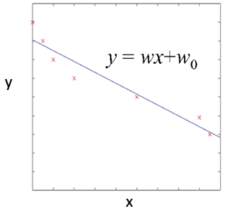
- *"Fitting data with complex functions"*
- Focus on predicting, rather than the parameters of model
- Model generalization

Predictive:

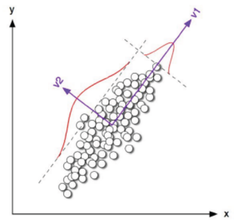


classification

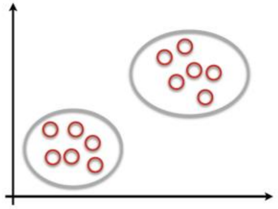
regression



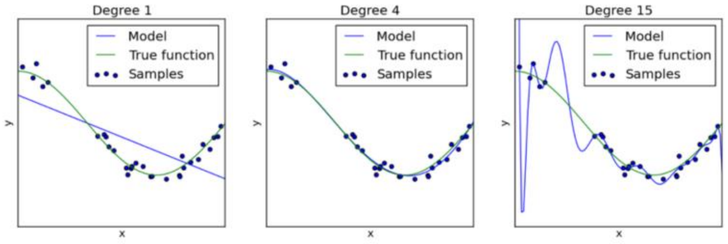
Descriptive:



dimensionality reduction

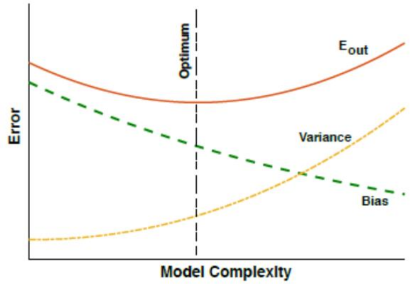


clustering

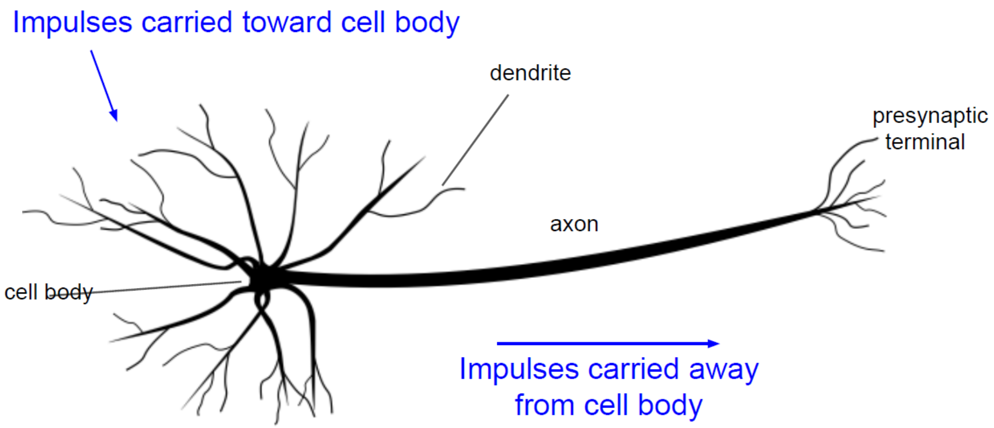


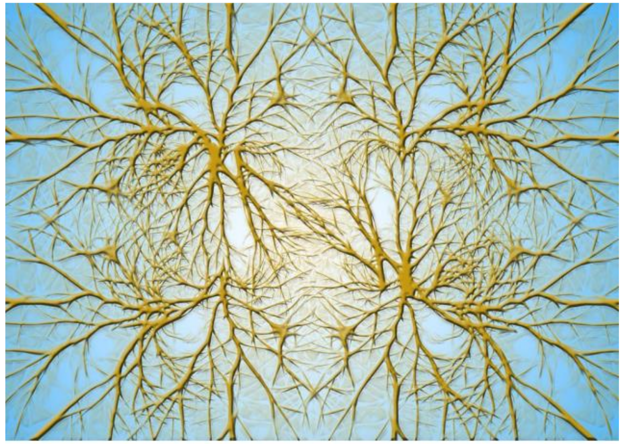
Underfitting

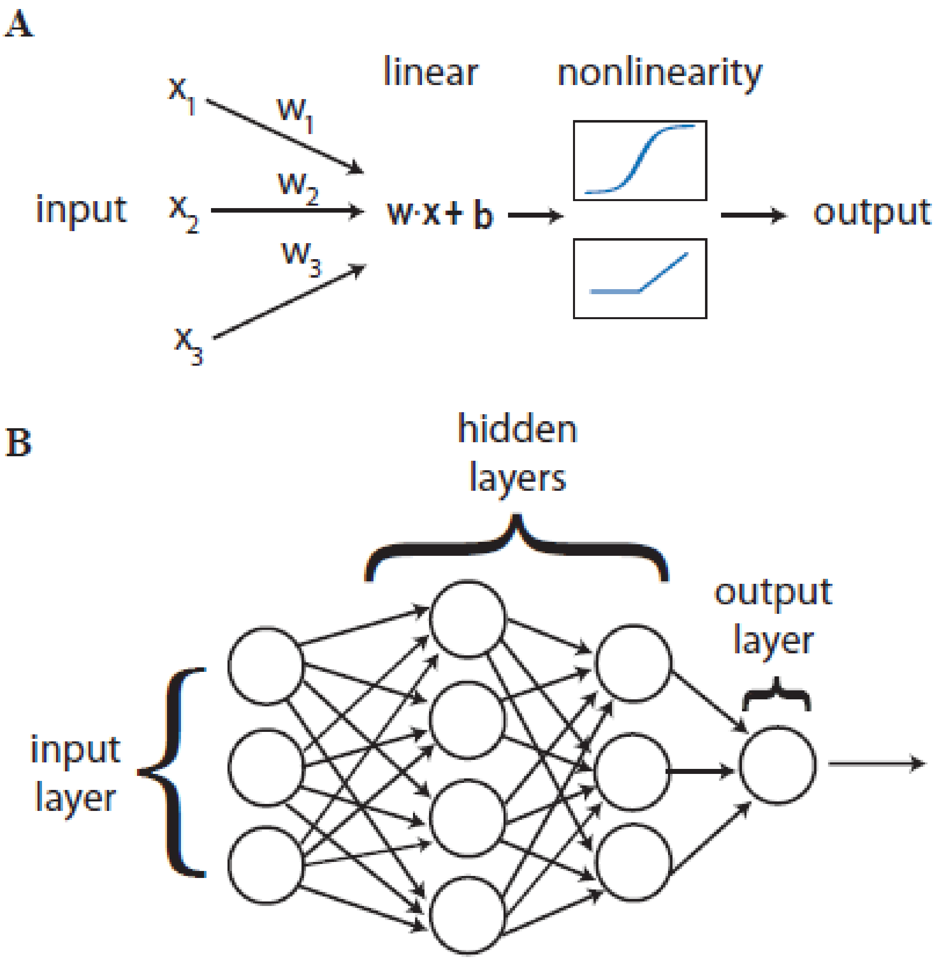
Overfitting



Artificial Neural Network



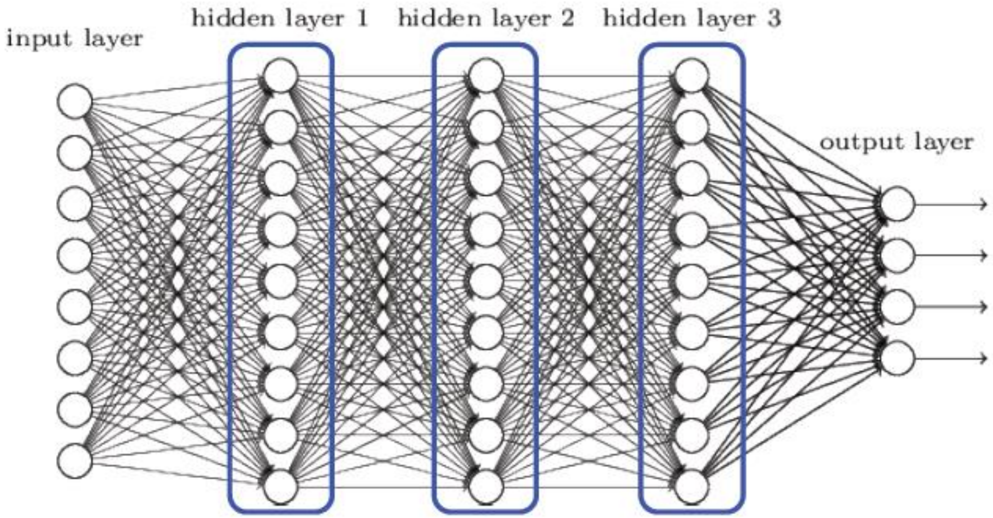




Universal Function Approximator

Cybenko 1989
Hornik, Stinchcombe, White 1989

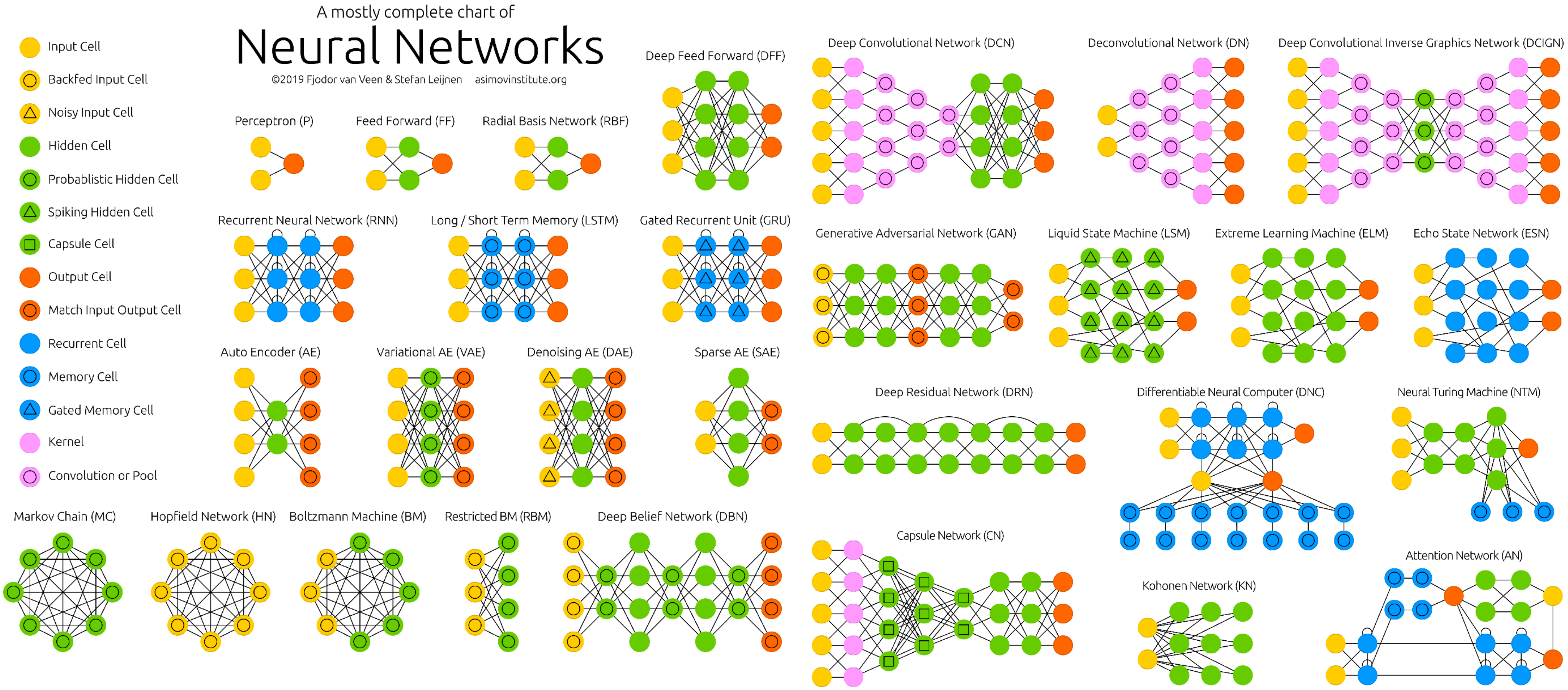
Deep Neural Networks



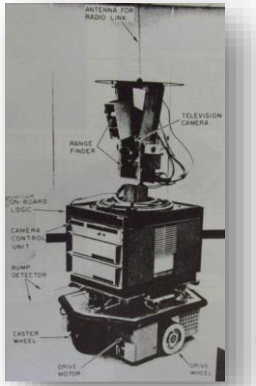
- As data complexity grows, need exponentially large number of neurons in a single-hidden-layer network to capture all the structure in the data
- Deep neural networks have many hidden layers

– Factorize the learning of structure in the data across many layers

- Difficult to train, only recently possible with large datasets, fast computing (GPU) and new training procedures / network structures (like dropout)


A mostly complete chart of
Neural Networks
©2019 Fjodor van Veen & Stefan Leijnen
asimovinstitute.org
Input Cell
Backfed Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Capsule Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Gated Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Differentiable Neural Computer (DNC)
Neural Turing Machine (NTM)
Capsule Network (CN)
Kohonen Network (KN)
Attention Network (AN)

Very brief history of artificial neural networks



"Perceptrons"
50s/60s

"Recurrent networks"
"Convolutional networks"
80s/90s

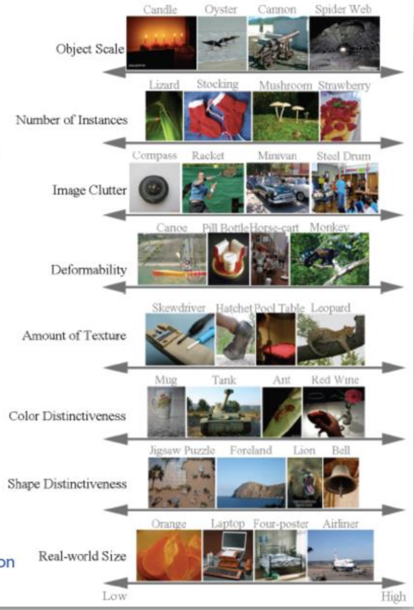
Deep nets for
image recognition
beat the competition
2012

ImageNet competition

1.2 million training pictures
(annotated by humans)
1000 object classes

2012: A deep neural network beats competition clearly (16% error rate; since then rapid decrease of error rate, down to about 7%)

Picture: "ImageNet Large Scale Visual Recognition Challenge", Russakovsky et al. 2014



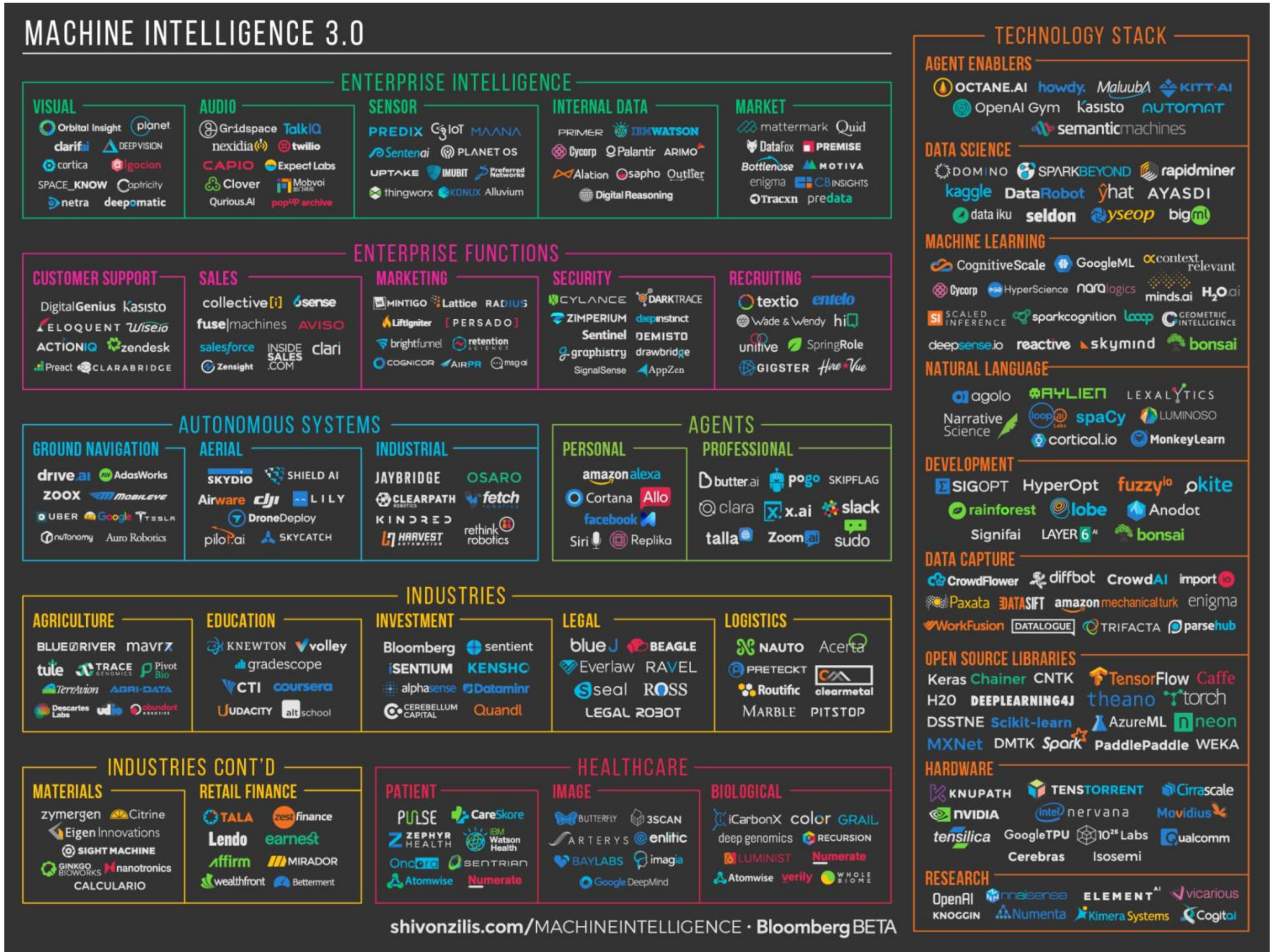
1956 Dartmouth
Workshop on
Artificial Intelligence

