Deep Learning

Gregor Kasieczka
(gregor.kasieczka@uni-hamburg.de)

BOOST 2018
2018-08-18

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

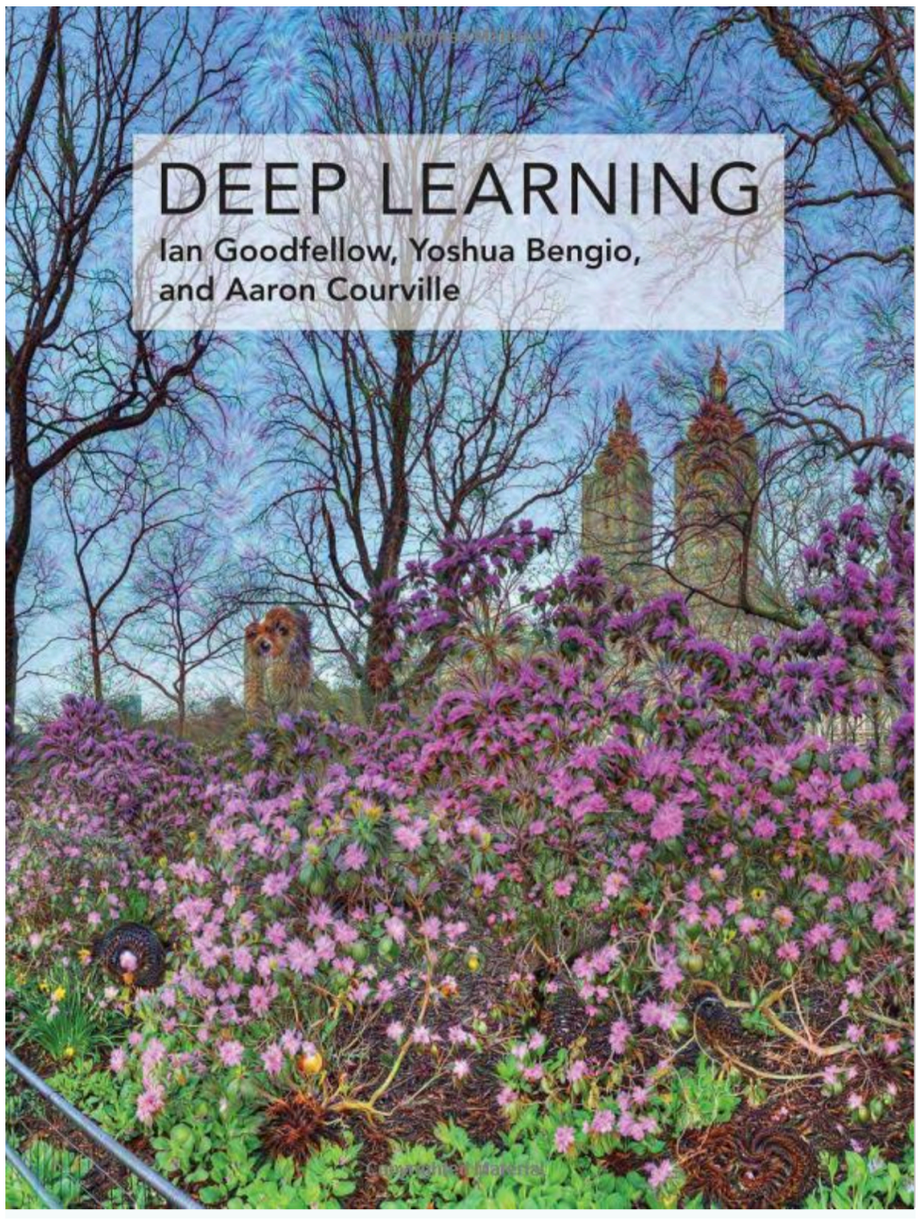
Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

Bundesministerium für Bildung und Forschung

Disclaimer

- This is a shortened set of slides used in an introduction block-lecture for graduate students in Freiburg. Included are the basic topics.
- The material was compiled Spring 2018 and therefor some of the recent developments (especially material shown at this BOOST) are not yet included
- If you need a sample to play with: https://goo.gl/XGYju3

Literature



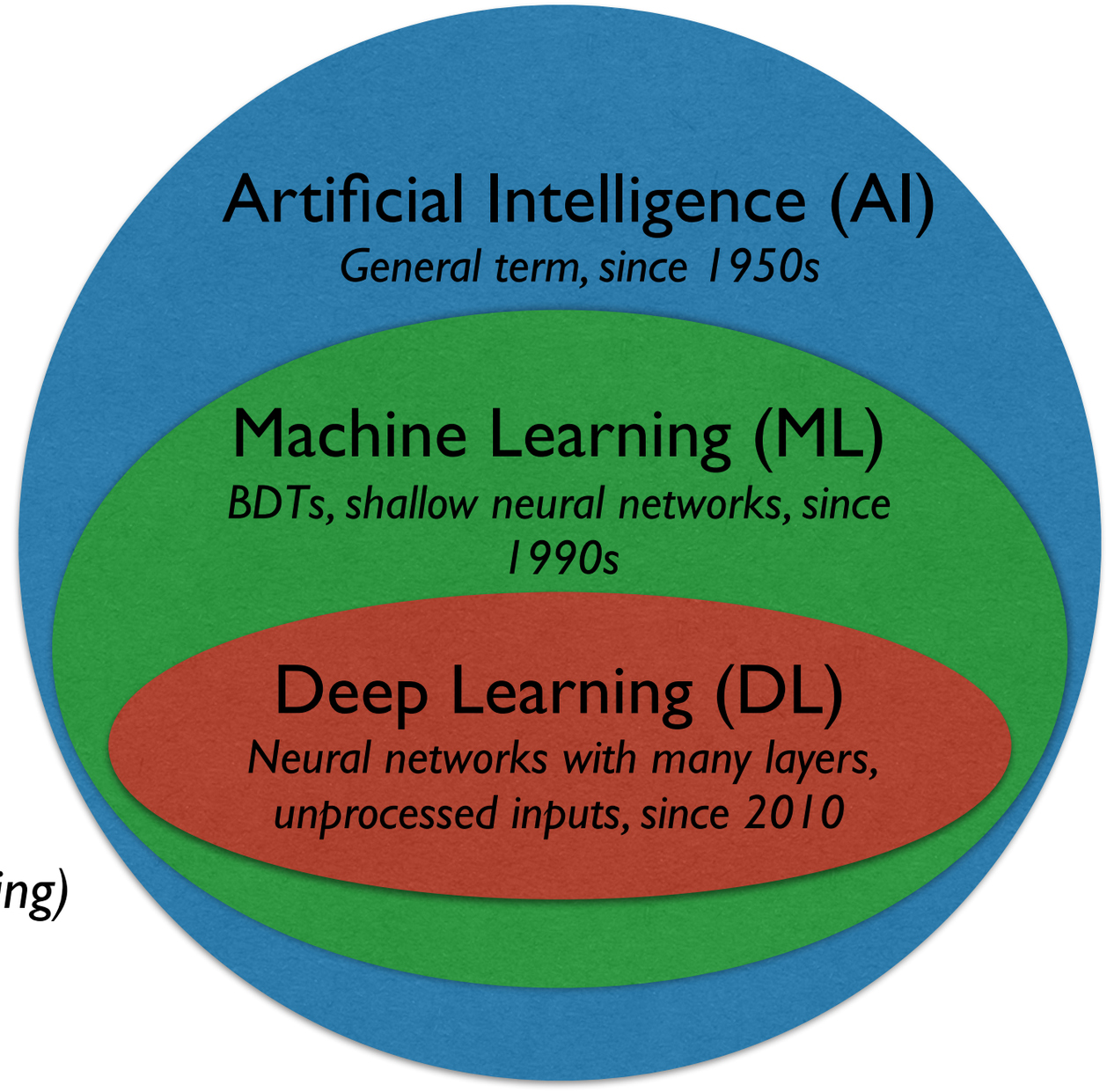
Free online version:
http://www.deeplearningbook.org/

- Publications on arXiv:
  - stat.ml and cs.lg
  - Prepare for O(50) papers/day
- Big conferences (outside physics)
  - ICML and NIPS

Basics

Terminology

- What do we want to learn?
  - Classification (*Cat or dog?*)
  - Generation (*New pictures of cats*)
  - Regression (*How old is the cat?*)
  - Compression (*Smaller cats*)
  - ?
- What do we have available?
  - Labelled examples (*supervised learning)*
  - Limited labelled examples (*weakly supervised learning*)
  - No examples (*unsupervised learning*)

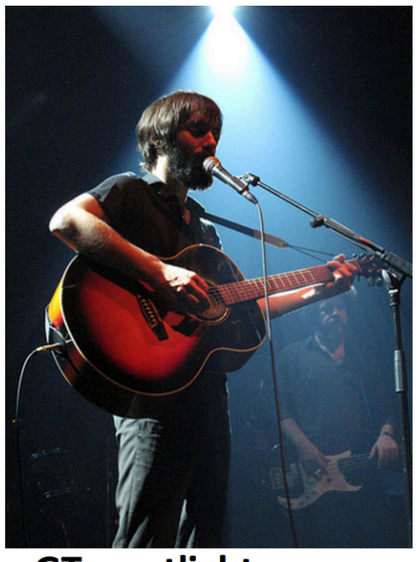


Humans vs Machines

- 2015 **Image Classification**:
  - K. He et al (Microsoft Research), *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,* 1502.01852
- 2016 **Go**:
  - Alpha Go (D. Silver et al, *Mastering the game of Go with deep neural networks and tree search*, Nature 529, pp484–489 and D. Silver et al *Mastering the game of Go without human knowledge,* Nature 550, pp354–359)
- 2016 **Speech recognition:**
  - W Xiong et al (Microsoft Research) *Achieving Human Parity in Conversational Speech Recognition,* 1610.05256
- 2017 **Poker** (heads-up no-limits Texas Hold'em):
  - N Brown and T Sandholm, *Superhuman AI for heads-up no-limit poker: Libratus beats top professionals, Science 359, Issue 6374, pp418-424*
- 2018 ***Translation*** (Chinese-English)
  - H H Awadalla et al (Microsoft AI & Research) *Achieving Human Parity on Automatic Chinese to English News Translation*
- 20?? **Particle Physics**
  - *to be seen*

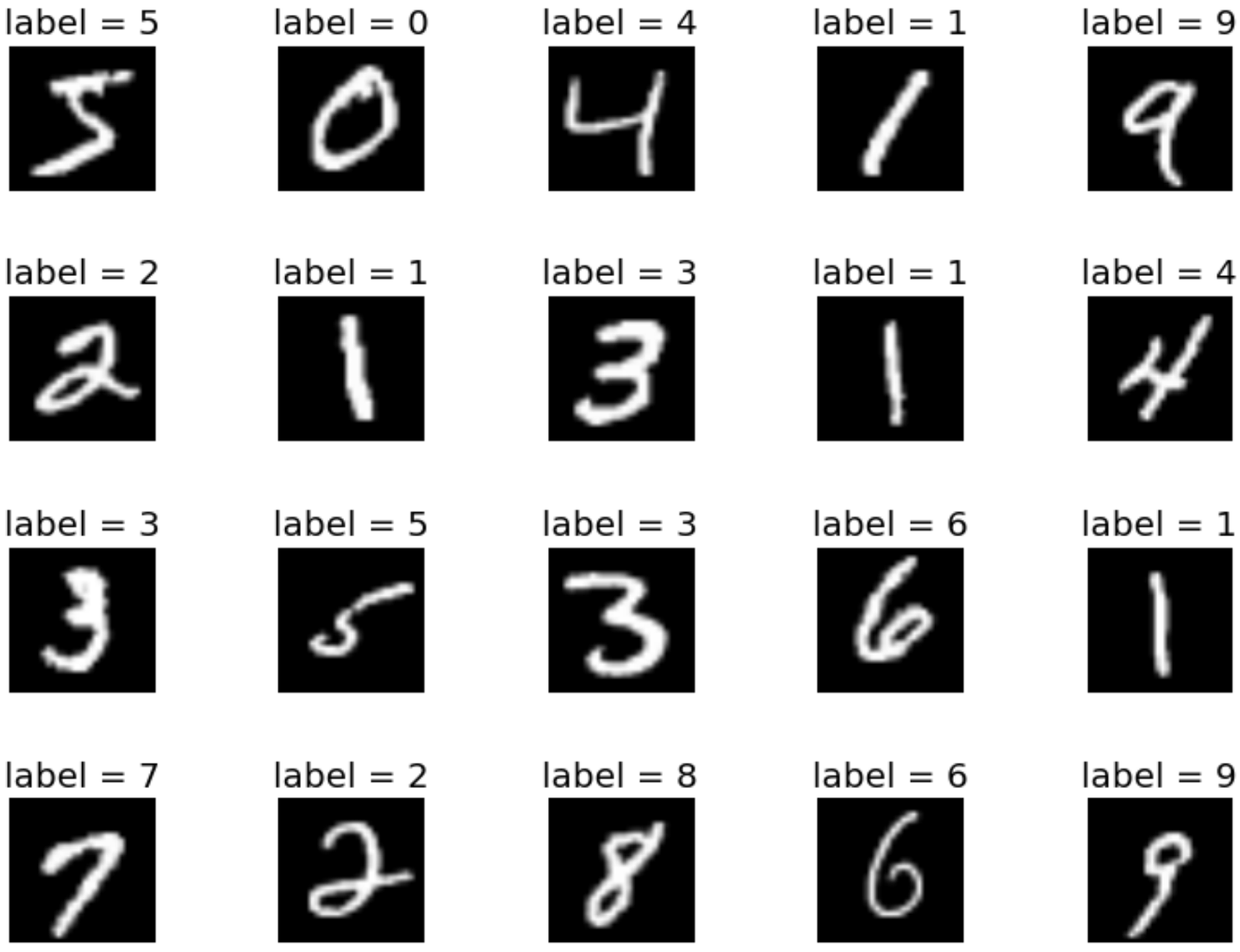
(from 1502.01852)

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog



GT: spotlight
1: acoustic guitar
2: stage
3: microphone
4: electric guitar
5: banjo

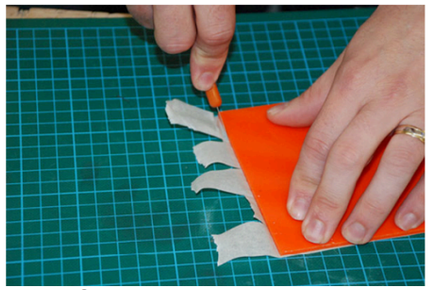
Image Datasets: MNIST



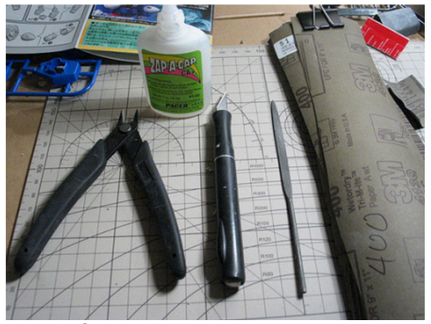
- http://yann.lecun.com/exdb/mnist/
- *Mixed NIST*
- Grayscale, 28x28 pixels

ImageNet

GT: letter opener
1: drumstick
2: candle
3: wooden spoon
4: spatula
5: ladle



GT: letter opener
1: Band Aid
2: ruler
3: rubber eraser
4: pencil box
5: wallet



GT: letter opener
1: fountain pen
2: ballpoint
3: hammer
4: can opener
5: ruler

GT: spotlight
1: grand piano
2: folding chair
3: rocking chair
4: dining table
5: upright piano

GT: spotlight
1: acoustic guitar
2: stage
3: microphone
4: electric guitar
5: banjo

GT: spotlight
1: altar
2: candle
3: perfume
4: restaurant
5: confectionery

- 14M hand annotated images
- 20k categories
- http://image-net.org

GT: horse cart
1: horse cart
2: minibus
3: oxcart
4: stretcher
5: half track

GT: birdhouse
1: birdhouse
2: sliding door
3: window screen
4: mailbox
5: pot

GT: forklift
1: forklift
2: garbage truck
3: tow truck
4: trailer truck
5: go-kart

GT: coucal
1: coucal
2: indigo bunting
3: lorikeet
4: walking stick
5: custard apple

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog

GT: yellow lady's slipper
1: yellow lady's slipper
2: slug
3: hen-of-the-woods
4: stinkhorn
5: coral fungus

GT: torch
1: stage
2: spotlight
3: torch
4: microphone
5: feather boa

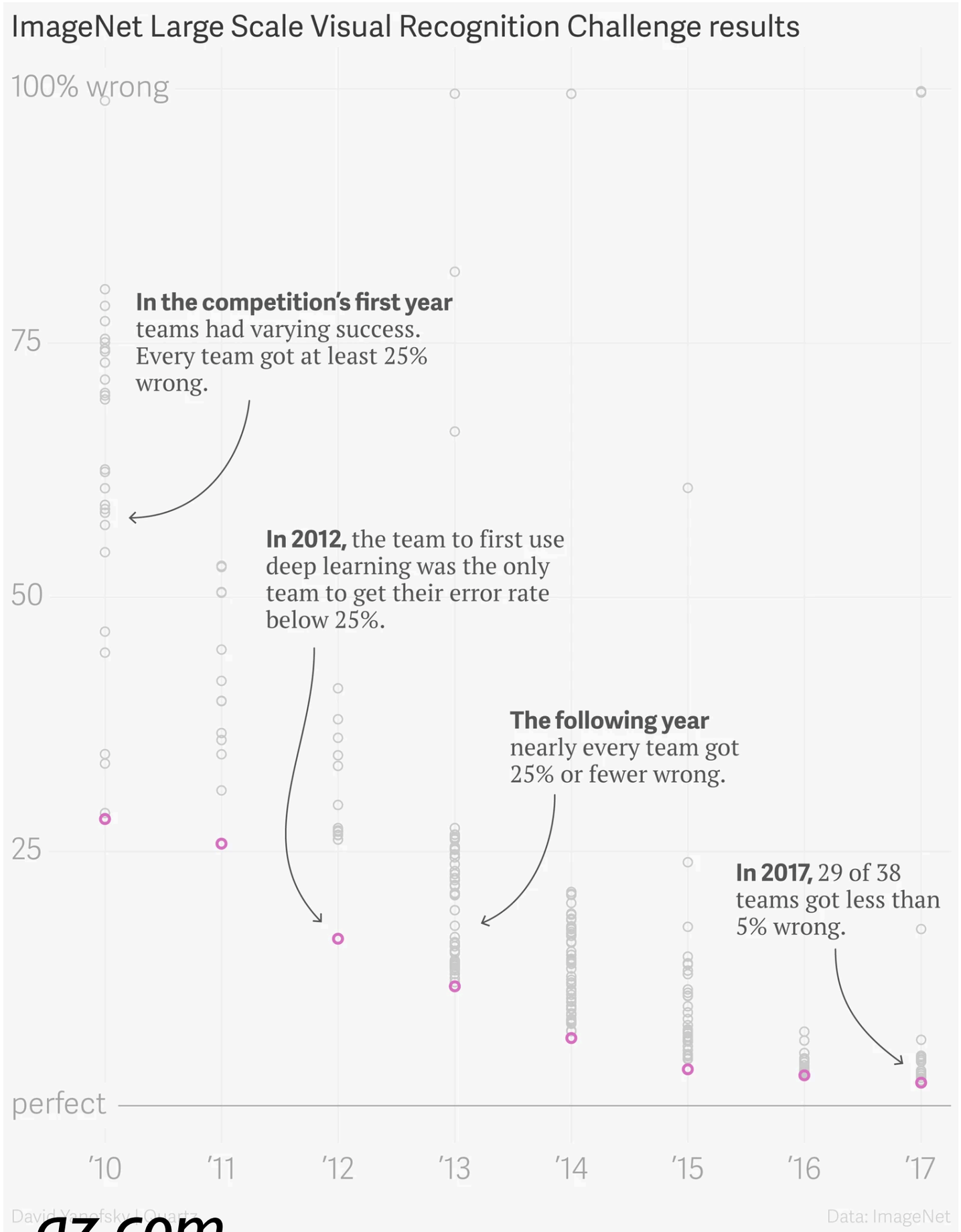
GT: banjo
1: acoustic guitar
2: shoji
3: bow tie
4: cowboy hat
5: banjo

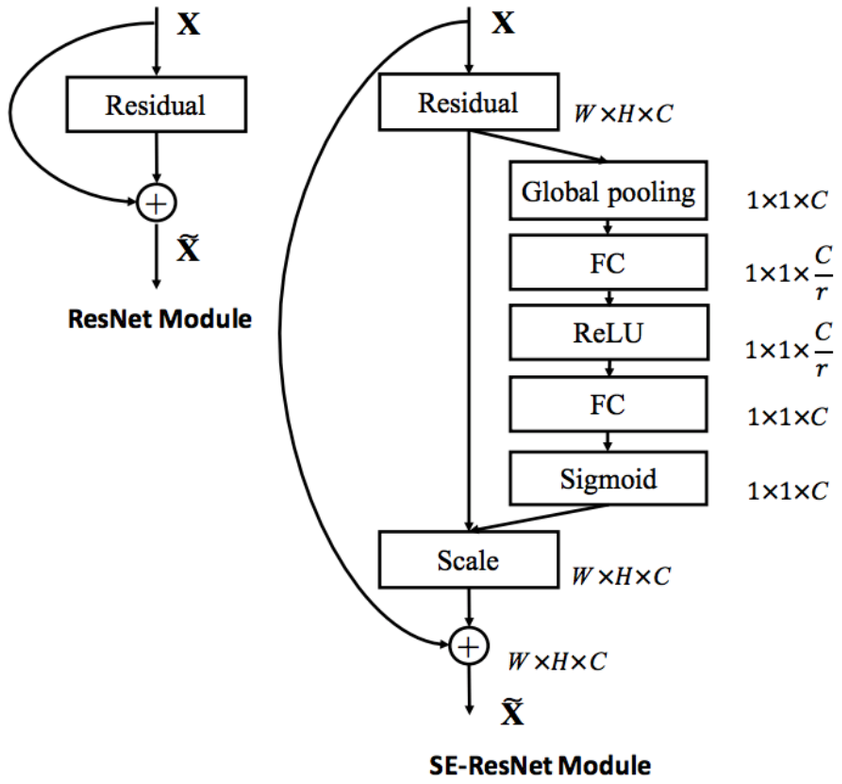
GT: go-kart
1: go-kart
2: crash helmet
3: racer
4: sports car
5: motor scooter

GT: mountain tent
1: sleeping bag
2: mountain tent
3: parachute
4: ski
5: flagpole

GT: geyser
1: geyser
2: volcano
3: sandbar
4: breakwater
5: leatherback turtle

GT: microwave
1: microwave
2: washer
3: toaster
4: stove
5: dishwasher





SE-ResNeXt-50 (32×4d)	
$\begin{bmatrix} conv, 1\times1, 128 \\ conv, 3\times3, 128 \\ conv, 1\times1, 256 \\ fc, [16, 256] \end{bmatrix}$	$C = 32$ $\times 3$
$\begin{bmatrix} conv, 1\times1, 256 \\ conv, 3\times3, 256 \\ conv, 1\times1, 512 \\ fc, [32, 512] \end{bmatrix}$	$C = 32$ $\times 4$
$\begin{bmatrix} conv, 1\times1, 512 \\ conv, 3\times3, 512 \\ conv, 1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix}$	$C = 32$ $\times 6$
$\begin{bmatrix} conv, 1\times1, 1024 \\ conv, 3\times3, 1024 \\ conv, 1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix}$	$C = 32$ $\times 3$

qz.com

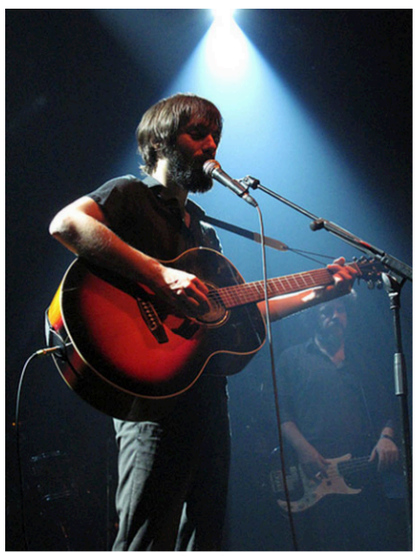
1709.01507

Humans vs Machines

- 2015 **Image Classification**:
  - K. He et al (Microsoft Research), *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,* 1502.01852
- 2016 **Go**:
  - Alpha Go (D. Silver et al, *Mastering the game of Go with deep neural networks and tree search*, Nature 529, pp484–489 and D. Silver et al *Mastering the game of Go without human knowledge,* Nature 550, pp354–359)
- 2016 **Speech recognition:**
  - W. Xiong et al (Microsoft Research) *Achieving Human Parity in Conversational Speech Recognition,* 1610.05256
- 2017 **Poker** (heads-up no-limits Texas Hold'em):
  - N Brown and T Sandholm, *Superhuman AI for heads-up no-limit poker: Libratus beats top professionals, Science 359, Issue 6374, pp418-424*
- 2018 ***Translation*** (Chinese-English)
  - H H Awadalla et al (Microsoft AI & Research) *Achieving Human Parity on Automatic Chinese to English News Translation*
- 20?? **Particle Physics**
  - *to be seen*

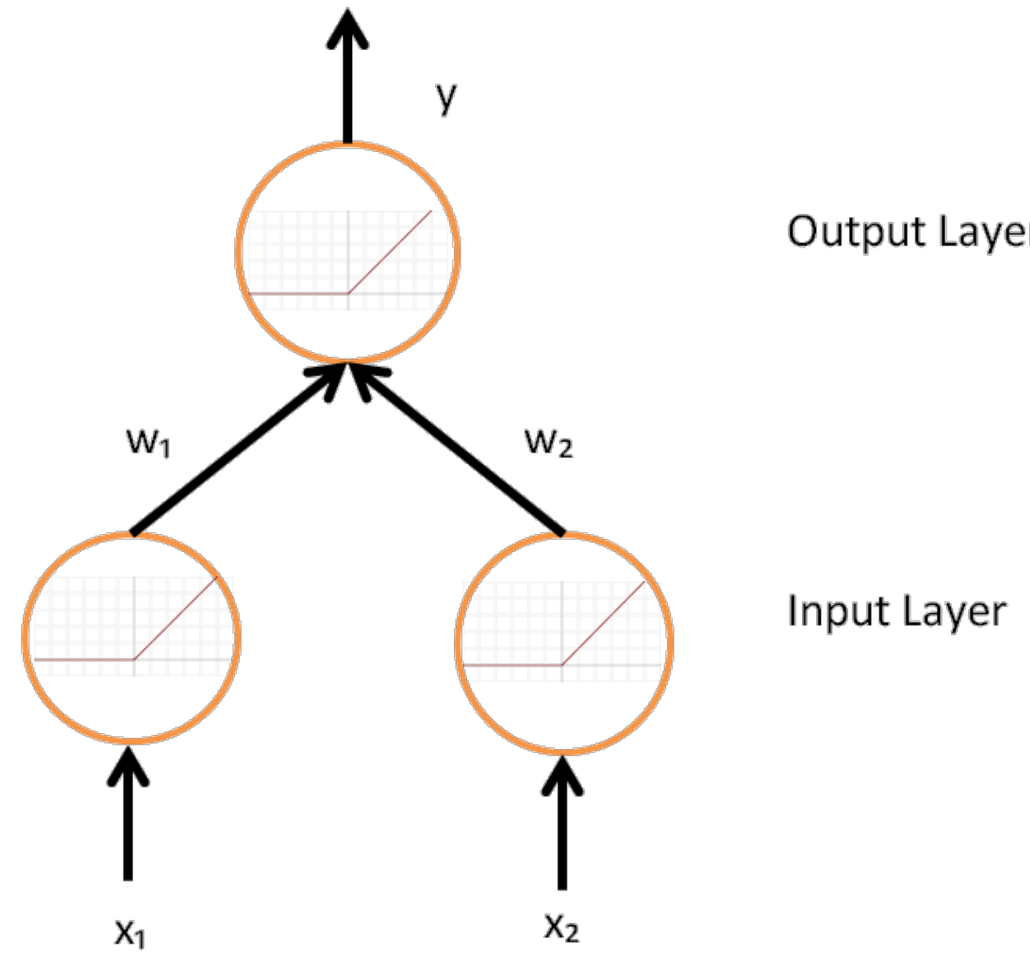
(from 1502.01852)

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog



GT: spotlight
1: acoustic guitar
2: stage
3: microphone
4: electric guitar
5: banjo

A Very Simple Network



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

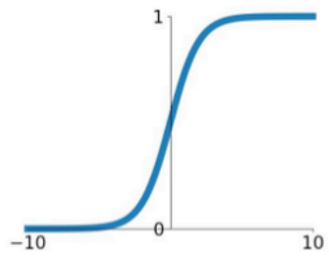
$$f(x) = \Theta(x) \cdot x$$

Activation Functions

Activation Functions

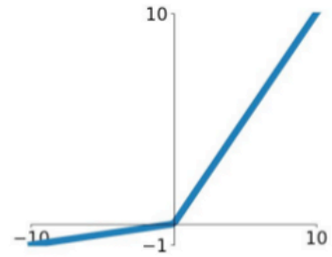
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



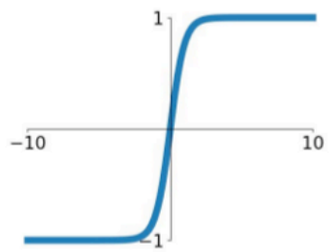
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

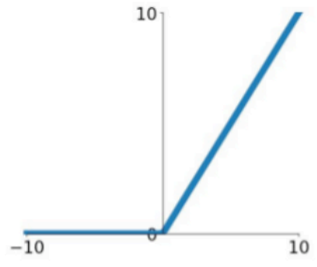


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

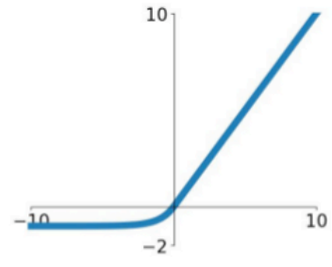
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Softmax
(for final classification layer)

$$\sigma : \mathbb{R}^K \to (0,1)^K$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

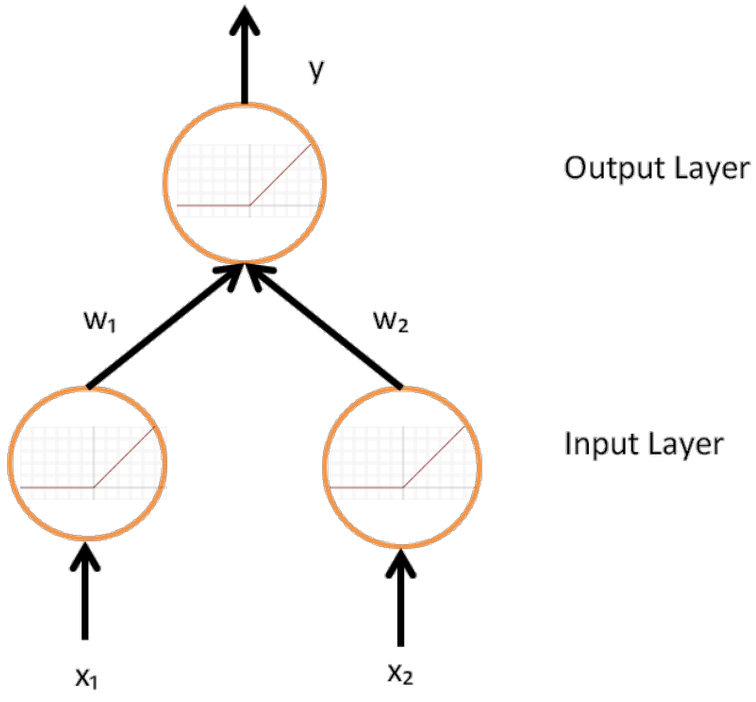
How do networks learn?

