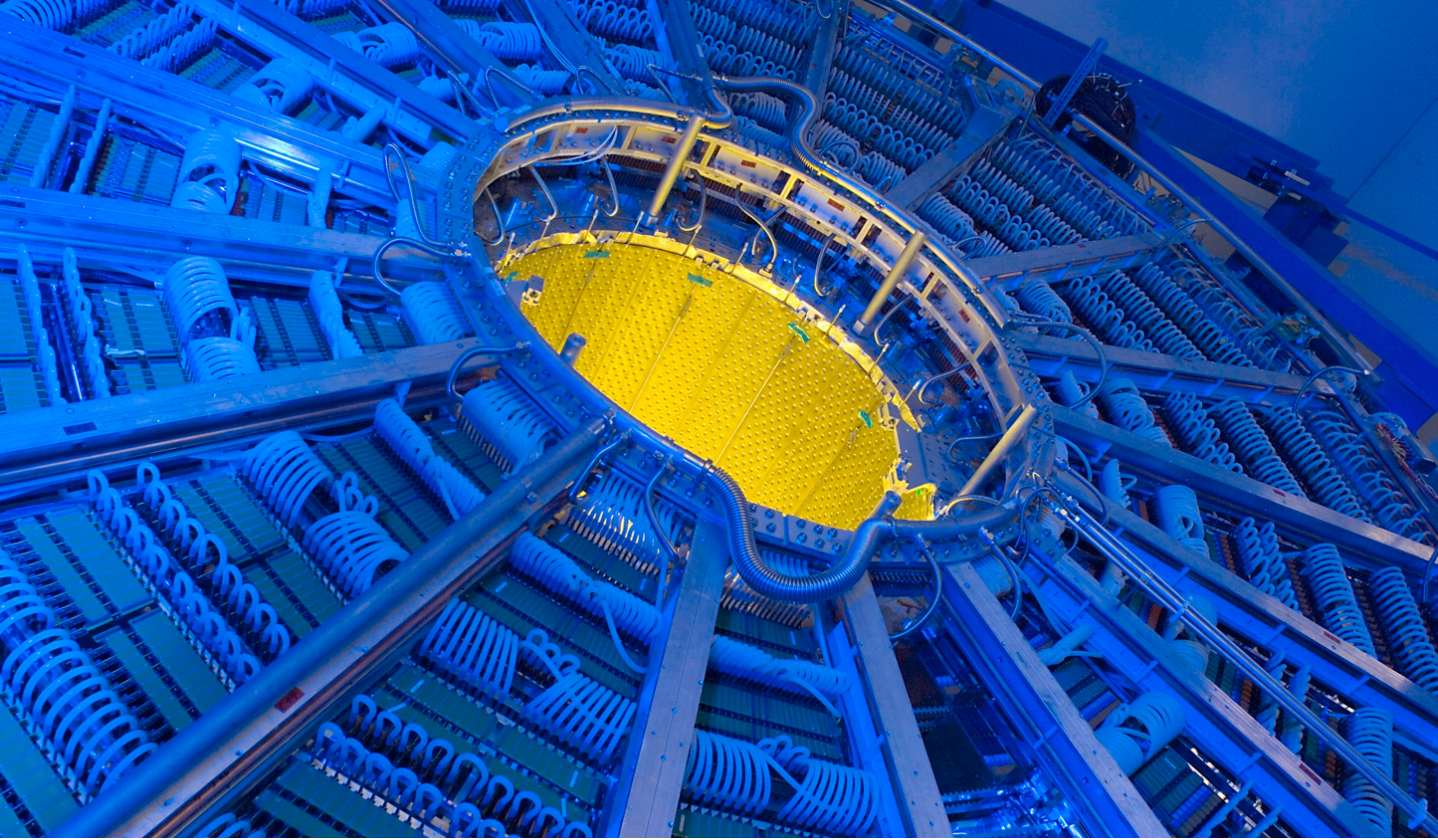


# Summary of the Machine Learning Workshop

(and why ML could be useful to ALICE)

Michele Floris

Alice Offline Week

March 29, 2017

# Outline

- **What is** machine learning
- **IML** and the IML Workshop
- How it can be **useful to ALICE**
  - Analysis
  - Quality Assurance
  - Fast Simulation
  - Reconstruction and calibration (not discussed in details)


Tom Gauld, https://flic.kr/p/9durkH

# What is machine learning?

“**Machine learning** is the subfield of computer science that […] gives computers the ability to **learn without being explicitly programmed**.”

(Wikipedia)

**Supervised**
Learn by example

- Classification
  - Binary
  - Multiclass
- Regression

**Unsupervised**
Finds patterns in the data

- Clustering / Anomaly detection
- Dimensionality Reduction

(only a few incomplete hints of where ML could be useful)

# What is machine learning?

"**Machine learning** is the subfield of computer science that […] gives computers the ability to **learn without being explicitly programmed**."

(Wikipedia)

**Supervised**
Learn by example

Classification

Regression

Binary

Multiclass

Most common application in ALICE (signal vs background)

**Unsupervised**
Finds patterns in the data

Clustering / Anomaly detection

Dimensionality Reduction

(only a few incomplete hints of where ML could be useful)

# What is machine learning?

“**Machine learning** is the subfield of computer science that […] gives computers the ability to **learn without being explicitly programmed**.”

(Wikipedia)

**Supervised**

Learn by example

Classification

Binary

Most common application in ALICE (signal vs background)

Multiclass

Obvious applications: PID, various BG types, QA …

Regression

Calibration, Reconstruction Fast Simulation

**Unsupervised**

Finds patterns in the data

Clustering / Anomaly detection

Applicable to QA?

Dimensionality Reduction

Very useful for visualization / Insight

(only a few incomplete hints of where ML could be useful)

# IML and The IML Workshop

The **Inter-experimental Machine Learning Group** is a LPCC working group:

- Discussion forum (monthly meetings, ofter topical)
- Foster integration of modern tools in HEP workflow
- Benchmark datasets for apples-to-apples comparison of methods
- Tutorials and documentation

https://lpcc.web.cern.ch/lpcc/index.php?page=ml_wg
http://iml.web.cern.ch

**1st IML Workshop (March 20-22)**

- Industry session (IBM, Yandex, Nvidia, Intel)
- CWP discussion and writing session [Google Doc]
- Contributed talks on "Tagging of physics objects"
- Tutorials (TMVA, scikit-learn, Keras, R)
- Mini-challenge

https://indico.cern.ch/event/595059/overview

# Tools and Tutorials

- **TMVA** is the de-facto standard in HEP
  - **Pro**: natural interfacing with our data sets, experience in the field
  - **Con**: sometimes less intuitive, lags behind with respect to industry tools (but lots of recent dev and interfaces with external packages!)
- Many **other tools** exist (often used also in HEP, mostly python)
  - **Scikit-learn**: uniform interface to a large number of methods and methodologies, tools to streamline data
  - **Keras**: go-to package to get started with “**deep learning**”: easy to use + deployment on GPUs
  - Many more in the market (e.g. **R,** Caffé, Torch, Theano, Tensorflow, …)
- **Tutorials** given at the workshop:
  - Getting started
  - Describe various ways to import our datasets in these tools
  - See also challenge examples
  - **Webcast recording** available soon

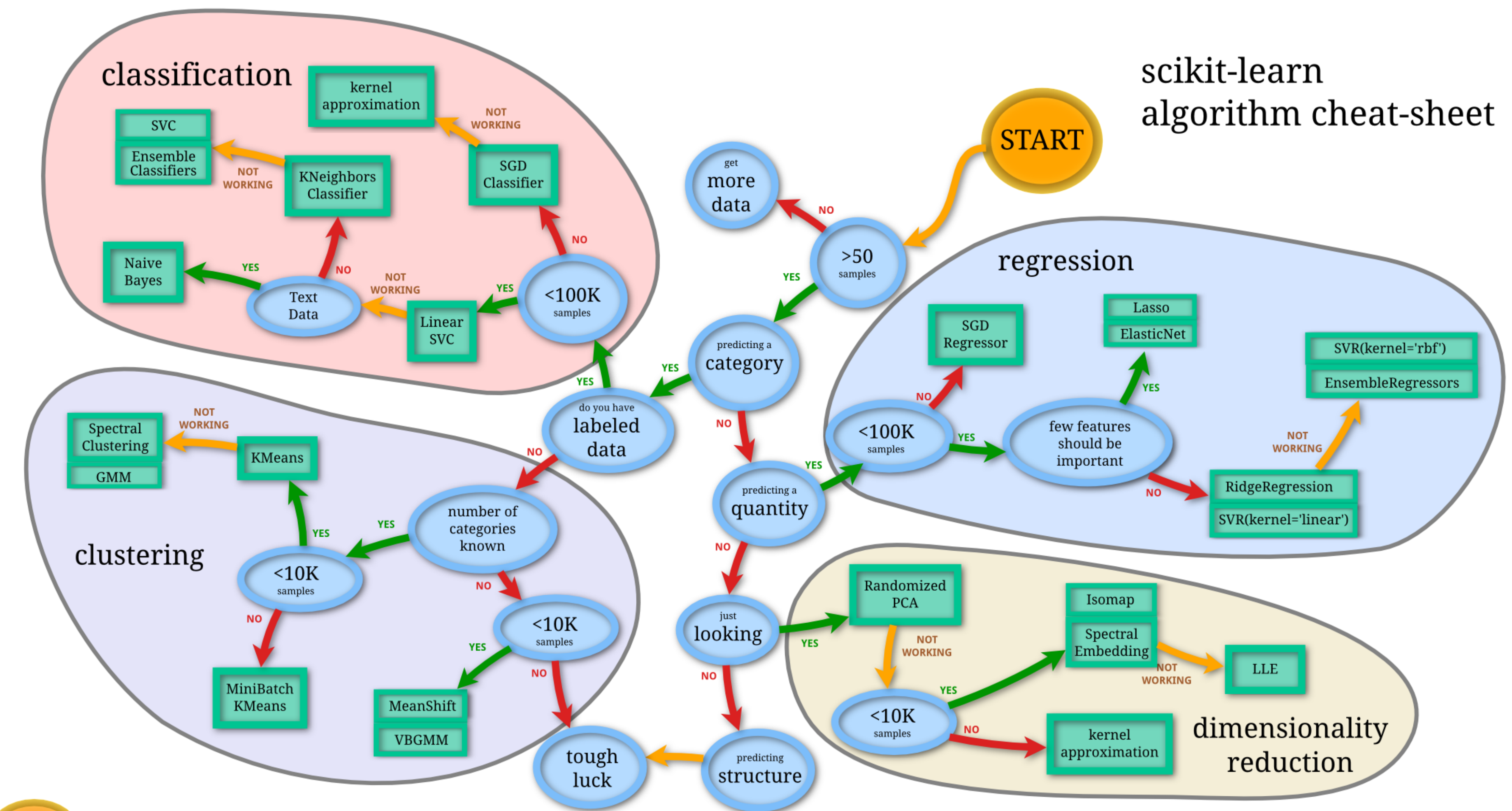
# Physics analysis

# ML For Analysis

- **In ALICE** several applications already, but:
  - Mostly **binary classification** signal-vs-BG
  - Many use **TMVA out of the box**
    ⇒ a lot to be gained optimizing methods and preprocessing input
  - **Recent efforts** to use other tools / approaches
    - For instance:
      *b*-jet tagging (presented at workshop),
      low mass dielectrics (poster at QM17),
      …
- **At the IML workshop**
  - Focus was on "**tagging**"
  - 2 classes of talks:
    - "**Traditional**" ML used to publish physics results
    - **Exploratory** talks, benefit of new techniques?