"Backpropagation"
80s (*1970)

early 2000s
"Deep networks"
become practical

2015
A deep net
reaches expert level in "Go"

Widely applied



Early applications in HEP

Computer Physics Communications 49 (1988) 429–448
North-Holland, Amsterdam

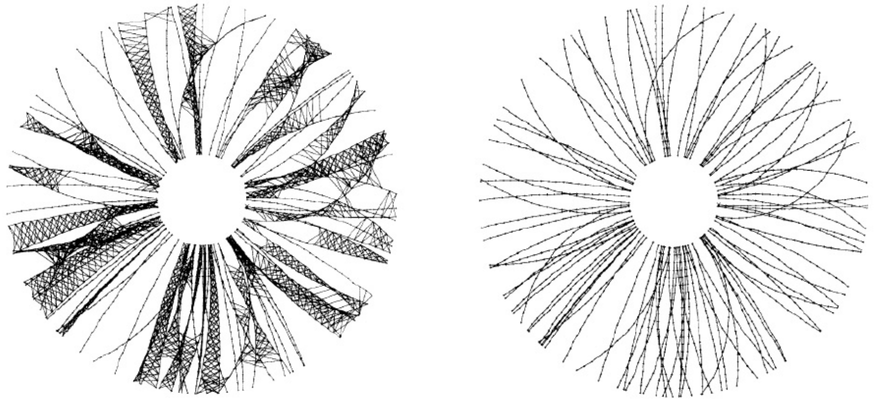
NEURAL NETWORKS AND CELLULAR AUTOMATA IN EXPERIMENTAL HIGH ENERGY PHYSICS

B. DENBY

Laboratoire de l'Accélérateur Linéaire, Orsay, France

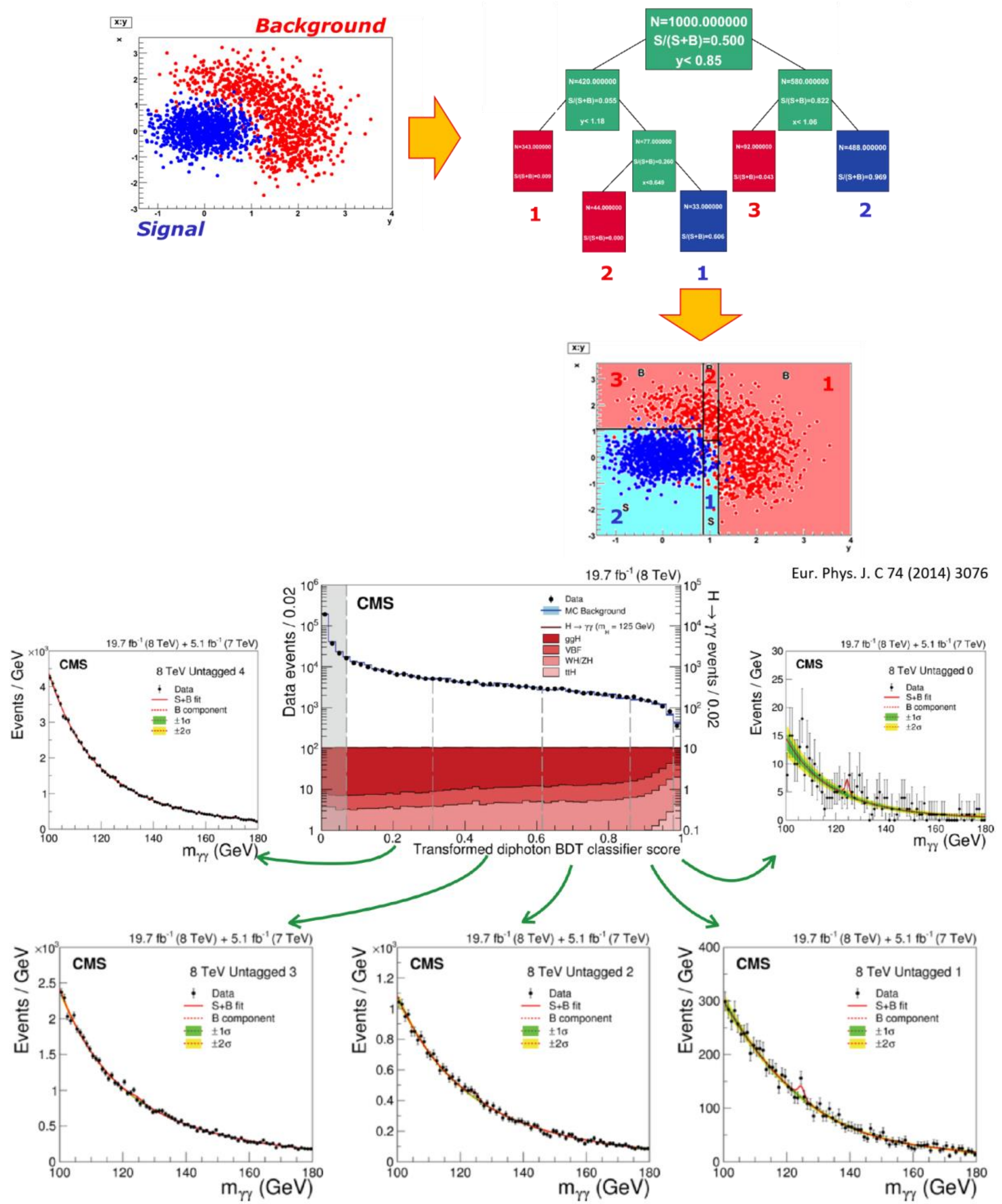
E.g. Peterson (1988) "Track finding with Neural Networks"

Full implementation in ALEPH Stimpfl & Garrido (1990)
Computer Physics Comm. 64 (1991) 46.



In early 2000's

- simple feed-forward neural networks were largely displaced by Boosted Decision Trees (BDTs)
  - MiniBooNe compared performance of different boosting algorithms and neural networks for particle ID (2005)
  - D0 claimed first evidence for single top quark production (2006) CDF (2008)

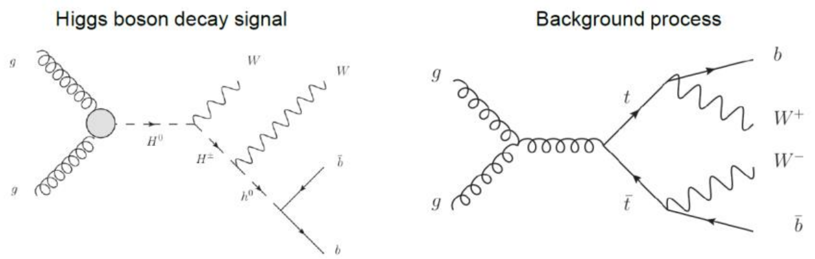


Since 2014, go “deep”

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

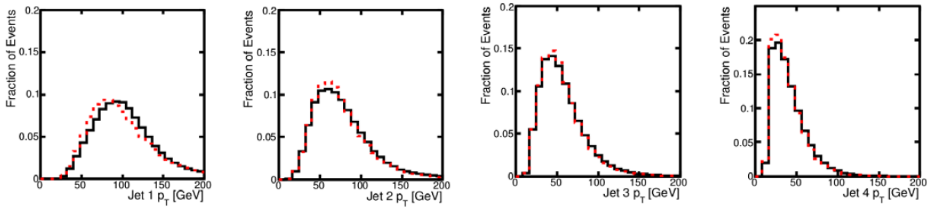
P. Baldi, P. Sadowski & D. Whiteson

Nature Communications **5**, Article number: 4308 (2014) | Cite this article

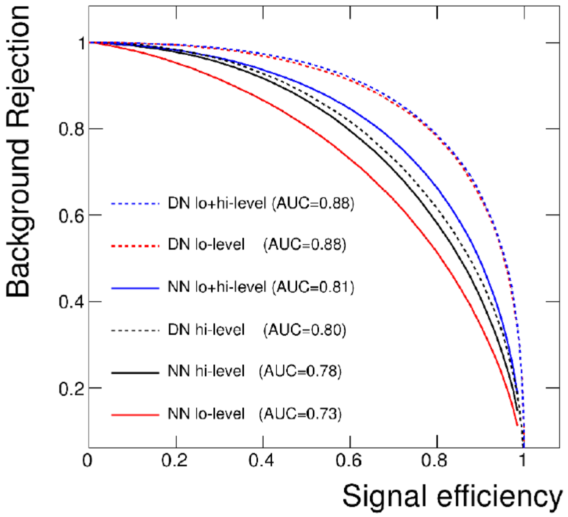


Supervised learning problem:

- Two classes
- 11 million training examples (roughly balanced)
- 28 features
  - 21 low-level features (momenta of particles)
  - 7 high-level features derived by physicists



Signal (black) vs background (red)



	AUC		
Technique	Low-level	High-level	Complete
BDT	0.73	0.78	0.81
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 (< 0.001)	0.885 (0.002)

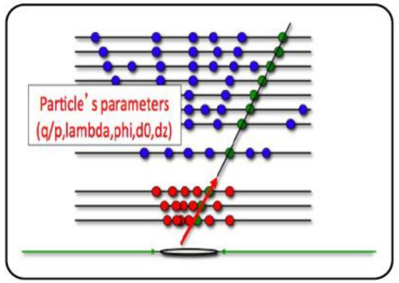
Deep network improves AUC by 8%

not only out-performed BDT, but also did not require engineered features to achieve the performance

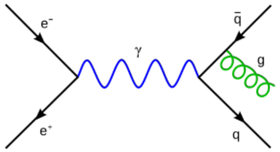
Work flow

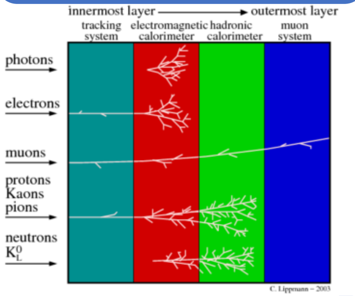
nature

detector



Classify
signal/background





reconstruction

event selection

Statistical inference

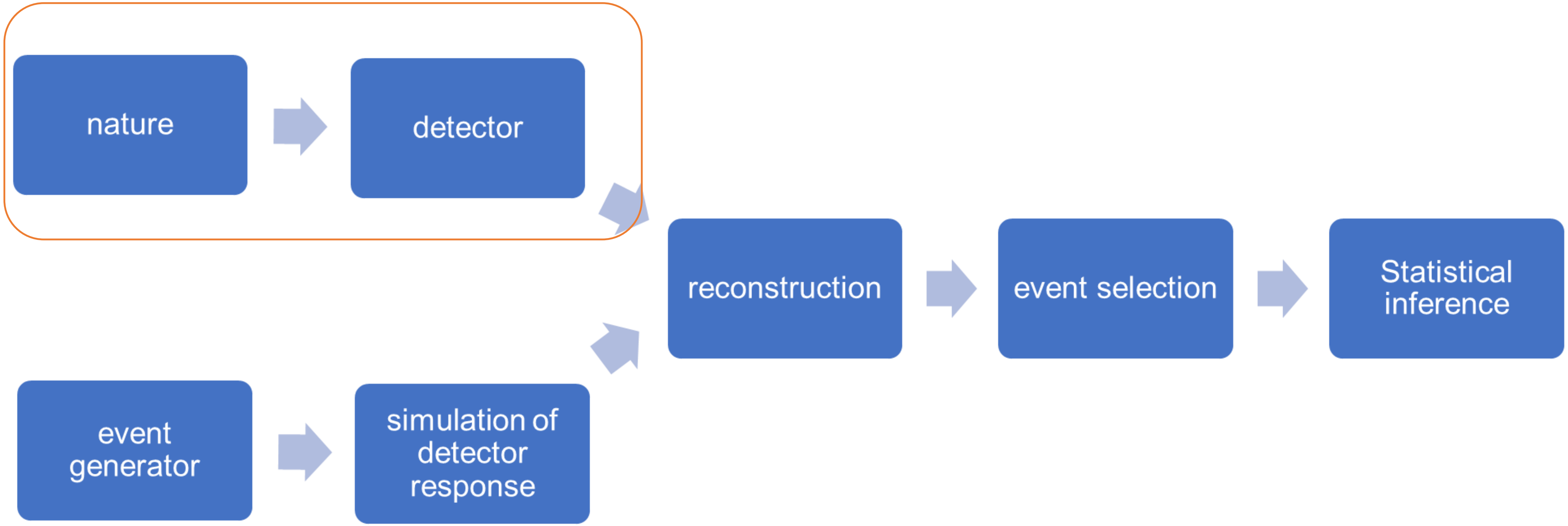
event generator

simulation of detector response

hits→ final state particle :
Cluster hits
Classify clusters as particles
Infer/regress properties

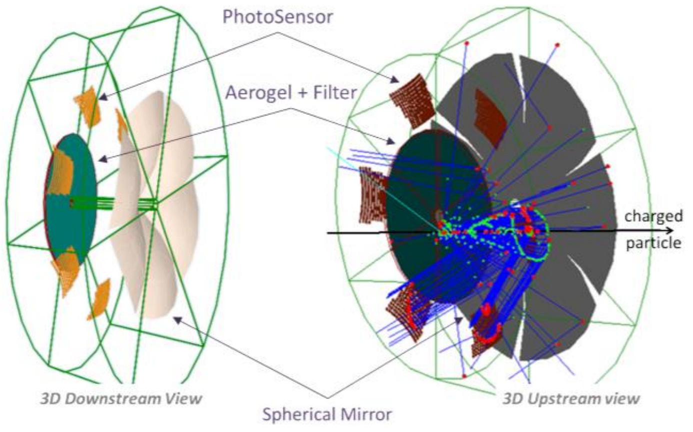
Hypothesis testing
Parameter estimation

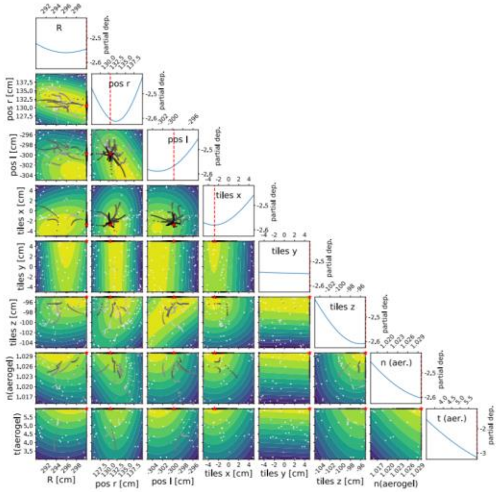
Accelerator/detector design

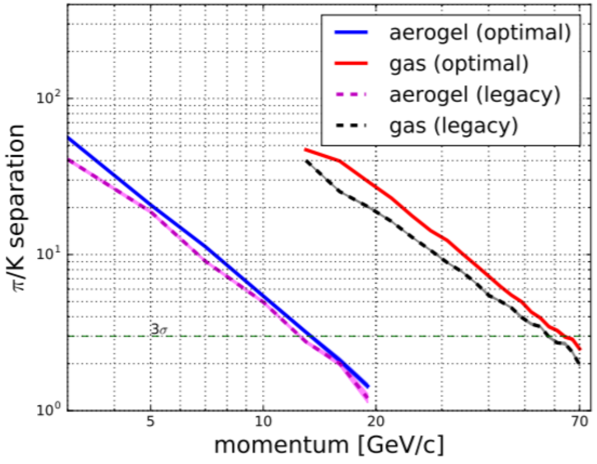


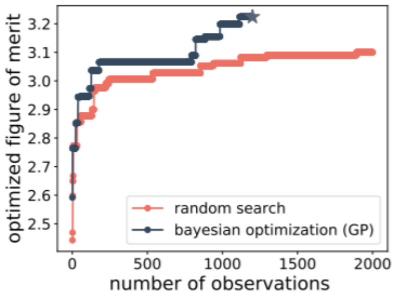
AI-optimized detector design for the future Electron-Ion Collider: the dual-radiator RICH case