- *Backpropagation + Gradient descent*
- Pass input (x_1, x_2) to ANN
- Calculate output ($\hat{y}$) and difference to true value (y)
  This is the loss function *L*
- Find gradient of loss function with respect to weights
- Use gradient to find new weights

Regression Problem:

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$w_{t+1} = w_t - \eta \frac{\partial L}{\partial w_t} \equiv w_t - \eta \nabla L(w_t)$$



Optimisers

$$w_{t+1} = w_t - \eta \nabla L(w_t)$$

(stochastic/batched) gradient descent

$$w_{t+1} = w_t - \eta \nabla L(w_t) + \alpha \Delta w_t$$

+ momentum term

$$
\begin{aligned}
m_{t+1} &= \beta_1 m_t + (1 - \beta_1) \nabla L \\
v_{t+1} &= \beta_2 v_t + (1 - \beta_2) (\nabla L)^2 \\
\hat{m}_{t+1} &= \frac{m_{t+1}}{1 - \beta_1^t} \\
\hat{v}_{t+1} &= \frac{v_{t+1}}{1 - \beta_2^t} \\
w_{t+1} &= w_t - \eta \frac{\hat{m}_{t+1}}{\sqrt{\hat{m}_{t+1}} + \epsilon}
\end{aligned}
$$

Adam

(a good starting point)

Information content of single event

- Likely events should habe low information content
- Less likely events should have higher information content
- Independent events should have additive information

$$I(x) = -\log P(x)$$

Entropy

Boltzmann:

$$S = k_B \ln W$$

(W: number of micro states)

Gibbs:

$$S = -k_B \sum p_i \ln p_i$$

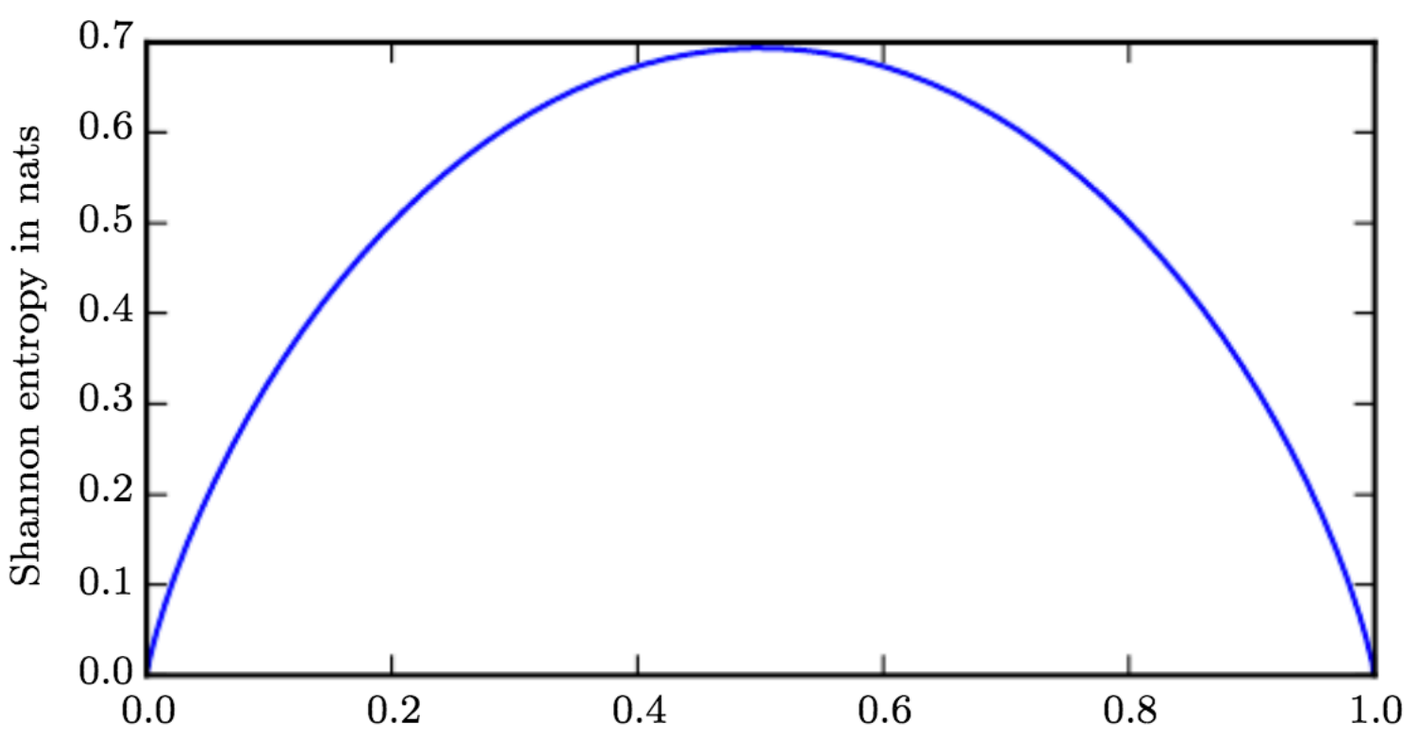
(p_i: probability of a micro state)

Shannon:

$$S = -\sum p_i \ln p_i$$

(p_i: probability of a measurement)

Entropy



Boltzmann:

$$S = k_B \ln W$$

(W: number of micro states)

Gibbs:

$$S = -k_B \sum p_i \ln p_i$$

(p_i: probability of a micro state)

Shannon:

$$S = -\sum p_i \ln p_i$$

(p_i: probability of a measurement)

Classification

$$S = -\sum p_i \ln p_i$$

- Entropy: *Optimal number of bits needed to encode when the probability distribution is known*

$$S = -\sum p_i \ln \hat{p}_i$$

- Cross Entropy: *We do not know the true probability*

$$L = \sum_{\text{Samples}} -y_s \ln \hat{y}_s - (1 - y_s) \ln(1 - \hat{y}_s)$$

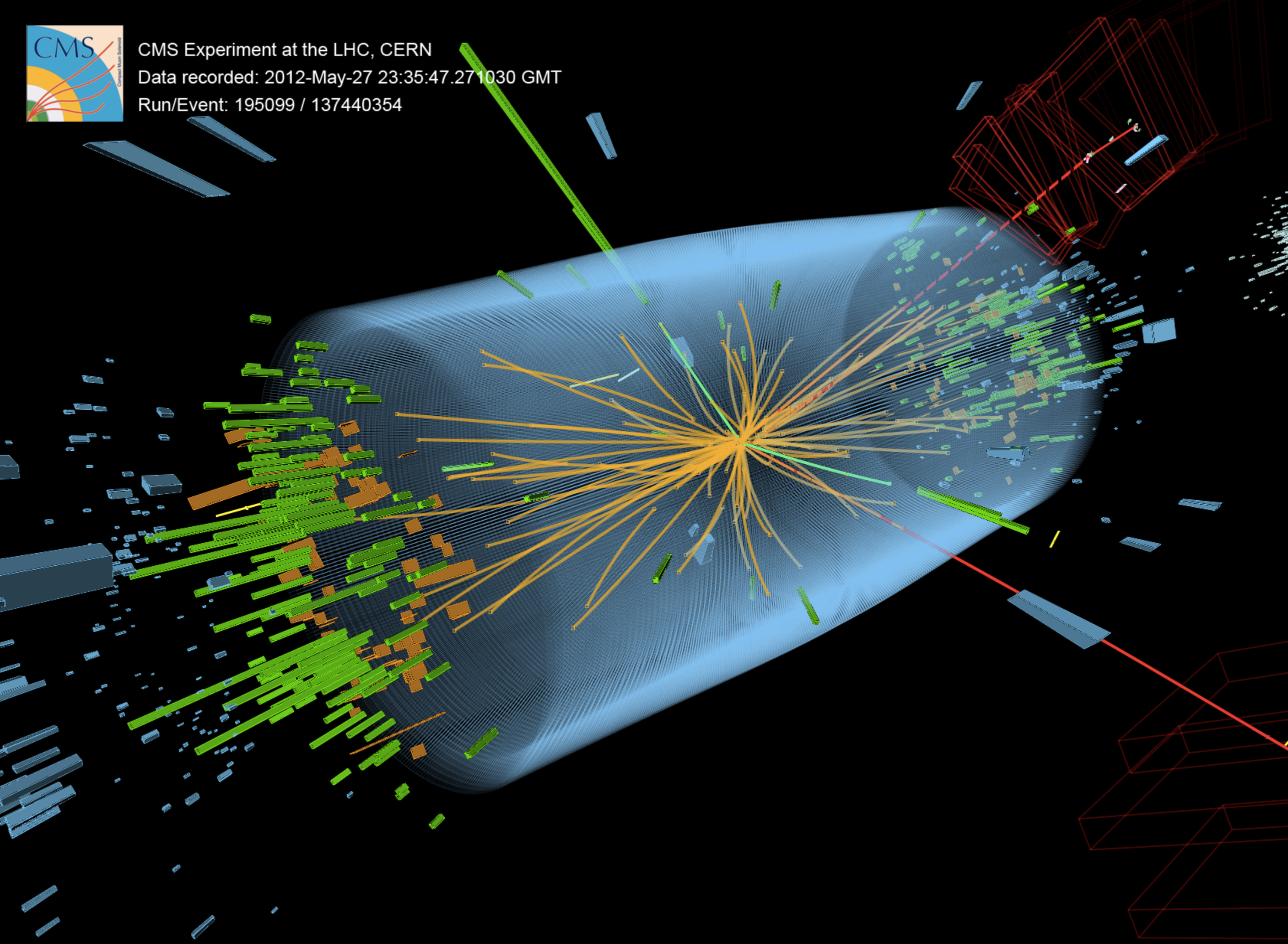
True class
image is cat: 0
image is dog: 1

Predicted class
DNN output between 0 and 1

Minimize cross entropy: approximate true distribution

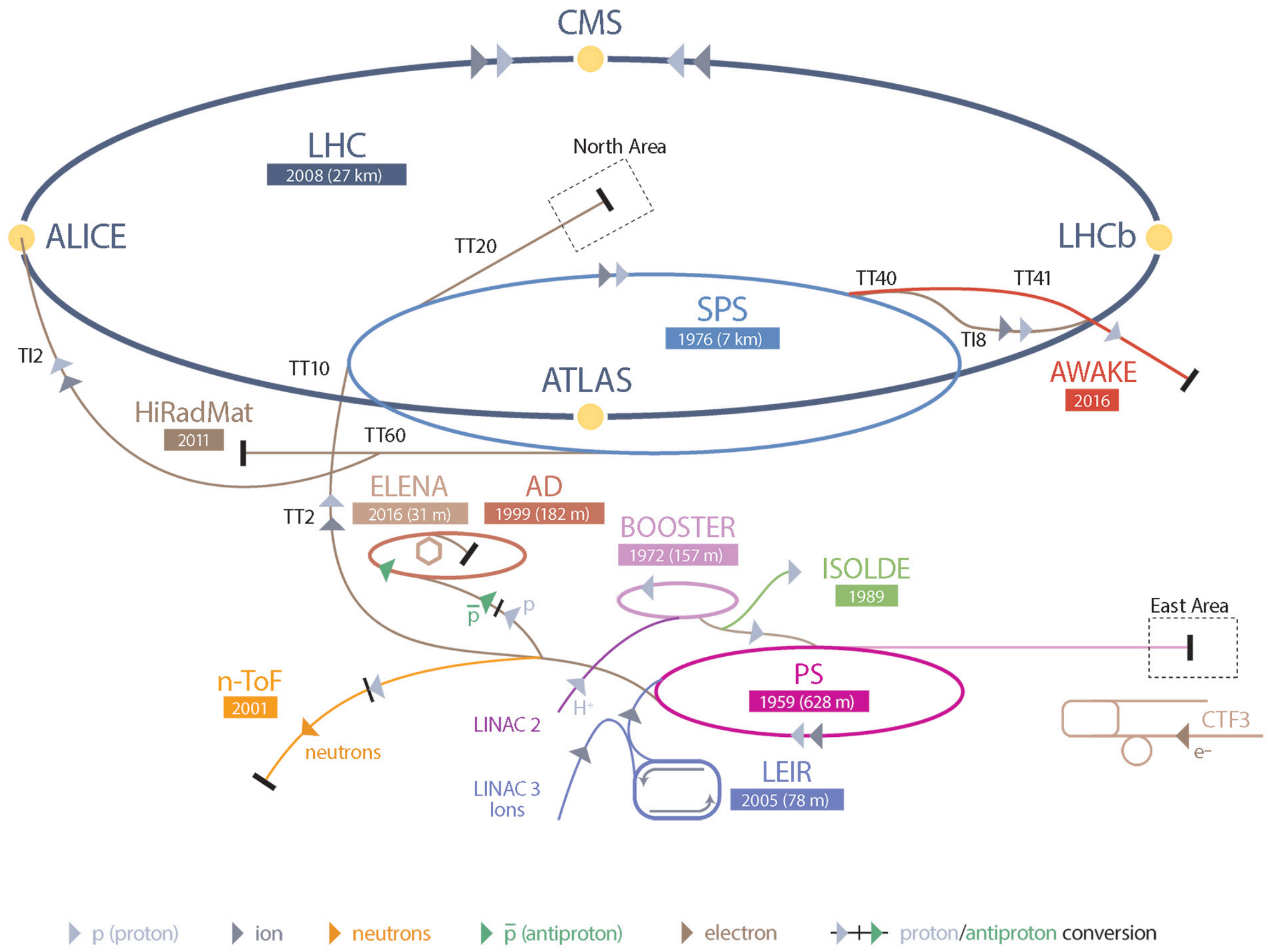
Physics

CMS Experiment at the LHC, CERN
Data recorded: 2012-May-27 23:35:47.271030 GMT
Run/Event: 195099 / 137440354

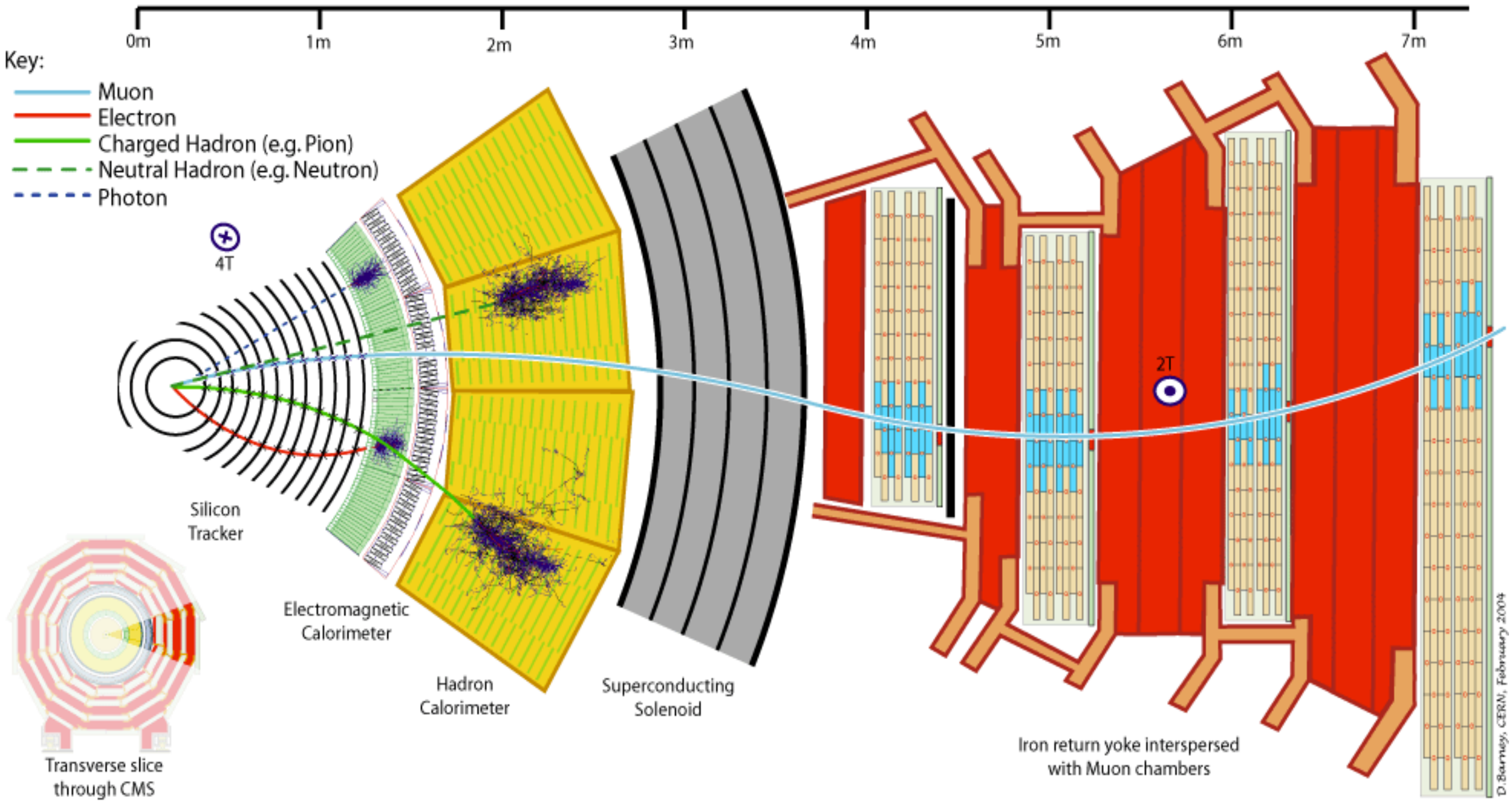


The LHC

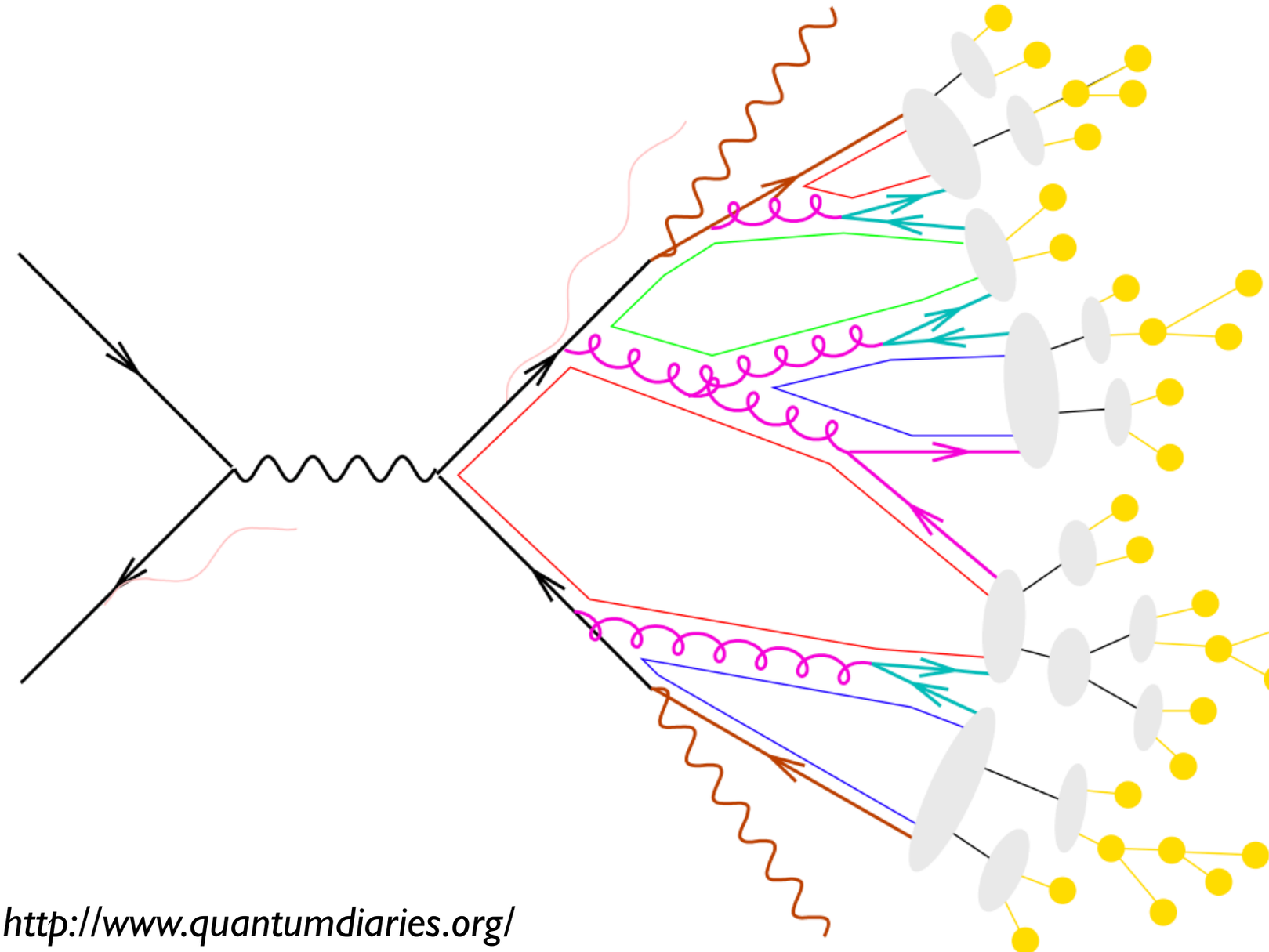
CERN's Accelerator Complex



CMS Experiment



Physics at the LHC



- hard scattering
- (QED) initial/final state radiation
- partonic decays, e.g. $t \to bW$
- parton shower evolution
- nonperturbative gluon splitting
- colour singlets
- colourless clusters
- cluster fission
- cluster $\to$ hadrons
- hadronic decays

http://www.quantumdiaries.org/

We want to infer underlying physics from measurements in the detector.
How can deep neural networks assist us?

Jet Clustering

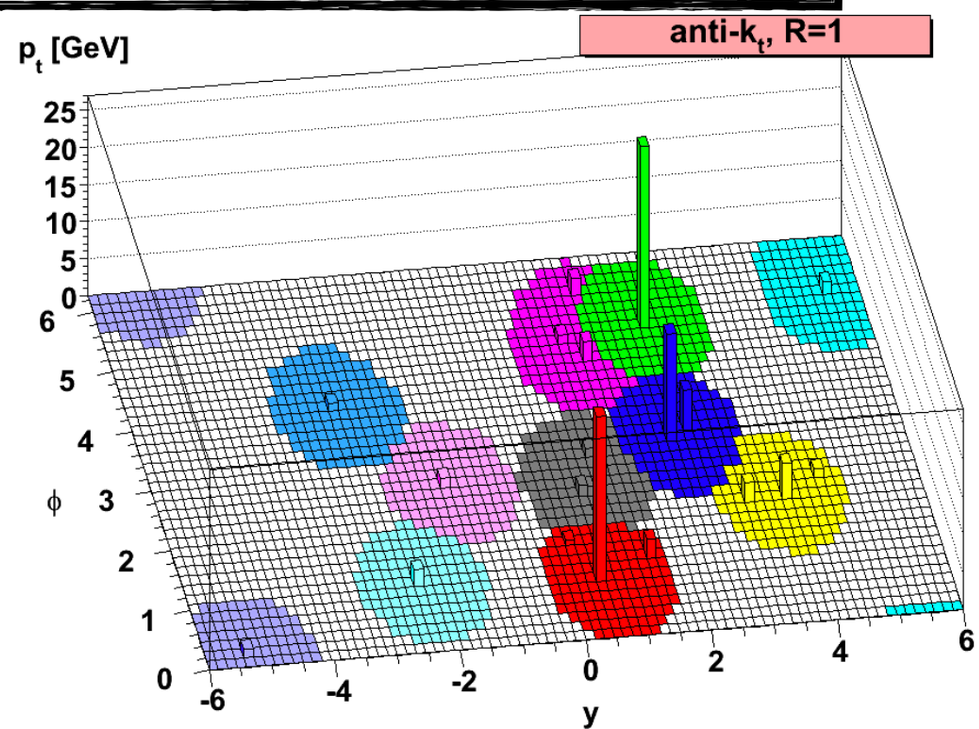
1112.4441
0005012
0802.1189

- Two approaches:
  - **Cone based** - Find stable axes of particles flow (ie. SisCone: 0704.0292)
  - **Sequential** - Pairwise combination of clusters according to a distance measure:

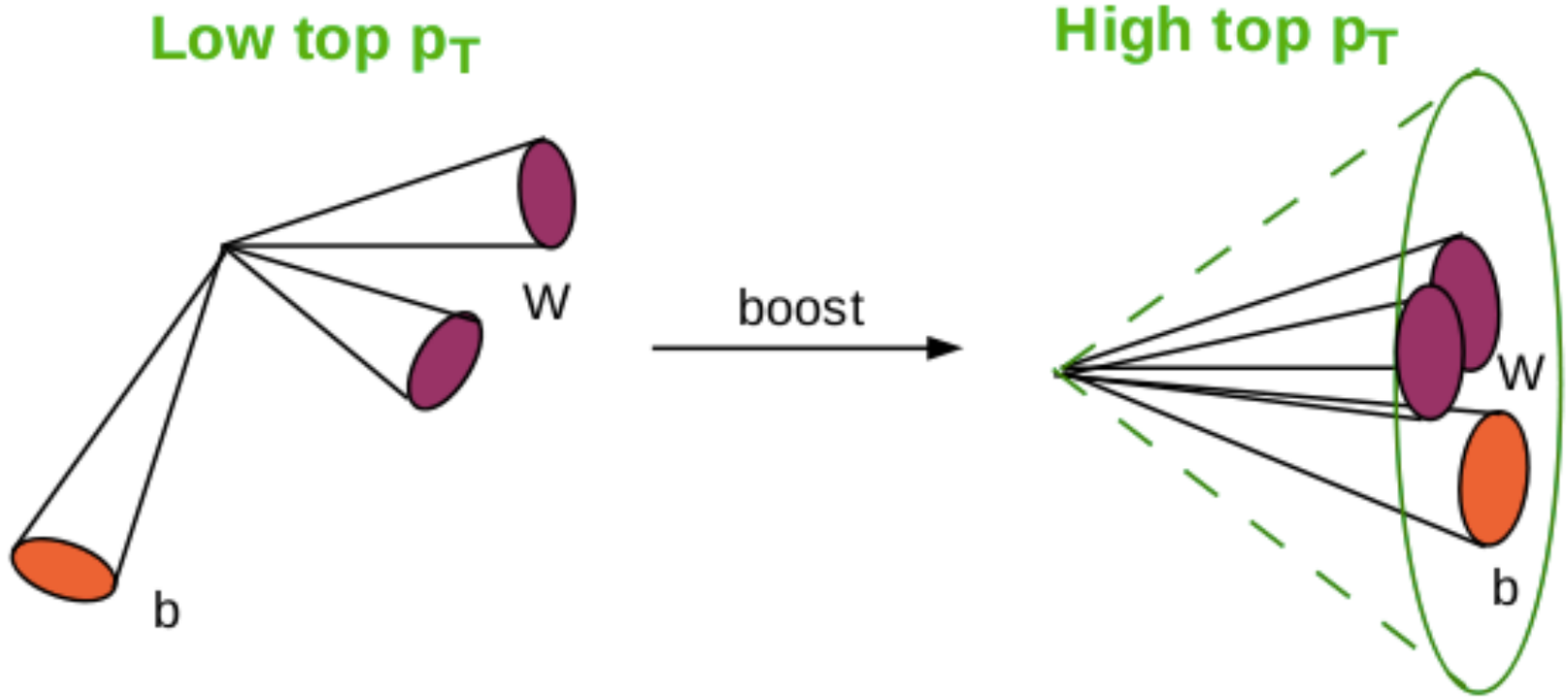
k_T	$d_{j_1 j_2} = \dfrac{\Delta R^2_{j_1 j_2}}{D^2} \min\left(p^2_{T,j_1}, p^2_{T,j_2}\right)$	$d_{j_1 B} = p^2_{T,j_1}$
Cambridge/Aachen	$d_{j_1 j_2} = \dfrac{\Delta R^2_{j_1 j_2}}{D^2}$	$y_{j_1 B} = 1$
anti-k_T	$d_{j_1 j_2} = \dfrac{\Delta R^2_{j_1 j_2}}{D^2} \min\left(\dfrac{1}{p^2_{T,j_1}}, \dfrac{1}{p^2_{T,j_2}}\right)$	$d_{j_1 B} = \dfrac{1}{p^2_{T,j_1}}$.

anti-k_T
Default ATLAS/CMS algorithm, nice and circular boundaries

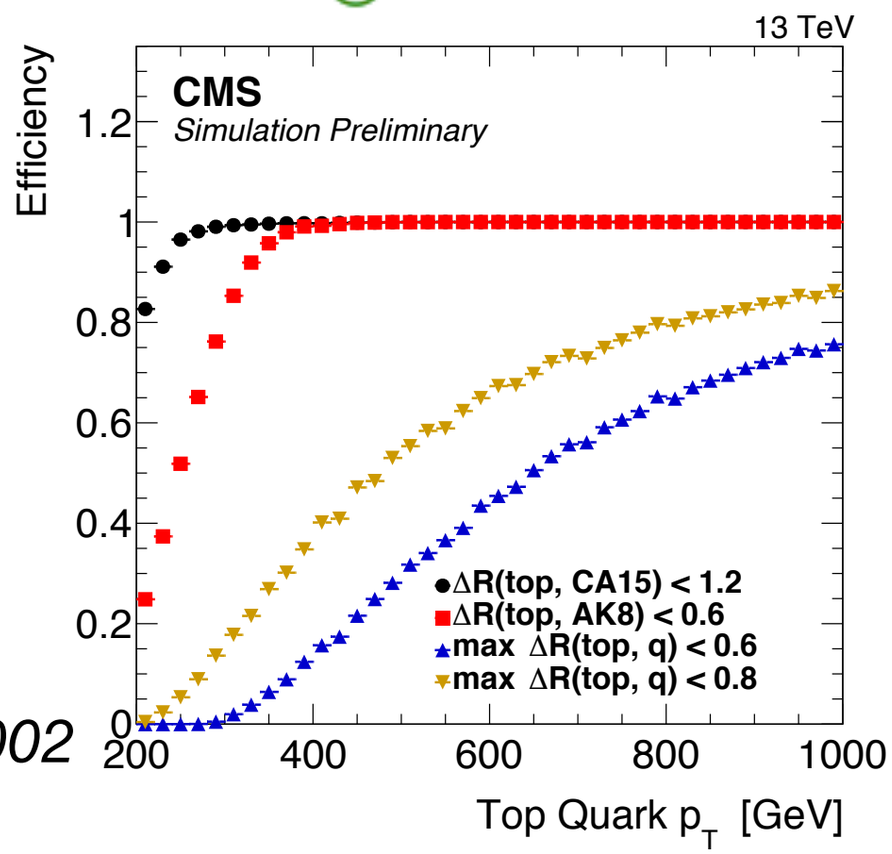
k_T and **C/A**
better interpretability in terms of QCD



The (very) basics



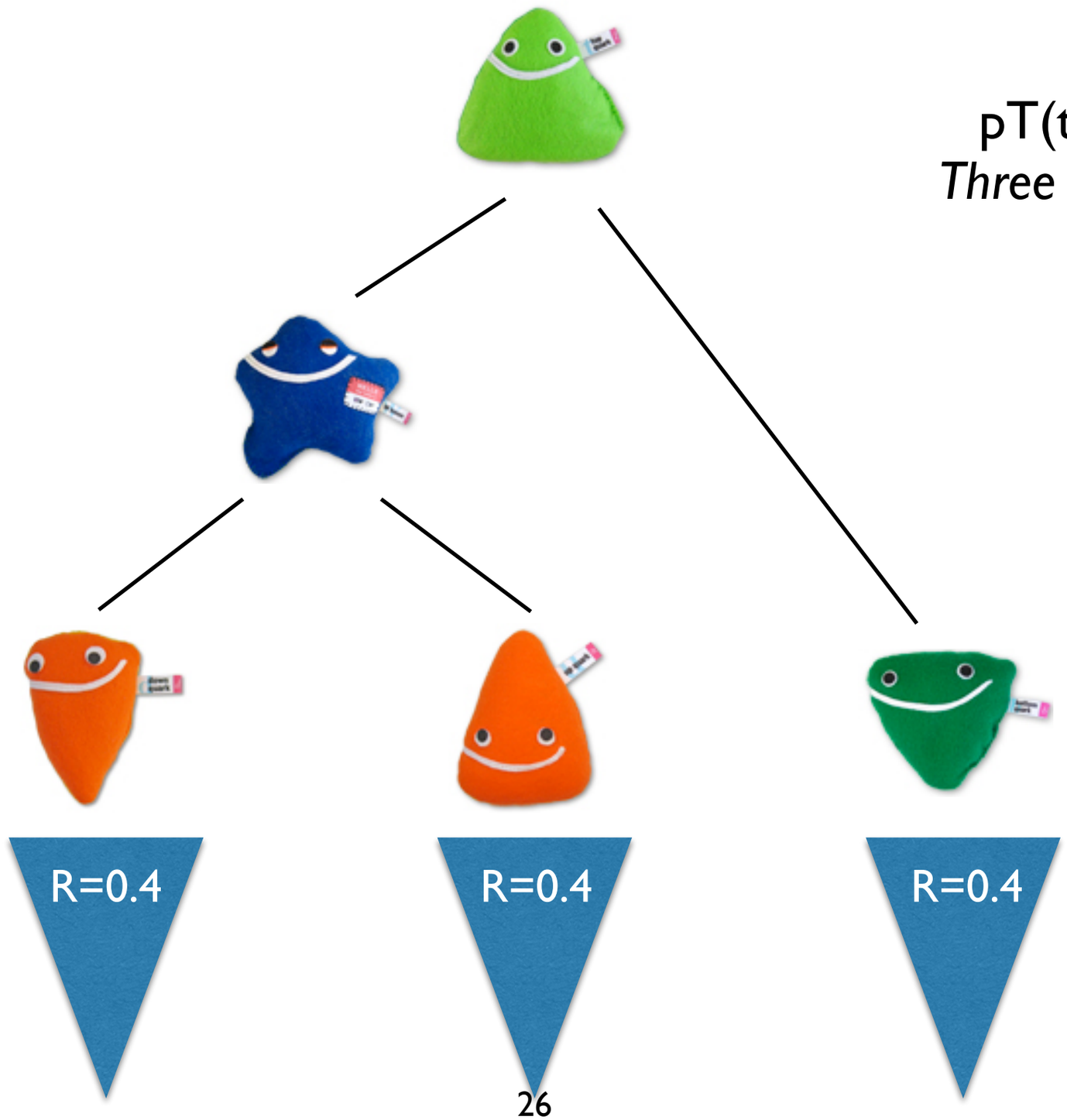
- Hadronically decaying top quarks
- Contained in one (large-R) jet
- How to distinguish
  from light quark/gluon jets?



CMS PAS JME-15-002

Jets Merging

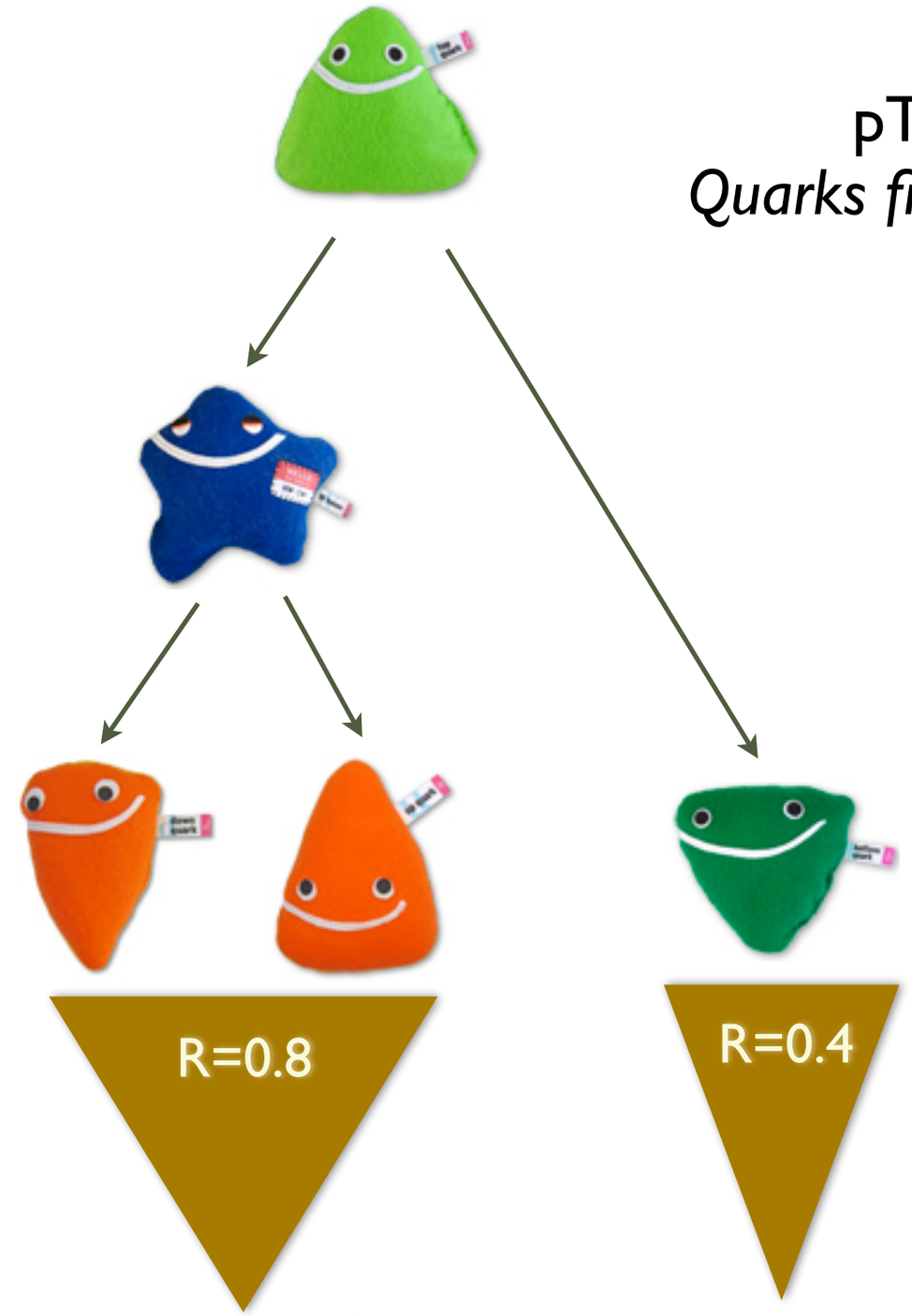
Example: Hadronic top quark decays



pT(top) ~ 150 GeV:
Three separate jets in the detector

Jets Merging

Example: Hadronic top quark decays

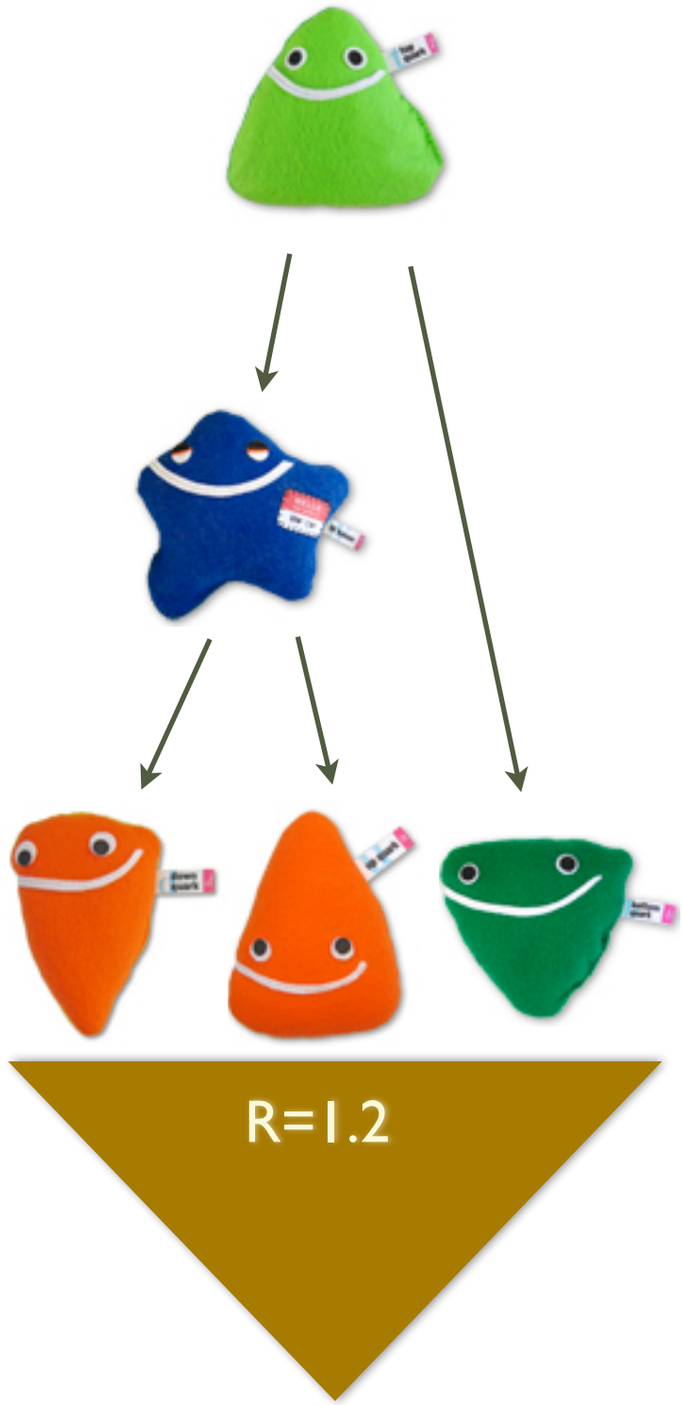


pT (top) ~ 250 GeV

Quarks from the W can merge into one jet

Jets Merging

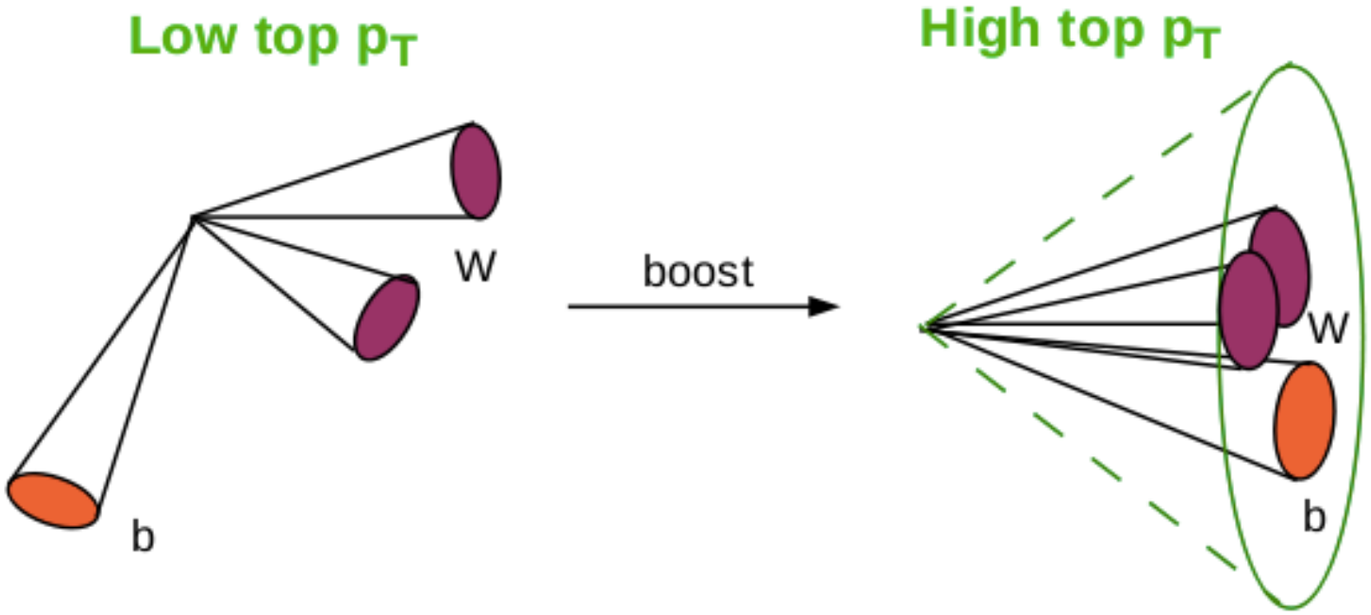
Example: Hadronic top quark decays



pT (top) ~ 500 GeV

One object in the detector

Heavy Resonance Tagging



- Hadronically decaying top/Higgs/W/Z
- Contained in one (large-R) jet
- How to distinguish from light quark/gluon jets (and from each other)
- For new physics searches (and SM studies)