# ML For Analysis

- **In ALICE** several applications already, but:
  - Mostly **binary classification** signal-vs-BG
  - Many use **TMVA out of the box**
    ⇒ a lot to be gained optimizing methods and preprocessing input
  - **Recent efforts** to use other tools / approaches
    - For instance:
      *b*-jet tagging (presented at workshop),
      low mass dielectrics (poster at QM17),
      …
- **At the IML workshop**
  - Focus was on "**tagging**"
  - 2 classes of talks:
    - "**Traditional**" ML used to publish physics results
    - **Exploratory** talks, benefit of new techniques? ⇐ (incomplete personal perspective)

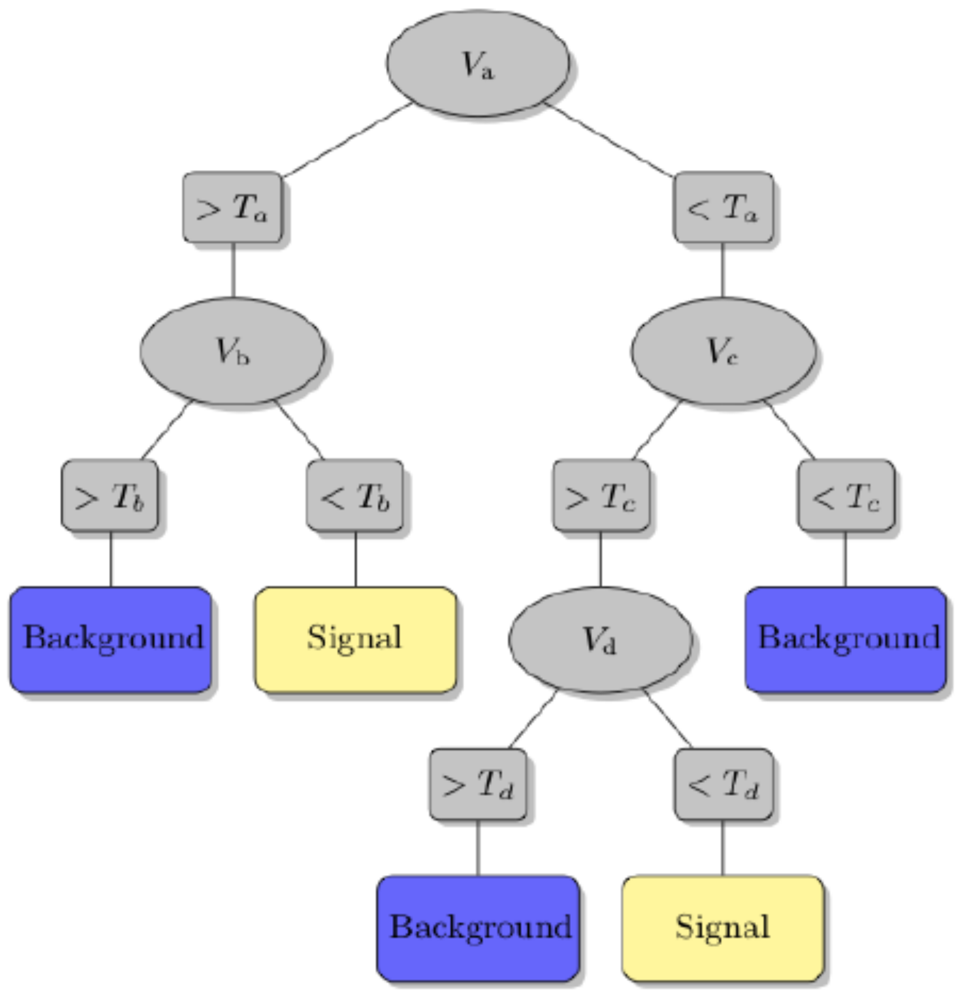


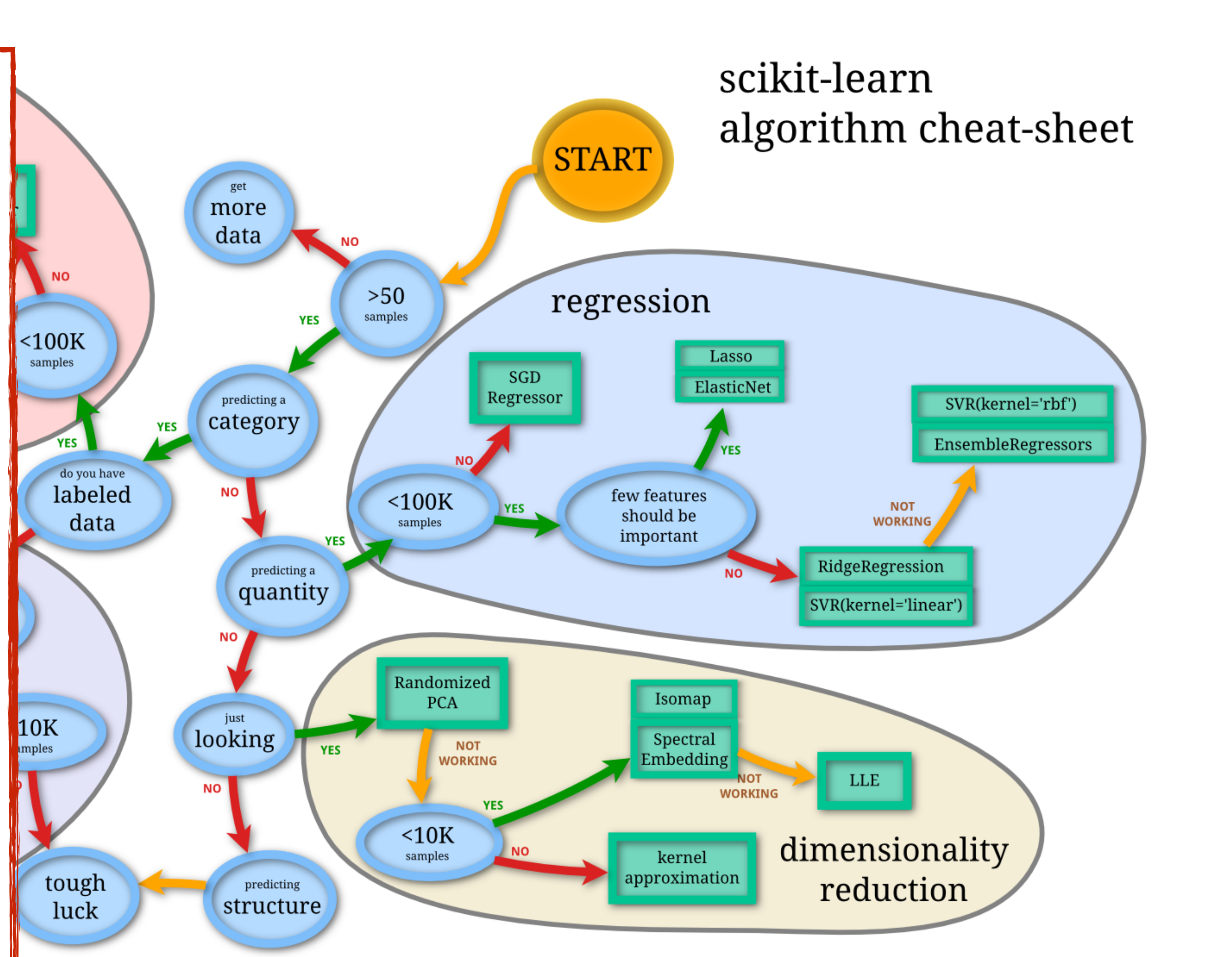

http://scikit-learn.org/stable/tutorial/machine_learning_map/

Dozens of methods available in modern ML packages

# Decision Trees

(Boosted DT probably the most common methods in HEP)