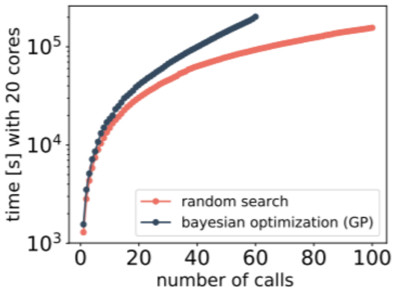
JINST 15 P05009(2020)









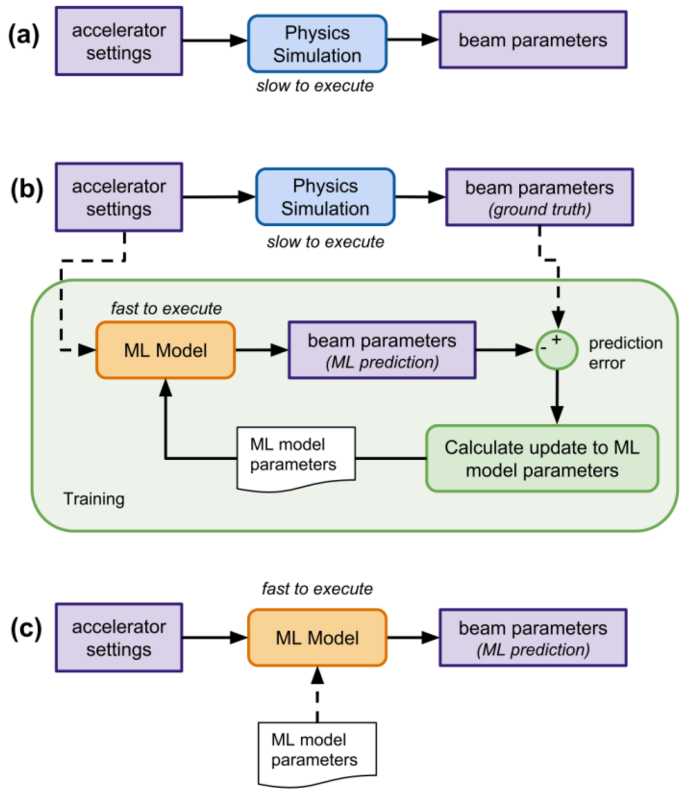


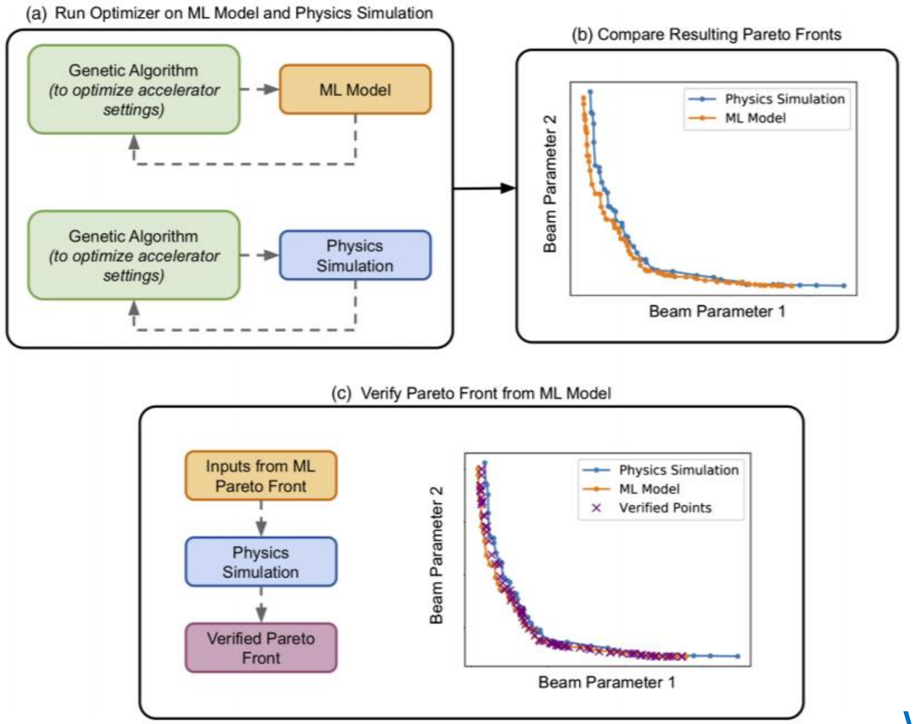
Bayesian optimization x Gradient Boosted Regression Trees (GBRT)

Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems

Auralee Edelen, Nicole Neveu, Matthias Frey, Yannick Huber, Christopher Mayes, and Andreas Adelmann

Phys. Rev. Accel. Beams 23, 044601 (2020)

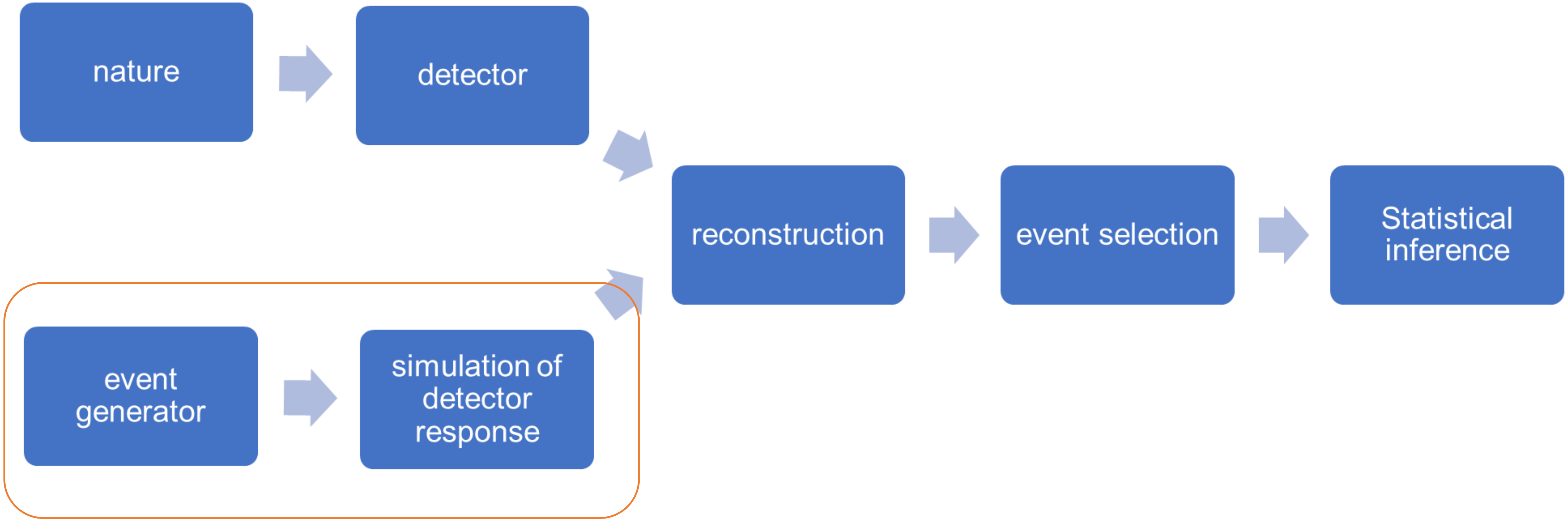




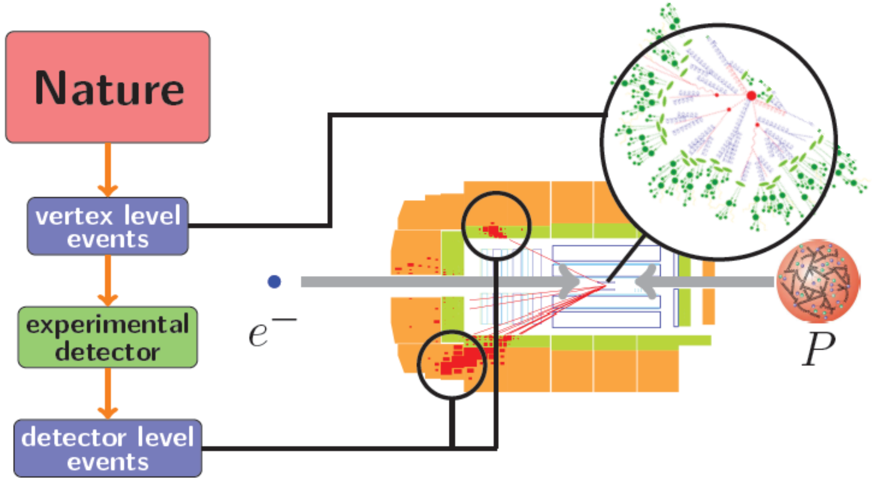
The models are $\mathcal{O}(10^6)$–$\mathcal{O}(10^7)$ times more computationally efficient to execute.

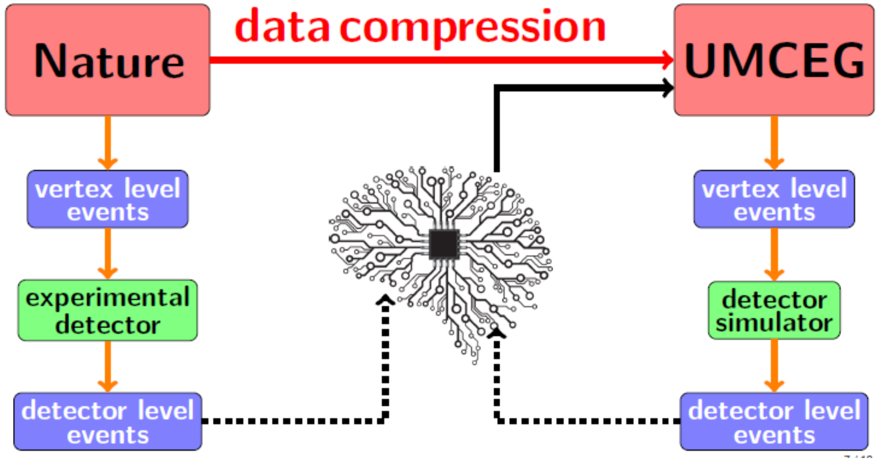
When considering the computation time required to generate the training data and to train the NN, the overall improvement is still substantial (*O(100)*)

Simulation

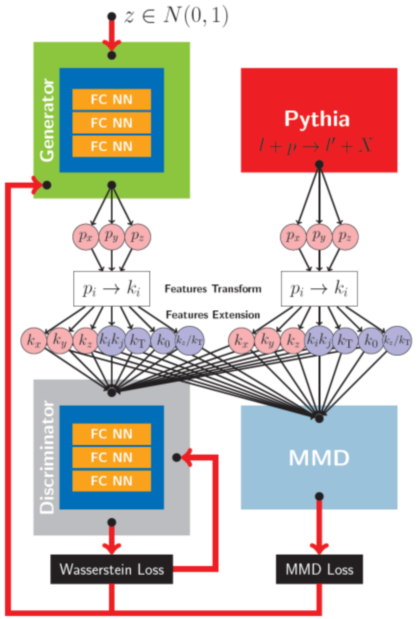

nature
detector
event generator
simulation of detector response
reconstruction
event selection
Statistical inference

Universal Monte Carlo Event Generator,
arXiv:2008.03151



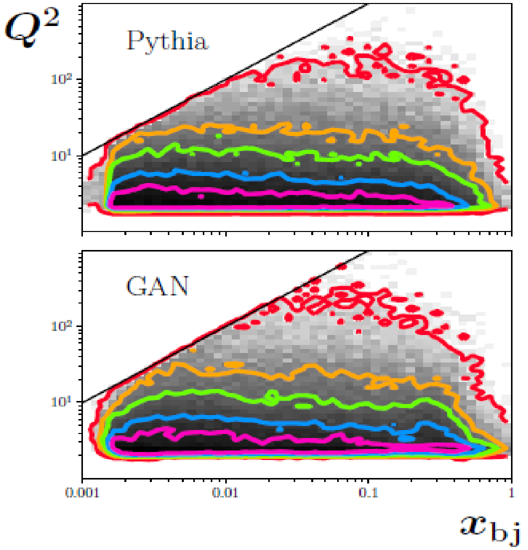


FAT-GAN



- Event image $= l'_{x,y,z}$
- Feature extension: $l'_i \cdot l'_j, \quad l'_0, \quad l'_z/l'_T$

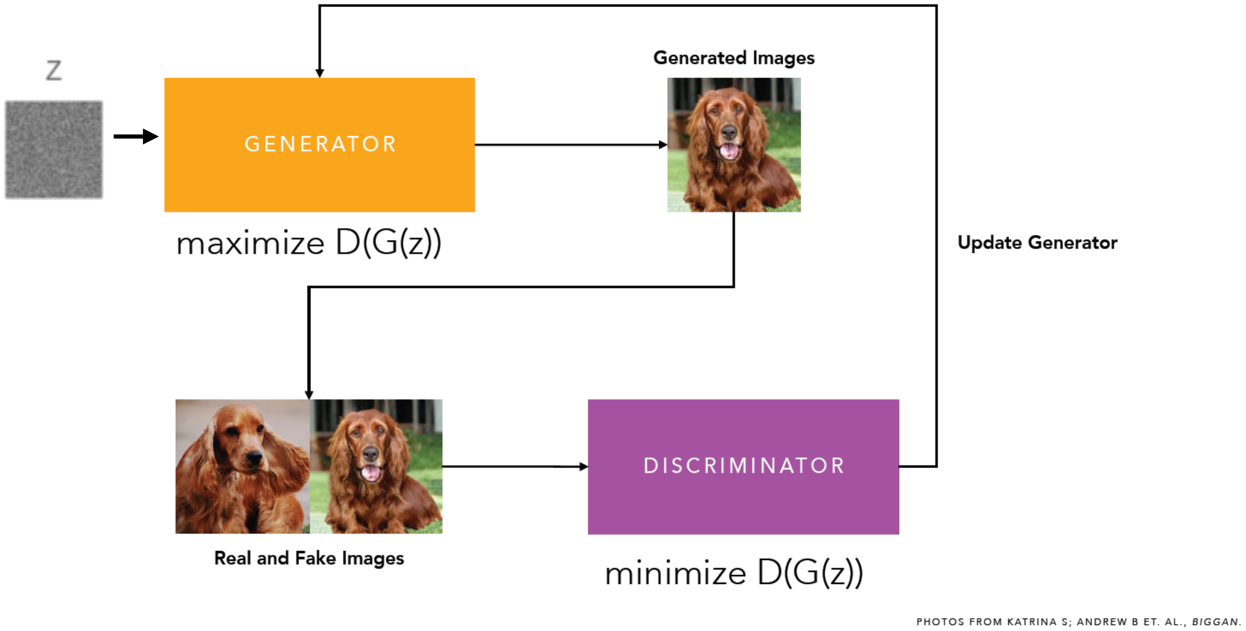
Vectors generator



theory-free

Generative Adversarial Networks GAN

Transform noise into
a realistic sample



Distinguish real samples
from fake samples

- How can we jointly optimize G and D?
- Construct a two-person zero-sum minimax game with a value V

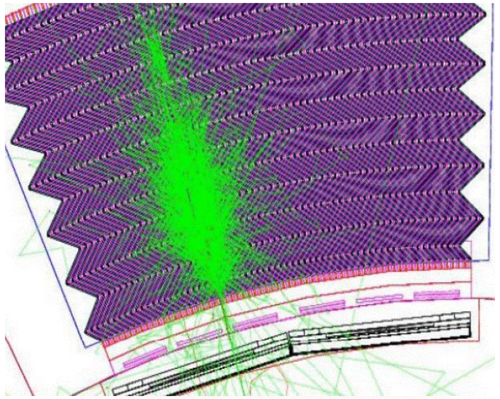
$$V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x;\theta_D)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z;\theta_G);\theta_D))]$$