Some Classical solutions:
(aka jet substructure)

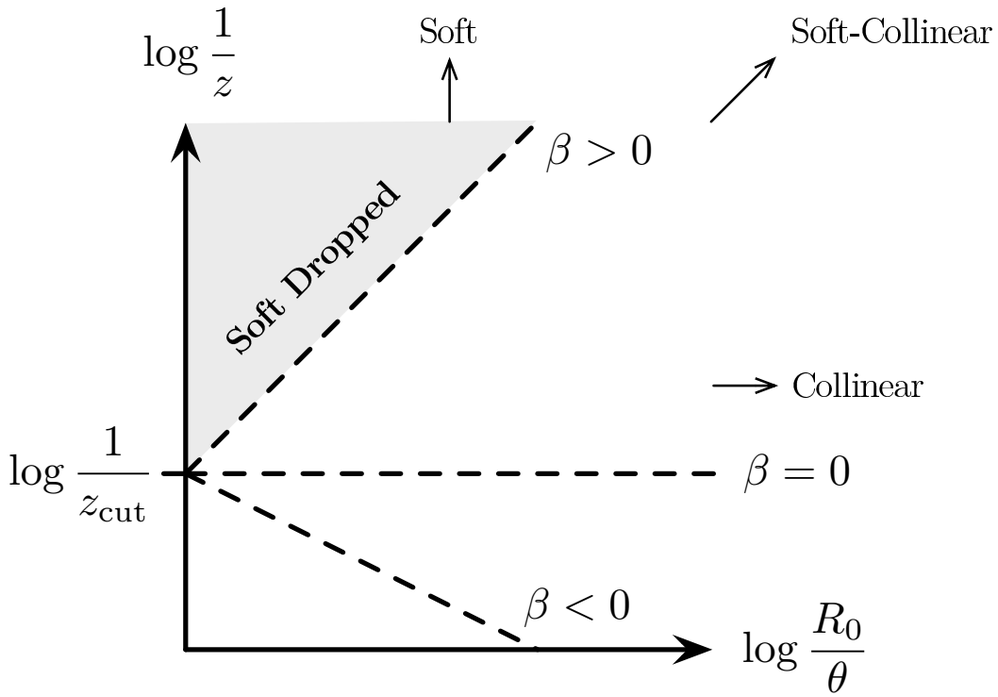
- Mass
  Calculate using a grooming algorithm (eg mMDT/softdrop or pruning)
- Centers of hard radiation
  n-subjettiness or energy correlation functions
- Flavour
  b tagging of large-R jets or subjets
- Soft substructure
  Color connection
- Inclusive reconstruction
  HEPTopTagger V2, HOTVR
- Other substructure variables
  Shower deconstruction, template tagger, …

Towards an Understanding of the Correlations in Jet Substructure
D Adams et al (BOOST 2013 Participants), Eur.Phys.J. C75
Top Tagging, T Plehn, M Spannowksy, J.Phys. G39 (2012) 083001
Boosted Top Tagging Method Overview, GK, Proc. Top2017

Jet Grooming

- Remove
  - soft radiation
  - underlying event
  - pile up
- from jet to access top mass

mMDT
Softdrop



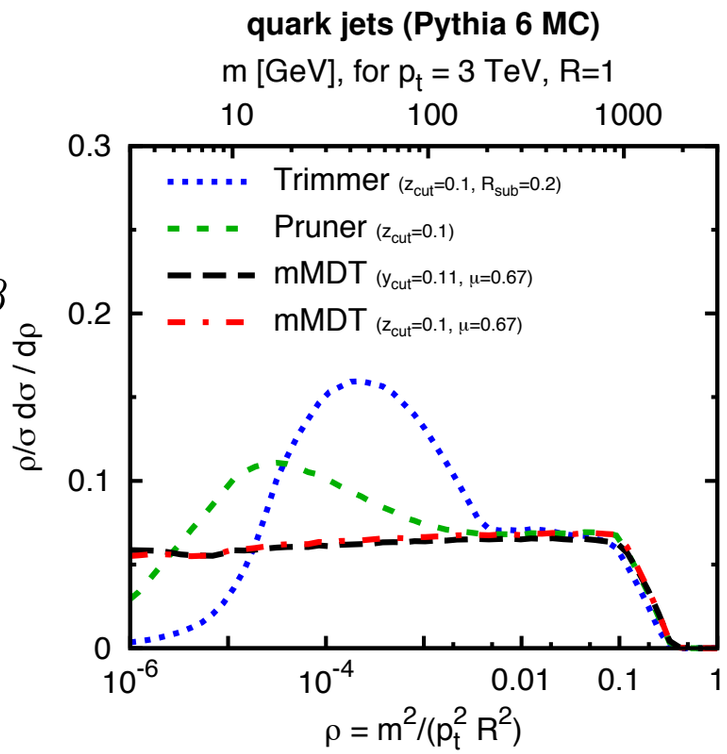
$$\frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{\rm cut} \left(\frac{\Delta R_{12}}{R_0} \right)^{\beta}$$

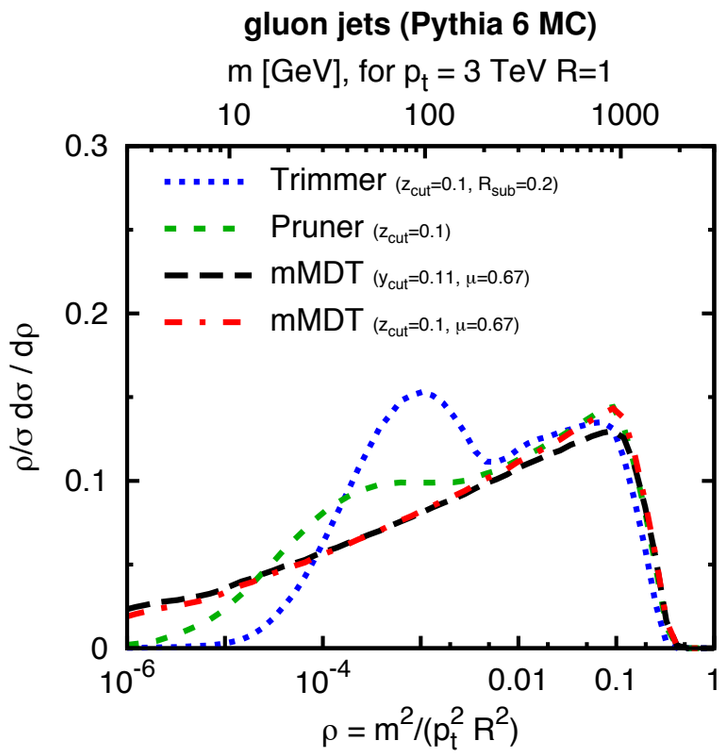
Towards an understanding of jet substructure
M Dasgupta, A Fregoso, S Marzani, G Salam
JHEP 1309 029

Soft Drop
A Larkoski, S Marzani, G Soyez, J Thaler
JHEP 1405 146

Factorization for groomed jet substructure beyond the next-to-leading logarithm
C Frye, AJ Larkoski, MD Schwartz, K Yan
JHEP 1607 064

- Find hard substructure using step-wise unclustering
- No pure soft divergences
- Analytically calculable to high precision





n-Subjettiness

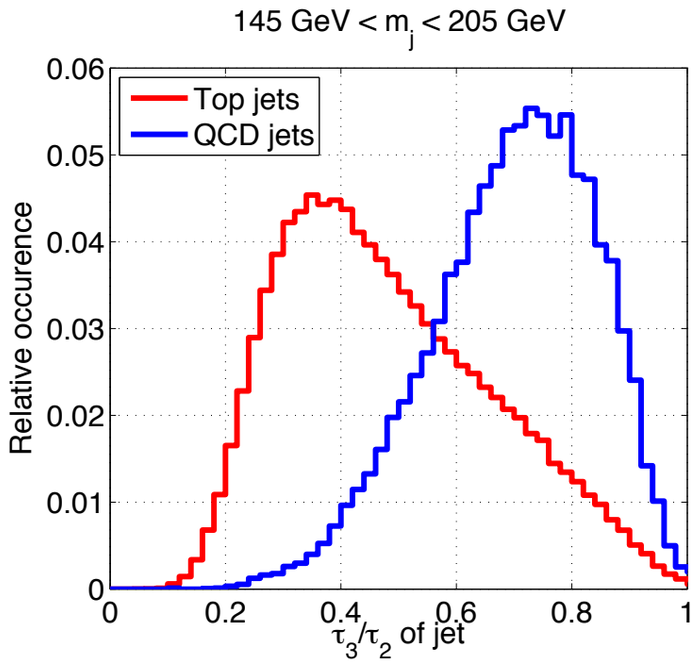
Identifying Boosted Objects with N-subjettiness
J Thaler, KV Tilburg,
JHEP 1103 015

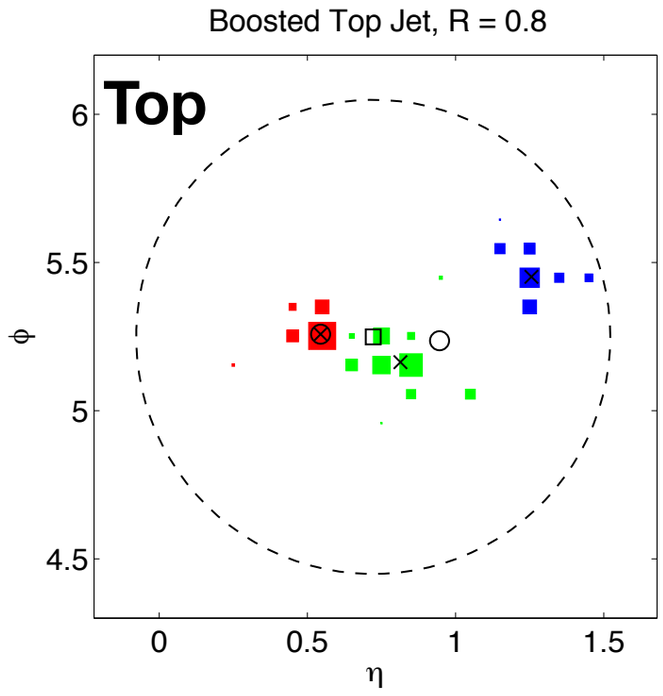
Dichroic subjettiness ratios to distinguish colour flows in boosted boson tagging
G Salam, L Schunk, G Soyez
JHEP 1703 022

$$\tau_N = \frac{1}{d_0} \sum_k p_{T,k} \min \{\Delta R_{1,k}, \Delta R_{2,k}, \cdots, \Delta R_{N,k}\}$$

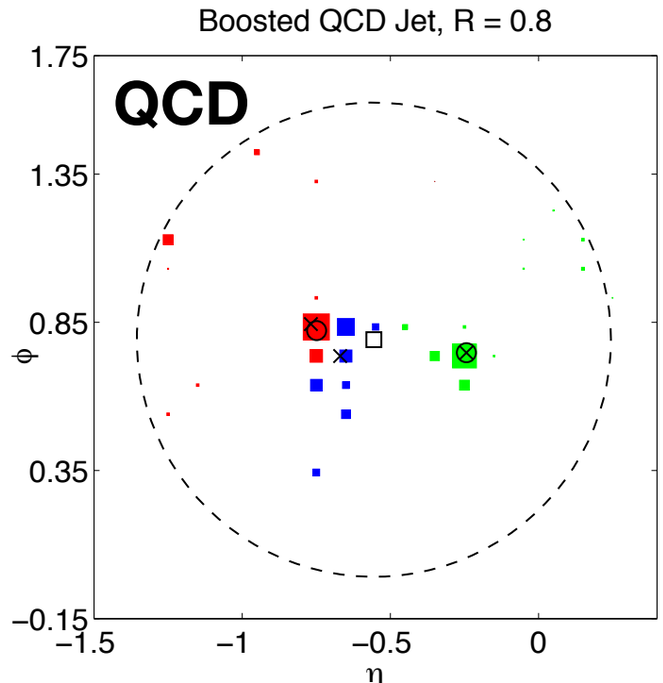
- n-subjettiness: Small when compatible with n-prong substructure

- Used for top-tagging: $\frac{\tau_3}{\tau_2}$





(b)

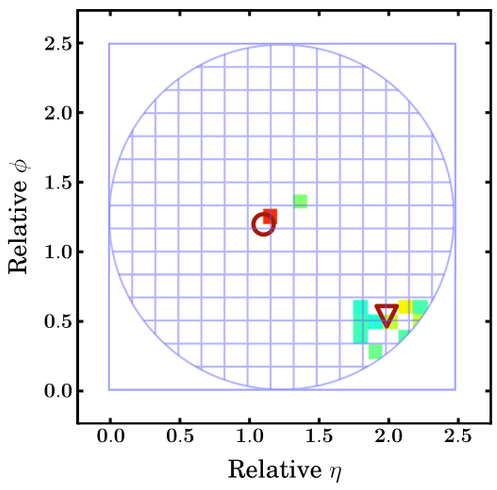


- Recent ideas:

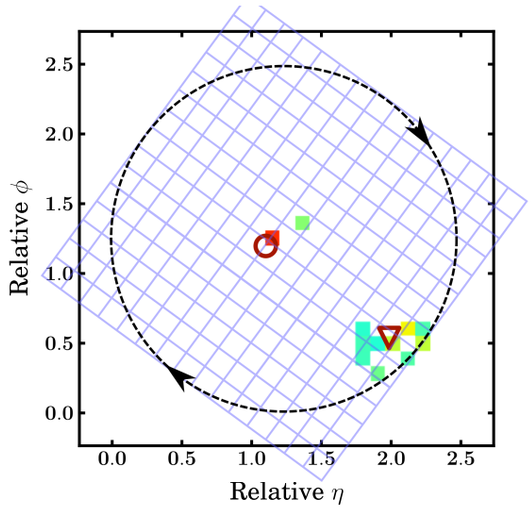
  - Dichroic n-subjettiness = ratio of n-subjettiness with different grooming (JHEP 1703 022)

  - Use for jet clustering (XCone: JHEP 1511 07)

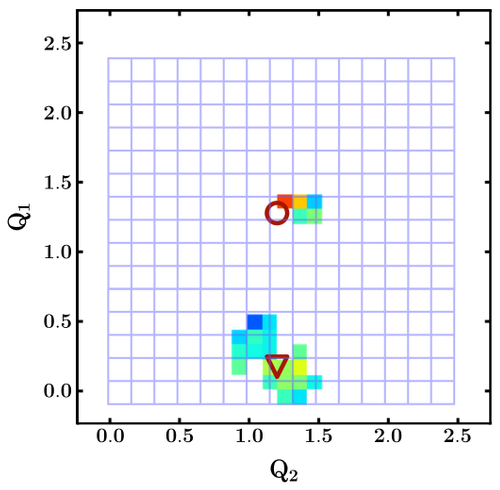
Jet Images



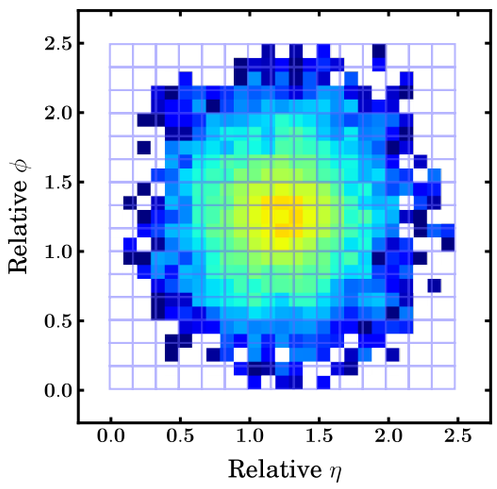
(a) Jet-image prior to rotation



(b) Rotated pixel grid



(c) Jet-image after projection onto rotated grid, before translation

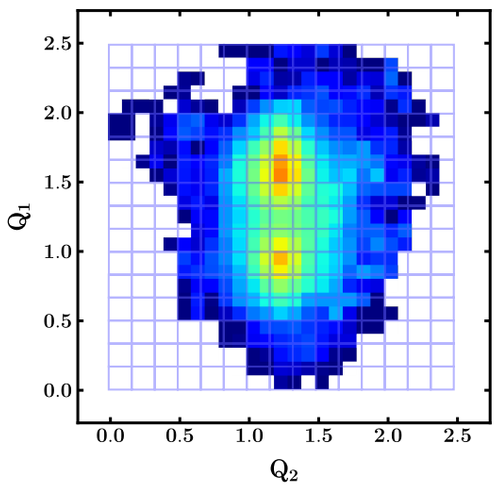


(d) Average jet-image, prior to rotation

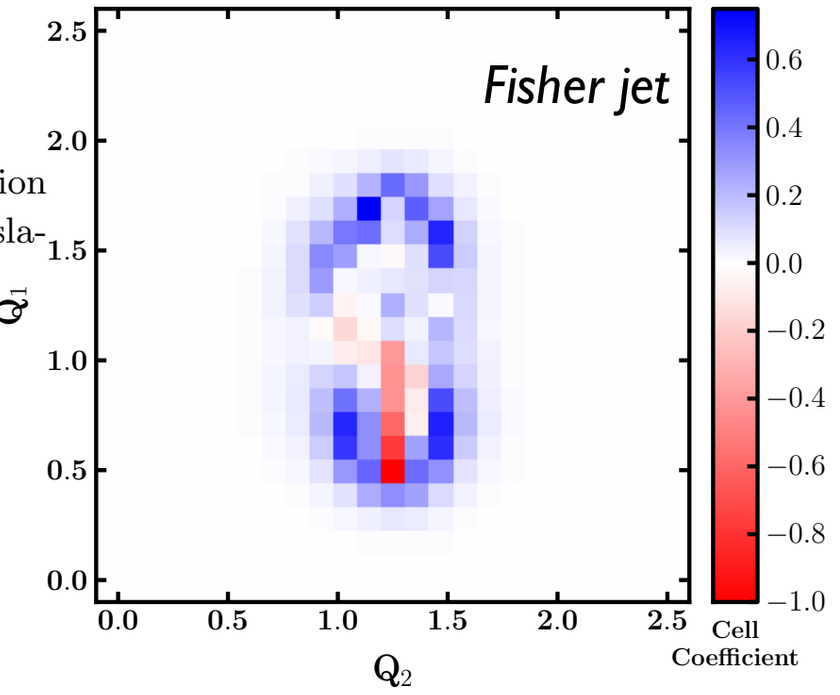
$$D[A] = \sum_{k=1}^{N^2} \bar{F}_k \cdot \bar{A}_k$$

discriminant *jet image*

Fisher jet



(e) Average jet-image, after pre-processing



- Apply ideas from computer vision for W tagging
- Use Fisher linear discriminant as classifier

Jet-Images: Computer Vision Inspired Techniques for Jet Tagging
J Cogan, M Kagan, E Strauss, A Schwartzman
arXiv:1407.5675

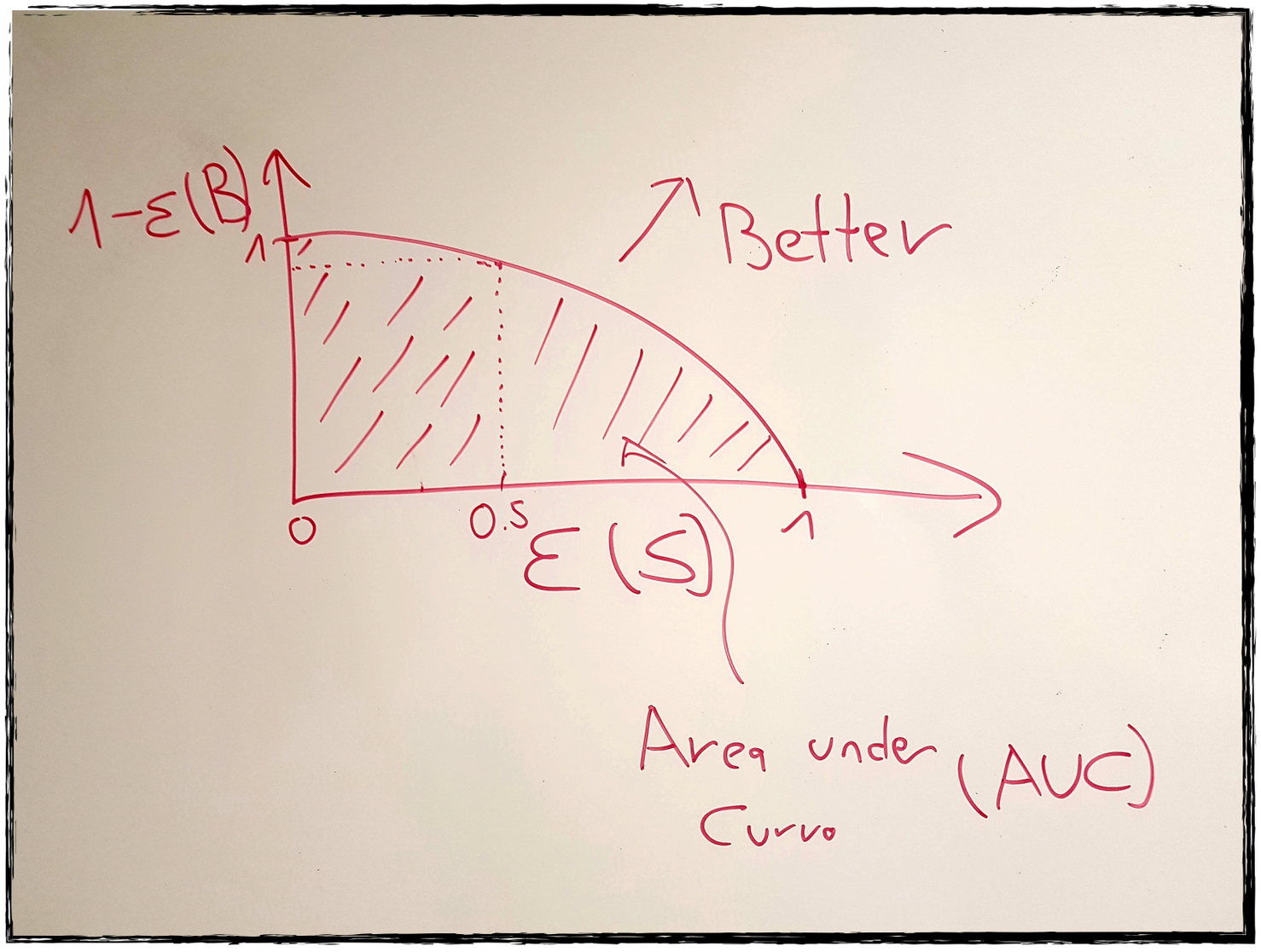
Architectures

Architecture

How to build a network?

- **Fully connected** networks
- **Image** recognition
- **Natural language** processing
- **Physics** based
- **Graph** networks

Performance Measures



ROC:
Receiver operation characteristic

AUC:
Area under curve

Background rejection at given signal efficincy

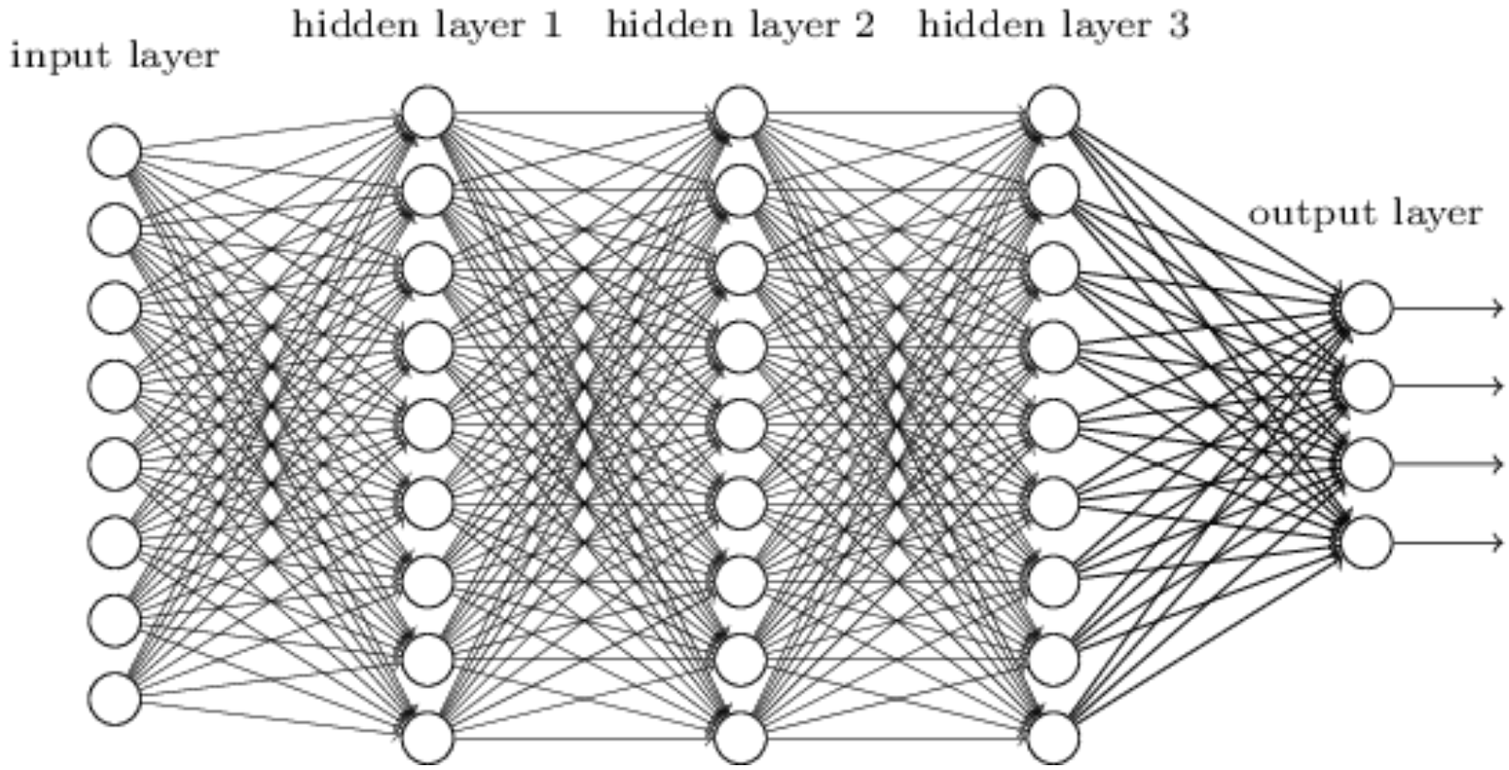
Accuracy

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

(fraction of correct predictions)

scikit-learn.org

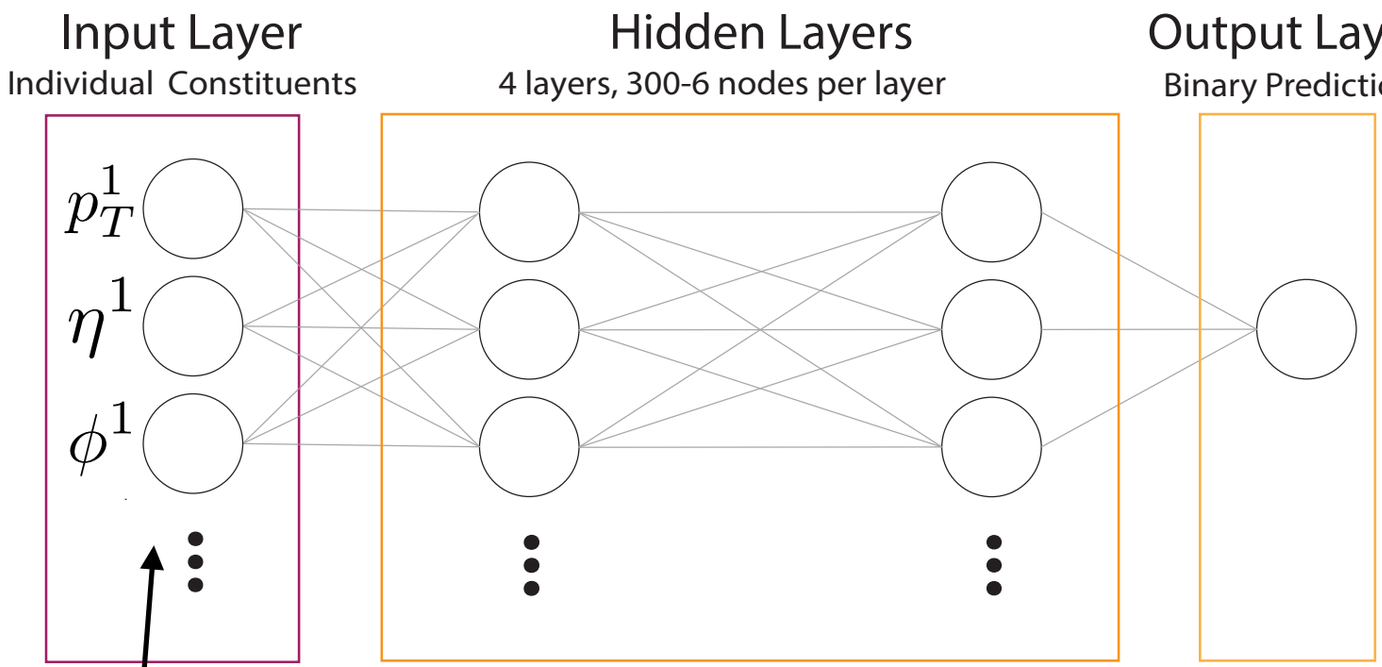
Fully Connected



- *Classical* artificial neural network
- Most generic structure
- Many weights, inefficient
- Can we use the symmetry of the problem to simplify matters?

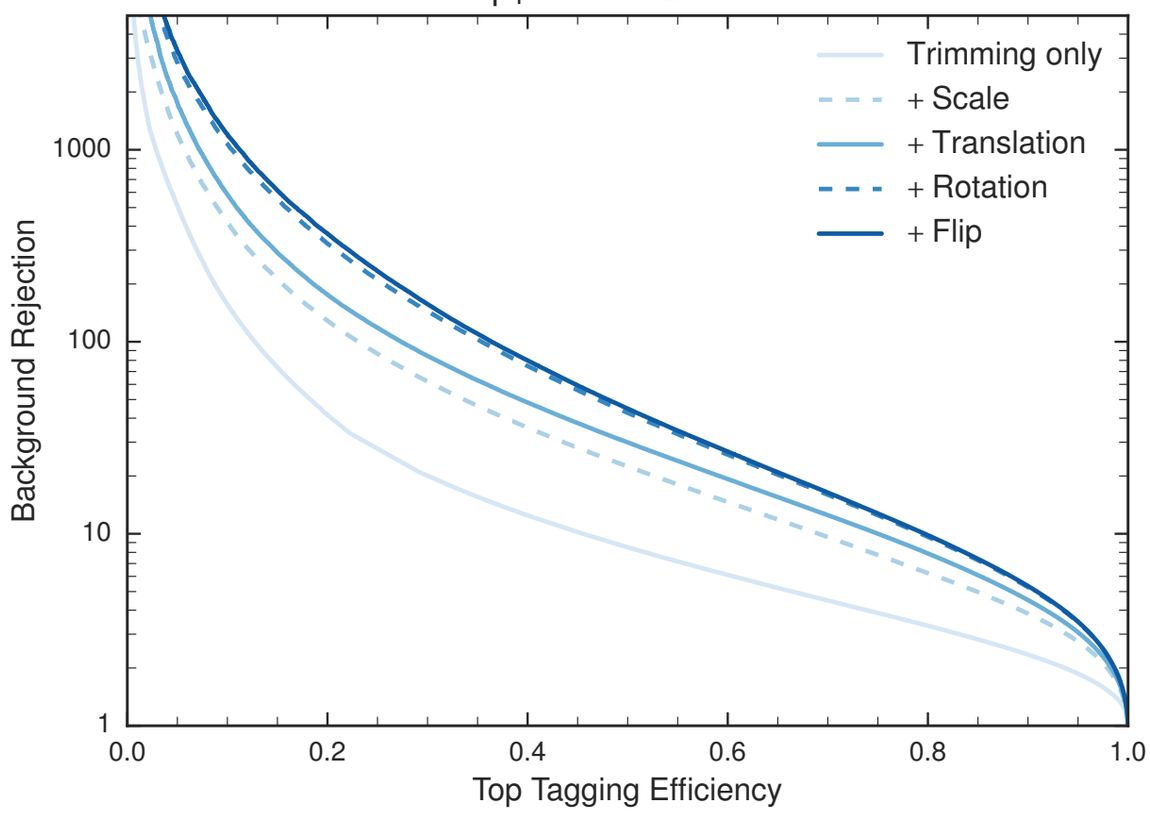
Jet Constituents for Deep Neural Network Based Top Quark Tagging
J Pearkes, W Fedorko, A Lister, C Gay
1704.02124
Playing Tag with ANN: Boosted Top Identification with Pattern Recognition
LG Almeida, M Backovic, M Cliche, SJ Lee, M Perelstein
1501.05968

Fully Connected Networks

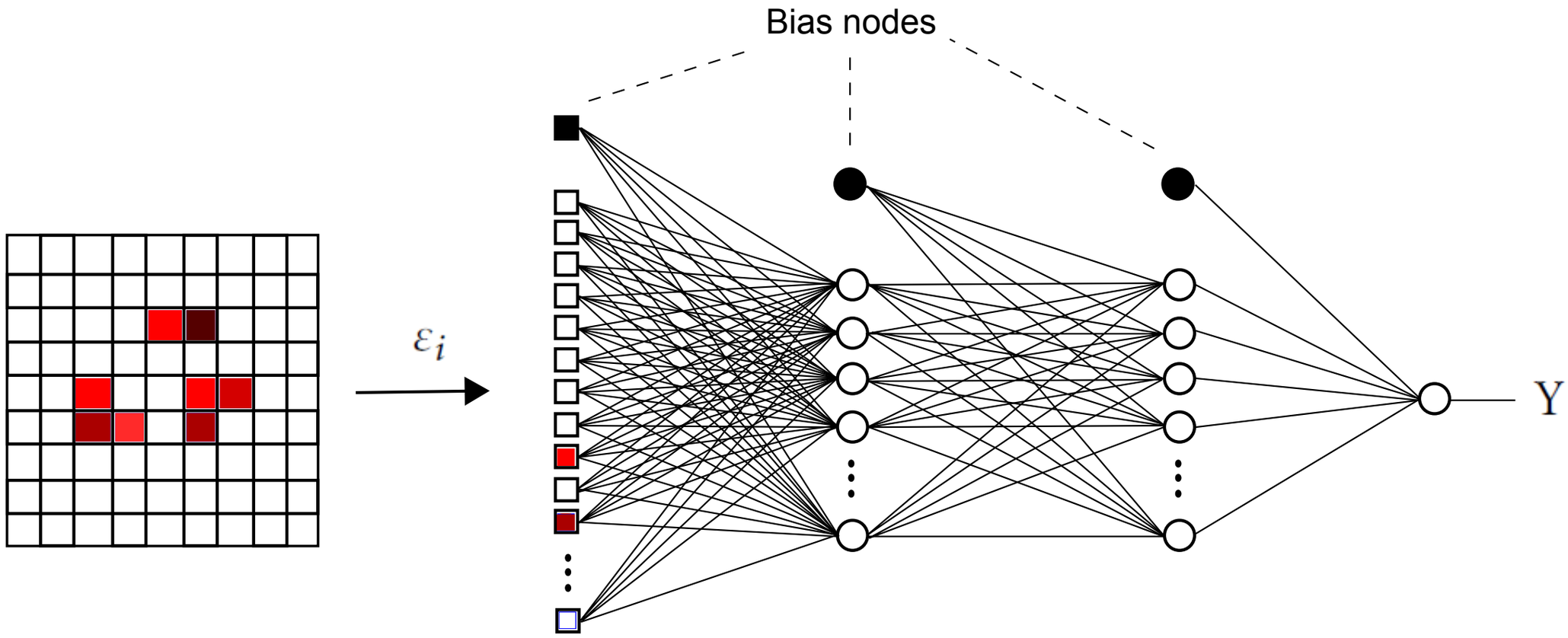


Directly input jet constituents

Preprocessing essential!

