# Introduction to Machine Learning

CERN Openlab Summer Student Lectures
14 July 2022

Jean-Roch Vlimant (California Institute of Technology)
jvlimant@caltech.edu @vlimant

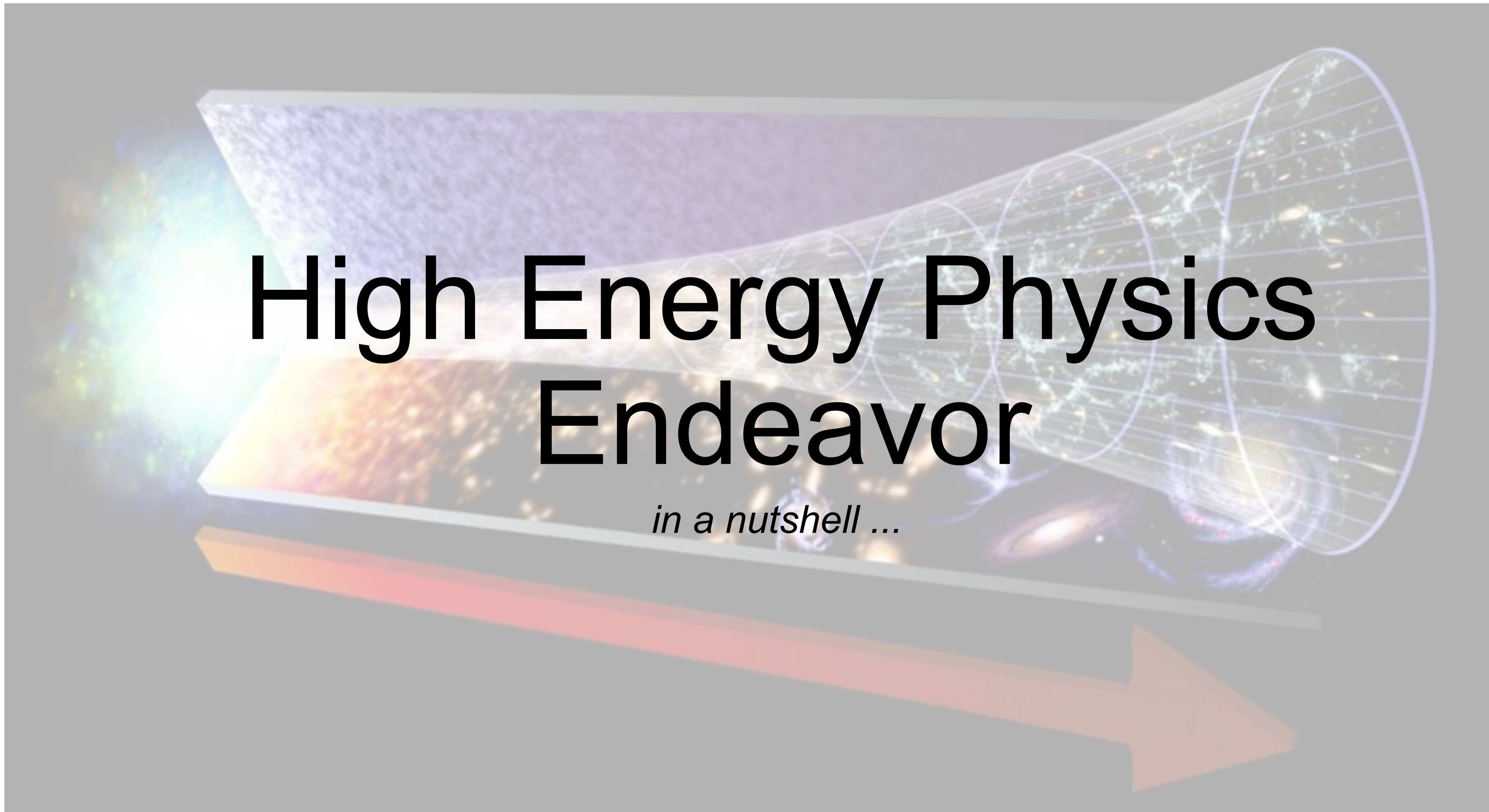
# Scope of this Lecture

- Basics of machine learning were introduced in Glen Cowan's Lecture https://indico.cern.ch/event/1132551/ . We will only **go over the essentials** here.
- Impossible to "learn machine learning" with this lecture. The aim is at pointing out key aspects for **doing "good machine learning"** and the specifics to high energy physics applications.
- Emphasis on what you need to bear in mind while **developing Machine learning at collider** to avoid common pitfalls.
- Lots of references provided therein for **further reading** and understanding. Also in https://github.com/iml-wg/HEPML-LivingReview

*This work is partially supported by the U.S. DOE, Office of Science, Office of High Energy Physics under Award No. DE-SC0011925, DE-SC0019219, and DE-AC02-07CH11359.*

# Outline

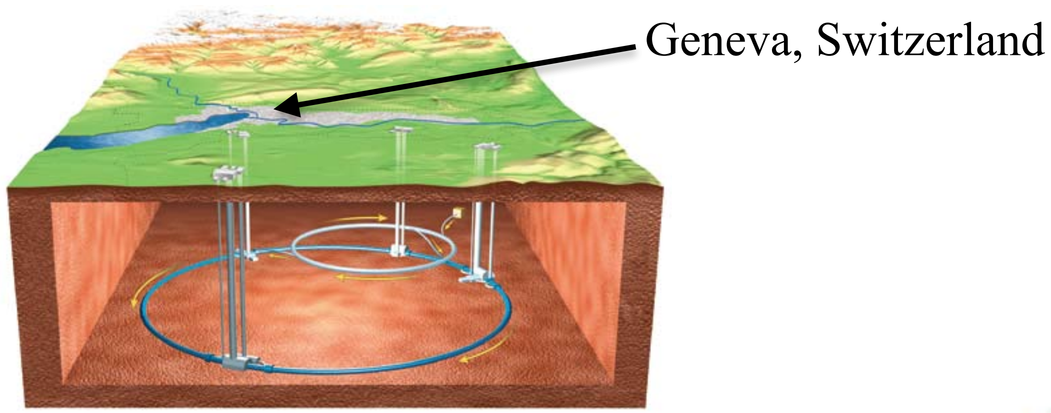
I. Physics at the Large Hadron Collider
II. A glimpse at the Machine Learning Landscape
III. Motivations for using Machine Learning
IV. Deep-learning in the HEP data pipeline
V. Collider-Specific AI

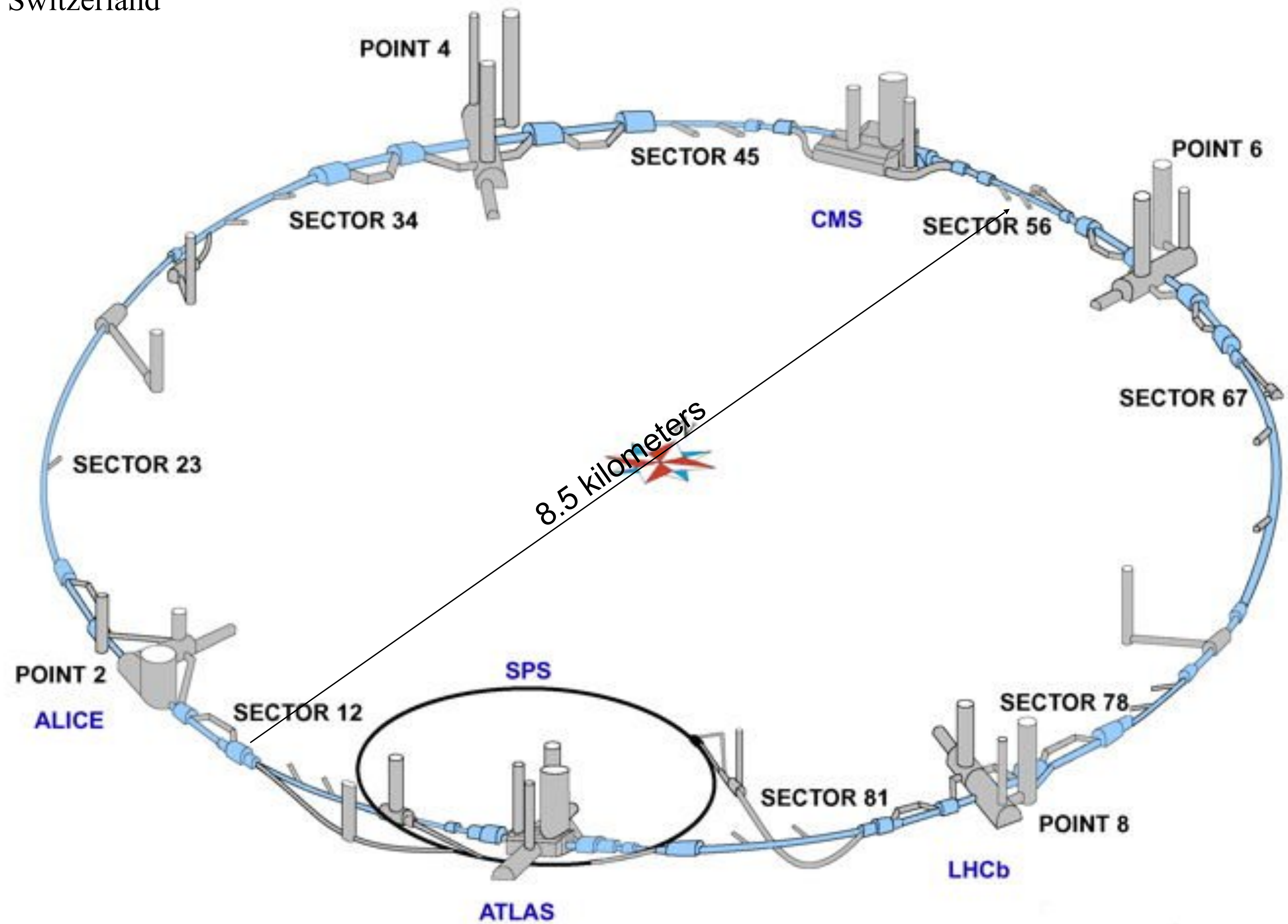


# High Energy Physics Endeavor

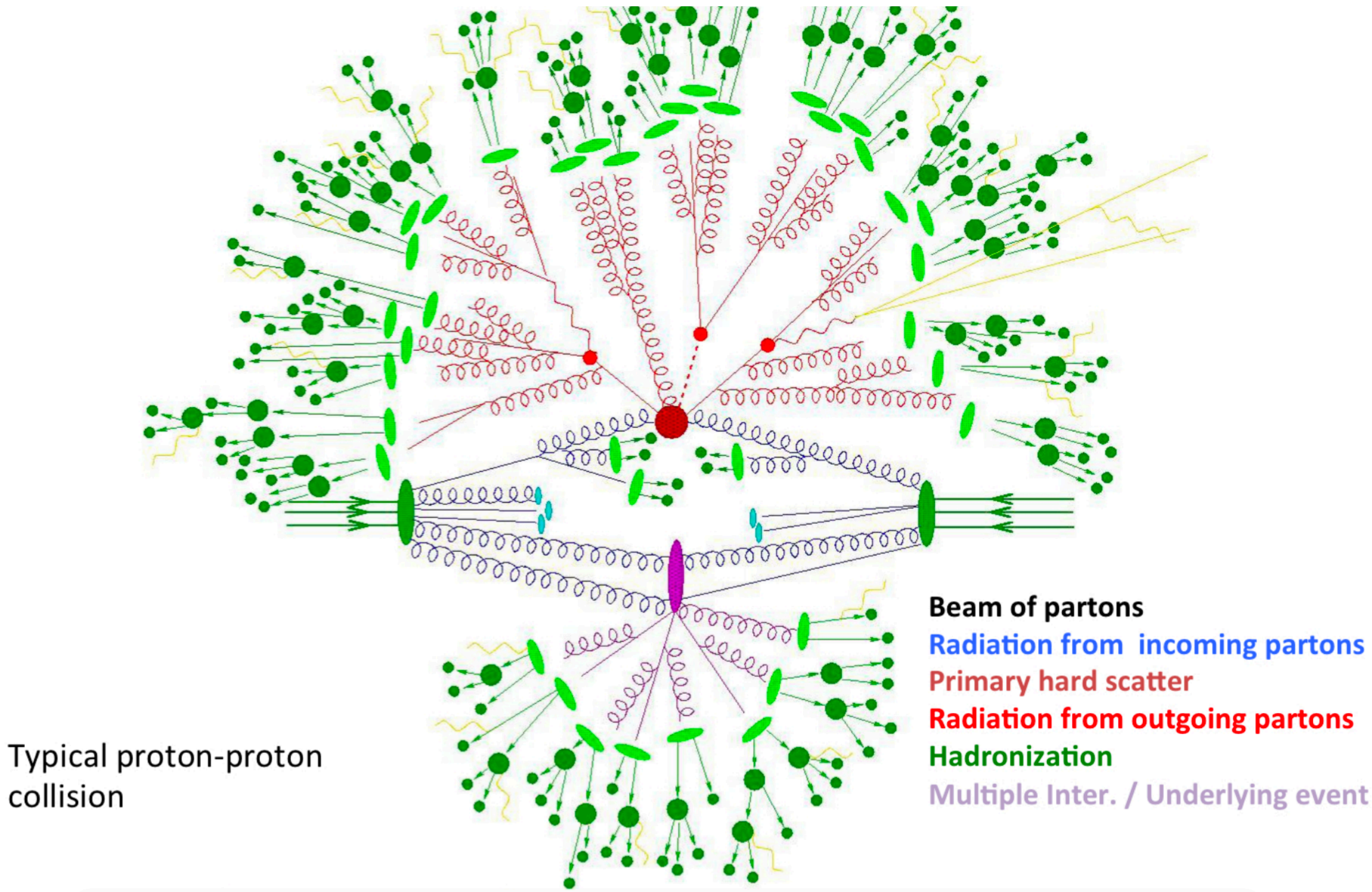
*in a nutshell ...*

# The Large Hadron Collider



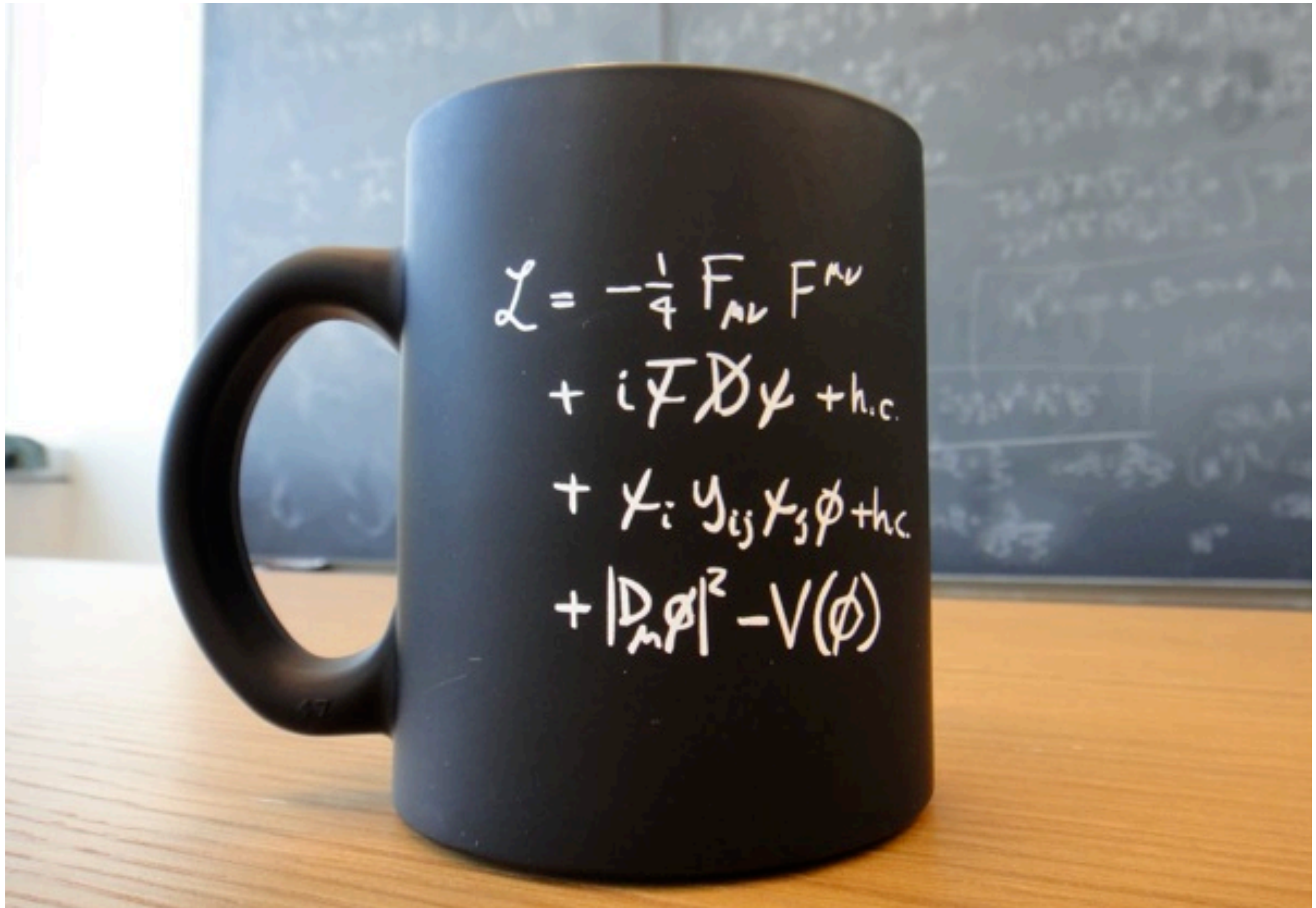



# Colliding Hadrons




Probing fundamental laws of physics as large spectrum of particles (known and unknown) can be produced

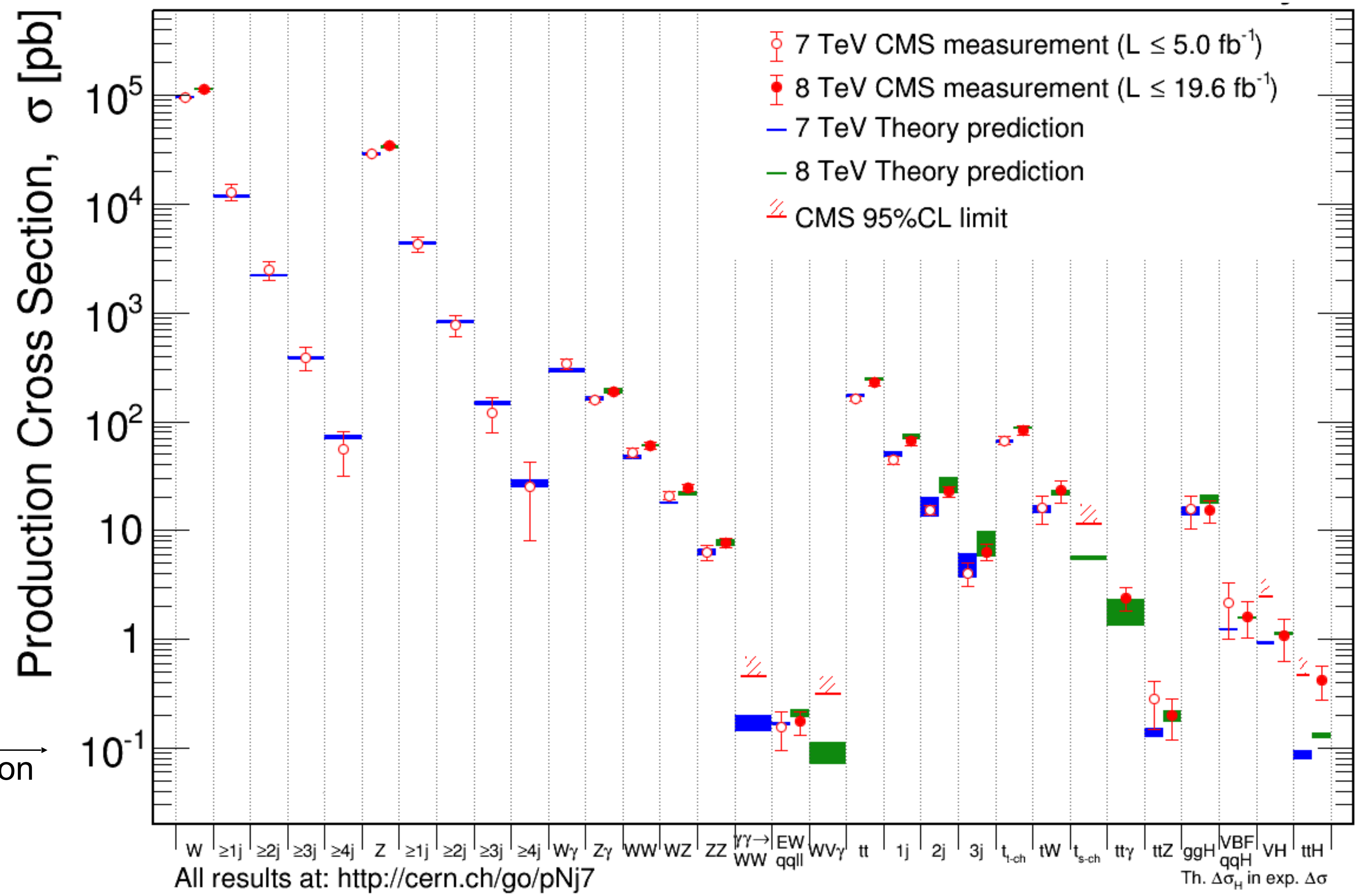
# The Standard Model




Well demonstrated effective model
We can predict most of the observations
We can use a large amount of simulation

# Size Of The Challenge

1 event every 500.000 proton collision

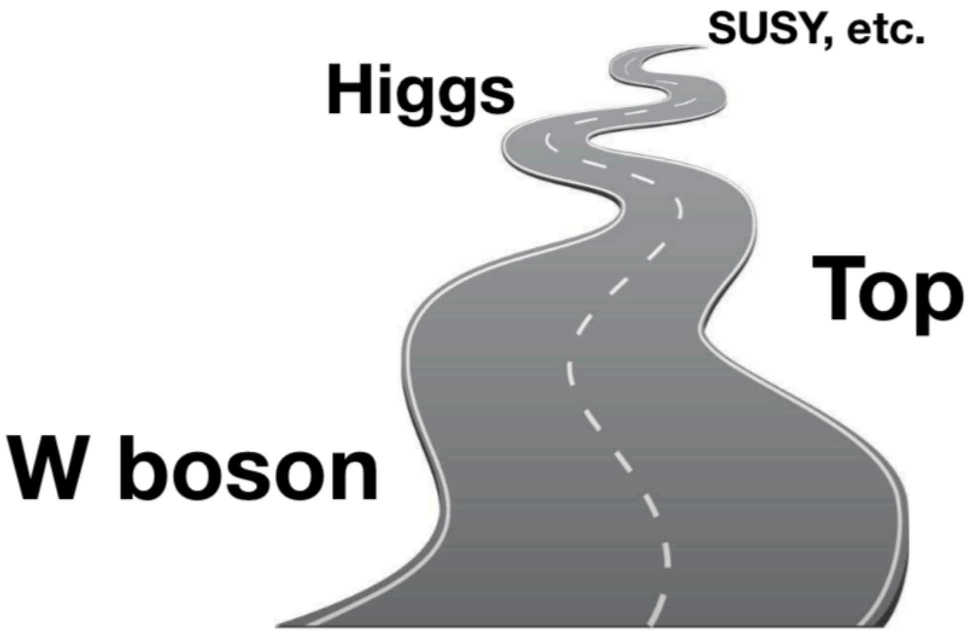



Low probability of producing exotic and interesting signals.
Observe rare events from a large amount of data.

# The Sea Beyond Standard Model

Slide: A. Wulzner [H&N]

## HEP yesterday




"Almost" **Simple $H_1$**

Focus on **few sharply-defined** alternative models (e.g., the Higgs)

Case-by-case design of **optimal test**

## HEP today

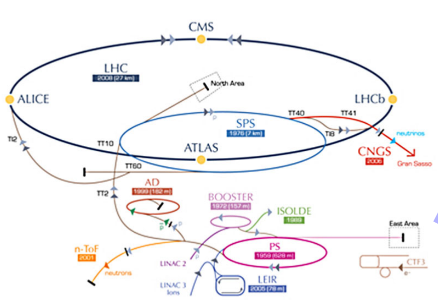

"Very" **Composite $H_1$**

**Huge set** of alternatives

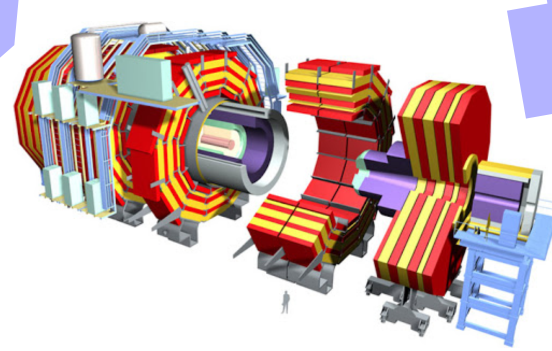
Case-by-case optimisation **unfeasible**

The **right $H_1$** likely **not yet formulated**

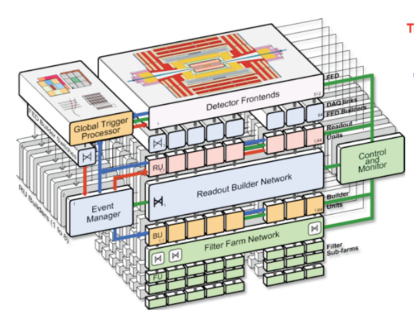
# HEP Data Pipeline




Large Hadron Collider
40 MHz of collision

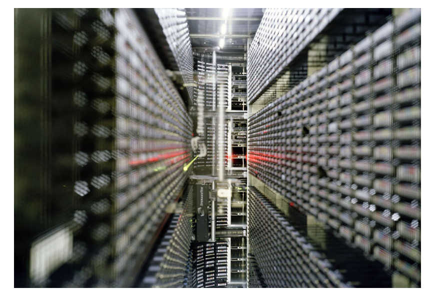


CMS Detector
1PB/s

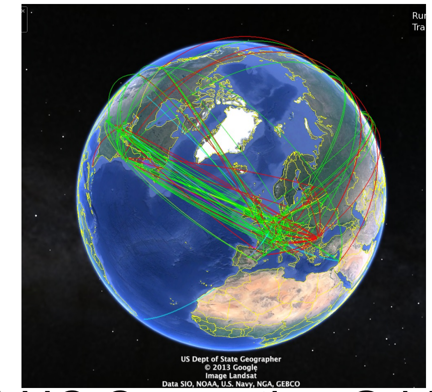



CMS L1 & High-Level Triggers
50k cores, 1kHz

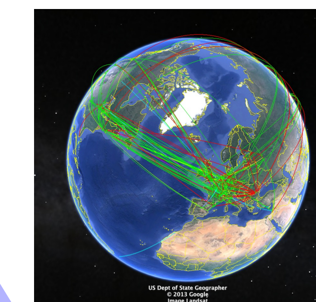
CERN Tier-0
Computing Center
20k cores



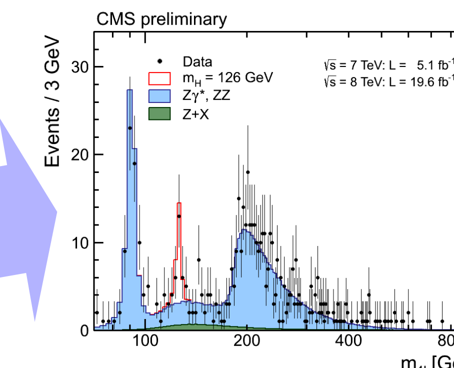
CERN Tier-0/Tier-1
Tape Storage
200PB total



LHC Computing Grid
200k cores pledge to
CMS over ~100 sites



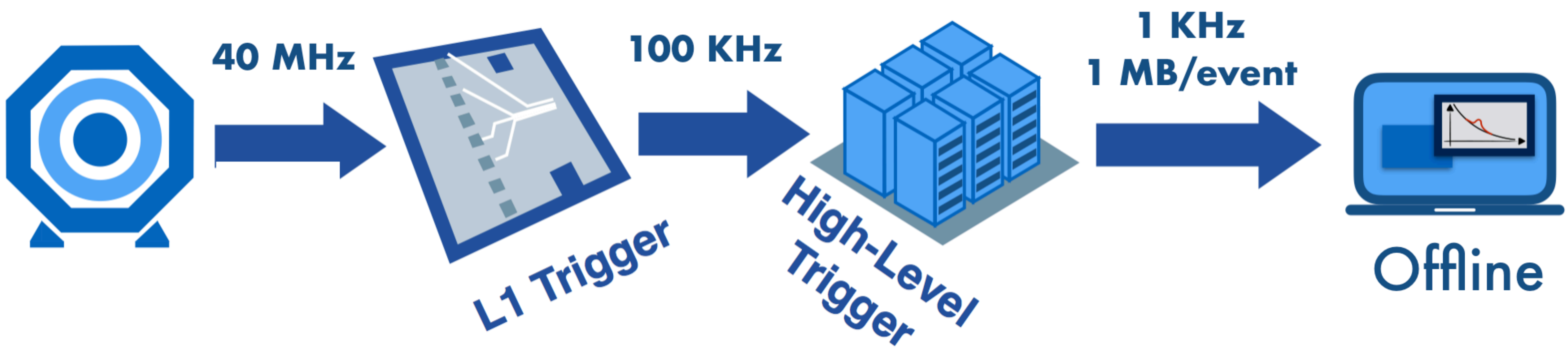
LHC Grid
Remote Access
to 100PB of data




Rare Signal
Measurement
~1 out of $10^6$

# Event Triggering

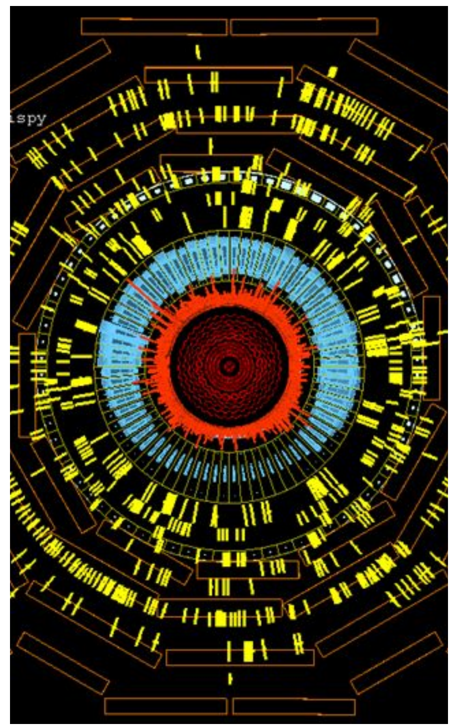
Select what is important to keep for analysis.
Ultra fast decision in hardware and software.

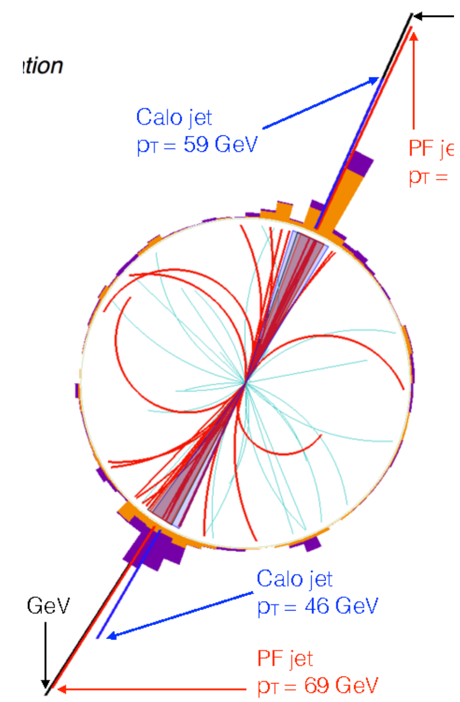


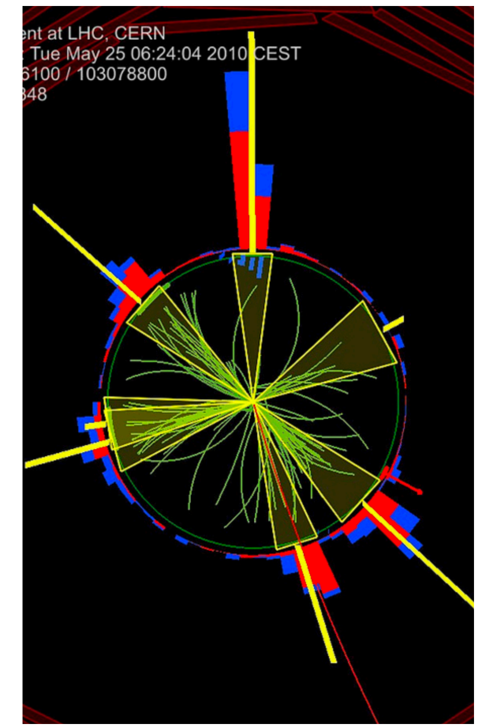

Reconstruction(s) of the event under limited latency.
Better resolution help lowering background trigger rates.
Approximate deep learning surrogates can help.

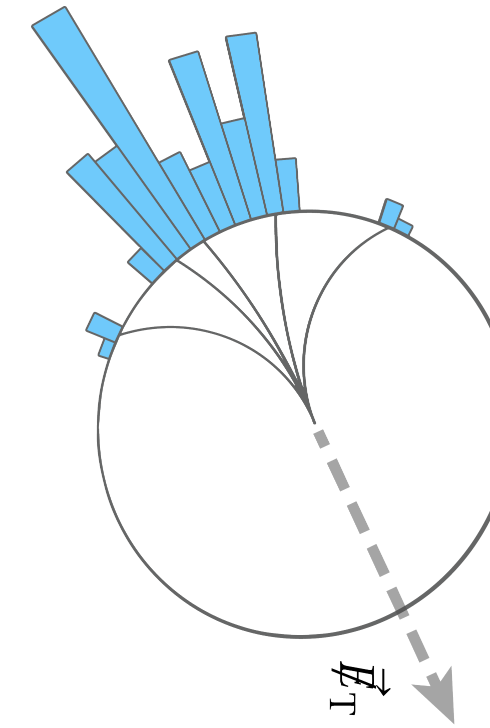
# Reconstructing Collisions

| Detector Data | Local reconstruction | Particle representation | Jet Clustering | High level features |
|---|---|---|---|---|




Event Processing

Dimensionality reduction

Globalization of information

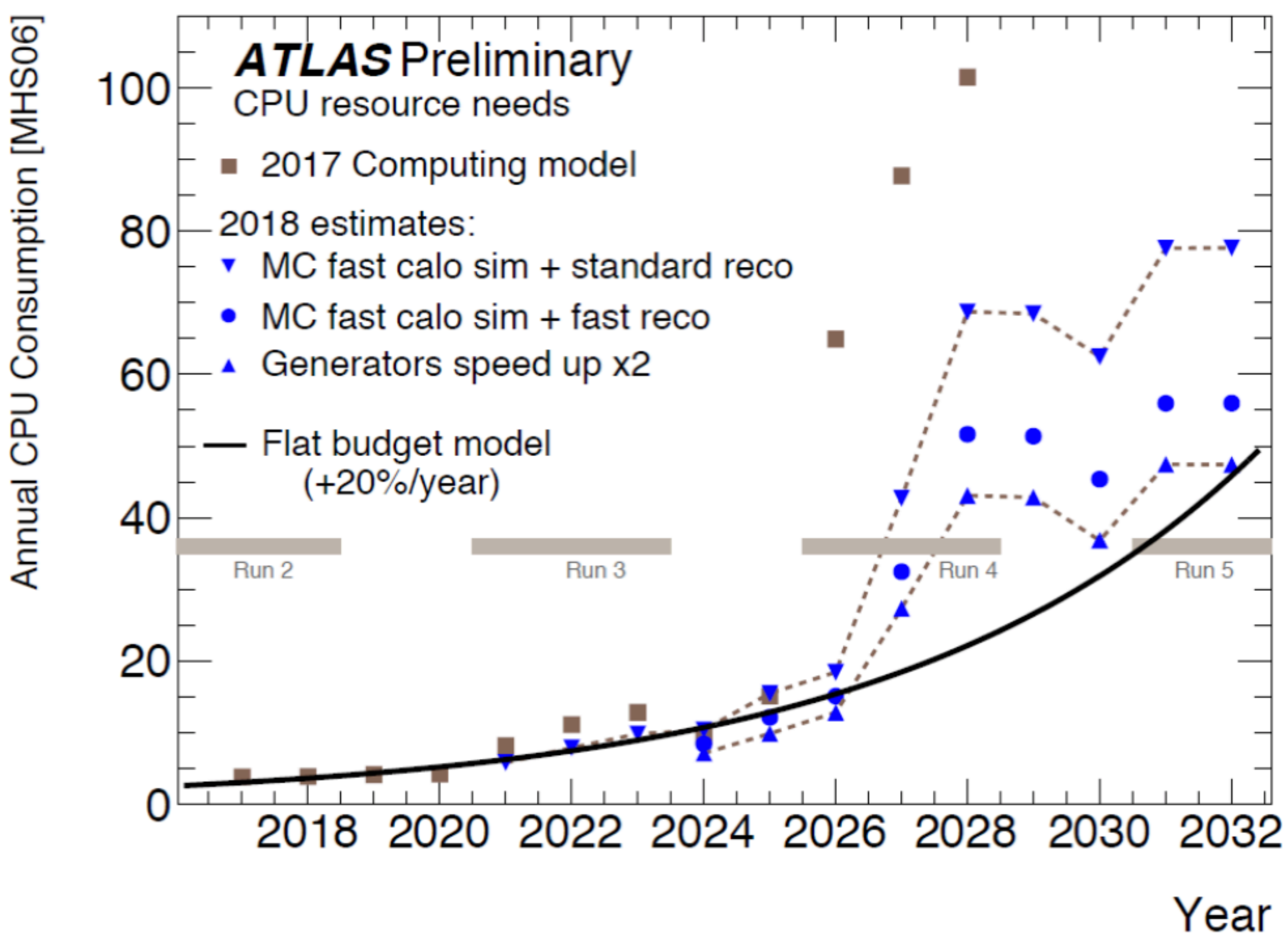
From digital signal, to local hits, to a sequence of objects, and high-level features.
Complex and computing intensive tasks.

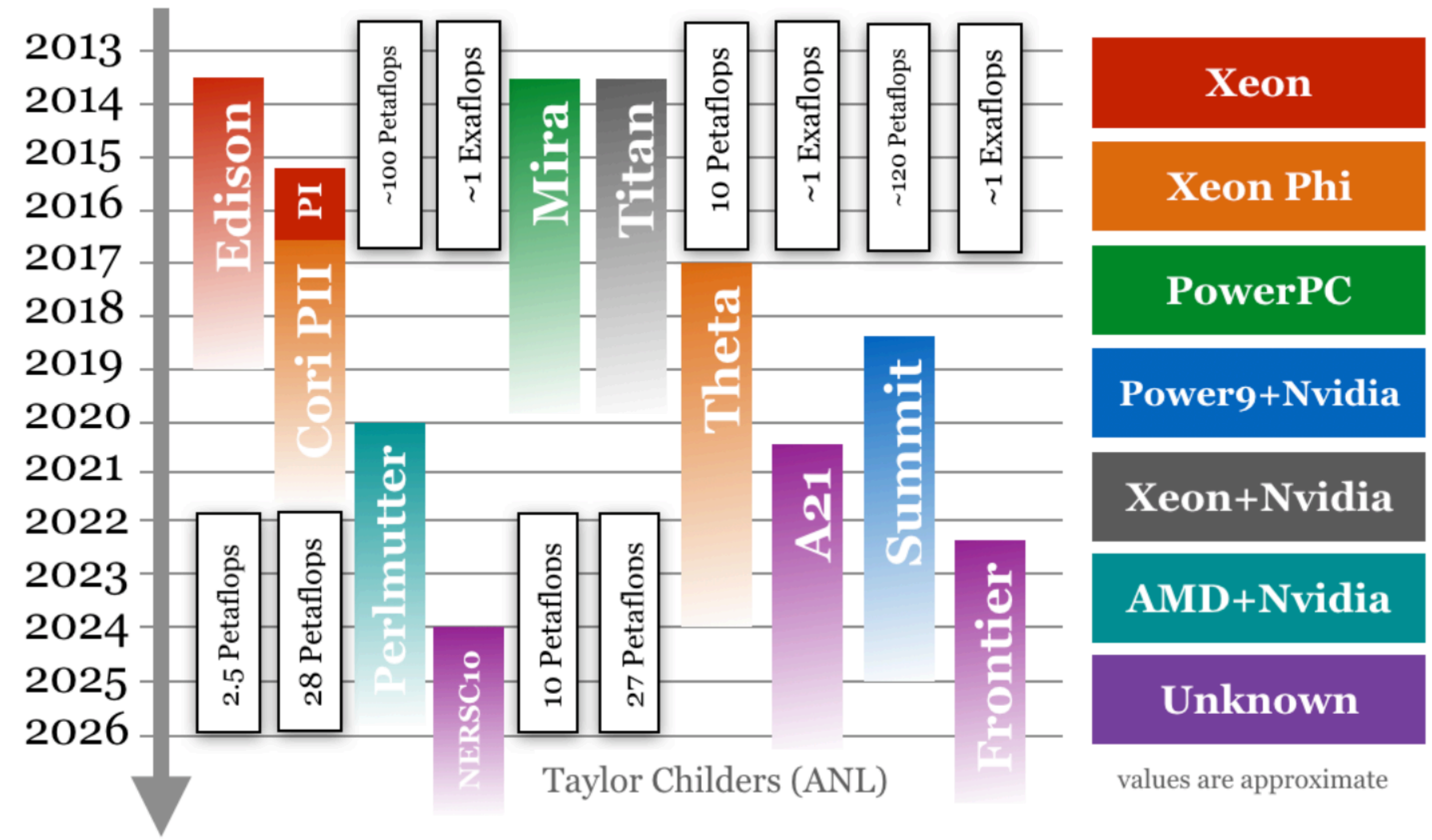
# Simulating Collisions

| | |
|---|---|
| *Madgraph, Pythia, Sherpa, ...* | **Event Generator**: compute predictions of the standard models to several orders of expansion in coupling constants (LO, NLO, NNLO, ...) using proton density functions. |
| *Pythia, ...* | **Hadronization**: phenomenological model of the evolution of hadrons under the effect of QCD. |
| *GEANT 4, GEANT V* | **Material simulator**: transports all particles throughout meters of detector, using high resolution geometrical description of the materials. |
| *Homegrown software* | **Electronic emulator:** converts simulated energy deposits in sensitive material, into the expected electronic signal, including noise from the detector. |

Non-differentiable, **computing intensive** sequence of **complex simulators** of the signal expected from the detectors.

# The Computing Cost of Science




https://indico.cern.ch/event/822126/contributions/3500169/

Ever growing needs for computing resource
Slowdown of classical architecture
Growth of GPU architecture

*Take home message :*

*Measure rare and exotic processes from orders of magnitude larger backgrounds.*

*The Standard Model predicts with precision what to expect from many processes.*

*Reconstruct, identify and reject large amount of event within resource constraints.*

# A Glimpse at the Machine Learning Landscape

# What Is Machine Learning

*“Giving computers the ability to learn without explicitly programming them”* A. Samuel (1959).

Is fitting a straight line machine learning ?

Models that have enough capacity to define its own internal representation of the data to accomplish a task : **learning from data.**

**In practice :** a statistical method that can extract information from the data, not obviously apparent to an observer.

→ Most approach will involve a **mathematical model** and a cost/reward function that needs to be **optimized.**

→ The more **domain knowledge** is incorporated, the better.