- We have an inner maximization by D and an outer minimization by G

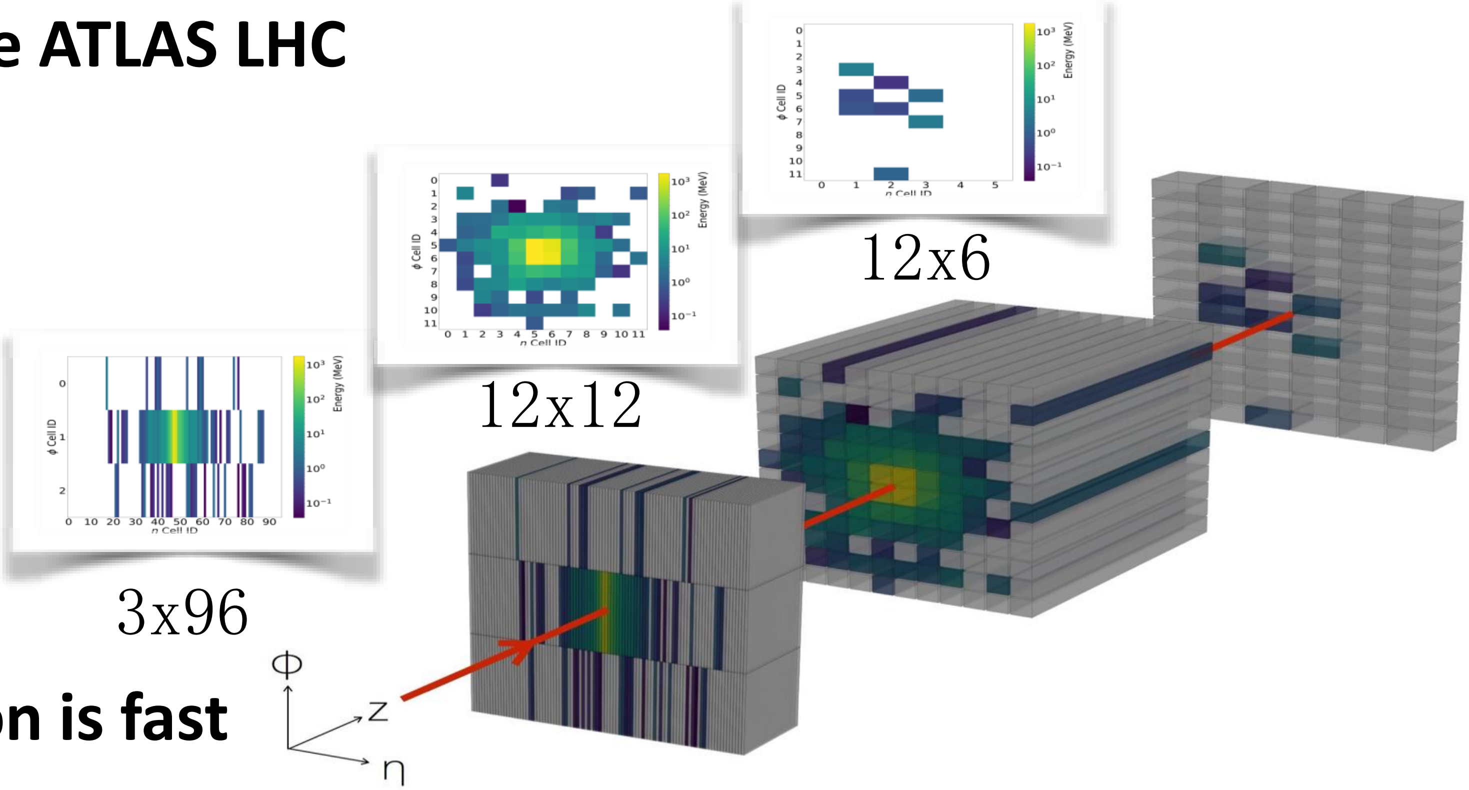
$$\min_G \max_D V(D,G)$$

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



Training NN’s is slow, but evaluation is fast

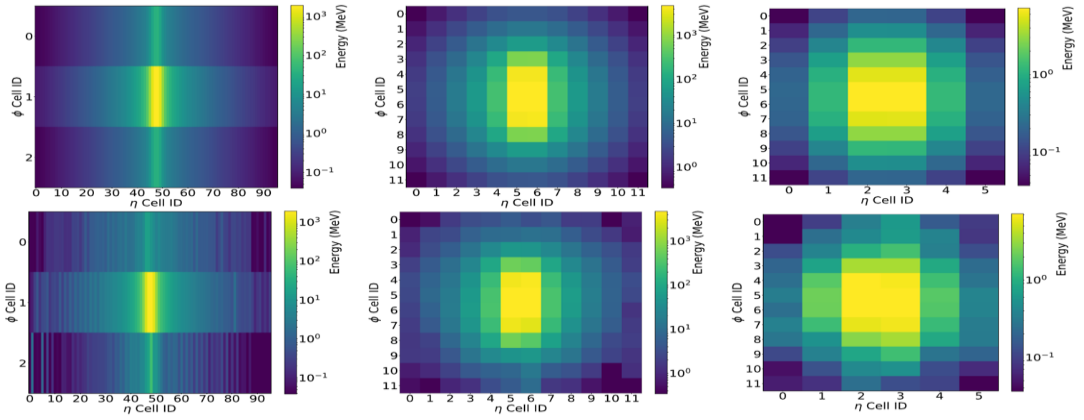
CaloGAN - results

Michela Paganini, Luke de Oliveira,
Benjamin Nachmann

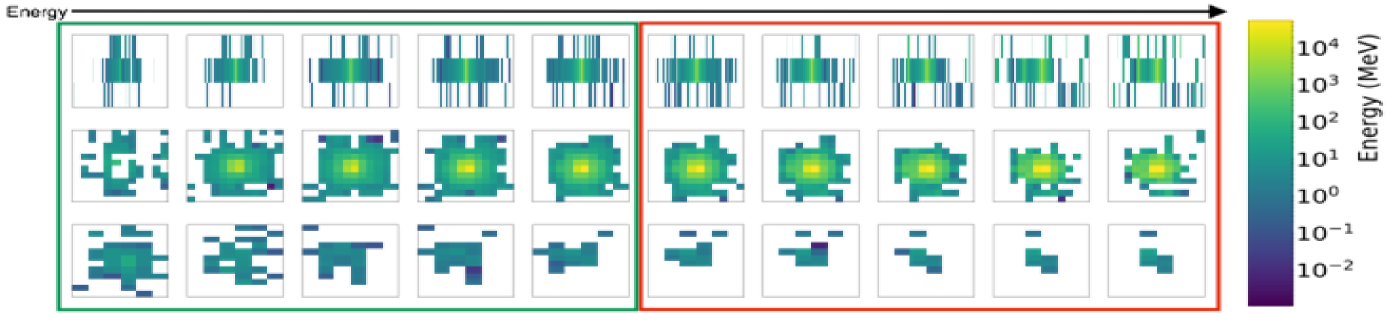
https://arxiv.org/abs/1705.02355

- **Realistic average and individual images**

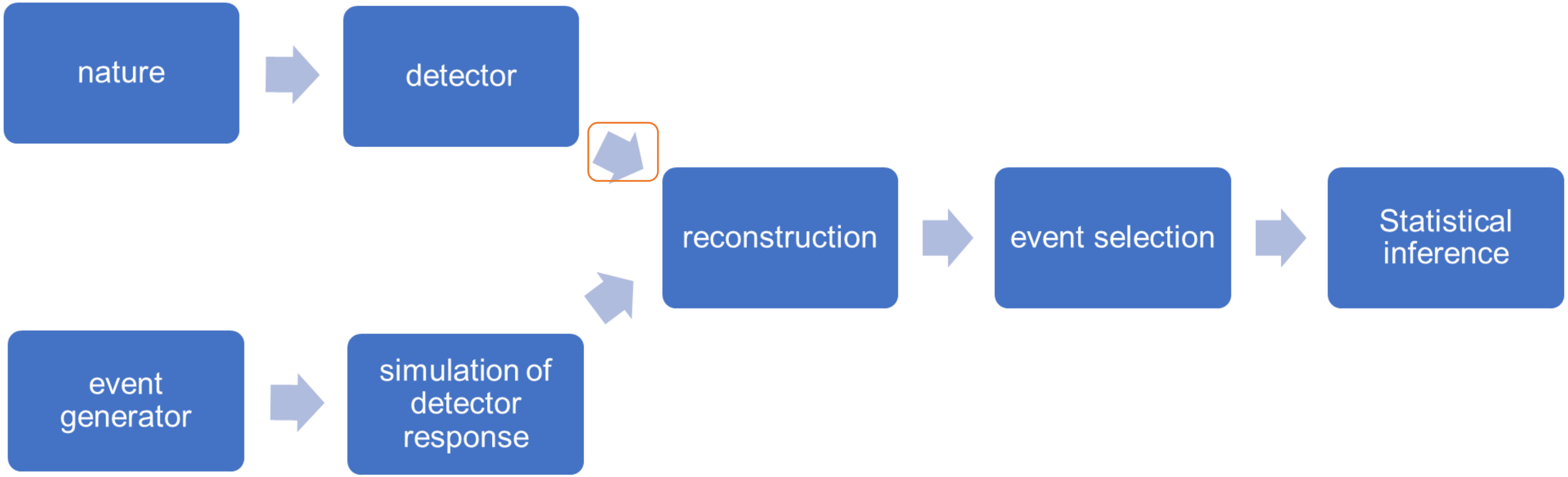
- **Conditional generation based on physical attributes**
  - Allowing parameter interpolation and extrapolation



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


nature
detector
event generator
simulation of detector response
reconstruction
event selection
Statistical inference

Real Time Analysis and Triggering

JINST 13 (2018) 07, P07027

A typical trigger system

Triggering typically performed in multiple stages @ ATLAS and CMS

Level-1 trigger (L1)

custom hardware

trigger decision to be made in O(μs) (“latency”)

rate in/out: 40 MHz / 100 KHz

Latencies necessitate all-FPGA design!

High-level trigger (HLT)

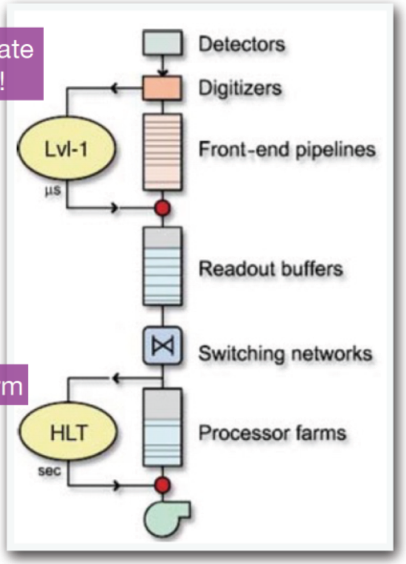
computing farm for detailed analysis of the full event

latency O(100 ms)

rate in/out: 100 KHz / 500 Hz

CPUs farm

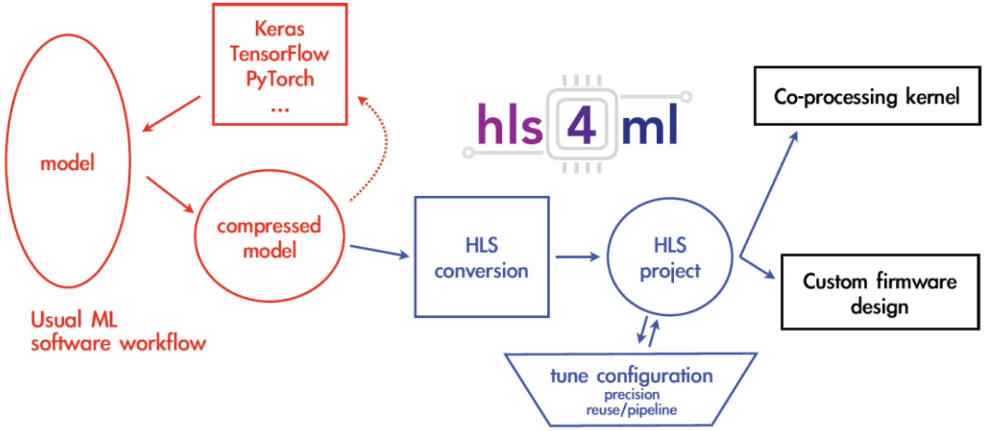
For HL-LHC upgrade: latency and output rates will increase by 5 (ex: for CMS 3.2 → 13 μs @ L1)



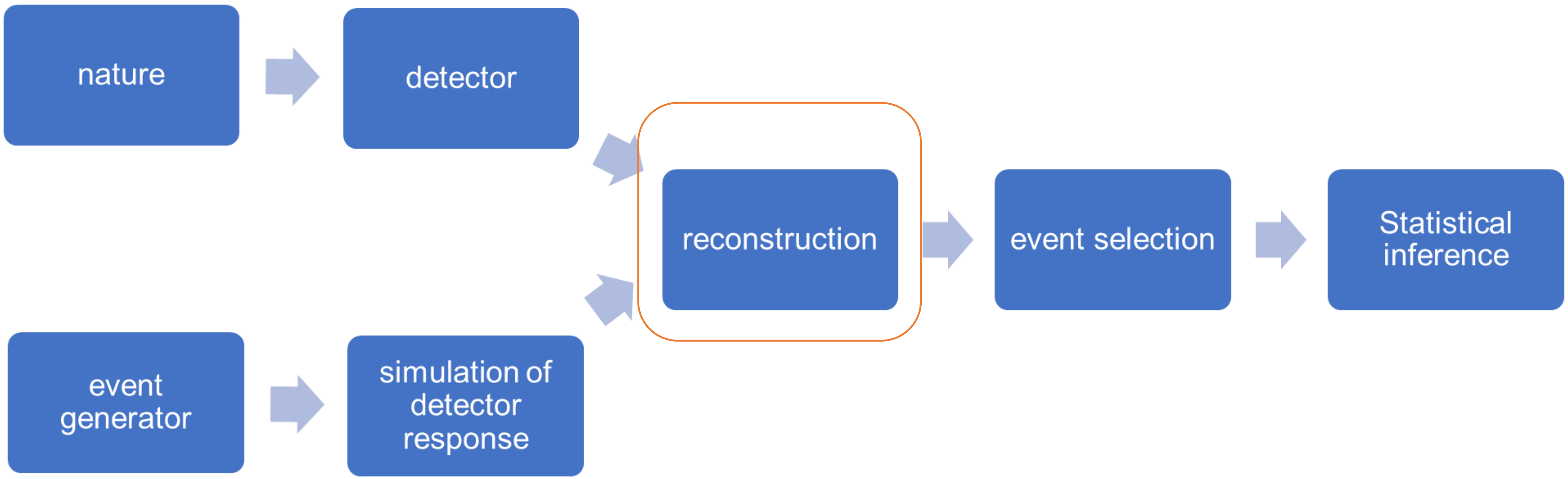
inference of deep neural networks in FPGAs for low-latency application

Compression, Quantization, and Parallelization made easy in

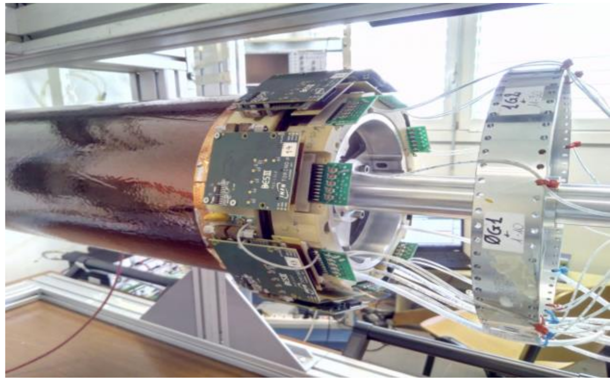
high level synthesis for machine learning

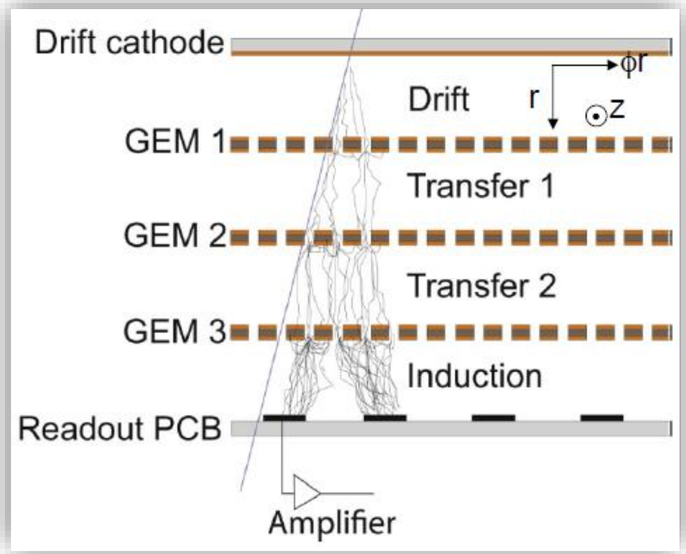


https://hls-fpga-machine-learning.github.io/hls4ml/

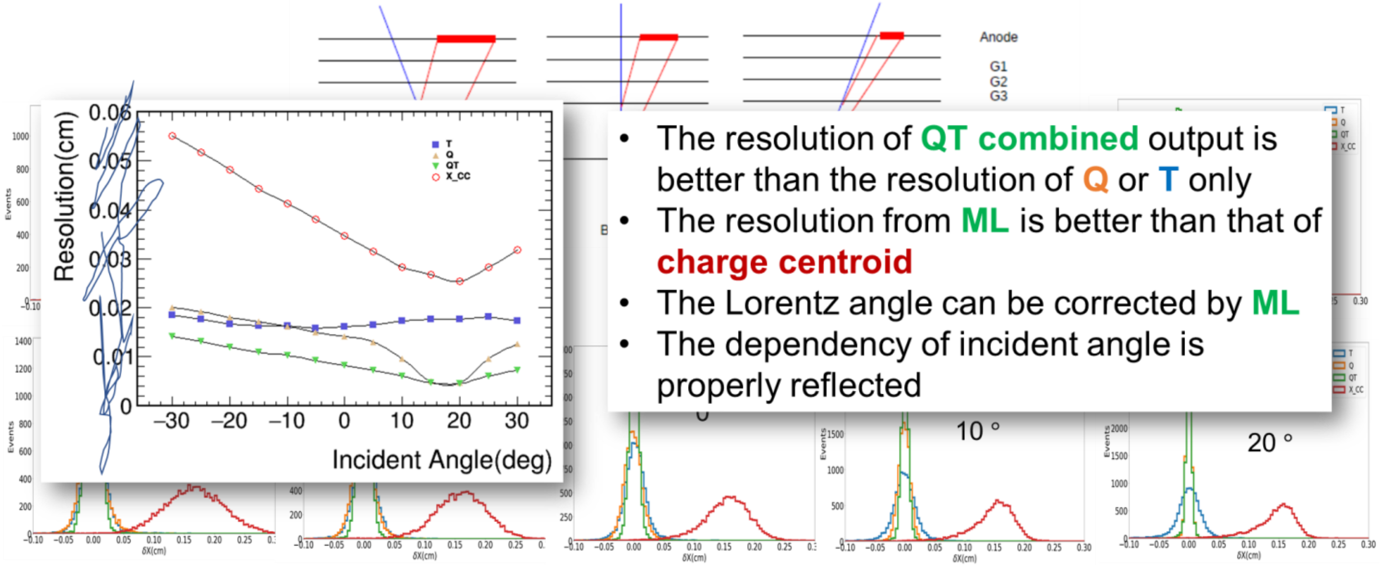

nature
detector
event
generator
simulation of
detector
response
reconstruction
event selection
Statistical
inference