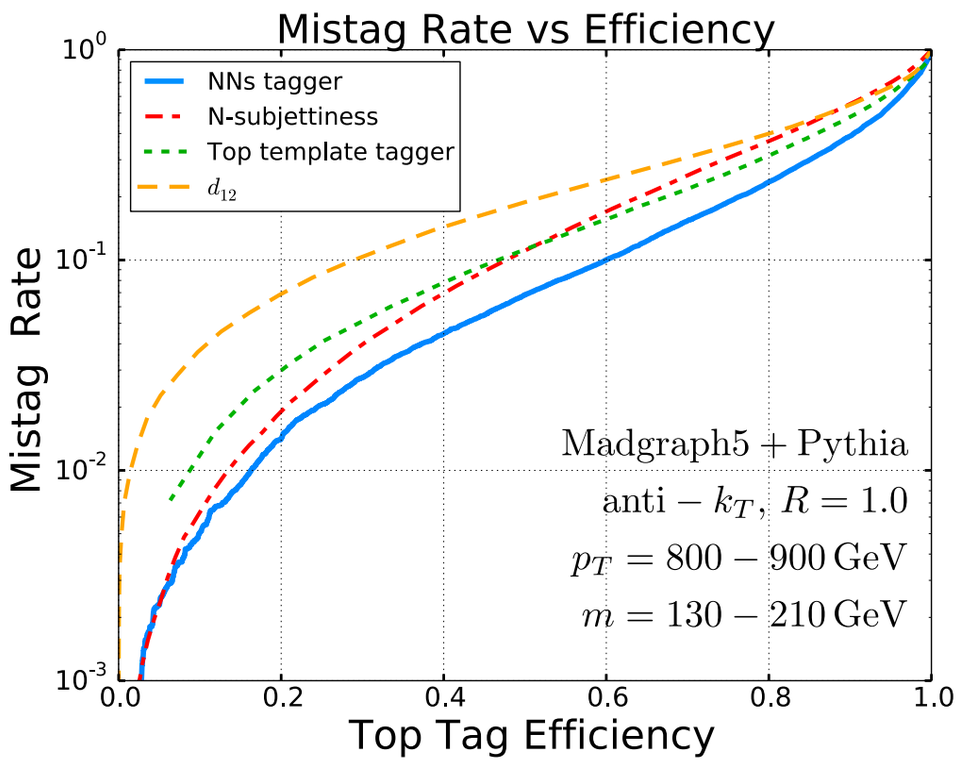


Jet Constituents for Deep Neural Network Based Top Quark Tagging
J Pearkes, W Fedorko, A Lister, C Gay
1704.02124



Fully connected + images

Playing Tag with ANN: Boosted Top Identification with Pattern Recognition
LG Almeida, M Backovic, M Cliche, SJ Lee, M Perelstein
arXiv:1501.05968



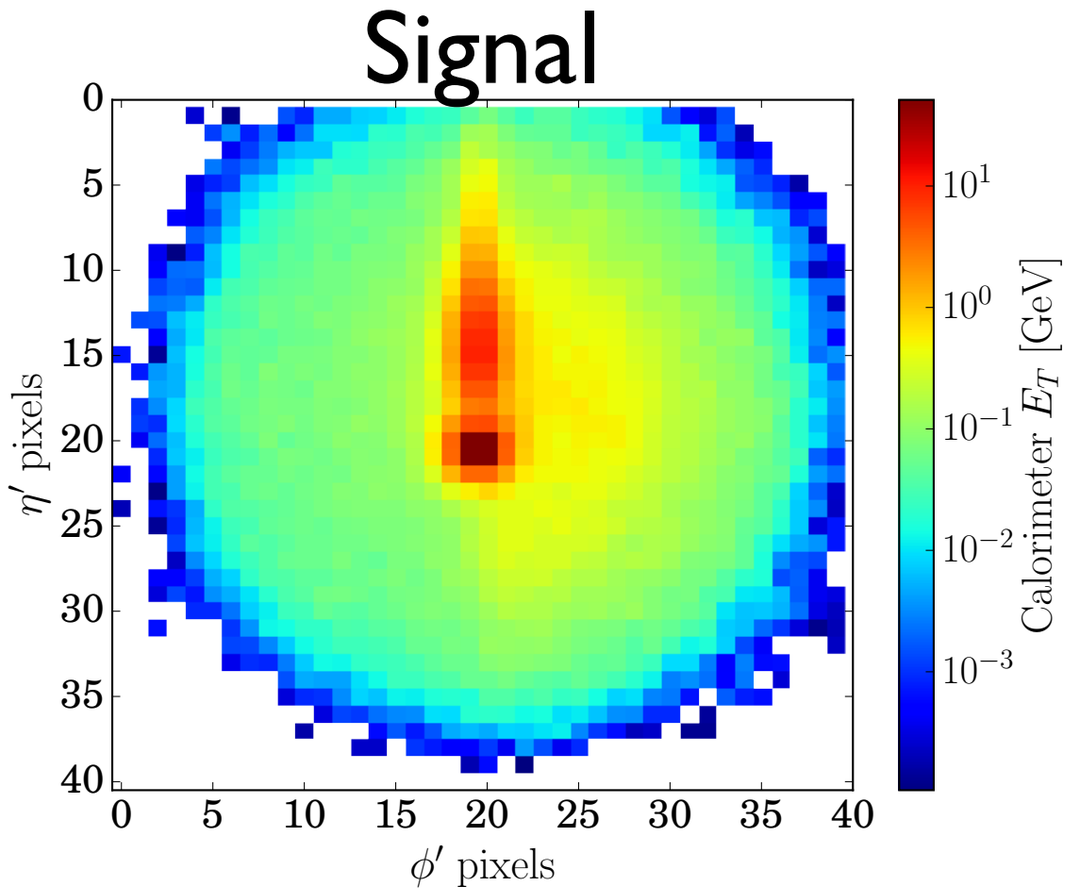
Convolutional

Image approach

- Jets = 2d grayscale images:
  - 1 pixel = 0.1 in eta, 5 degree in phi
  - pixel energy: calorimeter ET
- Preprocessing (for illustration!)
  - 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

Deep-learning Top Taggers or The End of QCD?
GK, Tilman Plehn, Michael Russell, Torben Schell
JHEP 05 (2017) 006

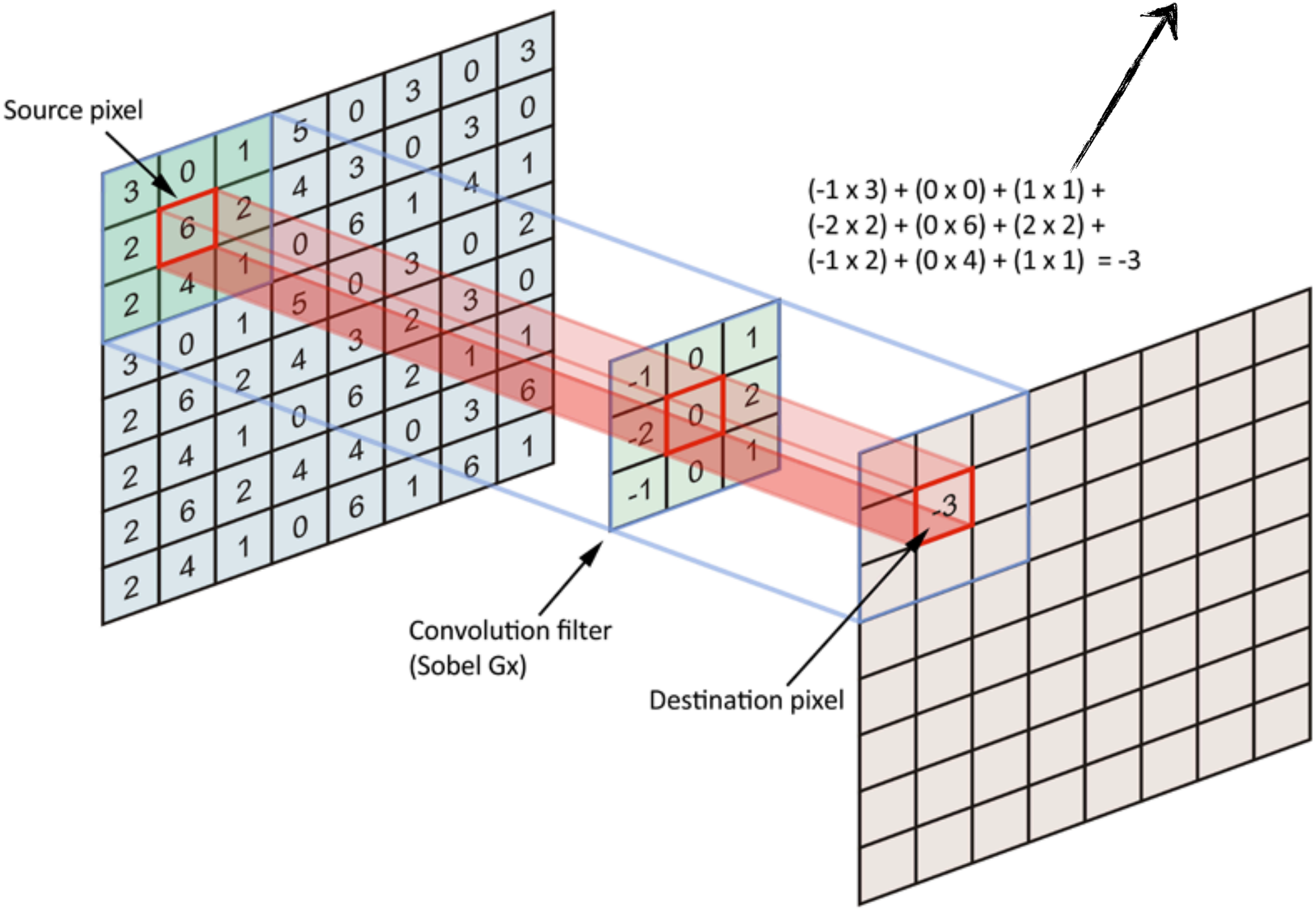
Origins:
Jet-Images: Computer Vision Inspired Techniques for Jet Tagging
J Cogan, M Kagan, E Strauss, A Schwartzman
arXiv:1407.5675
Jet-Images – Deep Learning Edition
Ld Oliveira, M Kagan, L Mackey, B Nachman, A Schwartzman
JHEP 1607 069



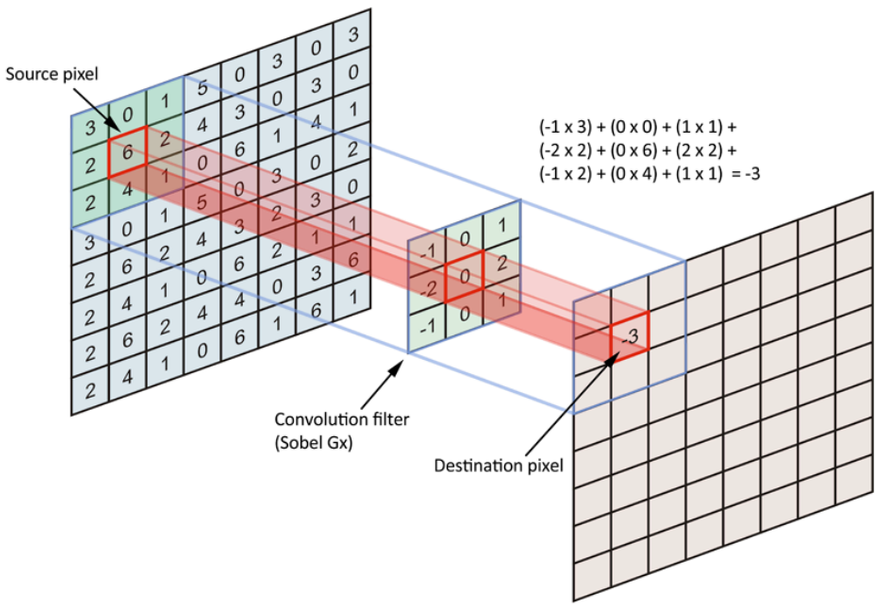
Overlay of 10k images

Convolutional Layer

That's the weights we want to train

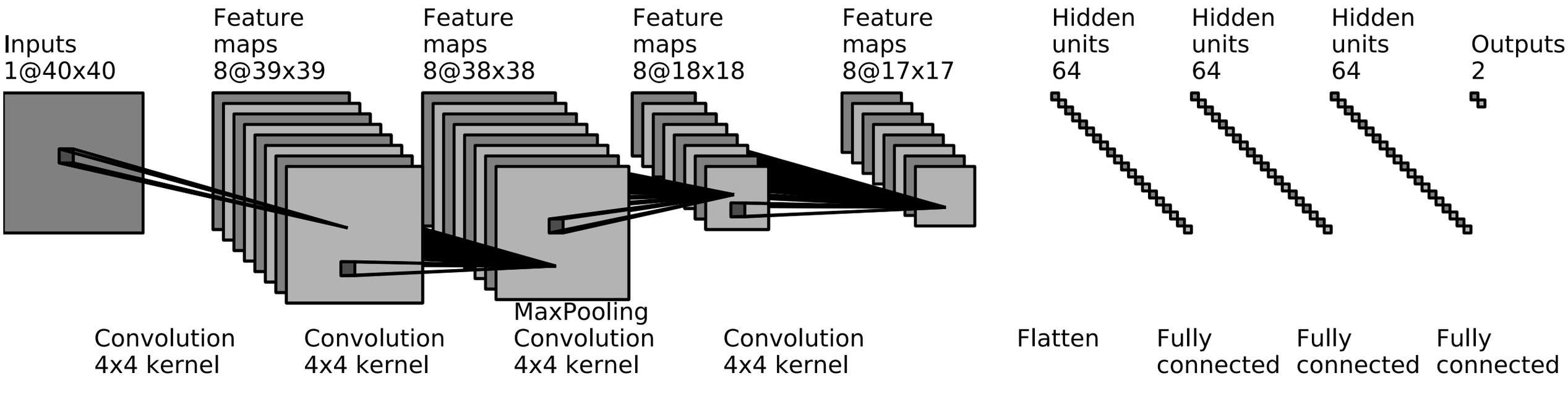


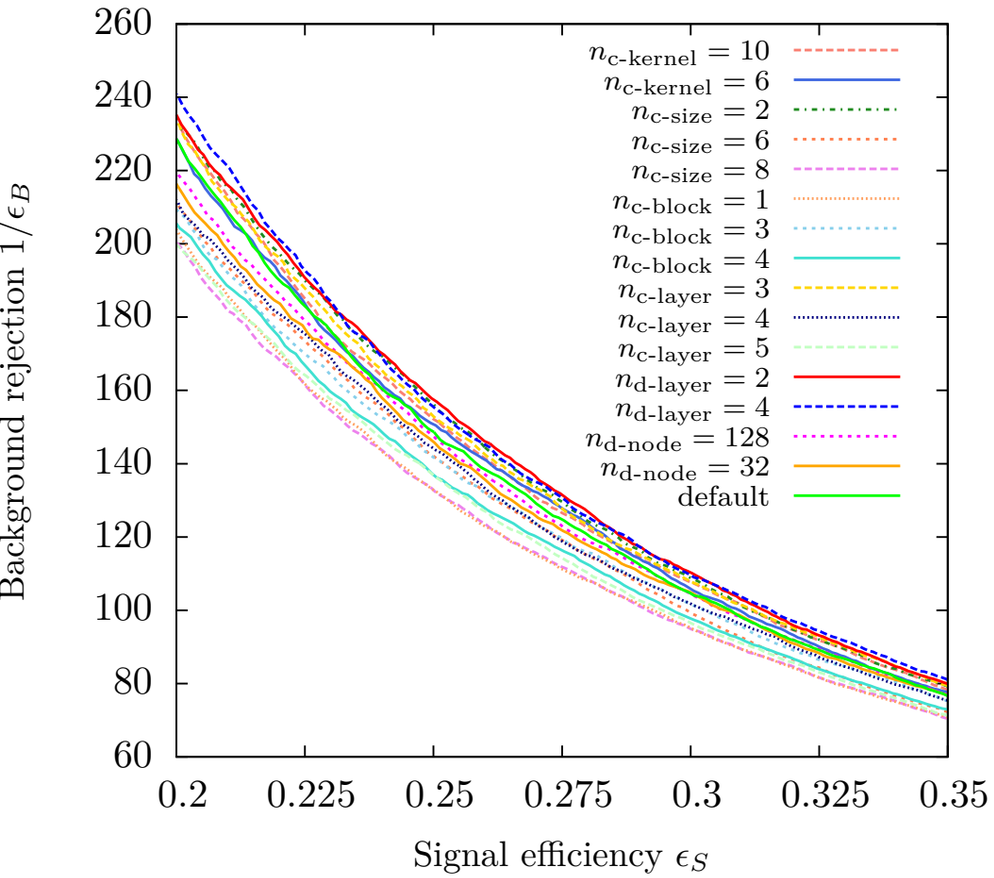
Convolutional Network



- ***How to build a convolutional network***
  - Chain multiple conv layers
  - Use multiple masks per layer
  - Pooling
    - Max Pooling
    - Average Pooling
  - Add a fully connected network in the end

Network architecture





Iterative tuning of hyper parameters.

Performance

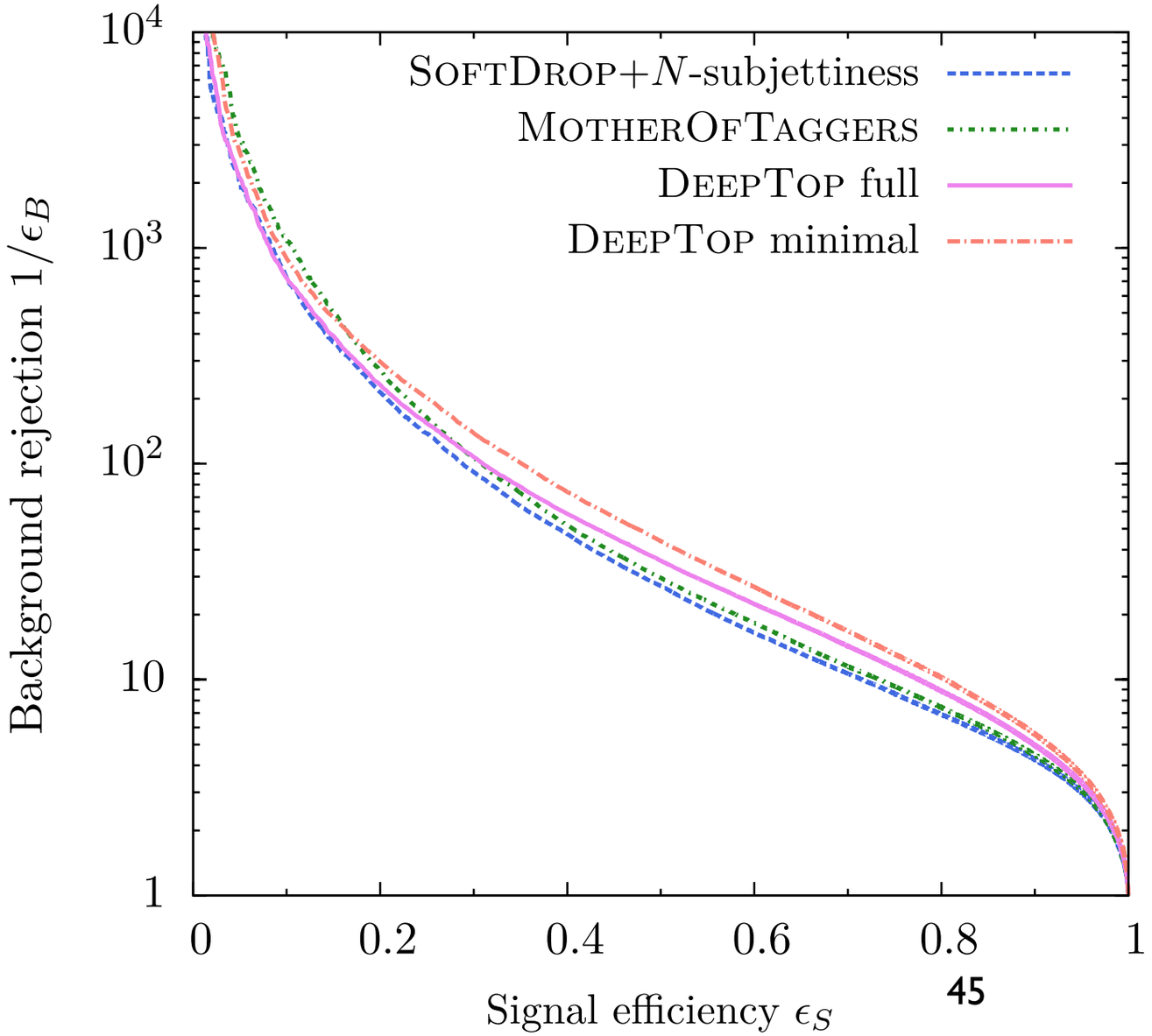
- Train a BDT on a set of standard tagging variables

SoftDrop + n-subjettiness:

$$\{ m_{\mathrm{sd}}, m_{\mathrm{fat}}, \tau_2, \tau_3, \tau_2^{\mathrm{sd}}, \tau_3^{\mathrm{sd}} \}$$

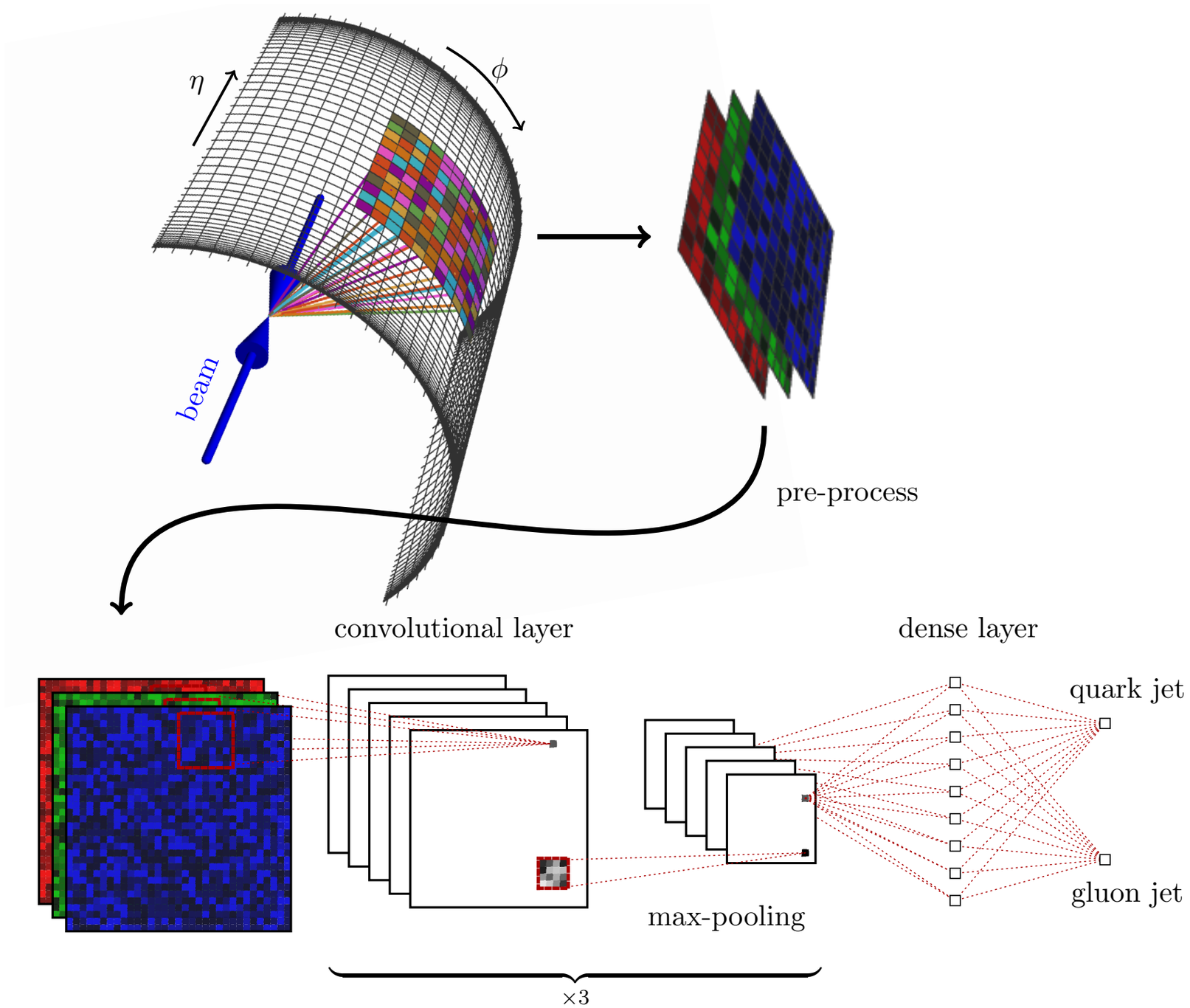
MotherOfTaggers:

$$\{ m_{\mathrm{sd}}, m_{\mathrm{fat}}, m_{\mathrm{rec}}, f_{\mathrm{rec}}, \Delta R_{\mathrm{opt}}, \tau_2, \tau_3, \tau_2^{\mathrm{sd}}, \tau_3^{\mathrm{sd}} \}$$



- Advantages:
  - Symmetry / structure
  - Straightforward
- Potential Problems
  - Resolution
  - Sparsity
  - How to encode complex information

Adding Color



`red` = transverse momenta of charged particles

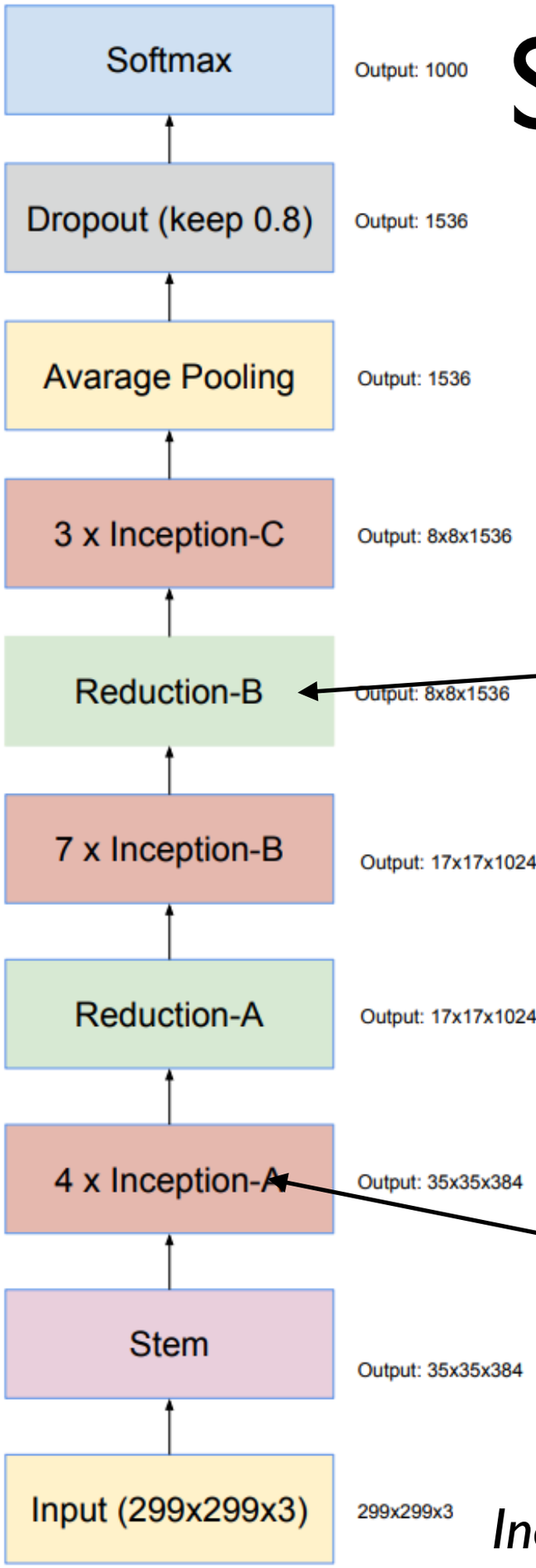
`green` = the transverse momenta of neutral particles

`blue` = charged particle multiplicity

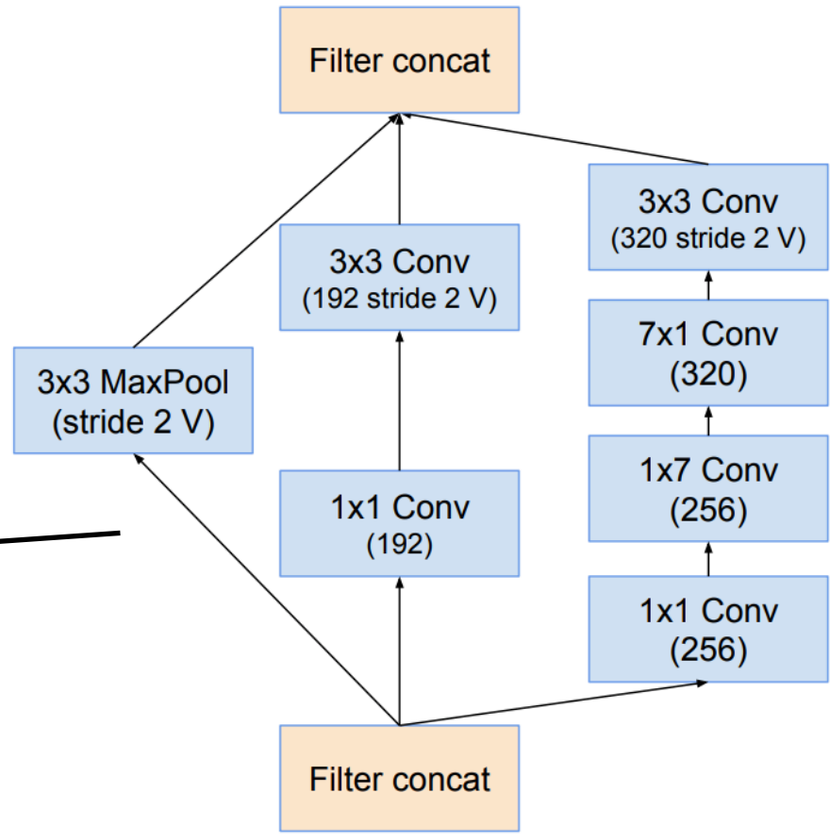
Deep learning in color: towards automated quark/gluon jet discrimination
PT Komiske, EM Metodiev, MD Schwartz
JHEP 01 (2017) 110

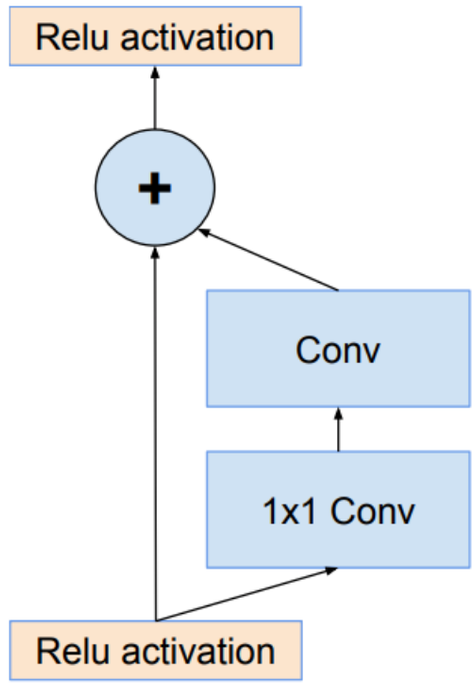
State of the art

Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning
C Szegedy, S Ioffe, V Vanhoucke, Alemi
arXiv:1602.07261