# Overview

**Reinforcement Learning (cherry)**
- The machine predicts a scalar reward given once in a while.
- **A few bits for some samples**

**Supervised Learning (icing)**
- The machine predicts a category or a few numbers for each input
- **10→10,000 bits per sample**

**Unsupervised Learning (cake)**
- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- **Millions of bits per sample**

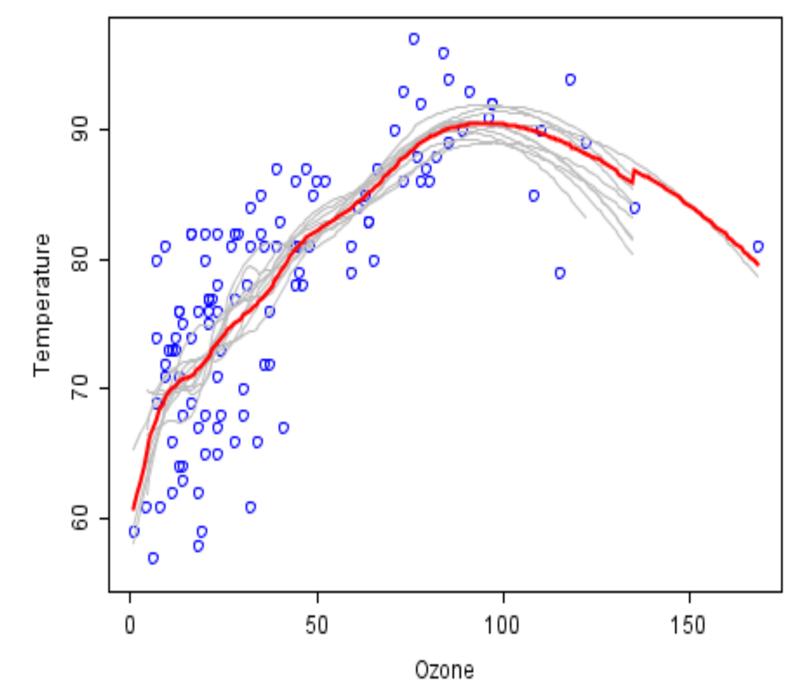
Yann Le cun, CERN, 2016

# Supervised Learning

- Given a dataset of samples, a subset of features is qualified as **target**, and the rest as **input**
- Find a **mapping from input to target**
- The mapping should **generalize to any extension** of the given dataset, provided it is generated from the same mechanism

$$dataset \equiv \{(x_i, y_i)\}_i$$

$$find \ \ function \ f \ \ s.t. \ \ f(x_i) = y_i$$

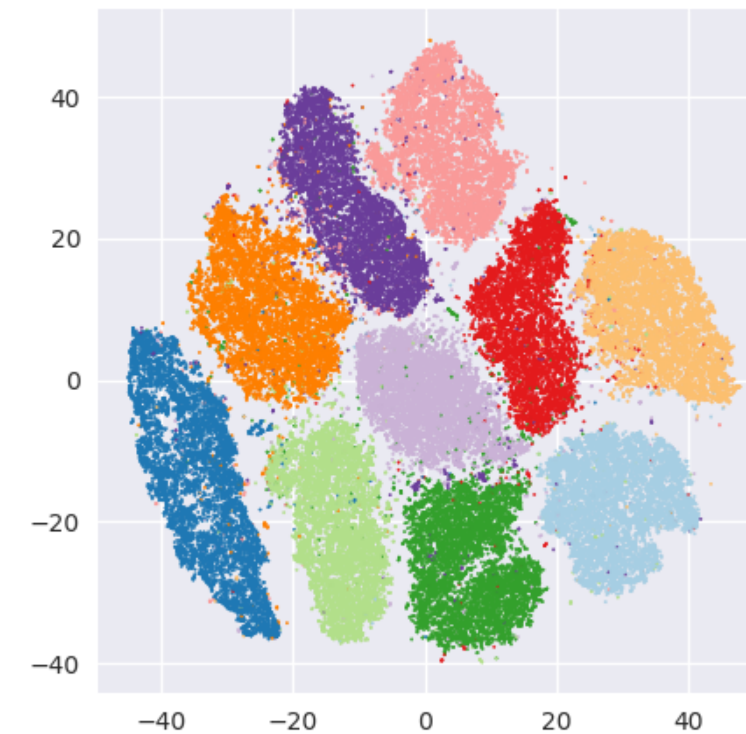



- Finite set of target values :
  → **Classification**
- Target is a continuous variable :
  → **Regression**

# Unsupervised Learning

- Given a dataset of samples, but there is no subset of feature that one would like to predict
- Find mapping of the samples to a lower dimension manifold
- The mapping should generalize to any extension of the given dataset, provided it is generated from the same mechanism

$$dataset \equiv \{(x_i)\}_i$$
$$find\ \ f\ \ s.t.\ \ f(x_i) = p_i$$



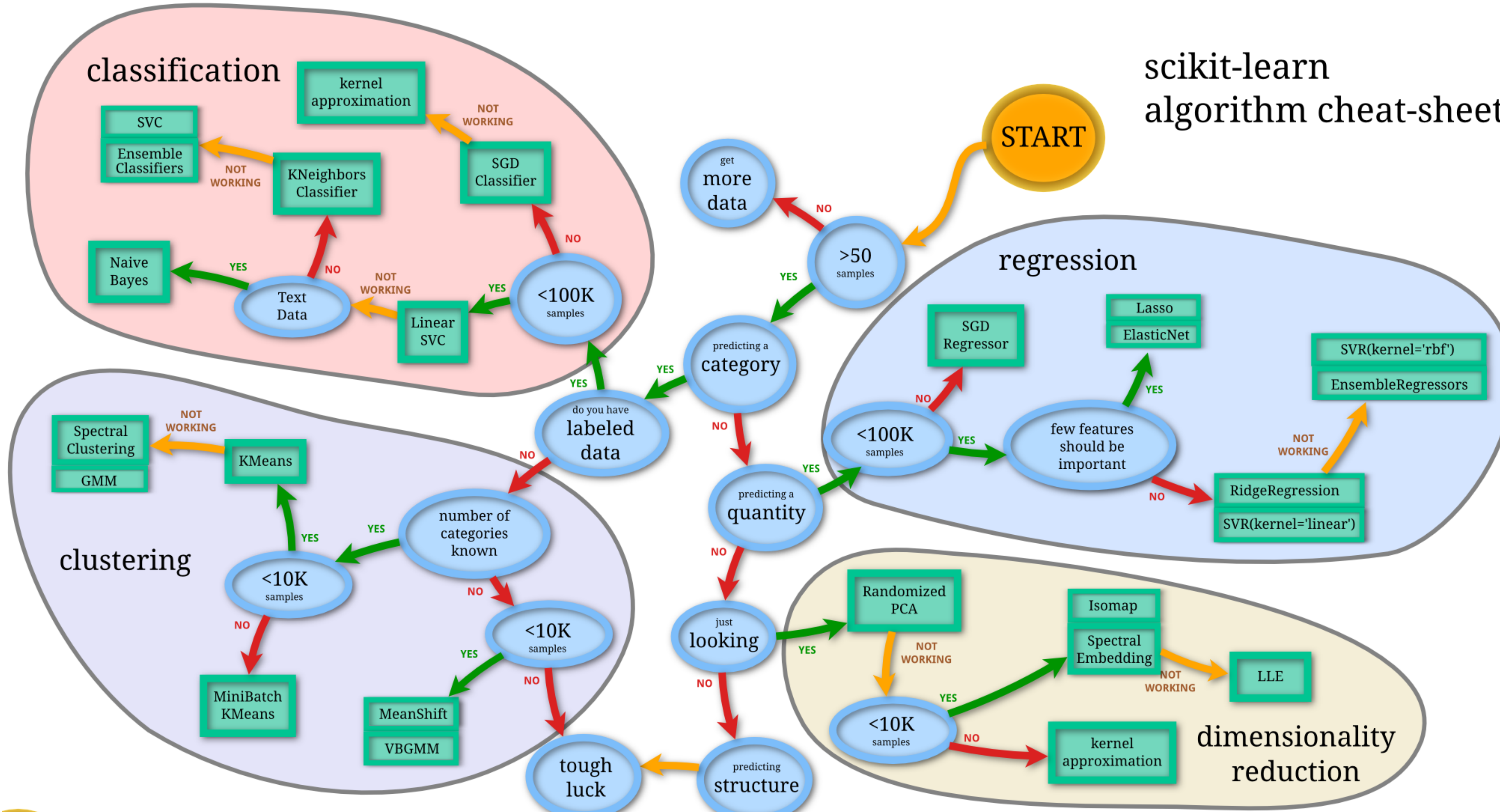

- Manifold is a finite set
  - → **Clusterization**
- Manifold is a lower dimension manifold :
  - → **Dimensionality reduction, density estimator**

# Reinforcement Learning

- Given an **environment** with multiple states, given a reward upon action being taken over a state
- Find an **action policy to drive** the environment toward maximum cumulative reward

$$s_{t+1} = Env(s_t, a_t)$$
$$r_t = Rew(s_t, a_t)$$
$$\pi(a|s) = P(A_t = a | S_t = s)$$
$$find\ \pi\ s.t.\ \sum_t r_t\ is\ maximum$$

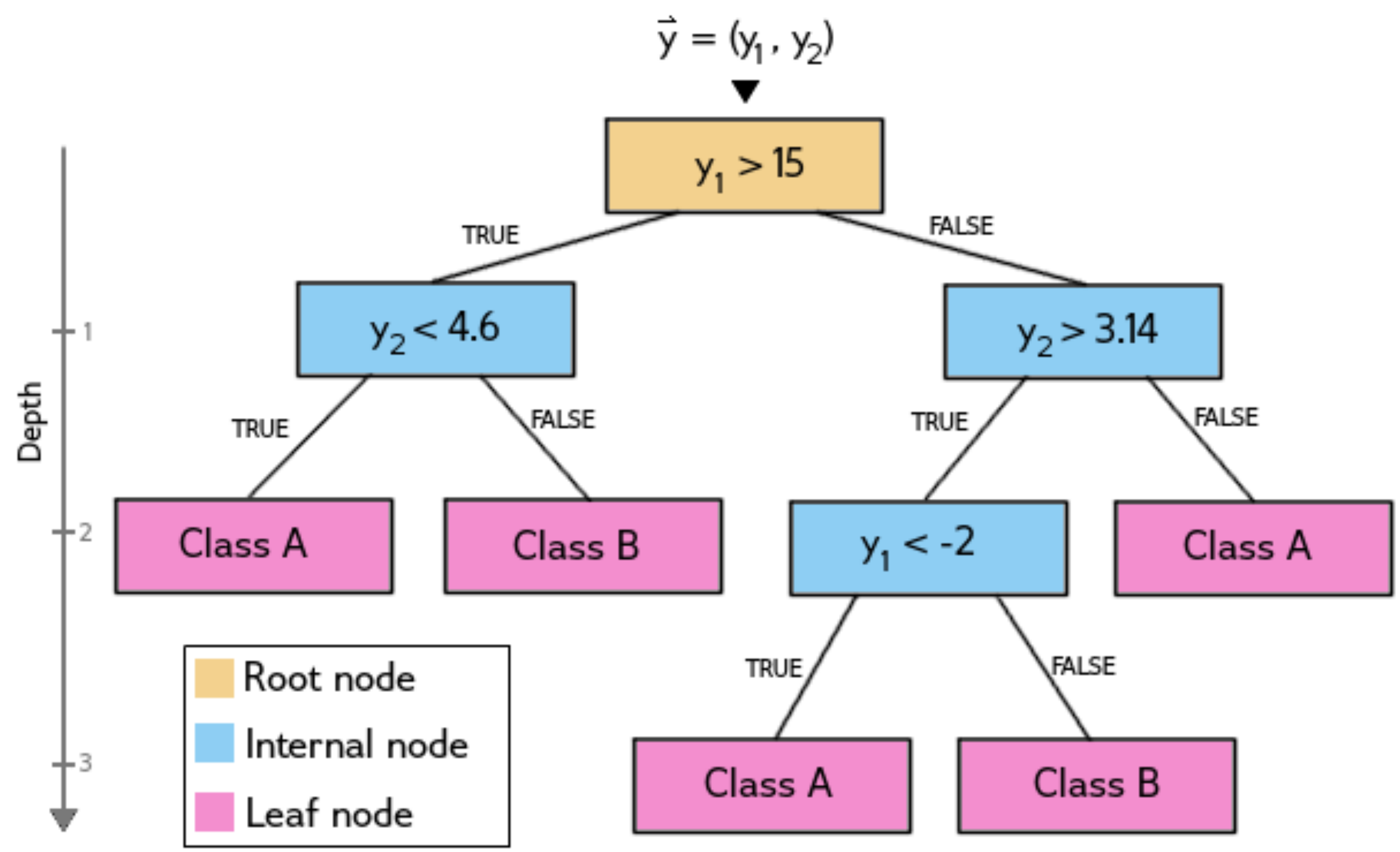
# (Some) Machine Learning Methods




http://scikit-learn.org/stable/tutorial/index.html

# Decision Tree

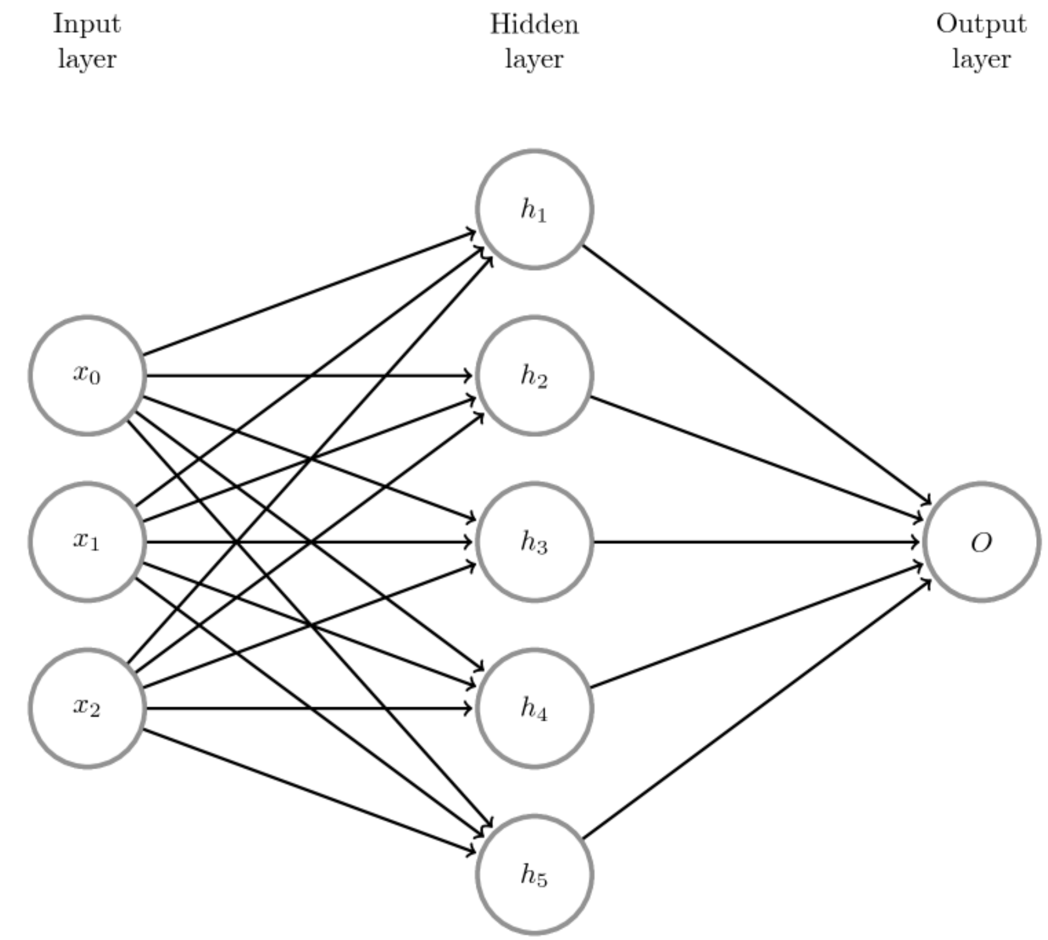
- Decision trees is a well known tool in supervised learning.
- It has the advantage of being easily interpretable
- Can be used for classification or regression



# Artificial Neural Network

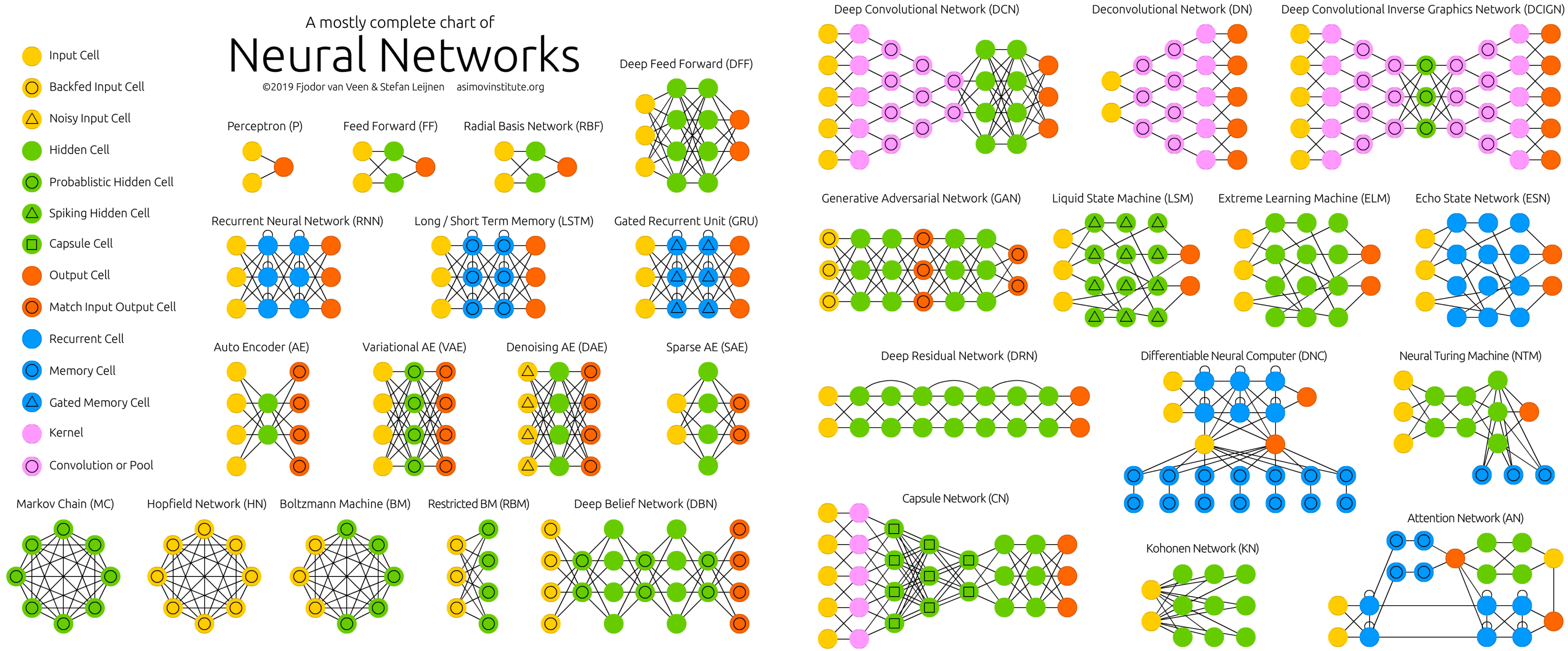
- **Biology inspired** analytical model, but **not bio-mimetic**
- Booming in recent decade thanks to large dataset, increased computational power and theoretical novelties
- Origin tied to logistic regression with change of data representation
- Part of any “deep learning” model nowadays
- Usually large number of parameters trained with stochastic gradient descent

$$h = \phi(Ux + v)$$
$$o(x) = \omega^T h + b$$
$$p_i \equiv p(y = 1 | x) \equiv \sigma(o(x)) = \frac{1}{1 + e^{-o(x)}}$$
$$loss_{XE} = -\sum_i y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i)$$



# Neural Net Architectures

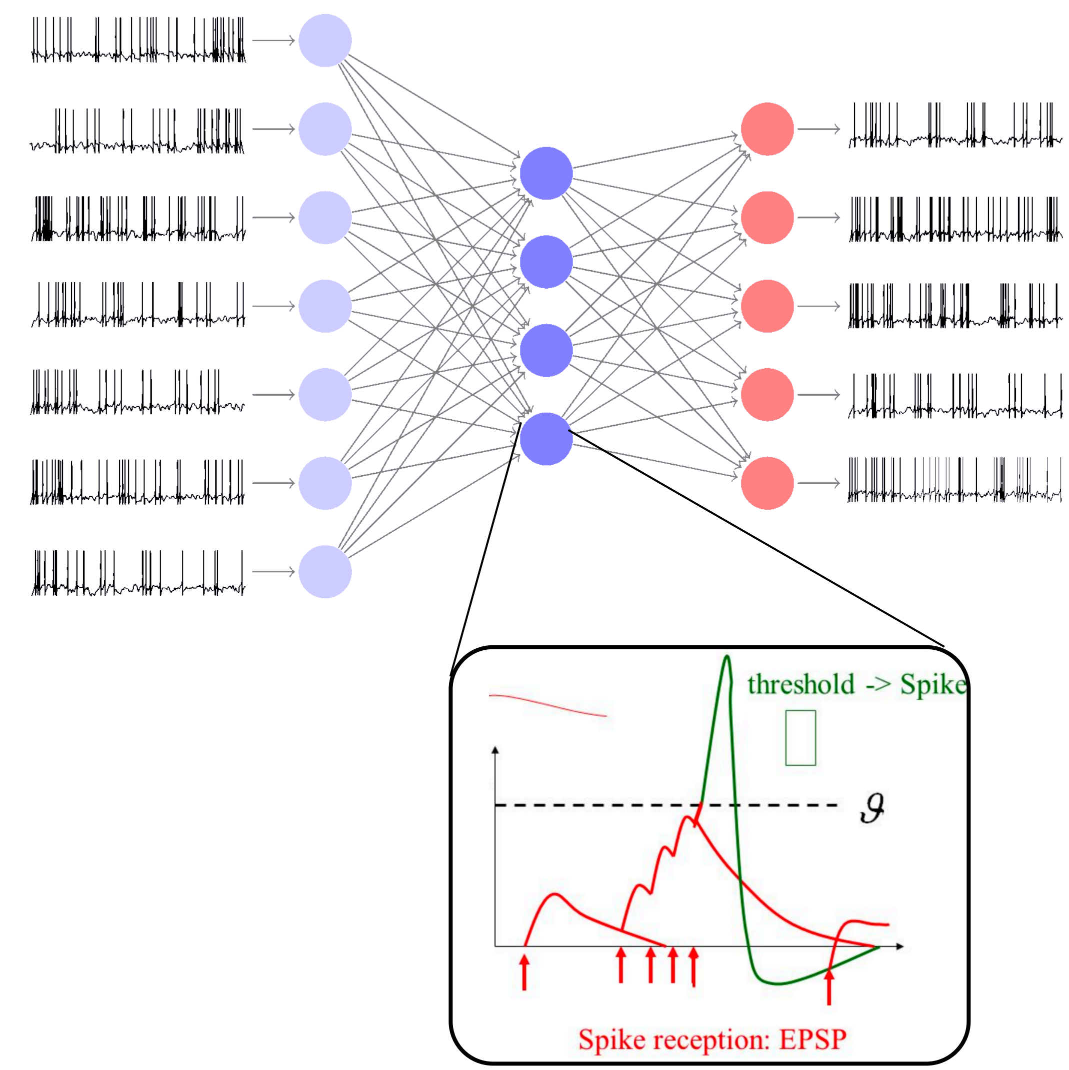
http://www.asimovinstitute.org/neural-network-zoo




➢ Does not cover it all : densenet, graph network, ...

# Spiking Neural Network

- Closer to the actual biological brain
- Adapted to temporal data
- Hardware implementation with low power consumption
- Trained using evolutionary algorithms, recent work on gradient-based methods [1706.02609] , [1901.09948], [2110.14092]
- Economical models
- Python libraries for spiking neural network : slayer, snntorch, spikingjelly, norse, …



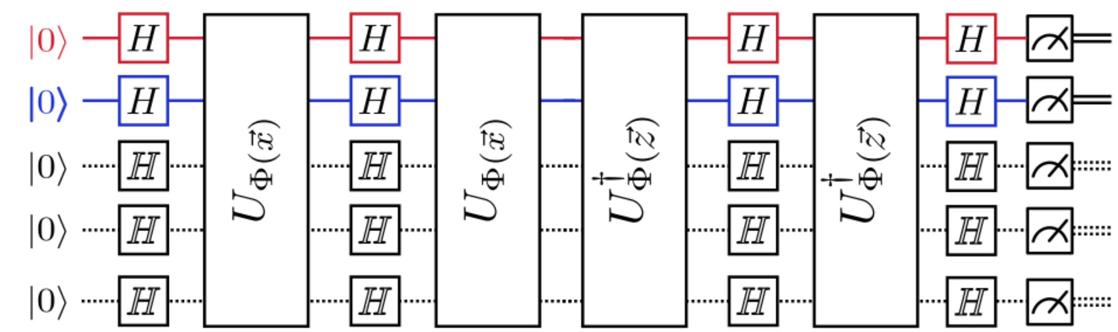

| | Deep Learning | Spiking |
|---|---|---|
| **Training Method** | Back-propagation | Not well established (here, genetic algorithms) |
| **Native Input Types** | Images/Arrays of values | Spikes |
| **Network Size** | Large (many layers, many neurons and synapses per layer) | Relatively small (fewer neurons and sparser synaptic connections) |
| **Processing Abilities** | Good for spatial | Good for temporal |
| **Performance** | Well understood and state-of-the-art | Not well understood |

# Quantum Machine Learning

Deep learning is computing intensive, and de-facto enabled by use of GPU. People are looking for ways to leverage possible quantum advantage to accelerate machine learning techniques.
Main algorithms used in recent studies

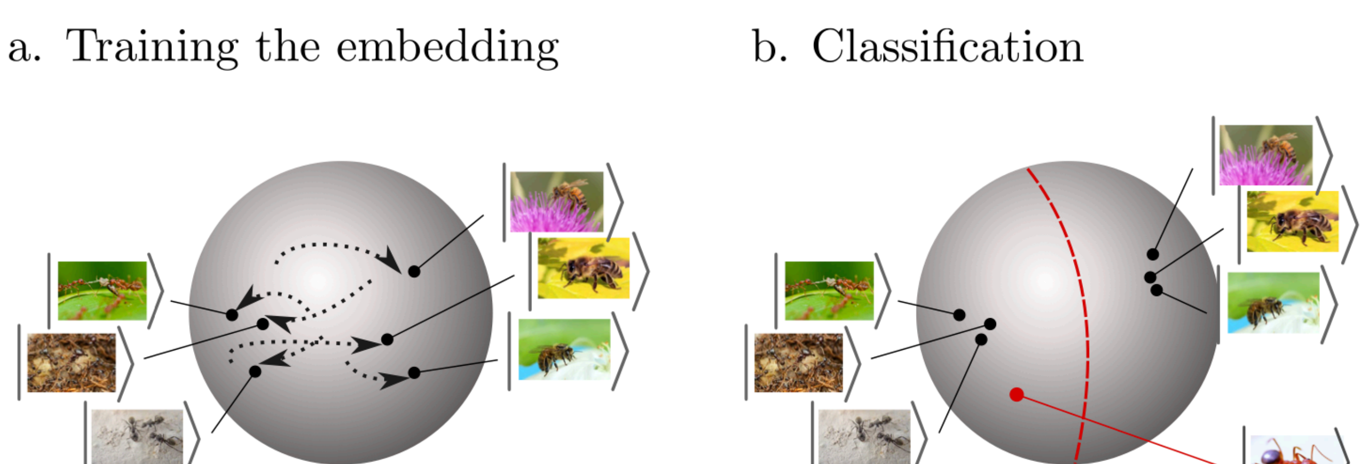
- ➡Variational Quantum Circuits (VQC)
- ➡Quantum Support Vector Machine (QSVM)
- ➡Quantum Restricted Bolztman Machine (QRBM)
- ➡Quantum Adiabatic Machine Learning (QAML)
- ➡Quantum Generative Adversarial Network (QGAN)
- ➡…

Field in constant evolution. Embedding is crucial.
Deep implications of kernel methods.

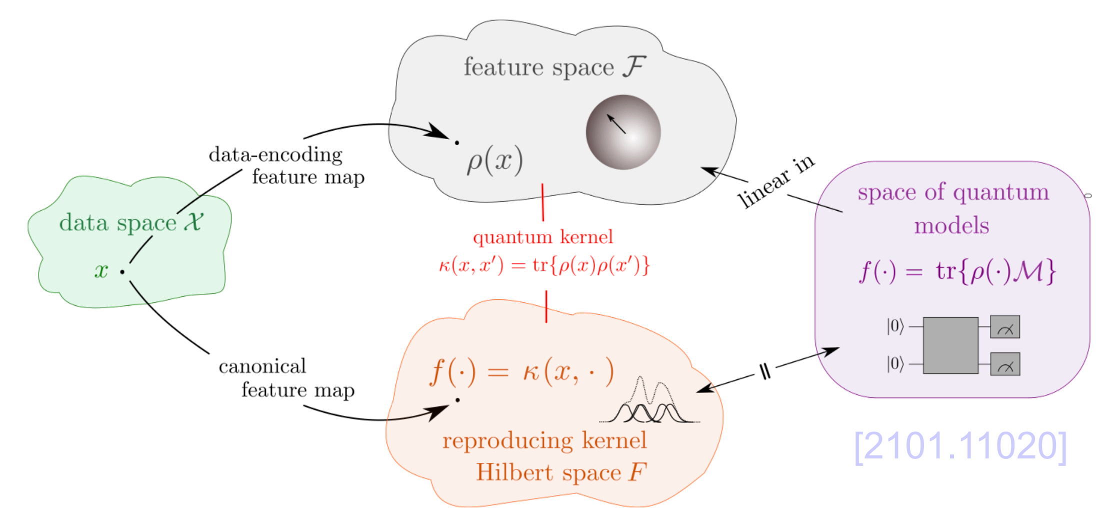



[1804.11326]

Software and toolkit available pennylane , tf-quantum

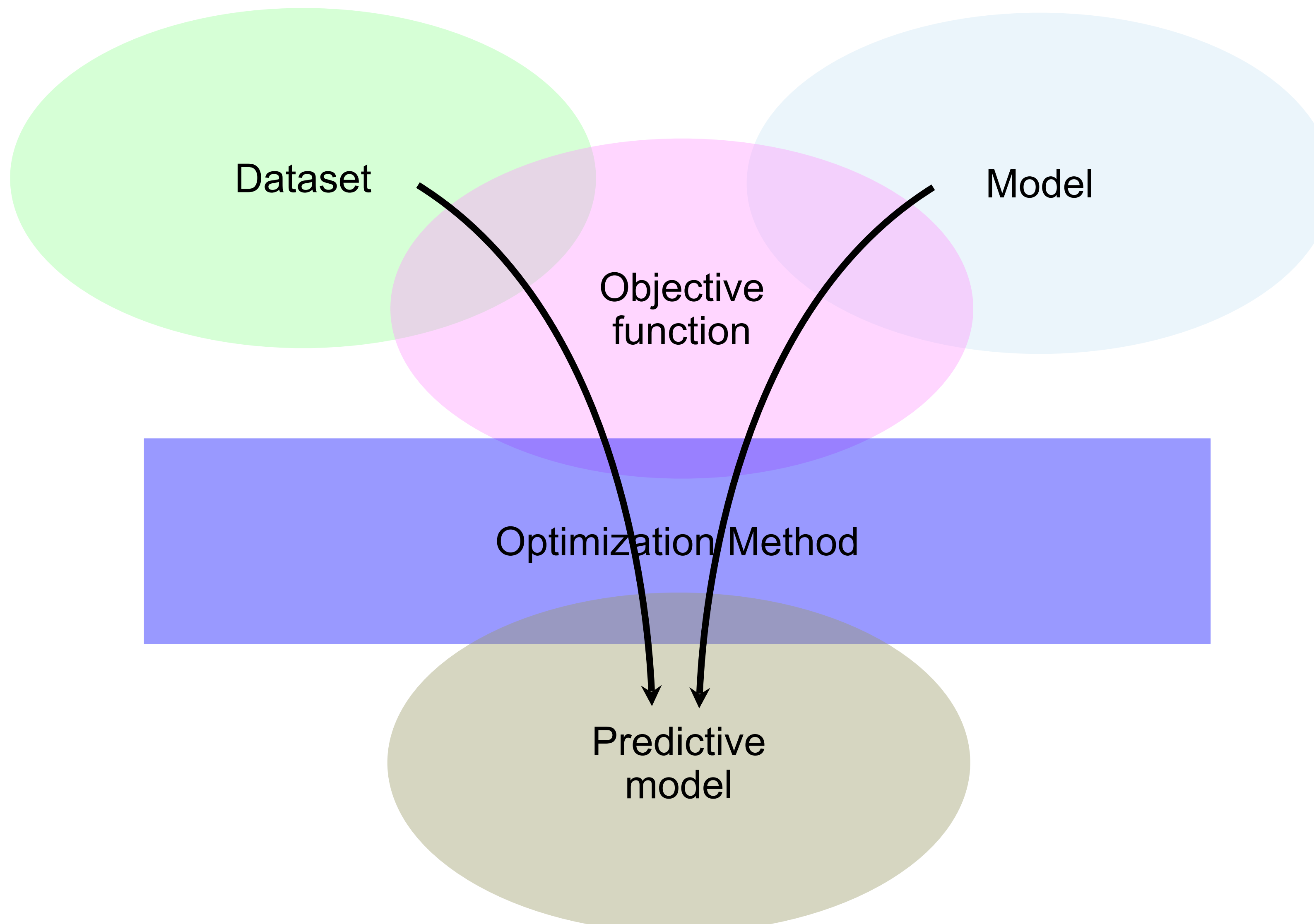



[2001.03622]



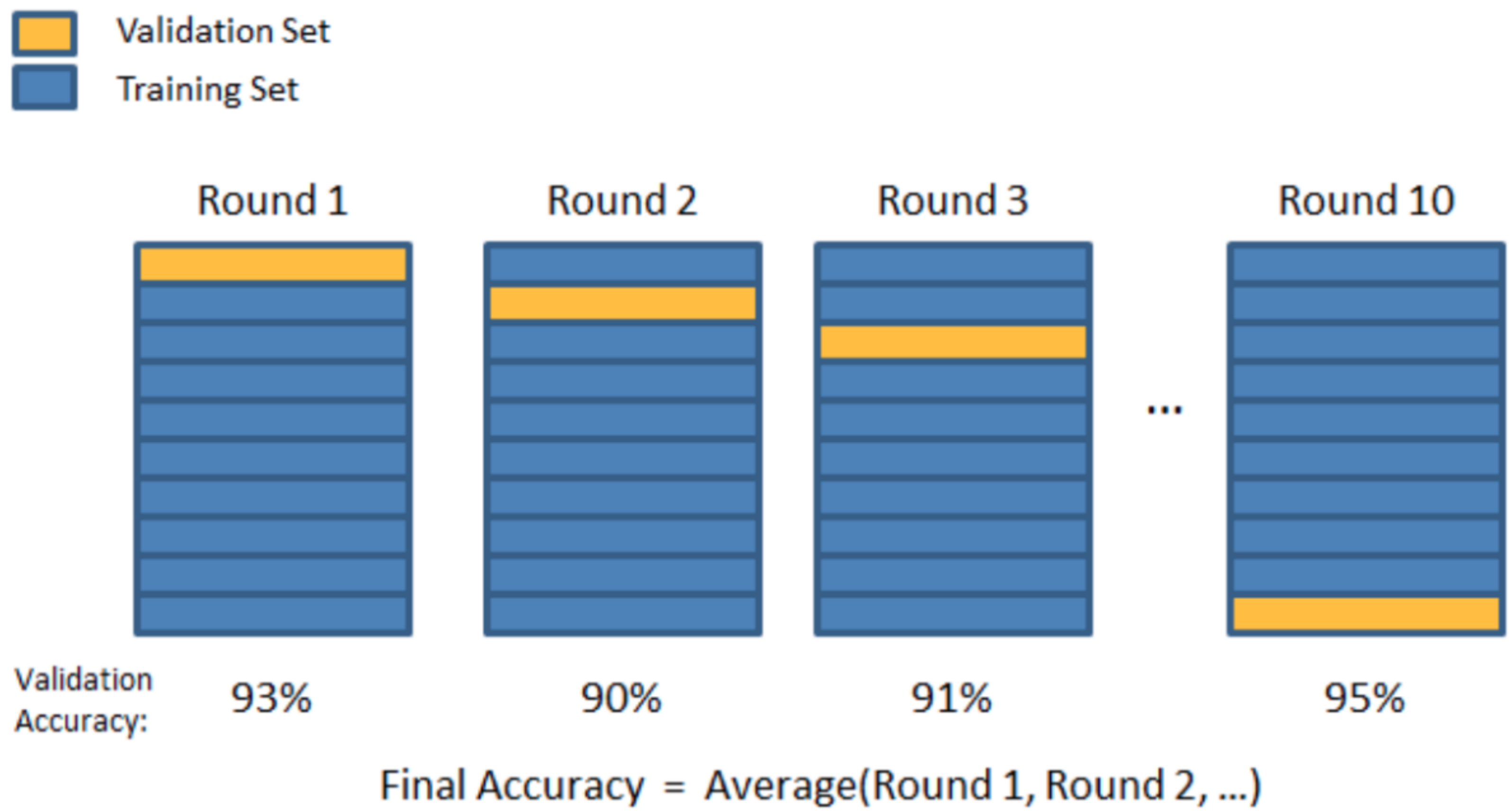

[2101.11020]

# Machine Learning Concept




All comes down to an optimization problem.
What follows are some of the things to keep an eye on when developing a machine learning solution

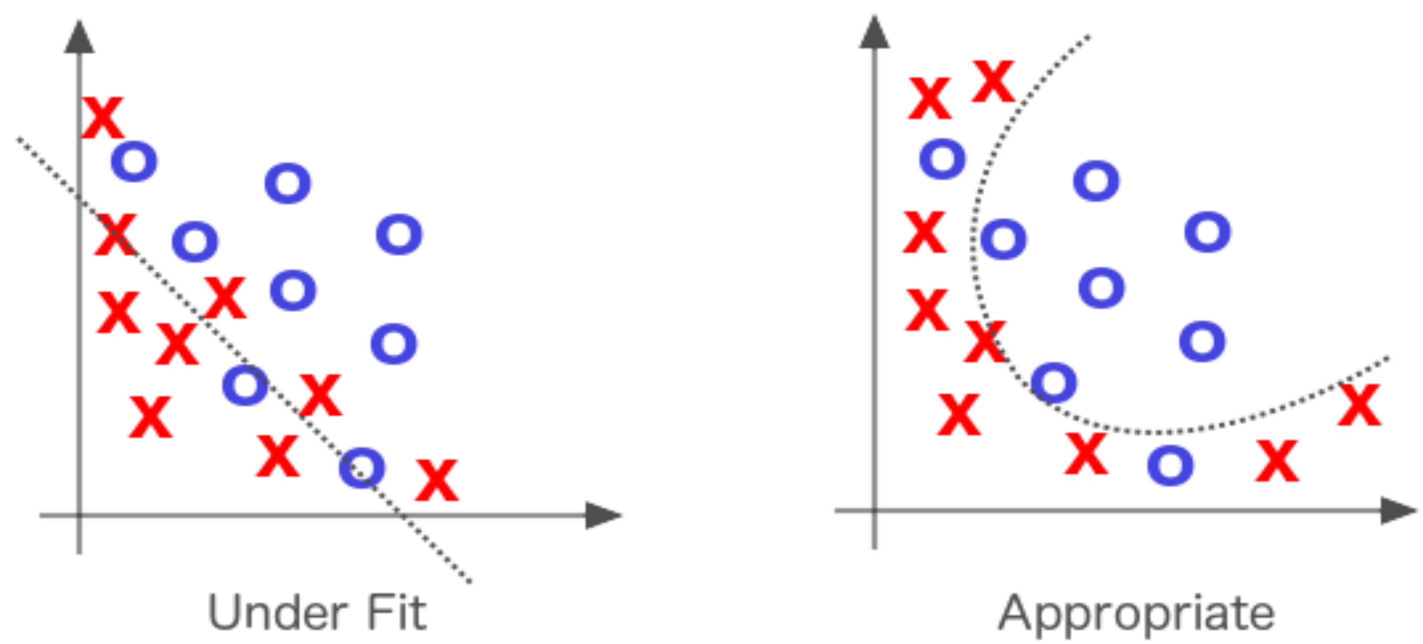
# Cross Validation




- Model selection requires to have an estimate of the uncertainty on the metric used for comparison
  - ➢ K-folding provides an un-biased way of comparing models
- Stratified splitting (conserving category fractions) protects from large variance coming from biased training
- Leave-one-out cross validation : number folds ≡ sample size

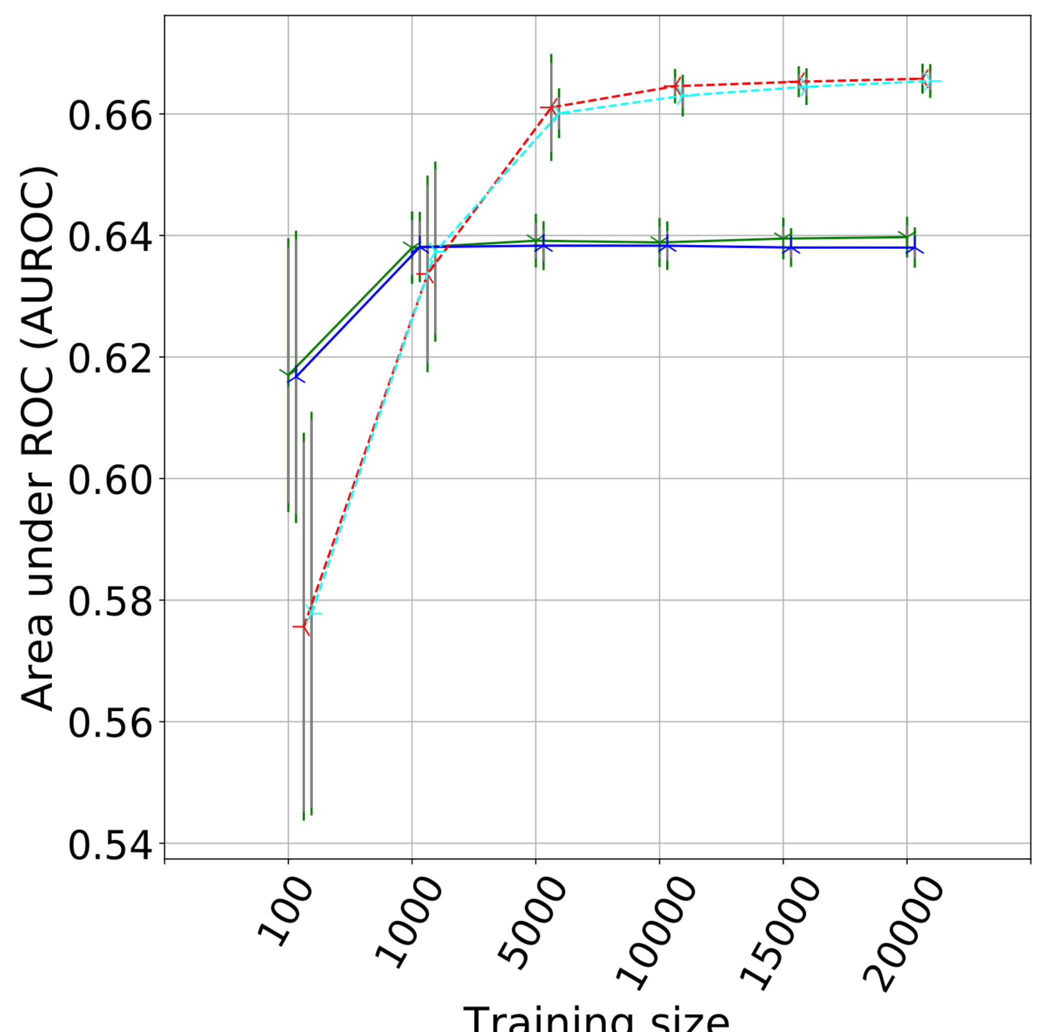
# Under-fitting

- Poor model performance can be explained
  - Lack of modeling capacity (not enough parameters, inappropriate parametrization, …)
  - Model parameters have not reached optimal values



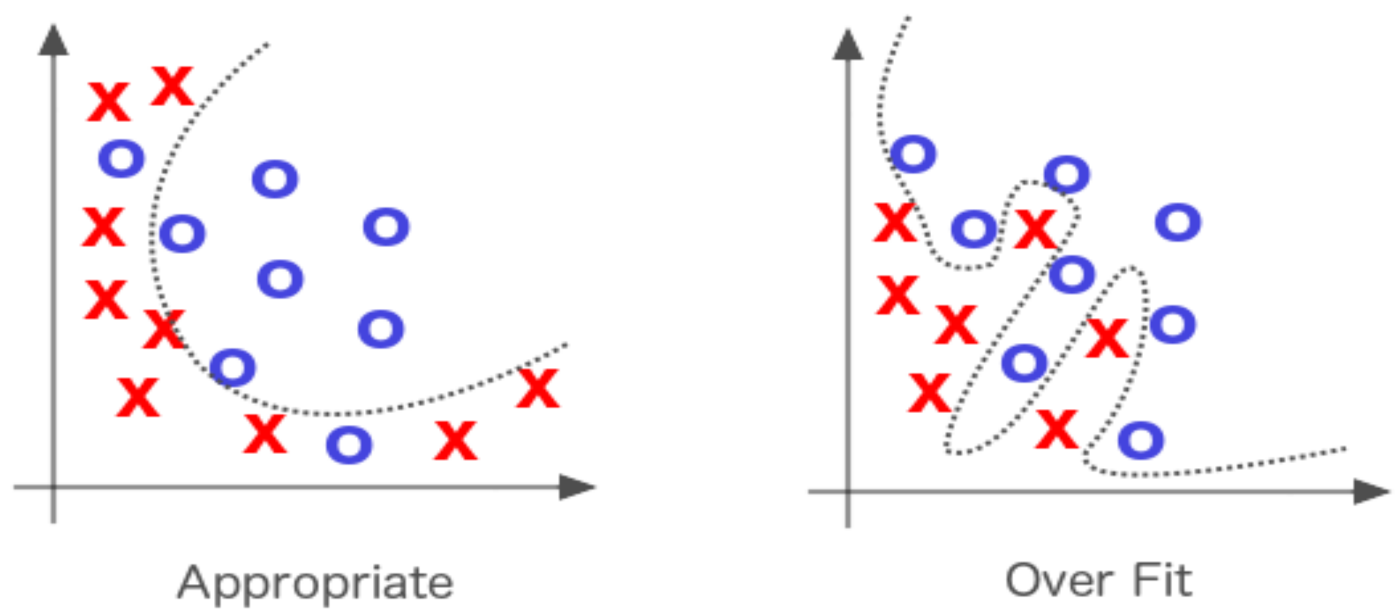
# Need for Data

- “What is the **best performance one can get** ?” rarely has an answer
- When comparing multiple models, one can answer “what is the **best of these models, for this given dataset** ?”
- It does not answer “what is the **best model at this task** ?”