Cluster reconstruction of CGEM-IT

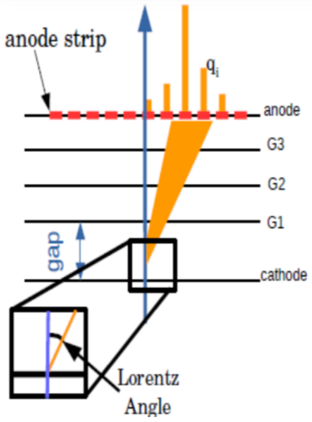




B. Liu et al., EPJ Web Conf. 214, 06033 (2019)
Using XGBOOST as a regressor to measure the initial ionizing particle position X from Q and T of the fired strips



- The resolution of **QT combined** output is better than the resolution of **Q** or **T** only
- The resolution from **ML** is better than that of **charge centroid**
- The Lorentz angle can be corrected by **ML**
- The dependency of incident angle is properly reflected



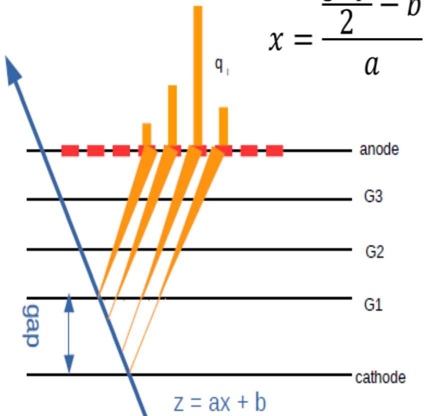
With Q: Charge Centroid

$$< x > = \frac{\sum_i x_i q_i}{\sum_i q_i}$$

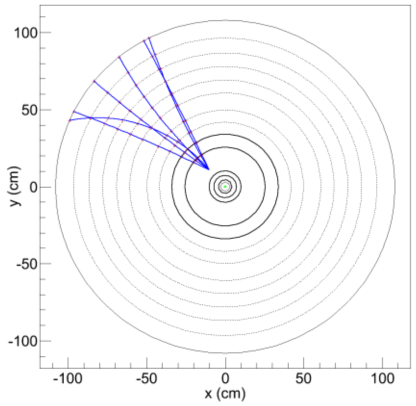
With T: Micro TPC

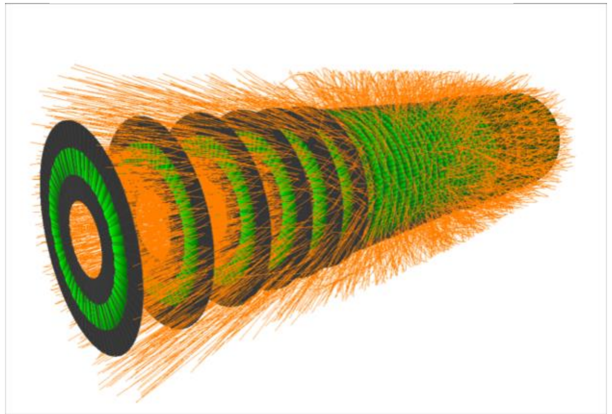
[Alexopoulos et al, NIM A617 (2010) 161]

$$x = \frac{\frac{gap}{2} - b}{a}$$



Tracking: *Charged particle reconstruction*



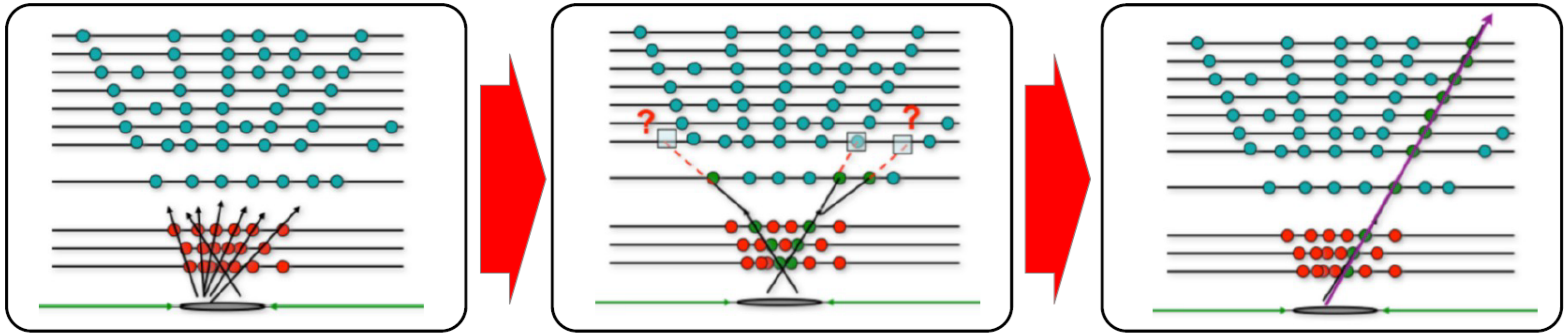


Challenging for HL-LHC

Tracking in a Nutshell

- Particle trajectory bended in a solenoidal magnetic field
- Curvature is a proxy to momentum
- **Thousands of sparse hits**
- Hits pollution from low momentum, secondary particles

Seeding | Combinatorial Kalman Filter | Fitting with Kalman Filter



Scaling performance and limits in computation budget
call for faster algorithms

Machine Learning in Tracking

- Seeding and Clustering
- Pattern recognition
- Track Selection
- Track Parameters
- Vertexing

A very active field

TrackML — Particle tracking challenge (kaggle)

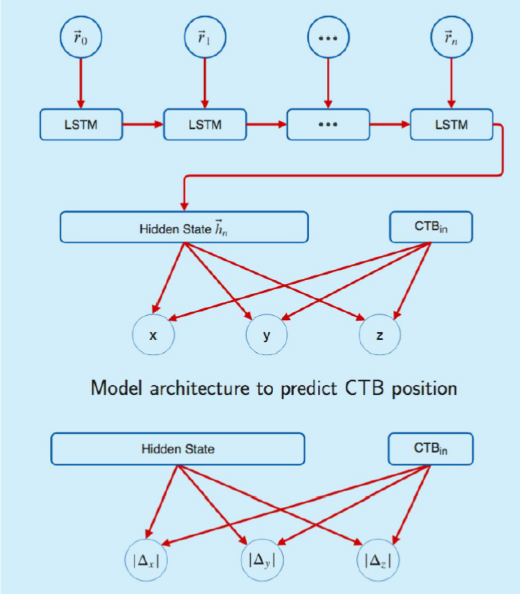
HEP.TrkX — HEP advanced tracking algorithms with cross-cutting applications (Project HEP.TrkX)

Exa.TrkX — HEP advanced tracking algorithms at the exascale (Project Exa.TrkX)

CTD/WIT 2019
Connecting the Dots and Workshop on Intelligent Trackers

track reconstruction in LHCb's Vertex Locator

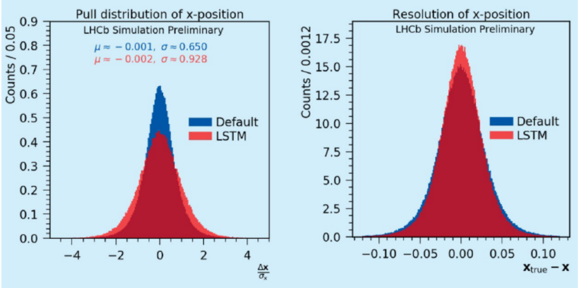
Impact Parameters

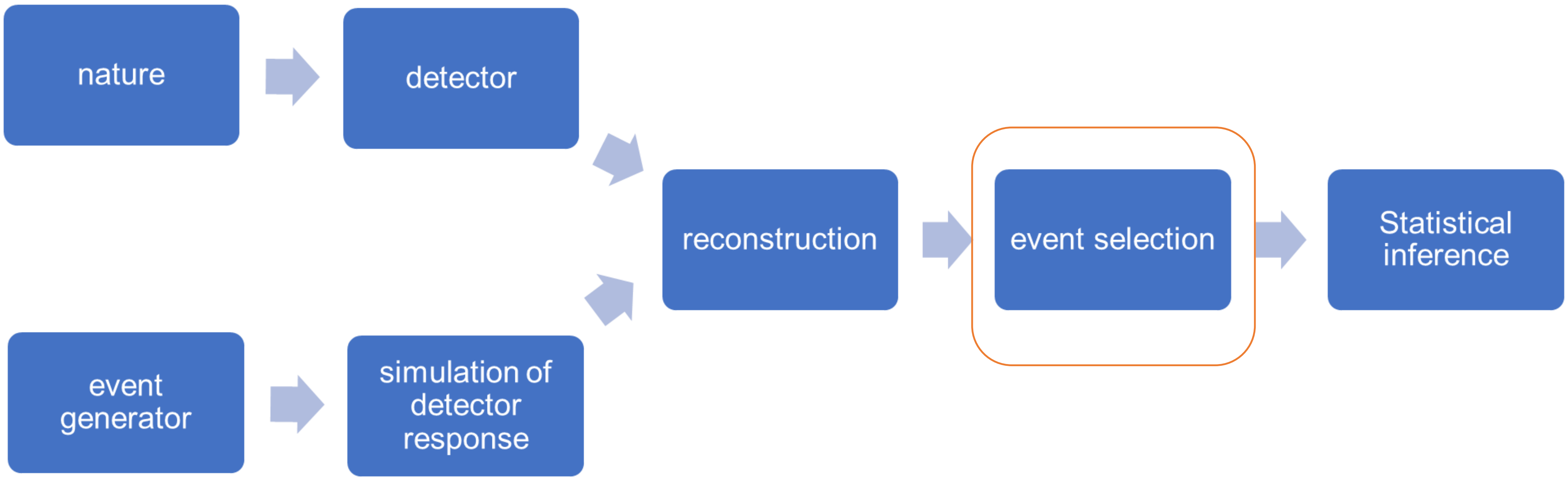


Model architecture to predict CTB position

Model extension to predict the uncertainty on the CTB position

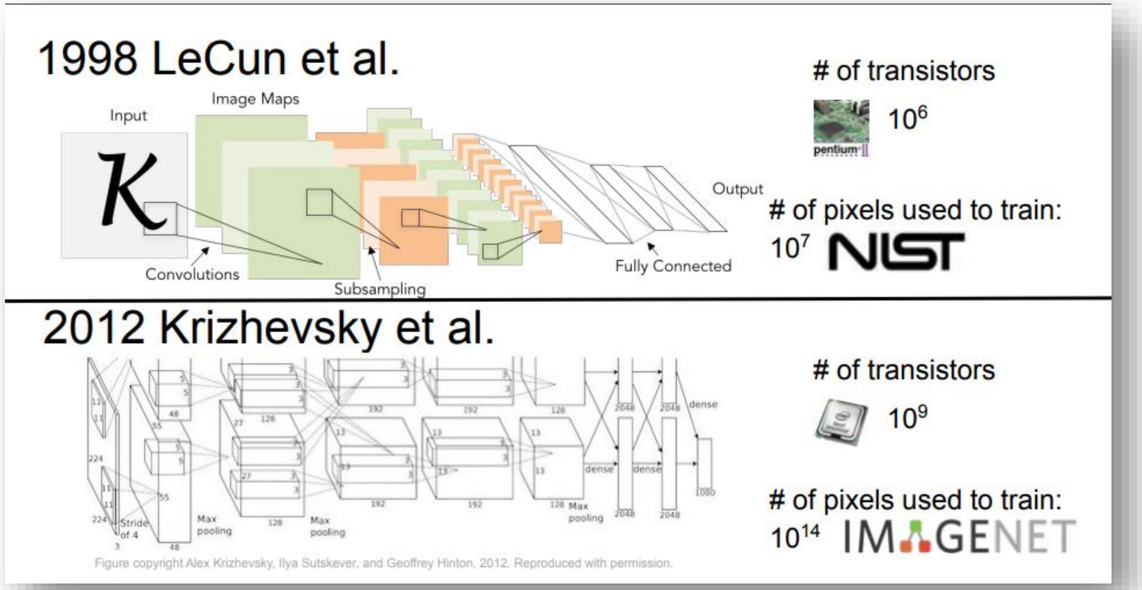
- LSTM model supplements a Kalman Filter approach
- Improve resolution and estimation of track impact parameters in LHCb



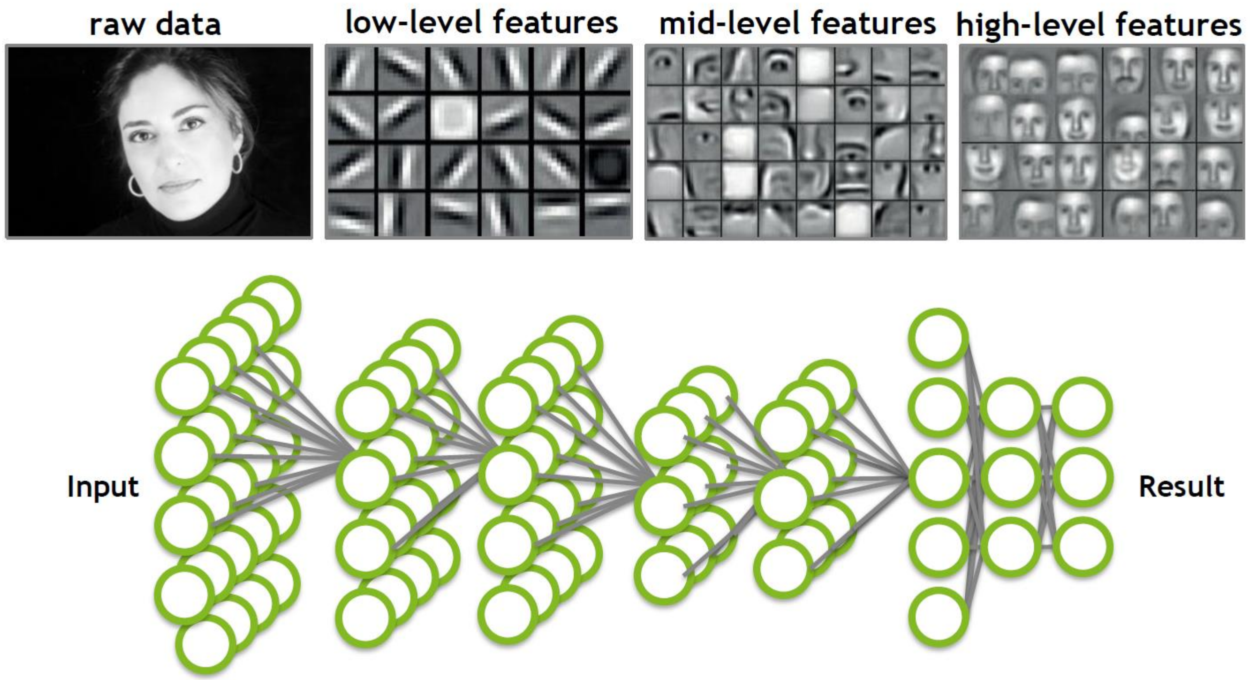

nature
detector
event generator
simulation of detector response
reconstruction
event selection
Statistical inference

Classification with Convolutional Neural Networks

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



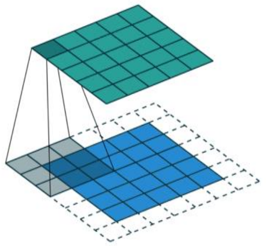
[CS231n]



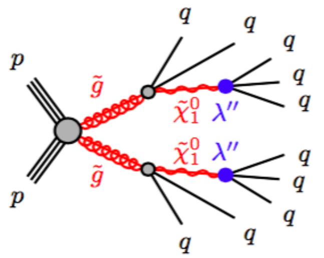
[Nvidia]

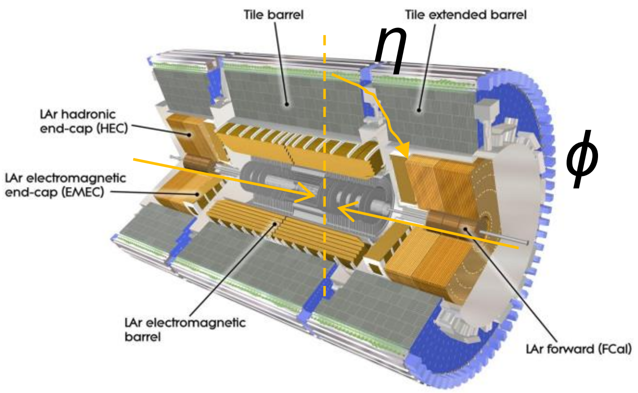
Classification with Convolutional Neural Networks

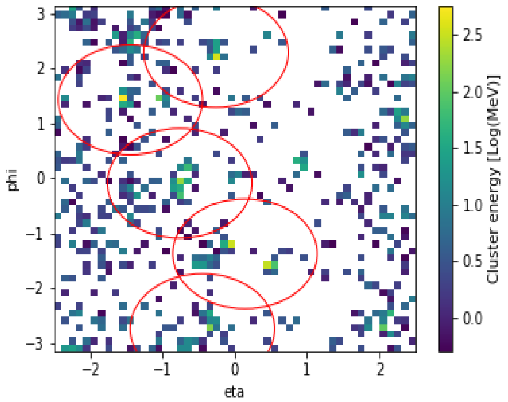
- CNN – shared non-linear filters; reduce weights; exploit locality and symmetries: now popular in many science studies
- E.g. LHC-CNN: Unroll cylindrical detector data for image[1]; classify known (QCD) vs new physics (RPV supersymmetry)
  - Use 3 channels for EM and HCal Calorimeters and number of 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.