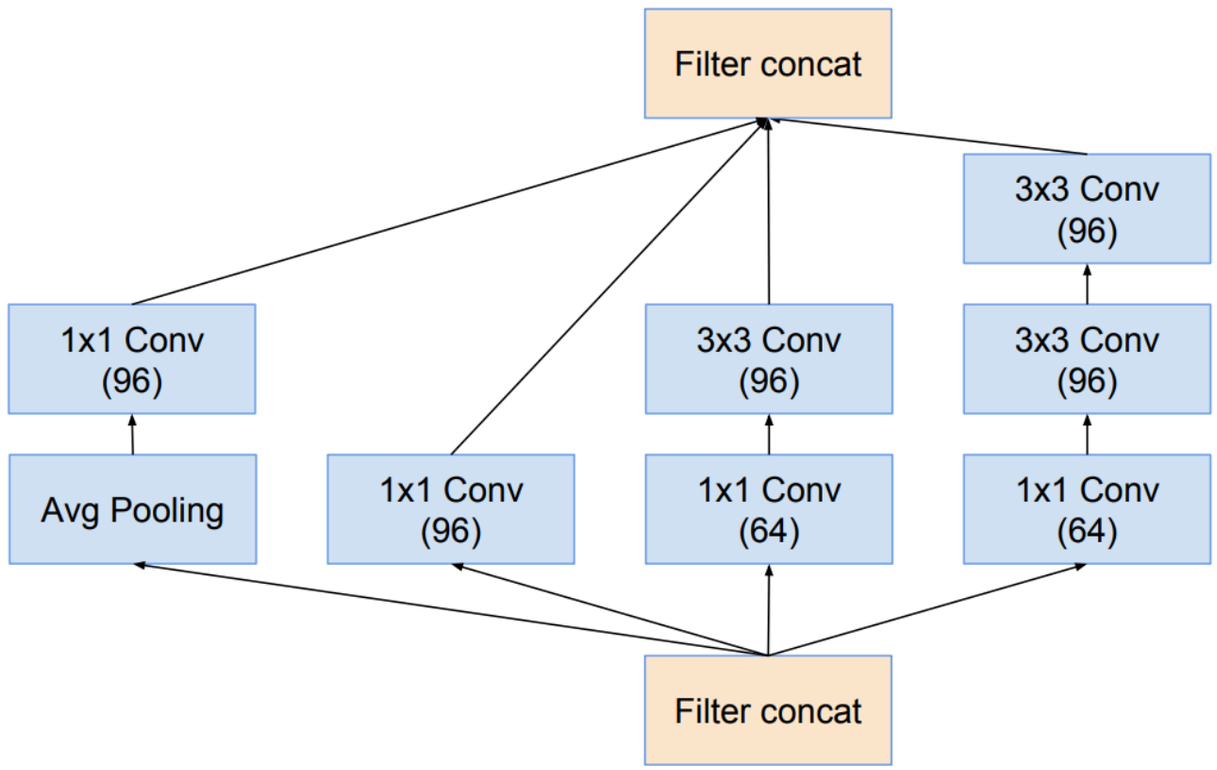


Inception v4





Residual connection



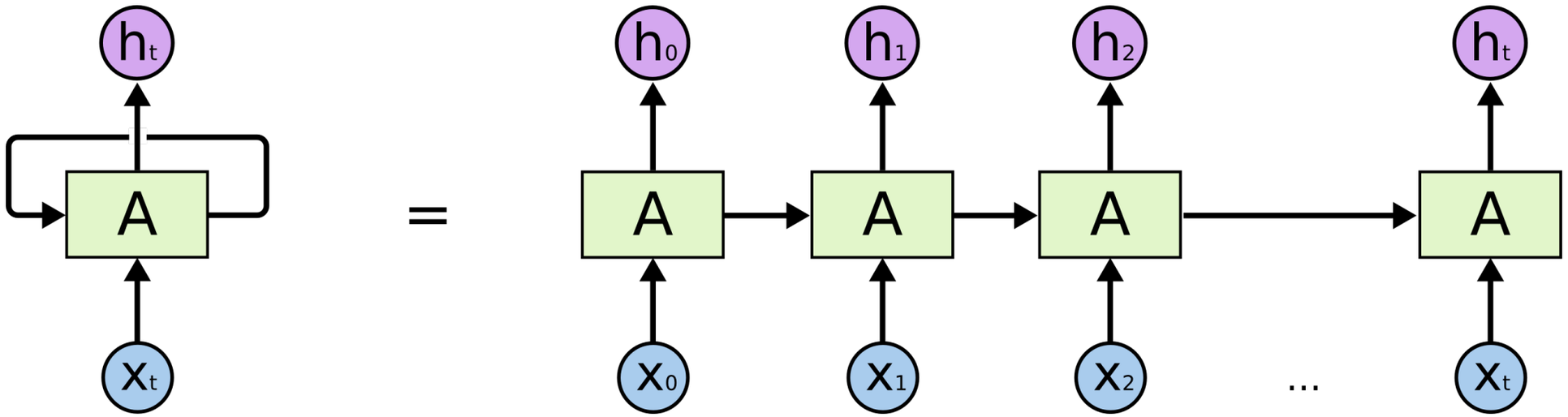
State of the art cont'd

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

https://keras.io/applications

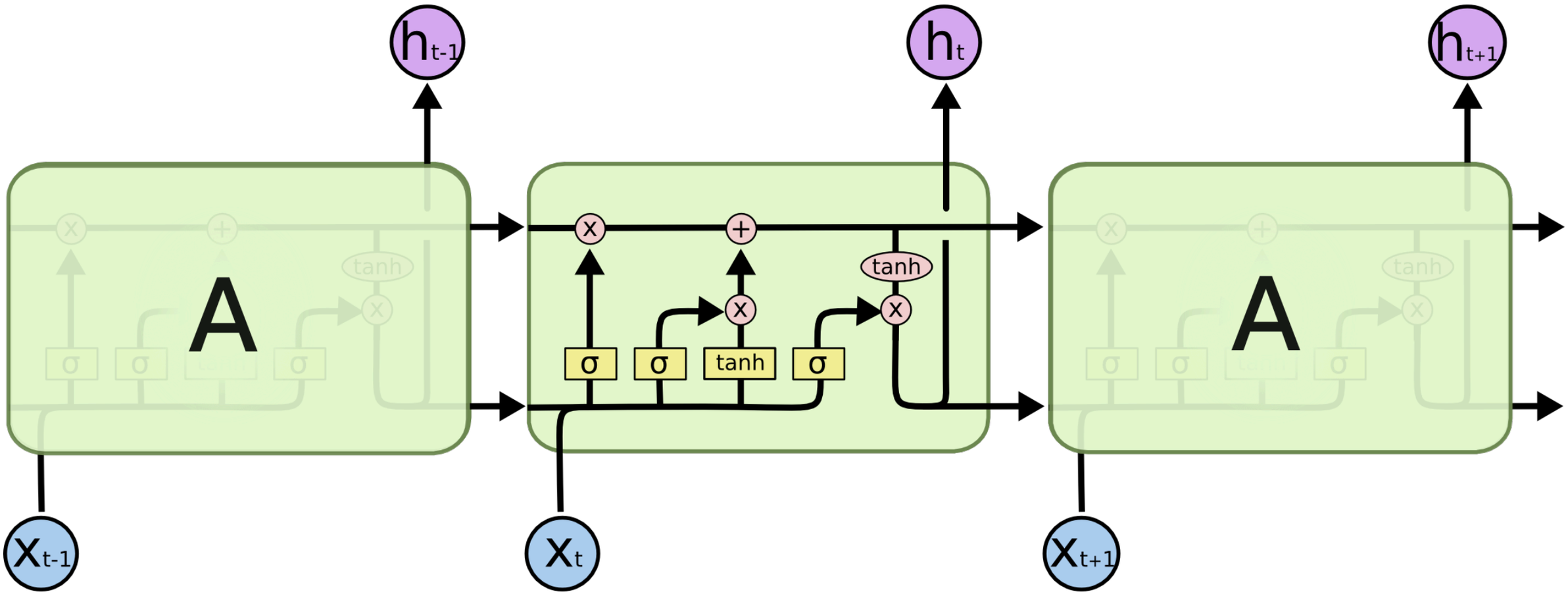
Recurrent

Recurrent



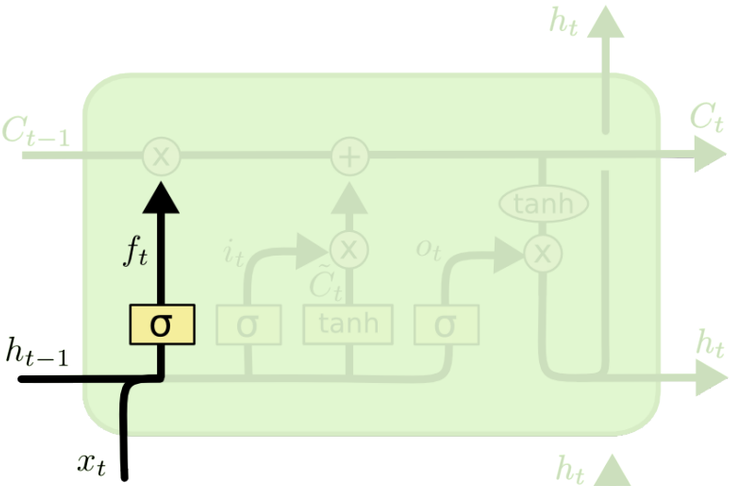
- Inspired by natural language processing
- Work with a sequence of inputs
- Inputs can change the state of the cell (*Long Short Term Memory*)

LSTM



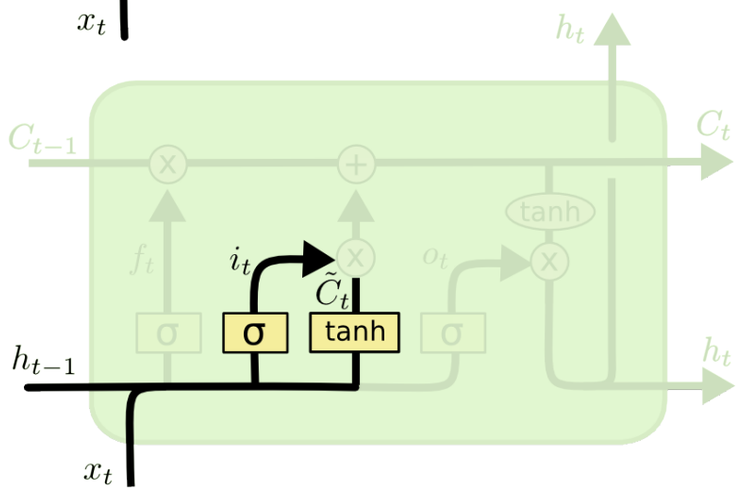
http://colah.github.io/

LSTM



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

forget

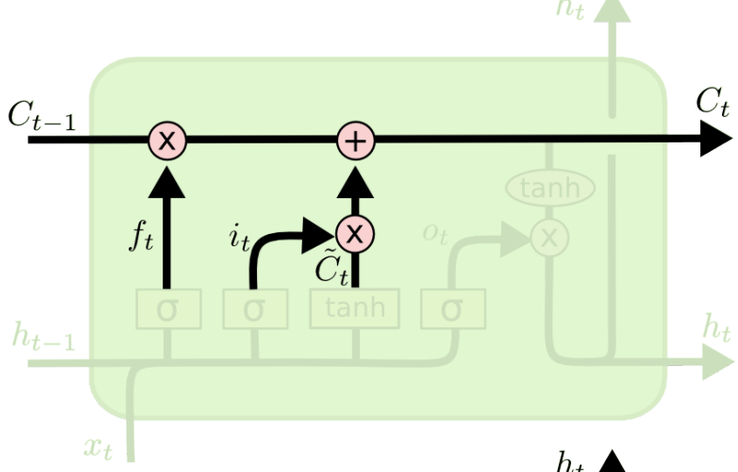


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

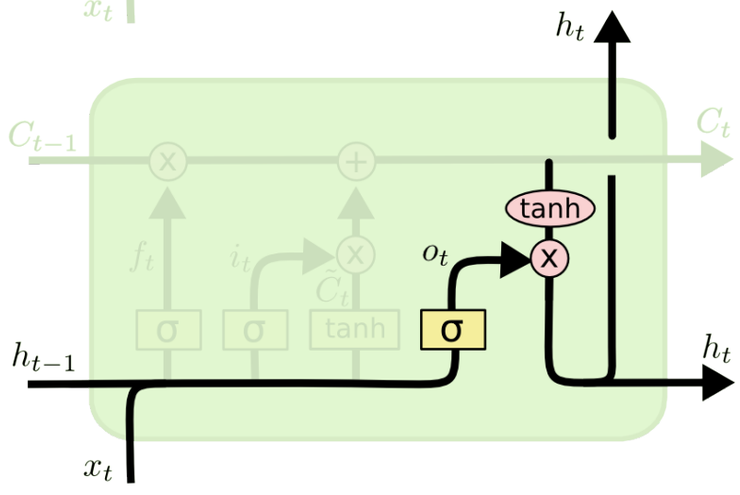
which inputs to keep?

Long Short Term Memory



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

update cell state



$$o_t = \sigma\left(W_o \; [\, h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

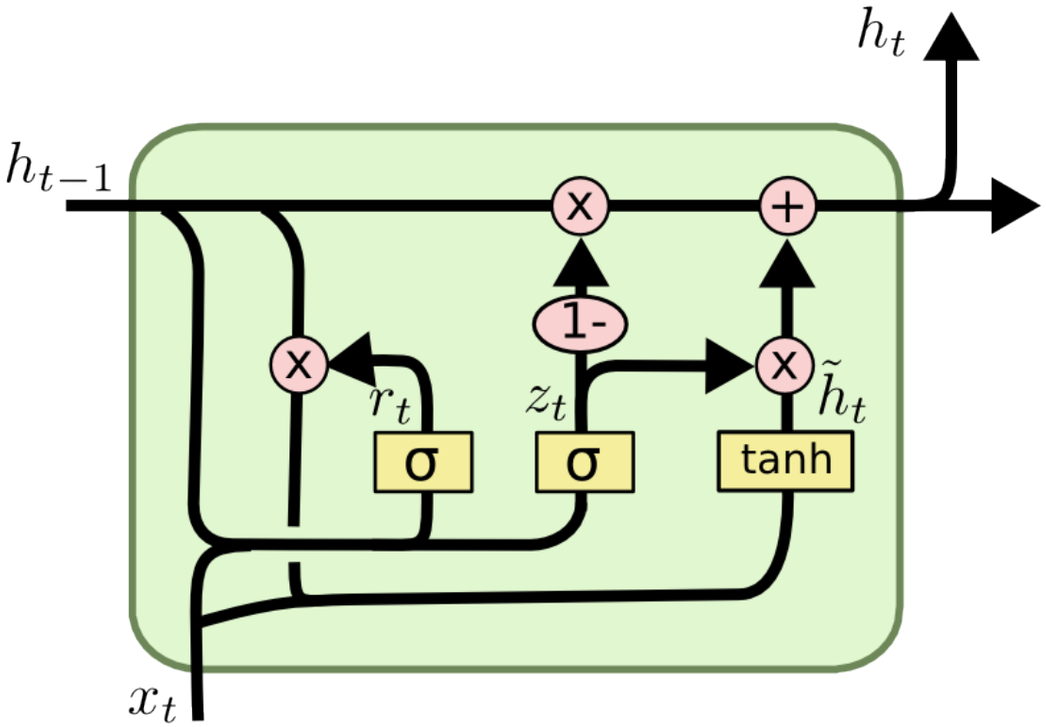
decide output

http://colah.github.io/

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
K Cho et al
arXiv: 1406.1078

GRU

Gated Recurrent Unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Combine forget and input gate
- Combine cell state and hidden state

Reminder: Jet Clustering

1112.4441
0005012
0802.1189

- Two approaches:
  - **Cone based** - Find stable axes of particles flow (ie. SisCone: 0704.0292)
  - **Sequential** - Pairwise combination of clusters according to a distance measure:

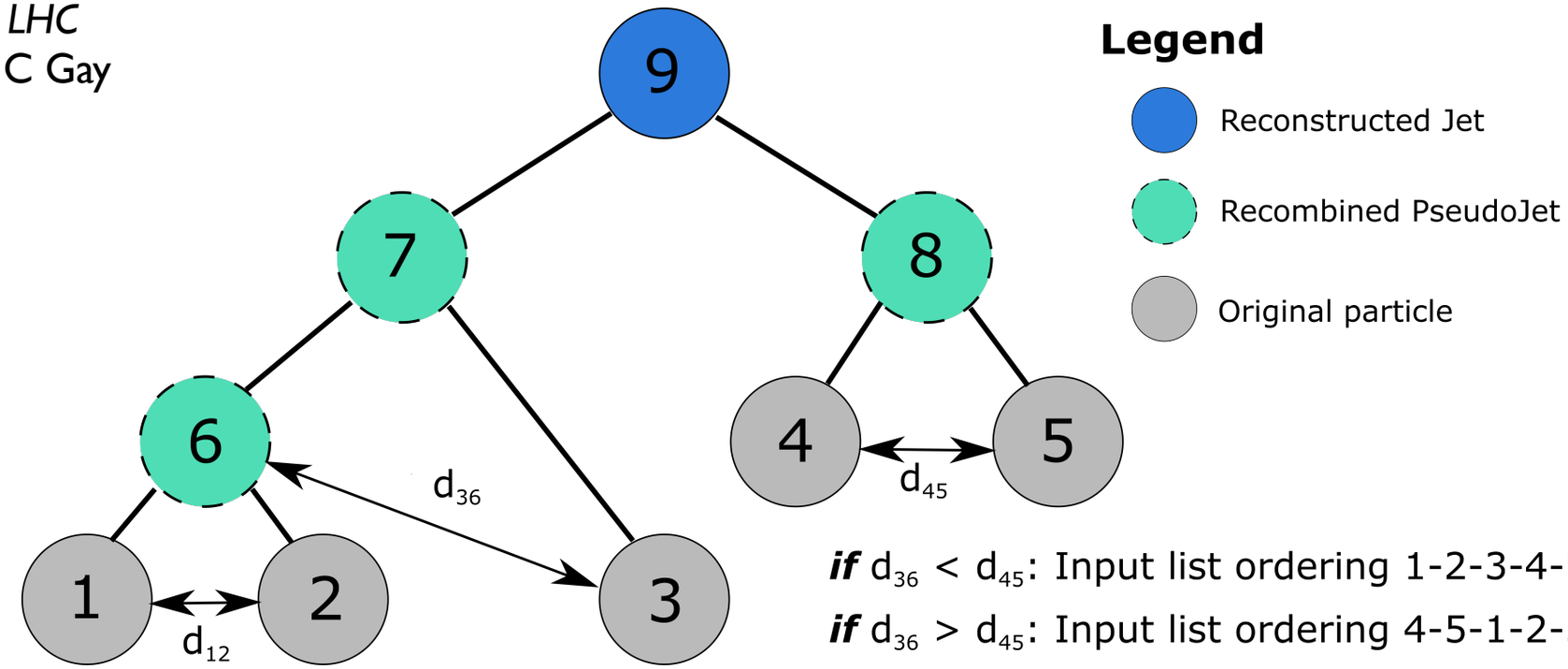
k_T	$d_{j_1 j_2} = \frac{\Delta R^2_{j_1 j_2}}{D^2} \ \min\left(p^2_{T,j_1}, p^2_{T,j_2}\right)$	$d_{j_1 B} = p^2_{T,j_1}$
Cambridge/Aachen	$d_{j_1 j_2} = \frac{\Delta R^2_{j_1 j_2}}{D^2}$	$y_{j_1 B} = 1$
anti-k_T	$d_{j_1 j_2} = \frac{\Delta R^2_{j_1 j_2}}{D^2} \min\left(\frac{1}{p^2_{T,j_1}}, \frac{1}{p^2_{T,j_2}}\right)$	$d_{j_1 B} = \frac{1}{p^2_{T,j_1}}$.

anti-k_T
Default ATLAS/CMS algorithm, nice and circular boundaries
k_T and **C/A**
better interpretability in terms of QCD

Physics Applications

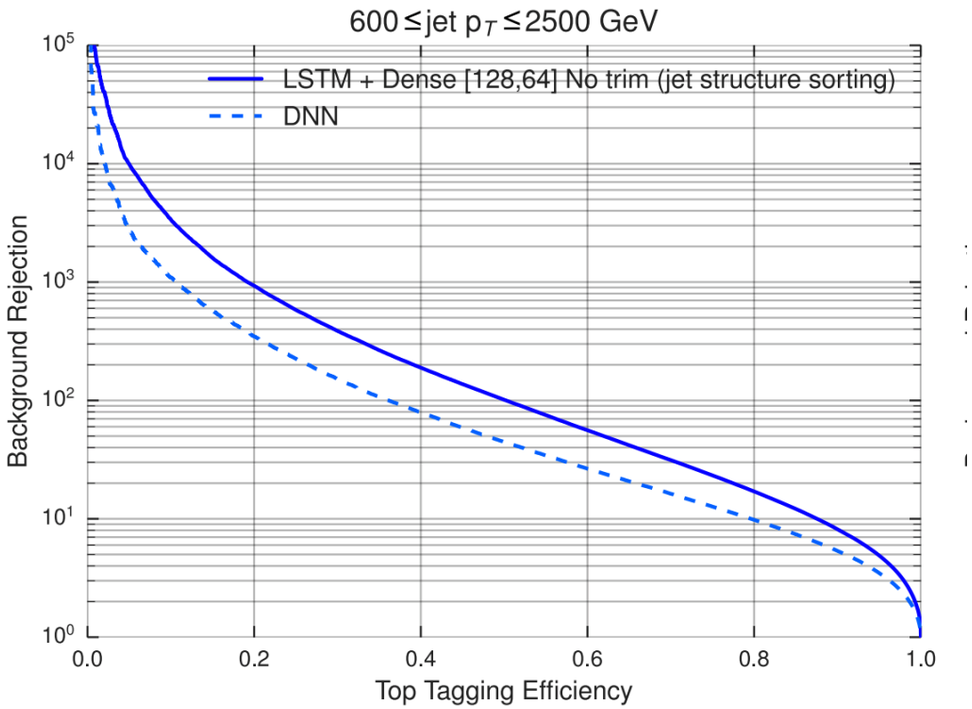
Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC
S Egan, W Fedorko, A Lister, J Pearkes, C Gay
arXiv: 1711.09059

LSTM width of 128, fully connected layer of 64 nodes



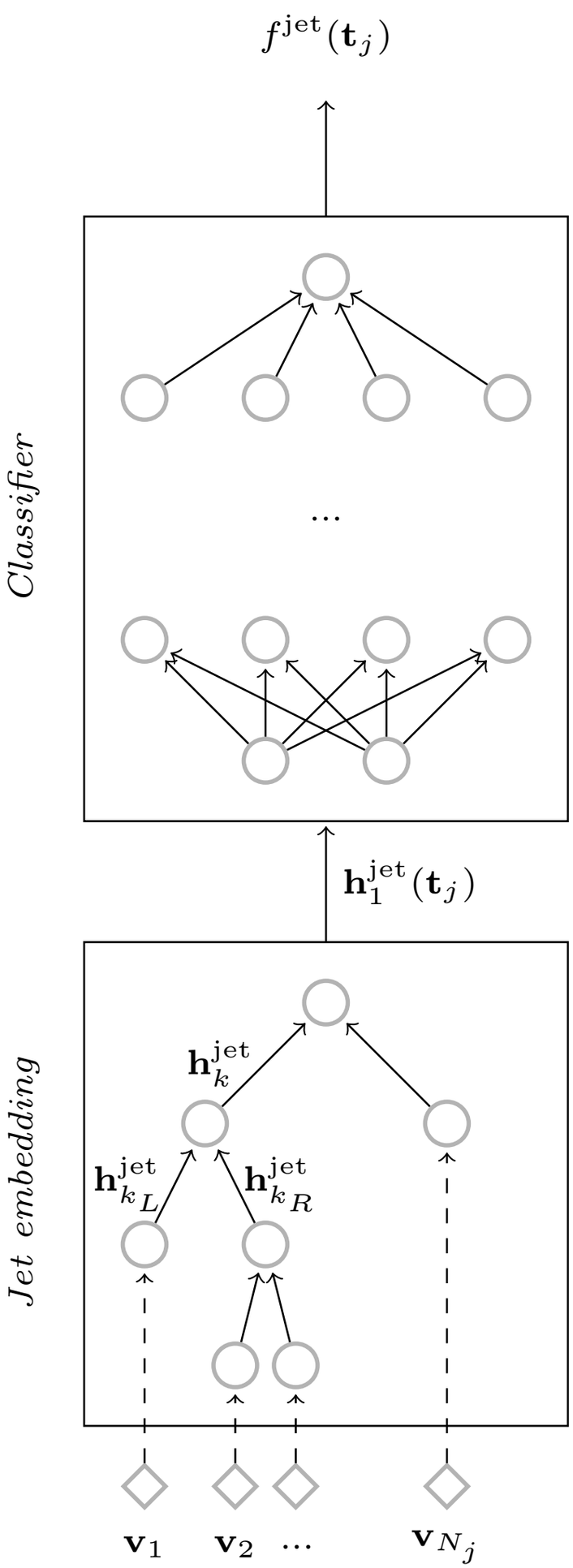
if $d_{36} < d_{45}$: Input list ordering 1-2-3-4-5

if $d_{36} > d_{45}$: Input list ordering 4-5-1-2-3



Sequentially uncluster jet using Anti-Kt order to define input order

Recursive



Recursive embedding using clustering history

Map one jet with arbitrary constituents to a fixed length vector

$$\mathbf{h}_k^{\text{jet}} = \begin{cases} \mathbf{u}_k & \text{if } k \text{ is a leaf} \\ \sigma\left(W_h \begin{bmatrix} \mathbf{h}_{k_L}^{\text{jet}} \\ \mathbf{h}_{k_R}^{\text{jet}} \\ \mathbf{u}_k \end{bmatrix} + b_h\right) & \text{otherwise} \end{cases} \quad (2)$$

$$\mathbf{u}_k = \sigma\left(W_u g(\mathbf{o}_k) + b_u\right) \quad (3)$$

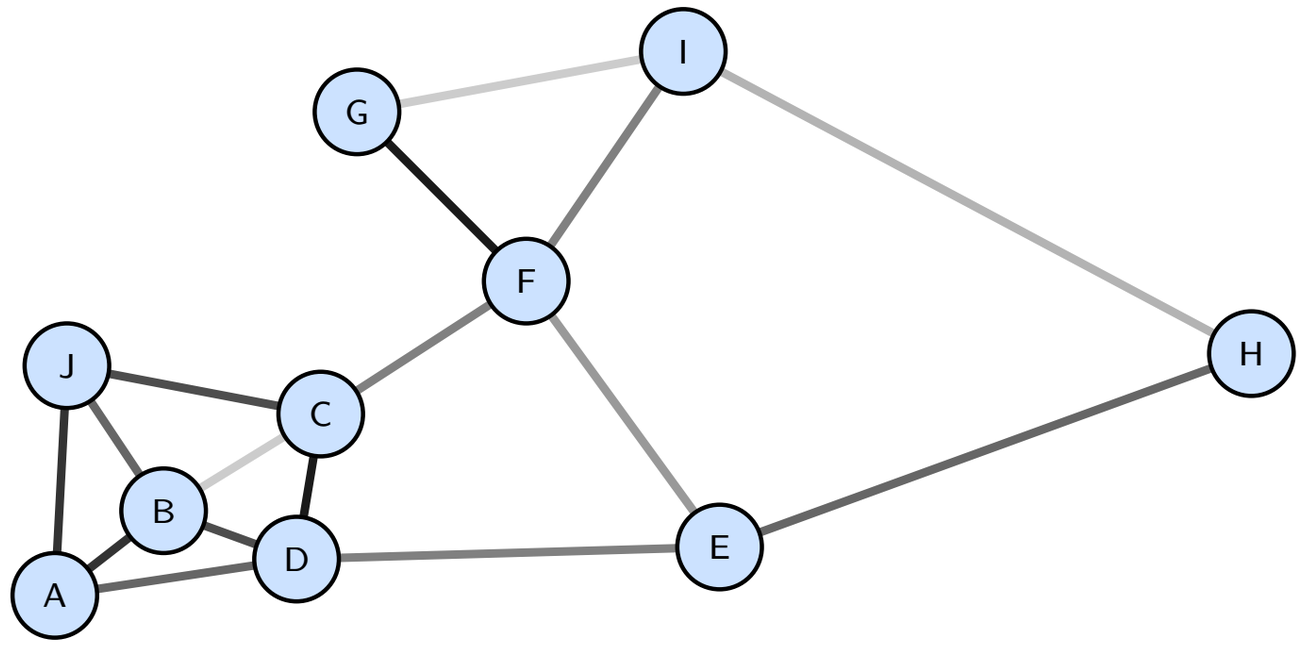
$$\mathbf{o}_k = \begin{cases} \mathbf{v}_{i(k)} & \text{if } k \text{ is a leaf} \\ \mathbf{o}_{k_L} + \mathbf{o}_{k_R} & \text{otherwise} \end{cases} \quad (4)$$

QCD-Aware Recursive Neural Networks for Jet Physics
G Louppe, K Cho, C Becot, K Cranmer
arXiv:1702.00748

Graphs

Message Passing

- *Nodes* in the graph: particles
- *Edges*: "closeness" of Nodes
  - Encoded in Adjacency matrix
  - Can also be learned by algorithm



- Model clustering structure by *sending messages* between nodes

$$\mathbf{h}^{(t+1)} = \mathbf{Gc}(\mathbf{h}^{(t)}) = \rho\left(\sum_{q=1}^{|\mathcal{A}|} A_q \mathbf{h}^{(t)} \theta_q^{(t)}\right)$$

Simple graph update

Neural Message Passing for Jet Physics
I Henrion et al
Procs. of the Deep Learning for Physical Sciences Workshop at NIPS (2017)

Neural Message Passing for Jet Physics
I Henrion et al
Procs. of the Deep Learning for Physical Sciences Workshop at NIPS (2017)

Message Passing

Algorithm 1 Message passing neural network

Require: $N \times S$ array of jet constituents $\mathbf{x}$

▷ N is the number of particles, S is their data dimension

$\mathbf{h} \leftarrow \tanh(W_e\mathbf{x} + \mathbf{b}_e)$ ▷ Embed the jets

for $t = 1, \ldots, T$ **do** ▷ Message passing

$A \leftarrow \textsc{AdjacencyMatrix}_t(\mathbf{h})$

$\mathbf{m} \leftarrow \textsc{Message}_t(A, \mathbf{h})$

$\mathbf{h} \leftarrow \textsc{VertexUpdate}_t(\mathbf{h}, \mathbf{m}, \mathbf{x})$

end for

return $\textsc{Readout}(\mathbf{h})$

Adjacency Matrix:

$$A_{i,j}^{(t)} = \mathrm{softmax}_{\mathrm{row}}\varphi(h_i^{(t)}, h_j^{(t)})$$

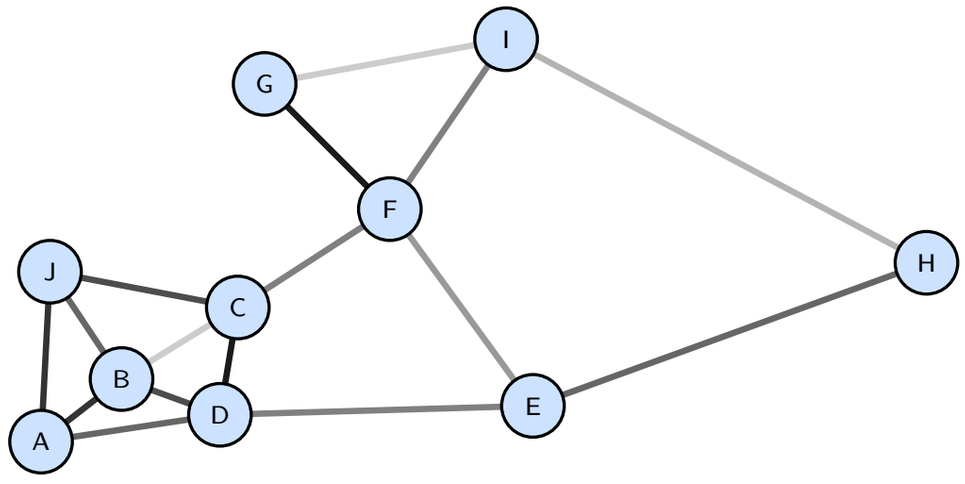
Message from all others to i:

$$m_i^{(t)} = \tanh\left(\sum_j A_{i,j}^{(t)} h_j^{(t)}\right)$$

Vertex Update:

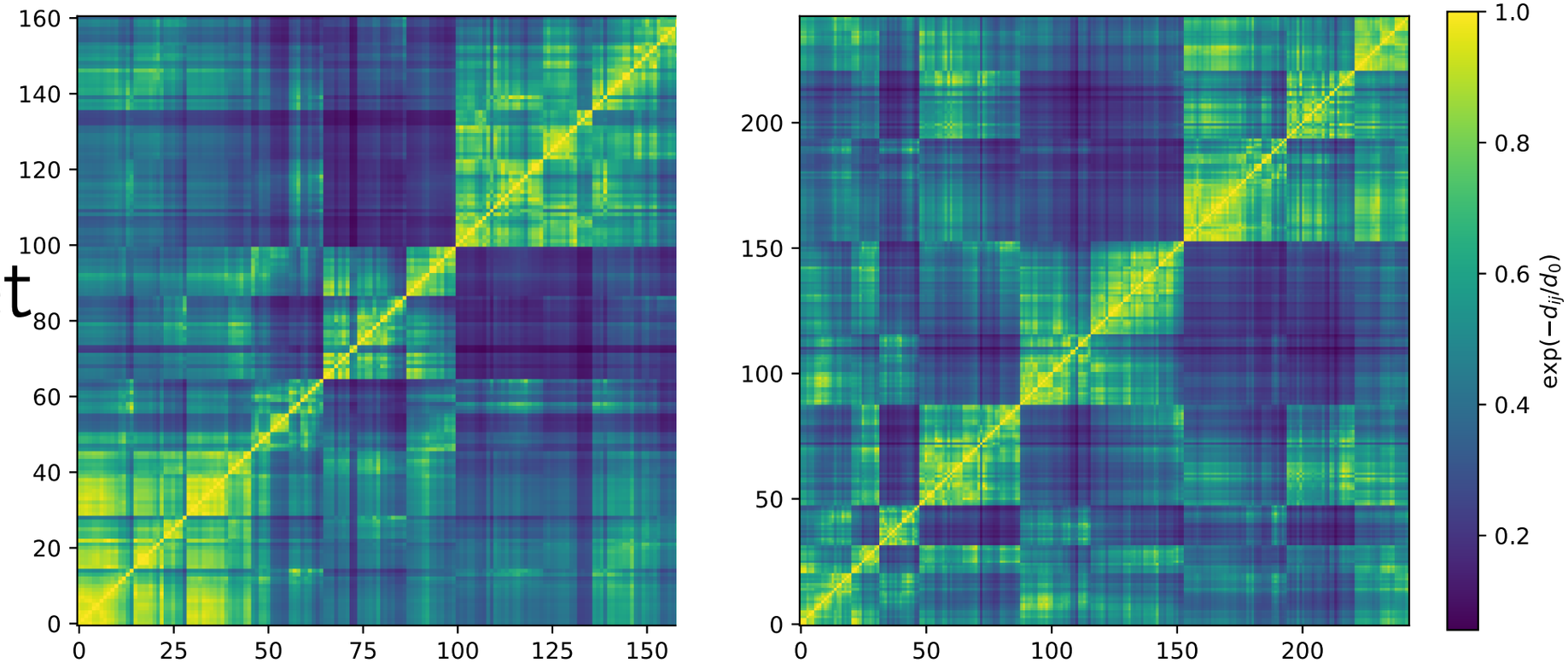
$$h_i^{(t+1)} = \mathrm{GRU}(h_i^{(t)}, [m_i^{(t)}, x_i])$$

h.. 40 feautres/vertex



Learned Distance Measure

Single QCD jet



Single W jet

Generalized learning of metric

Jets as graphs:W tagging with neural message passing
I Henrion et al

Physics

Physics Approach

Input is a p_T sorted list of Lorentz four-vectors:
(calo towers or particle flow objects)

$$k_{\mu,i} = \begin{pmatrix} E_0 & E_1 & \cdots & E_N \\ p_{x,0} & p_{x,1} & \cdots & p_{x,N} \\ p_{y,0} & p_{y,1} & \cdots & p_{y,N} \\ p_{z,0} & p_{z,1} & \cdots & p_{z,N} \end{pmatrix}$$

Combination Layer (**CoLa**): create linear combinations: $k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$

Lorentz Layer (**LoLa**): Use resulting matrix to extract physics features.
Main assumption is the Minkowski metric

Fully connected layers for final output

Deep-learning Top Taggers & No End to QCD
A Butter, GK, T Plehn, M Russell
1707.08966

CoLa

- Goal: Allow network to reconstruct substructure axes (top, W, hard subjets, ..) by summing constituents
- (M - (N+1)) x N trainable weights

$$k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$$

$$C = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & C_{1,N+2} & \cdots & C_{1,M} \\ 1 & 0 & 1 & & \vdots & C_{2,N+2} & \cdots & C_{2,M} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 1 & C_{N,N+2} & \cdots & C_{N,M} \end{pmatrix}$$

Sum of all constituents

trainable linear combinations

Diagonal matrix
(pass-through constituents)

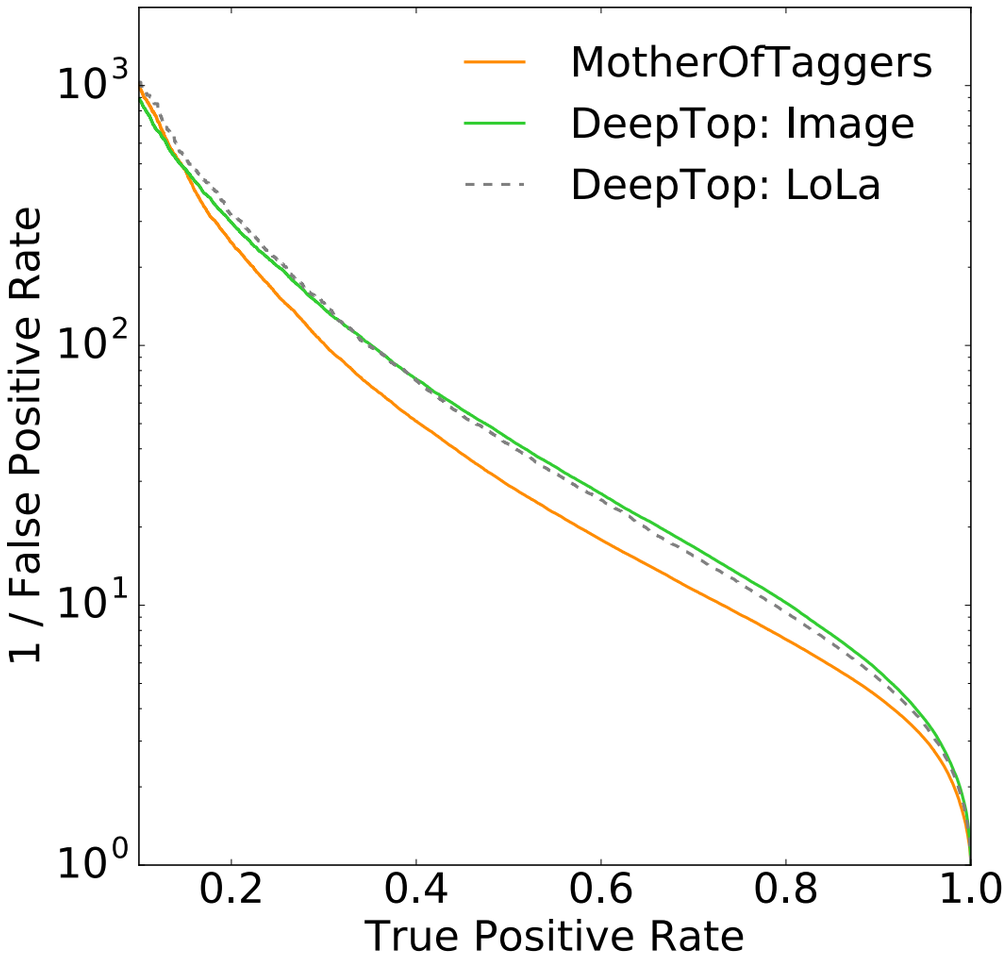
LoLa

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

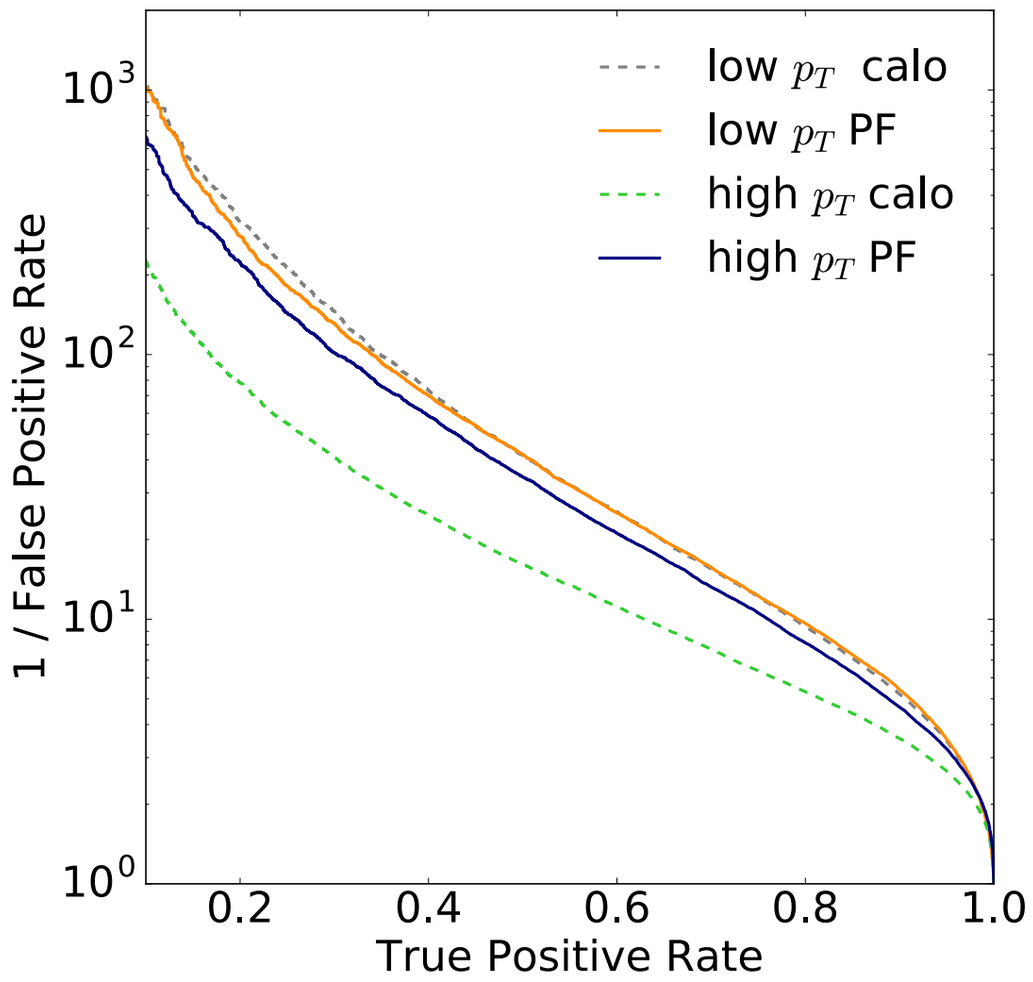
- Transforms M Lorentz-vectors into M vectors with P components
- Using:
  - Per pseudo-jet variables: $\widetilde{k}_{\mu,i} \rightarrow \widetilde{k}_{0,i}$

    $\widetilde{k}_{\mu,i} \rightarrow k_{\mu,i} k_{\nu,i} \eta^{\mu\nu}$
  - Trainable sums: $\widetilde{k}_{\mu,i} \rightarrow \widetilde{k}_{0,j} A_{ij}$
  - Sum of differences: $\widetilde{k}_{\mu,i} \rightarrow \sum_j (\widetilde{k}_i - \widetilde{k}_j)_\mu (\widetilde{k}_i - \widetilde{k}_j)_\nu \eta^{\mu\nu} B_{ij}$

Still fine-tuning what needs to be included..