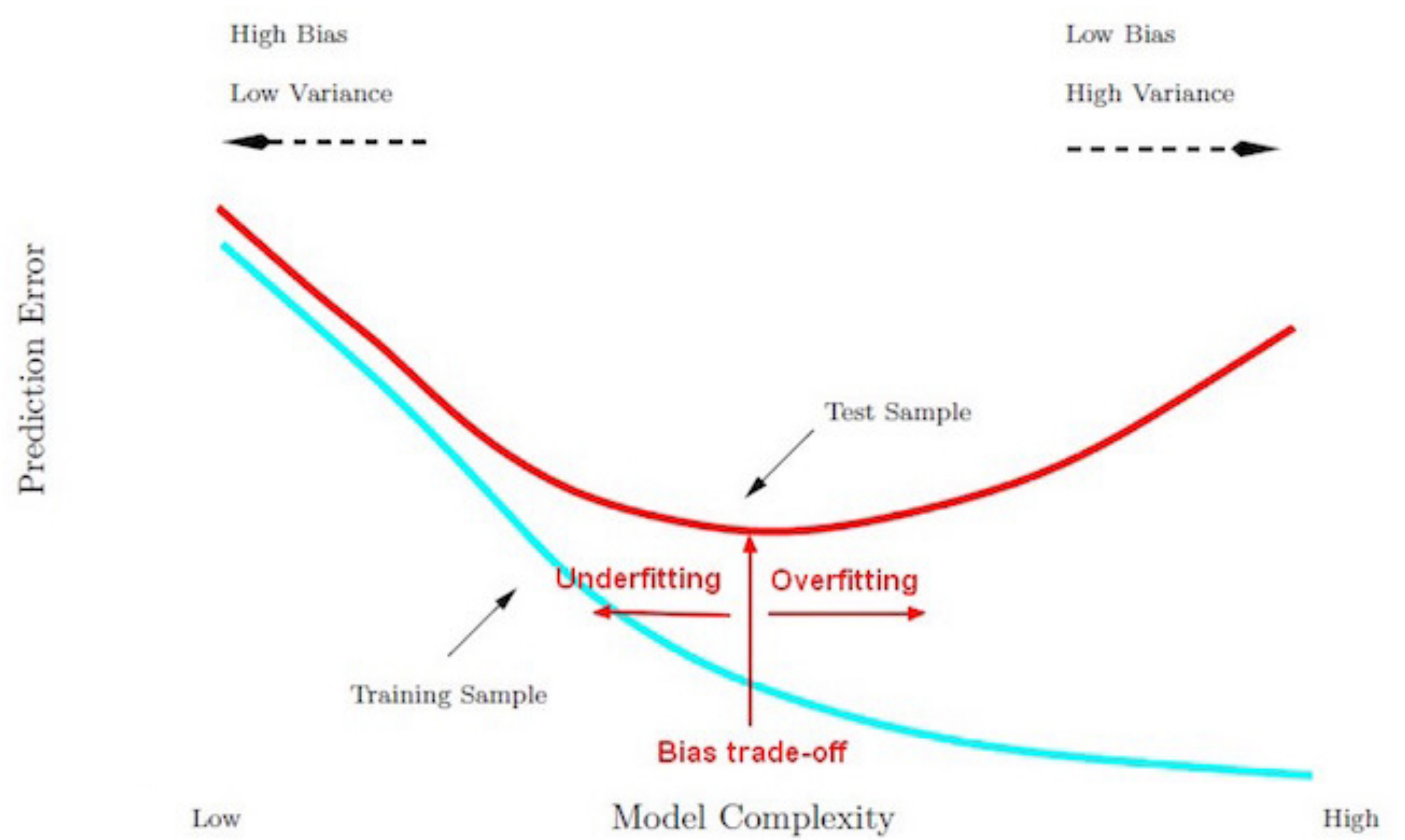
# Over-fitting

- “Too good to be true” model performance can be explained
  - ➢ Excessive modeling capacity (too many parameters, parametrization is too flexible, ...)
  - ➢ Model parameters have learn the trained data by heart
- Characterized by very good performance on the training set and (much) lower performance on unseen dataset



# Generalization

- Systematic error ≡ bias
- Sensitivity of prediction ≡ variance
- A good model is a tradeof of both
  - ➢ Early stopping can help with halting the model



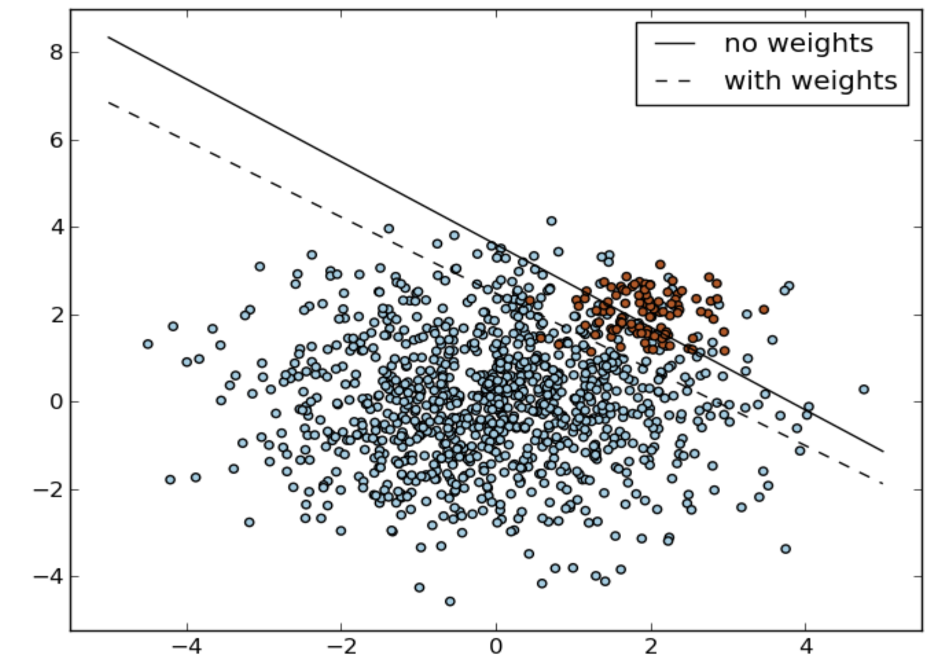
# Figure(s) of Merit(s)

- Objective function in optimization might be chosen for computational reason (differentiable, …)
- **Objective function might only be a proxy** to the actual figure of merit of the problem at hand
- Multi-objective optimization is subject to trade-off between objectives

➢ While model optimization is based on the loss function over the training set, following the evolution of a more interesting (non-usable) metric over the validation can help selecting models that are better for the use case

# Class Imbalance

- In many cases the number of samples varies significantly from class to class
- Class imbalance biases the performance on the minority class
- Multiple ways to tackle the issue
  - ➢ Over-sample the minority class
  - ➢ Synthetic minority over-sampling
  - ➢ Under-sample the majority class
  - ➢ Weighted loss function
  - ➢ Active learning




- NB: metrics can be sensitive to class imbalance and be misguiding if not correct : e.g. 99% accuracy with 0% recall

# Training

- Training phase or learning phase is when the parameters of the model are adjusted to best solve the problem
- For some model/technique (especially deep learning) this **can become computationally prohibitive**
- General purpose graphical processing units (GP-GPU) offer an enormous amount of parallel compute power, applicable to specific numerical problems
- Matrix calculation, minibatch computation, deep learning, … can get a significant boost from GP-GPU.
- Further parallelization can be obtained across multiple nodes/GPU using : most of the deep learning framework offer **distributed training solutions** : tf.distribute, torch.nn.DataParallel, NNLO, …

# Hyper-parameter Optimization

- Most optimization methods and models require hyperparameters
    - number of layers/nodes in an ANN, number of leaves in a decision tree, learning rates, …
- In most cases these parameters cannot be optimized while the model is trained ; i.e **not optimized with gradient descent**
- Their values can however significantly influence the final performance

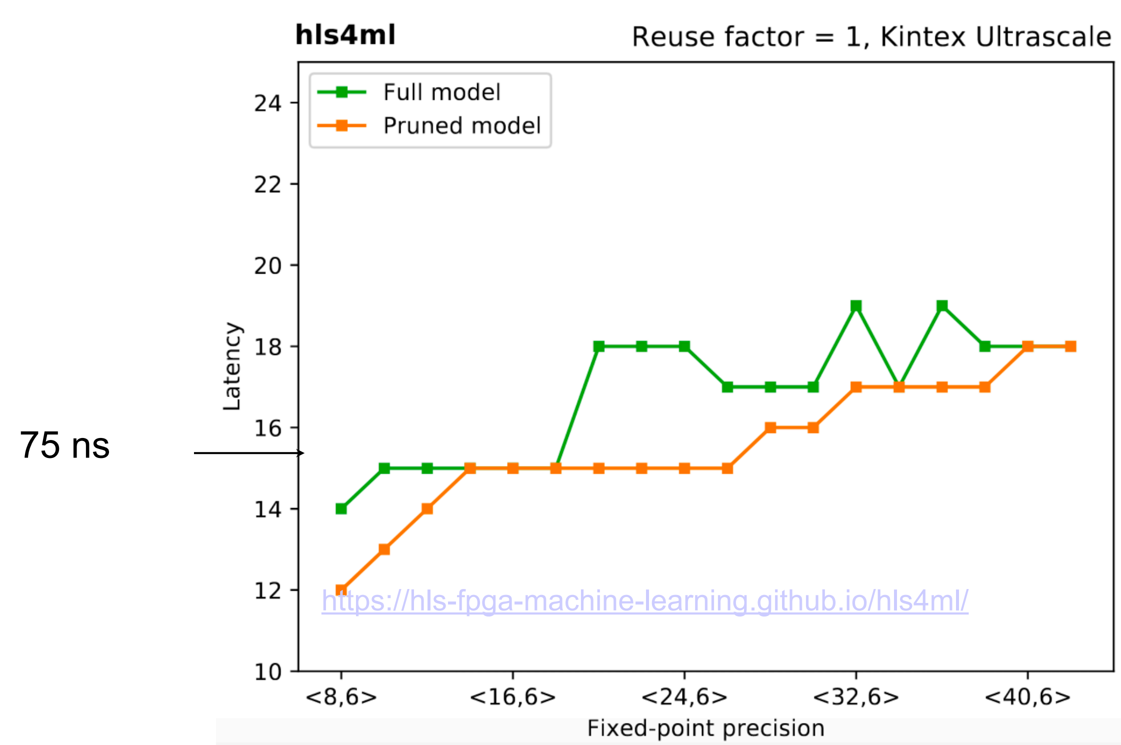
➢ These can be optimize in various ways
    - Simple grid search
    - Bayesian optimization
    - Evolutionary algorithm

➢ Model comparison should be done very carefully
    - K-folding is a “must”

➢ Multiple libraries available skOpt, hyperopt, GPyOpt, ray.tune, Spearmint, deap, …

# Cost of Running the Model

- Contrary to training, making prediction from a trained model is usually rather fast, even on CPU
- However fast is may be, it might still not be fast enough for the particular application
- Faster inference can be obtained on specialized hardware GP-GPU, TPU, FPGA, neuromorphic, … when the application allows it (trigger, onboard electronics, …)
- “Inference as a service” can be a solution to get access to accelerators remotely, at the cost of communication



# Take Home Message

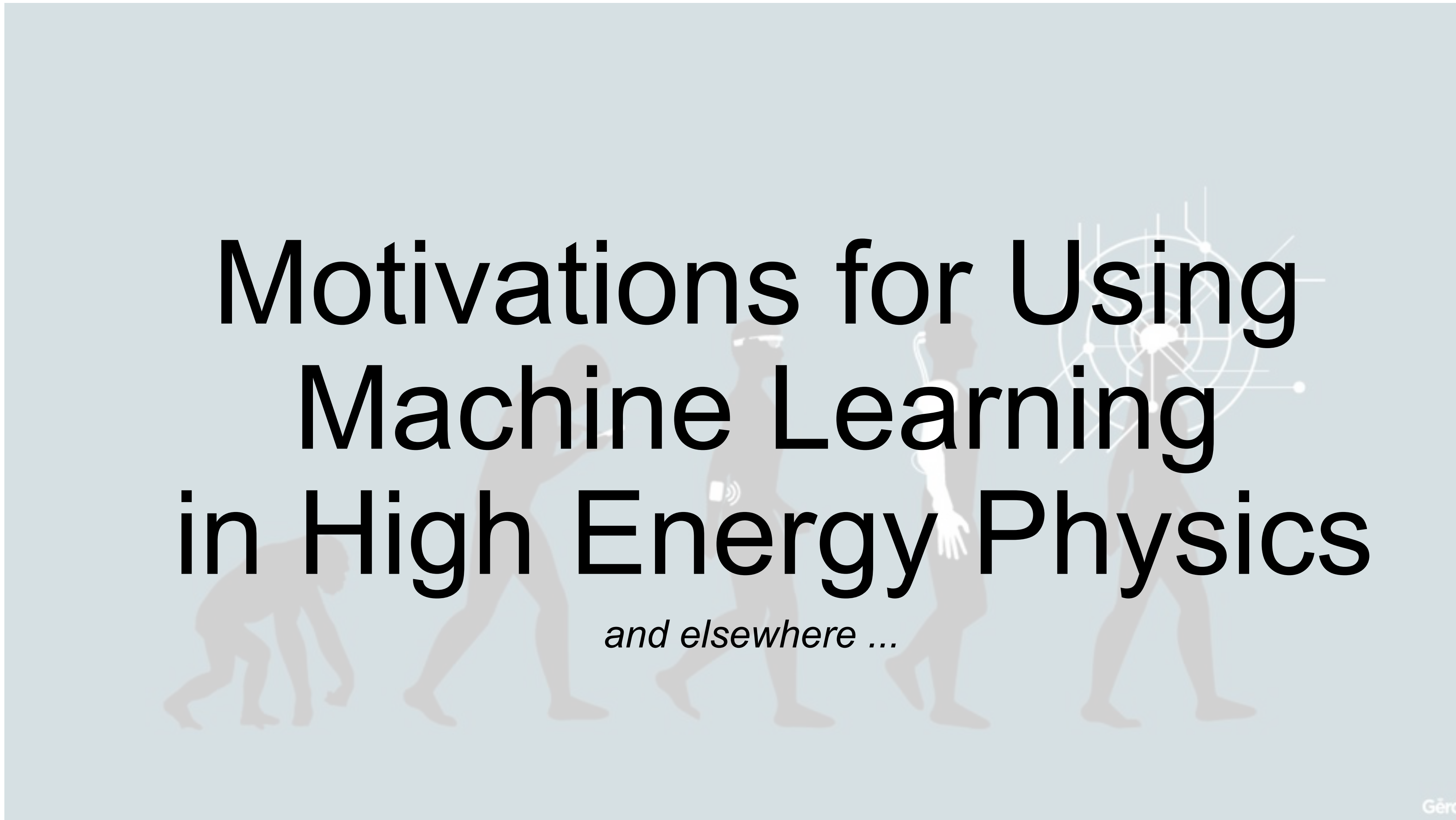
*Machine learning applications need to be developed with scientific rigor.*

*Lots of interesting studies possible on statistics/theory of learning.*

*Keep an eye on cost of making prediction.*

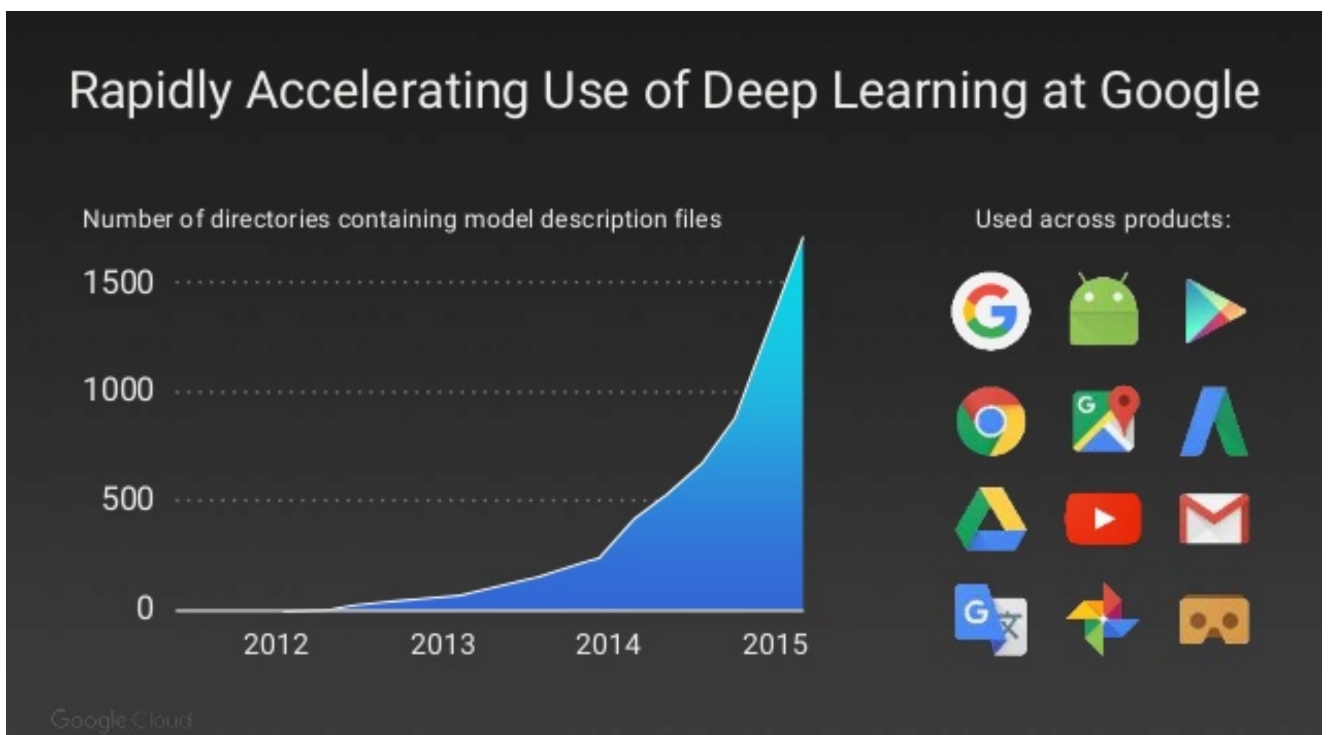
# Motivations for Using Machine Learning in High Energy Physics

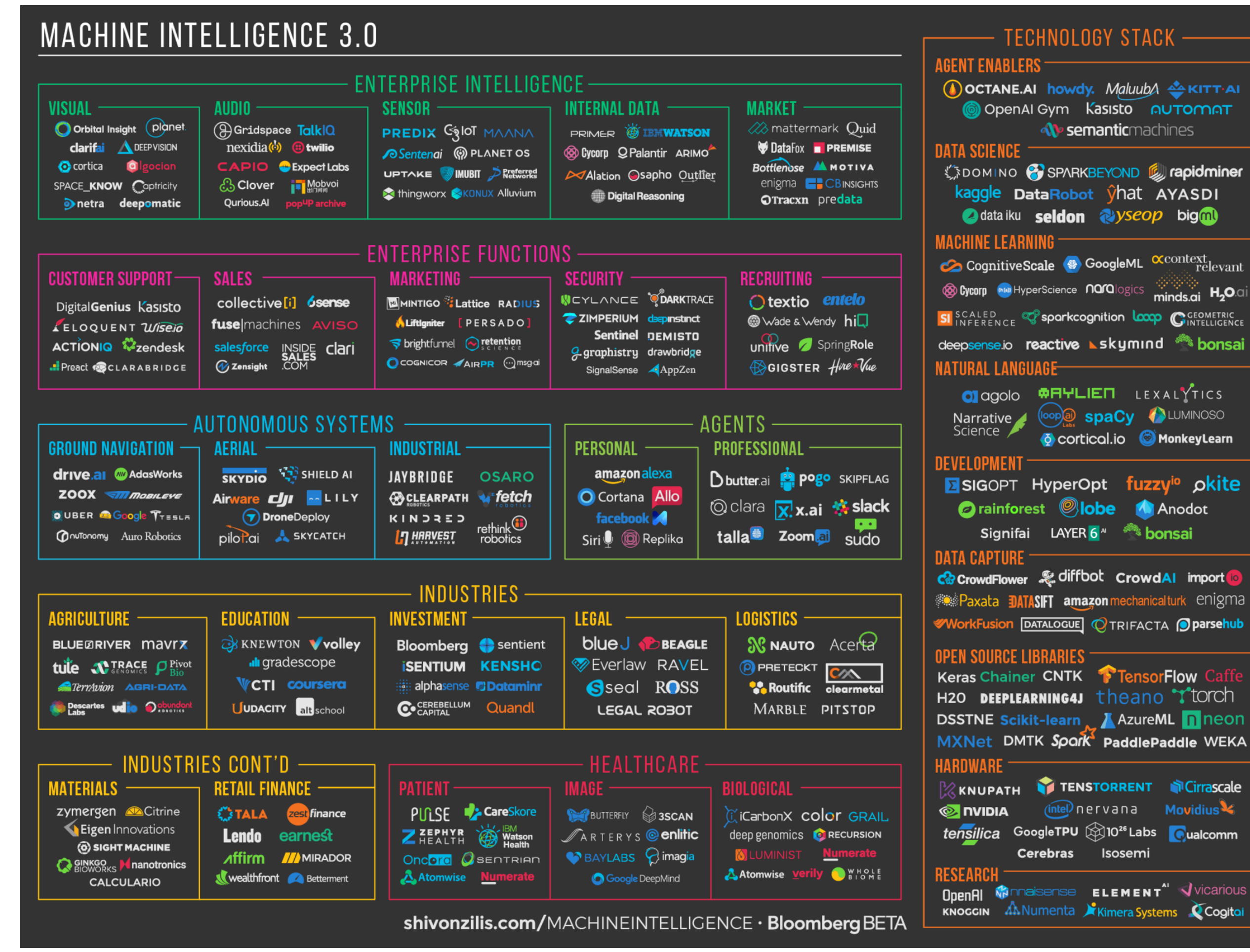
*and elsewhere ...*



# Machine Learning in Industry


https://www.nvidia.com/en-us/deep-learning-ai/

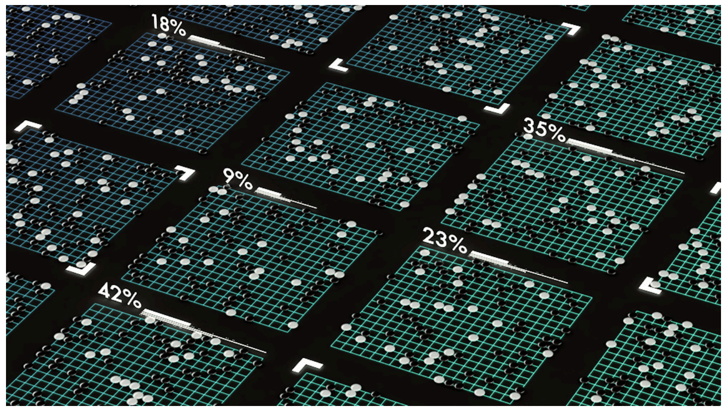


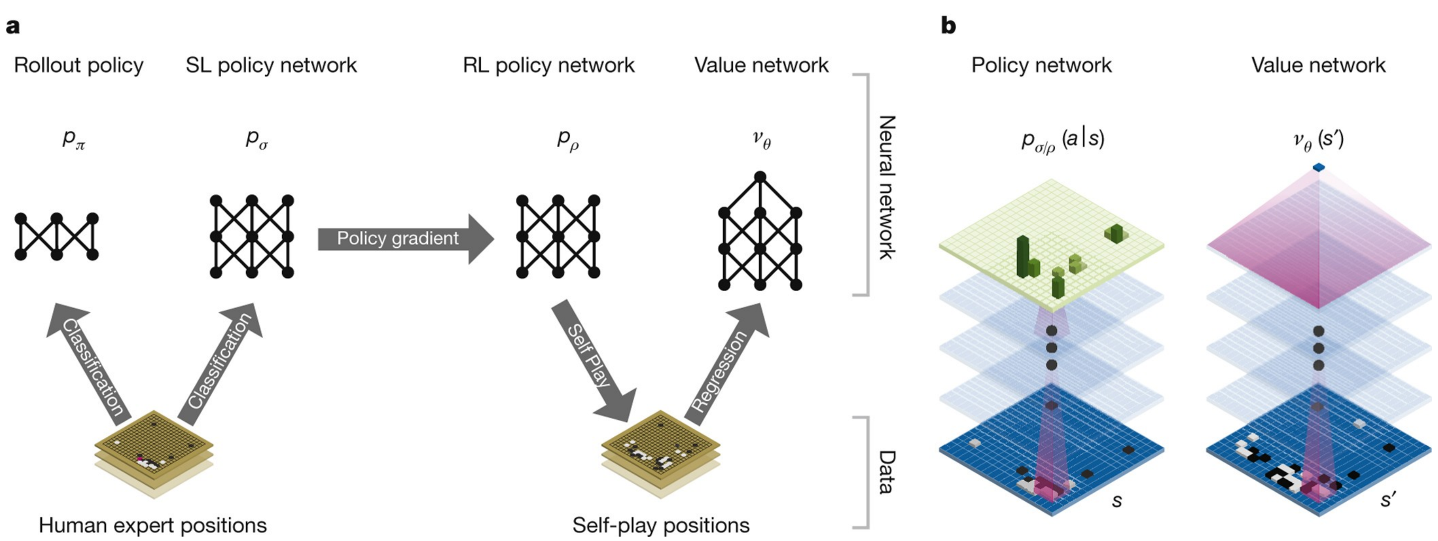




http://www.shivonzilis.com/machineintelligence

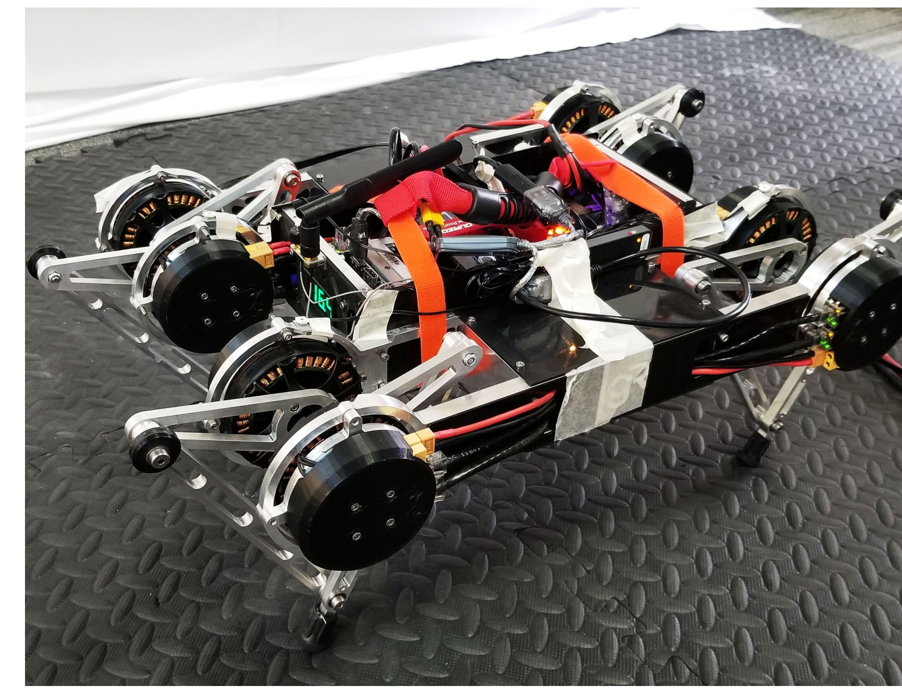
Prominent skill in industry nowadays. Lots of data, lots of applications, lots of potential use cases, lots of money. Knowing machine learning can open significantly **career horizons.**

# Learning to Control



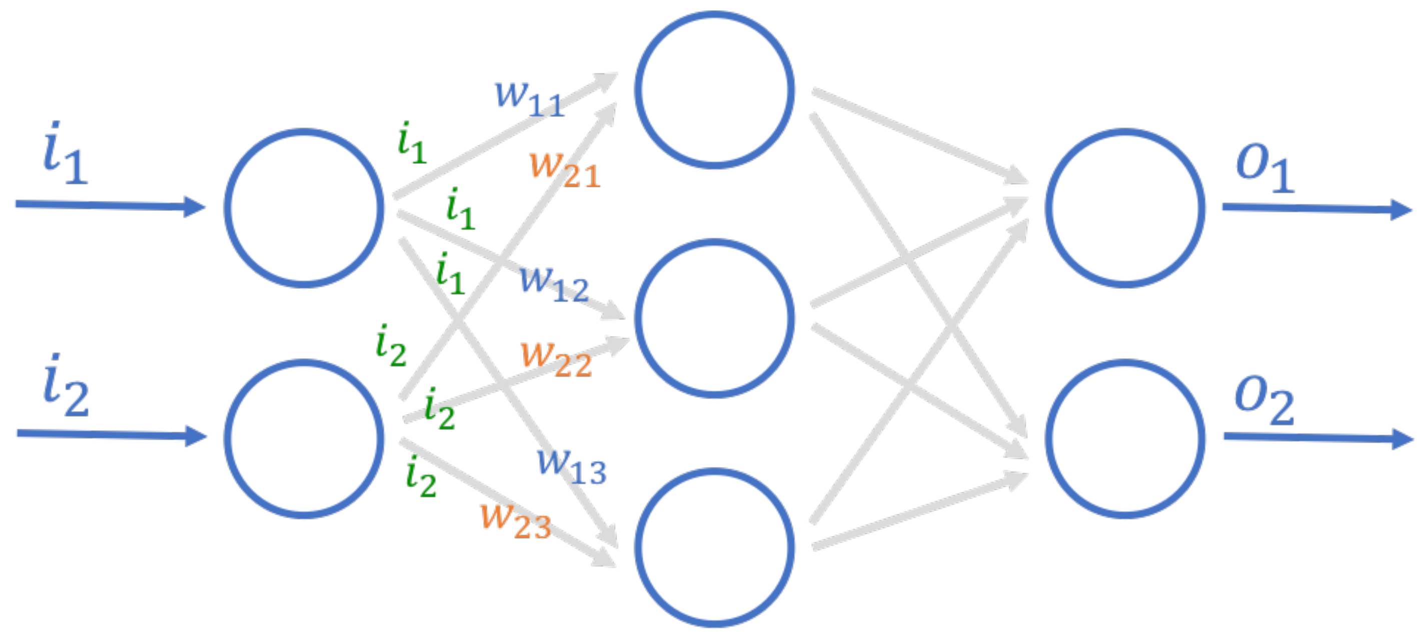

Mastering the game of Go with deep neural networks and tree search,
https://doi.org/10.1038/nature16961



Learning to Walk via Deep Reinforcement Learning
https://arxiv.org/abs/1812.11103

Modern machine learning **boosts control technologies**.
AI, gaming, robotic, self-driving vehicle, etc.

# Operation Vectorization

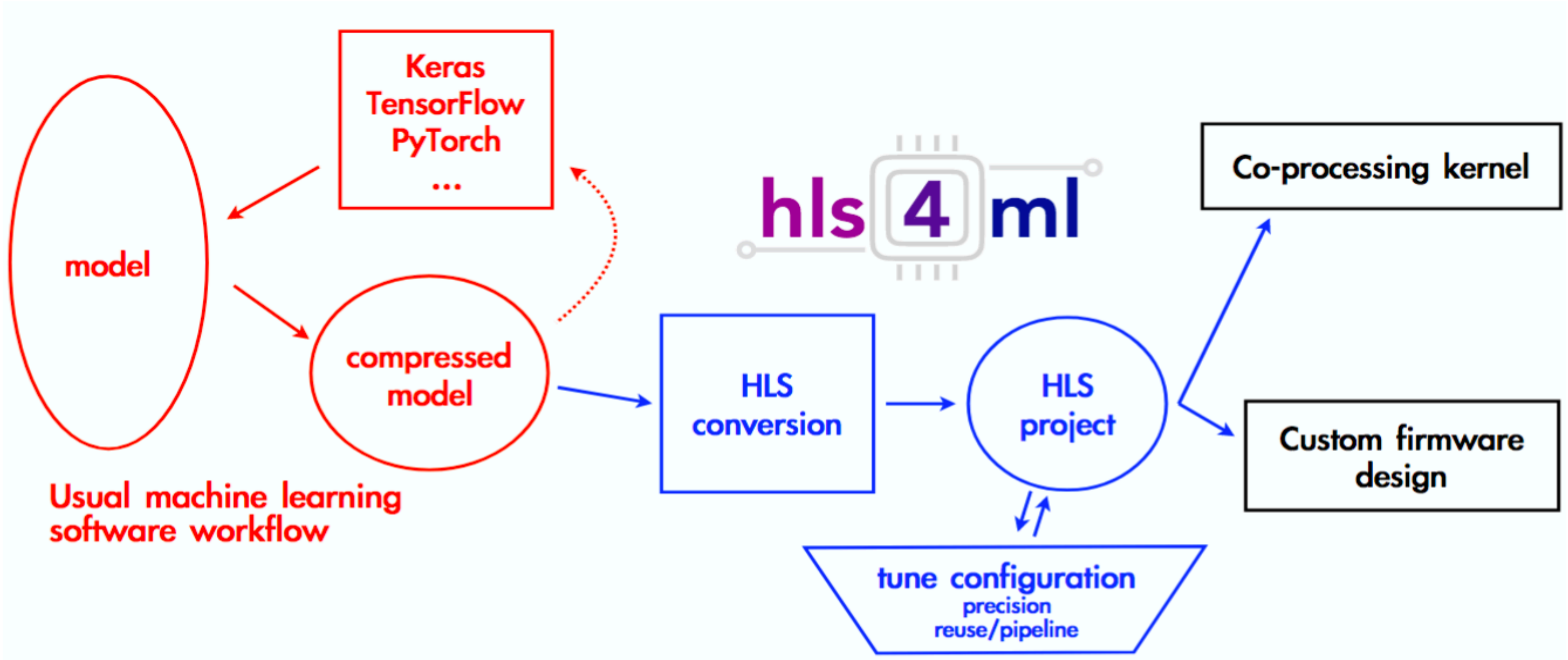



ANN ≡ matrix operations ≡ parallelizable

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

Computation of prediction from artificial neural network model can be **vectorized to a large extend.**

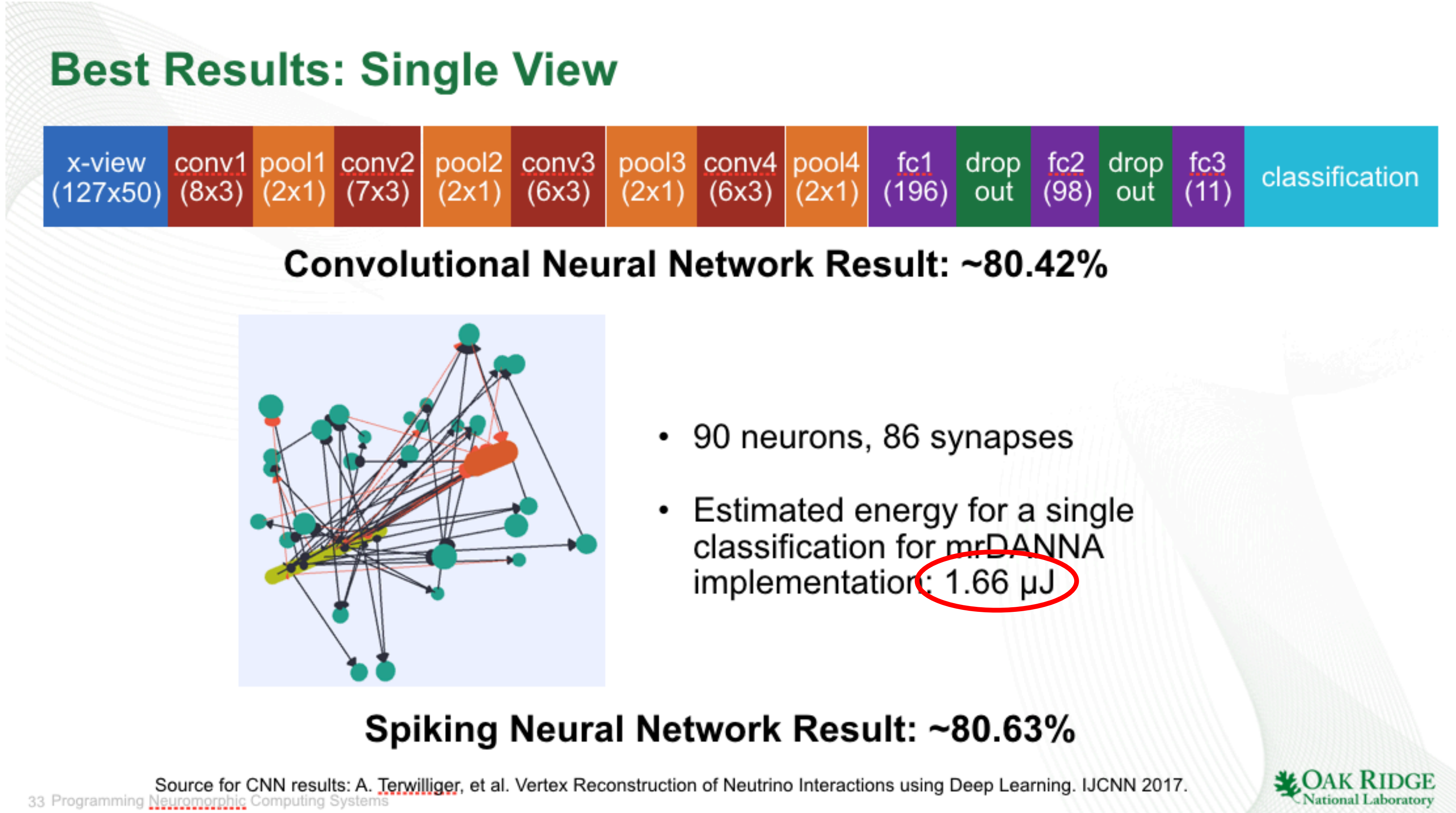
# Hyper-Fast Prediction




Synthesizing FPGA firmware from trained ANN
https://fastmachinelearning.org/hls4ml/
J. Duarte et al.[1804.06913]

Artificial neural network model can be **executed efficiently on FPGA**, GPU, TPU, ...

# Low Power Prediction




https://indico.fnal.gov/event/13497/contribution/0 *Slide C. Schuman*

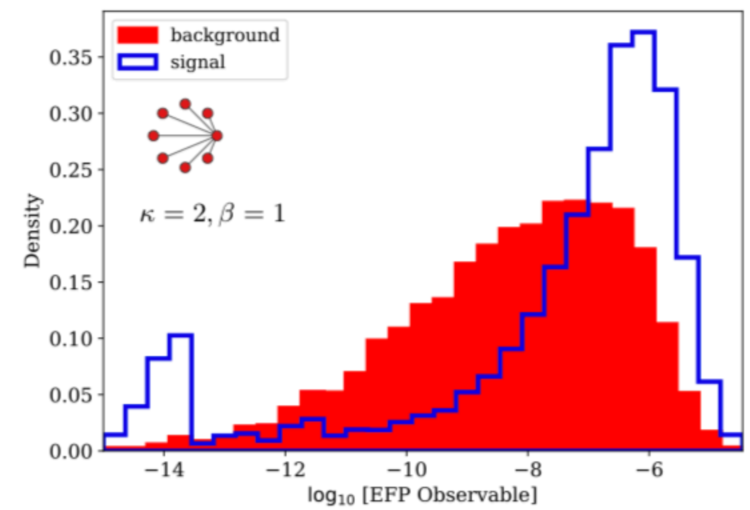
Neuromorphic hardware dedicated to **spiking neural networks**
**Low power** consumption by design

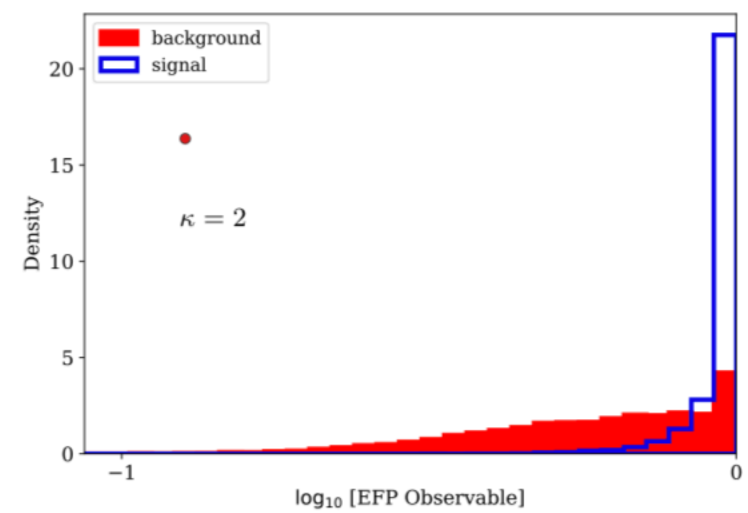
# Learning Observables

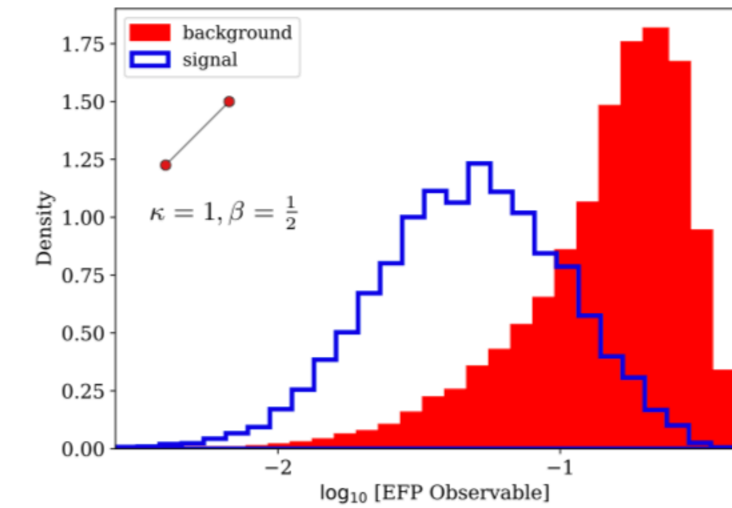
Electron classification performance

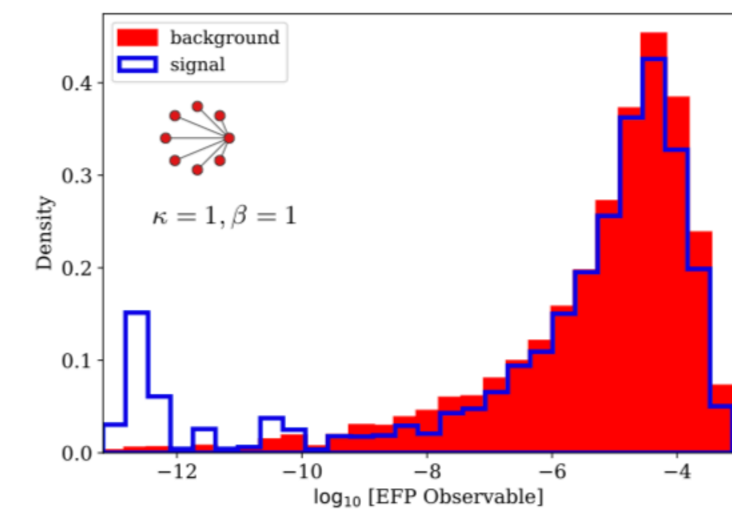
| Base | | Additions $(\kappa, \beta)$ | | (AUC) |
|---|---|---|---|---|
| 7HL | | | | 0.945 |
| 7HL | $+M_{\text{jet}}$ | | | 0.956 |
| 7HL | | $(1, \frac{1}{2})$ | | 0.970 |
| 7HL | $+M_{\text{jet}}$ | $(1, 1)$ | $(1, \frac{1}{2})$ | 0.971 |
| 7HL | | $(2, -)$ | | 0.970 |
| 7HL | $+M_{\text{jet}}$ | $(2, 1)$ | $(2, -)$ | 0.971 |
| CNN | | | | 0.972 |

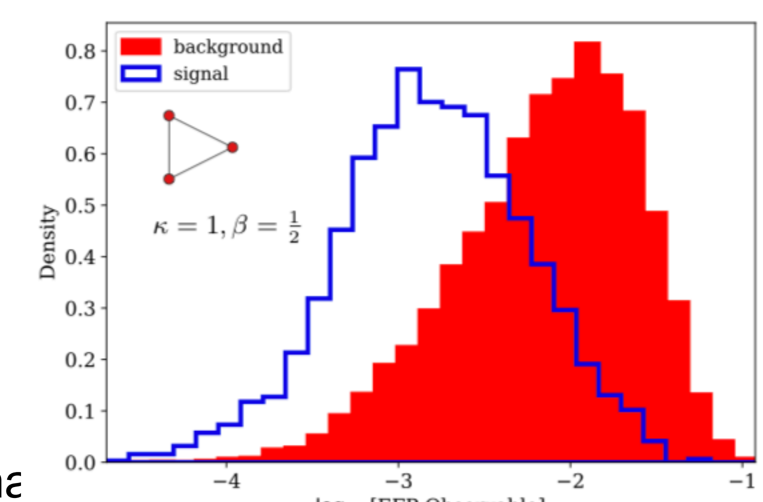
[2010.11998], [2011.01984]



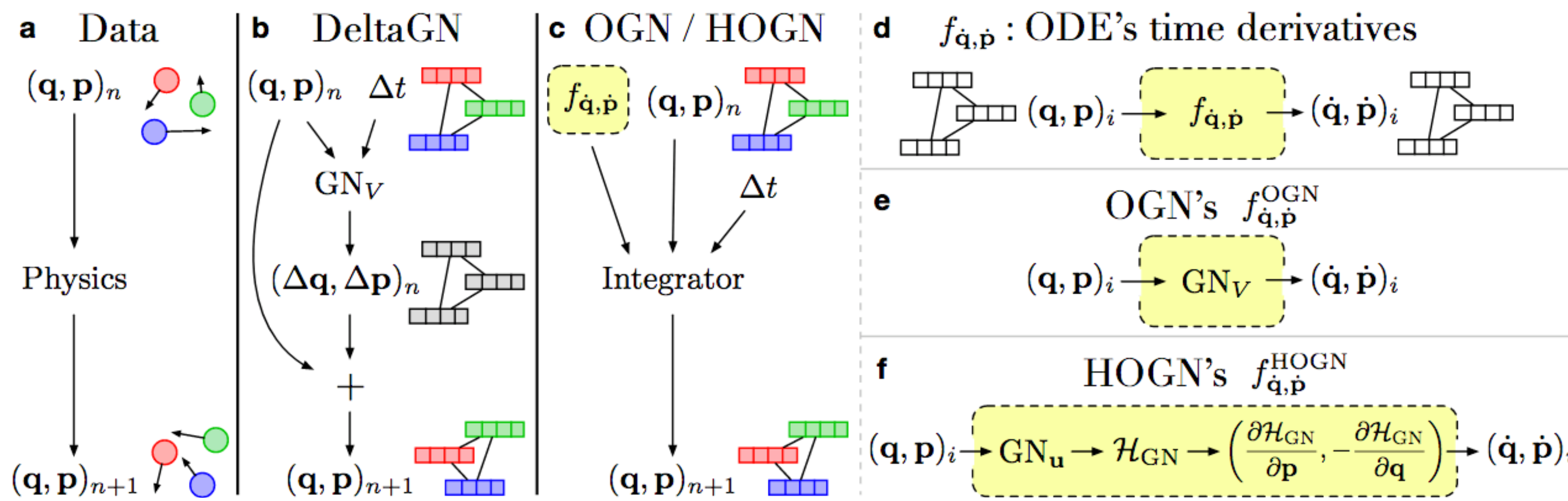



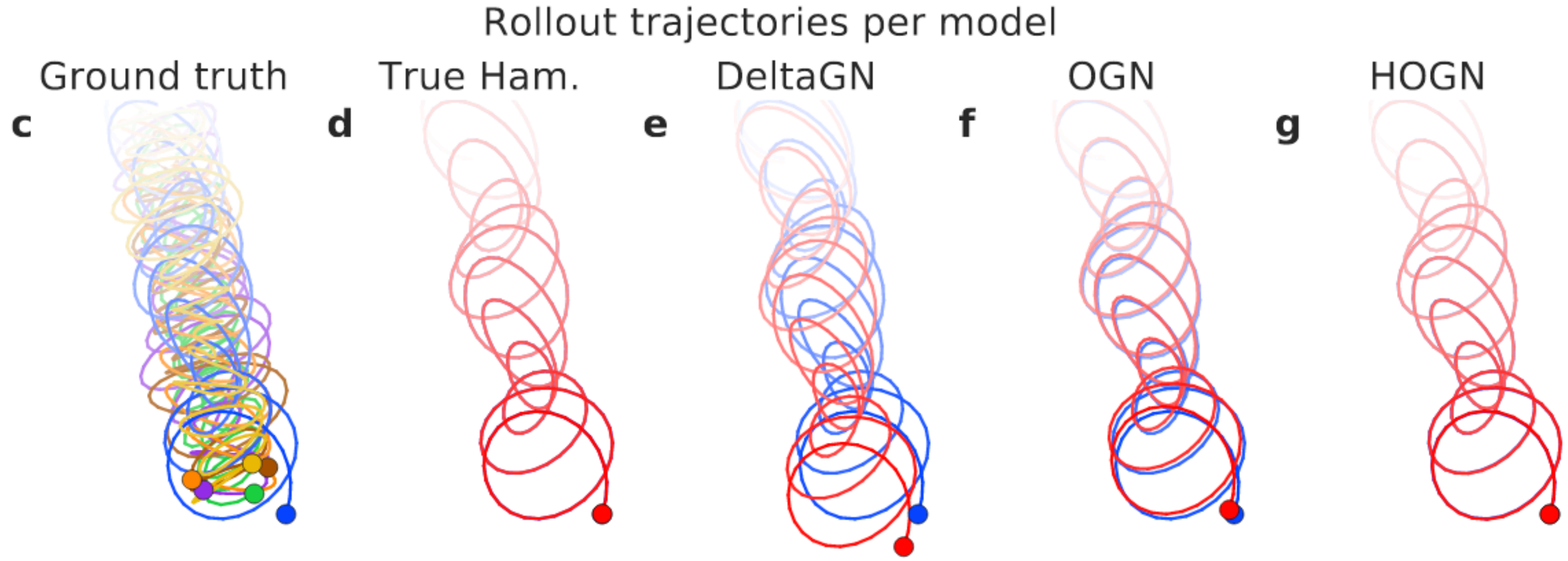




Machine Learning can **help understand Physics**.