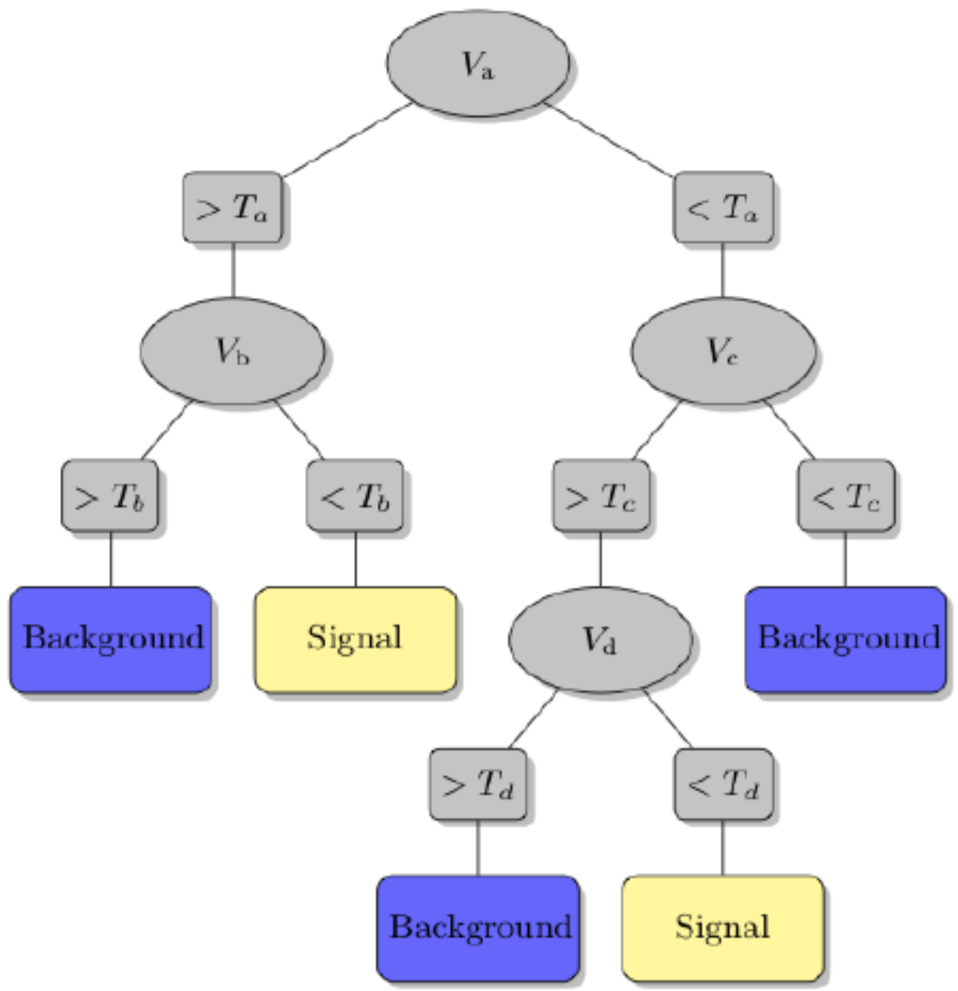




http://scikit-learn.org/stable/tutorial/machine_learning_map/

Dozens of methods available in modern ML packages

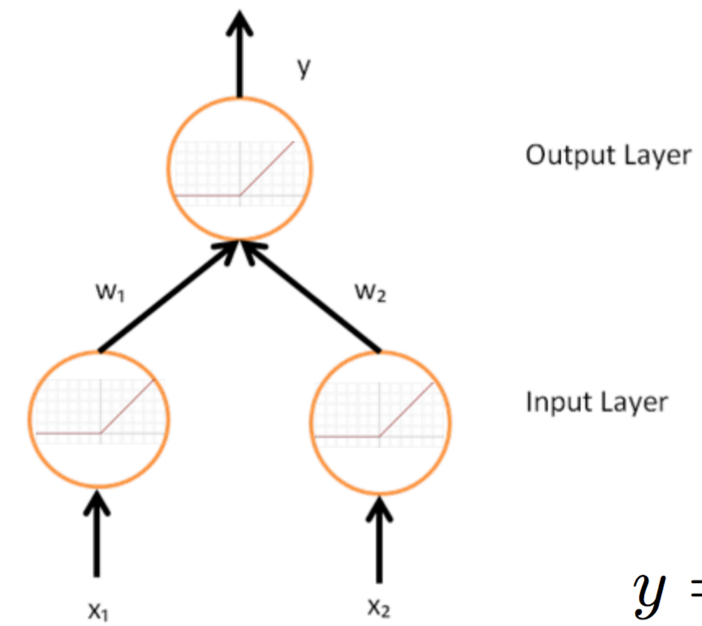
## Decision Trees

(Boosted DT probably the most common methods in HEP)




## A Very Simple Network

Increasing popularity in HEP




$$y = f(f(x_1)w_1 + f(x_2)w_2)$$

$$f(x) = \Theta(x) \cdot x$$

Slide by Gregor Kasieczka

Dozens of methods available in modern ML packages

# Deep learning

- What is **Deep Learning**:
  - Use a cascade of many layers of nonlinear processing units for **feature extraction** and **transformation**
  - Learn **multiple levels of representations** that correspond to different levels of **abstraction** (adapted from wikipedia)
  - Depending on the problem, specific architecture can exploit symmetries in your problem (e.g. convolutional networks for translational invariance)
- **Why?**
  - Traditional methods need properly “engineered” features
  - Deep methods can “discover” new representations
  - Significant improvements in some fields (e.g. computer vision)
  - **Not a silver bullet:** simpler methods may be better for many applications
- **Examples**:
  - Jet images and CNV
  - Deep networks for b-jets tagging
  - (Generative) **adversarial networks**

# Deep learning

- What is **Deep Learning**:
  - Use a cascade of many layers of nonlinear processing units for **feature extraction** and **transformation**
  - Learn **multiple levels of representations** that correspond to different levels of **abstraction**

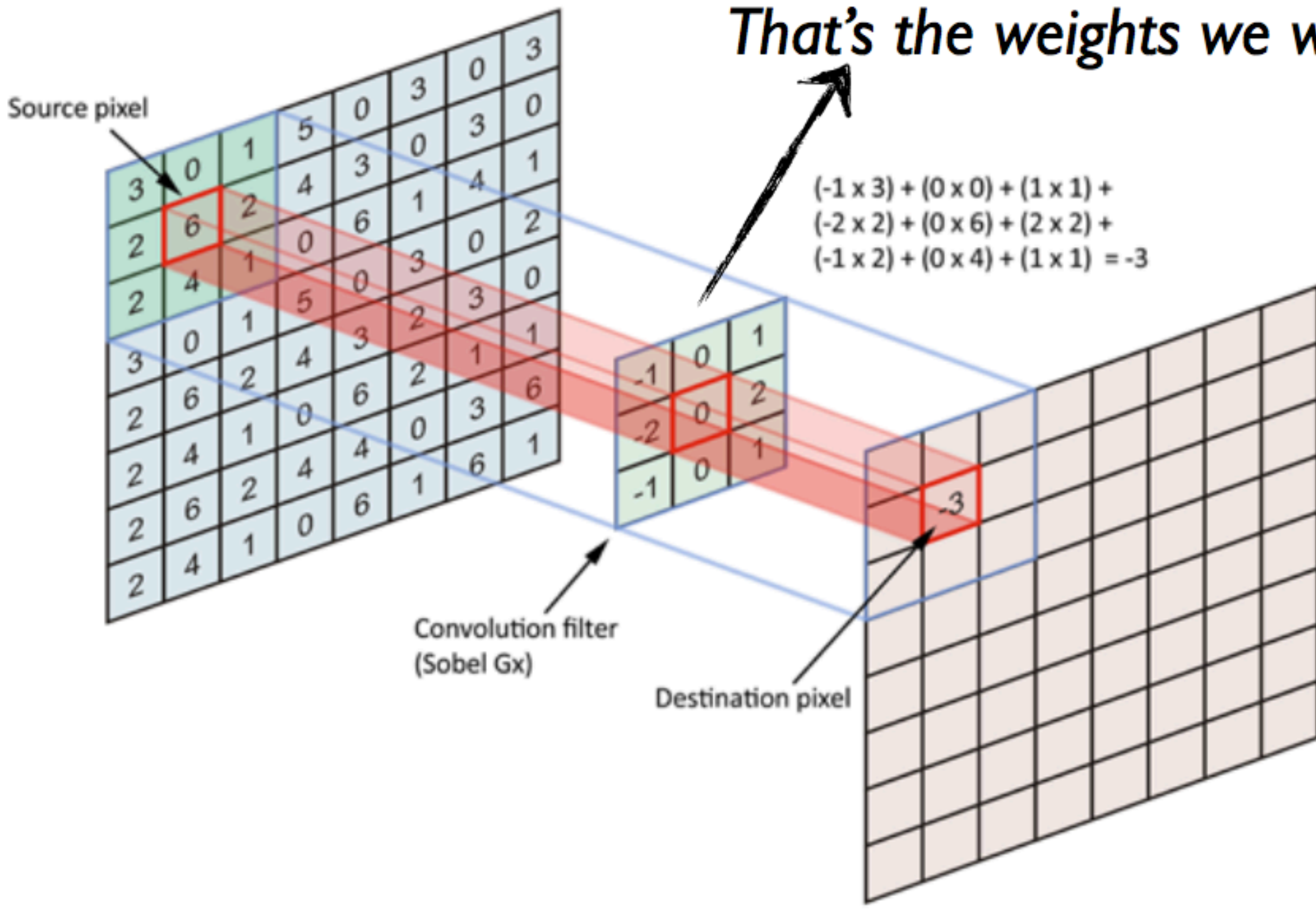
(adapted from wikipedia)

  - Depending on the problem, specific architecture can exploit symmetries in your problem (e.g. convolutional networks for translational invariance)
- **Why?**
  - Traditional methods need properly “engineered” features
  - Deep methods can “discover” new representations
  - Significant improvements in some fields (e.g. computer vision)
  - **Not a silver bullet:** simpler methods may be better for many applications
- **Examples**:
  - Jet images and CNV
  - Deep networks for b-jets tagging
  - (Generative) **adversarial networks**

**In the following:**
Slides stolen (mostly) from the IML Workshop Talks

# Convolutional Network

*That's the weights we want to train*




*Fold a mask with the input to get output*
*What is learned are the parameters of the mask*