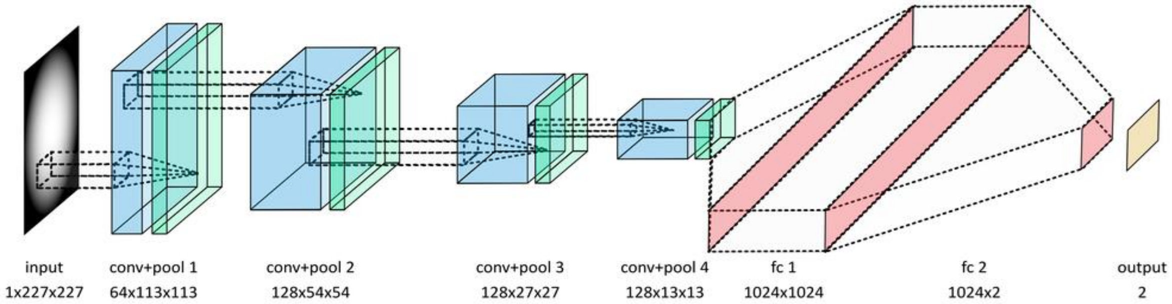


ATLAS-CONF-2016-057









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

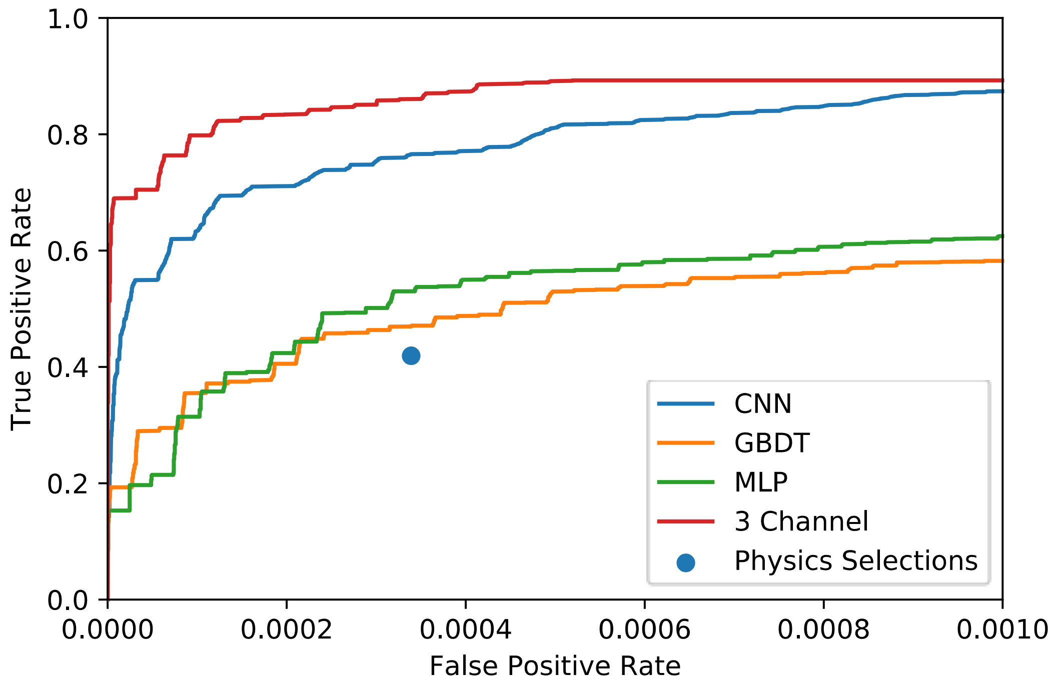
[1] As also in de Oliviera et. al. (*arXiv:*1511.05190) and others

[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

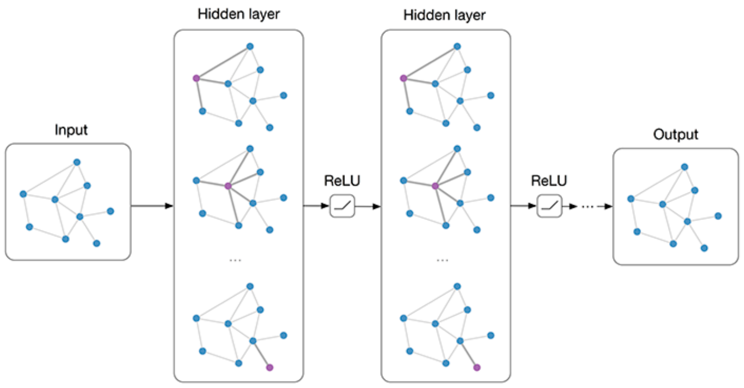
- Use re-implementation of existing physics selections on jet variables from ATLAS-CONF-2016-057 as a benchmark
- Also compare to boosted decision tree (GBDT) and 1-layer NN (MLP)
- Input to these jet variables used in the physics analysis (Sum of Jet Mass, Number of Jets, Eta between leading 2 jets)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



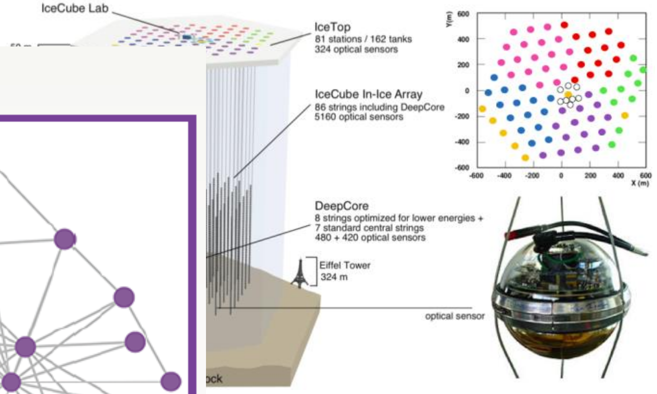
Potential to increase signal efficiency (from 0.41 to 0.77) at same background rejection as selections without using jet variables (approximate significance increase of 1.8x)

Further improvement from using 3-channels: Energy in E-Cal, H-Cal and No. tracks https://arxiv.org/abs/1809.06166

Graph CNNs

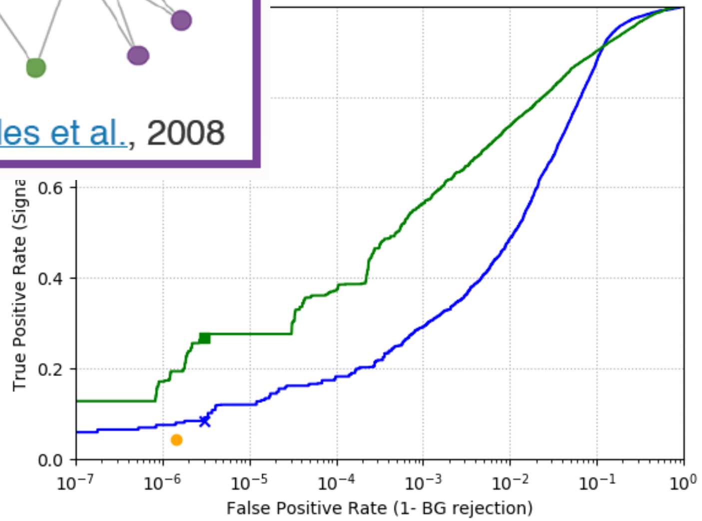


- **Use detector deposits rather than an image in a GraphCNN: Represent signals as nodes of a graph with similarity as edge weights**



- Graph networks
  - No sorting required
  - No grid
  - Sense of connection
  - Basic principle: information exchange through edges (connections)
  - Very active area of research in CS

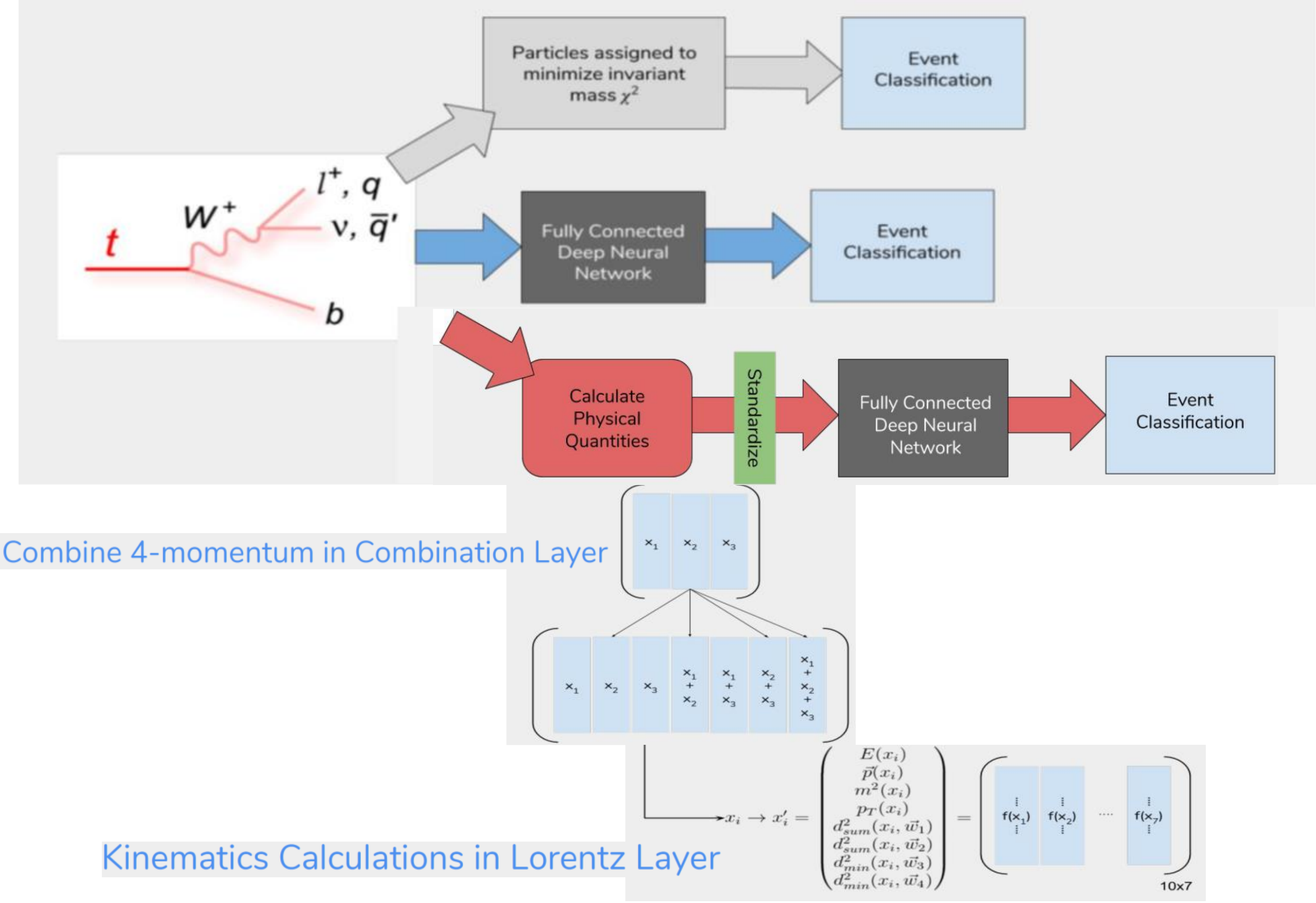
Brandes et al., 2008

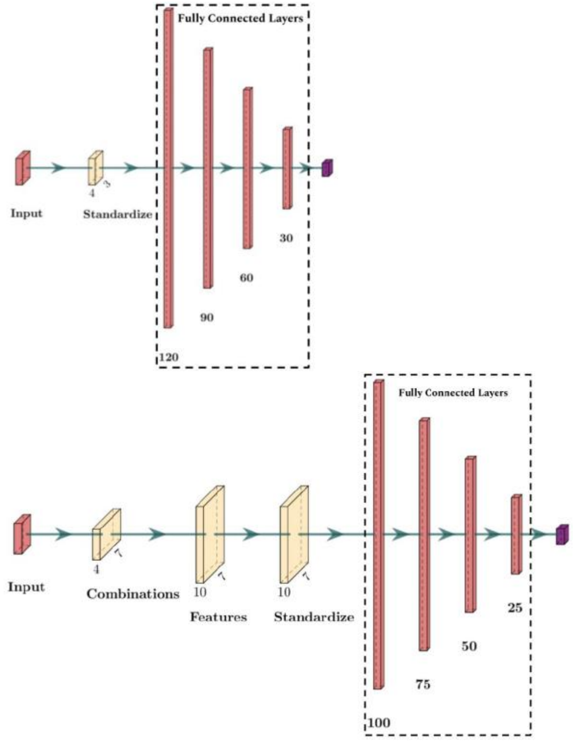


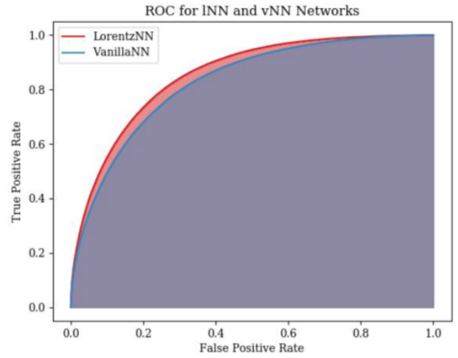
- **Compared with ResNet-18 3D CNN with data on grid and physics baseline (tuned cuts on stochasticity)**

Domain aware / physics informed / physics inspired ML algorithms

“Physics Inspired” DNN: “*Deep Learned Top Tagging with Lorentz Layer*, SciPost Phys. 5, 028 (2018)”

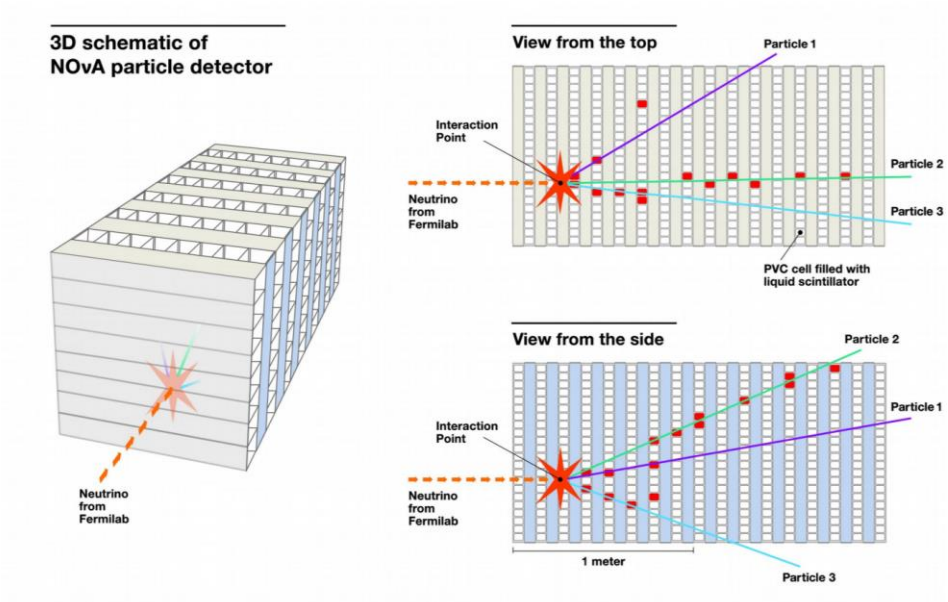






Physics inspired DNNs provided a modest performance boost over standard fully connected DNNs