# Use Physics

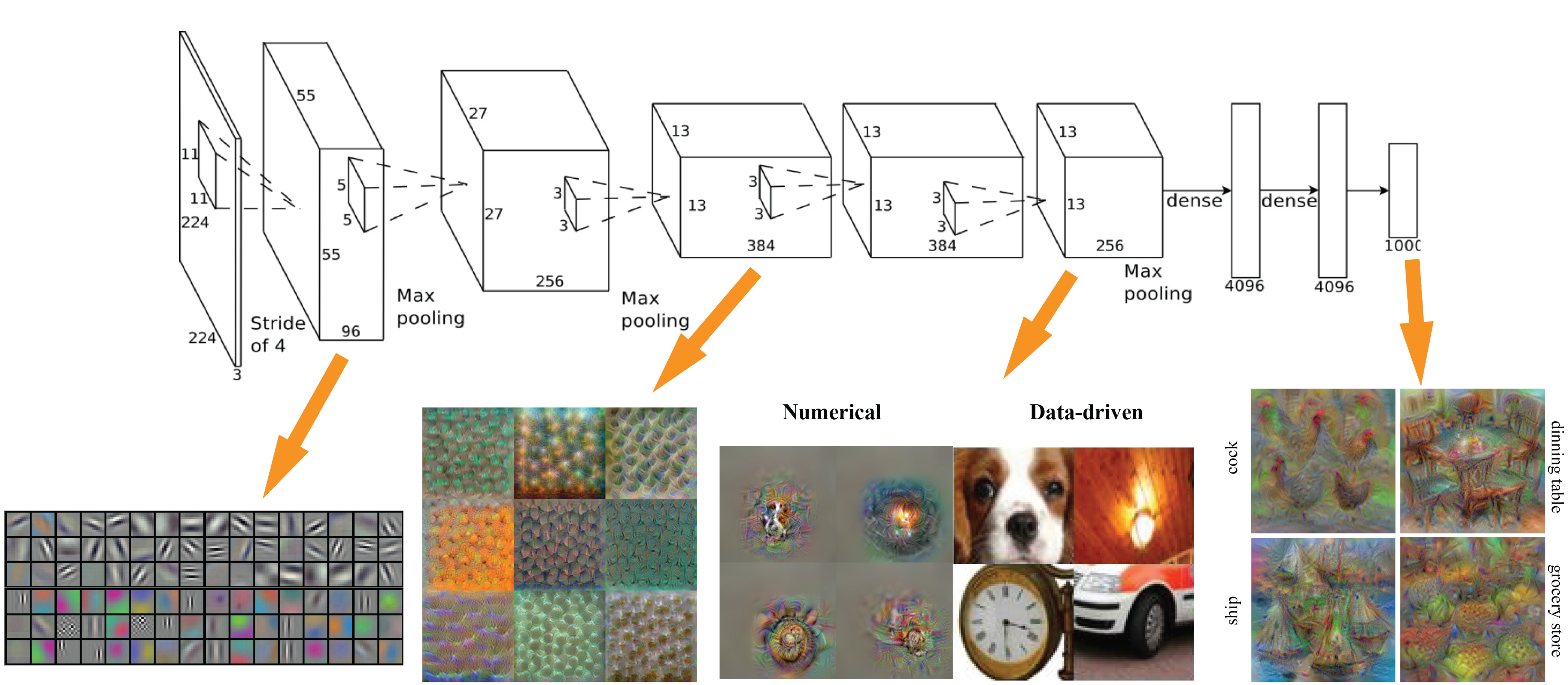



A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, P. Battaglia [1909.12790]

Let the model **include Physics principles** to master convergence

# Learning from Complexity

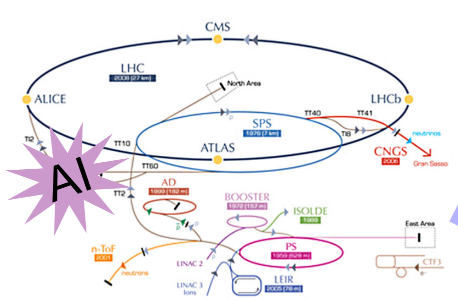



Conv 1: Edge+Blob Conv 3: Texture Conv 5: Object Parts Fc8: Object Classes

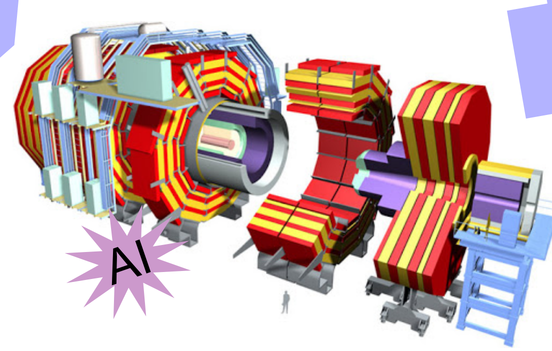
Machine learning model can **extract information from complex dataset.**
More classical algorithm counter part may
take **years of development.**

# AI in HEP

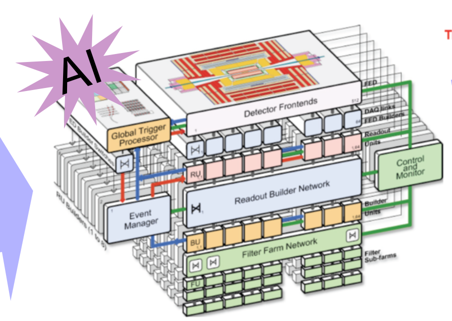
**Role of AI**: accelerator control, data acquisition, event triggering, anomaly detection, new physics scouting, event reconstruction, event generation, detector simulation, LHC grid control, analytics, signal extraction, likelihood free inference, background rejection, new physics searches, ...




Large Hadron Collider
40 MHz of collision



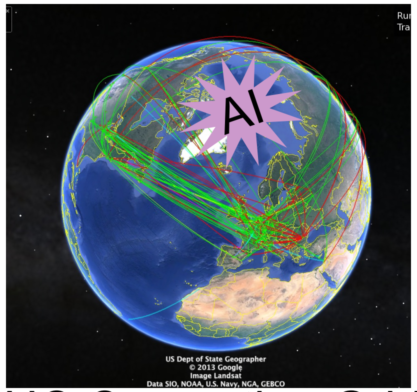

CMS Detector
1PB/s



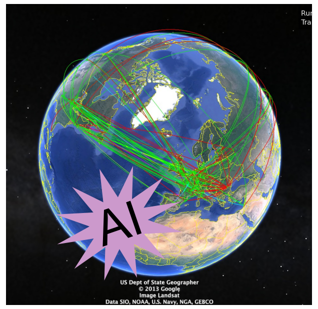

CMS L1 & High-Level Triggers
50k cores, 1kHz


CERN Tier-0
Computing Center
20k cores

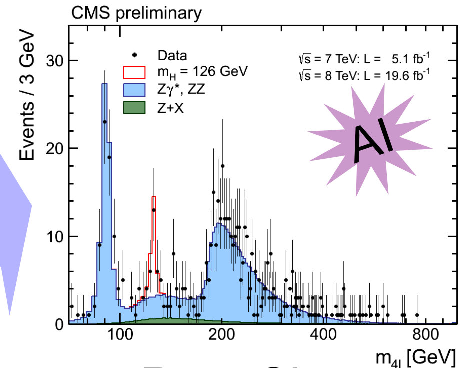

CERN Tier-0/Tier-1
Tape Storage
200PB total




LHC Computing Grid
200k cores pledge to CMS over ~100 sites



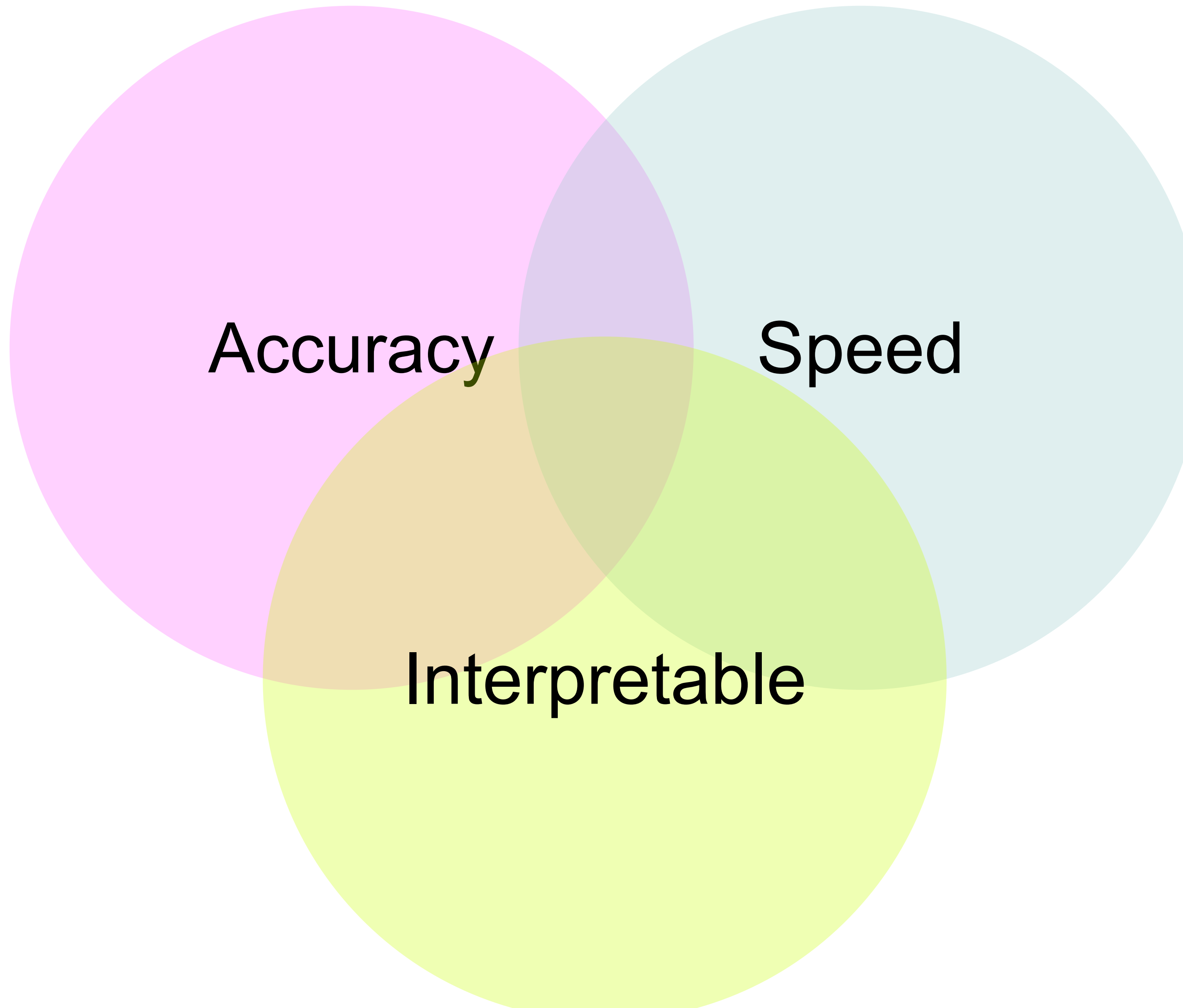

LHC Grid
Remote Access
to 100PB of data




Rare Signal
Measurement
~1 out of $10^6$

- Up to date listing of references: https://github.com/iml-wg/HEPML-LivingReview

# Possible Utilizations




- **Fast surrogate** models (trigger, simulation, etc) ; even better if more accurate.
- **More accurate** than existing algorithms (tagging, regression, etc) ; even better if faster.
- Model performing **otherwise impossible tasks** (operations, etc)

# Growing Literature


https://inspirehep.net/literature?q=machine learning or deep learning

Community-based up to date listing of references
https://iml-wg.github.io/HEPML-LivingReview/

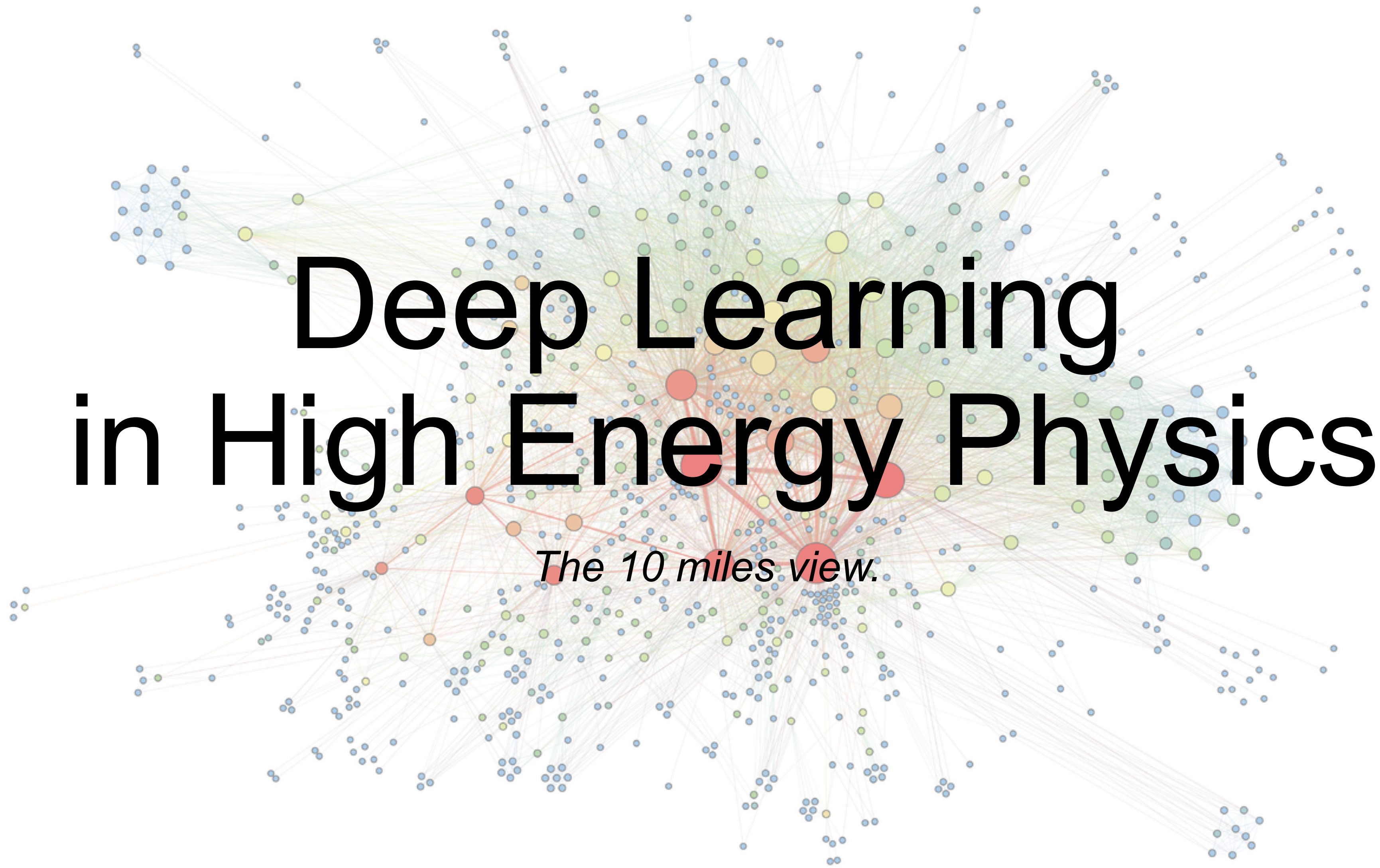
*Take home message :*

*Machine Learning is a widely recognized and used technology in industry*

*Deep Learning has the potential of helping Science to make progress*

*Neural Networks could help with the computing requirements of Science*

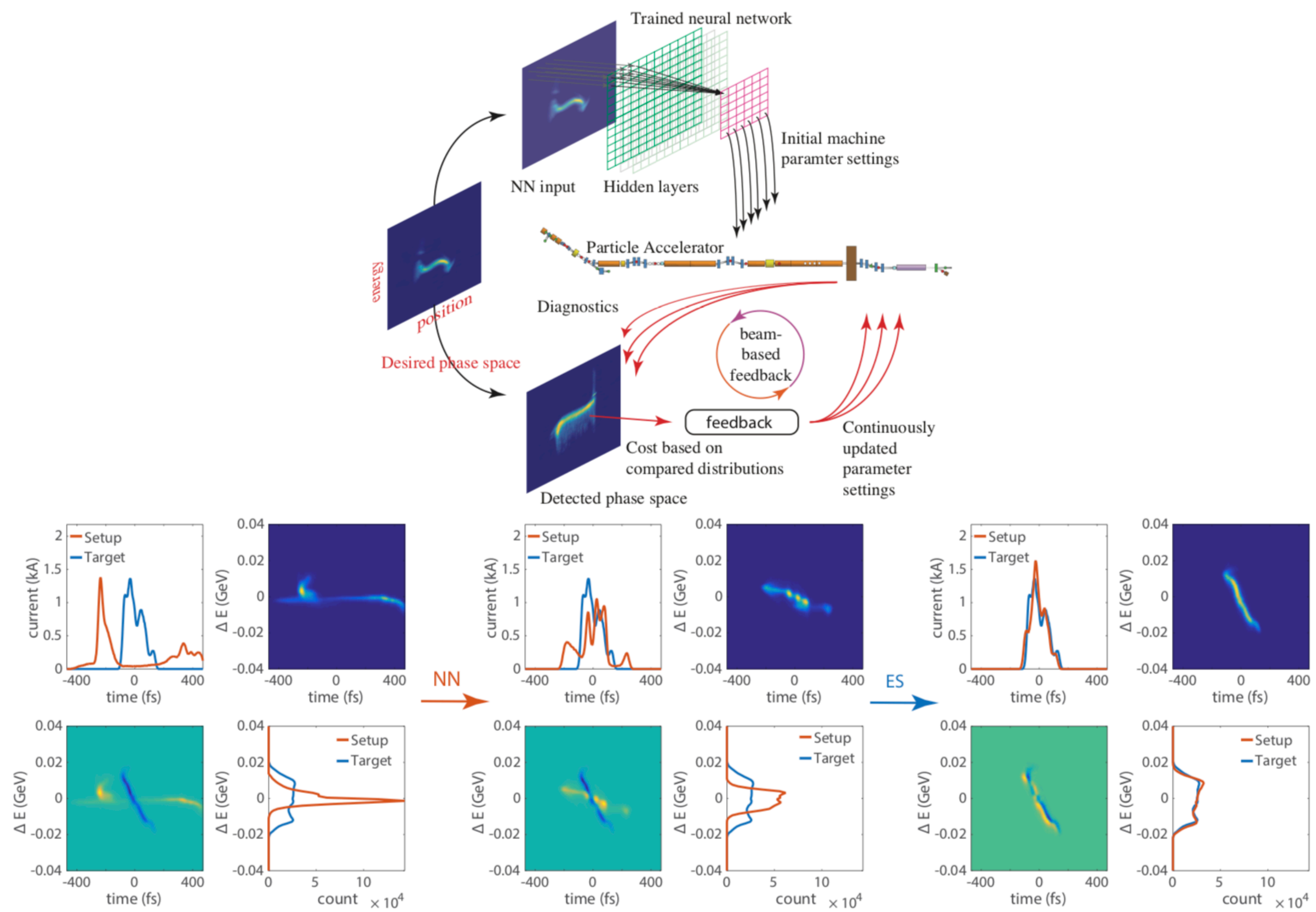
*Wide range of potential applications*



# Deep Learning in High Energy Physics

*The 10 miles view.*

# Producing the Data

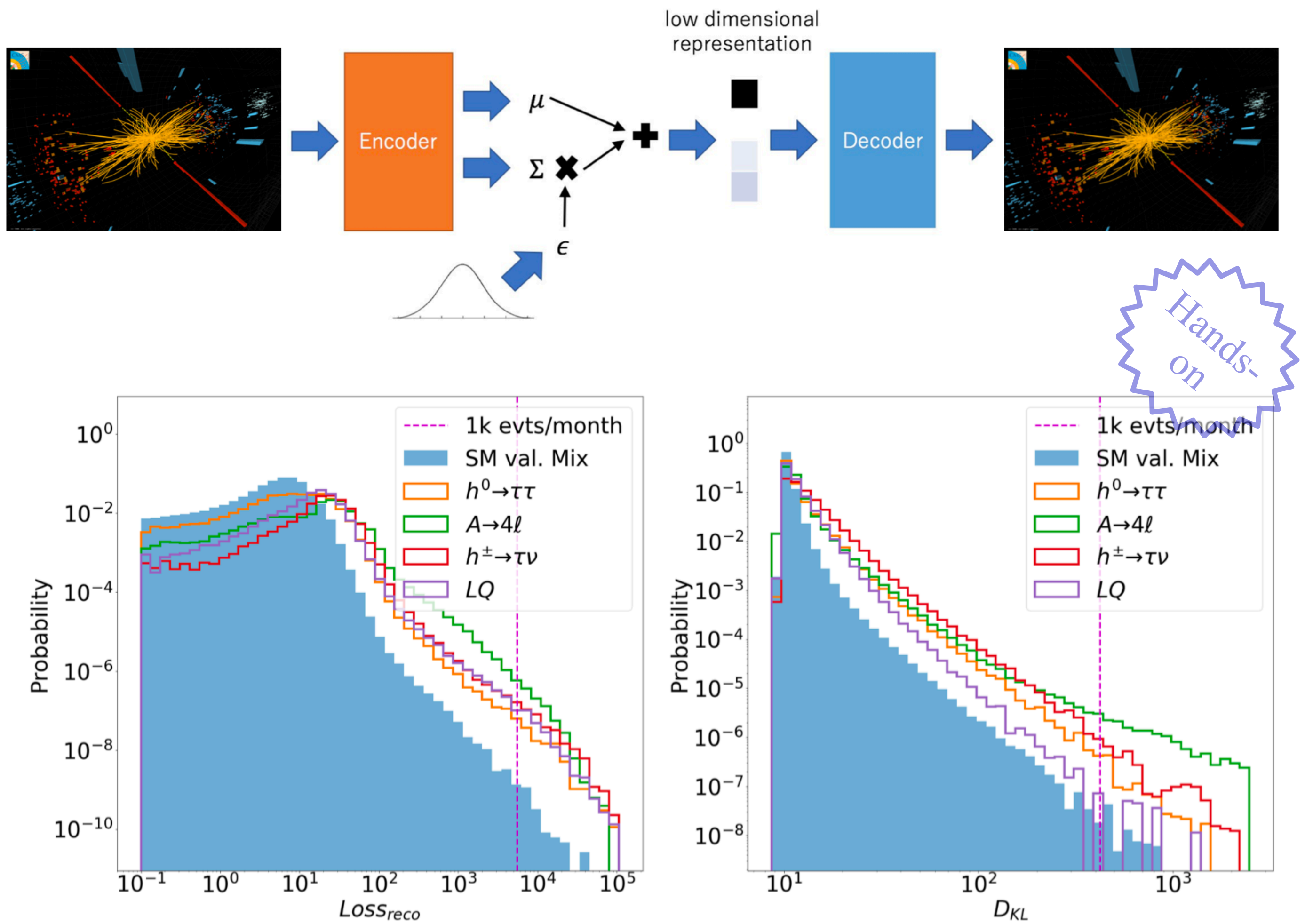



A. Scheinker, C. Emma, A.L. Edelen, S. Gessner [2001.05461]

- Machine learning can be used to tune devices, control beams, perform analysis on accelerator parameters, etc.
- Already successfully deployed on accelerator facilities.
- More promising R&D to increase beam time.

Opportunities in Machine Learning for Particle Accelerators [1811.03172]
Machine learning for design optimization of storage ring nonlinear dynamics [1910.14220]
Advanced Control Methods for Particle Accelerators (ACM4PA) 2019 Workshop Report [2001.05461]
Machine learning for beam dynamics studies at the CERN Large Hadron Collider [2009.08109]
…

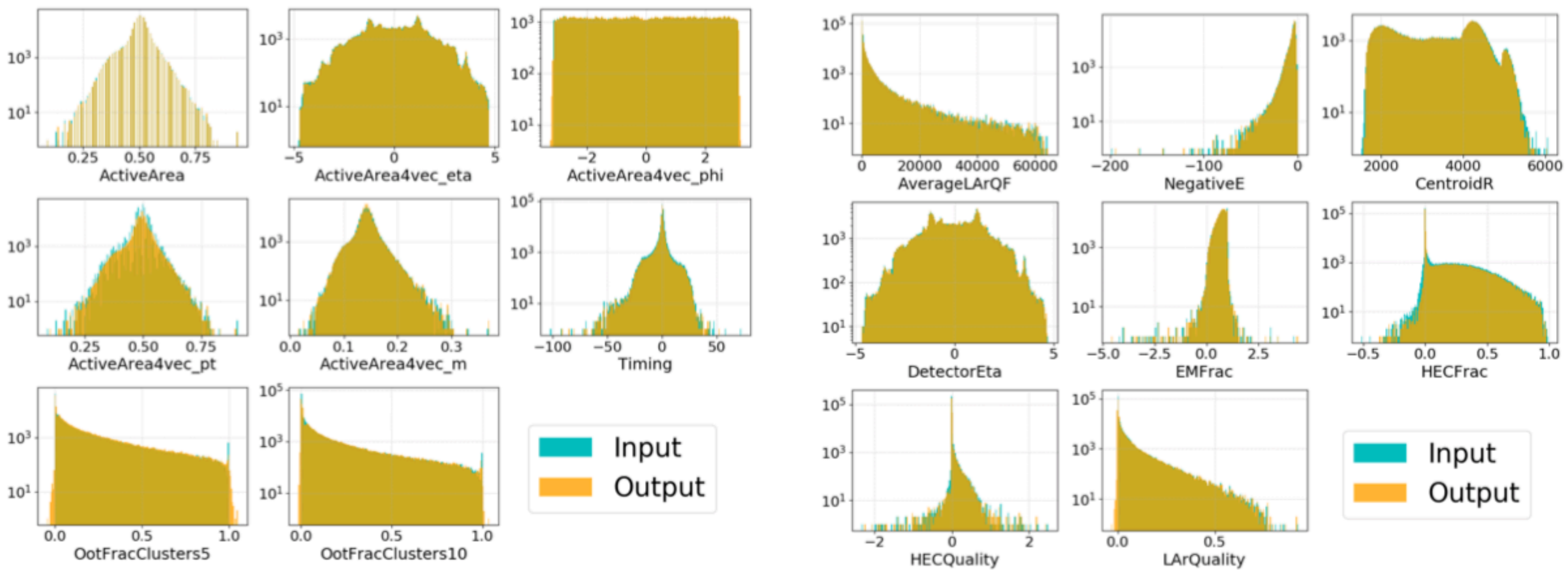
# Acquiring Data




Use of variational auto-encoders directly on data to marginalize outlier events, for anomalous event hotline operation.
[doi:0.1007/JHEP05(2019)036]

- Machine learning since long deployed in the trigger for selected signatures.
- Further potential for background trigger rate reduction.
- Emerging opportunity for triggering on unknown signatures.
- More promising R&D and experiment adoption.

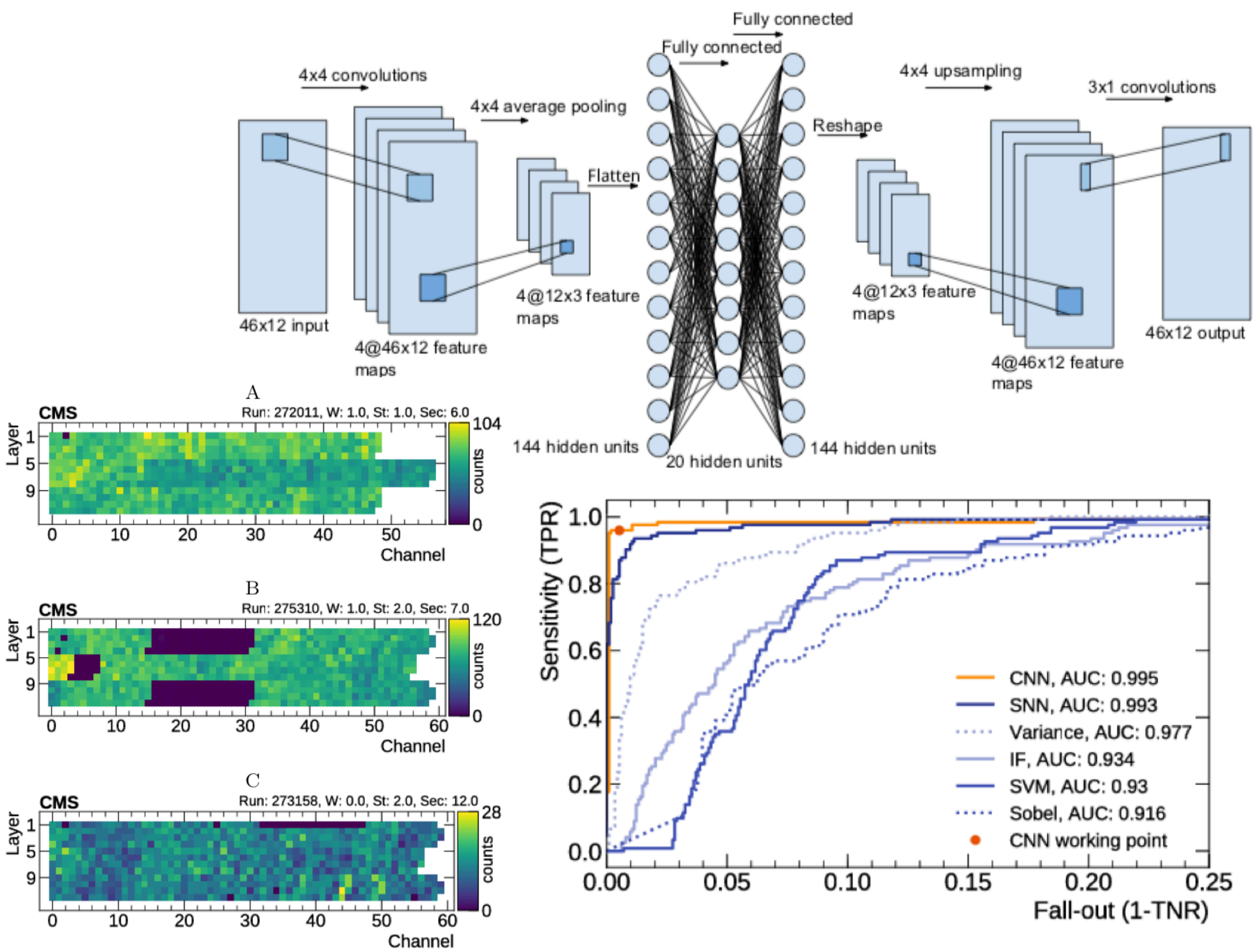
# Compressing Data




Use of auto-encoder model
http://lup.lub.lu.se/student-papers/record/9004751

- Rich literature on data compression of image with neural network.
- Make use of abstract semantic space for image compression.
- Image compression can suffer some loss of resolution.
- Saving on disk/tape cost. Potential in scouting data analysis.
- R&D needed to reach the necessary level of fidelity.

# Cleaning Data

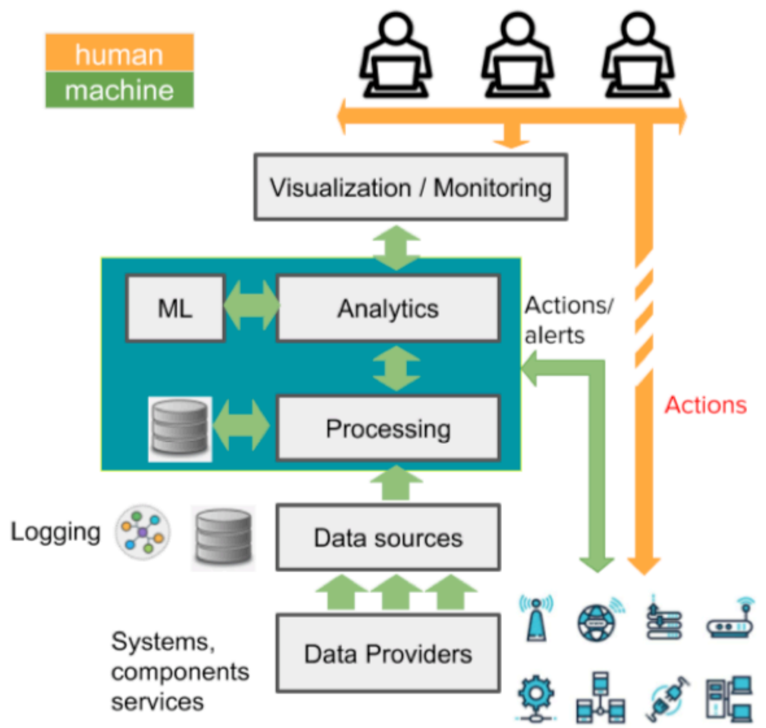



A.A. Pol, G. Cerminara, C. Germain, M. Pierini, A. Seth
[doi:10.1007/s41781-018-0020-1]

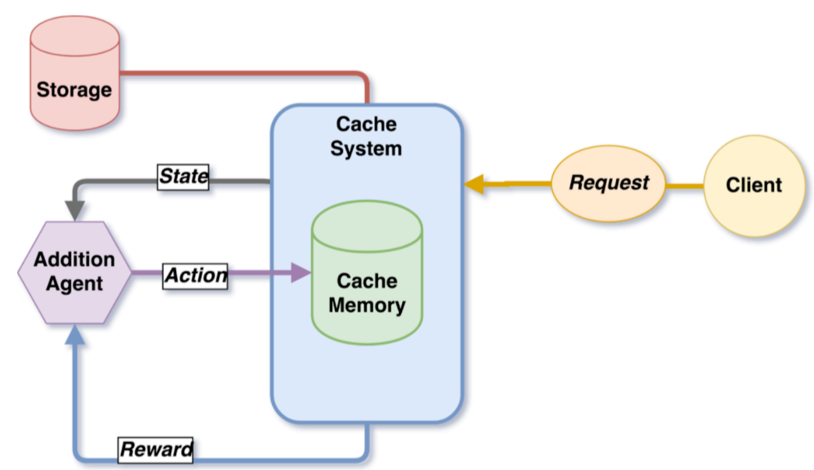
- Data quality is a person power intensive task, and crucial for swift delivery of Physics
- Machine learning can help with automation.
- Learning from operators, reducing workload.
- Continued R&D and experiment adoption.

Towards automation of data quality system for CERN CMS experiment [doi:10.1088/1742-6596/898/9/092041]
LHCb data quality monitoring [doi:10.1088/1742-6596/898/9/092027]
Detector monitoring with artificial neural networks at the CMS experiment at the CERN Large Hadron Collider [1808.00911]
Anomaly detection using Deep Autoencoders for the assessment of the quality of the data acquired by the CMS experiment [doi:10.1051/epjconf/201921406008]
…

# Managing Data




[cds:2709338]
https://operational-intelligence.web.cern.ch

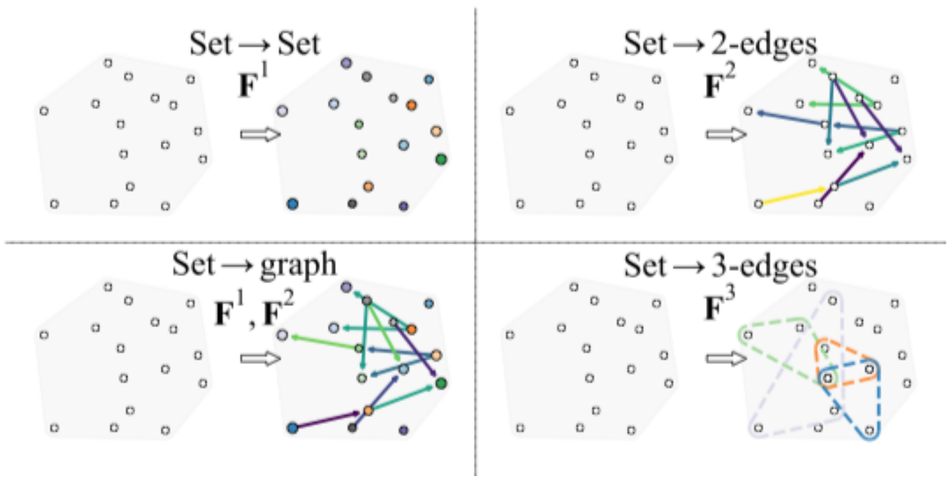


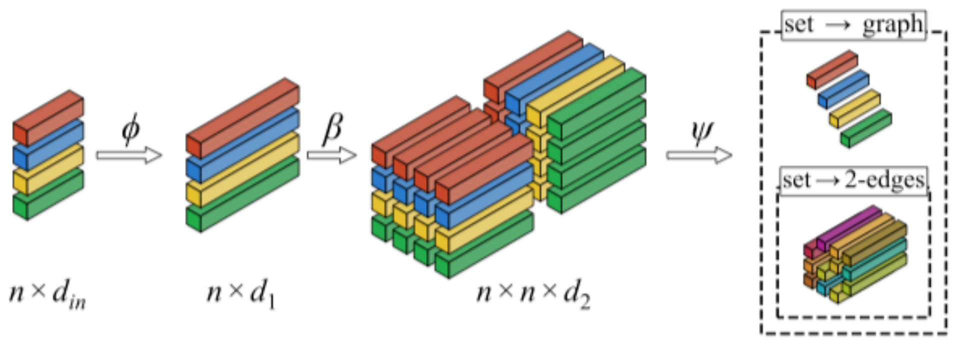

| Cache Type | Throughput | Cost | Read on hit ratio | Band sat. | CPU Eff. |
|---|---|---|---|---|---|
| SCDL | **79.43%** | **50.68%** | 21.22% | 58.94% | 58.75% |
| LFU | 65.01% | 104.73% | **33.29%** | **51.00%** | **60.92%** |
| Size Big | 49.02% | 111.73% | 28.55% | 54.40% | 60.41% |
| LRU | 47.15% | 112.84% | 27.64% | 54.93% | 59.90% |
| Size Small | 46.71% | 113.01% | 27.39% | 55.01% | 59.73% |

Caching suggestions using Reinforcement Learning
LOD 2020, in proceedings

- The LHC-grid is key to success of the LHC experiments.
- Complex ecosystem with dedicated operation teams.
- Person power demanding, and inefficient in some corner of the phase space.
- Potential for AI-aided operation.
- Lots of modeling and control challenges.
- R&D to increase operation efficiency.

# Reconstructing Data

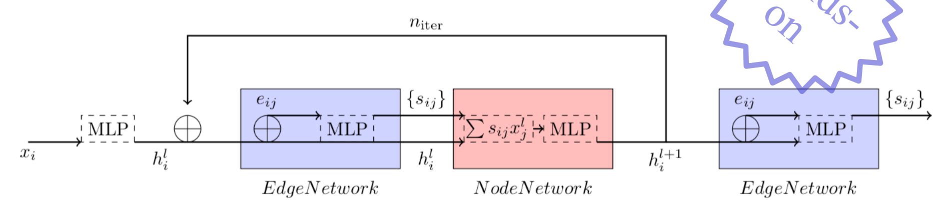


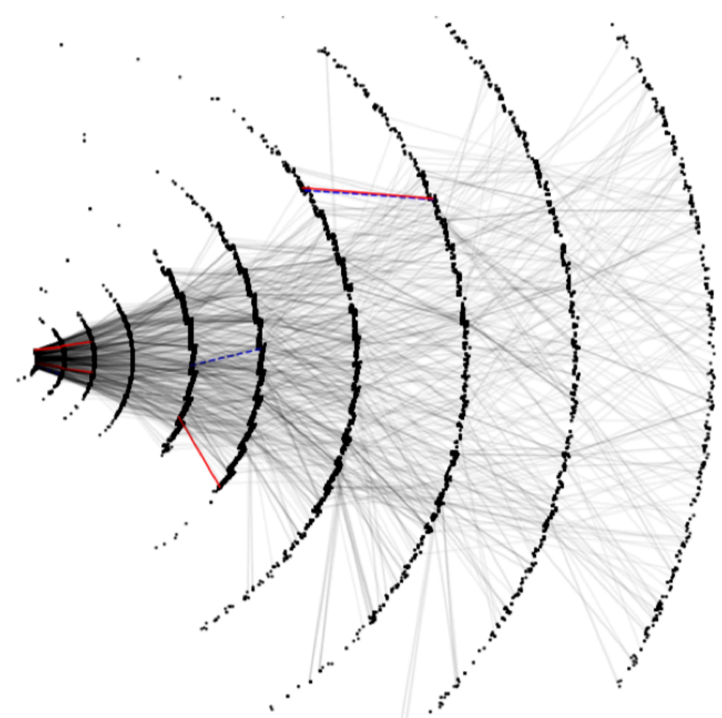




Learning graphs from sets, applied to vertexing
[2002.08772]

GNN applied to charged particle tracking
[2007.00149]

Hands-on

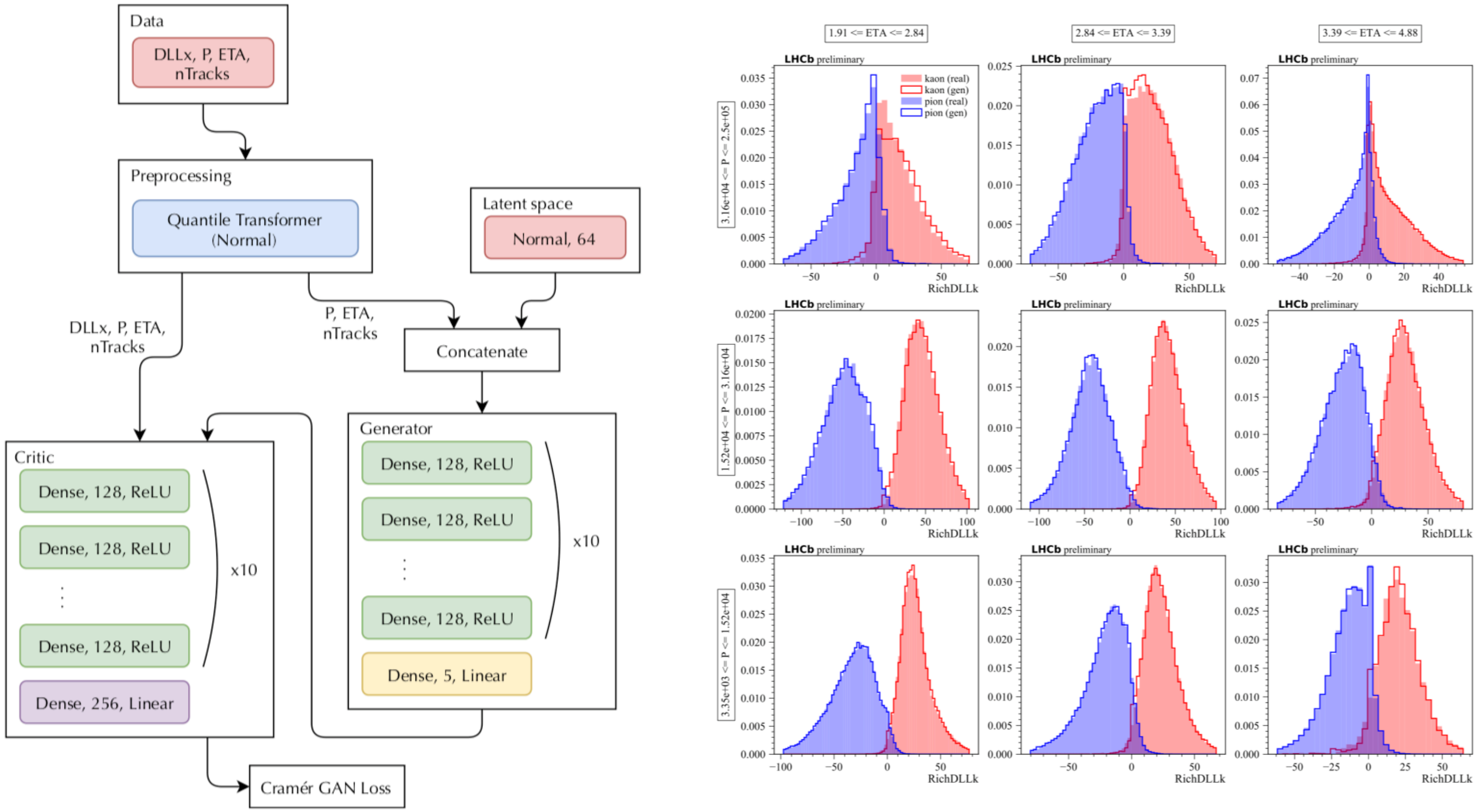


- Event reconstruction is pattern recognition to a large extend. Advanced machine learning techniques can help.
- Learn from the simulation, and/or data.
- Learn from existing "slow reconstruction" or simulation ground truth.
- Automatically adapt algorithm to new detector design.
- Image base methods evolving towards graph-based methods.
- Accelerating R&D to exploit full potential.

*Much more relevant work going on.*
*https://iml-wg.github.io/HEPML-LivingReview/*

# Simulating Data

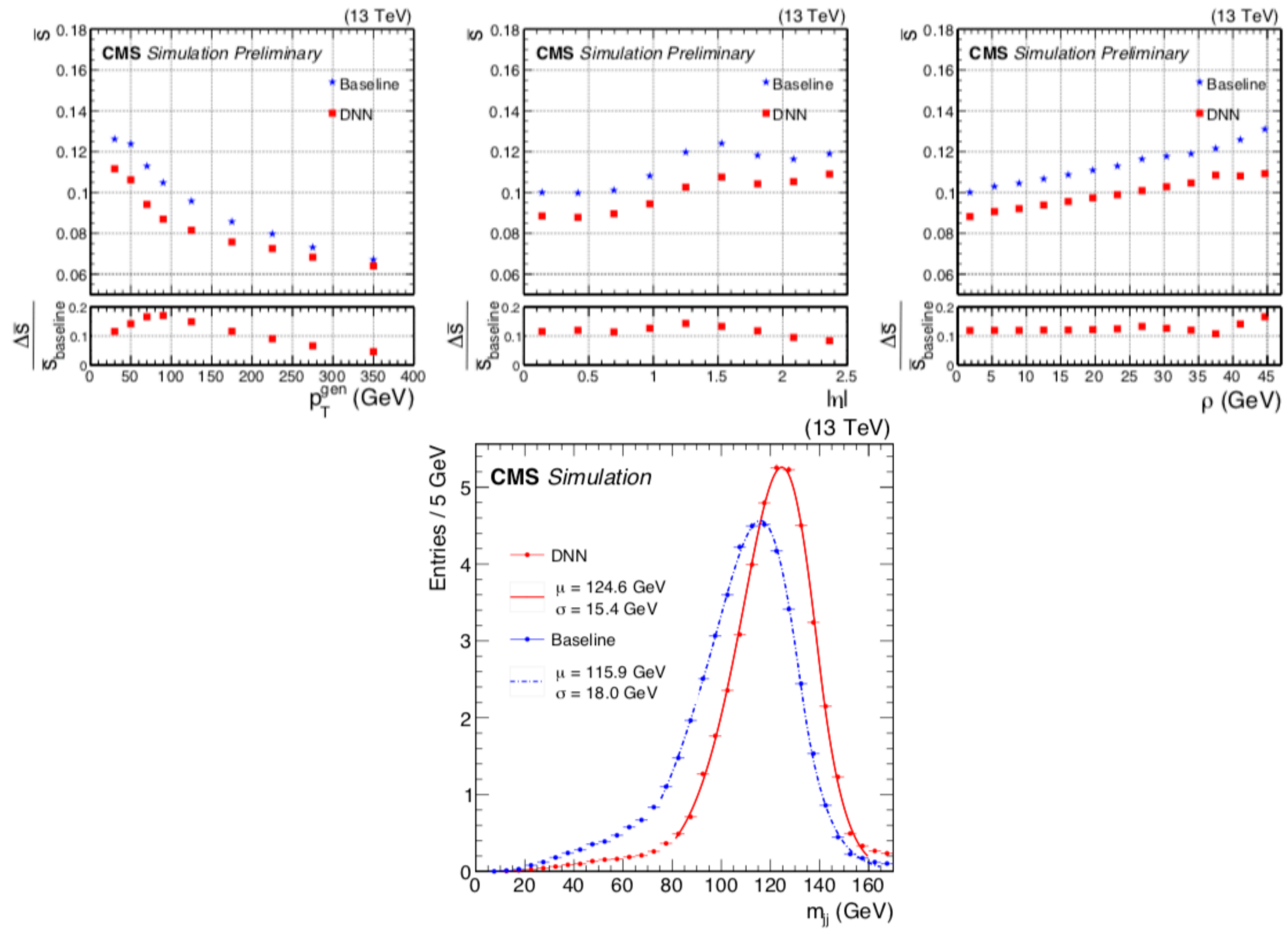



Generative Adversarial Networks for LHCb Fast Simulation
[2003.09762]

- Fully detailed simulation is computing intensive.
- Fast and approximate simulators already in operation.
- Applicable at many levels : sampling, generator, detector model, analysis variable, etc
- Generative models can provide multiple 1000x speed-up.
- Careful study of statistical power of learned models over training samples.
- Many R&D, experiment adoption starting.