Convolutional (conv) layer

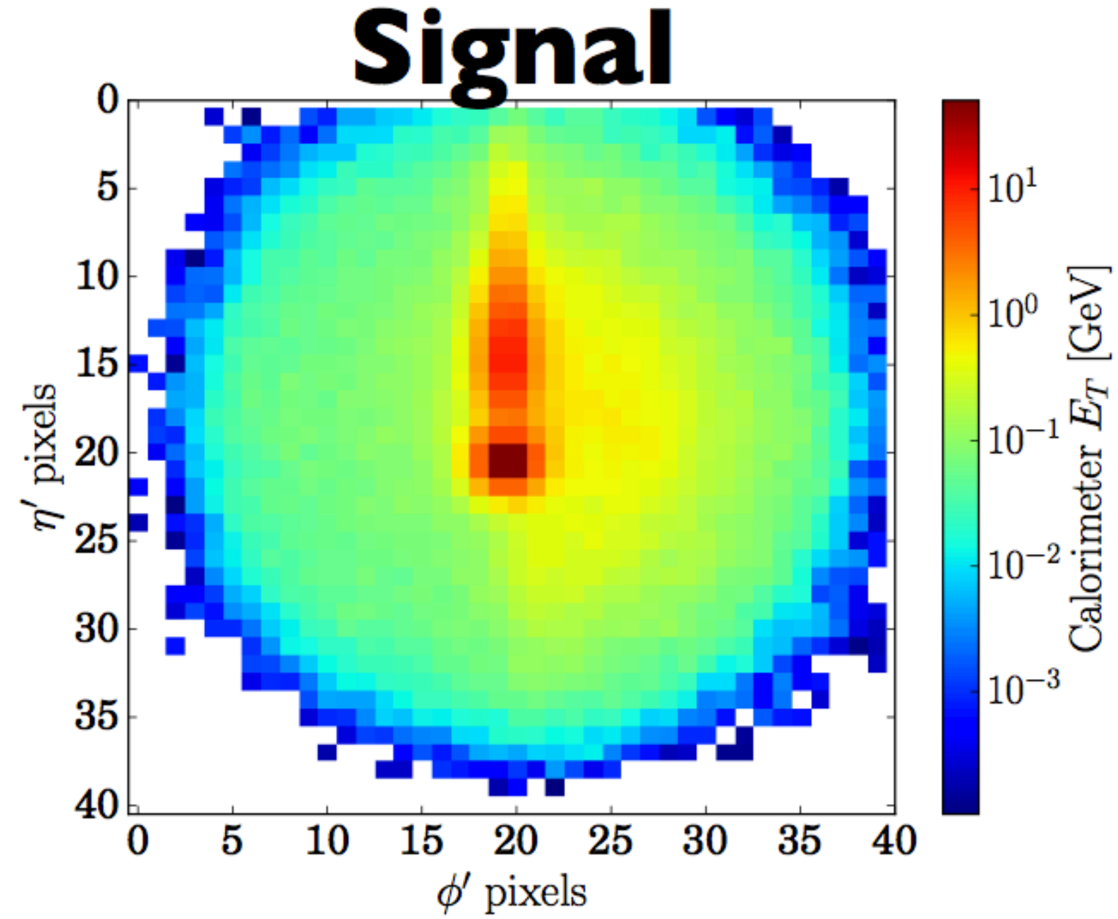
- ***How to build a convolutional network***
  - Chain multiple conv layers
  - Reduce image resolution in between (optional)
  - Use multiple masks per layer
  - Add linear ANN in the end (optional)

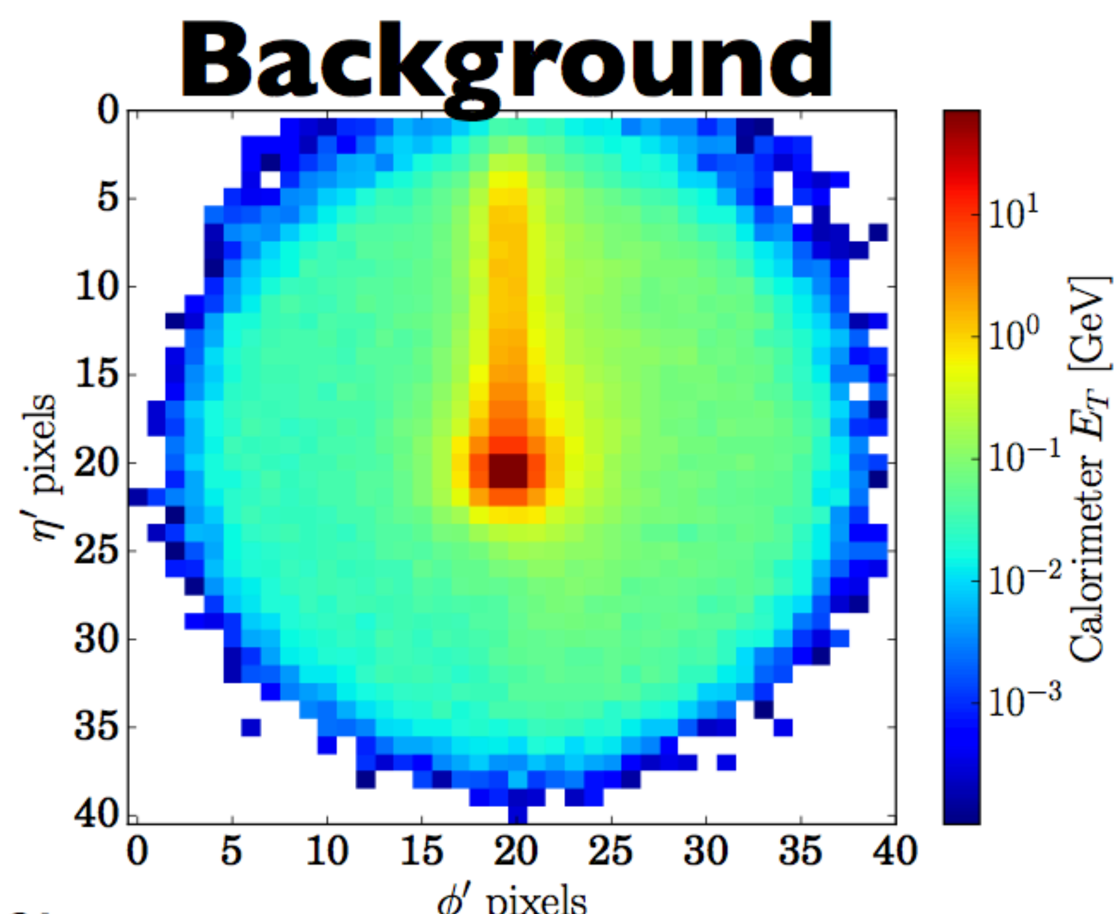
*(This is still a network. We just use a fancy idea to decide which nodes to connect to each other)*

# Image approach

- Jets = 2d grayscale images:
  - 1 pixel = 0.1 in eta, 5 degree in phi
  - pixel energy: calorimeter ET
- Preprocessing
  - Center maximum
  - Rotate so that second maximum is 12 o'clock
  - Flip so that third maximum is on the right side
  - Crop to 40x40 pixels

(Signal = top jets, BG = QCD jets)




*Overlay of 10k images*

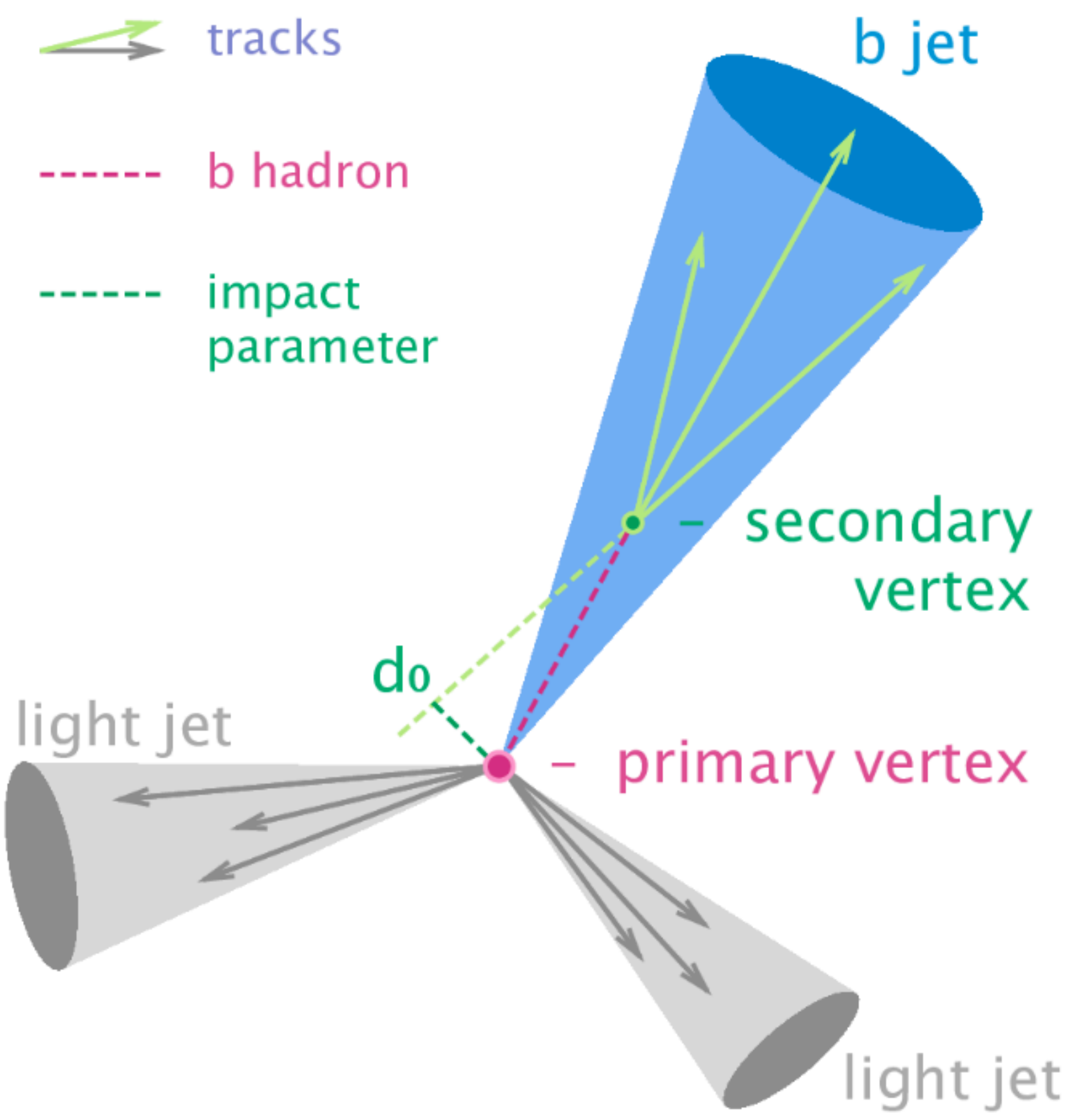
Gregor Kasieczka

# b-jet identification

- B-hadrons decay in the (sub-)millimeter range ($c\tau \sim 500\ \mu$m),
  → displaced from primary vertex
- Common discriminators:
  - Reconstructed secondary vertices
  - Track impact parameters
- Secondary vertex reconstruction:
  - Here: All three-track combinations considered (3-prong vertices)
  - Dispersion as vertex quality measure