Perfomance equal to image approach



Clear gain at high pT

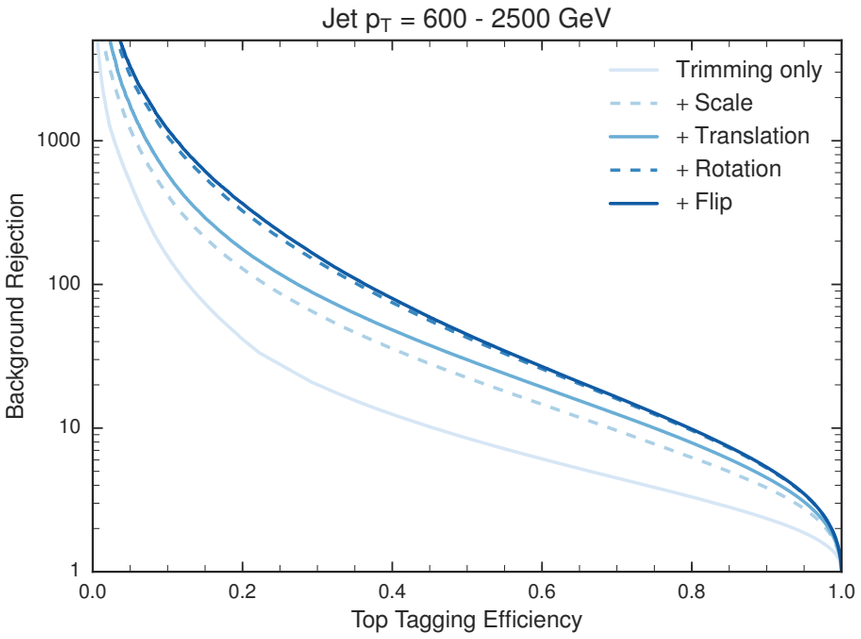
Ansatz

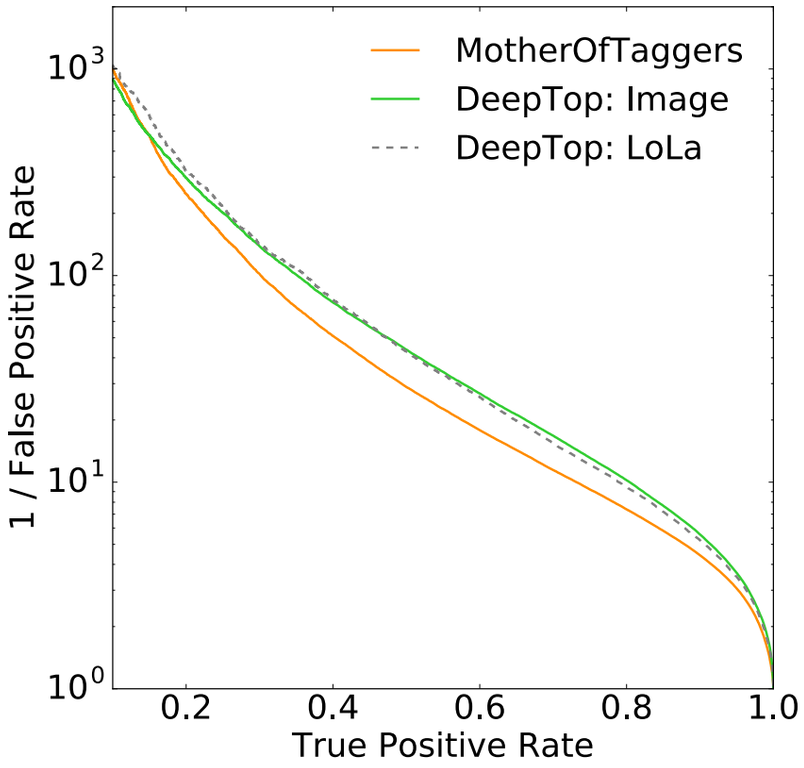
$$\mathrm{diag}(-1,1,1,1) \rightarrow \mathrm{diag}(K,L,M,N)$$

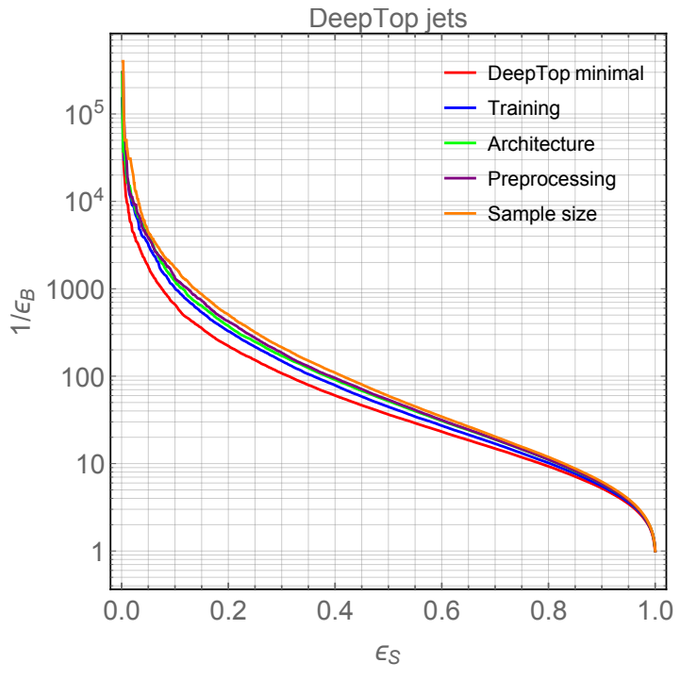
Metric learned

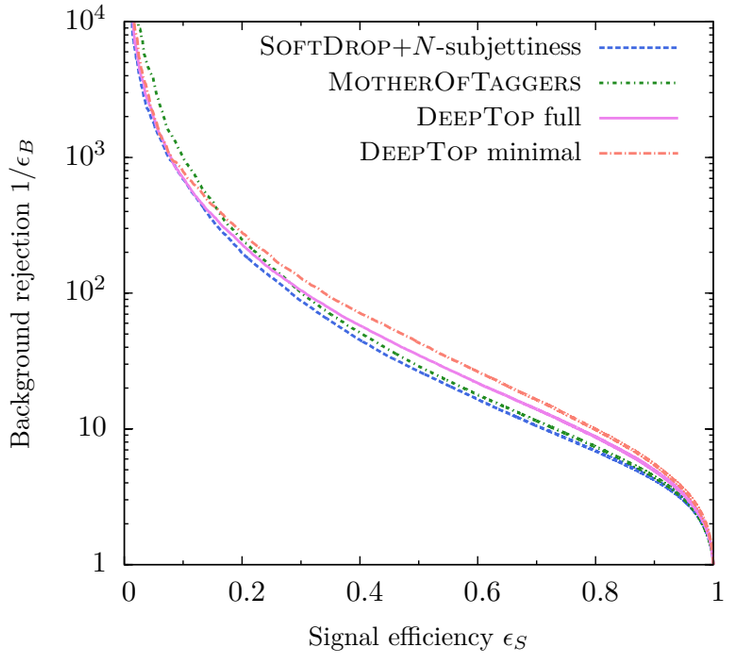
$$g = \mathrm{diag}(\quad 0.99 \pm 0.02,$$
$$-1.01 \pm 0.01, -1.01 \pm 0.02, -0.99 \pm 0.02)$$

Performance Overview









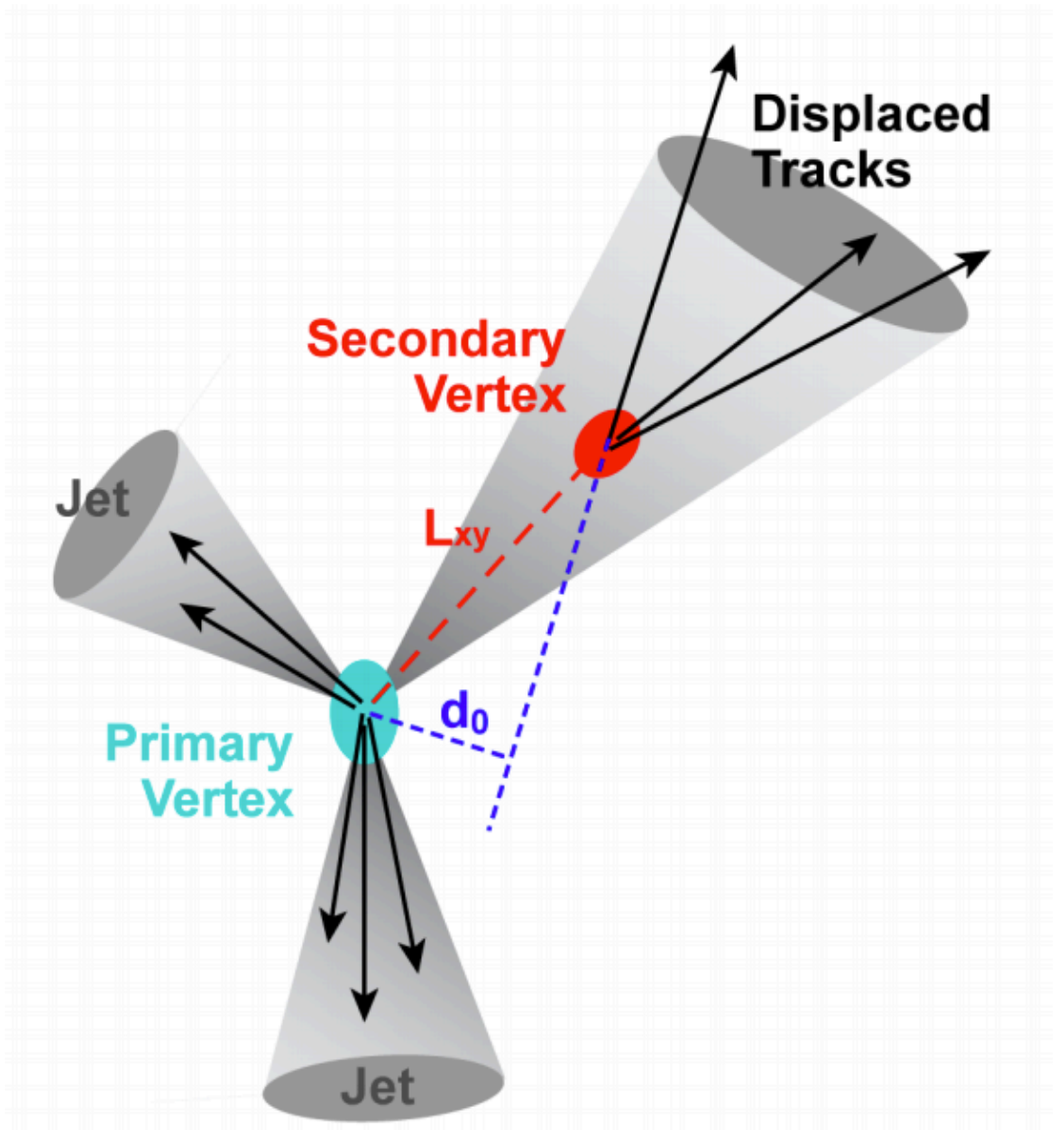
(Urgently) needed: Comparison of algorithms/architectures on a common samples

Our top tagging reference sample:
https://goo.gl/XGYju3

Other Applications

B Tagging

B Mesons



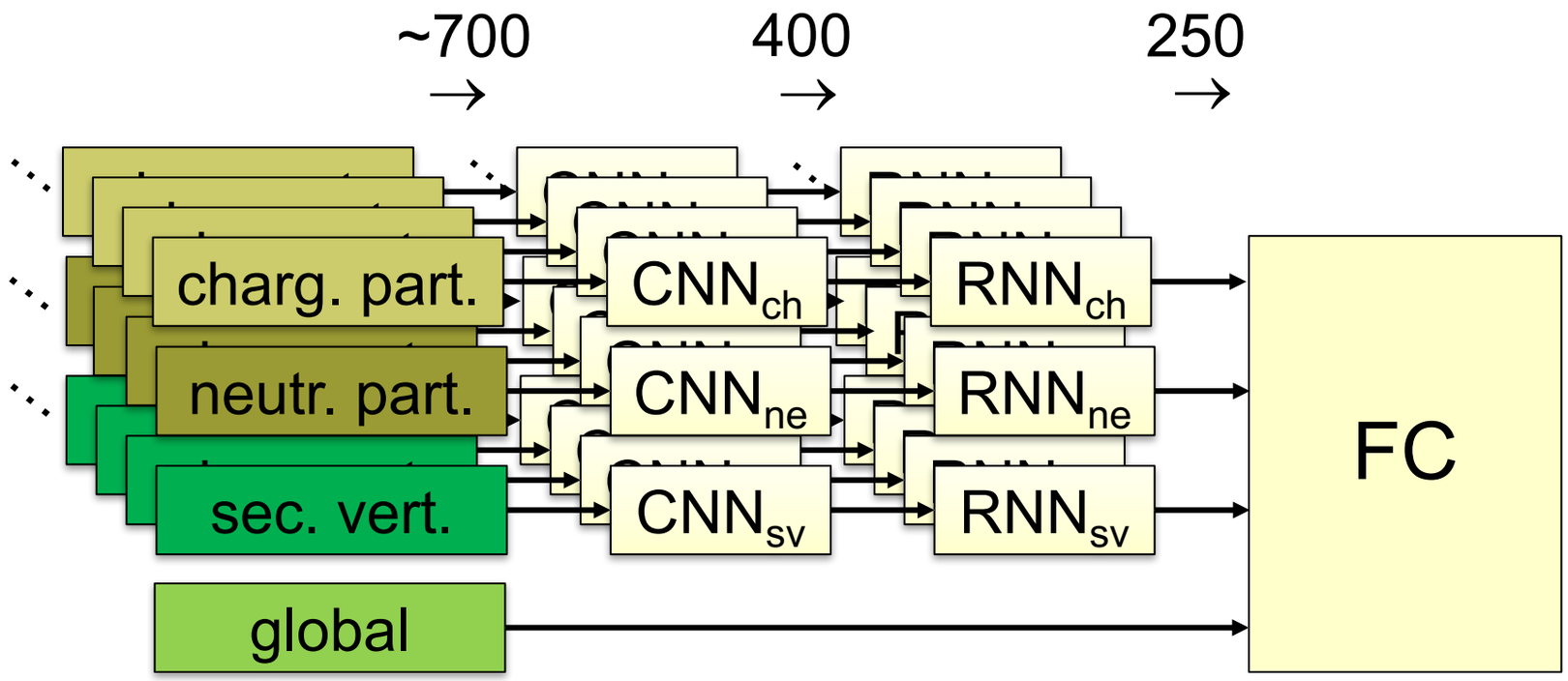
- Mass: 5-6 GeV
- Lifetime (in restframe): 1.5 x 10^{-12} s
- Decay length (in restframe): 0.45 mm
- Lab frame decay length:
  - Assume momentum ~30 GeV
  - v: ~0.99 c
  - γ: ~2
  - **Decay length: ~1 mm**

b quark identification in CMS

- *Classic approach*
  - CSVv2 (Combined Secondary Vertex version 2), uses secondary vertex and track-based lifetime variables combined using a (shallow) neural network
- DeepCSV
  - Same variables, more charged particle tracks, fully connected network with 5 hidden layers and 100 nodes/layer
- DeepFlavour
  - 25 charged particle flow candidates (16 properties)
  - 25 neutral particle flow candidates (6 properties)
  - 4 secondary vertices (17 properties)
  - Deep architecture

CMS DP 2017-013

b Quark Identification



~ 700 inputs and 250.000 model parameters

1x1 Convolutions

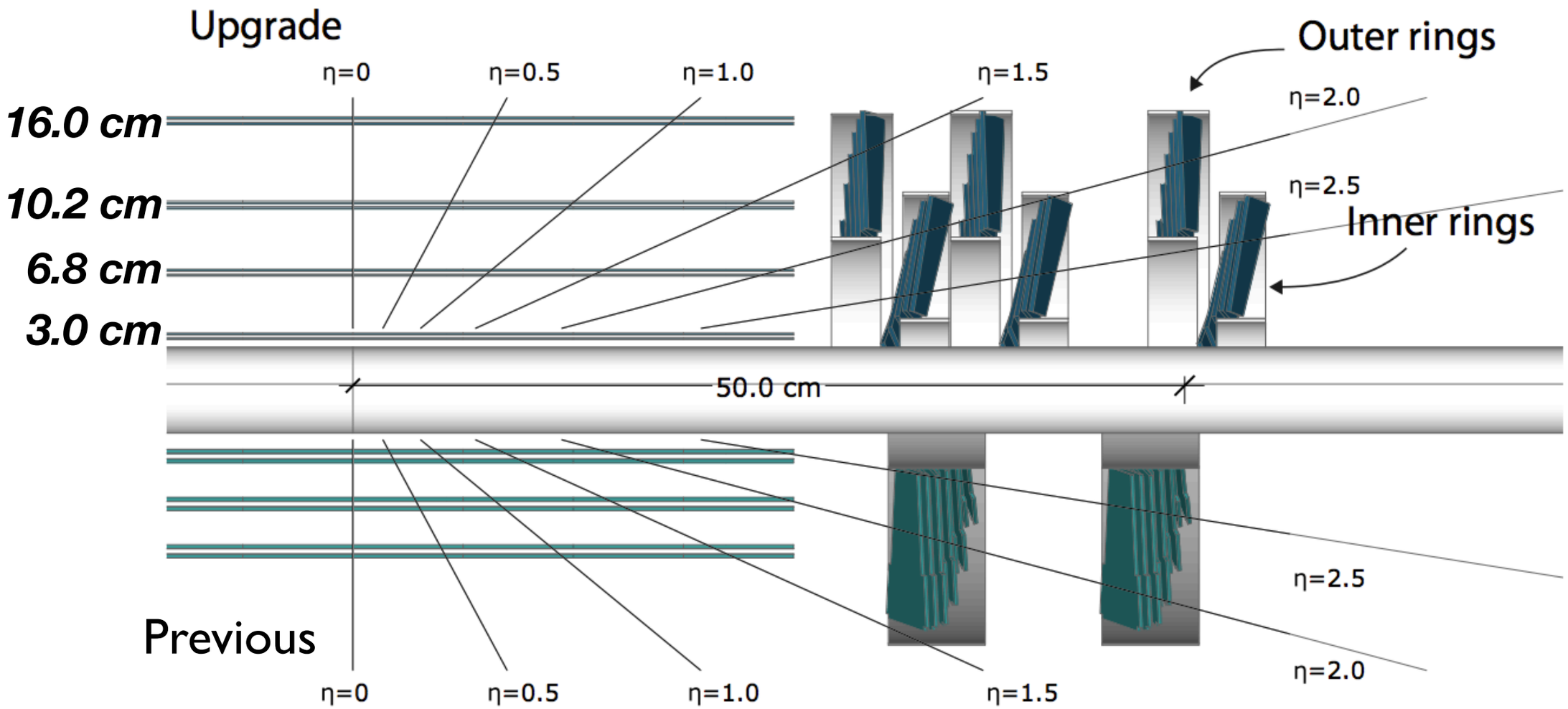
64-32-32-8 for charged/vertices

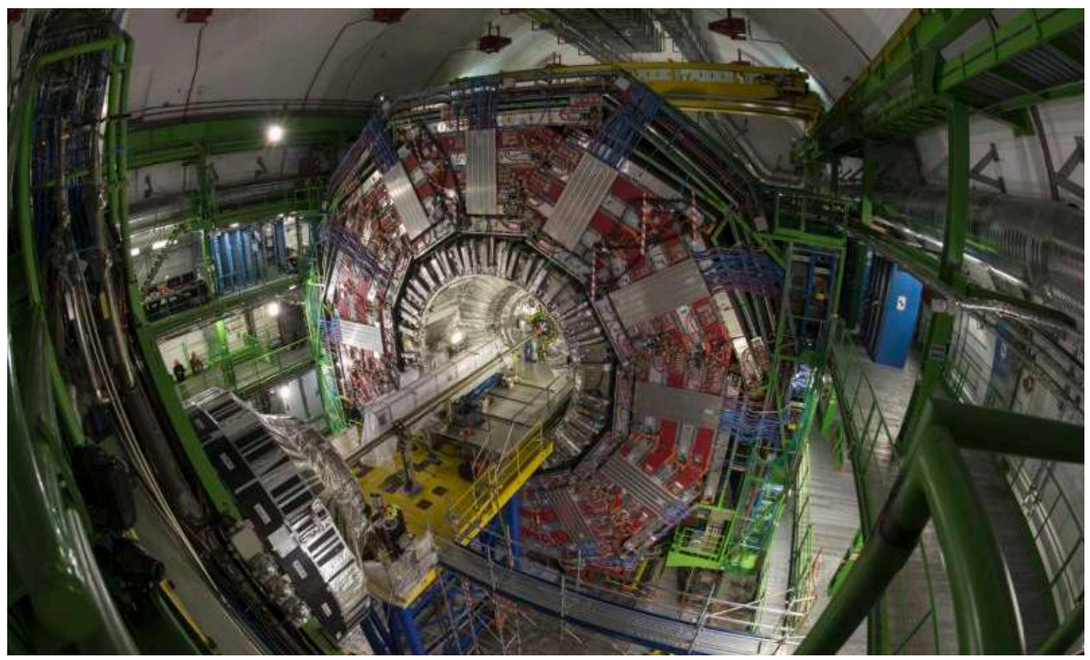
32-16-8 for neutral

Machine Learning for Jet Physics in CMS
Markus Store (for CMS)
Jets in ML Workshop, Berkeley, 2017

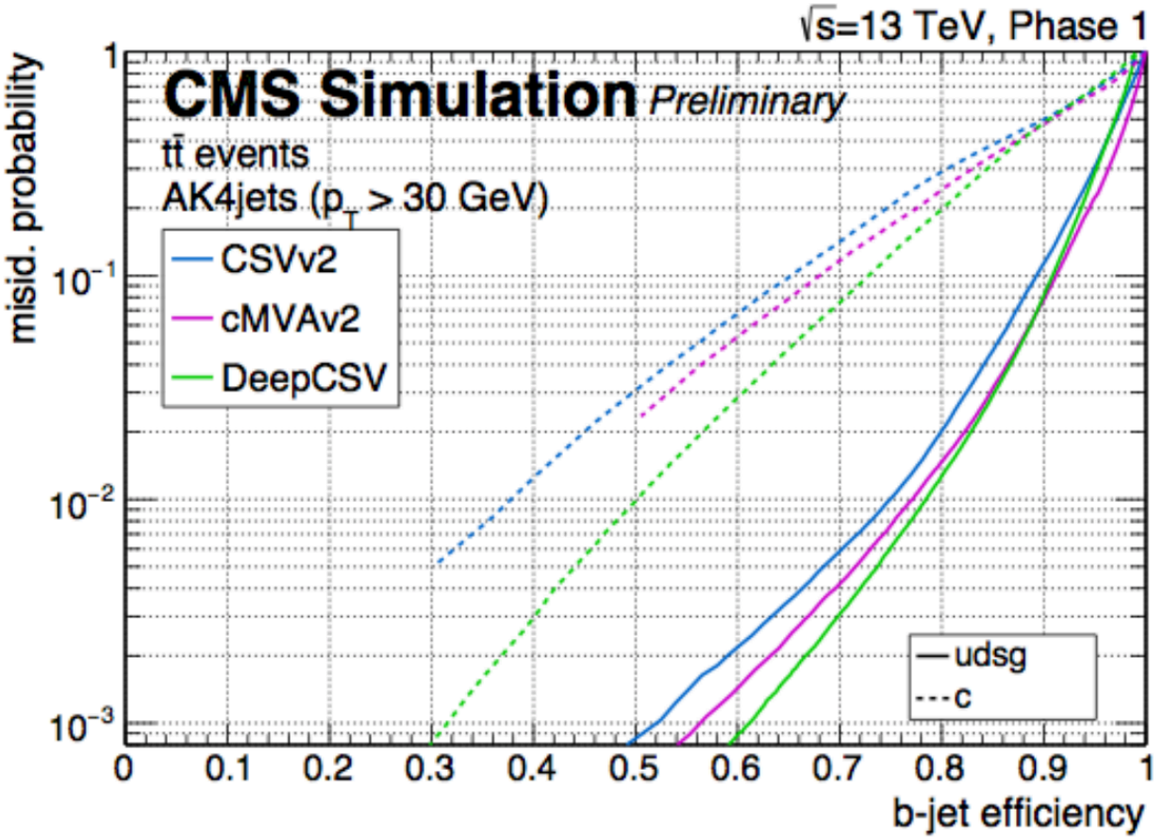
CMS DP 2017-013

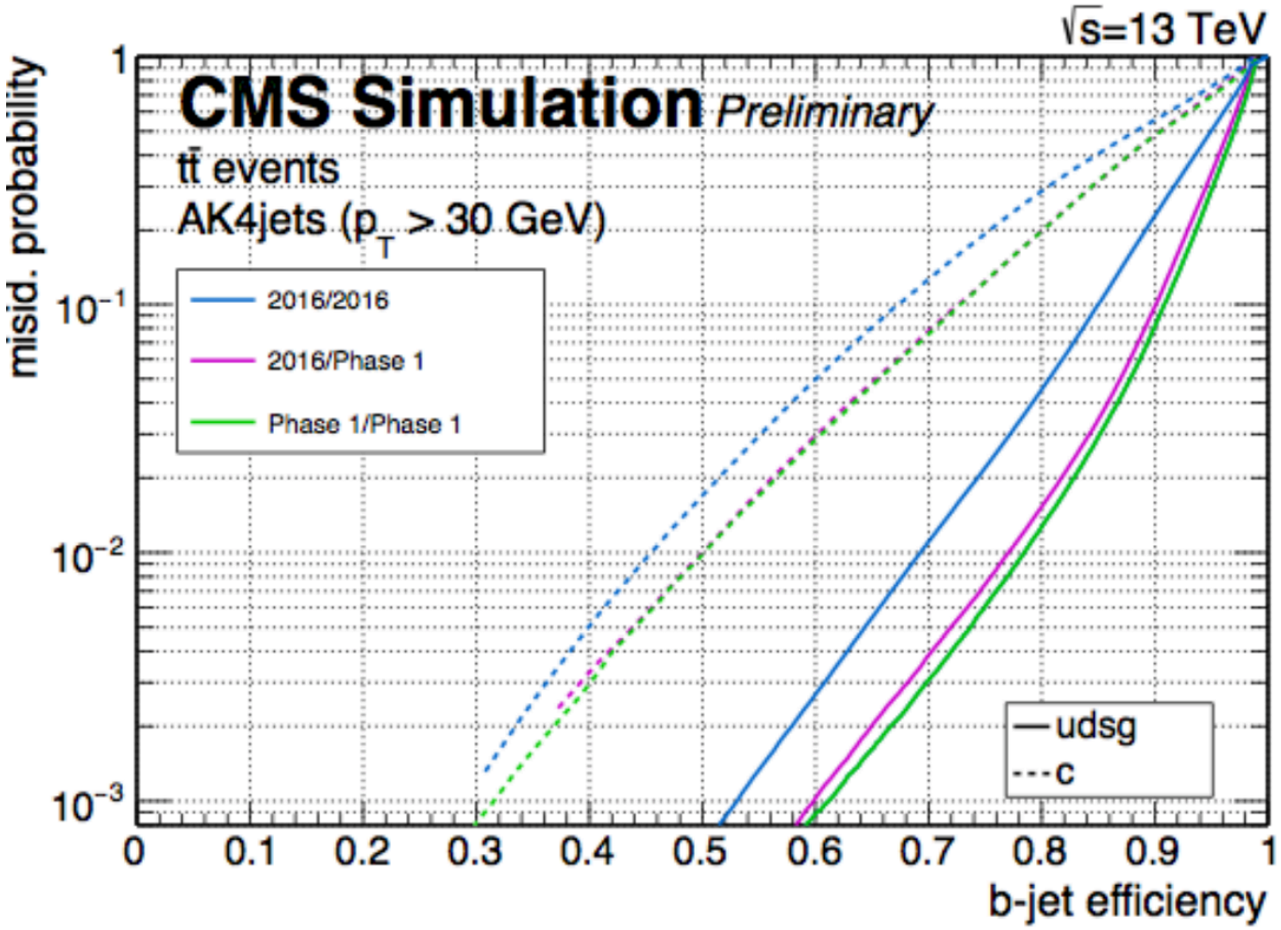
CMS Pixel Upgrade

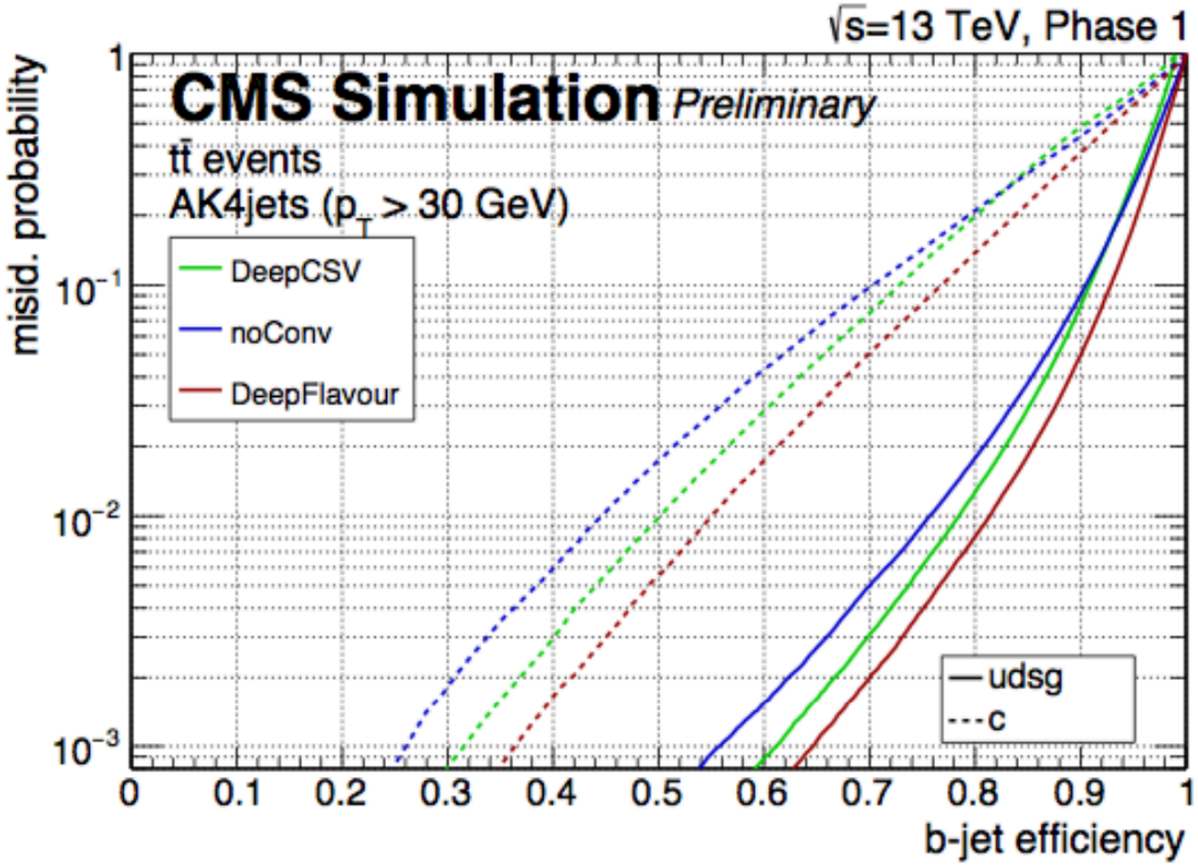




Impact



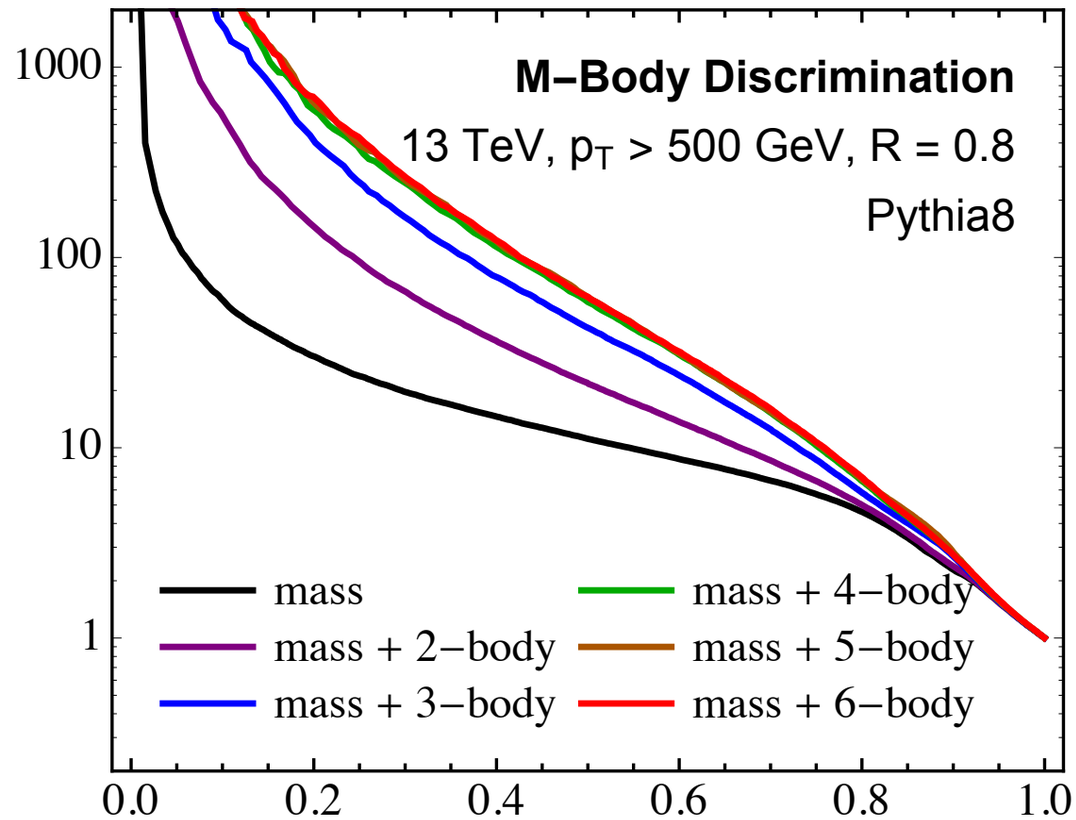




CMS DP 2017-013

New Variables

How much information is in a jet?