*Much more relevant work going on.*
https://iml-wg.github.io/HEPML-LivingReview/

# Calibrating Data

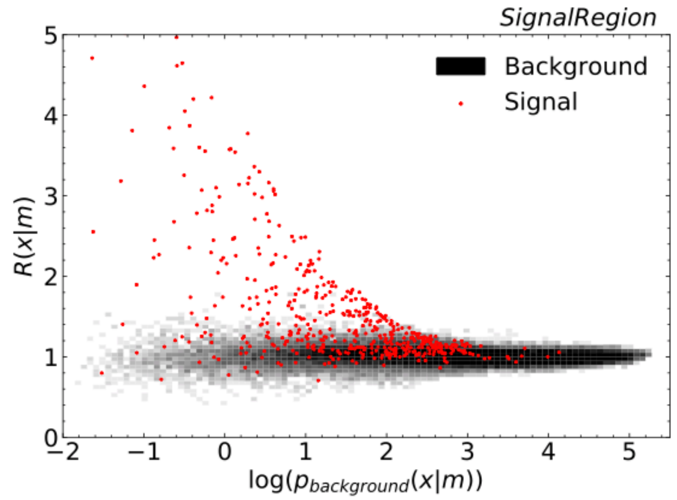


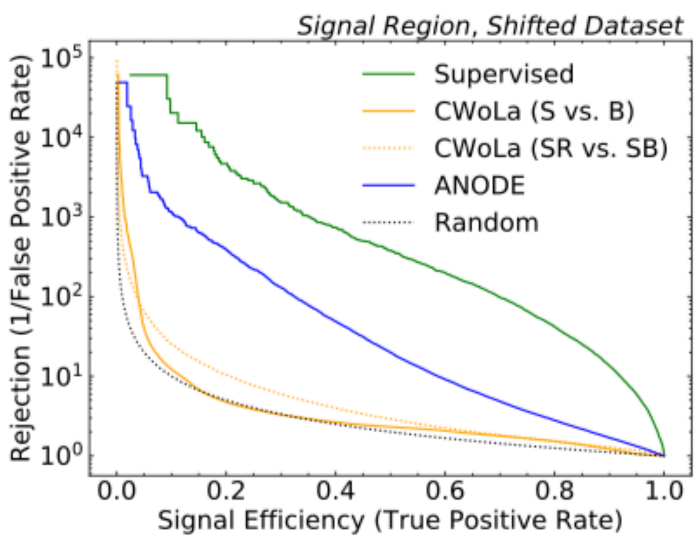

A deep neural network for simultaneous estimation of b jet energy and resolution [1912.06046]

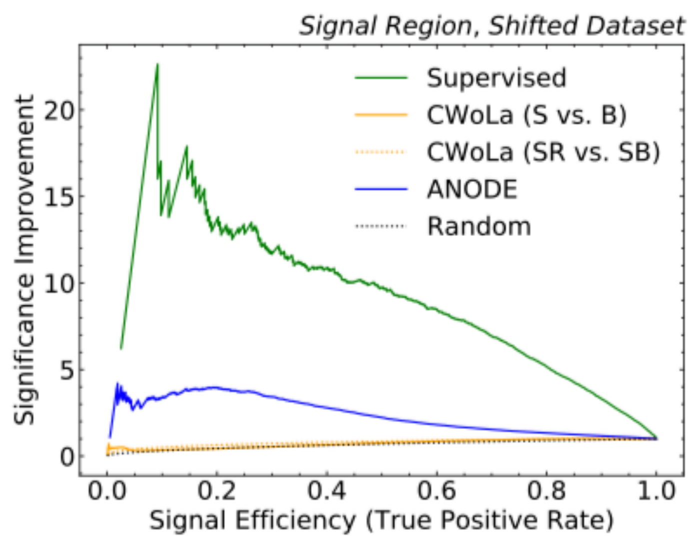
- Energy regression is the most obvious use case.
- Learning calibrating models from simulation and data.
- Parametrization of scale factors using neural networks.
- Reducing data/simulation dependency using domain adaptation.
- Continued R&D

*Much more relevant work going on.*
https://iml-wg.github.io/HEPML-LivingReview/

# Analyzing Data



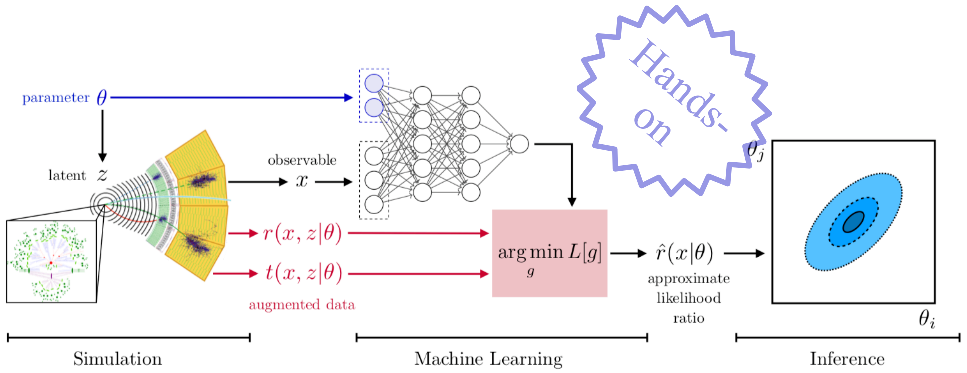



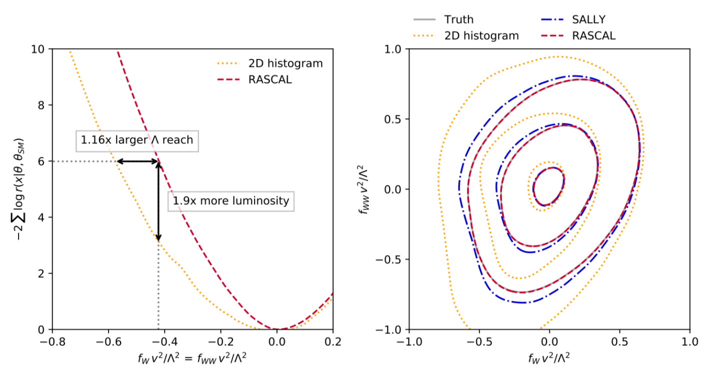

Use of masked autoregressive density estimator with normalizing flow as model-agnostic signal enhancement mechanism.
[doi:10.1103/PhysRevD.101.075042]

*Much more relevant work going on.*
*https://iml-wg.github.io/HEPML-LivingReview/*

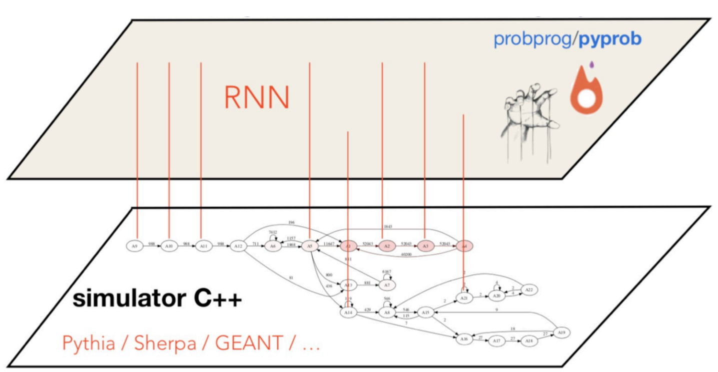
- Machine learning has long infiltrated analysis for signal/bkg classification.
- Increasing number of analysis with more complex DNN.
- Application to signal categorization, bkg modelling, kinematics reconstruction, decay product assignment, object identification, …
- Breadth of new model agnostic methods for NP searches.
- Continued R&D and experiment adoption initiated.

# Theory Behind the Data

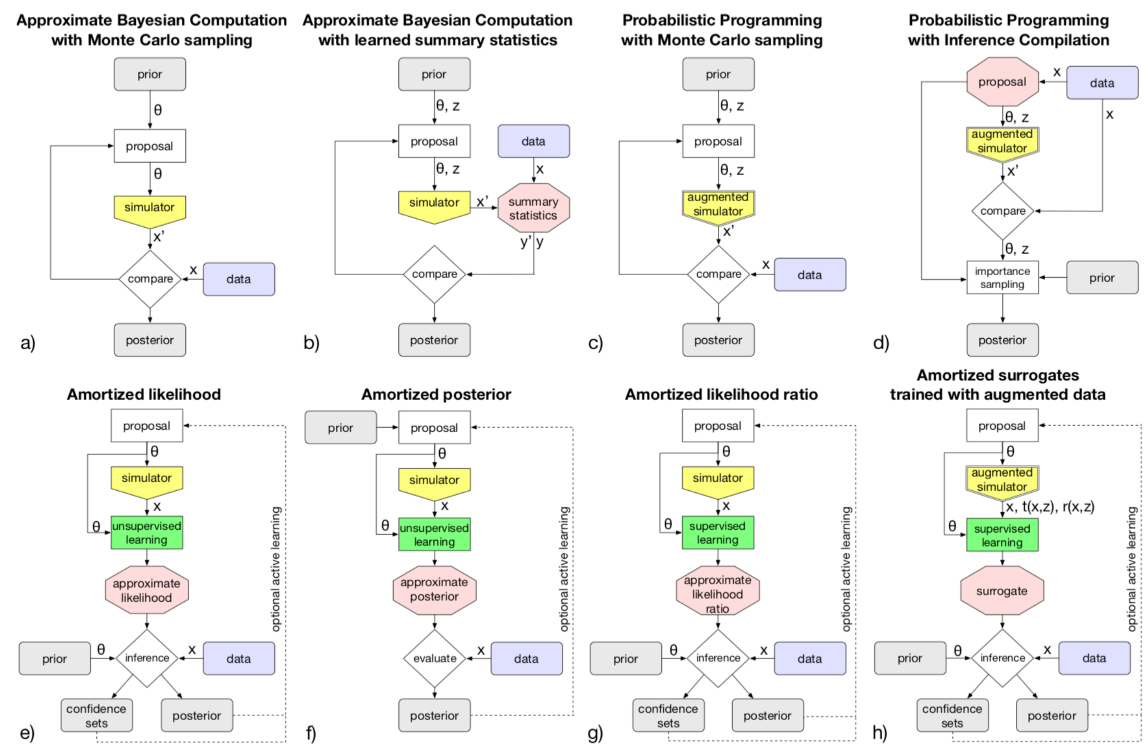



Constraining EFT with ML
[1805.00013]




https://github.com/probprog/pyprob




The frontiers of simulation-based inference
[1911.01429]

- Hypothesis testing is the core of HEP analysis.
- Intractable likelihood hinders solving the inverse problem.
- Going beyond the standard approach using machine learning and additional information from the simulator.
- More precise evaluation of the priors on theory's parameters.
- May involve probabilistic programming instrumentation of HEP simulator.
- R&D to bring this in the experiment.

*Take home message :*

*Rapid growth of machine learning applications in HEP*

*(too) Slowly turning proofs of concept into production*

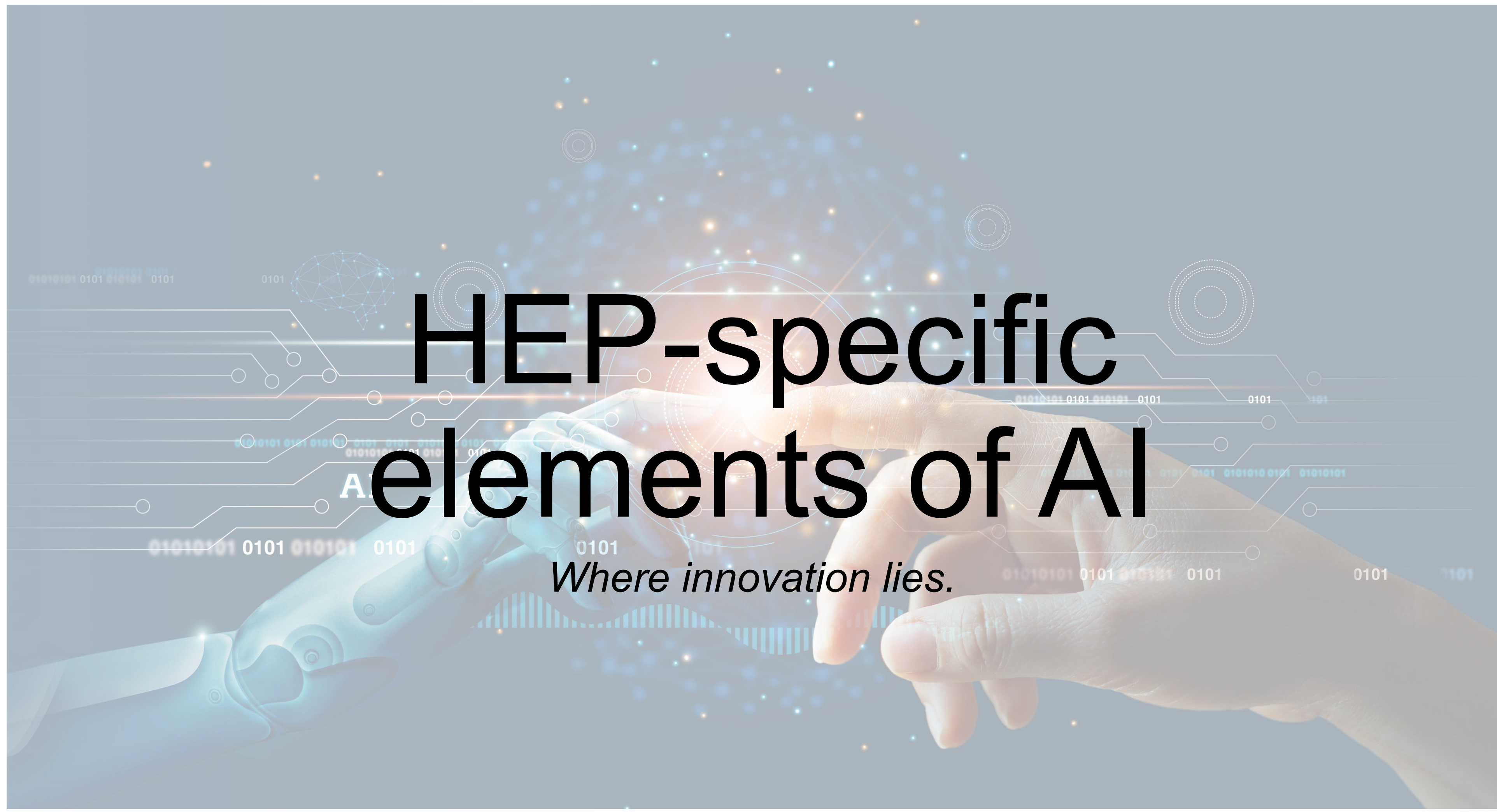
*Exciting time ahead exploiting further the potential of AI*

# QML in HEP

Applied where "classical machine learning" has already been applied

- Classification:
  - ➡[1908.04480], [2002.09935], [2010.07335], [2012.11560], [2012.12177], [2103.12257], [2103.03897], [2104.07692], …
- Event reconstruction
  - ➡Pattern recognition, tracking : [2003.08126], [2007.06868], [2012.01379], [2109.12636], [2202.06874], [2204.06496] …
- Anomaly detection
  - ➡[2112.04958] , …
- Generative Models:
  - ➡[2101.11132], [2103.15470], [2110.06933], [2201.01547], [2203.03578], …
- Density Estimation:
  - ➡[2011.13934], …

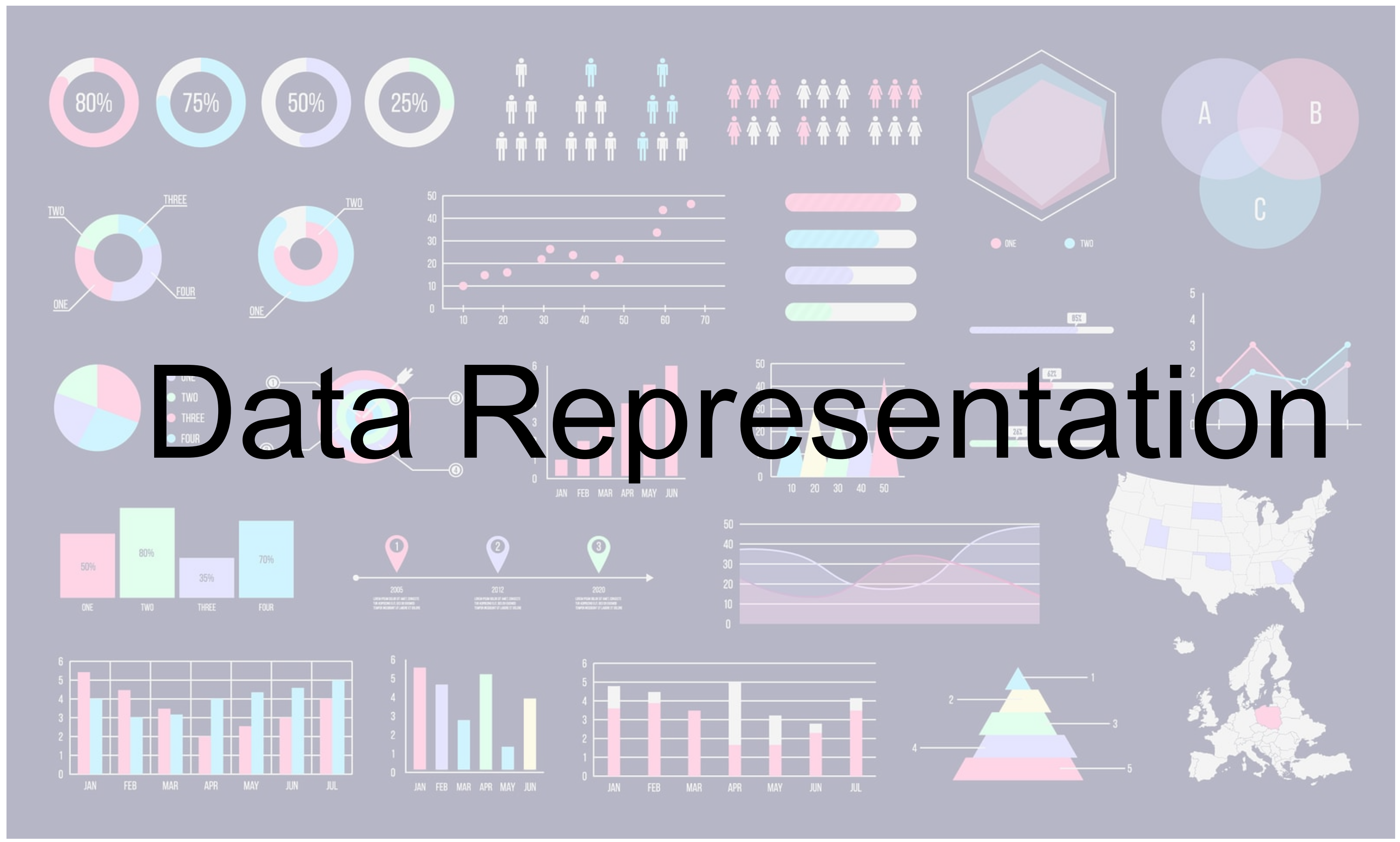
*Reference list might be incomplete, please let me know …*



# HEP-specific elements of AI

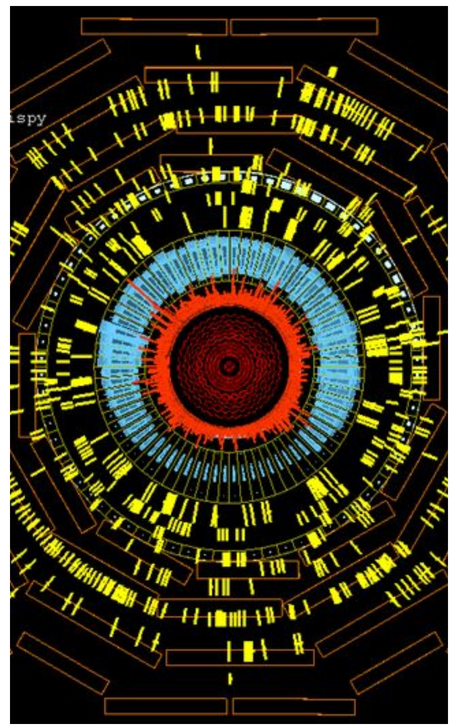
*Where innovation lies.*

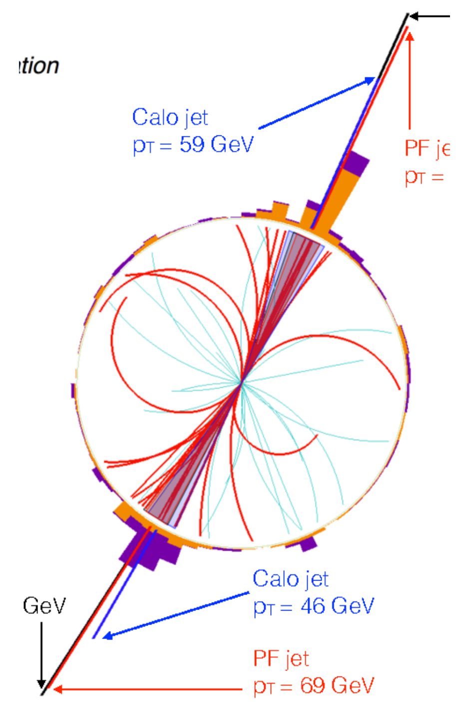
# Data Representation

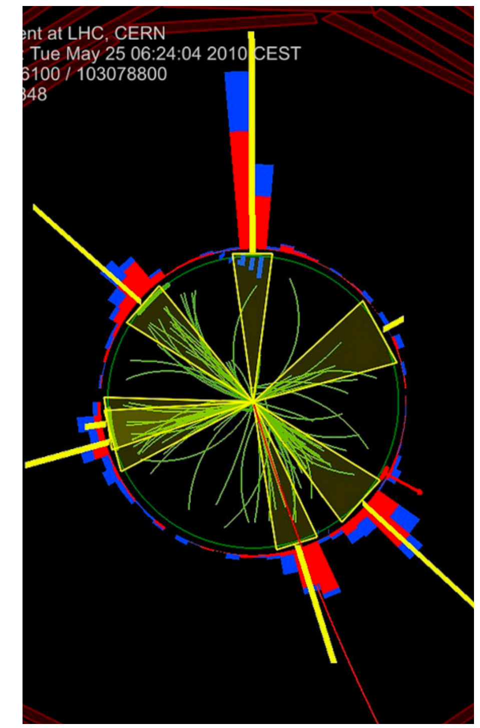


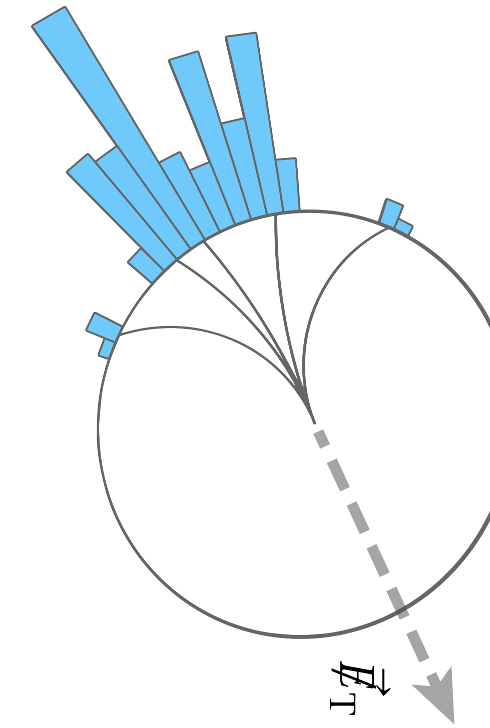
# From RAW to High Level Features

| Detector Data | Local reconstruction | Particle representation | Jet Clustering | High level features |
|---|---|---|---|---|






Event Processing

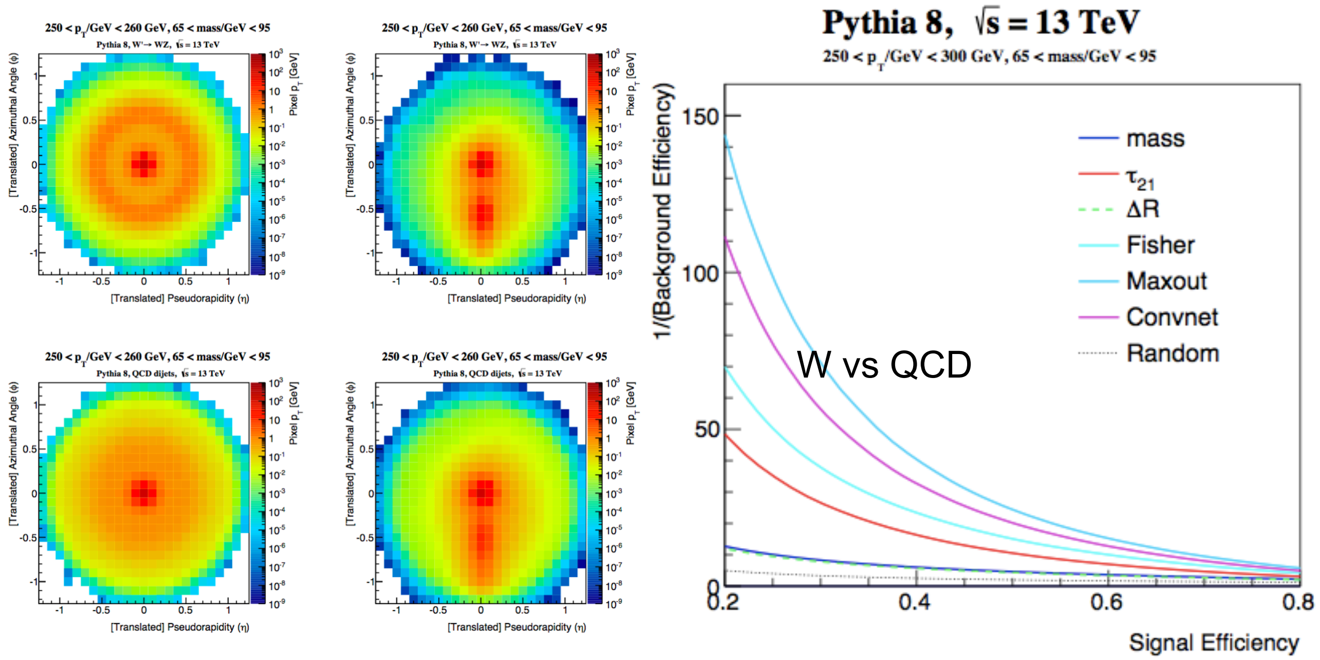
Dimensionality reduction

Globalization of information

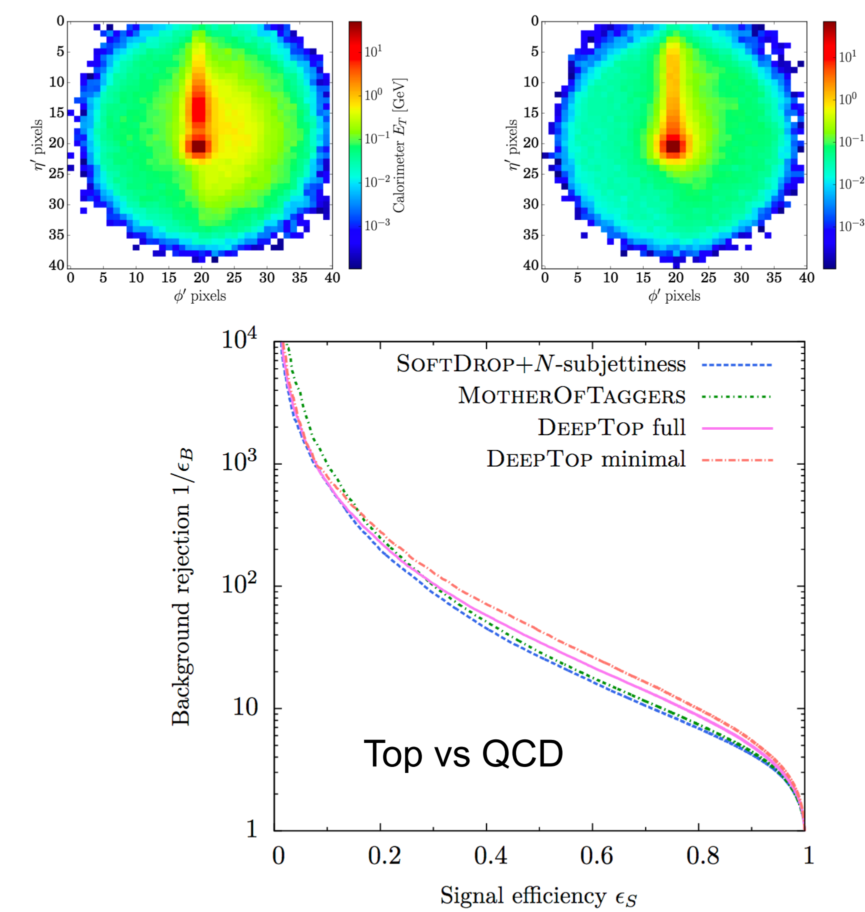
From digital signal, to local hits, to a sequence of objects, and high-level features.
Complex and computing intensive task that could find a match in ML application.

# Image Representation

Jet-Images – Deep learning edition
[1511.05190]

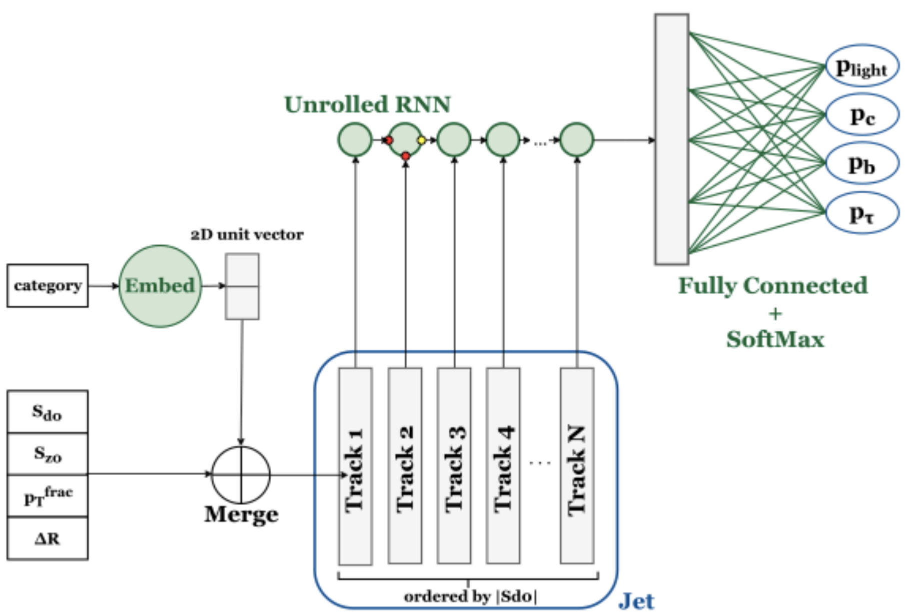



Deep-learning top taggers or the end of QCD?
[1701.08784]

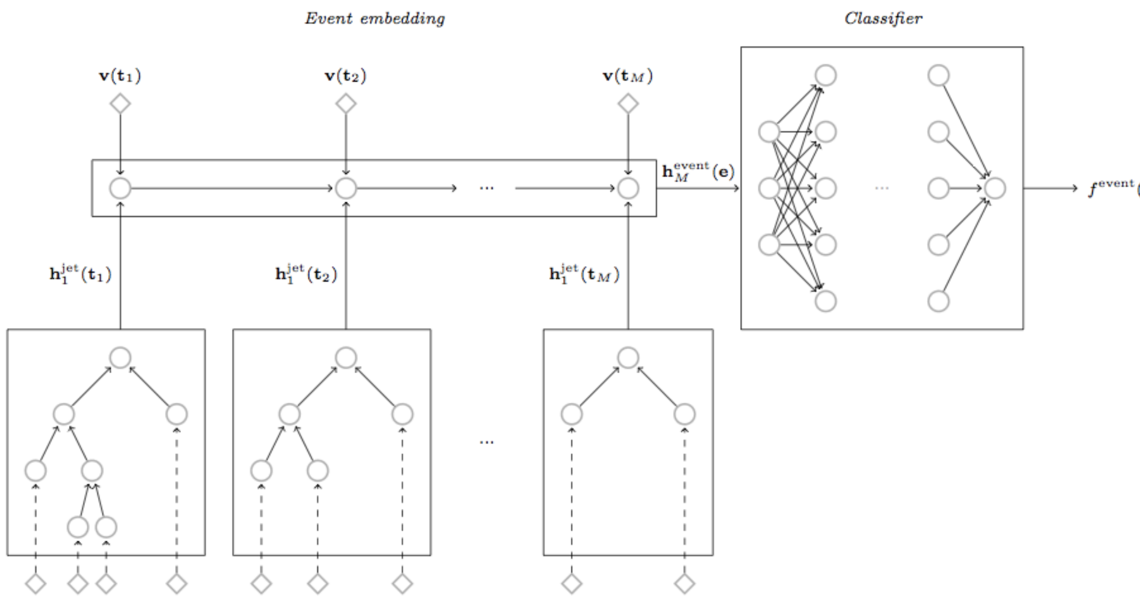



Calorimeter signal are image-like.
Projection of reconstructed particle properties onto images possible.
Potential loss of information during projection.

# Sequence Representation



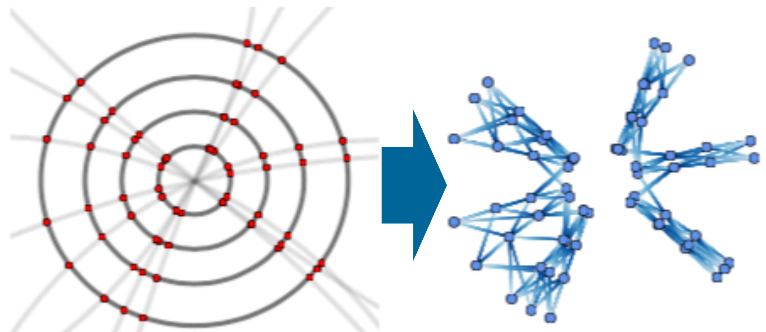

B-Jet with Recurrent Neural Networks
[cds:2255226]



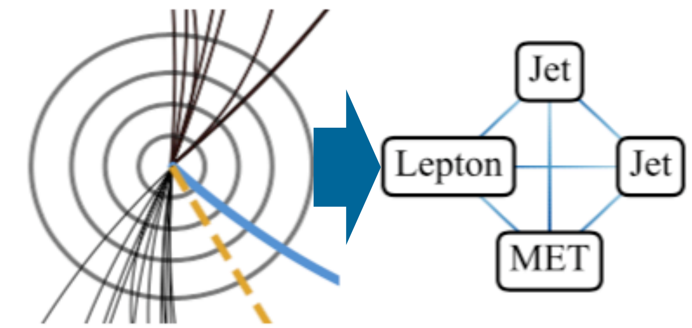

QCD-Aware Recursive Neural Networks for Jet Physics.
[1702.00748]

Somehow arbitrary choice on ordering with sequence representation.
Physics-inspired ordering as inductive bias.
Ordering can be learned too somehow.

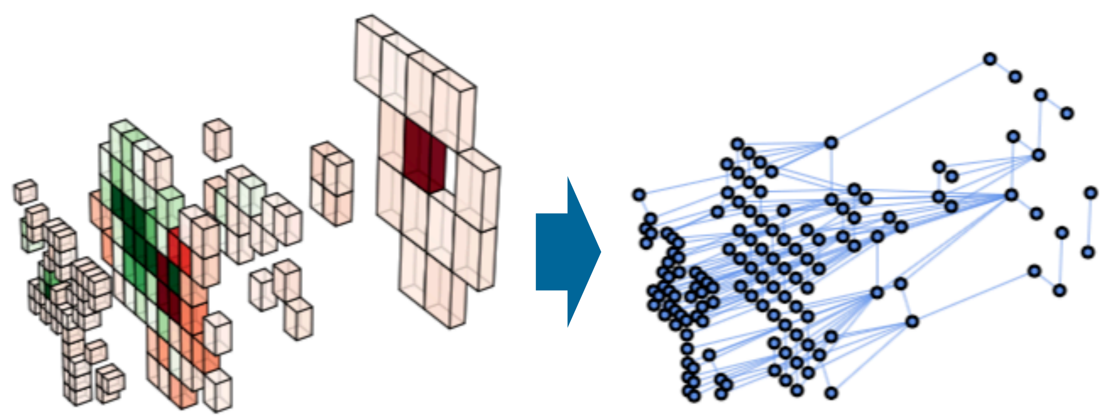
# Graph Representation



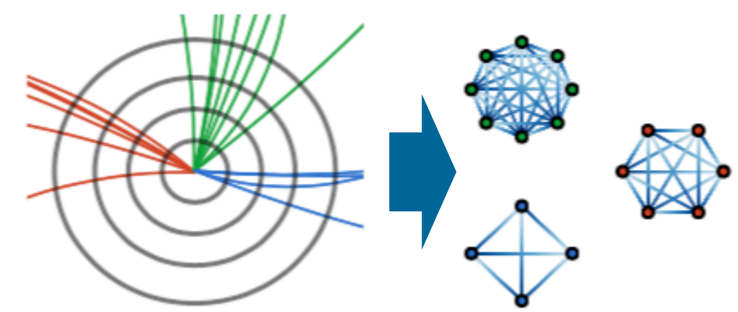
Hits in tracking detector




Objects in an event



Hits in calorimeter detector



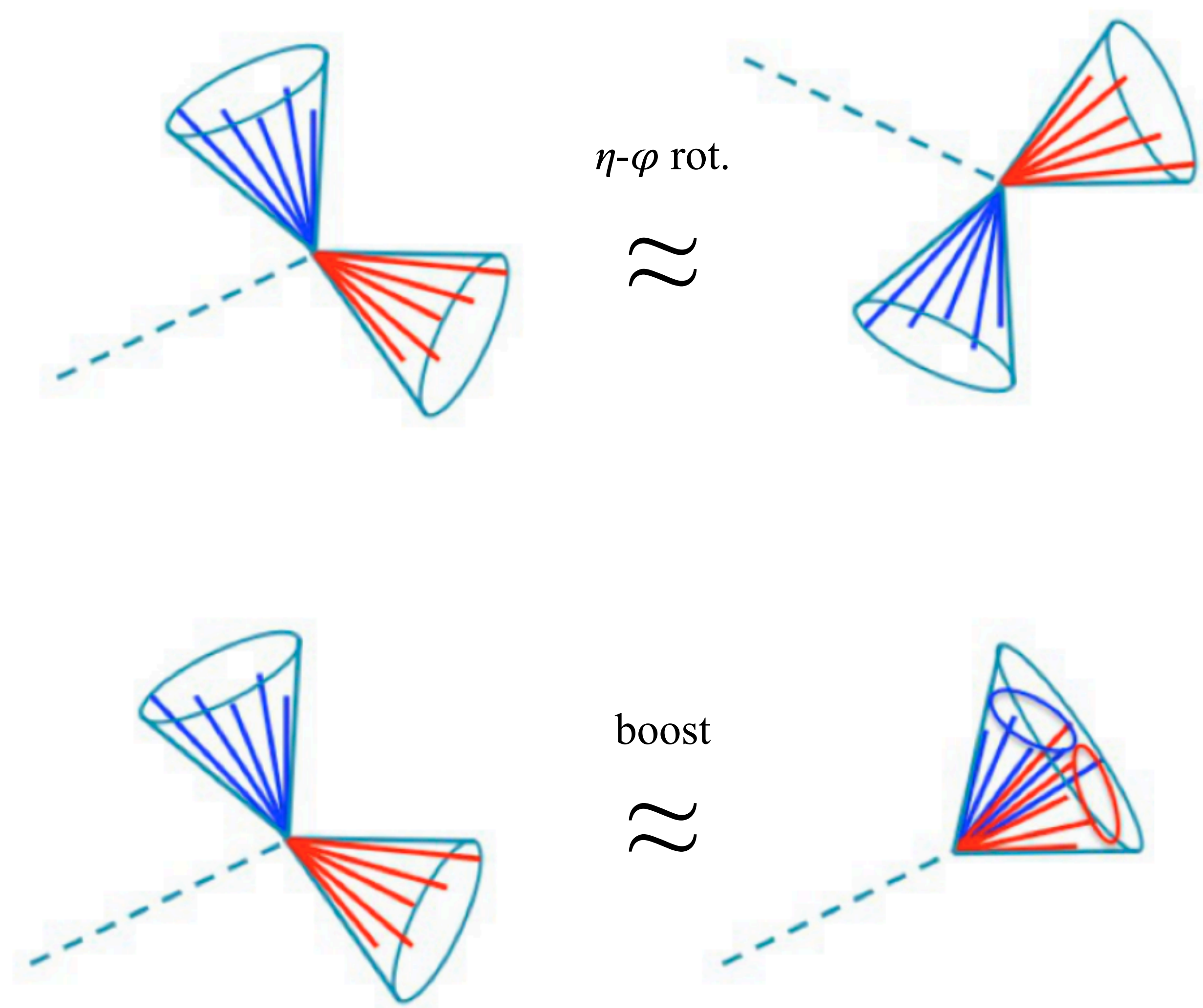
Object sub-structure in an event

Graph Neural Networks in Particle Physics
[2007.13681]

Heterogenous data fits well in graph/set representation.

# Invariance and Symmetries

# Dataset Degeneracy



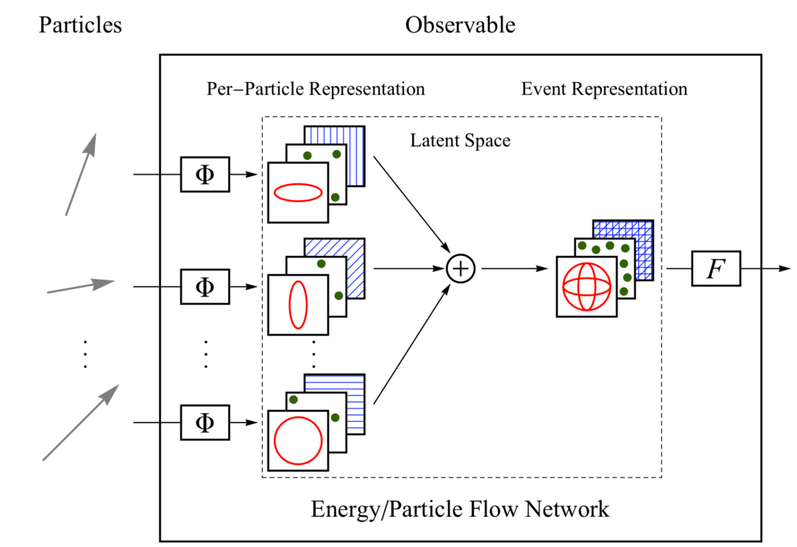

Pre-process the dataset to reduce degeneracy.
Model training improves as the invariance does not have to be learned.