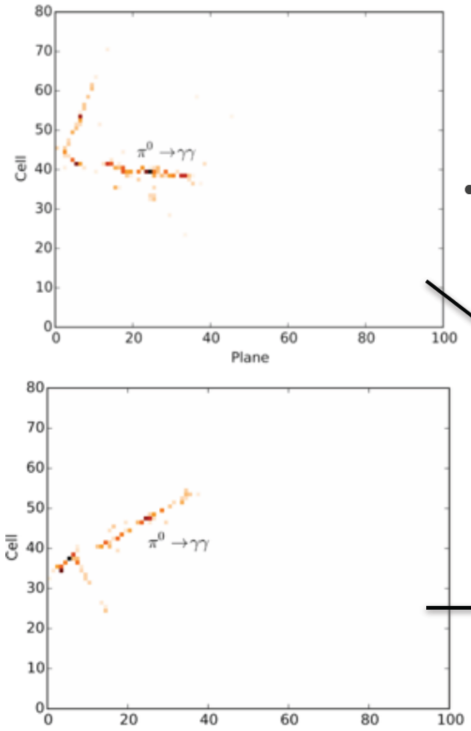
Neutrino Flavor Classification



- NOvA was the first HEP experiment to use CNN to extract published physics results

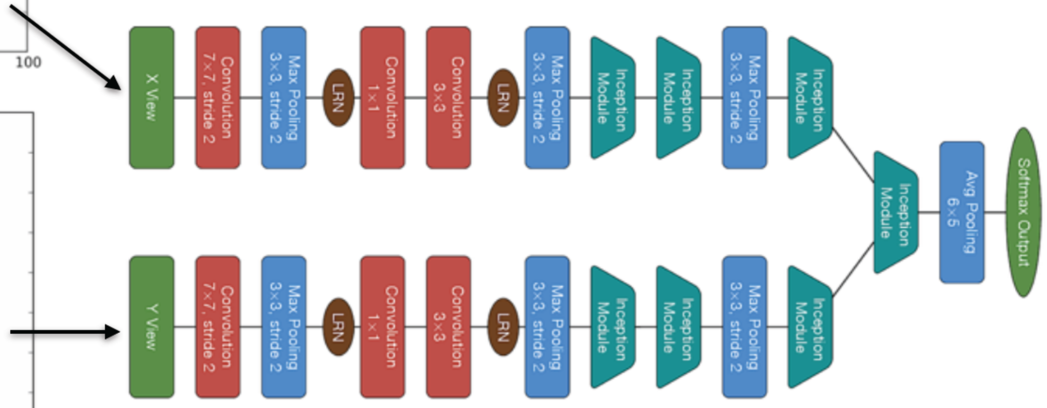
- It improved the headline analysis performance by 30%, equivalent to an equipment savings of approximately $72 million

A. Aurisano and A. Radovic and D. Rocco et. al, JINST 11 P09001 (2016)



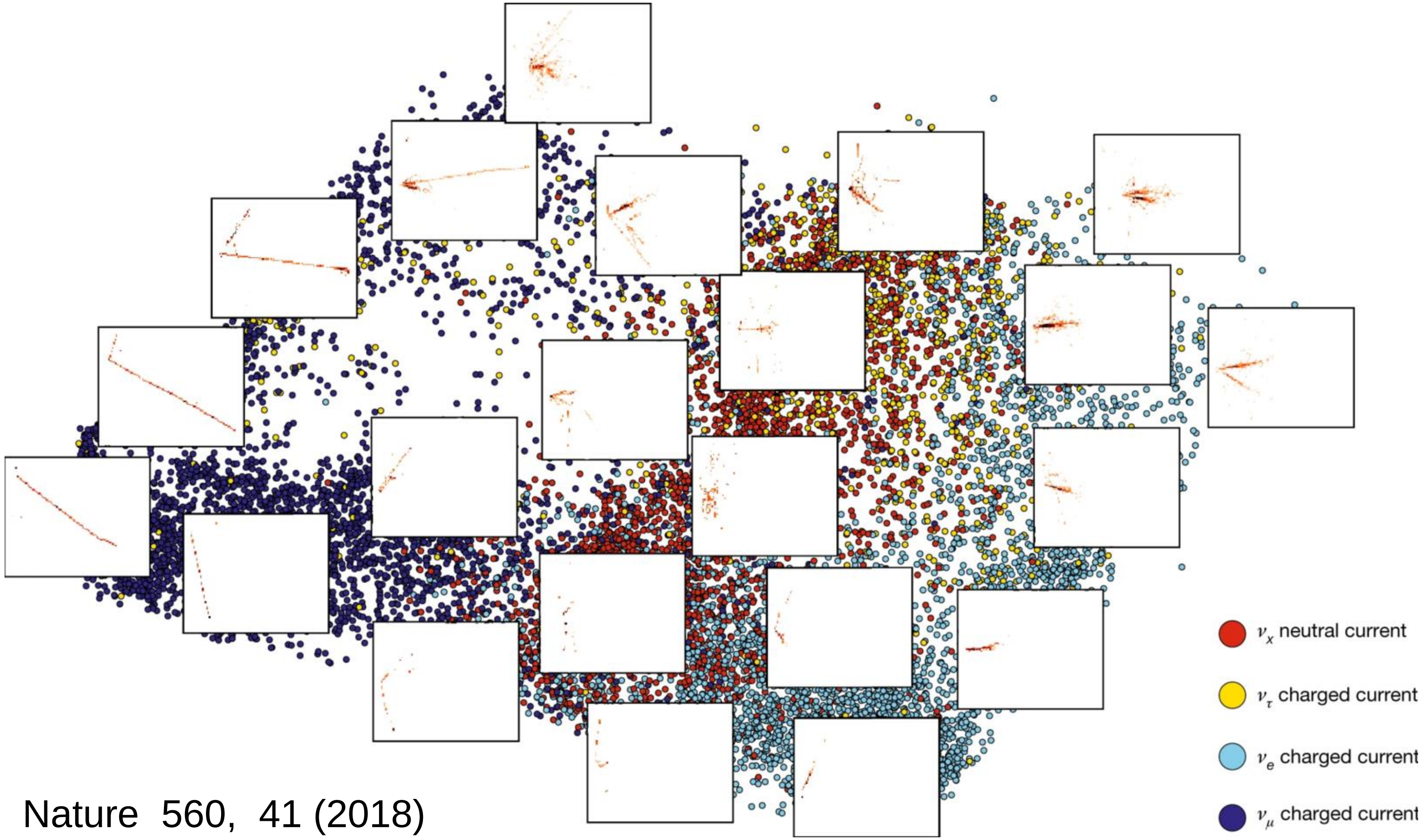
» Create a bi-columnar networks with shared weights
» Split views early to extract parallel features
» Merge together at the end before going through fully connected layers
» Ends with a feed forward neural network to create multi-classification

- ***Trained on 4.5+ million Monte Carlo beam events combined with cosmic ray data***



Going Deeper with Convolutions (arXiv:1409.4842)

Understanding the Network: Feature Embedding with t-SNE



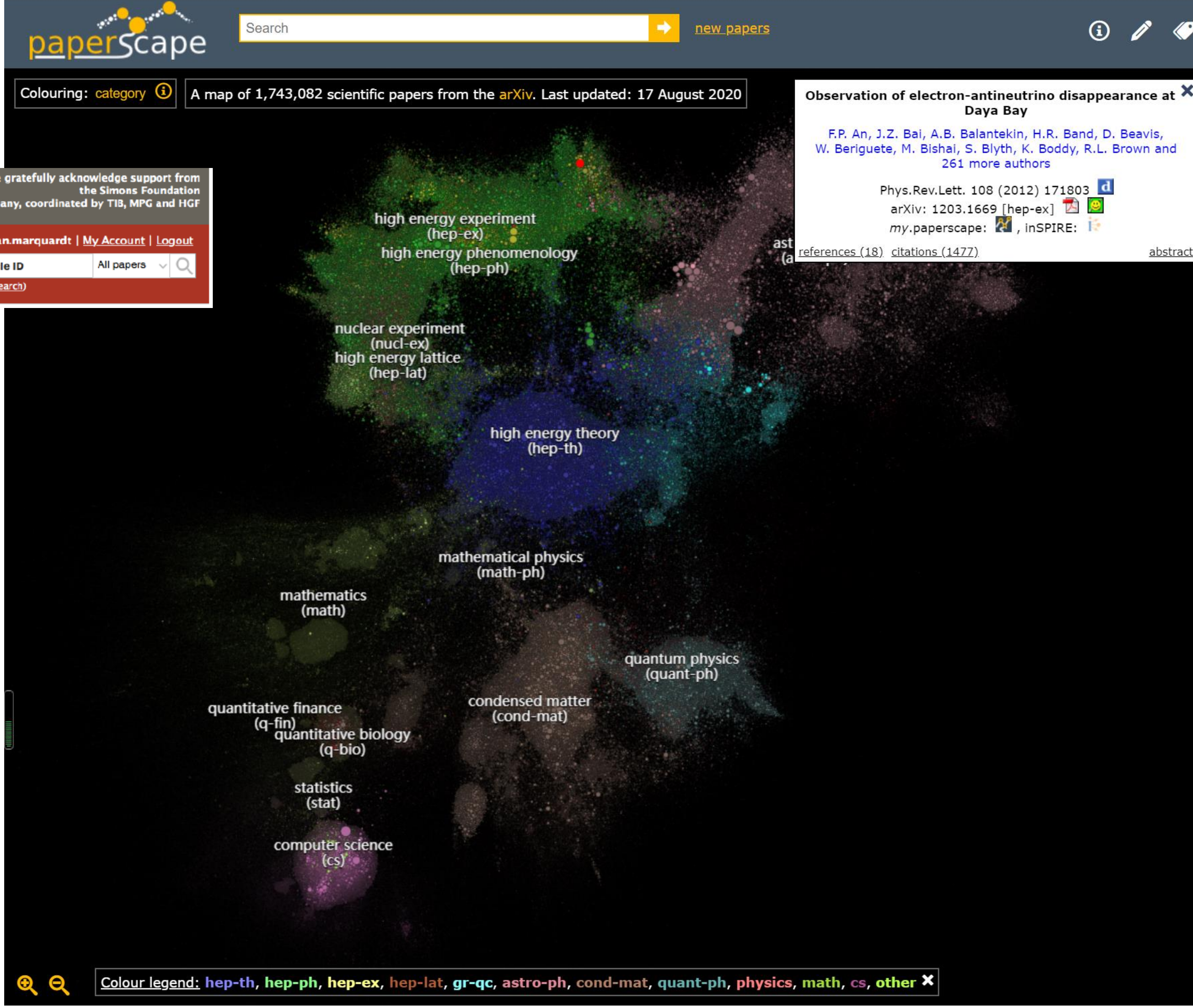
Nature 560, 41 (2018)

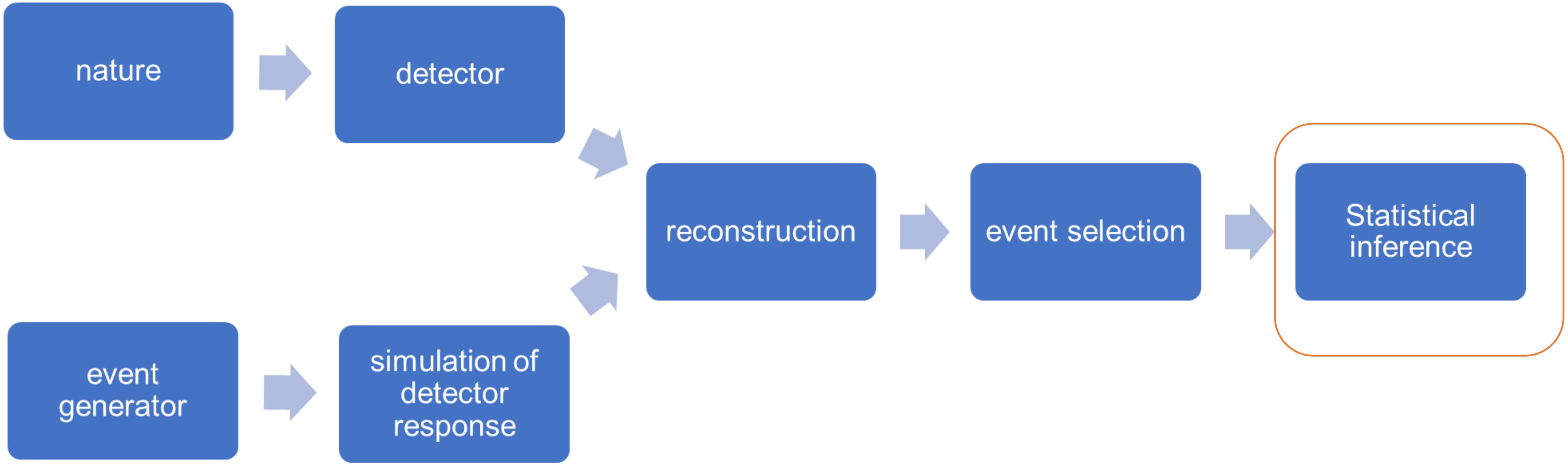
The various types of event are clustered into distinct regions in the horizontal direction, while the multiplicity of the particles in each event is found to be correlated with the location of the events in the vertical direction.

https://indico.io/blog/visualizing-with-t-sne/

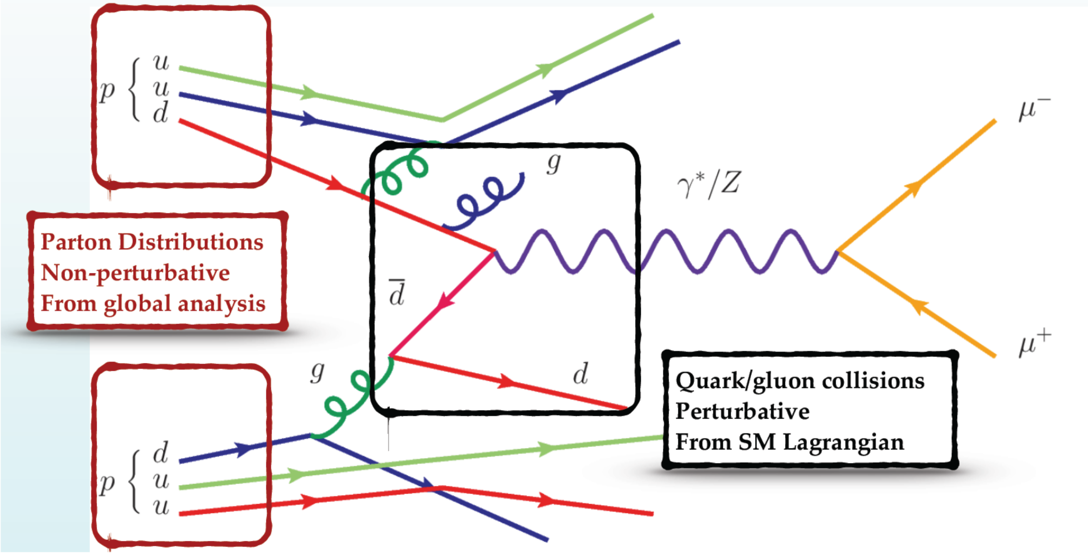
An other example

Paperscape uses a simple physical
(similar to t-SNE)




nature
detector
event generator
simulation of detector response
reconstruction
event selection
Statistical inference

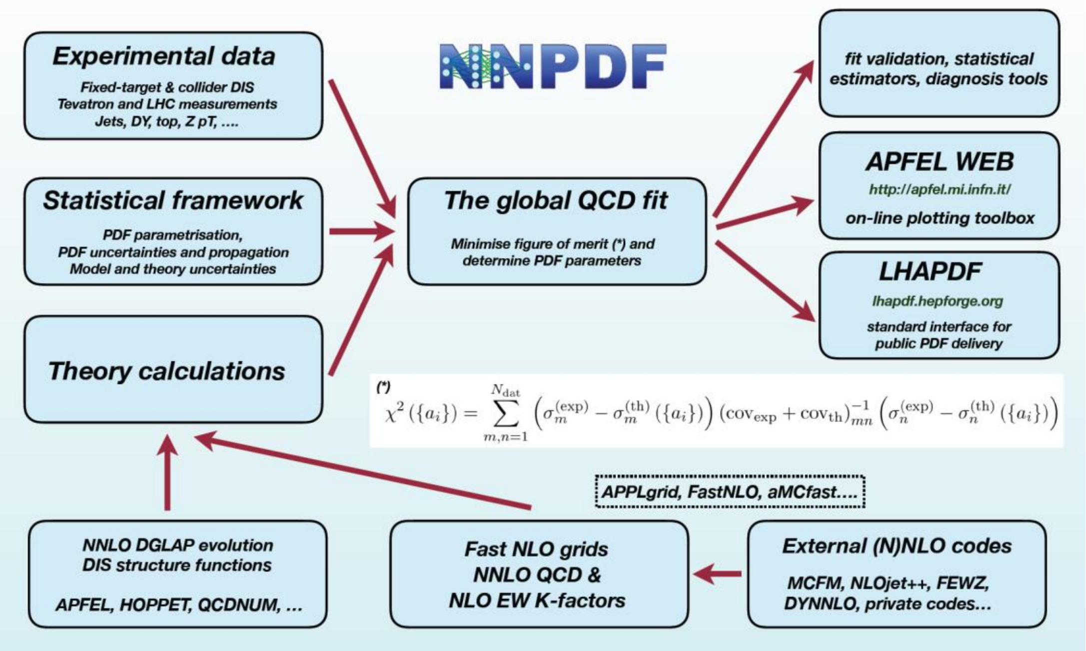
The determination of Parton Distribution Functions (PDFs)