$$\tau_N^{(\beta)} = \frac{1}{p_{TJ}} \sum_{i \in \text{Jet}} p_{Ti} \min \left\{ R_{1i}^{\beta}, R_{2i}^{\beta}, \ldots, R_{Ni}^{\beta} \right\}$$

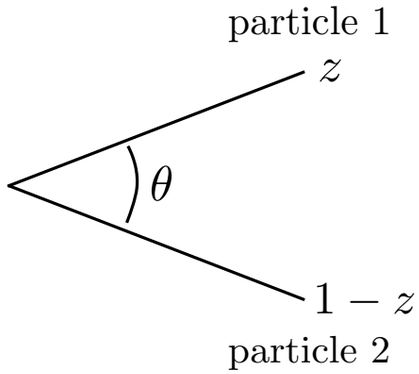
2-body: $\tau_1^{(1)}, \tau_1^{(2)}$

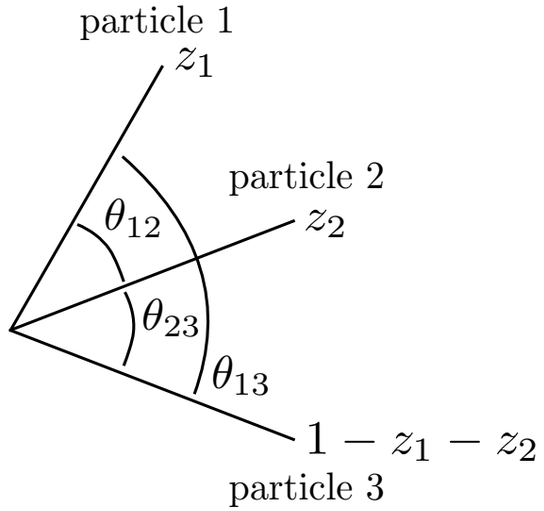
3-body: $\tau_1^{(0.5)}, \tau_1^{(1)}, \tau_1^{(2)}, \tau_2^{(1)}, \tau_2^{(2)}$

4-body: $\tau_1^{(0.5)}, \tau_1^{(1)}, \tau_1^{(2)}, \tau_2^{(0.5)}, \tau_2^{(1)}, \tau_2^{(2)}, \tau_3^{(1)}, \tau_3^{(2)}$

5-body: $\tau_1^{(0.5)}, \tau_1^{(1)}, \tau_1^{(2)}, \tau_2^{(0.5)}, \tau_2^{(1)}, \tau_2^{(2)}, \tau_3^{(0.5)}, \tau_3^{(1)}, \tau_3^{(2)}, \tau_4^{(1)}, \tau_4^{(2)}$

6-body: $\tau_1^{(0.5)}, \tau_1^{(1)}, \tau_1^{(2)}, \tau_2^{(0.5)}, \tau_2^{(1)}, \tau_2^{(2)}, \tau_3^{(0.5)}, \tau_3^{(1)}, \tau_3^{(2)}, \tau_4^{(0.5)}, \tau_4^{(1)}, \tau_4^{(2)}, \tau_5^{(1)}, \tau_5^{(2)}$





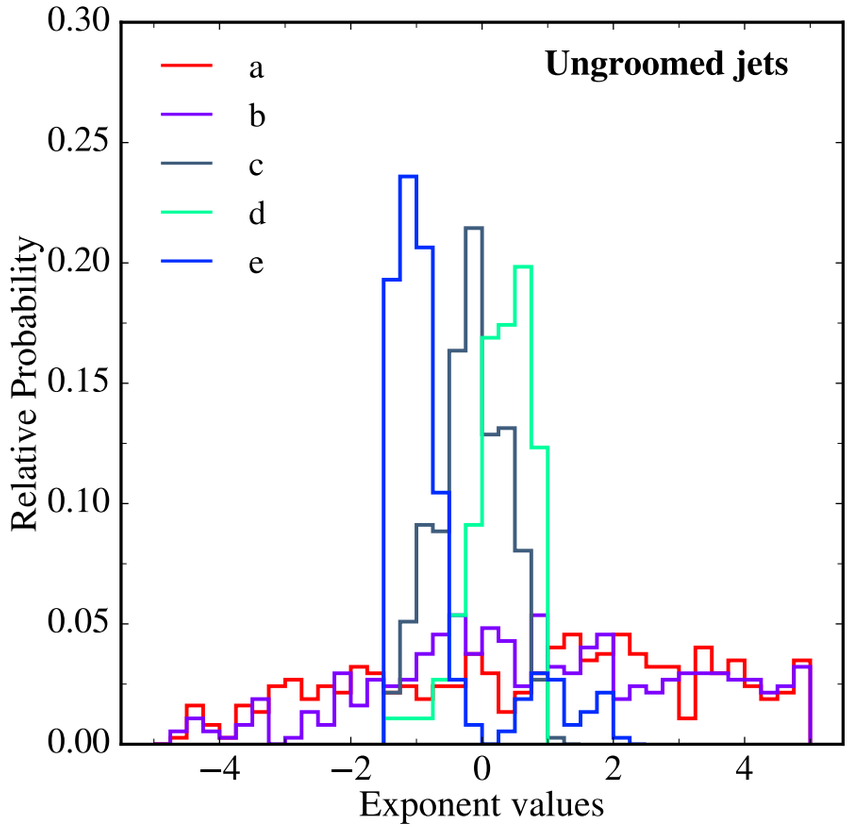
- Study W tagging
- Use n-subjettiness with different exponents to build basis
- Train ANNs with different numbers of variables

How Much Information is in a Jet?
K Datta, A Larkoski
arXiv:1704.08249

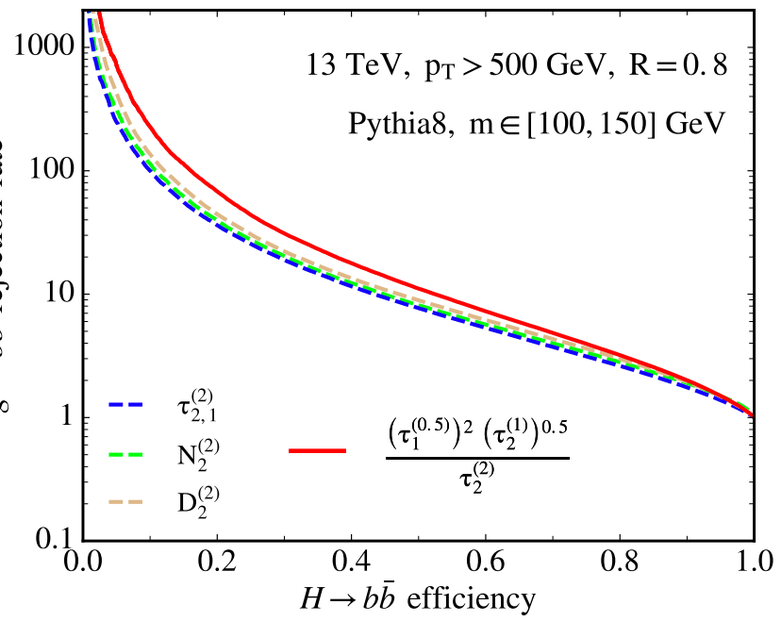
New Variables

$$\beta_3 \equiv \left(\tau_1^{(0.5)}\right)^a \left(\tau_1^{(1)}\right)^b \left(\tau_1^{(2)}\right)^c \left(\tau_2^{(1)}\right)^d \left(\tau_2^{(2)}\right)^e$$

Parametrise phase space

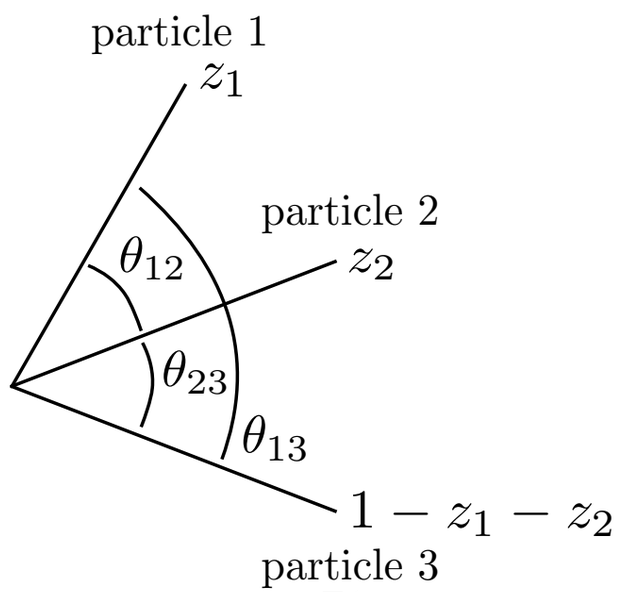


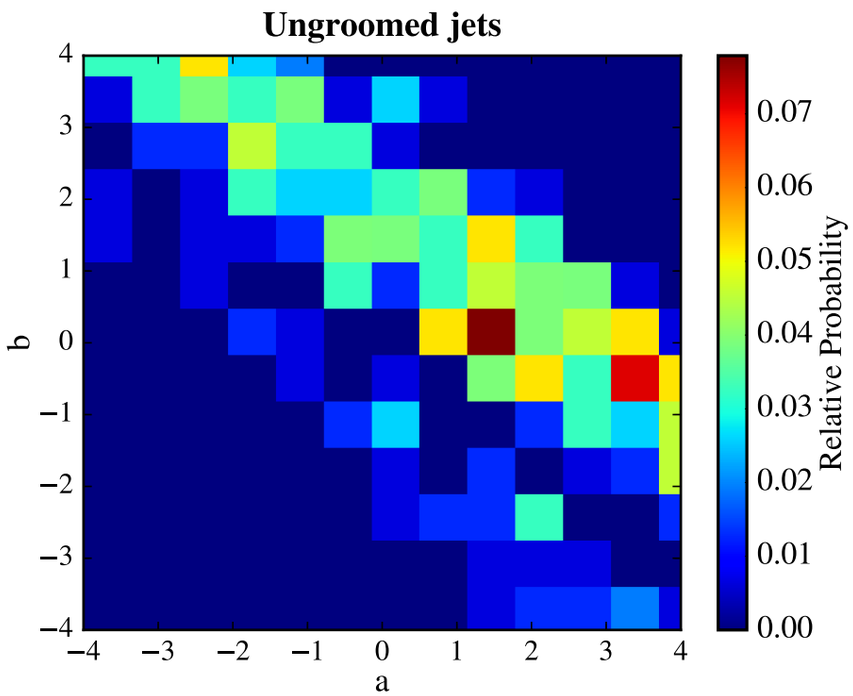
Find optimal exponents using MC



$$\beta_3 = \frac{\left(\tau_1^{(0.5)}\right)^2 \left(\tau_2^{(1)}\right)^{0.5}}{\tau_2^{(2)}}$$

Final variable

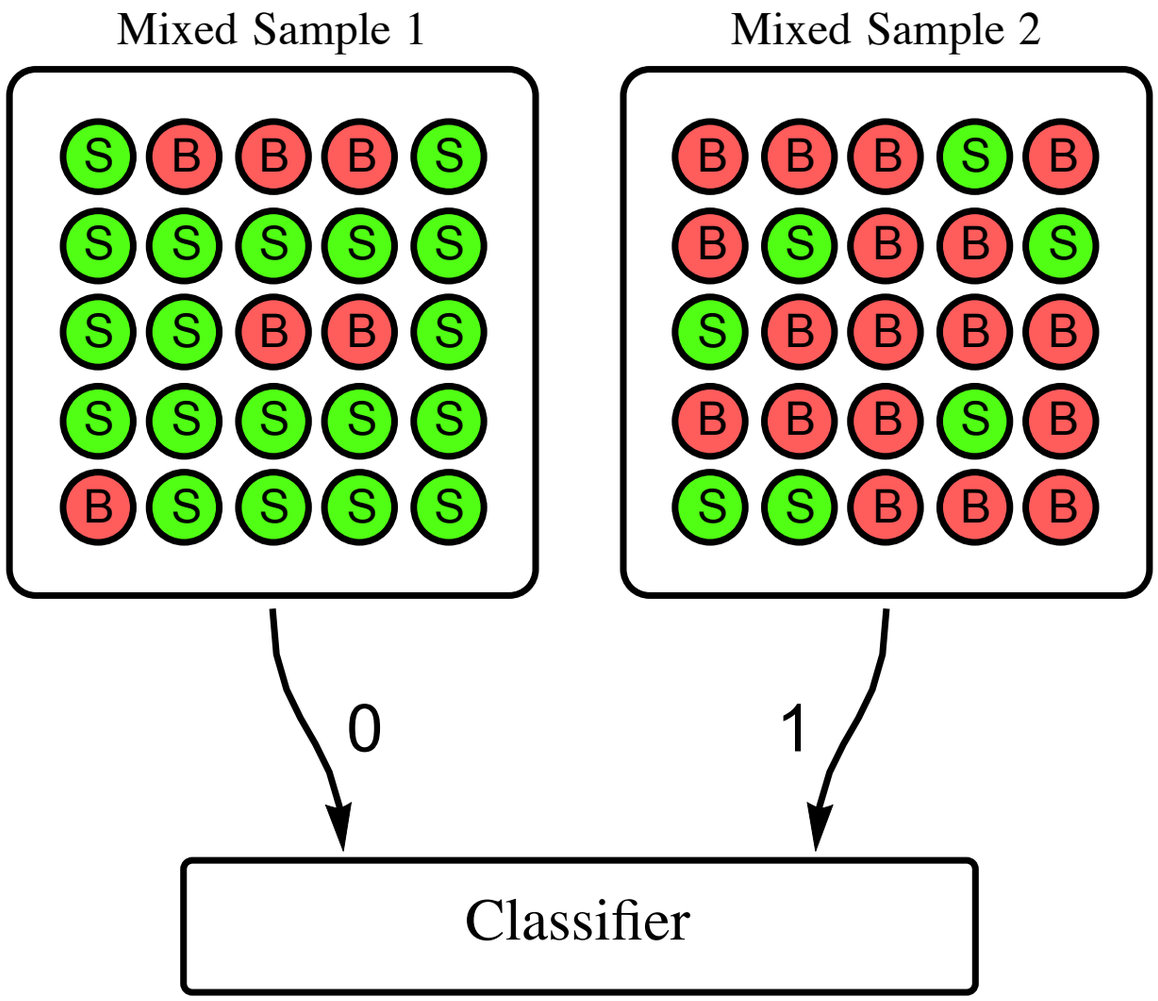


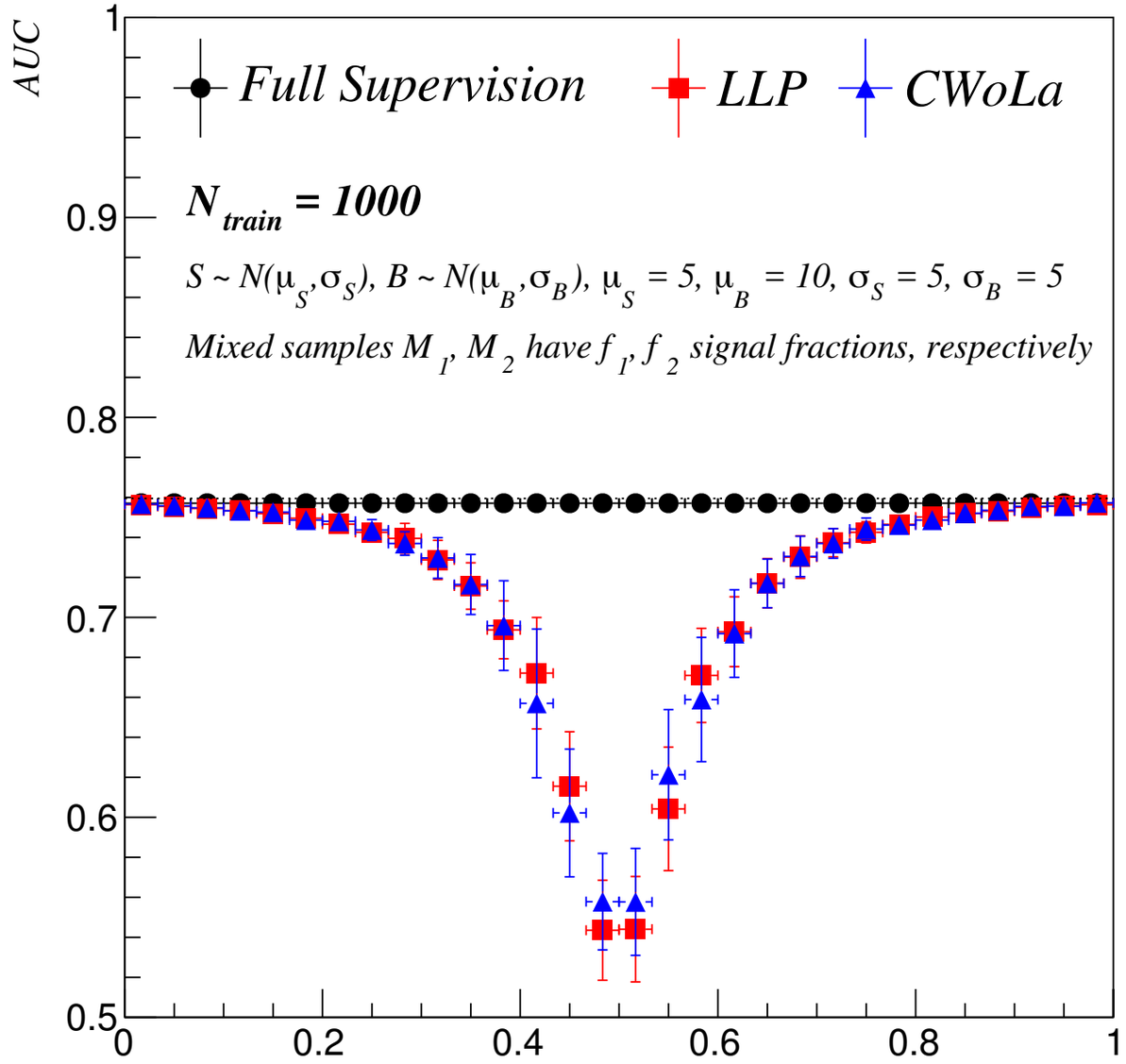


Novel Jet Observables from Machine Learning
K Datta, A Larkoski
arXiv:1710.01305

Weak Supervision

No Labels



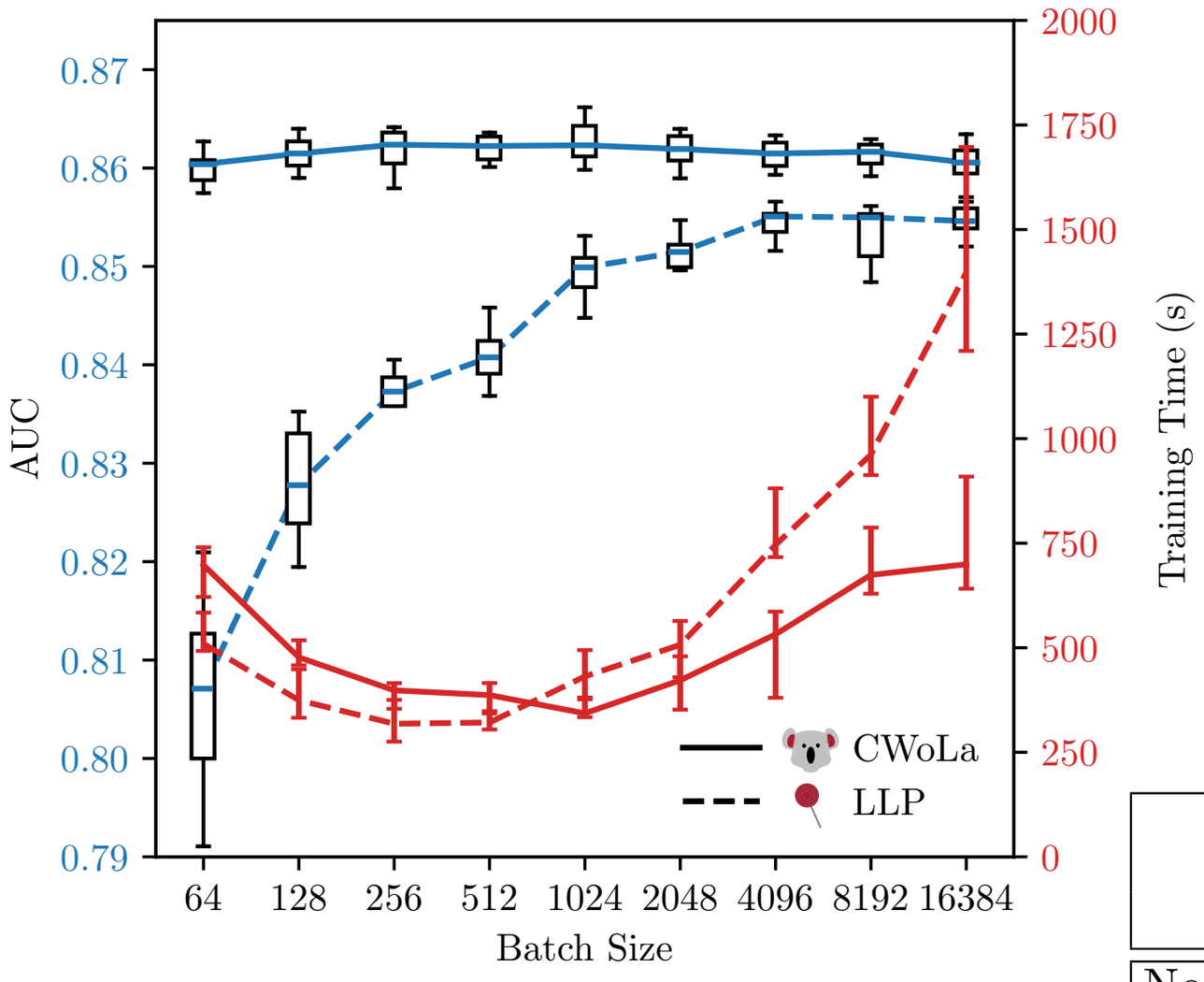


$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1-f_1)\, p_B}{f_2\, p_S + (1-f_2)\, p_B} = \frac{f_1\, L_{S/B} + (1-f_1)}{f_2\, L_{S/B} + (1-f_2)}$$

Weakly Supervised Classification in High Energy Physics
LM Dery, B Nachman, F Rubbo, A Schwartzman, 1702.00414
Learning to Classify from Impure Samples
PT Komiske, EM Metodiev, B Nachman, MD Schwartz, 1801.10158
Classification without labels: Learning from mixed samples in high energy physics, EM Metodiev, B Nachman, J Thaler, 1708.02949

Distinguishing mixed samples is equivalent to signal/background classification!

Learning from fractions



$$\ell_{\text{WCE}} = \sum_a \text{CE}\left(f_a,\ \frac{1}{N}\sum_{i=1}^{N} h(\mathbf{x}_i)\right)$$

- LLP: Learning from Label Proportions
- Modify loss function to learn fractions per mini-batch

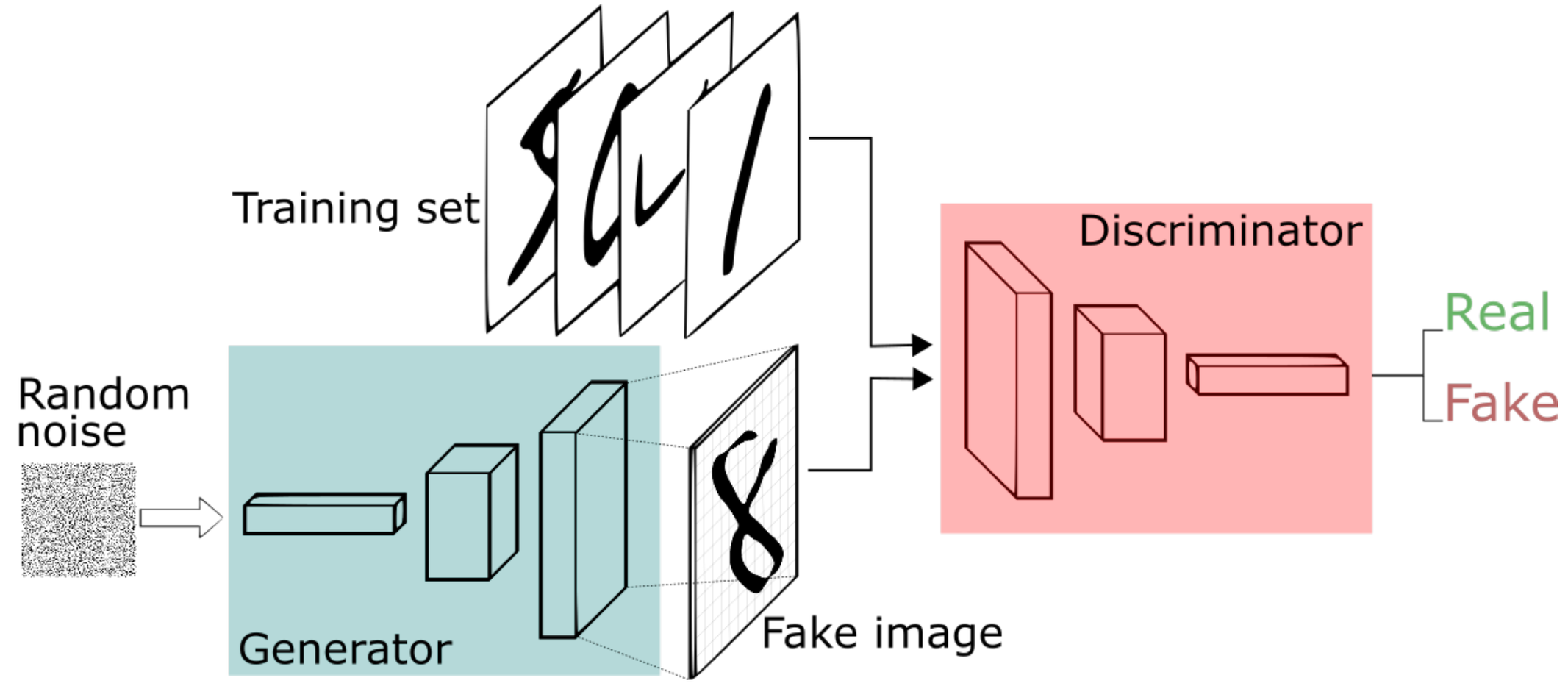
Property	LLP	CWoLa
No need for fully-labeled samples	✓	✓
Compatible with any trainable model	✓	✓
No training modifications needed	✗	✓
Training does not need fractions	✗	✓
Smooth limit to full supervision	✗	✓
Works for > 2 mixed samples	✓	?

Weakly Supervised Classification in High Energy Physics
LM Dery, B Nachman, F Rubbo, A Schwartzman
arXiv:1702.00414

Learning to Classify from Impure Samples
PT Komiske, EM Metodiev, B Nachman, MD Schwartz
arXiv: 1801.10158

Generative Networks

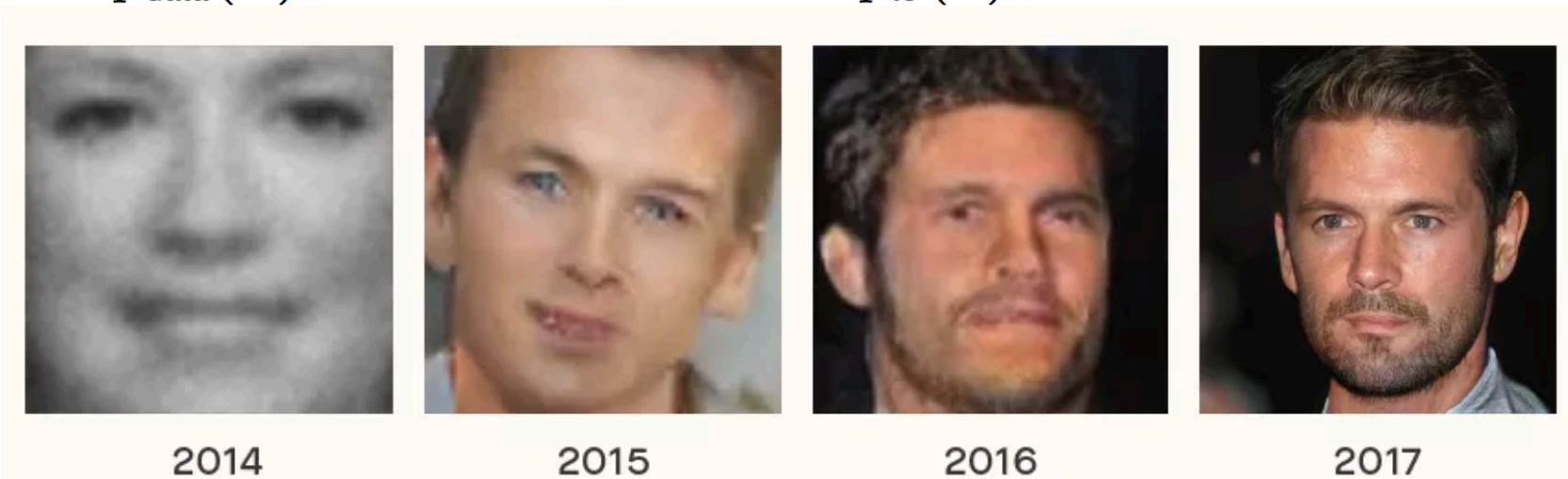
Generative Adverserial



- Two player min-max game:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Generative Adversarial Networks
IJ Goodfellow et al
1406.2661
Image: B. Amos
Progressive Growing of GANs for Improved Quality, Stability, and Variation
T Karras et al, 1710.10196



Divergences

Kullback-Leibler

A measure for the difference of probability distributions:

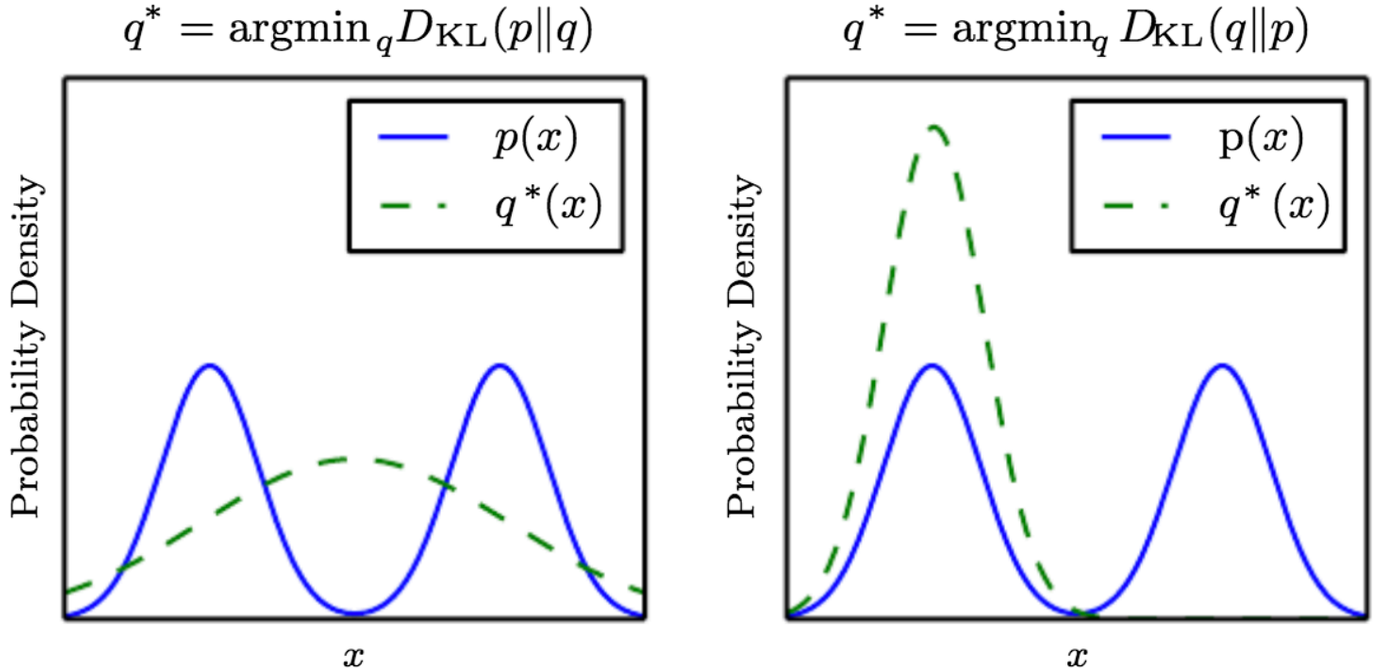
$$D_{\mathrm{KL}}(P\|Q) = -\sum_i P(i)\,\log\frac{Q(i)}{P(i)}$$

Also known as information gain

Amount of information gained when P is used instead of Q

Jensen-Shannon

$$M = \frac{1}{2}(P+Q)$$



What does a GAN do?