# Inductive Bias

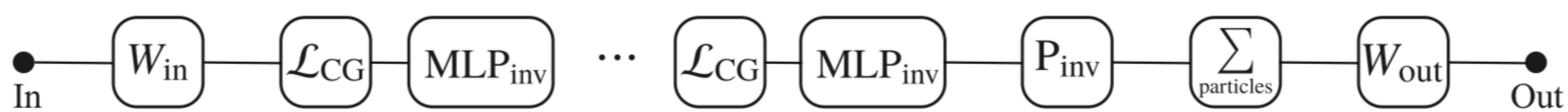
$$k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i} \begin{pmatrix} 1 & 0 & \cdots & 0 & C_{1,N+2} & \cdots & C_{1,M} \\ 0 & 1 & & \vdots & C_{2,N+2} & \cdots & C_{2,M} \\ \vdots & \vdots & \ddots & 0 & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & C_{N,N+2} & \cdots & C_{N,M} \end{pmatrix}$$

$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ w_{jm}^{(E)}\, E(\tilde{k}_m) \\ w_{jm}^{(d)}\, d_{jm}^2 \end{pmatrix}$$

Lorentz Learning Layer
[1707.08966]




Deep set
[1810.05165]

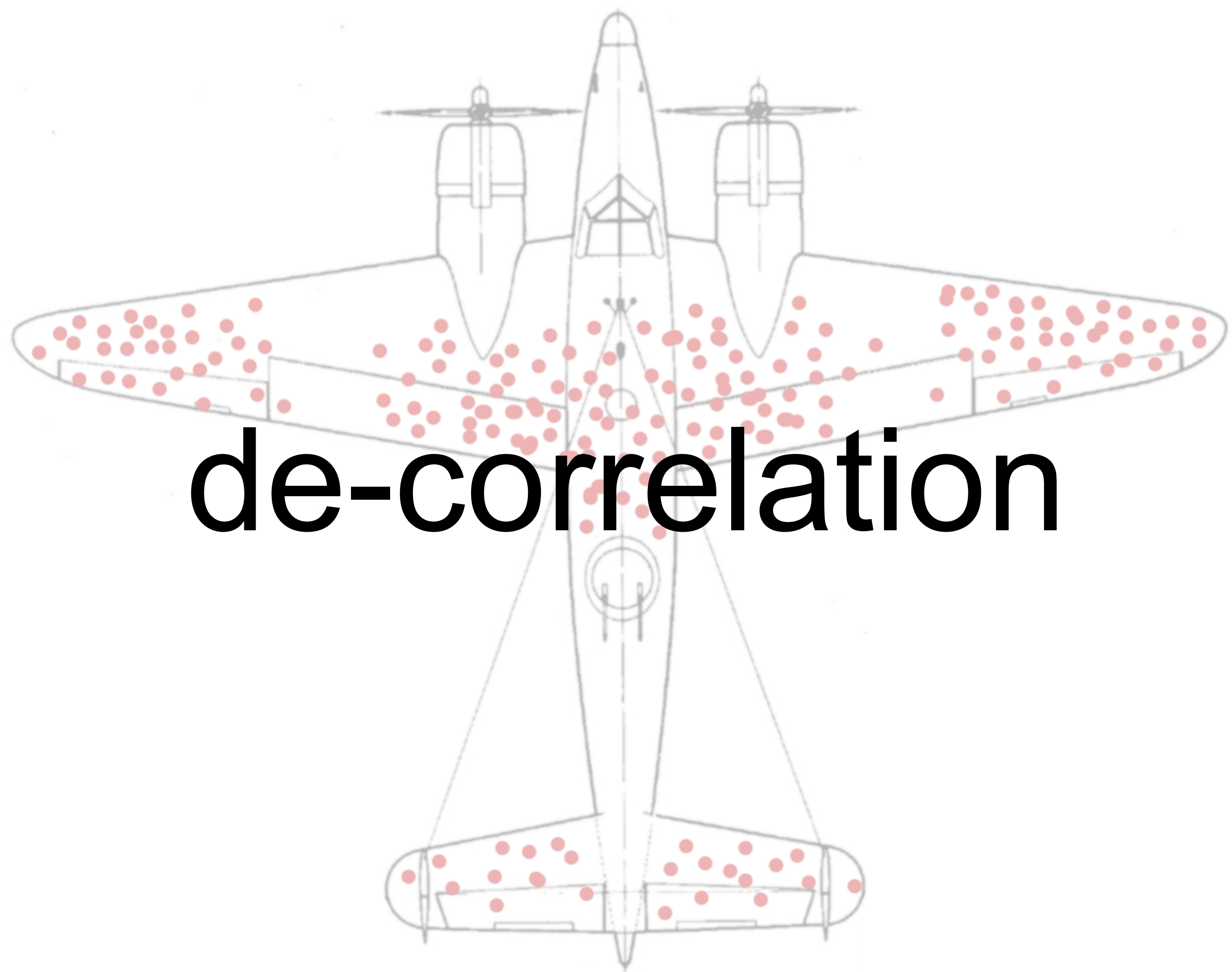



$$\mathscr{F}_i \mapsto W \cdot \left( \mathscr{F}_i \oplus \mathscr{F}_i^{\otimes 2} \oplus \sum_j f\left(p_{ij}^2\right) \cdot p_{ij} \otimes \mathscr{F}_j \right)$$

Lorentz group quivariant networks
[2006.04780]

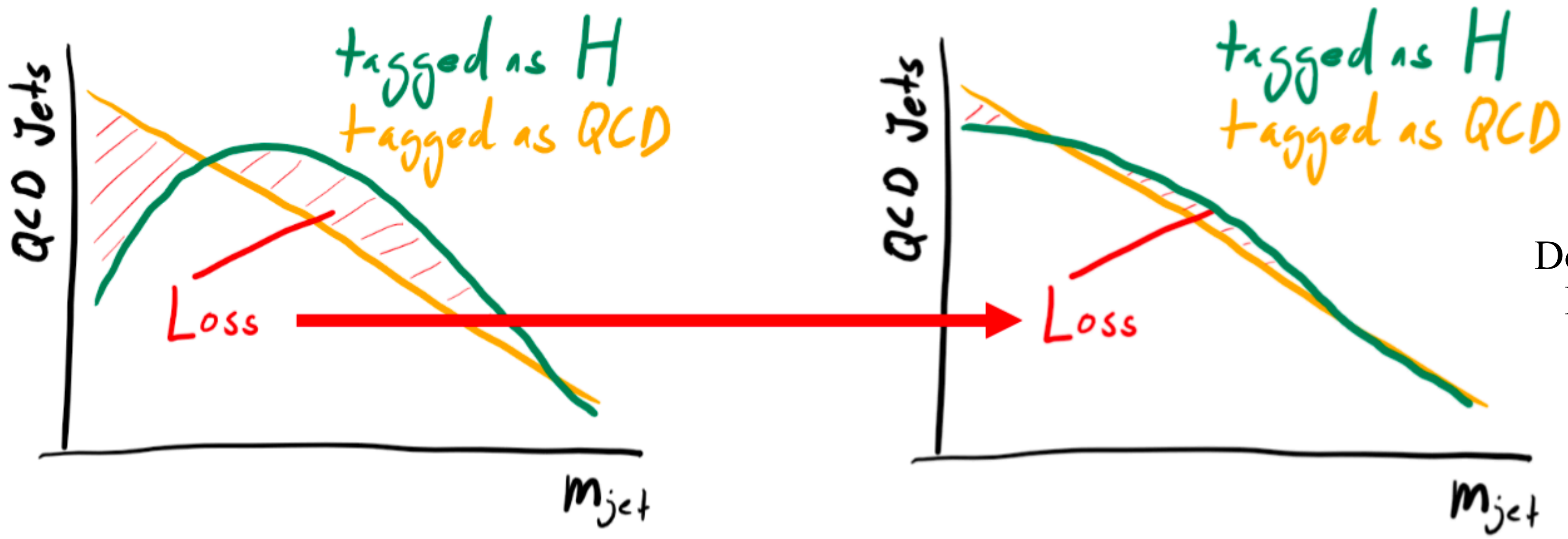
Embed the symmetry and invariance in the model.
Economy of model parameters.

# de-correlation



# De-correlation

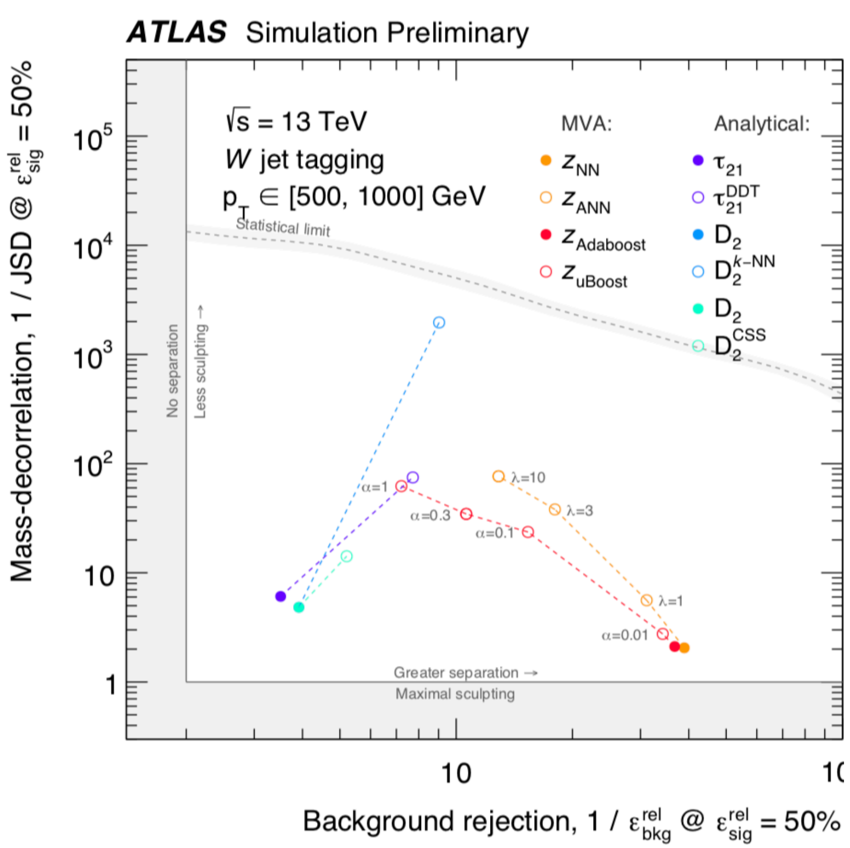
Most background estimation methods (side-bands, ABCD, parametrized fit, …) will require background shape to somehow be independent of analysis selections/processing (not only when using machine learning BTW).



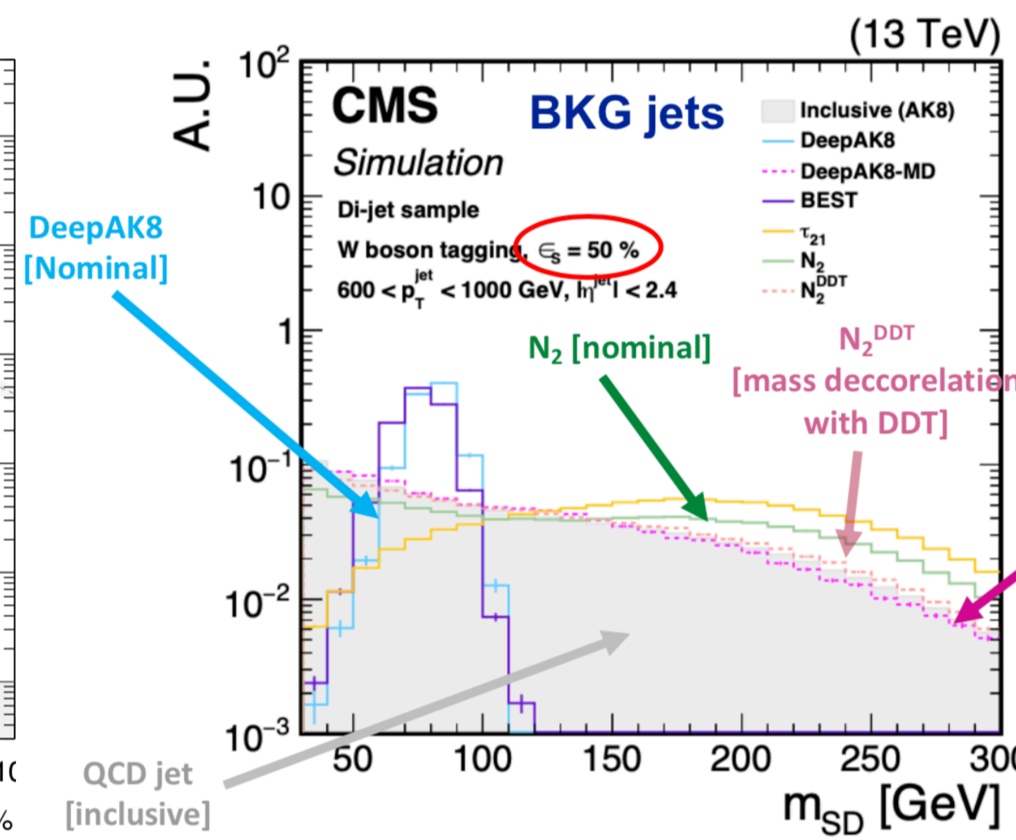

Domain adaptation [1409.7495]
Learn to Pivot [1611.01046]

Numerous methods proposed to de-correlate model predictions and quantities of interest ($p_T$, mass, … ).
Usually adding a term in the loss to constrain de-correlation.

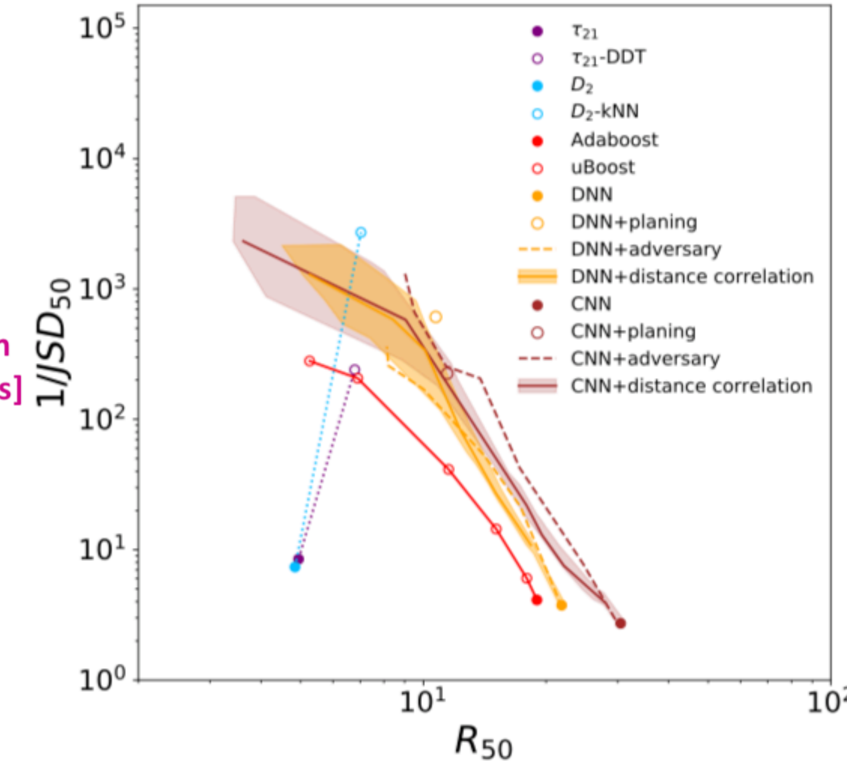
# Performance




ATLAS Collab. [cds:2630973]



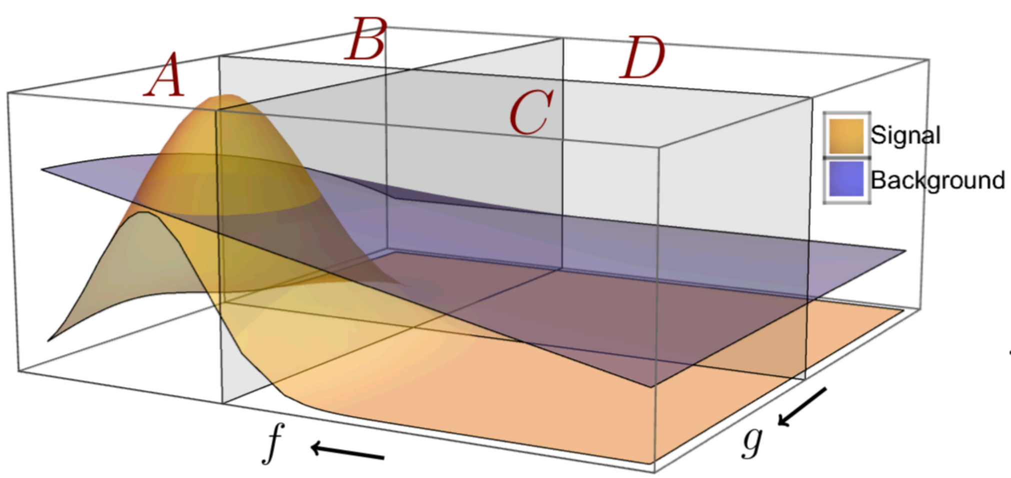

CMS Collab. [doi:10.1088/1748-0221/15/06/P06005]

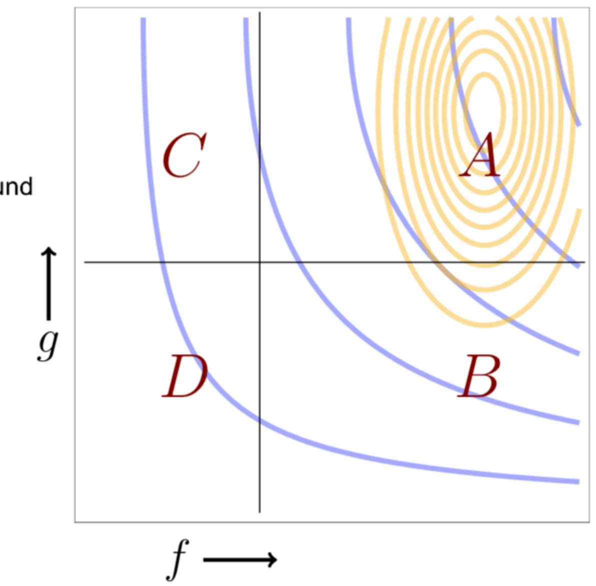



DISCO: Distance Correlation [2001.05310]

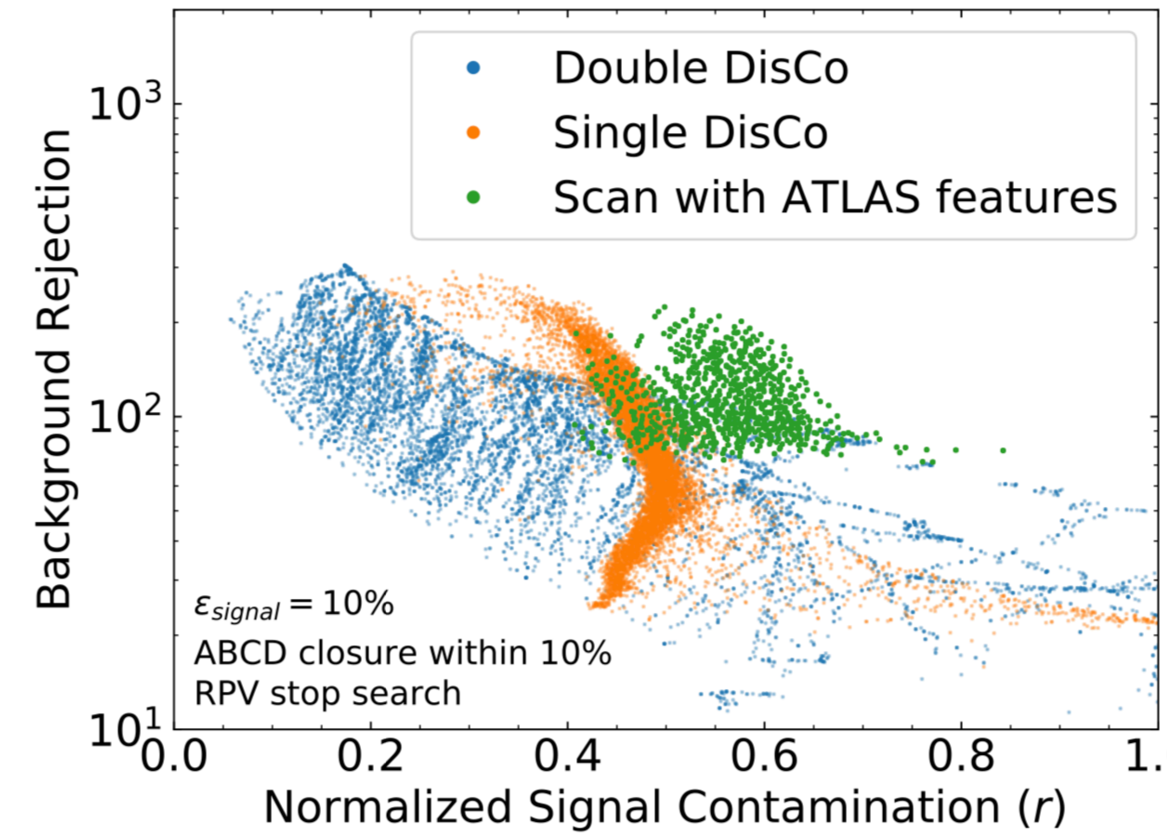
Jenson-Shannon Divergence (JSD) as the comparison metric for shaping.
Residual shaping needs to enter systematics uncertainty estimation.

# Background Estimation



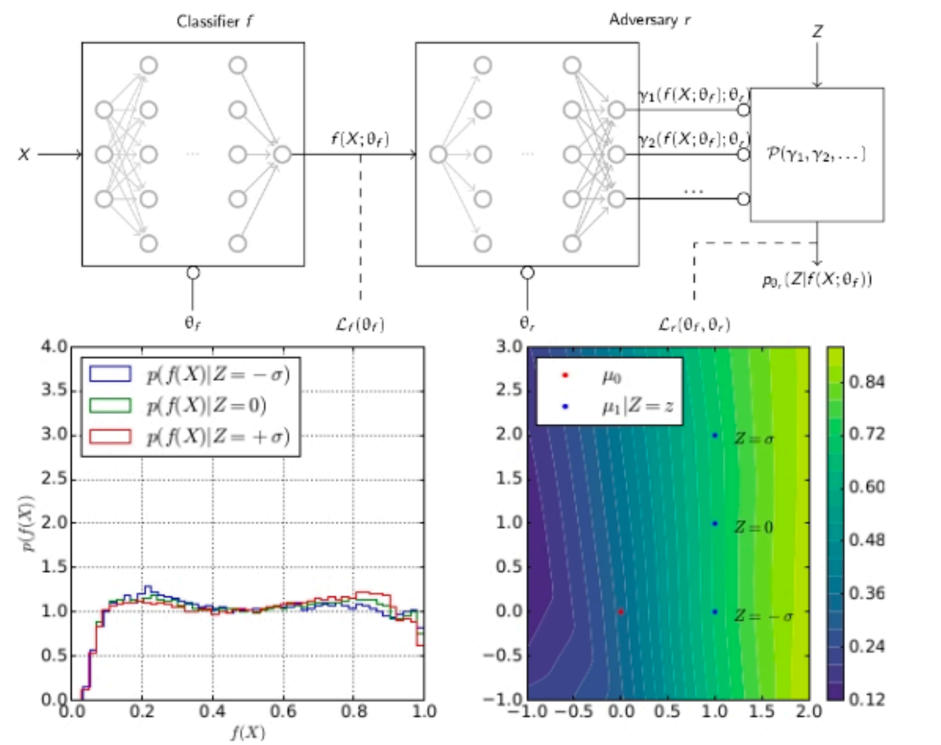

ABCD + Disco
[2007.14400]



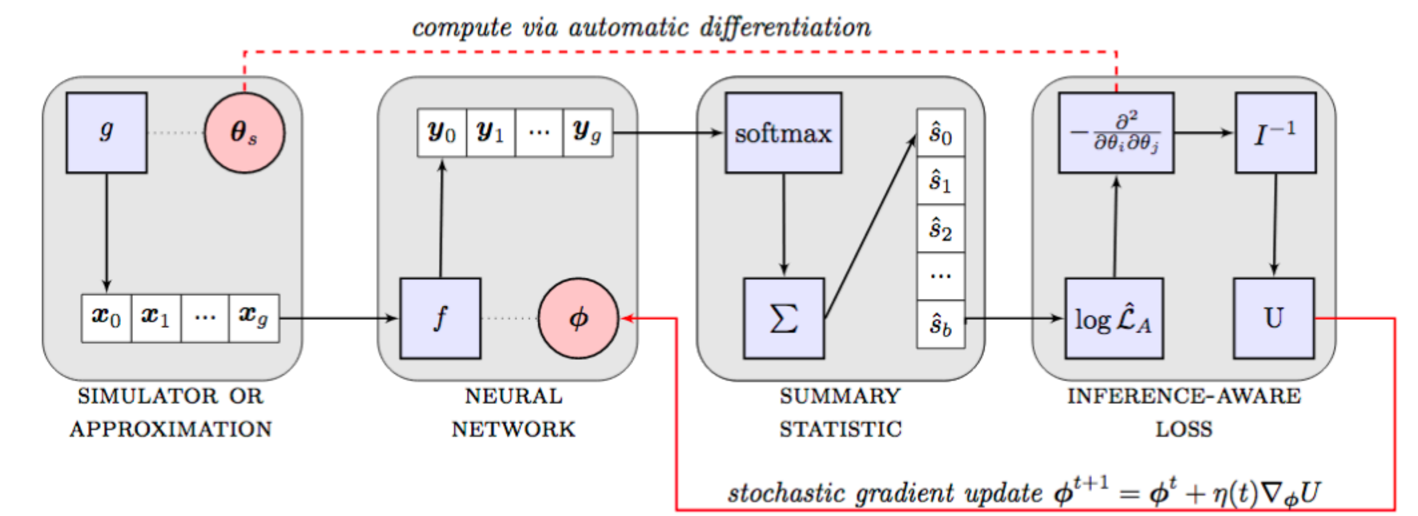

Most popular background estimation method (ABCD), can be optimized for de-correlation, yielding increased significance.

# Systematic Uncertainties

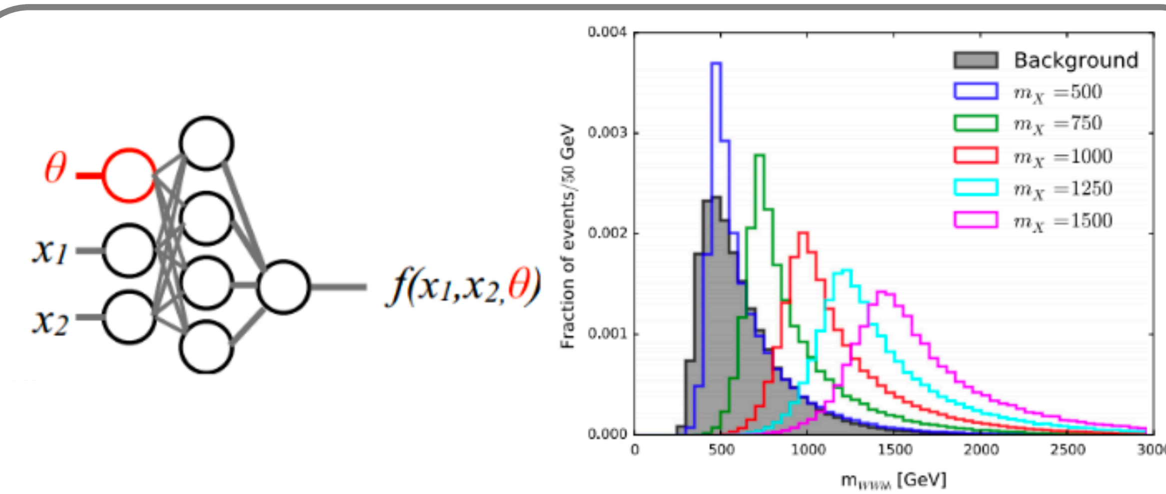
# Syst. Estimation and Mitigation




Learn to pivot [1611.01046]




INFERNO: Inference-Aware Optimisation [1806.04743]

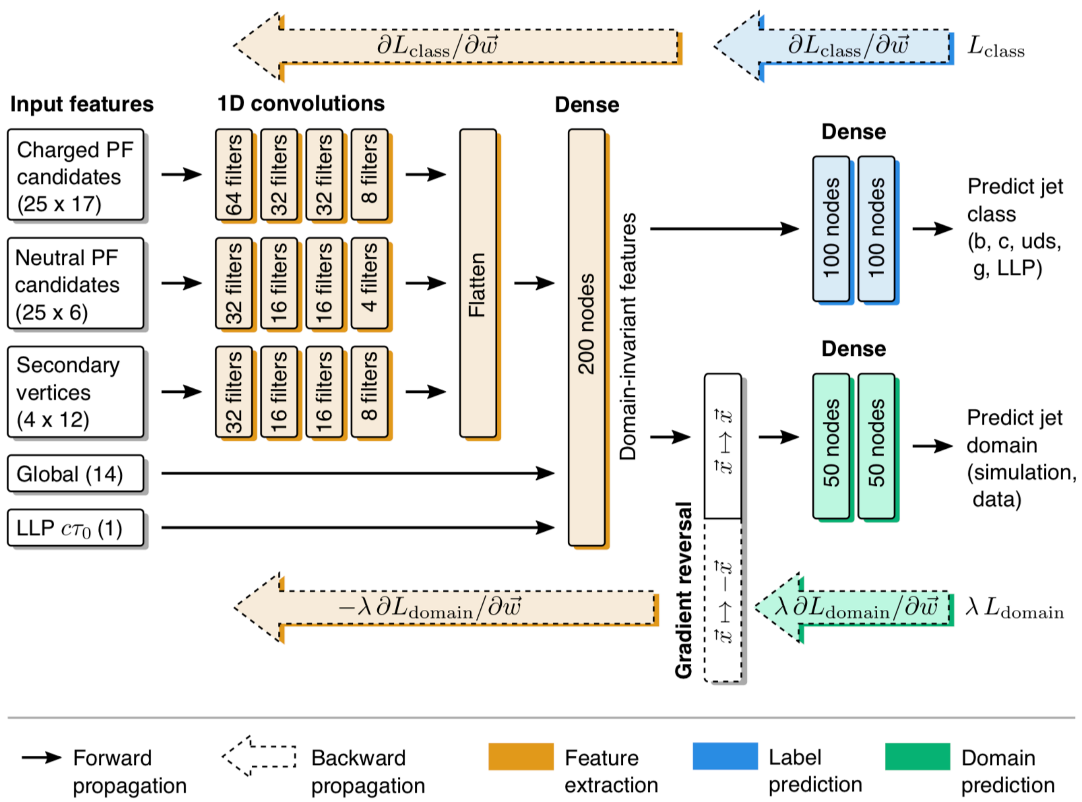



Parametrized Learning [1601.07913]

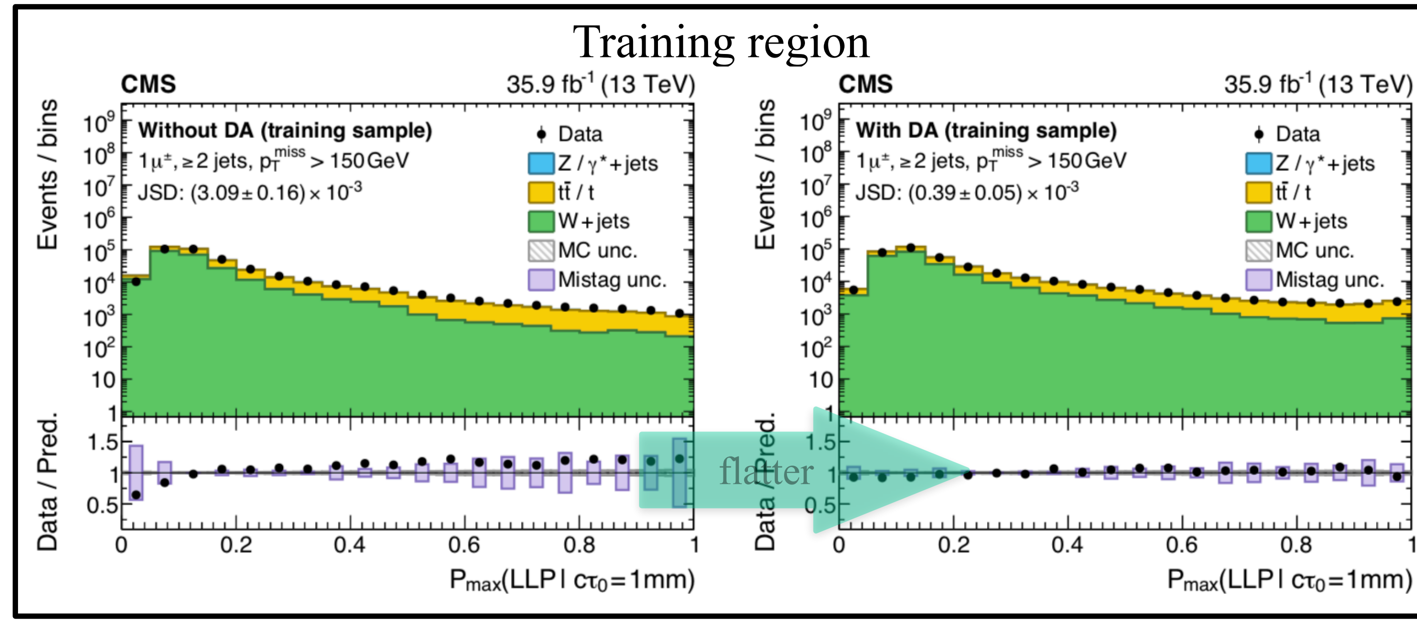
Systematic uncertainties can be propagated the usual ways.
No additional systematic from the model itself.
Methods to mitigate, propagate and optimize against systematic uncertainties.

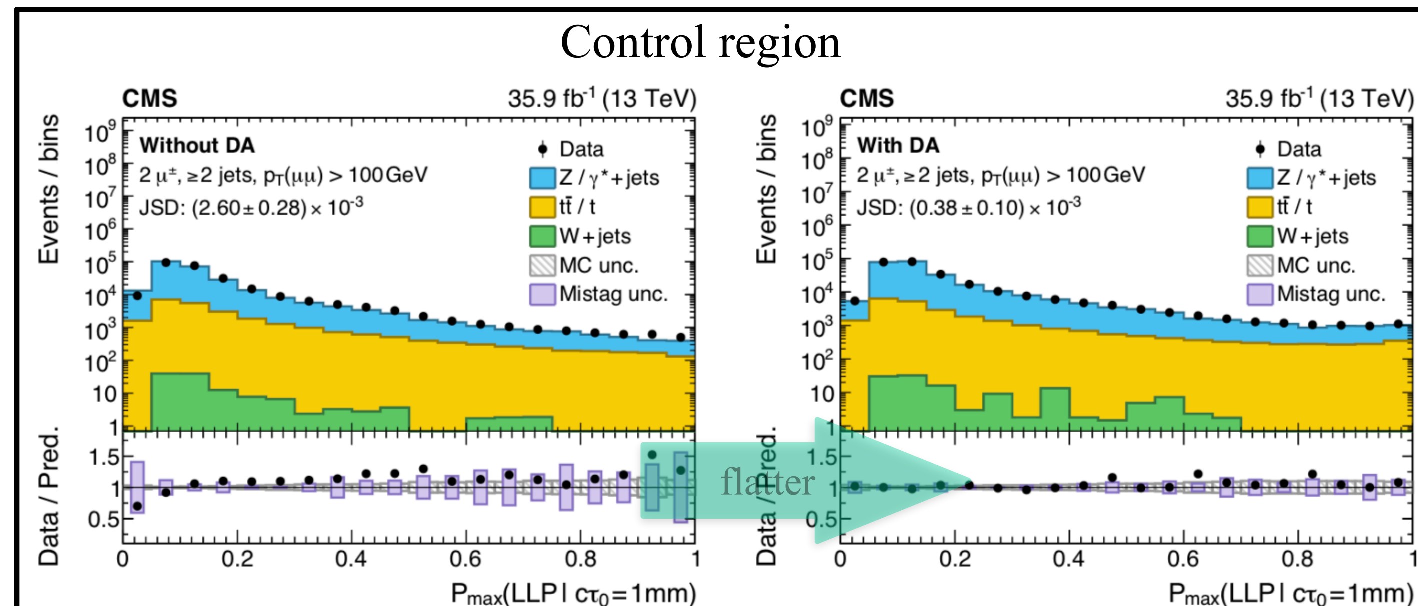
# Domain Dependence

# Domain in-Dependence



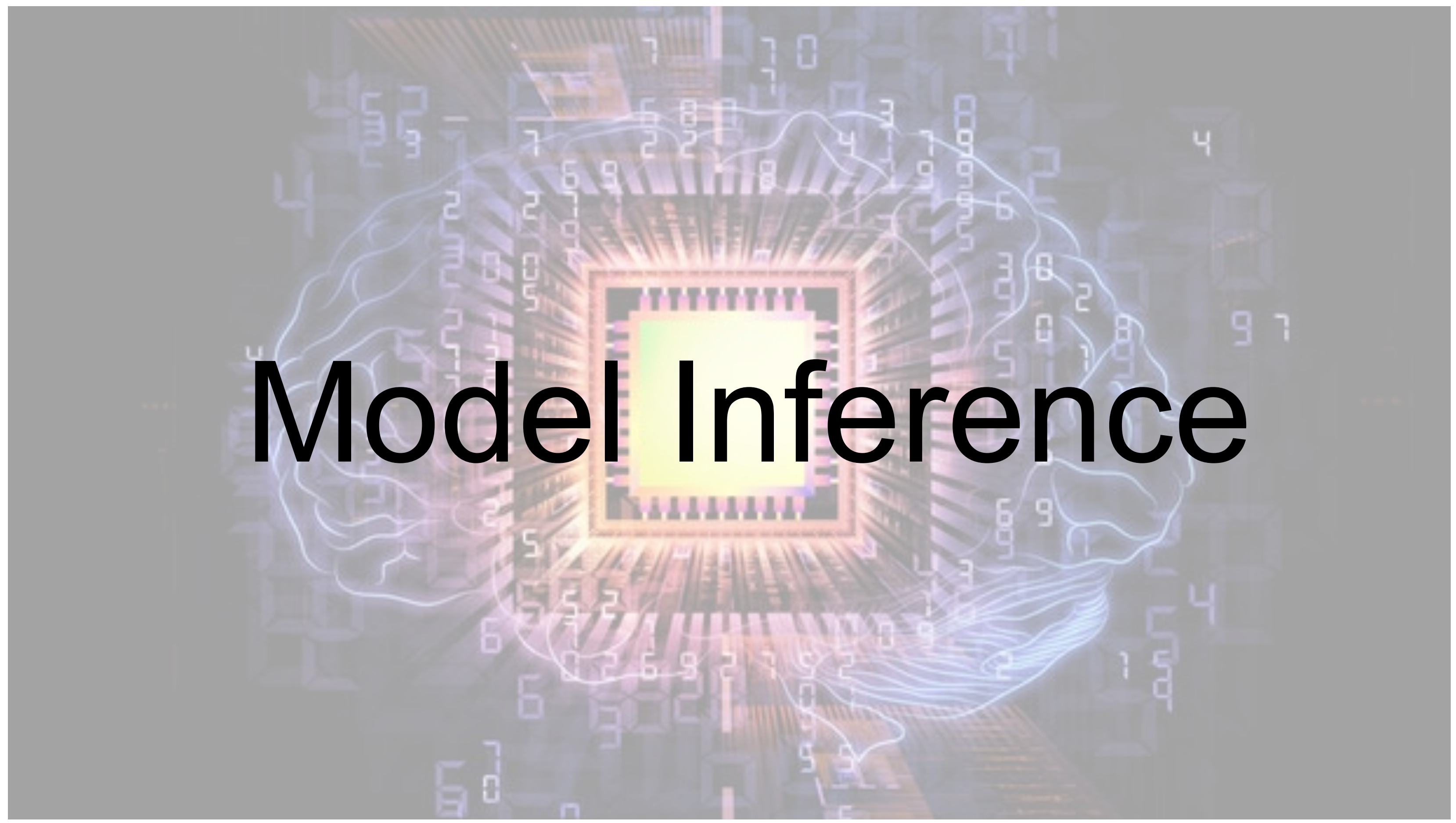

LLP jet tagger
[doi:10.1088/2632-2153/ab9023]




Gradient reversal on a domain-classifier to mitigate the discrepancies of classifier output between data and simulation.

# Model Inference



# Inference Engines

"On-Board accelerator"

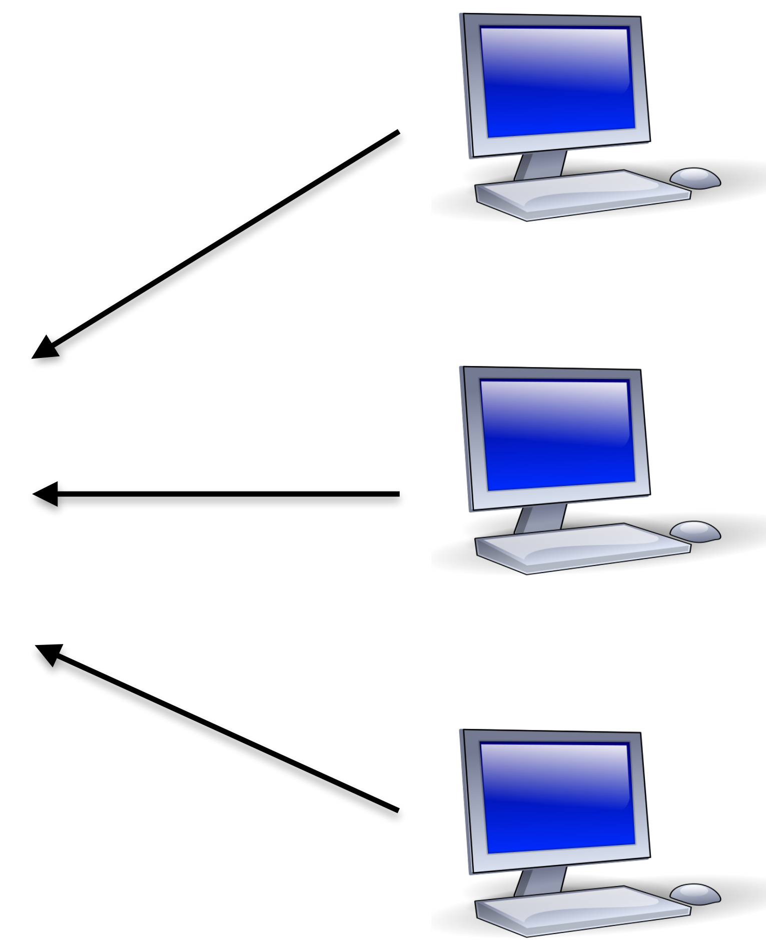
**CPU**
- Small models
- Small datasets
- Useful for design space exploration

**GPU**
- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL

**TPU**
- Matrix computations
- Dense vector processing
- No custom TensorFlow operations

**FPGA**
- Large datasets, models
- Compute intensive applications
- High performance, high perf./cost ratio

"Remote accelerator"

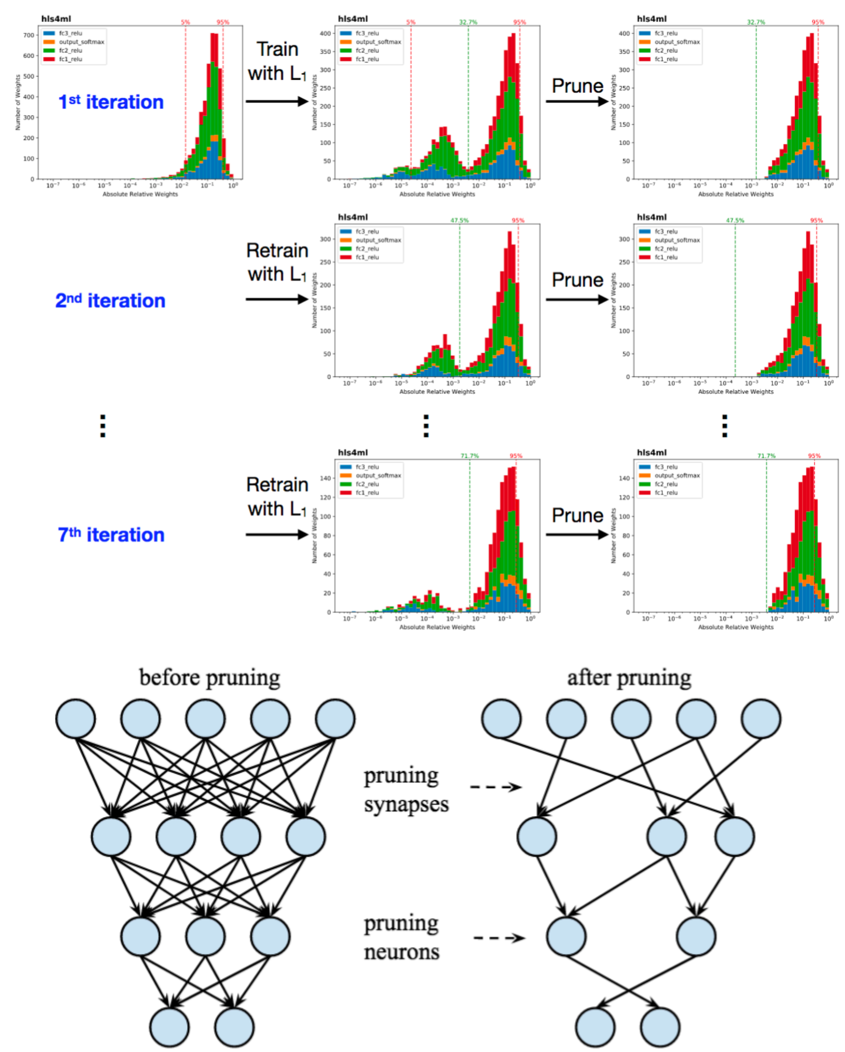


https://arxiv.org/abs/1811.04492
https://arxiv.org/abs/2007.10359
https://arxiv.org/abs/2007.14781

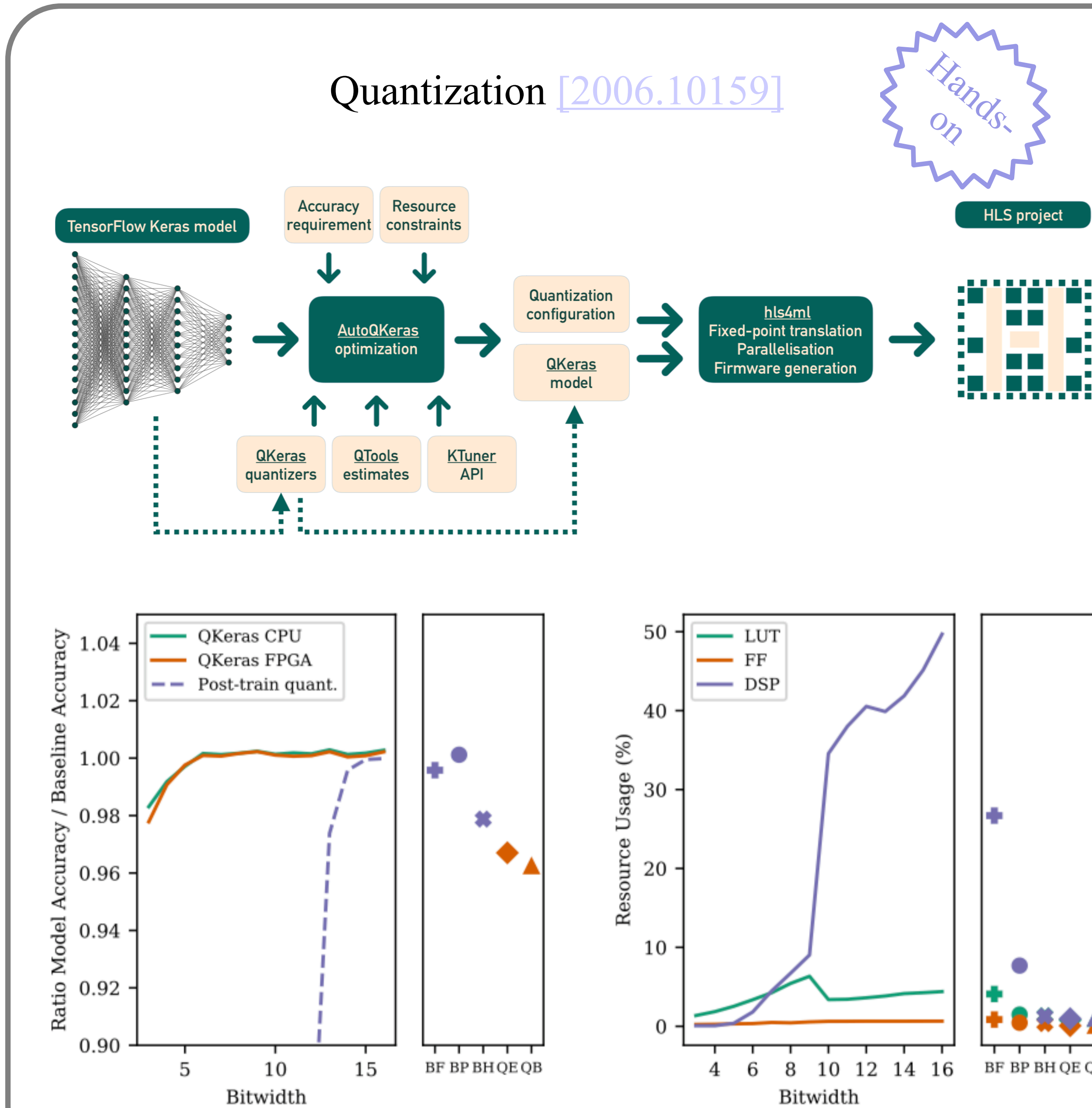
Growing list of deep learning accelerators.
Location of the device is driven by the environment (HLT, Grid, … ).