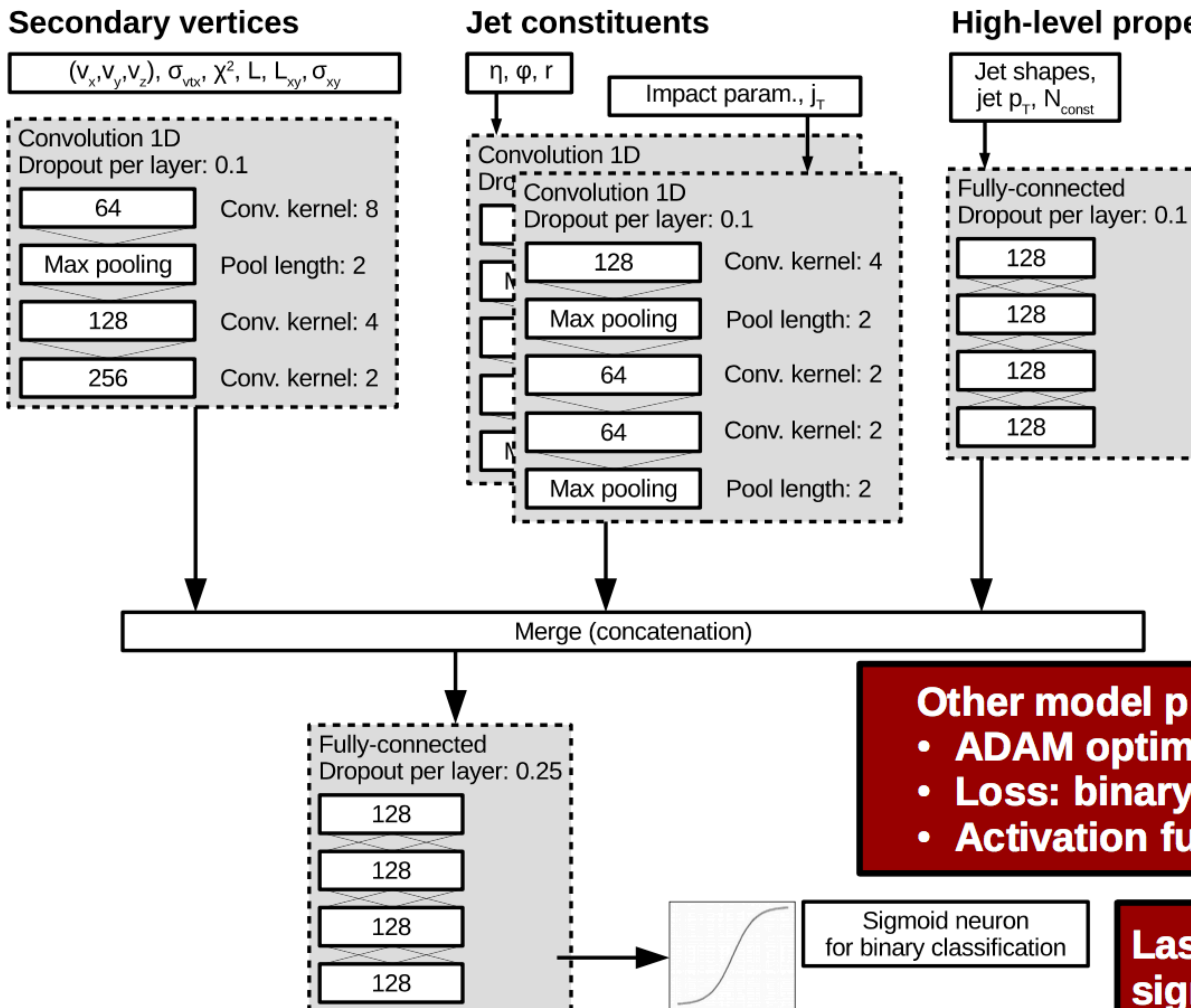
“Conventional” approach:

Application of rectangular cuts on properties of most displaced vertices




http://bartosik.pp.ua/hep_sketches/btagging

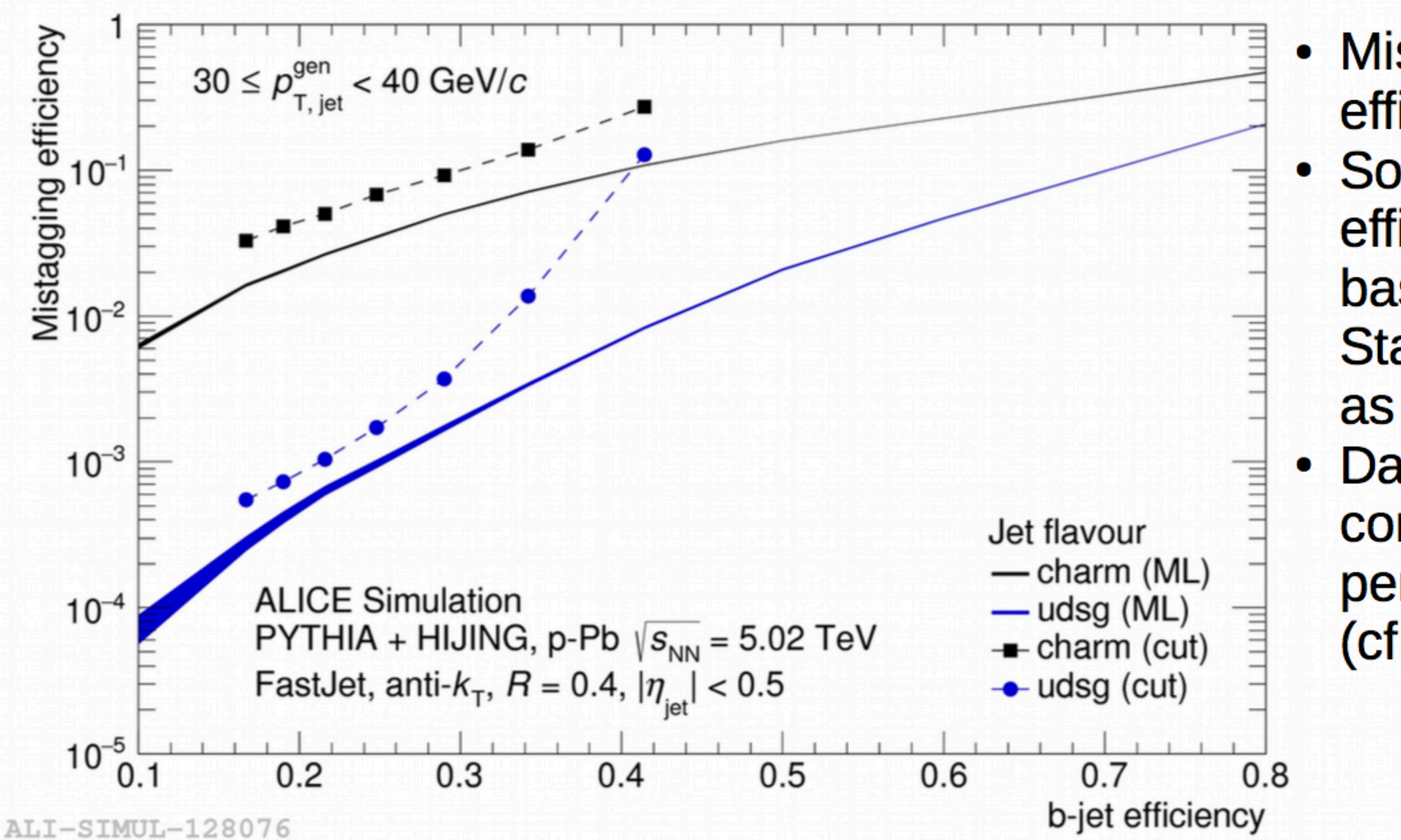
# Model design




**Other model properties**
- **ADAM optimizer**
- **Loss: binary crossentropy**
- **Activation function: ReLU**

**Last neuron is sigmoid-activated**

# Mistagging efficiencies vs. b-jet efficiency

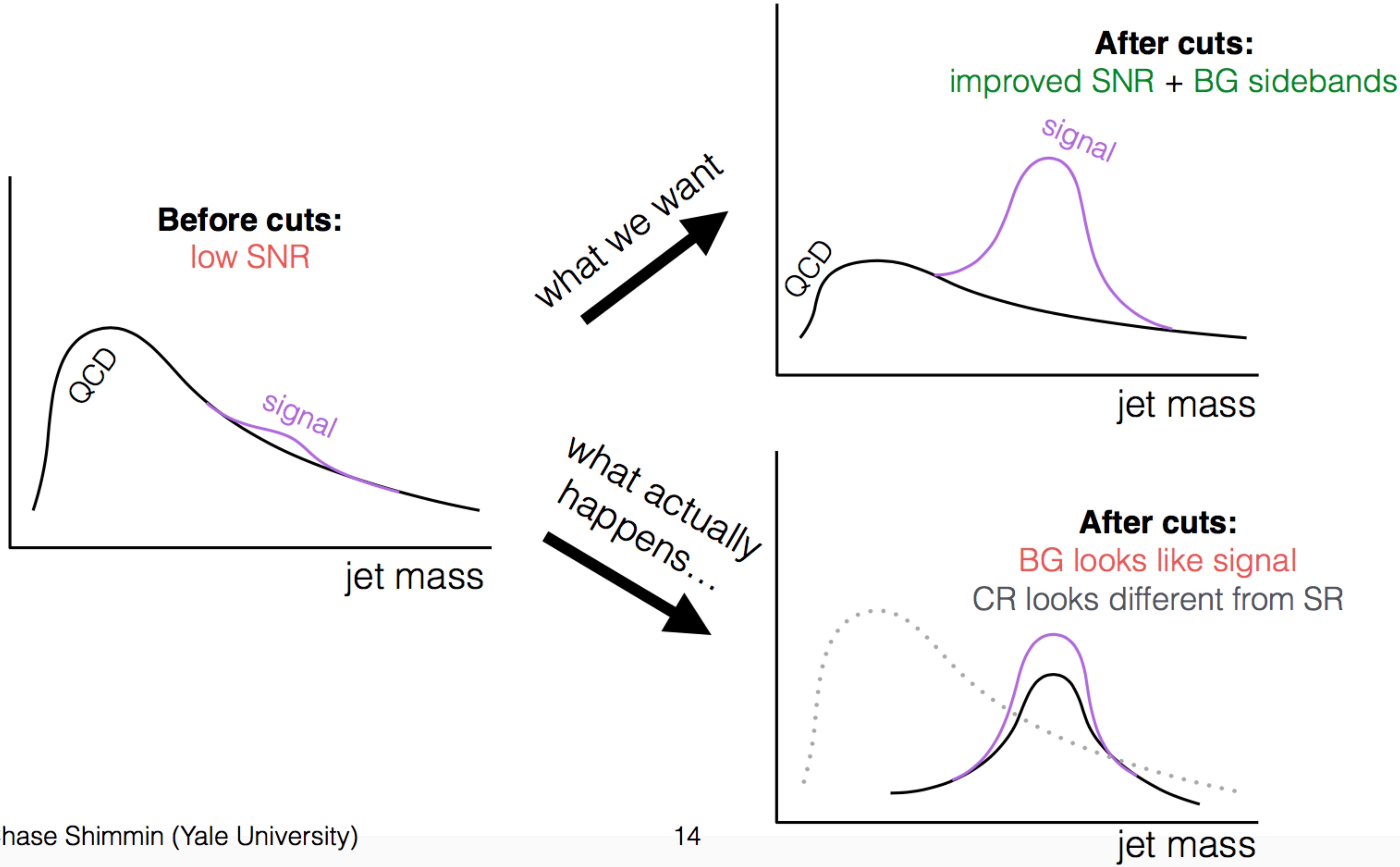



- Mistagging efficiency vs. b-jet efficiency
- Solid lines represent efficiencies with present ML-based method
  Statistical uncertainties shown as width of line
- Dashed lines show conventional, cut-based performance
  (cf. arXiv:1605.00143)

The present ML-assisted tagging method is very promising, compared to conventional method
– mistagging efficiency lower for c- and udsg-jets
– mistagging efficiencies rise less steep when considering higher b-jet tagging efficiency

# Mass Correlation

## Correlation with the observable of interest is bad!



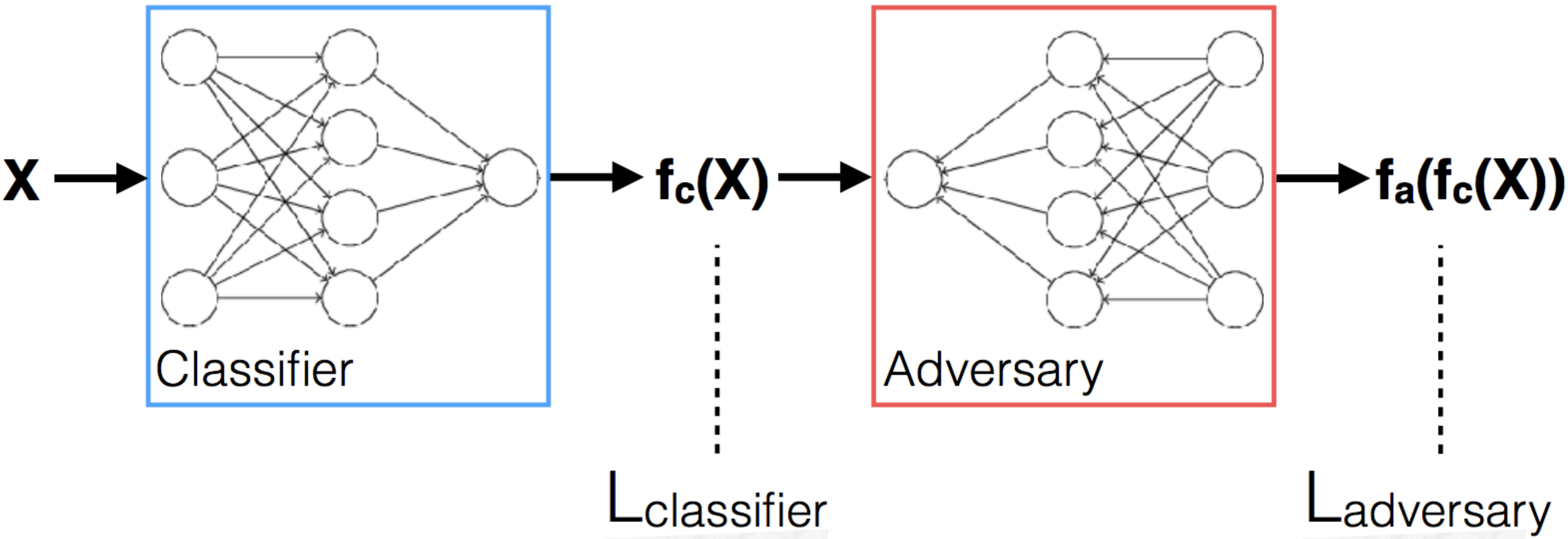
# Adversarial Decorrelation

**Simultaneously minimize:**

$$L_{adversary}$$

and

$$L_{tagger} = L_{classifier} - \lambda L_{adversary}$$




(distinguishes signal vs bg)

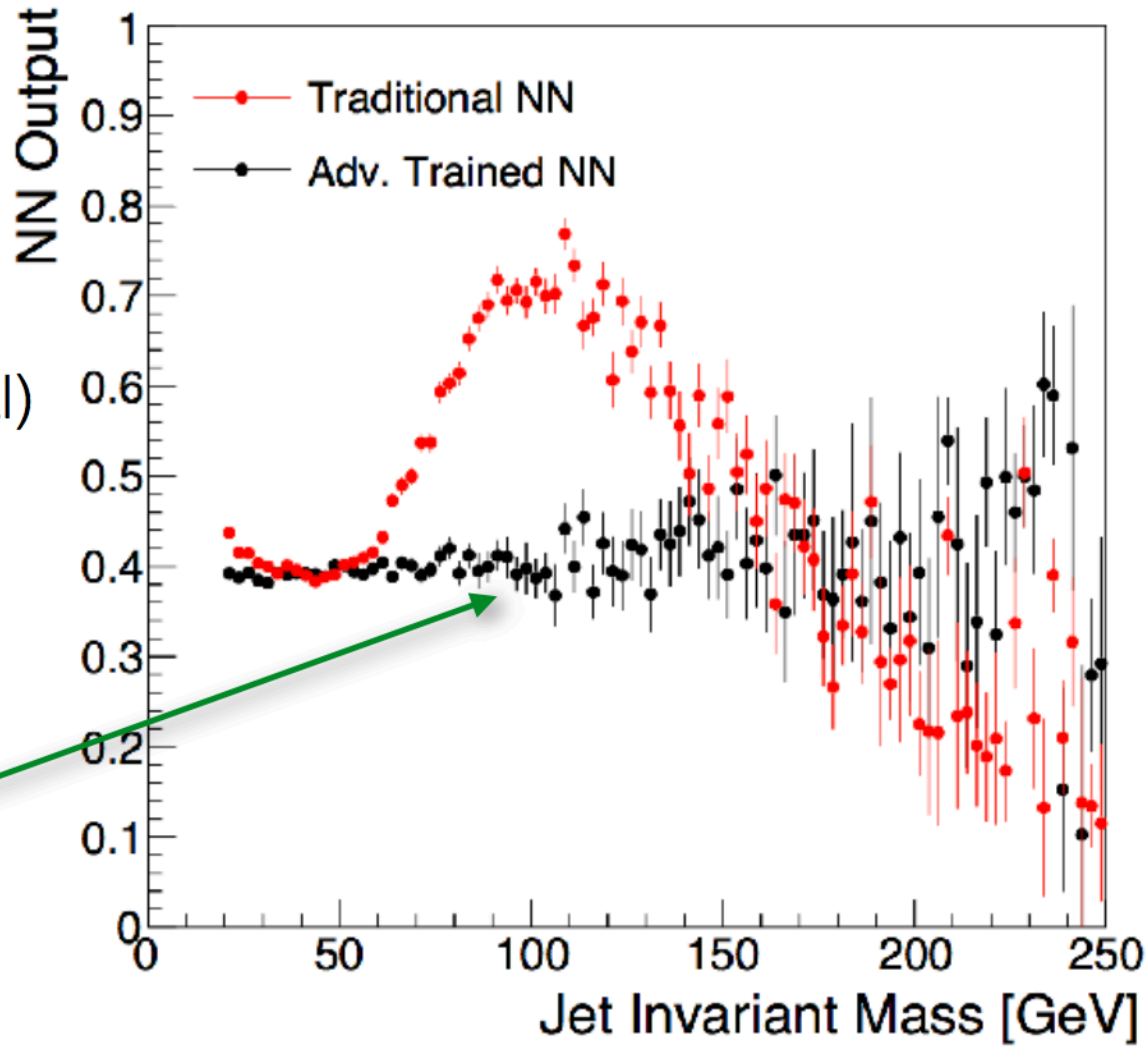
(predicts mass)

# Results

Training on ~200k MC events:

Sherpa γ+jet (BG)
MG5 γ+Z' (Signal)
Pythia + Delphes (Both)

✓ Tagger profile much flatter



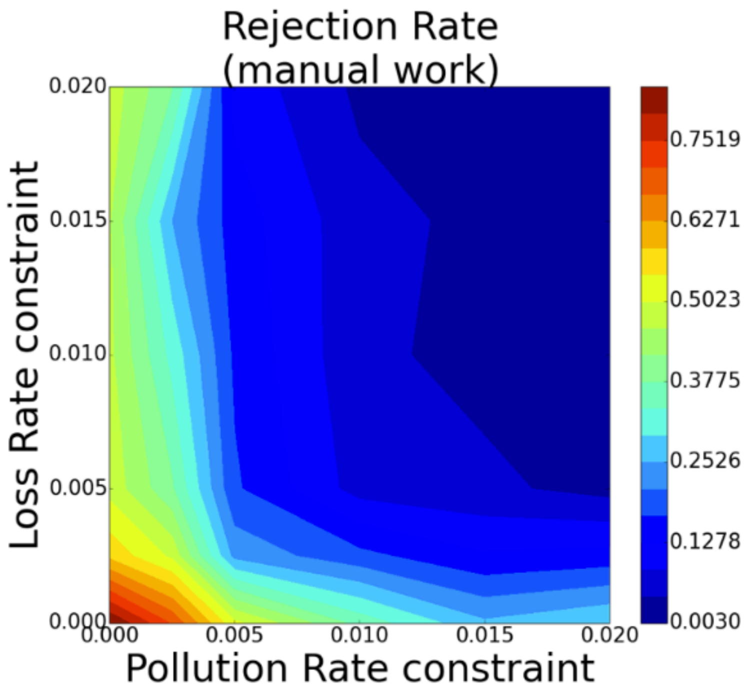
# DQM and QA

# DQM and QA

- **DQM and QA** are almost text book examples for application of ML
- Many **variables** (and **correlations**) need to be monitored
- **Traditionally**: human looks at histograms and trending plots, compares to reference and expectations
  - These tasks are easily automatized
- **Machine learning** can
  - use all data (variables) simultaneously
  - use more abstract representation of the data
- **Other experiments** (CMS/LHCb) are already developing/using these systems
  - Partnership with **industry** (Yandex, IBM)
- See also IML meeting on anomaly detection last year
  - https://indico.cern.ch/event/532992/

# CMS data certification / anomaly detection

Case




**80%**

**saving on manual work on data certification tasks**

## Task

Traditionally, quality of the data at CMS experiment is determined manually. It requires considerable amount of human efforts;

## Data used

— CERN open data portal 2010;

— Features: Particle flow jets, Calorimeter Jets, Photons, Muons;

— The dataset was labeled by CMS experts (~3 FTEs).