Traditional approach $g(x,Q_0) = A_g(1-x)^{a_g}x^{-b_g}\left(1 + c_g\sqrt{s} + d_g x + \ldots\right)$

NNPDF approach $g(x,Q_0) = A_g \mathrm{ANN}_g(x)$

Machine Learning for PDF fits

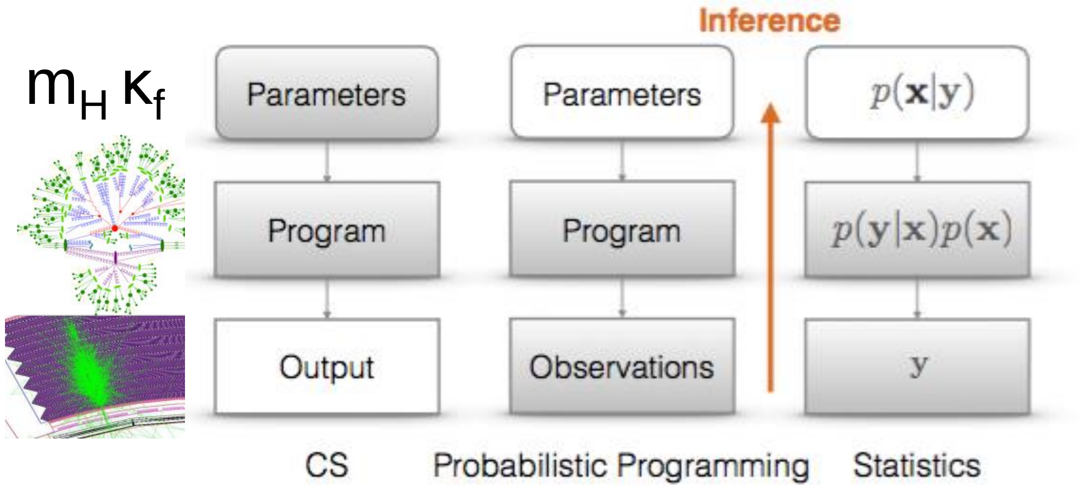


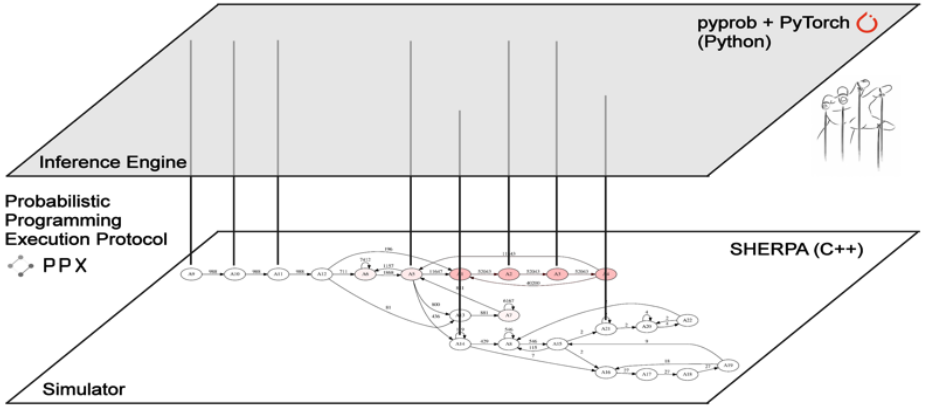
Simulation-based (`likelihood-free') Inference

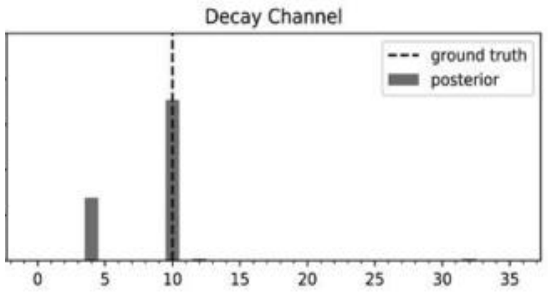
Atilim Gunes Baydin, Bradley Gram-Hansen (Oxford)
Lukas Heinrich, , Kyle Cranmer (NYU) Wahid Bhimji,
Prabhat (NERSC) Gilles Louppe (Liege), Lei Shao (Intel),
Frank Wood (UBC) https://arxiv.org/abs/1807.07706

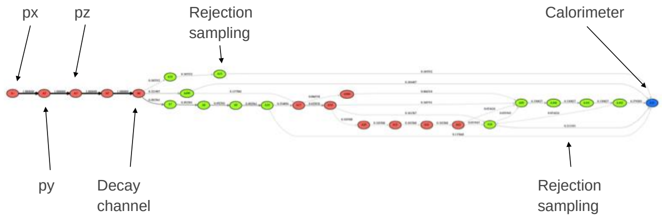
- In HEP/NP often have detailed simulation (forward model) of physics and detector
  - Ideally could ‘invert’ this to perform inference on real data – not easily done
- ‘Invert’ via probabilistic program (PPL) and embedding approach
  - PPL: ***Sample*** from distribution (already in HEP sim. E.g. SHERPA) and ***Condition*** on observation
  - 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









More …

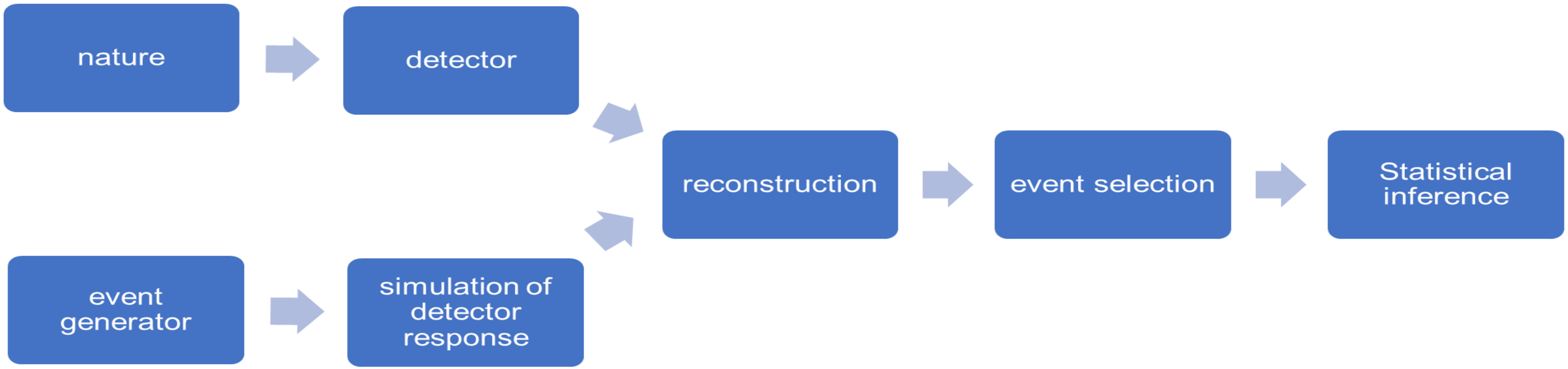
Anomaly Detection

E.g. Hardware monitoring, Comput.Softw.Big Sci. 3 (2019) 1, 3

Model-Independent Searches for New Physics, EPJC 79, 289 (2019)

Computing Resource Optimization

E.g. J.Phys.Conf.Ser. 1525 (2020) 1, 012042



ML←→HEP(physics)

Physics-inspired ML approaches

Simulated Annealing
MCMC techniques
Gibbs sampling
Gaussian process
Gradient descent
Boltzmann Machine
Energy-based GANs

Incorporation of domain knowledge

Why Does Deep and Cheap Learning Work So Well?, arXiv:1608.08225
Interaction Networks for Learning about Objects, Relations and Physics, arXiv:1612.00222
Covariance in Physics and Convolutional Neural Networks, arXiv:1906.02481
…

One of ML Challenges in HEP

Robustness to systematic uncertainties

- develop techniques that are more data efficient by incorporating domain knowledge directly into the machine learning models;
- incorporate the uncertainties in the simulation into the training procedure;
- develop weakly supervised procedures that can be applied to real data and do not rely on the simulation;
- improve the tuning of the simulation, reweight or adjust the simulated data to better match the real data, or use machine learning to model residuals between the simulation and the real data;

Some general advice

- No free lunch
  - Try many algorithms, starting with simple ones

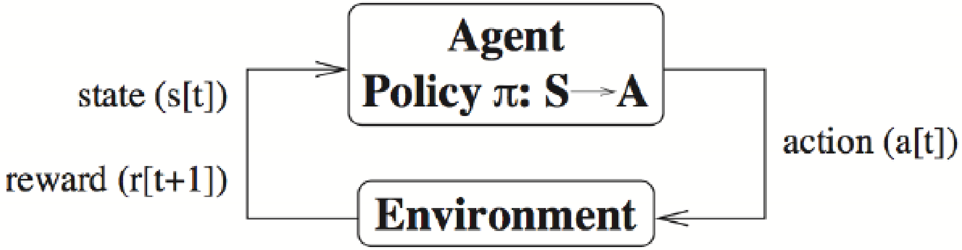
- Mapping your problem to ML field
  - Check the literature

- **Incorporating domain knowledge into the machine learning models**

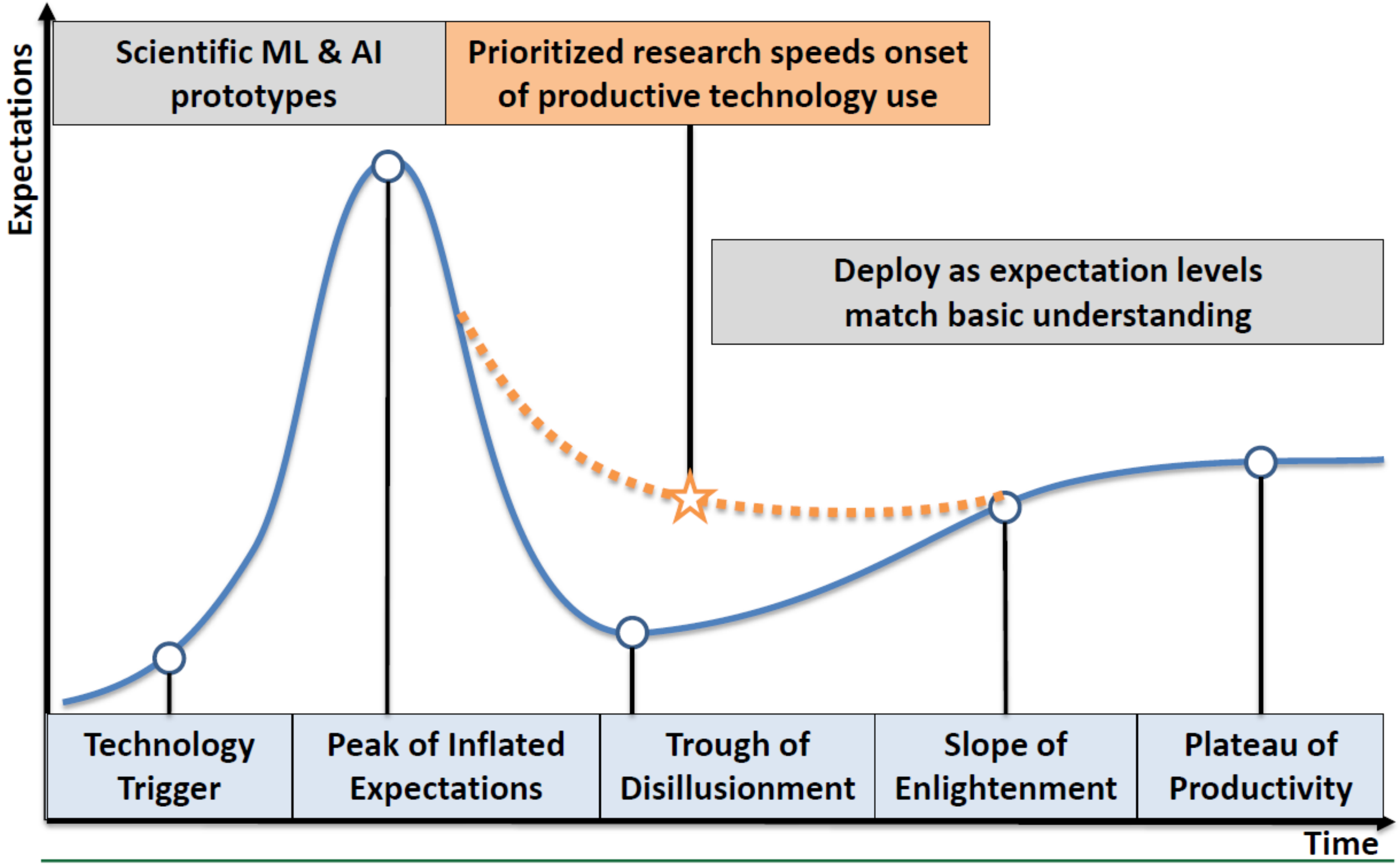
Thank you for your attention

Reinforcement Learning

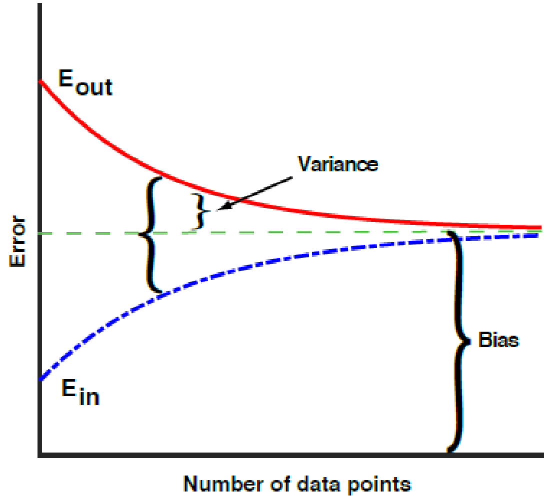
- Models for agents that take actions depending on current state
  - Actions incur rewards, and affect future states ("feedback")
- Learn to make the best sequence of decisions to achieve a given goal when feedback is often delayed until you reach the goal

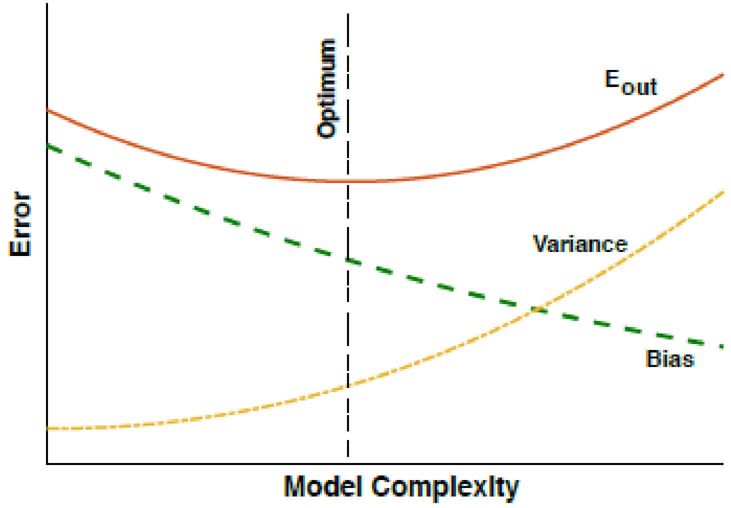


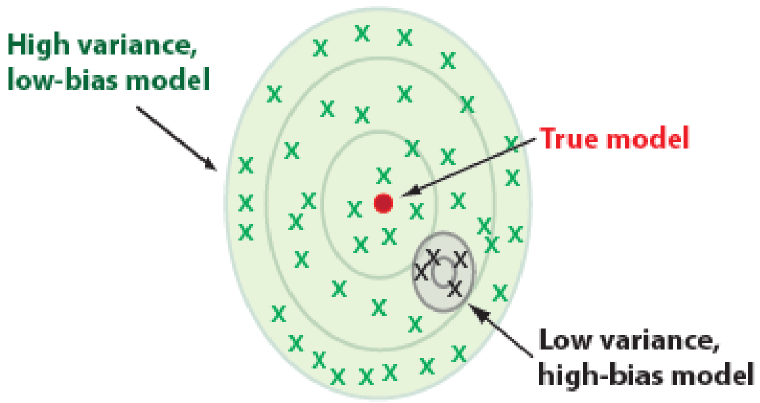
Foundational Research will increase our basic understanding of Scientific Machine Learning & AI technologies



arXiv:1803.08823







A bit of history:

Hubel & Wiesel,

1959

RECEPTIVE FIELDS OF SINGLE NEURONES IN
THE CAT'S STRIATE CORTEX

1962

RECEPTIVE FIELDS, BINOCULAR INTERACTION
AND FUNCTIONAL ARCHITECTURE IN
THE CAT'S VISUAL CORTEX

1968...