# Model Compression

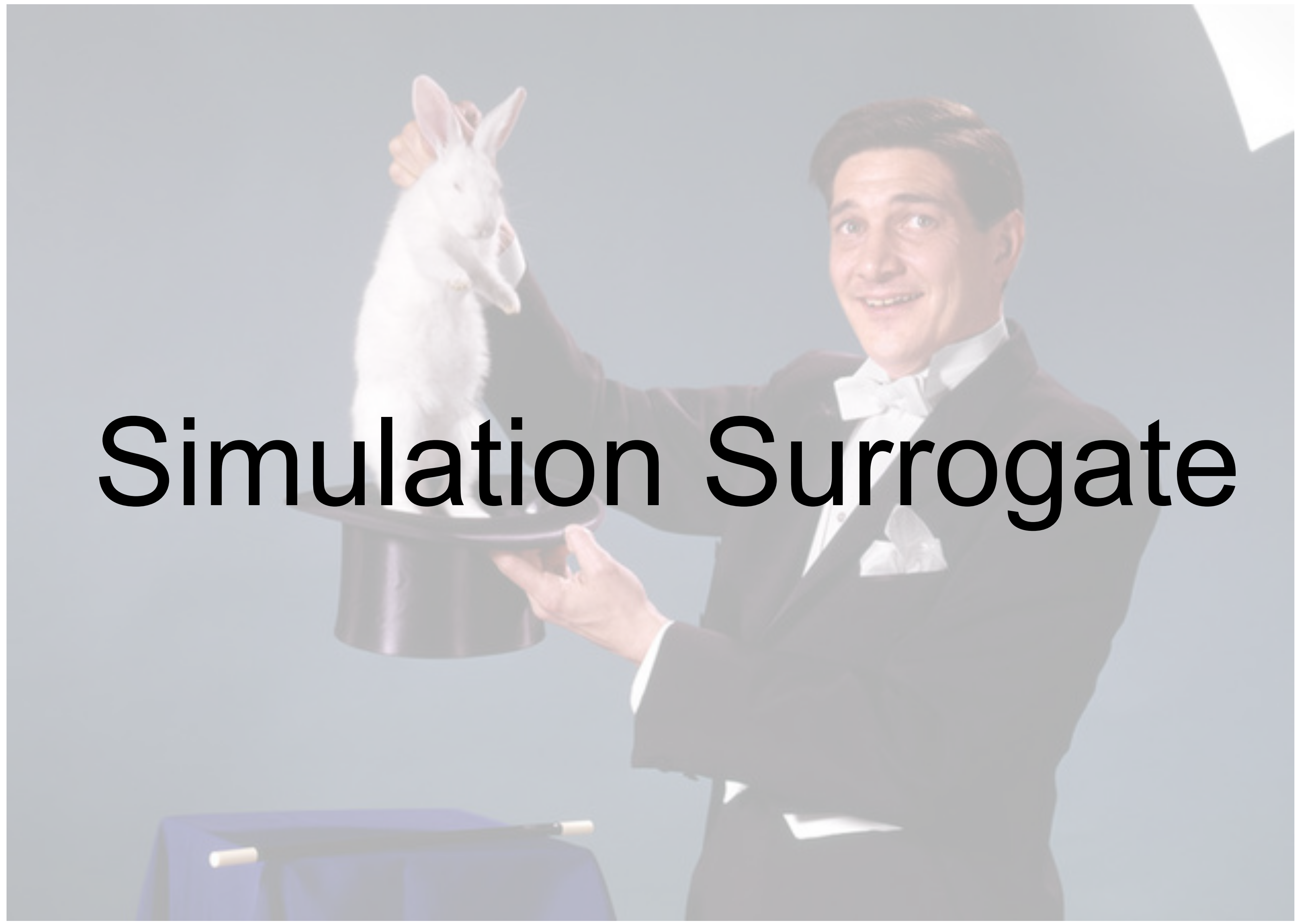
Pruning weights [1804.06913]




Quantization [2006.10159]

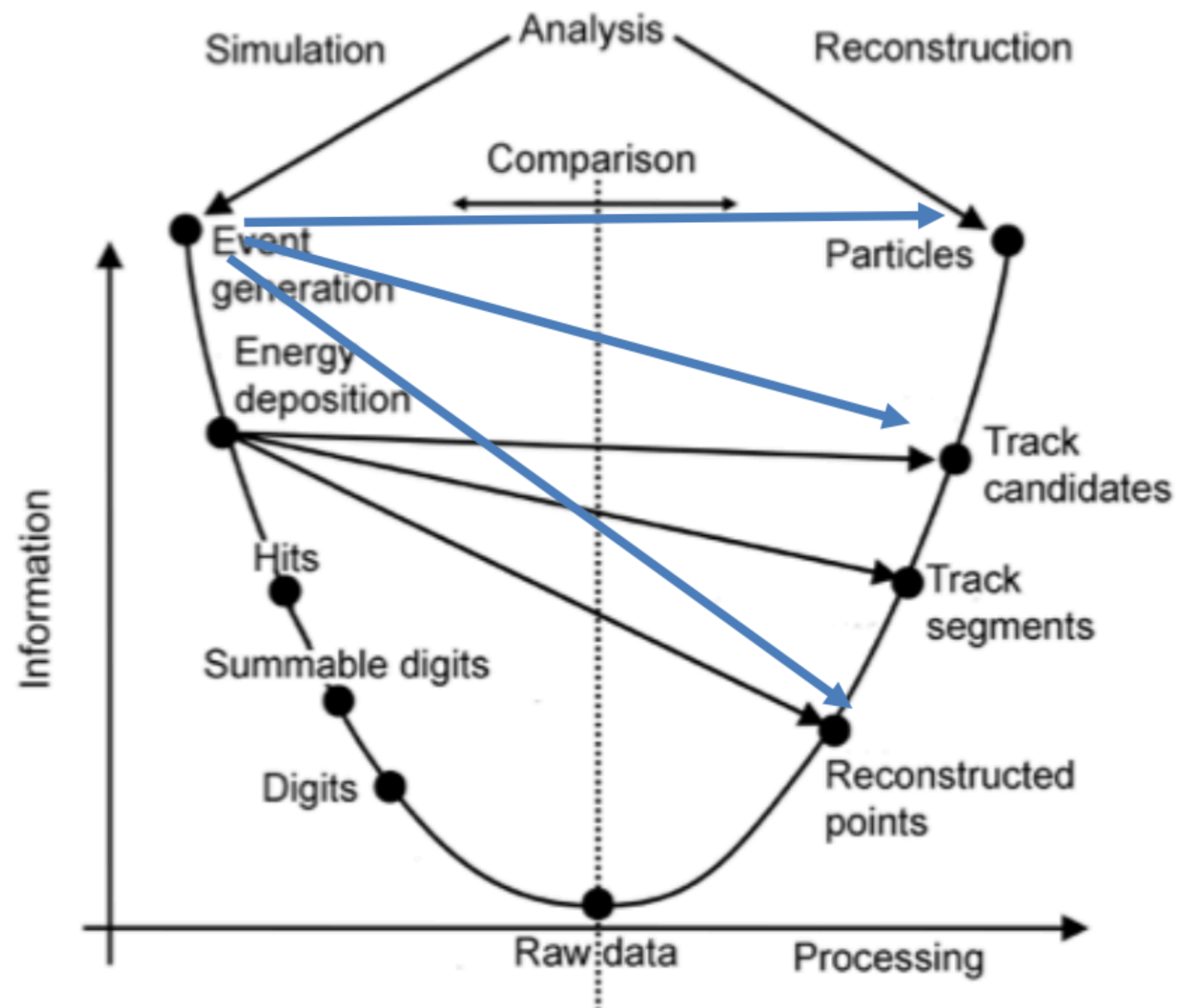



Model inference can be accelerated by reducing the number and size of operations.



# Simulation Surrogate

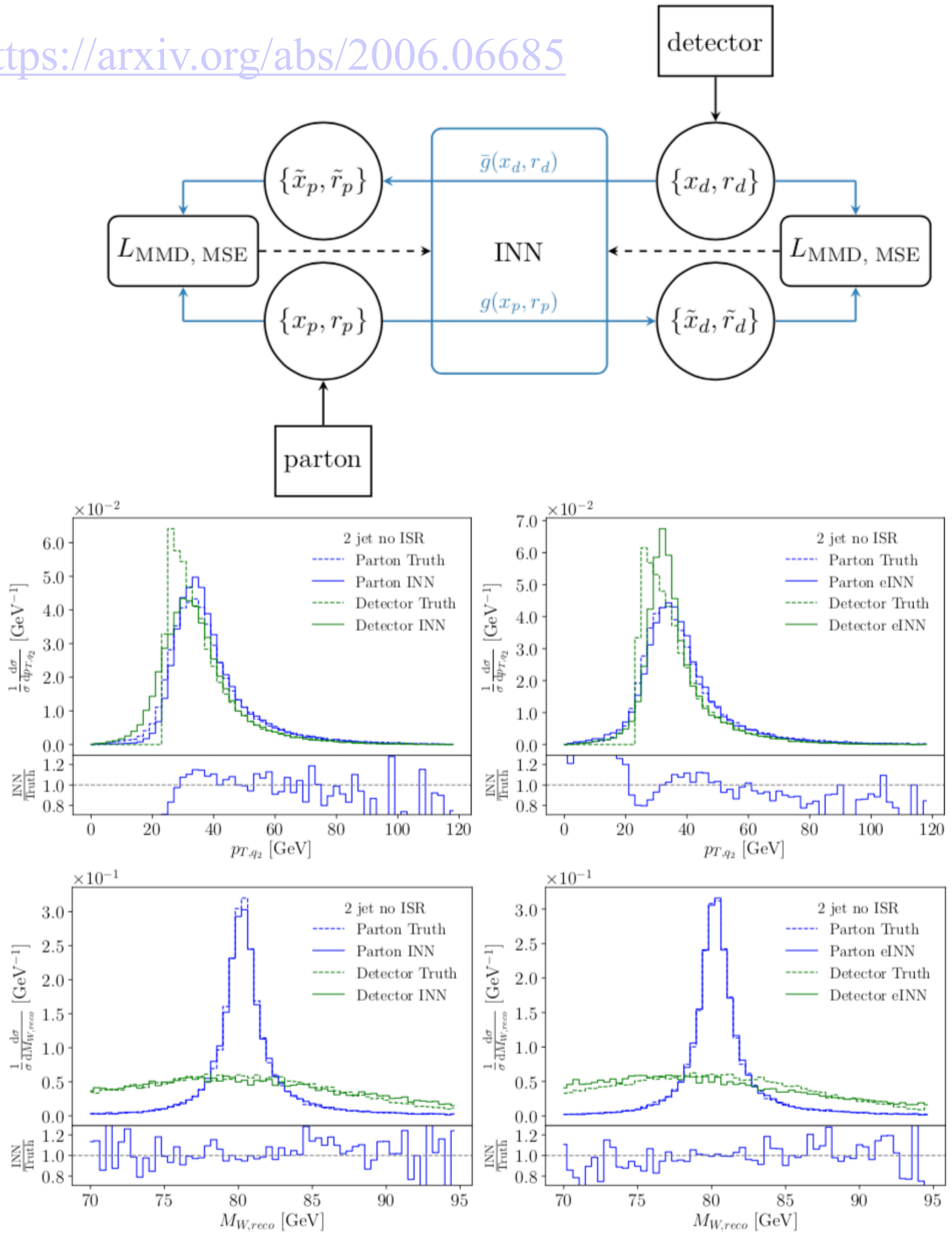
# Reconstruction ∘ Simulation ~ Identity



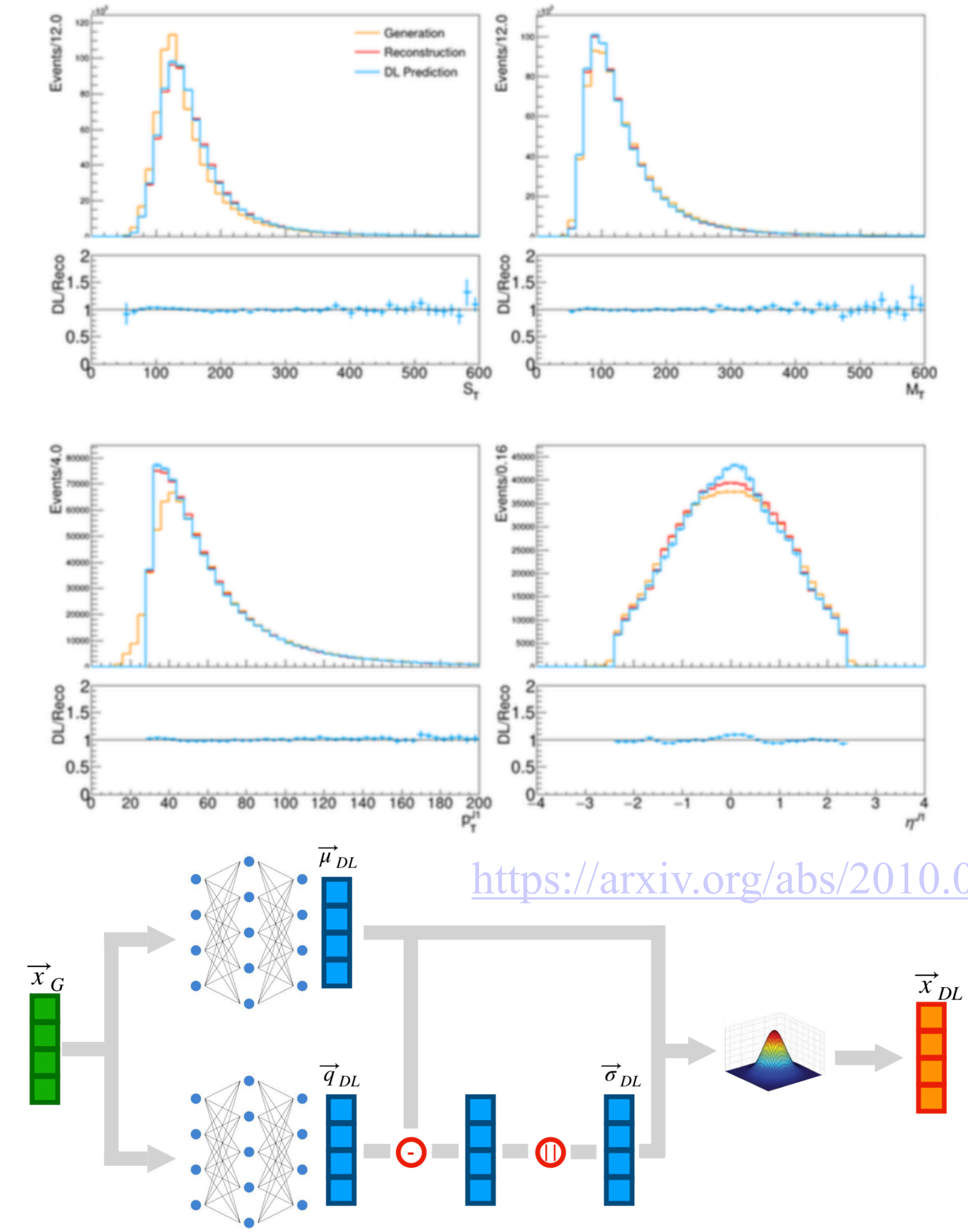

Simulation aims at predicting the outcome of collisions.
Reconstruction aims at inverting it.
Multiple ways to connect intermediate steps with deep learning.

# Suiting Models

https://arxiv.org/abs/2006.06685

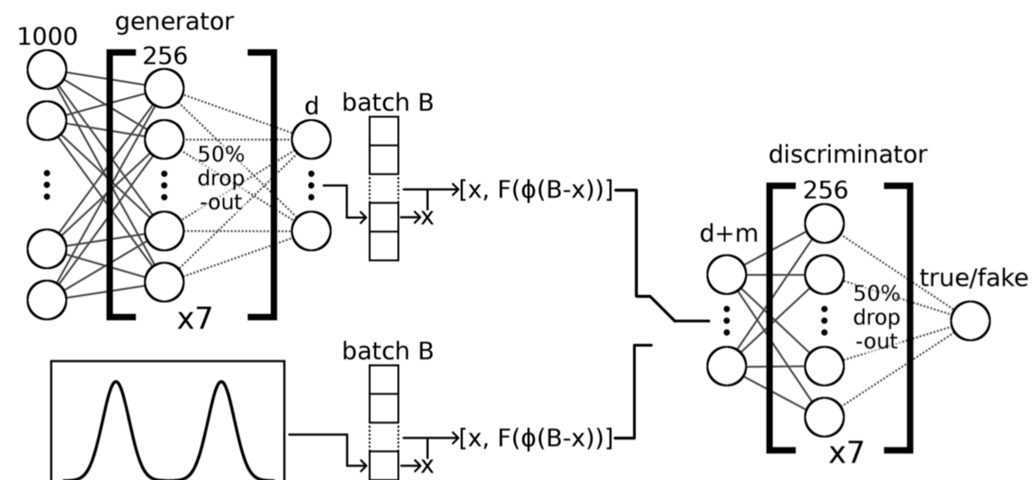



https://arxiv.org/abs/2010.01835

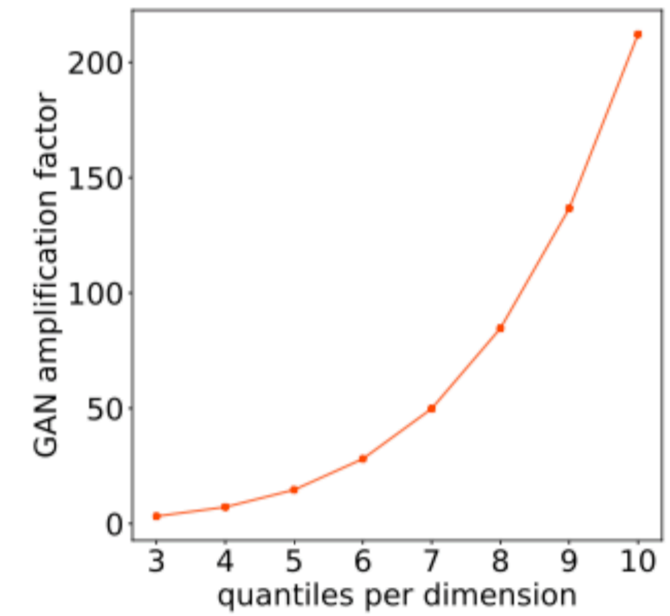


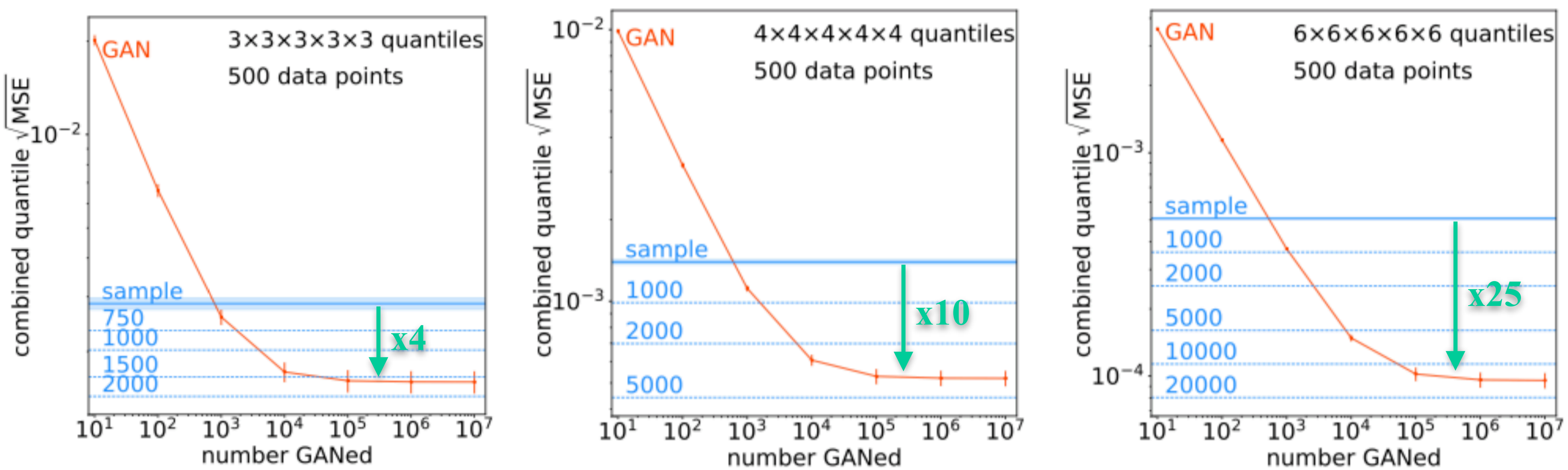

Learn the parton⇛detector function instead of generating samples from vacuum.

# Statistical Power




https://arxiv.org/abs/2008.06545



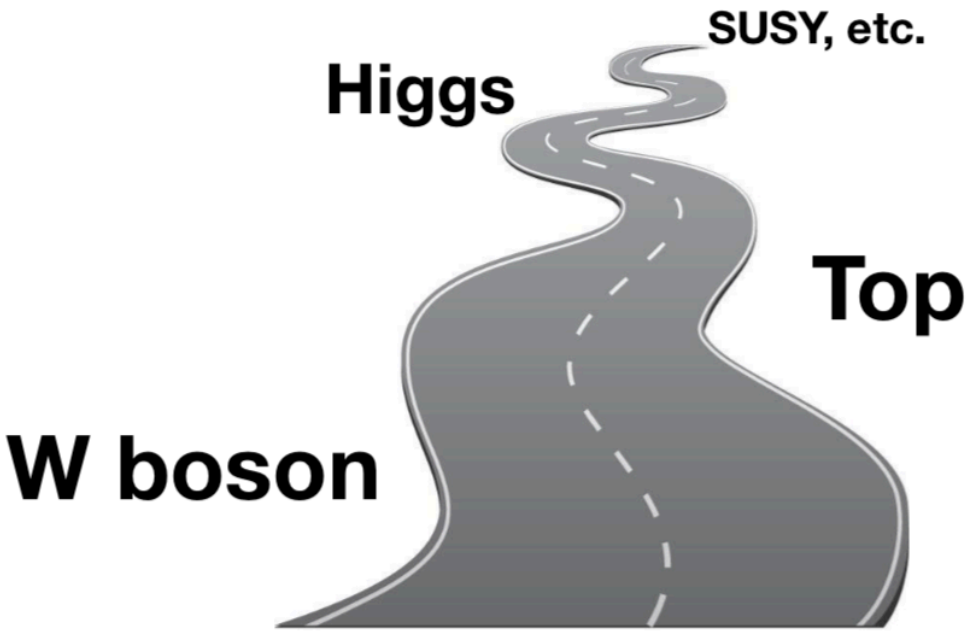

Generative adversarial network may help producing samples with higher statistical power than the one used for training.

# Anomaly Search

# The Sea Beyond Standard Model

Slide: A. Wulzner [H&N]

## HEP yesterday




“Almost” **Simple $H_1$**

Focus on **few sharply-defined** alternative models (e.g., the Higgs)

Case-by-case design of **optimal test**

## HEP today


“Very” **Composite $H_1$**

**Huge set** of alternatives

Case-by-case optimisation **unfeasible**

The **right $H_1$** likely **not yet formulated**

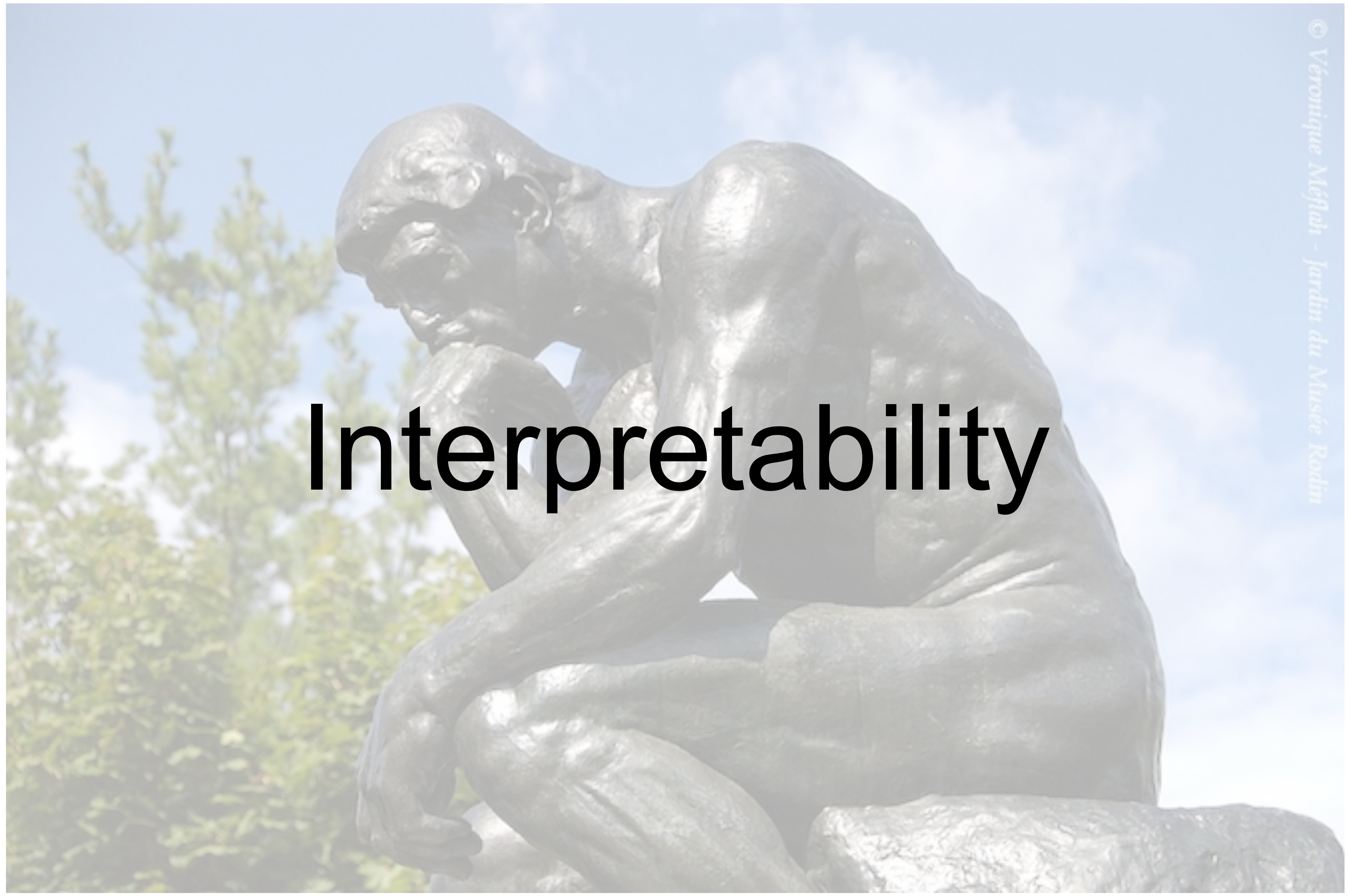
# “One-Sided” Hypothesis Testing

- Rigor in calibrating the rate of anomaly is HEP specific (Anomaly detection is not).
- Some methods can serve as a hotline: notification of odd signals.
- Some methods can serve in analysis: calibrated rate of novelty.
- Also of great importance in data quality monitoring/certification.

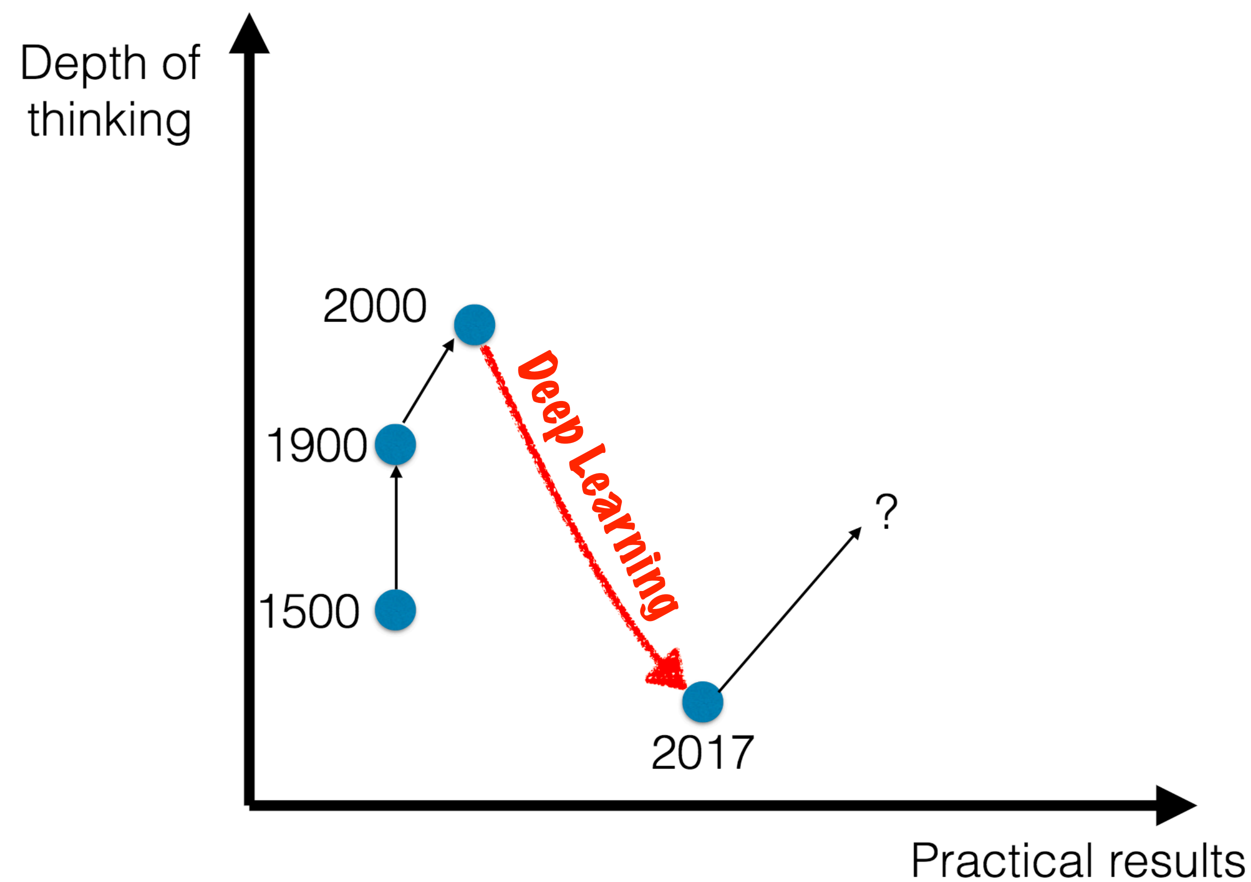
**Individual Approaches**

LHC Olympics 2020 [2101.08320]

# Interpretability



# The Black-box Dilemma



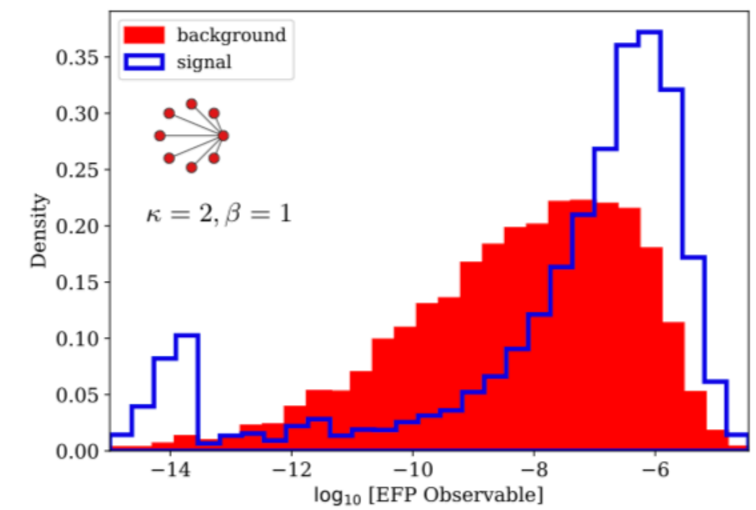

Deep learning may yield great improvements.
Having the "best classification performance" is not always sufficient.
Forming an understand of the processes at play is often crucial.

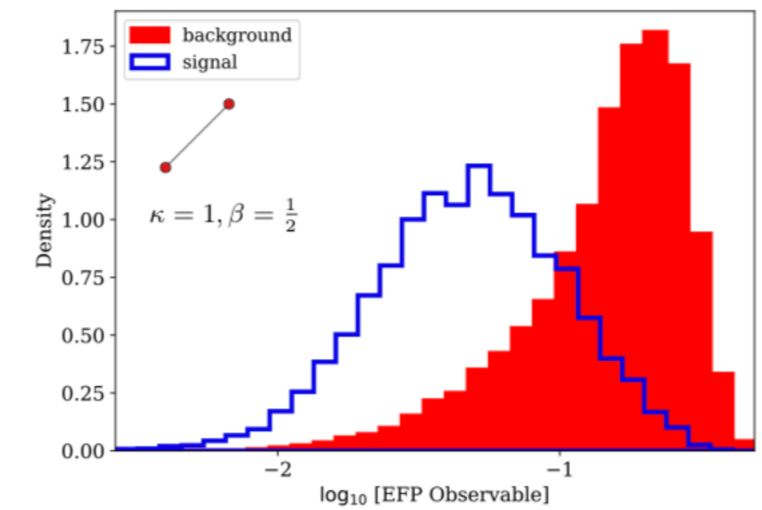
# Learning Observables

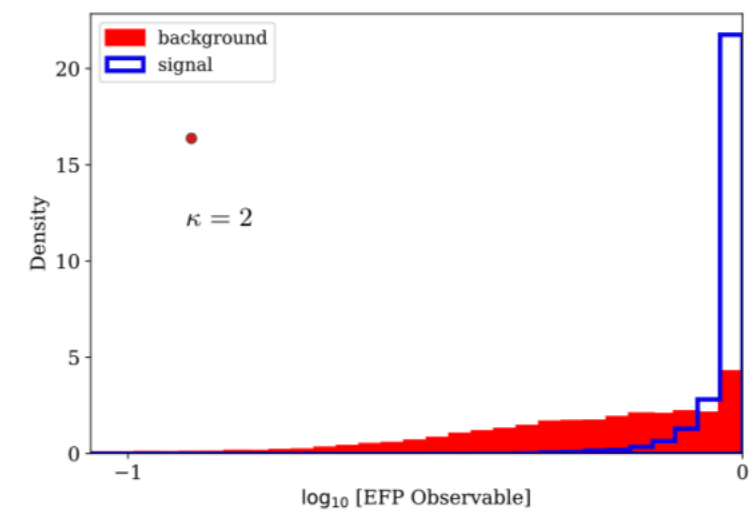
Electron classification performance

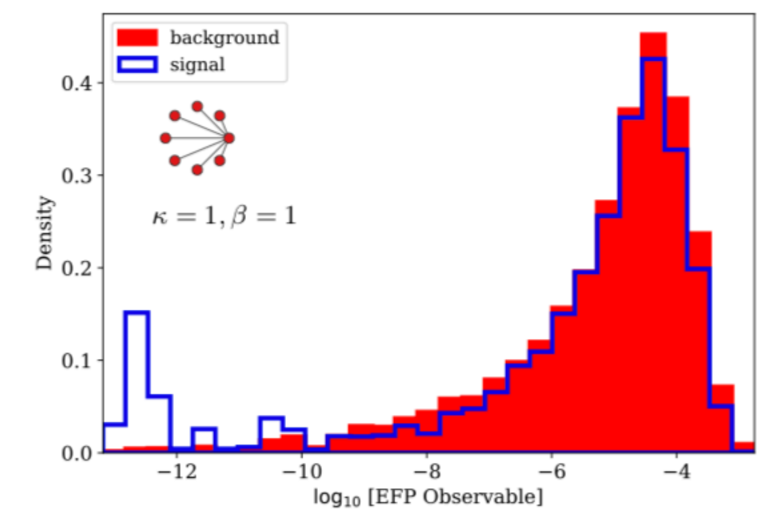
| Base | | Additions $(\kappa, \beta)$ | | (AUC) |
|---|---|---|---|---|
| 7HL | | | | 0.945 |
| 7HL | $+M_{\text{jet}}$ | | | 0.956 |
| 7HL | | $(1, \frac{1}{2})$ | | 0.970 |
| 7HL | $+M_{\text{jet}}$ | $(1, 1)$ | $(1, \frac{1}{2})$ | 0.971 |
| 7HL | | $(2, -)$ | | 0.970 |
| 7HL | $+M_{\text{jet}}$ | $(2, 1)$ | $(2, -)$ | 0.971 |
| CNN | | | | 0.972 |

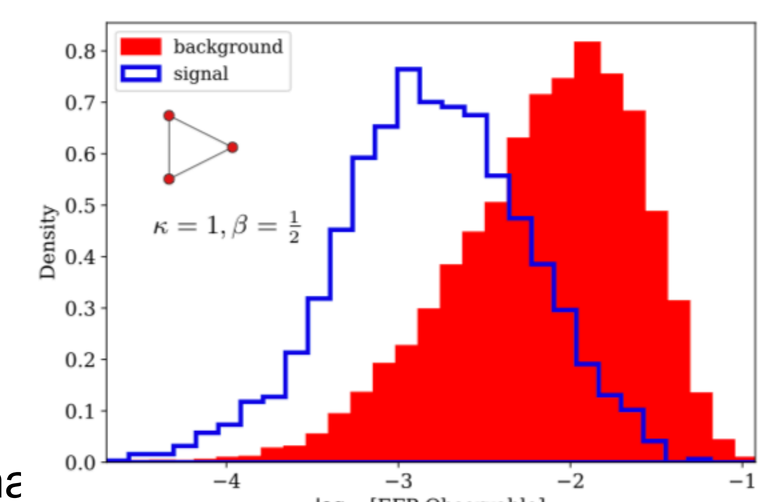
https://arxiv.org/abs/2010.11998
https://arxiv.org/abs/2011.01984



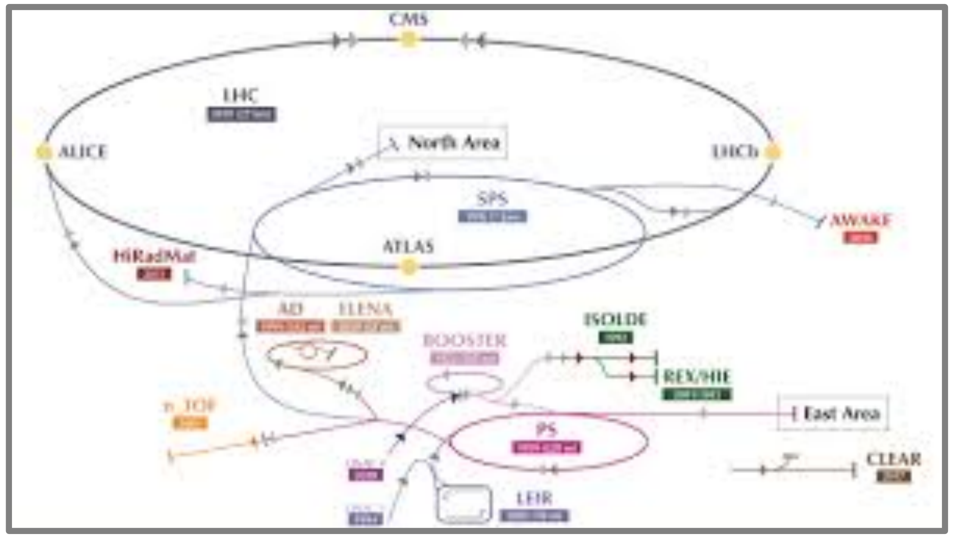



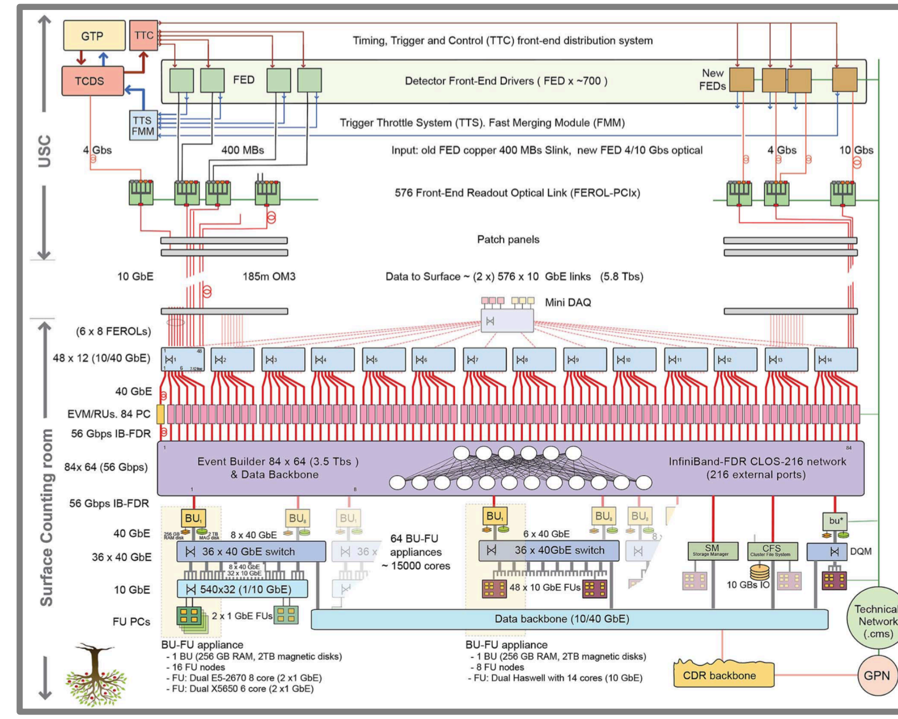

Search in the space of functions using decision ordering.
Simplified to the energy flow polynomial subspace.
Extract set of EFP that matches DNN performance.


# Taking Control

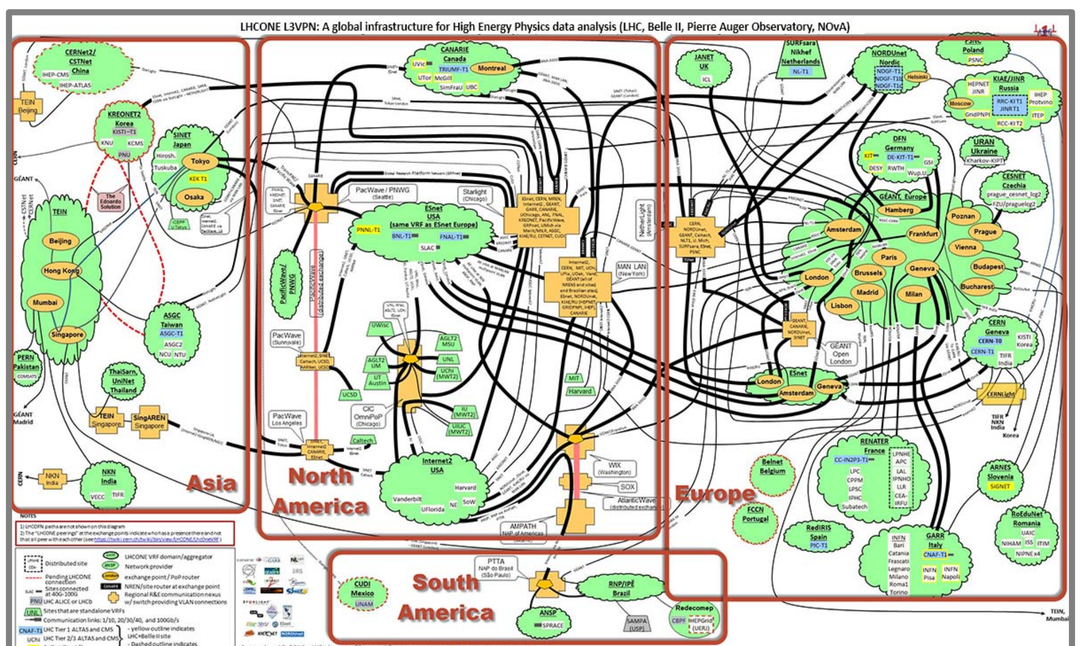
# HEP Instruments



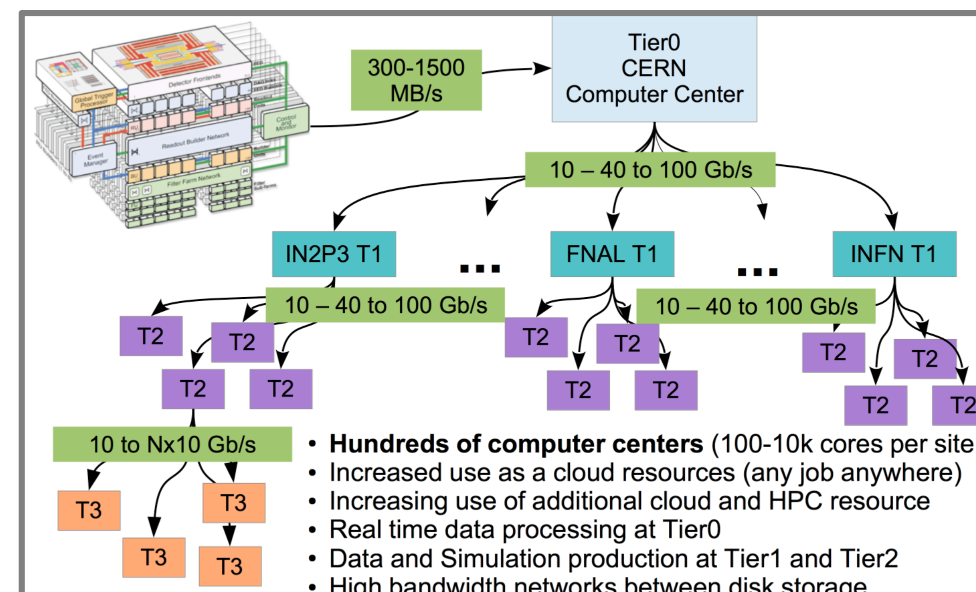

https://home.cern/science/accelerators/




DAQ [IEEE:7111380]




https://home.cern/science/computing/grid




- **Hundreds of computer centers** (100-10k cores per site)
- Increased use as a cloud resources (any job anywhere)
- Increasing use of additional cloud and HPC resource
- Real time data processing at Tier0
- Data and Simulation production at Tier1 and Tier2
- High bandwidth networks between disk storage

Unique set of complex apparatus for doing Science.

# Summary

➡Physics at collider is a **computing intensive endeavor**. Extracting, simulating, reconstructing rare signal from large amount of data.

➡Deep learning offers **great prospects for Science** and Physicists. Fast and efficient data processing.

➡Doing AI at colliders requires to **keep an eye on particular aspects**. Also relevant to other fields of Science.