## ML Metrics

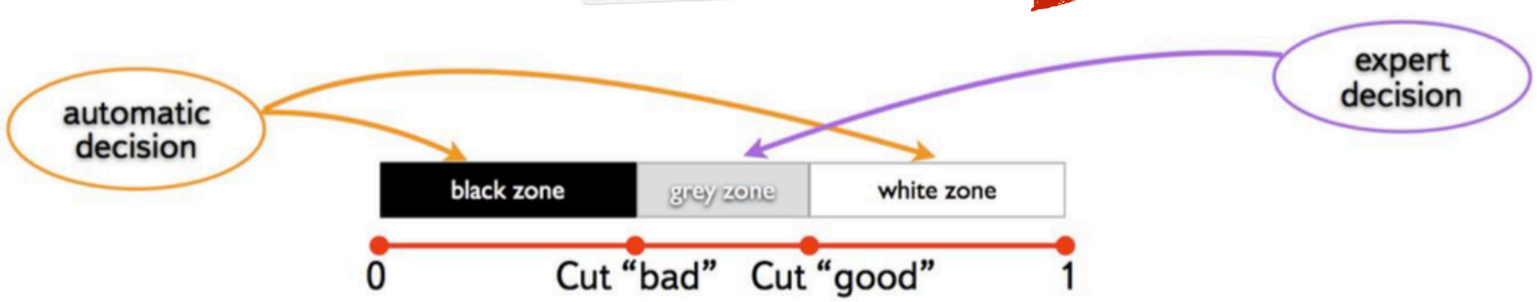
— ROC AUC, precision

## Result

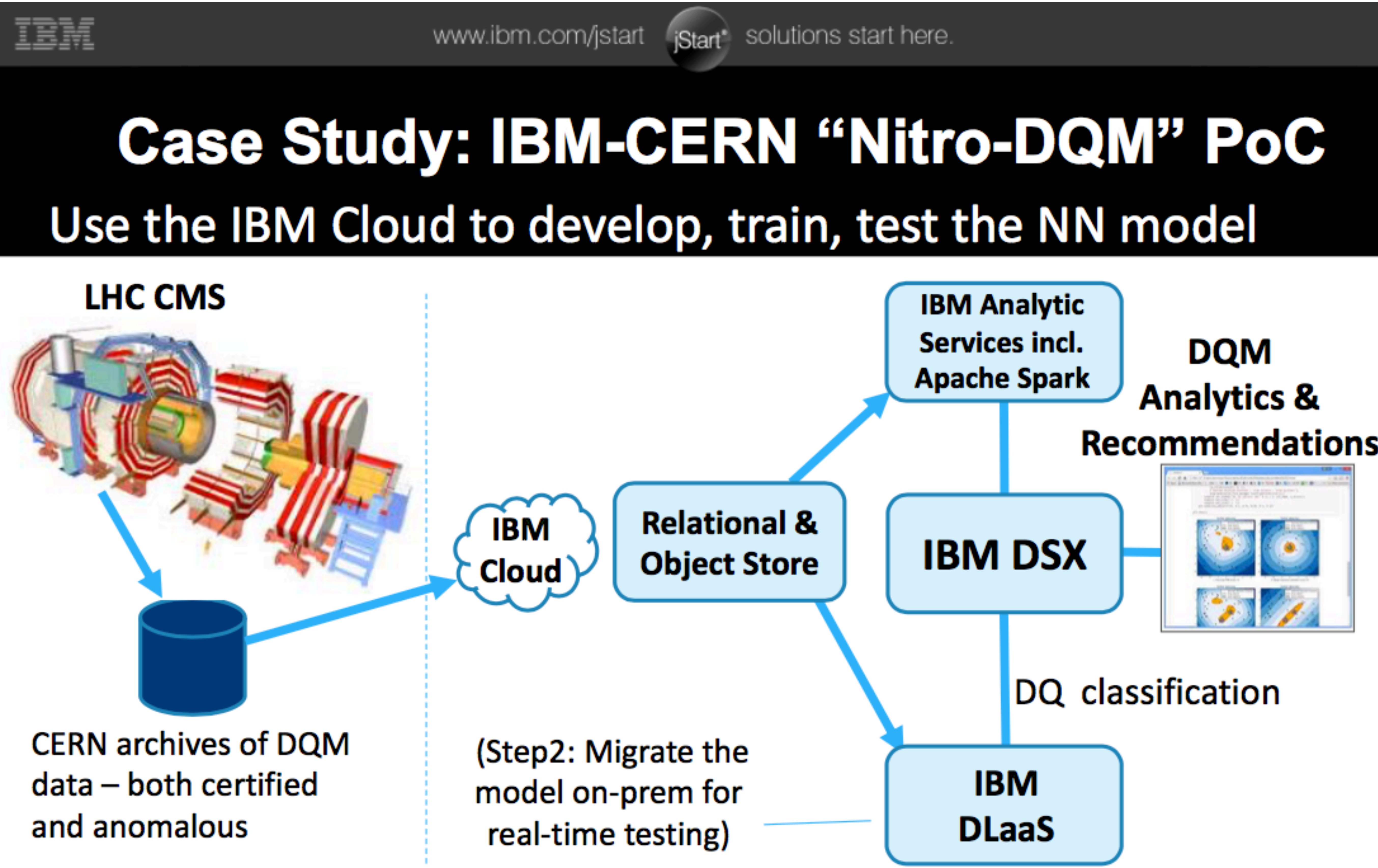
— ~80% saving on manual work is feasible for Pollution & Loss rate of 0.5%.

— Next steps: adopt technique for 2016 data & run in production

— http://bit.ly/2I0MLiN


Yandex Talk

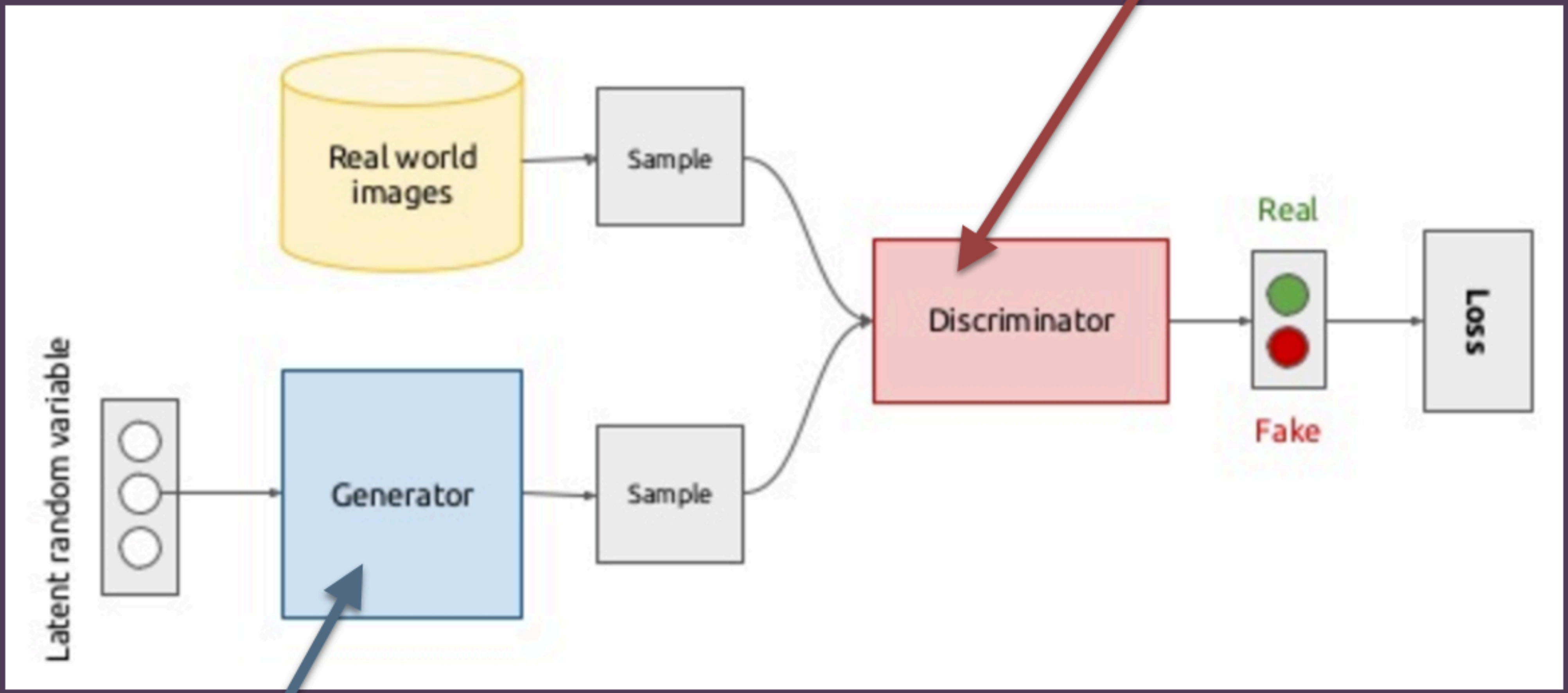

IBM
www.ibm.com/jstart
jStart
solutions start here.
Case Study: IBM-CERN "Nitro-DQM" PoC
Use the IBM Cloud to develop, train, test the NN model
LHC CMS
CERN archives of DQM data – both certified and anomalous
IBM Cloud
Relational & Object Store
IBM Analytic Services incl. Apache Spark
IBM DSX
IBM DLaaS
DQM Analytics & Recommendations
DQ classification
(Step2: Migrate the model on-prem for real-time testing)