➡Deep learning is entering High Energy Physics data processing at all levels. A lot done, a long way to go. **You can make a difference**

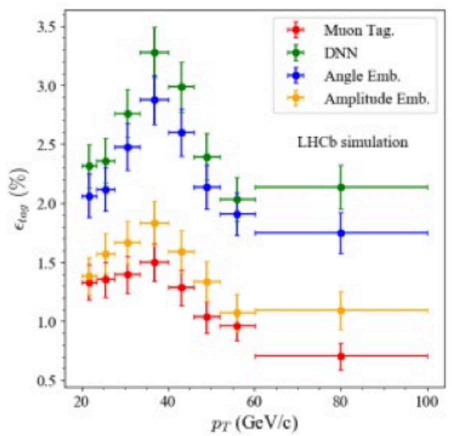
*This work is partially supported by the U.S. DOE, Office of Science, Office of High Energy Physics under Award No. DE-SC0011925, DE-SC0019219, and DE-AC02-07CH11359.*

DEPARTMENT OF ENERGY
UNITED STATES OF AMERICA

mPP

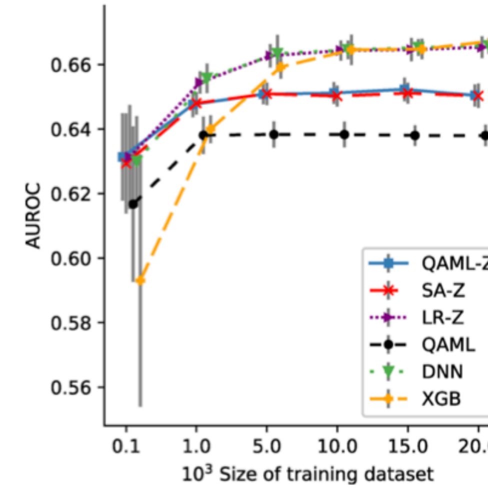
# Classification Task

## QML in High Energy Physics

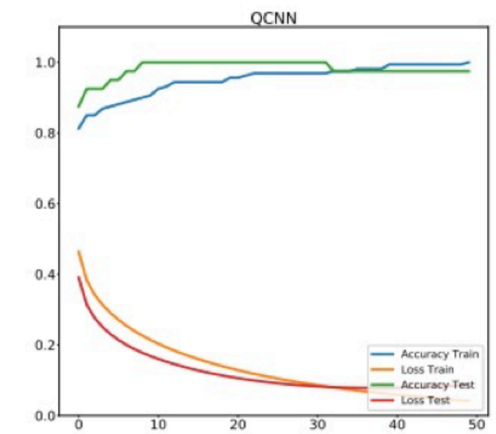



Alexander Zlokapa, Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar, and Maria Spiropulu. **Quantum adiabatic machine learning by zooming into a region of the energy surface.** Physical Review A, 102:062405, 2020. DOI:10.1103/PhysRevA.102.062405.

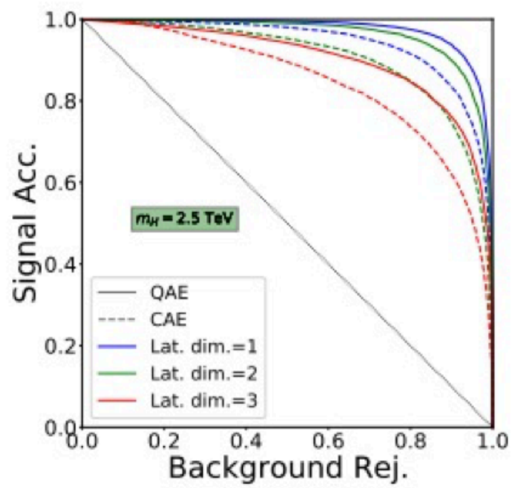
Alessio Gianelle, Patrick Koppenburg, Donatella Lucchesi, Davide Nicotra, Eduardo Rodrigues, Lorenzo Sestini, Jacco de Vries, and Davide Zuliani. **Quantum Machine Learning for *b*-jet identification.** arXiv:2202.13943, 2022.



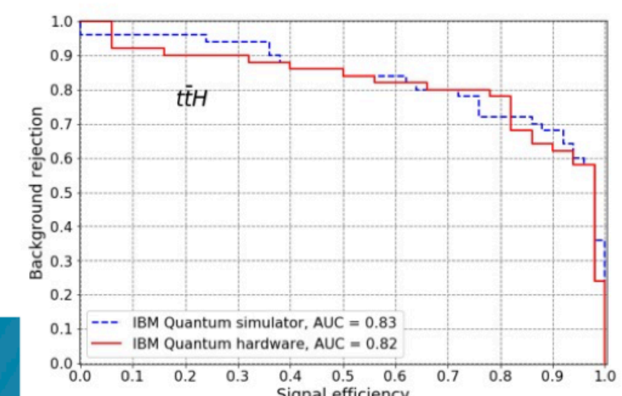

Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. **Quantum convolutional neural networks for high energy physics data analysis.** arXiv preprint: 2012.12177, 2020.



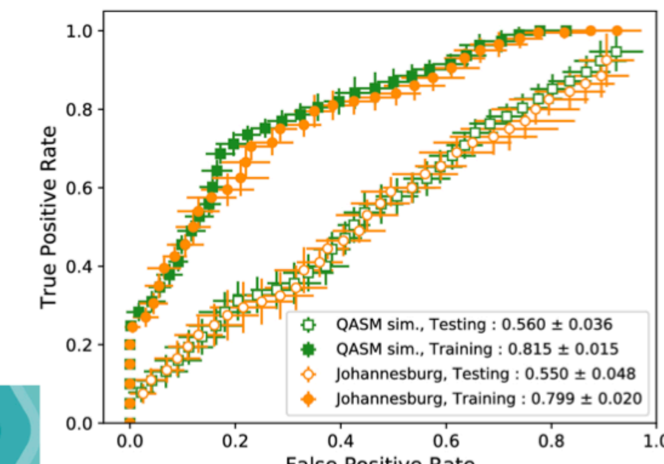

Vishal S Ngairangbam, Michael Spannowsky, and Michihisa Takeuchi. **Anomaly detection in high-energy physics using a quantum autoencoder**. arXiv preprint arXiv:2112.04958, 2021.




Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy C Y Li, and et al. **Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lhc on ibm quantum computer simulator and hardware with 10 qubits**. Journal of Physics G: Nuclear and Particle Physics, 48(12):125003, Oct 2021



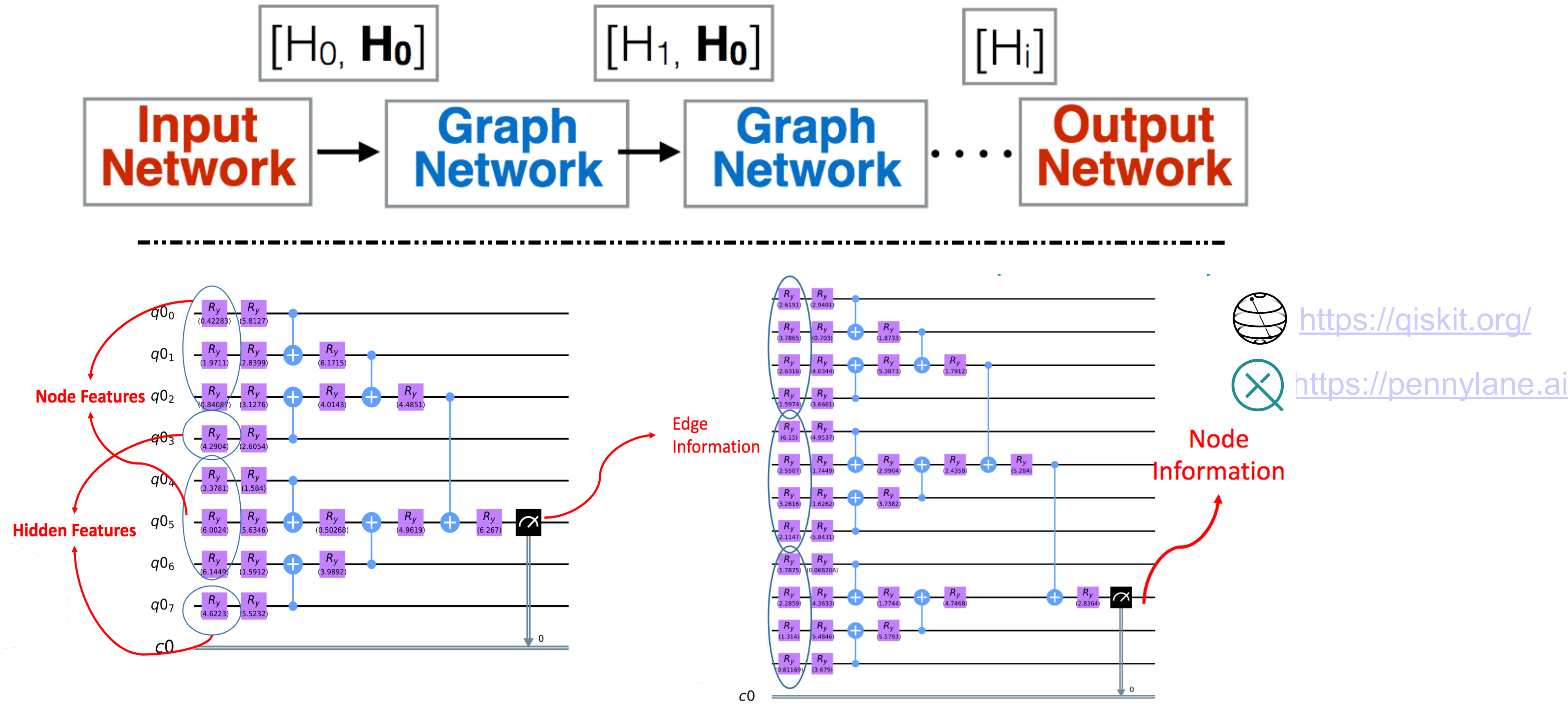

Koji Terashi, Michiru Kaneda, Tomoe Kishimoto, Masahiko Saito, Ryu Sawada, and Junichi Tanaka. **Event classification with quantum machine learning in 20 high-energy physics**. Computing and Software for Big Science, 5(1), January 2021.




*Slide: S. Vallecorsa*

- QA and QC approaches applied to various classification tasks
- Recurring hint of advantage a small training dataset size

# Tracking with Q-GNN

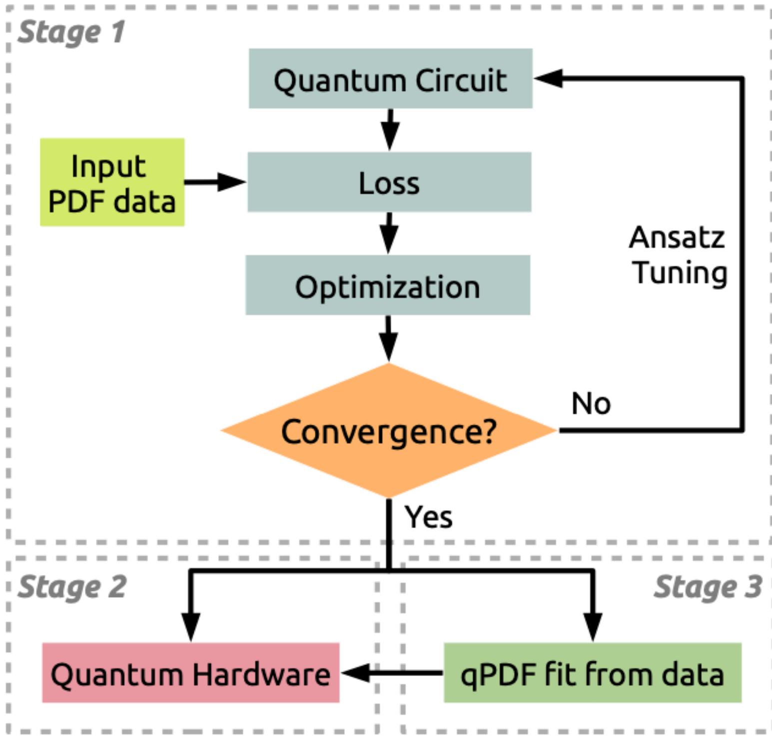



[2007.06868]

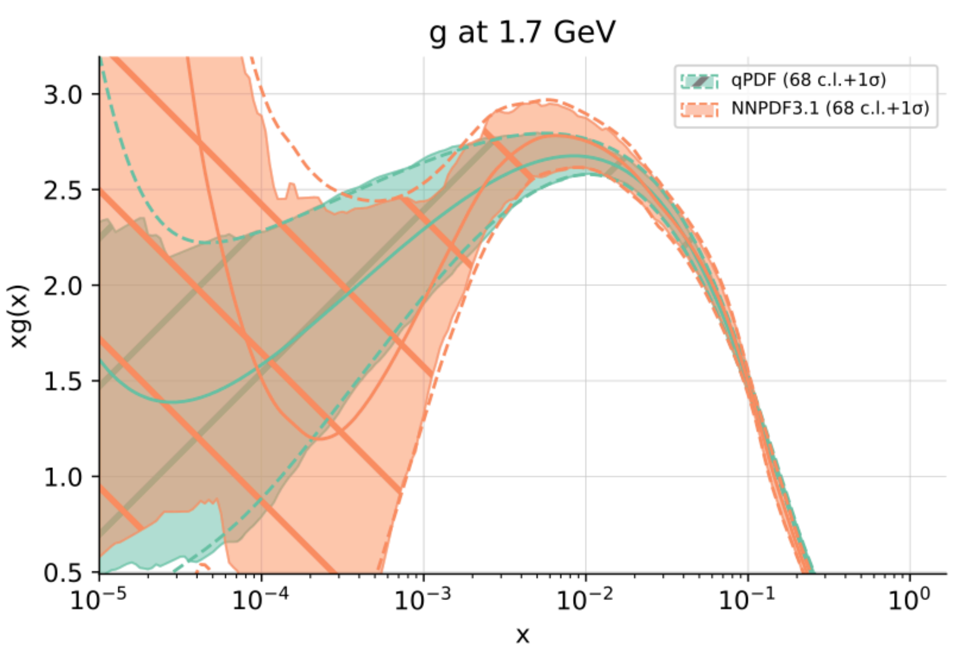
- Quantum/Classical hybrid graph neural network inspired by exatrkx work.
- Promising performance.
- However limites by large number of circuits and training time.

# PDF with Variational Quantum Circuit

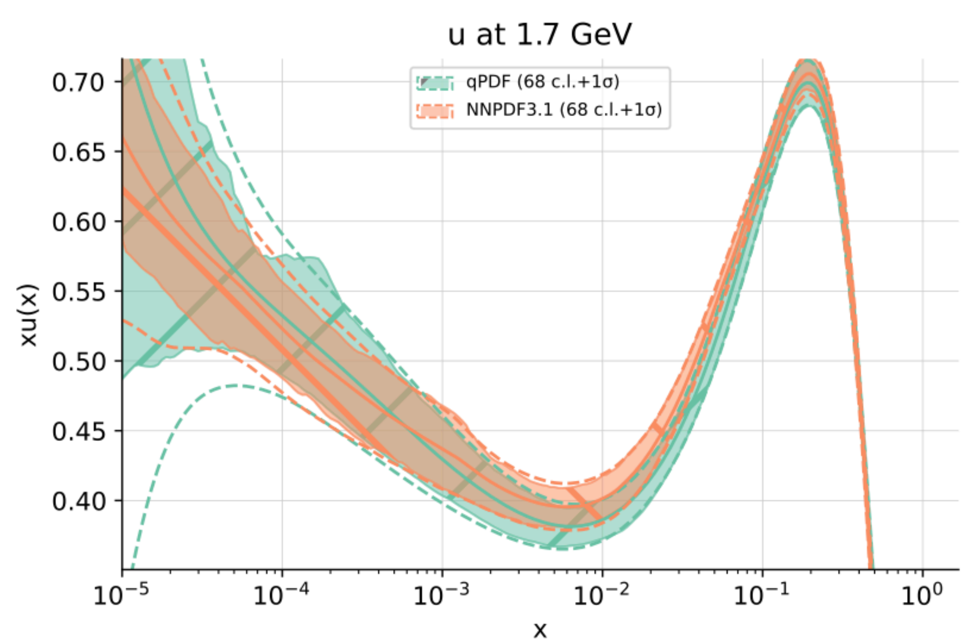
**qPDF Workflow**



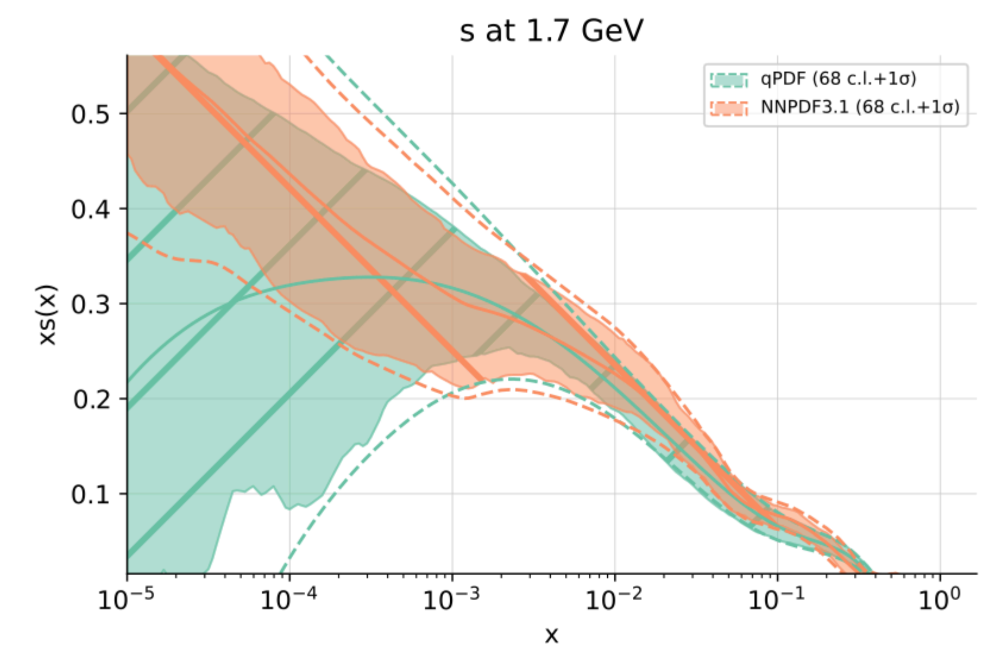

- VQC optimized at each energy scale value
- Parametrization of VQC on x
- Each qbit used represent a parton fraction
- Trained with standard NNPDF procedure
- Remarkable capability to produce PDF with much less parameters than DNN




(a) Gluon pdf.



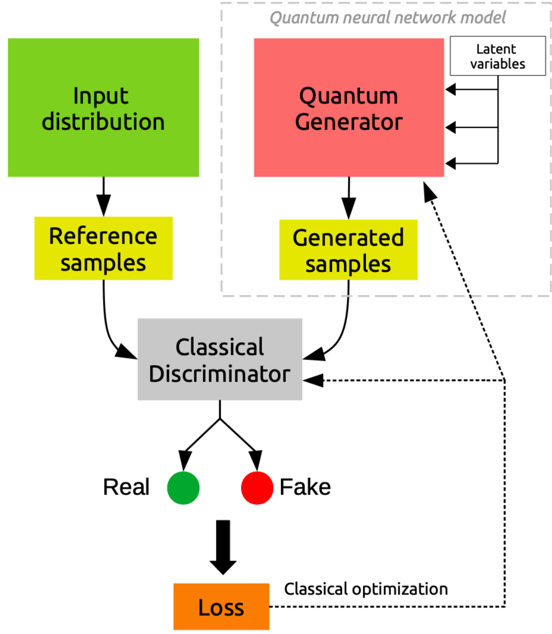

(b) u quark pdf.

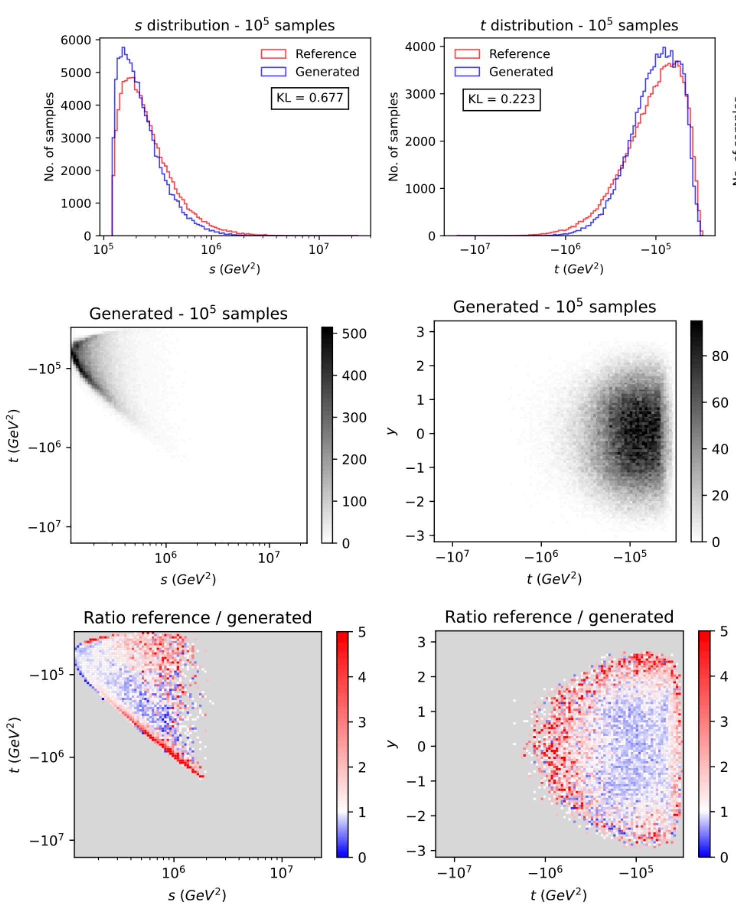



(c) s quark pdf.

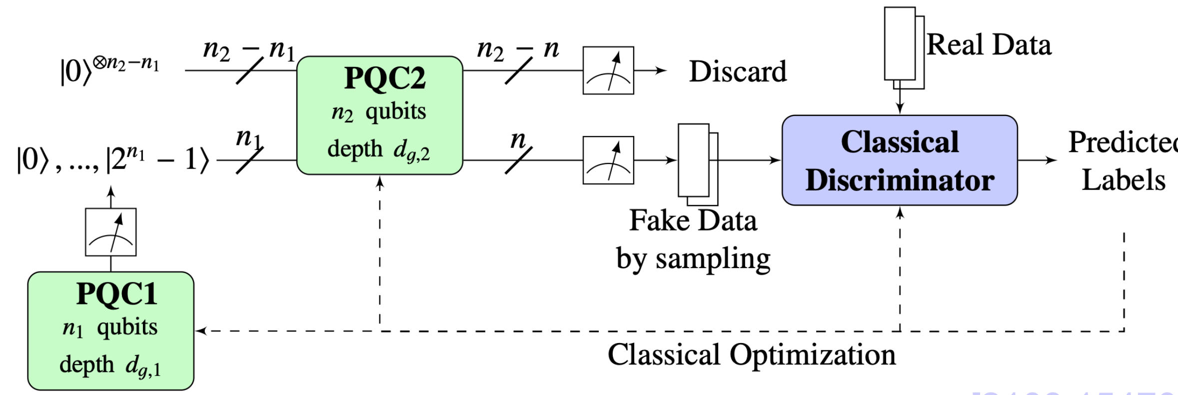
[2011.13934]

# Generative Models

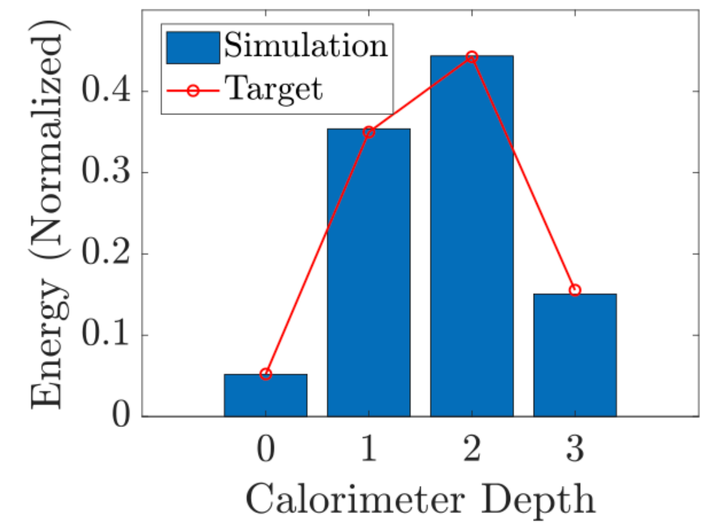



[2110.06933]



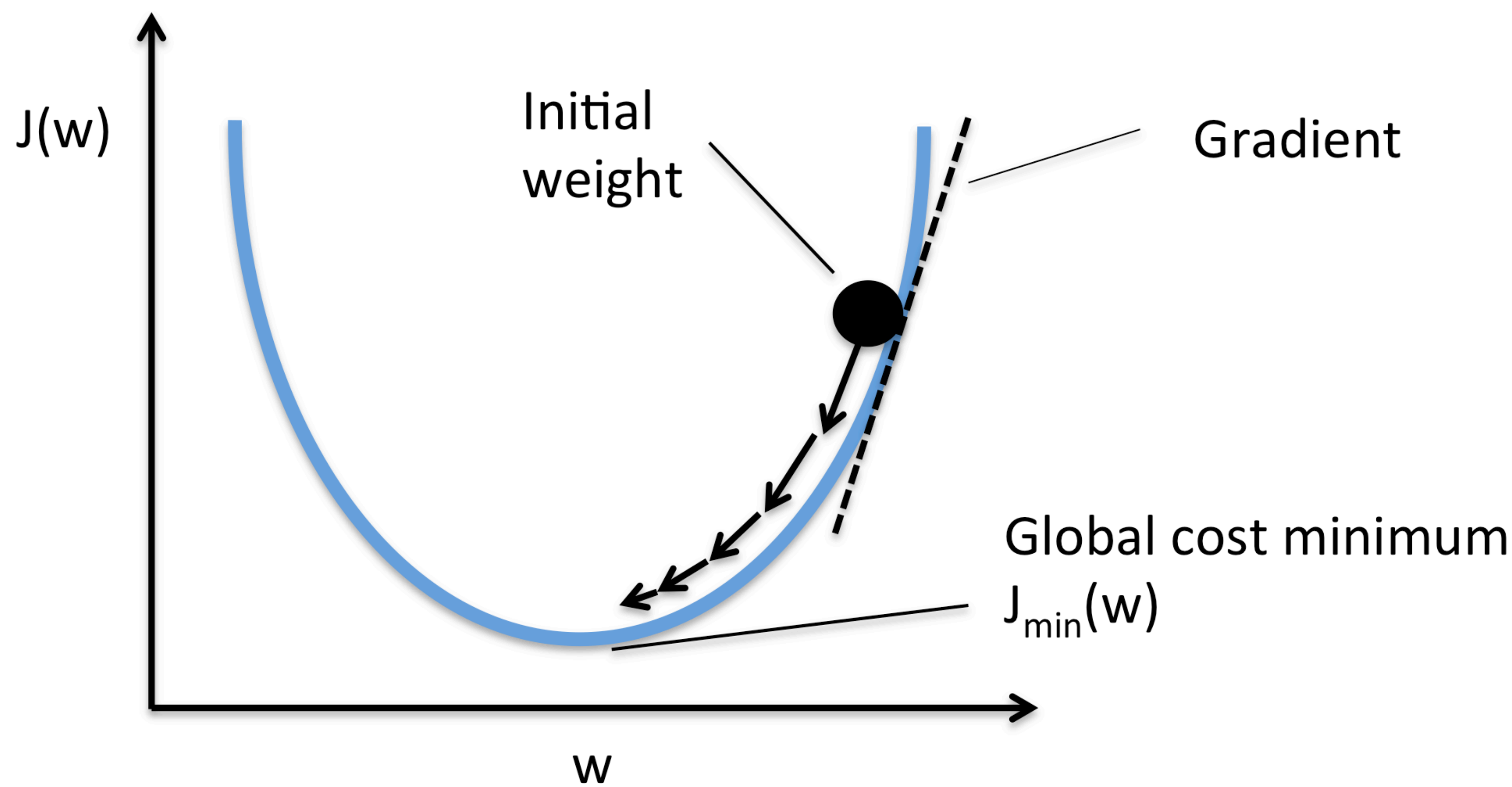

[2103.15470]




- Quantum Generative Adversarial Models inspired from “classical” Generative Adversarial Networks
- Models use various latent vector embedding
- Multiple ways of mapping qbits value/ expectations to original sample format
- Good fidelity of model, slightly decreased due to hardware noise

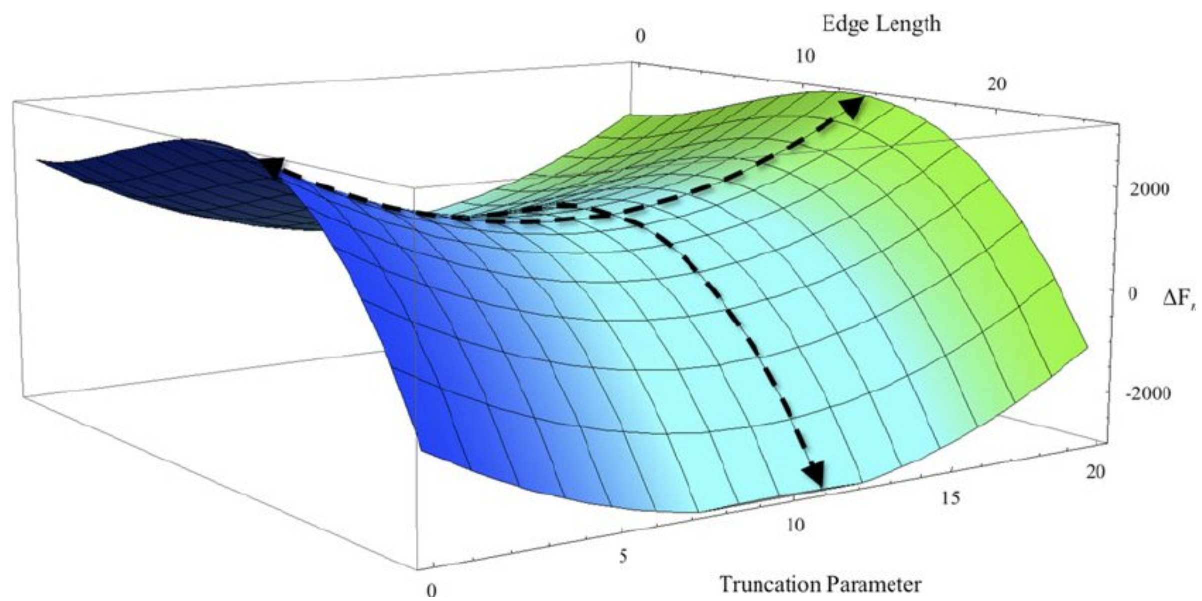
# Optimization Methods

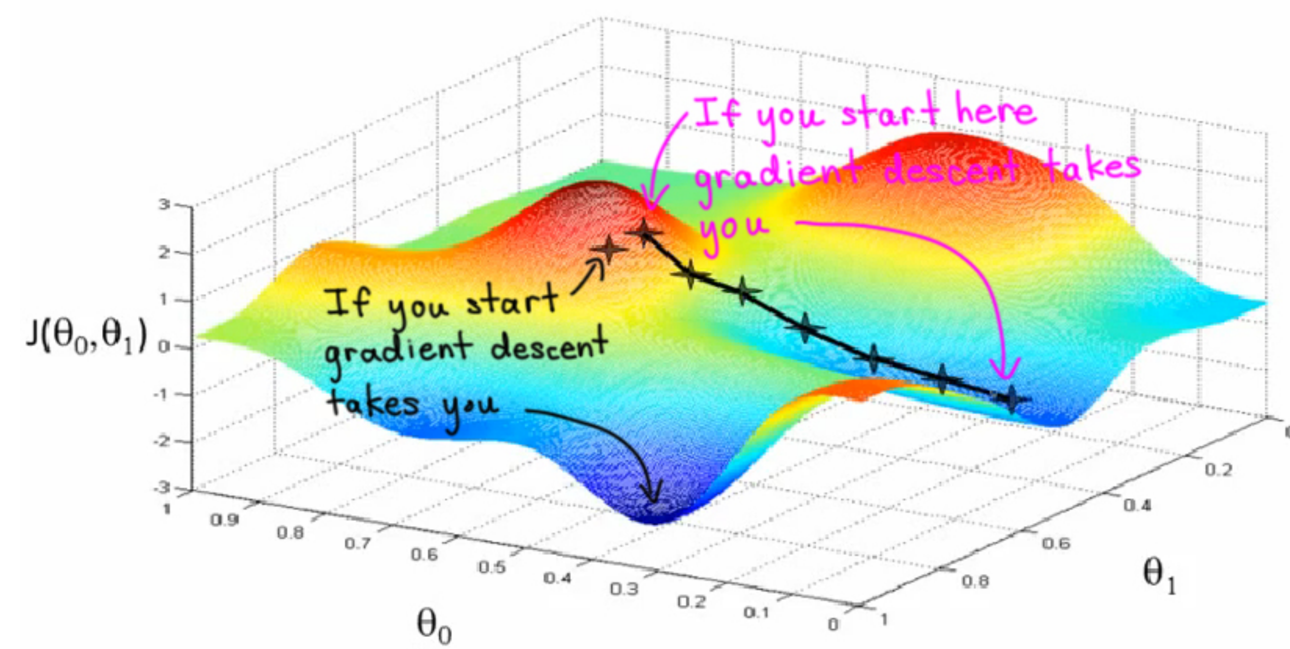
# Gradient Descent Optimization




- For a differentiable loss function f, the first Taylor expansion gives $f(x+\varepsilon) = f(x) + \varepsilon \nabla f(x)$
- The direction to locally maximally decrease the function value is anti-collinear to the gradient $\varepsilon = -\gamma \nabla f(x)$
- Amplitude of the step $\gamma$ to be taken with care to prevent overshooting

# Non-Convex Optimization




- The objective functions optimized in machine learning are usually non-convex
- Non guaranteed convergence of gradient descent
- Gradients may vanish near local optimum and saddle point

# Stochastic Gradient Descent

- Application of one gradient descent is expensive. Can be prohibitive with large datasets
- Following the gradient update from each and every sample of a dataset leads to tensions
    - In binary classification, samples from opposite categories would have “opposite gradients”
- Gradients over multiple samples are independent, and can be computationally parallelyzed

→ Estimate the effective gradient over a batch of samples

$$\nabla_{eff} f(x) = \frac{1}{N} \sum_{i \in batch} \nabla_i f(x)$$

# Non Analytical SGD

- Some valuable loss function might not be analytical and their gradients cannot be derived
- Used finite element method to estimate the gradient numerically

$$\nabla f(x) = \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

- Method can be extended to using more sampling and better precision
- Quite expensive computationally in number of function calls and impractical in large dimension
- Robust methods available in most program library

# Second Order Methods

- Newton-Raphson method defines a recursive procedure to find the root of a function, using its gradient.
- Finding optimum is equivalent to finding roots of the gradient, hence applying NR method to the gradient using the Hessian

$$f(x+\varepsilon) = f(x) + \varepsilon \nabla f(x) + \frac{1}{2} \varepsilon^T H(x) \varepsilon$$

$$\varepsilon \sim -H(x)^{-1} \nabla f(x)$$

- Convergence guaranteed in certain conditions
- Alternative numerical methods tackle the escape of saddle points and computation issue with inverting the Hessian
- In deep learning “hessian-free” methods are prohibitive computationally wise

# Approximate Bayesian Computation

$$\pi(model \mid data) = \frac{\pi(data \mid model)\,\pi(model)}{\pi(data)}$$

- ABC is applicable when the likelihood $\pi(data \mid model)$ is intractable/unknown
- The method requires a simulator or surrogate model
- Generate simulated data for models drawn from the prior, accept/reject whether matching data

- Overly expensive in calls to simulator
  - Introduce summary statistics to enhance border cases
  - Efficient sampling to boost acceptable models
  - Generalized methods for comparing simulated samples with data

→ Principle for likelihood-free inference in HEP : [1805.12244] , …

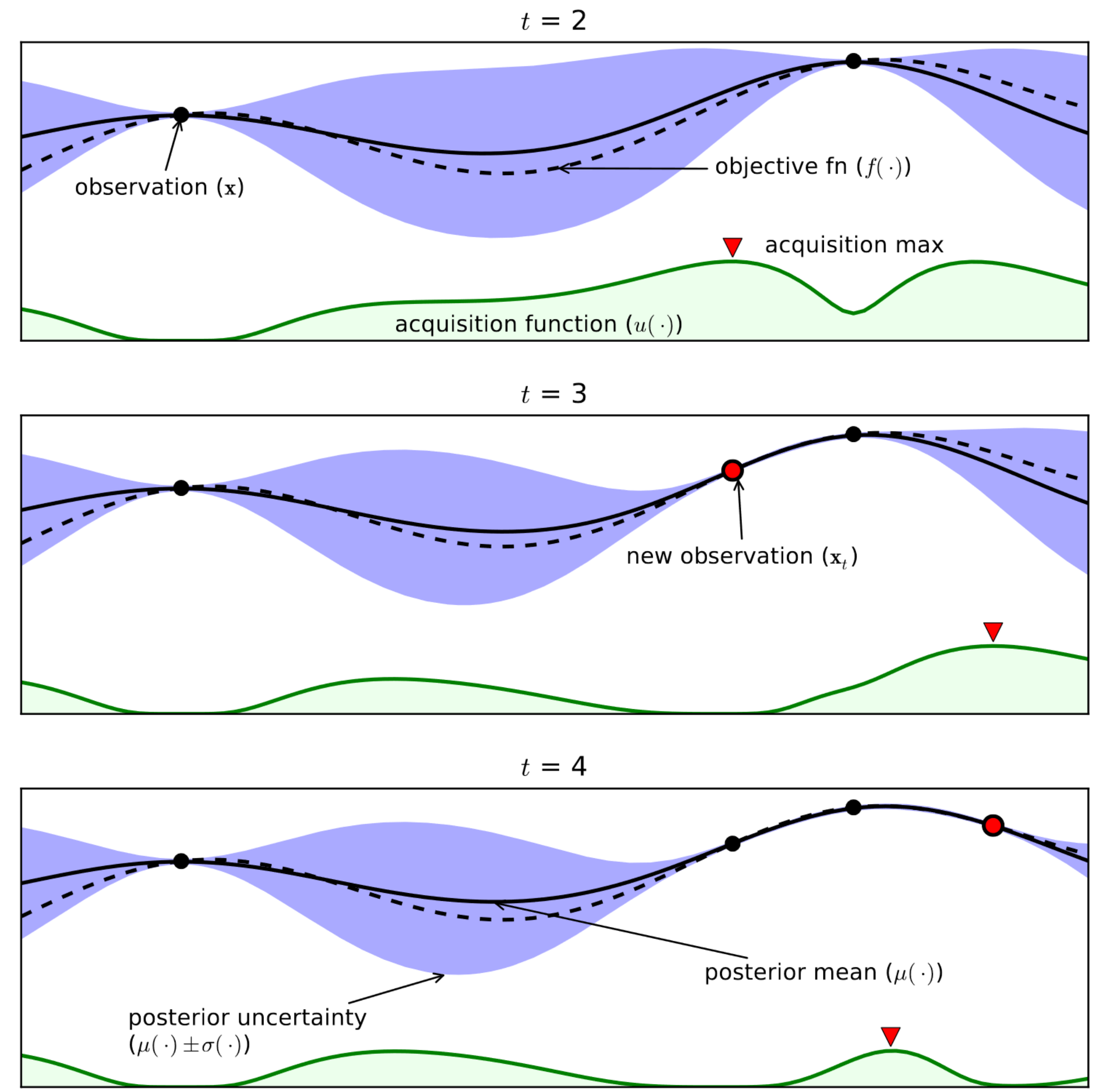
# Bayesian Optimization

- Applicable to optimize function **without close form** and that are **expensive to call** (numerical gradient impractical)
- Approximate the objective function with **Gaussian processes** (GP)
- Start at random points, then sample according to optimized acquisition function
  - ➢ Expected improvement

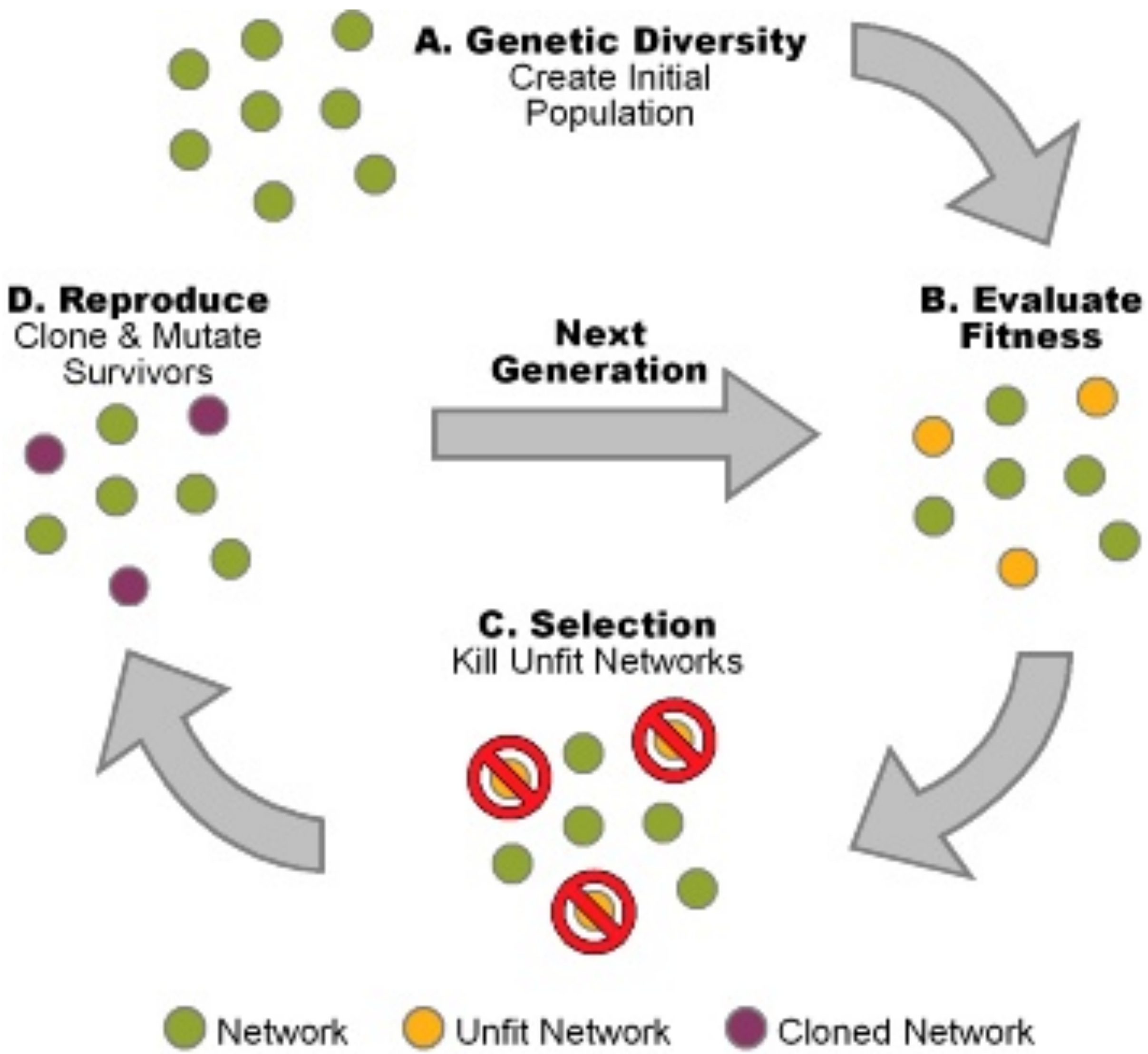
$$-EI(x) = -E(f_{GP}(x) - f(x_{best}))$$

$$LCB(x) = \mu_{GP}(x) + \kappa\,\sigma_{GP}(x)$$

$$-PI(x) = -P(f_{GP}(x) \geq f(x_{best}) + \kappa)$$



# Evolutionary Algorithms

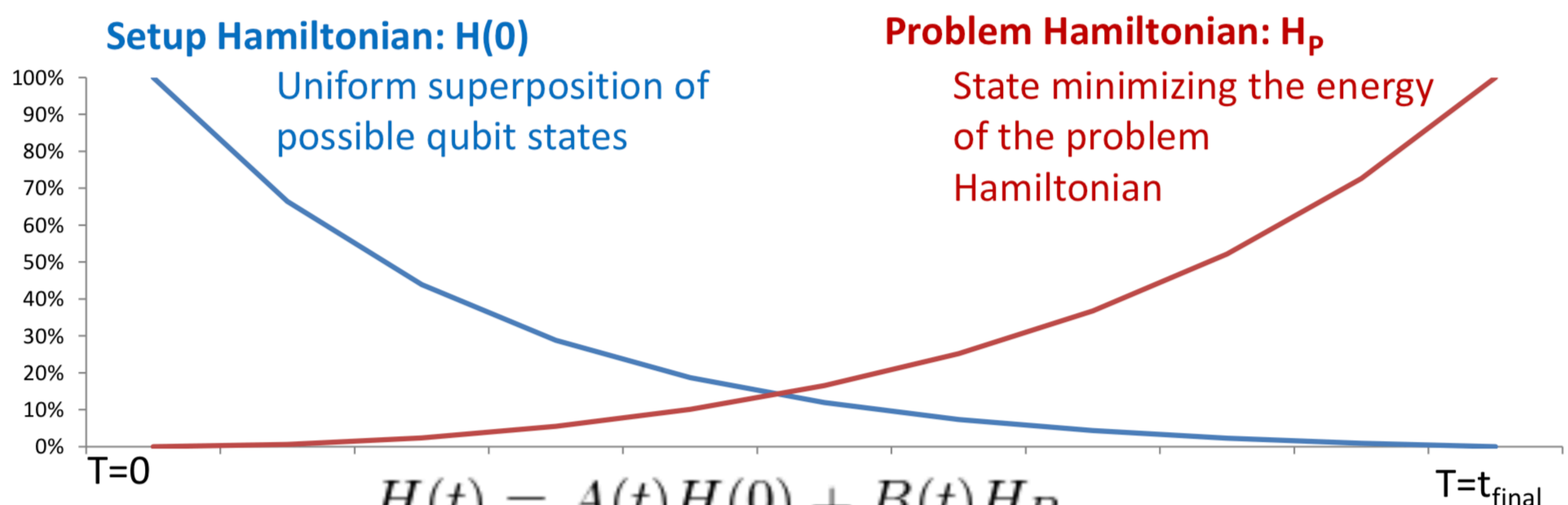



- Applicable to function in high dimensions, with a non regular landscape
- Start from random population
- Estimate fittest fraction of individuals
- Bread and mutate individuals

- Direction of optimization is given by the cross-over and mutation definition
- Multiple over algorithms : particle swarn, ...

# Adiabatic Quantum Annealing

- System setup with trivial Hamiltonian H(0) and ground state
- Evolve adiabatically the Hamiltonian towards the desired Hamiltonian $H_p$
- **Adiabatic theorem** : with a slow evolution of the system, the state stays in the ground state.




$$H(t) = A(t)H(0) + B(t)H_P$$

https://arxiv.org/abs/quant-ph/0001106
https://arxiv.org/abs/quant-ph/0104129

# Simulated Annealing

- Monte-Carlo based method to find ground state of energy functions
- Random walk across phase space
  - → accepting descent
  - → accepting ascent with probability $e^{-\Delta E/kT}$
- Decrease T with time