IBM Talk

# Fast simulation

# Generative Adversarial Networks

tries to distinguish real images from generated images




tries to turn noise into credible samples

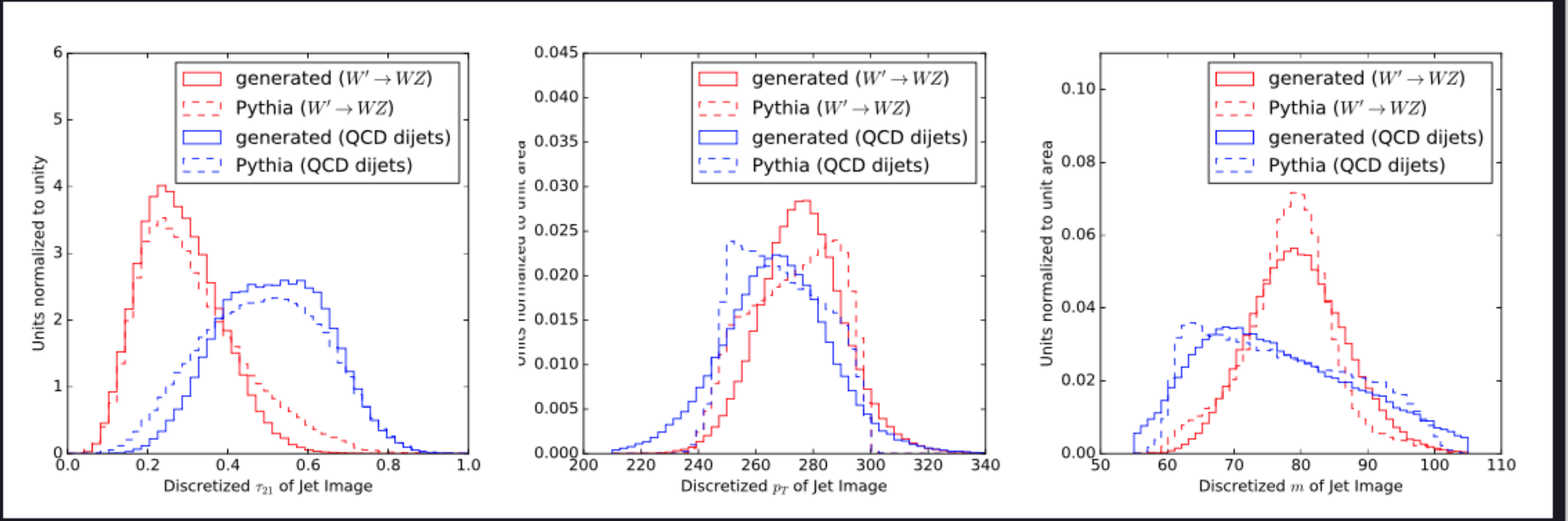
M. Paganini

# Physical Distributions

*Check*: does the LAGAN recover the true data distribution as projected onto a set of meaningful 1D manifolds? ✓

(Fast simulation of physics signals)

— signal
— background

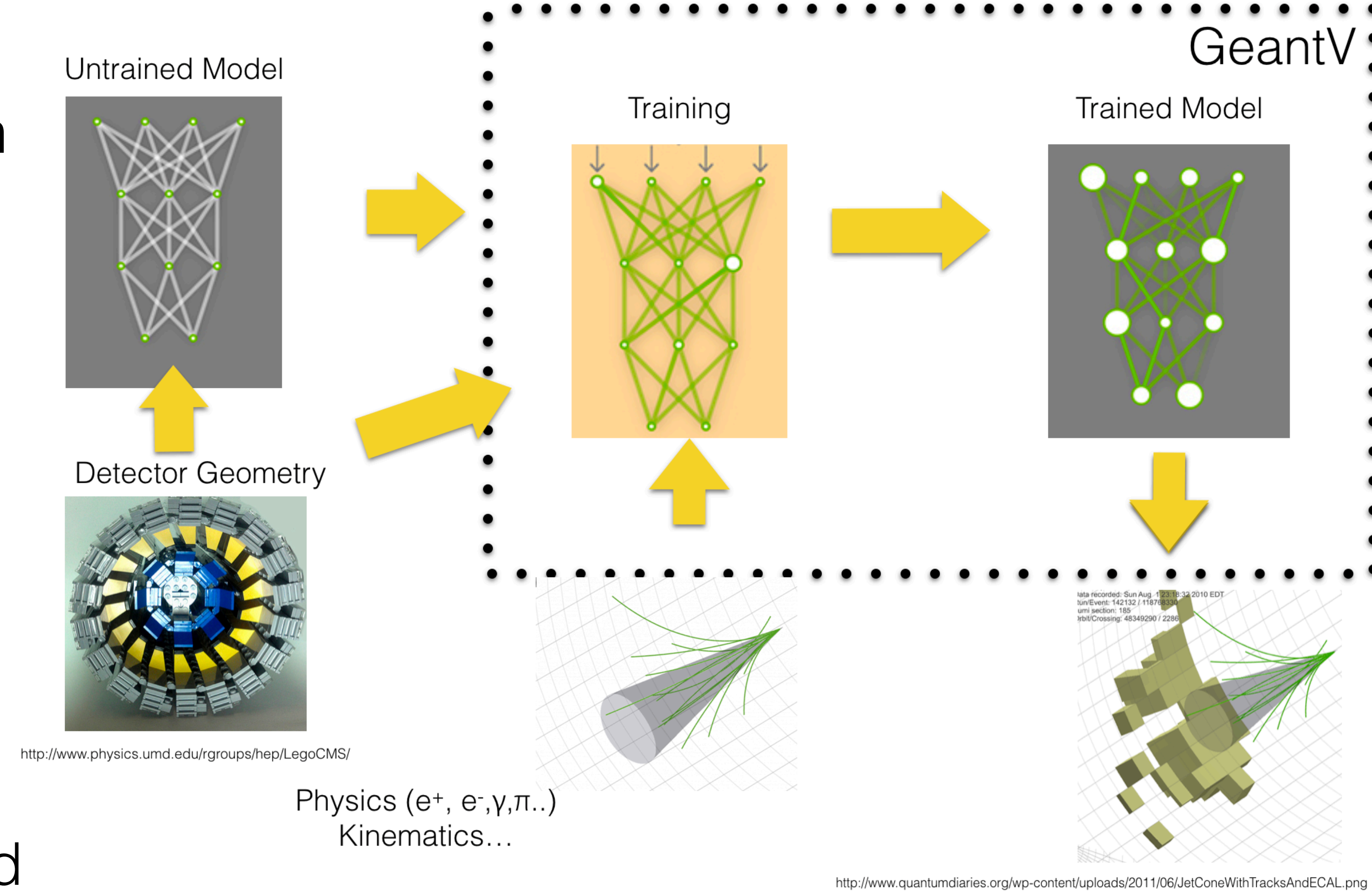


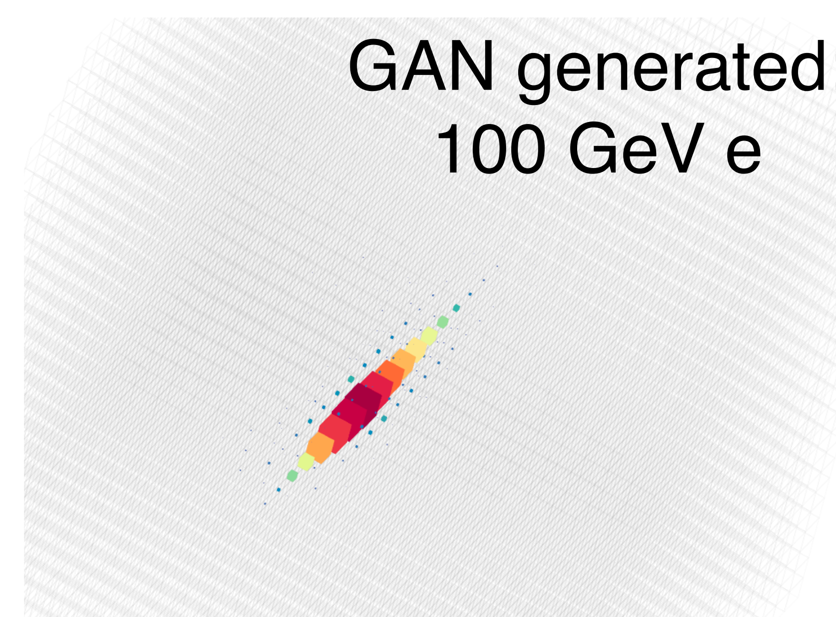

M. Paganini

# ML/DL engine for fastsim in GeantV

- Integrate inference (and training) step in GeantV
- Provide a configurable interface
- Train on full simulation
  - Test training on real data
- Test different techniques/models
  - Multi Objective regression, Feature extraction
  - Predictive Clustering Trees & Standard Perceptron (TMVA)
  - Generative adversarial networks (GANs)
- Later: embedded algorithm for hyper-parameters tuning and meta-optimization
- Possibly back-ported to Geant4

- Ex. first 3D images of single particle showers in LCD ECAL obtained training GAN




http://geant.cern.ch
S. Vallecorsa, https://indico.cern.ch/event/623453/

# Summary

- The **IML workshop** was mostly dedicated to a **specific problem** (tagging)
- **Many** other **areas of application** for ML, potentially useful for ALICE
- We are using ML less heavily than other experiments: many **low hanging fruits** still available
  - Good improvements with relatively little effort possible
- Some ML culture is starting to spread within the experiment (good!)
  - There will be an **ALICE ML meeting** during the physics week, slot to be announced
- Potential for **partnership with industry**: should we be more proactive?

*Thank you*