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D^*_G(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

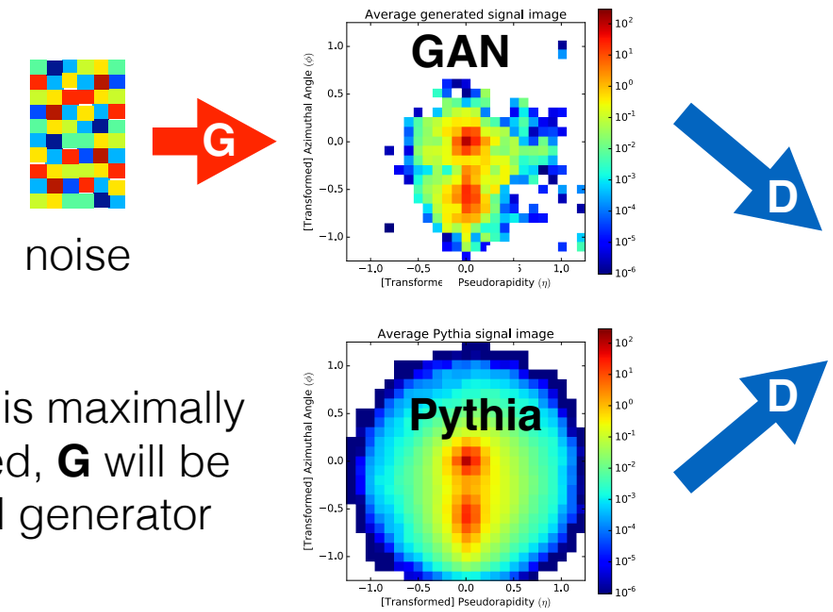
Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

Optimal D and G minimise Jensen-Shannon divergence between data and generator

Generative Adversarial Networks
IJ Goodfellow et al
1406.2661

Generation of 3D EM Showers

Generative Adversarial Networks (GAN):
*A two-network game where one **maps noise to images** and one **classifies images as fake or real**.*



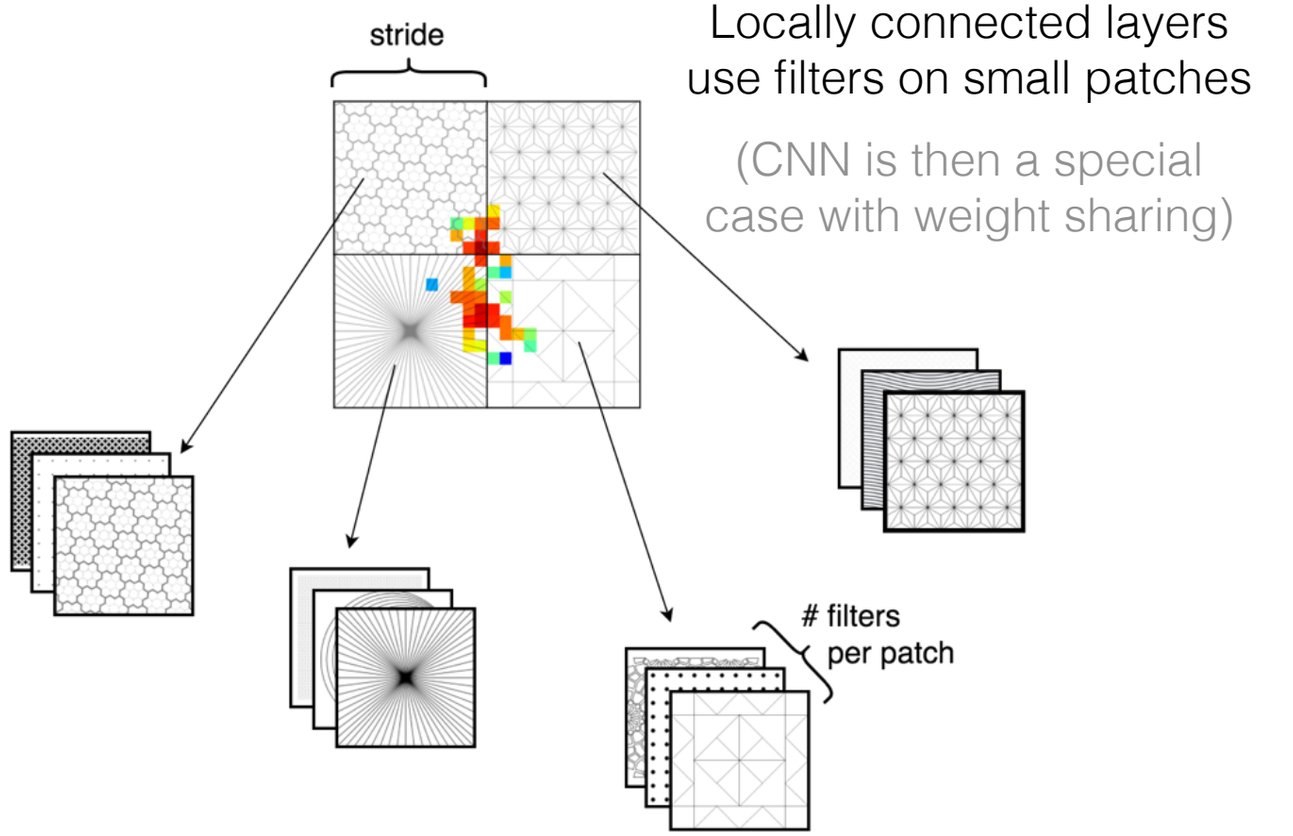
When **D** is maximally confused, **G** will be a good generator

- Generate: Electron, Photon, Pion
- Energy: 1..100 GeV (uniform)
- 3 Layer, LAr Calorimeter, Geant4

LAGAN

Locally connected layers use filters on small patches

(CNN is then a special case with weight sharing)

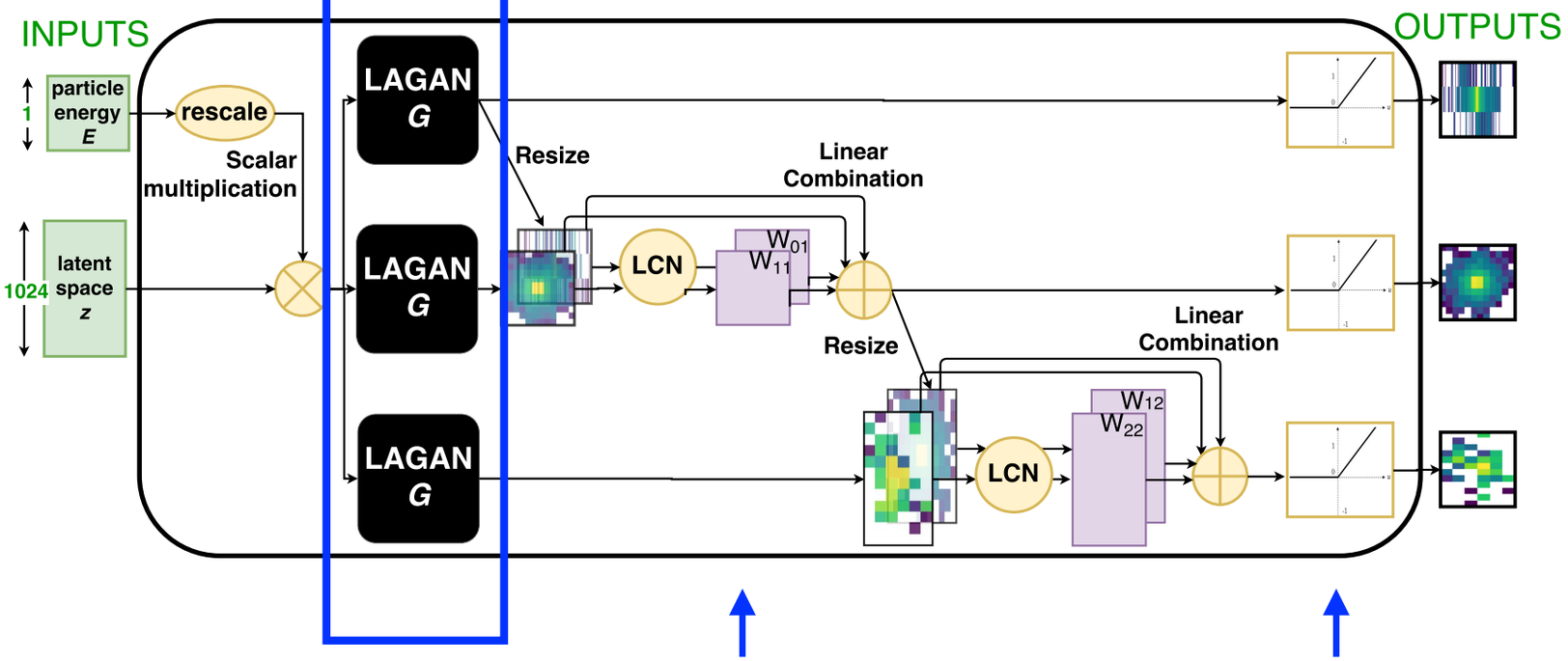


L. de Oliveira, M Paganini, B Nachman:
Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis,, Comput Softw Big Sci *(2017) 1:4,*
Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeters, PRL 120, 042003
CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks, PR D 97, 014021

Generator:

One ‘jet image’ per calo layer

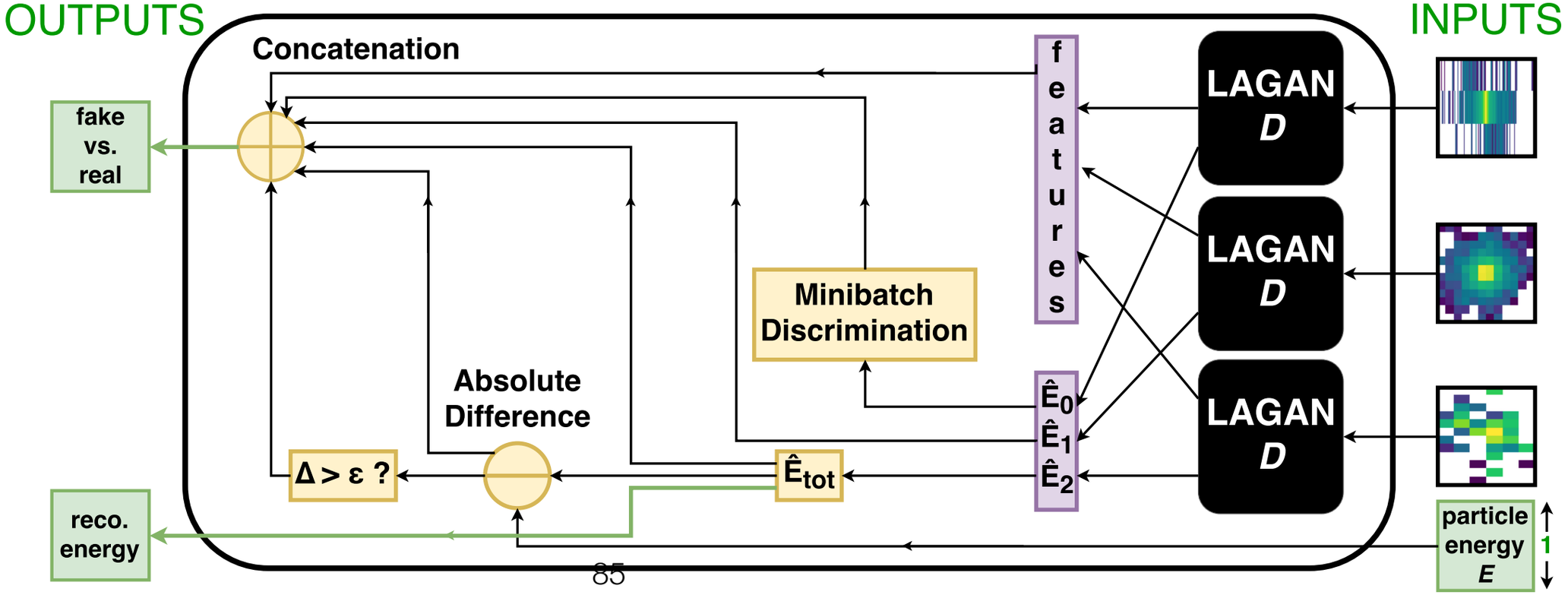
One network per particle type; input particle energy



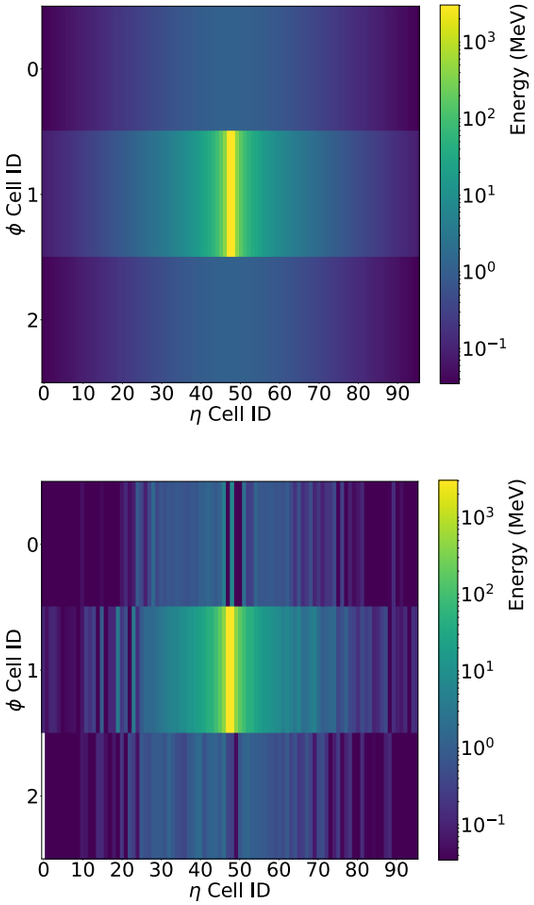
use layer i as input to layer i+1

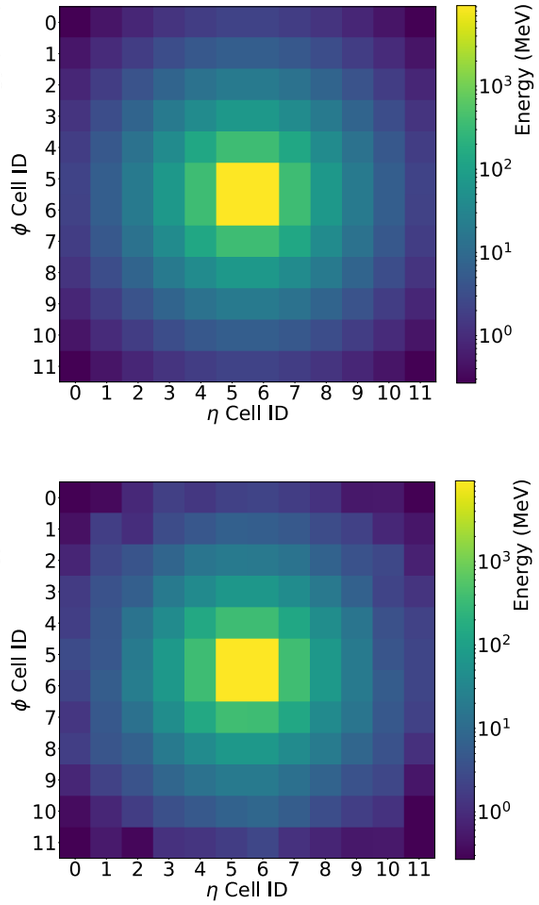
ReLU to encourage sparsity

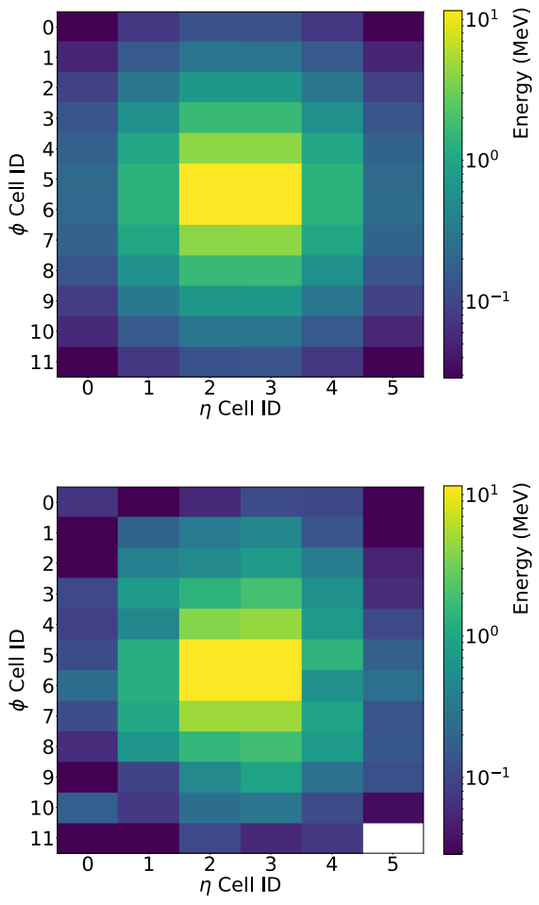
Discriminator:

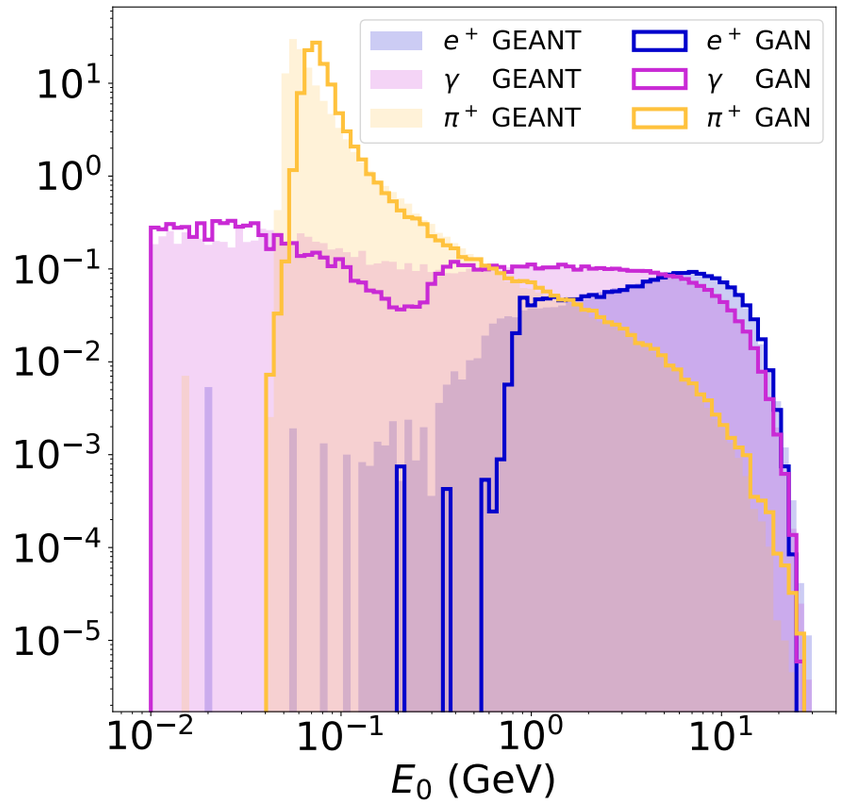


Geant4



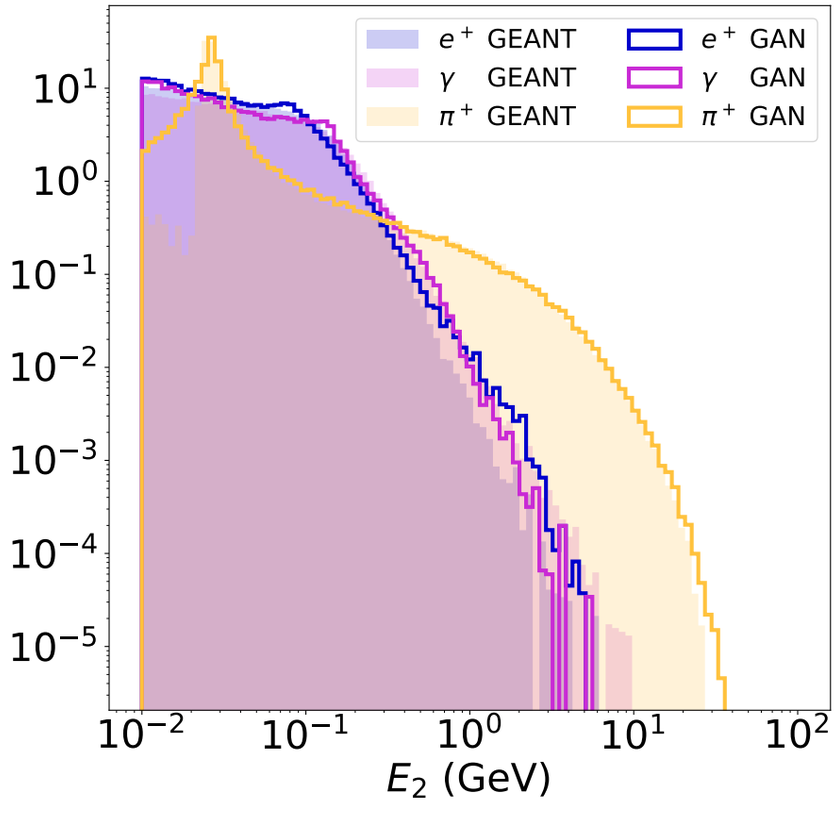






CaloGAN

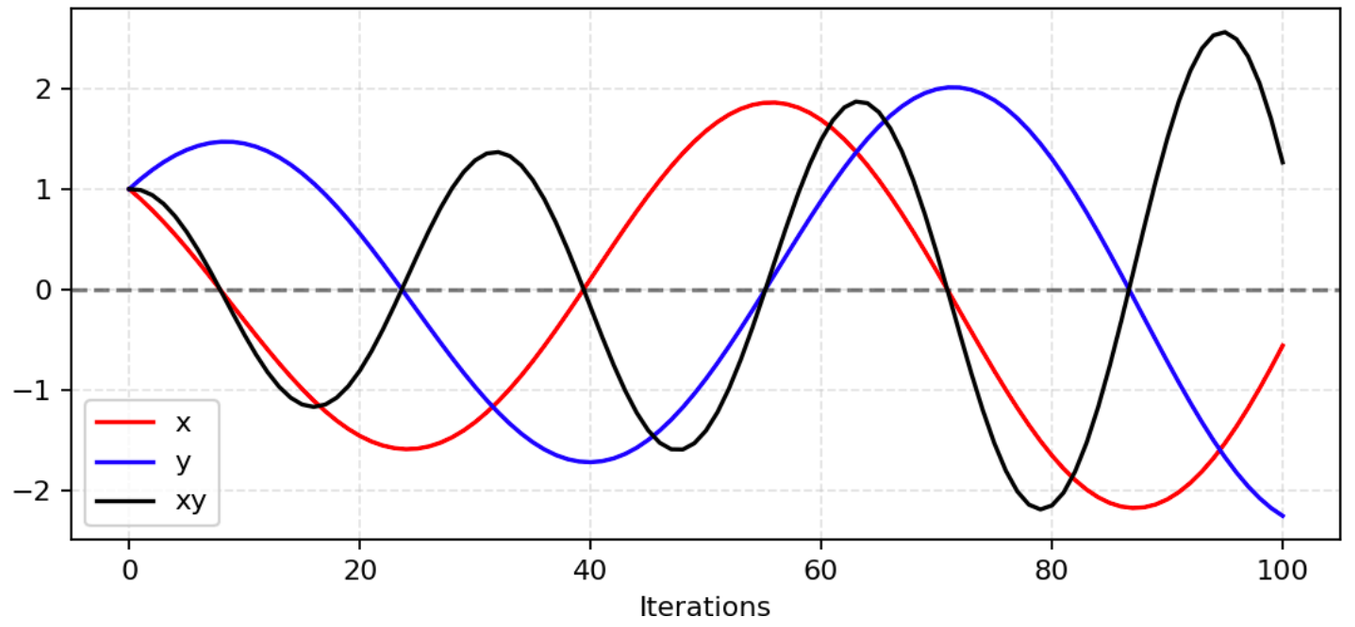
Generation Method	Hardware	Batch Size	milliseconds/shower
GEANT4	CPU	N/A	1772 ←
CALOGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012 ←

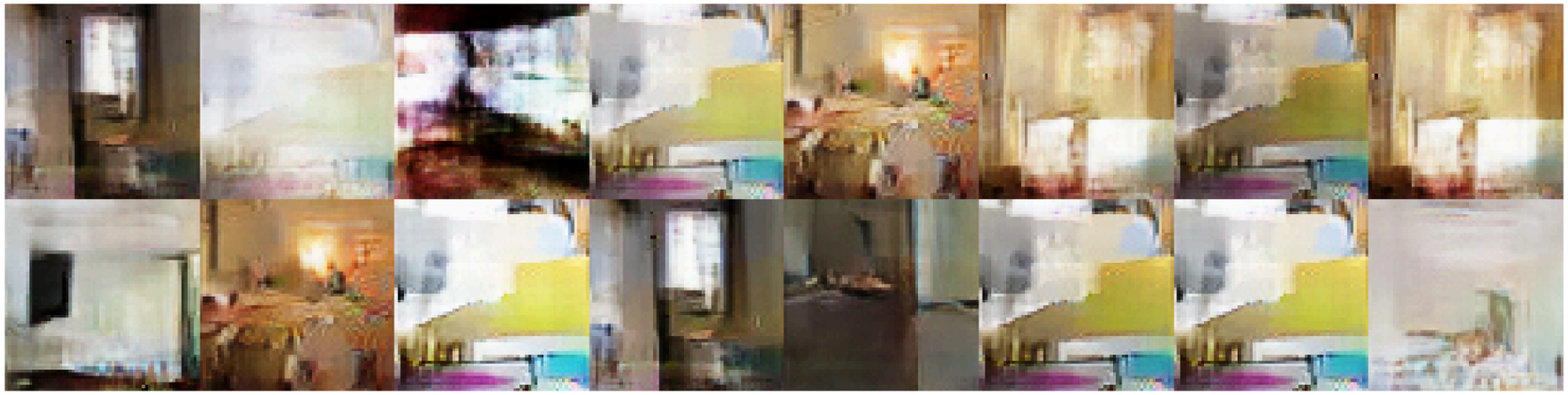


GAN Problems

- GANs optimises the Jensen-Shannon divergence
- Typical problems when training GANs:
  - Stability of learning
  - Mode collapse
  - Low dimensional support

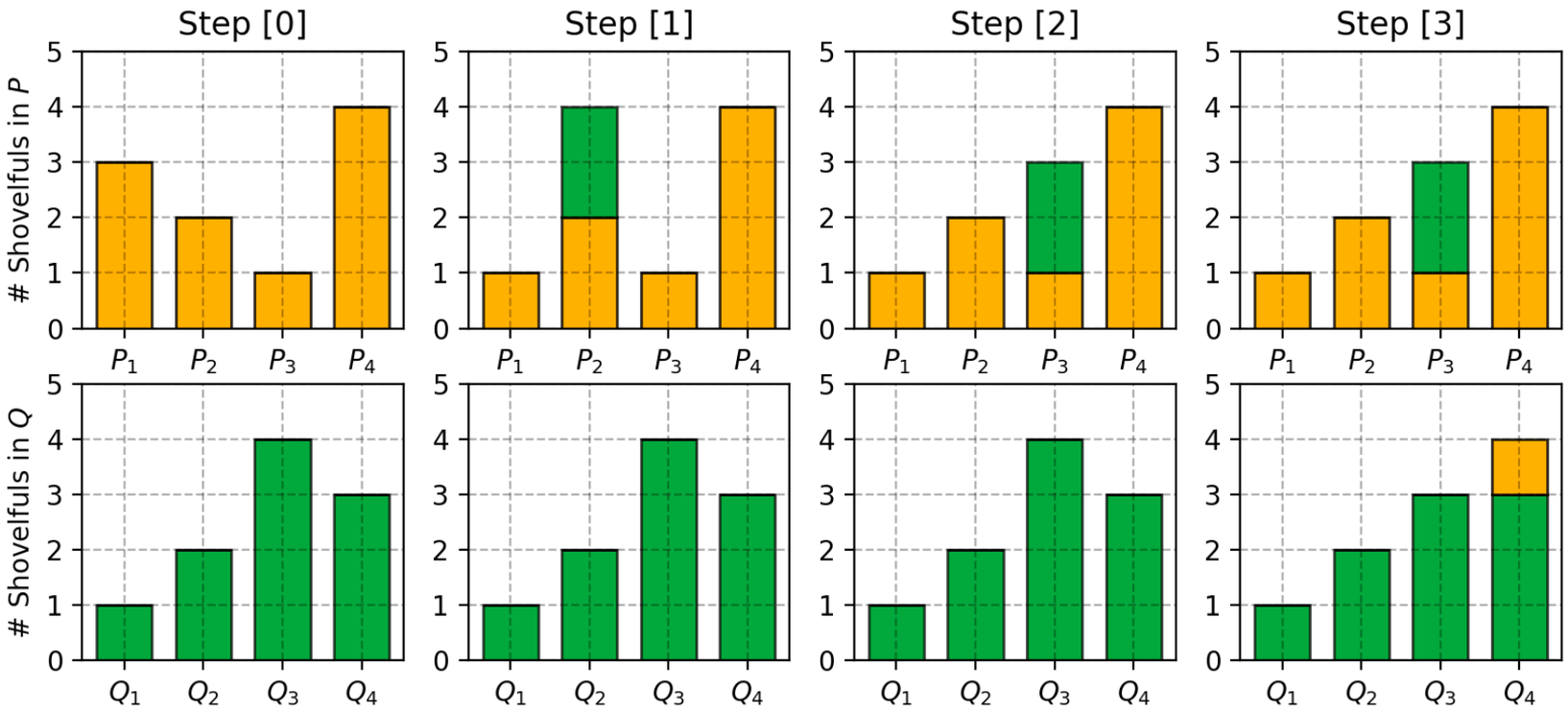
*Player 1: Change x to minimise x*y*
*Player 2: Change y to maximise x*y*

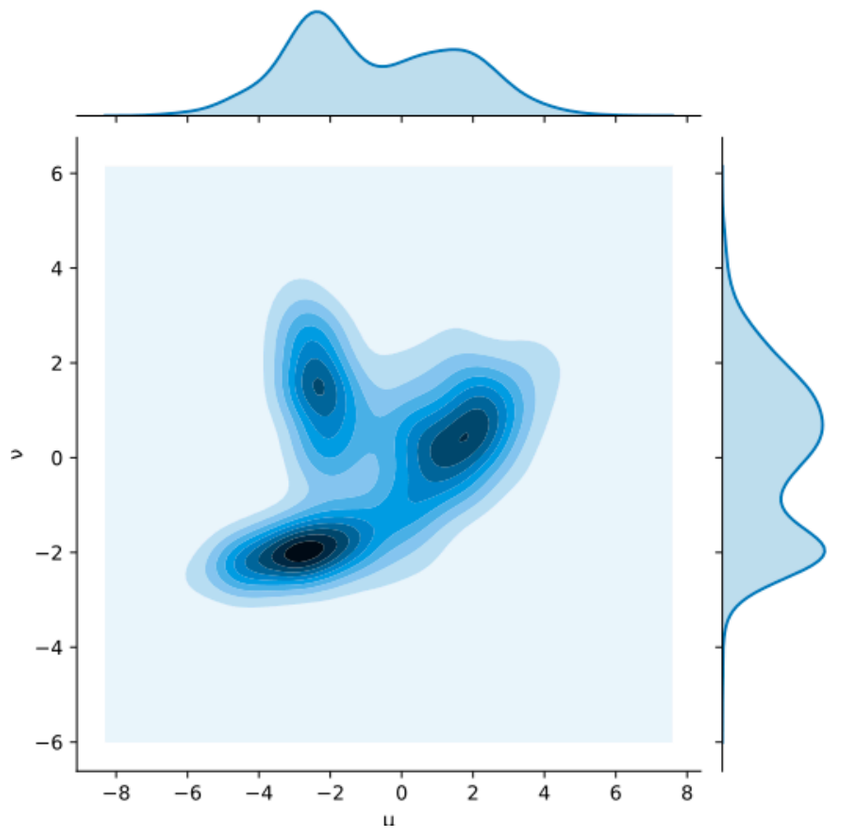




lilianweng
Wasserstein GAN, M Arjovsky, S Chintala, L Bottou, 1701.07875

Wasserstein Distance





- Earth mover's distance

- Metric compares two probability distributions

How much to move from x to y

$$W_p(\mu,\nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{M \times M} d(x,y)^p \, \mathrm{d}\gamma(x,y) \right)^{1/p}$$

Distance between x and y

lilianweng.github.io

Wasserstein GAN

$$D_W = \sup_{f \in \mathrm{Lip}_1} (\mathbb{E}[f(x)] - \mathbb{E}[f(\tilde{x})])$$

Equivalent formulation, according to Kantorovich-Rubinstein duality

$$|f(x_1) - f(x_2)| \leq L \cdot |x_1 - x_2|$$

Lipschitz continuity

$$C_1 = \mathbb{E}[f_w(x)] - \mathbb{E}[f_w(\tilde{x})]$$

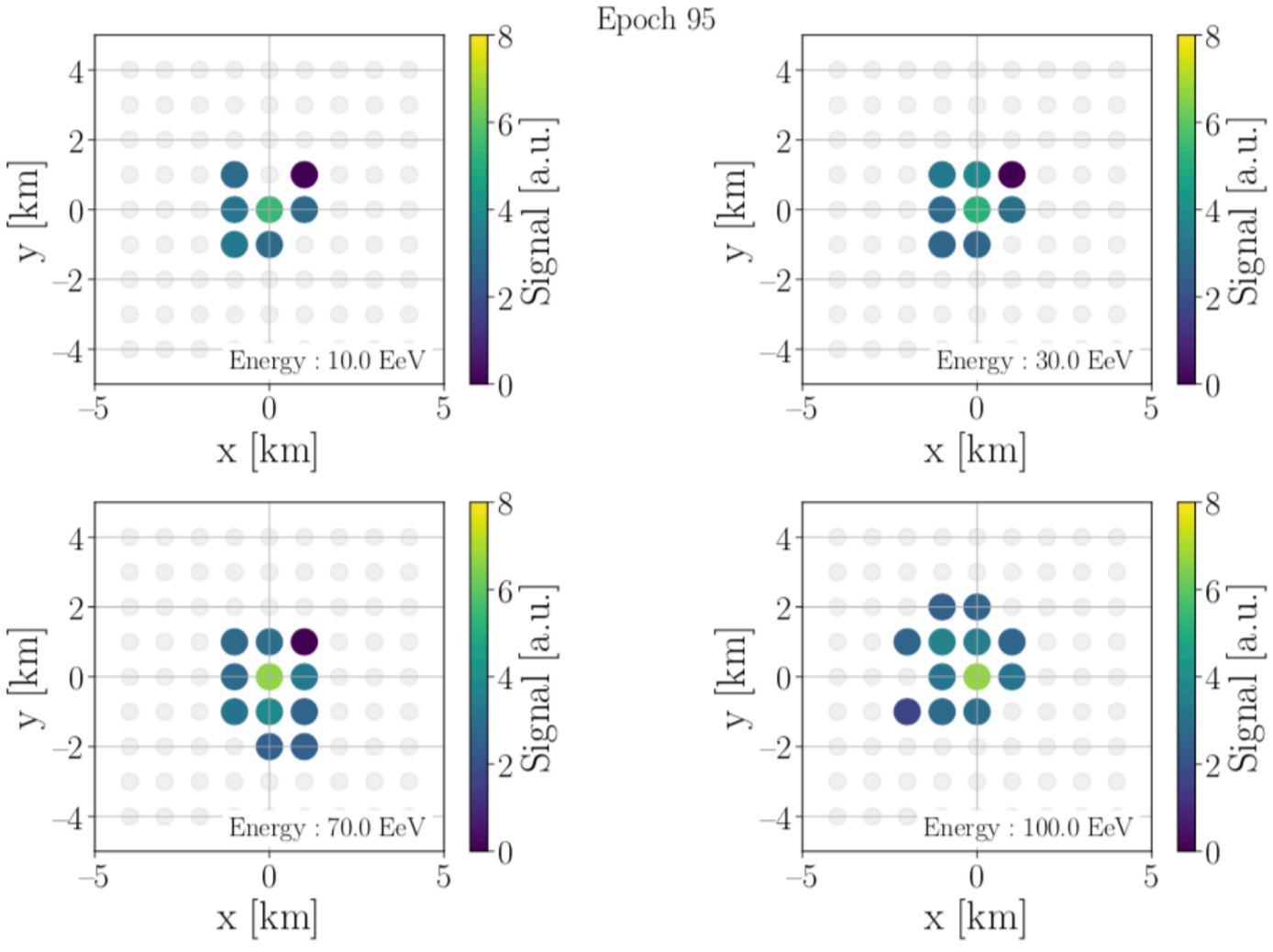
Encode using neural networks

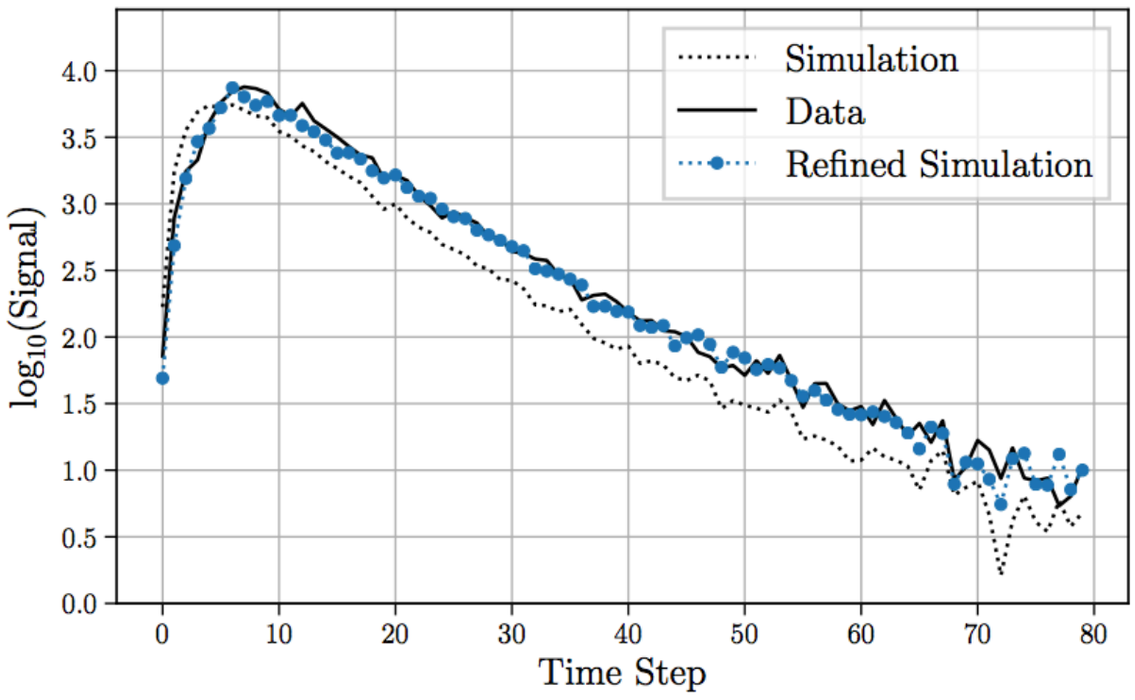
$$C_2 = \lambda \, \mathbb{E}[(||\nabla_{\hat{u}} f_w(\hat{u})||_2 - 1)^2]$$

Ensure continuity

Wasserstein GAN, M Arjovsky, S Chintala, L Bottou, 1701.07875
Improved Training of Wasserstein GANs I Gulrajani et al, 1704.00028
Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks M Erdmann et al, 1802.03325

WGAN

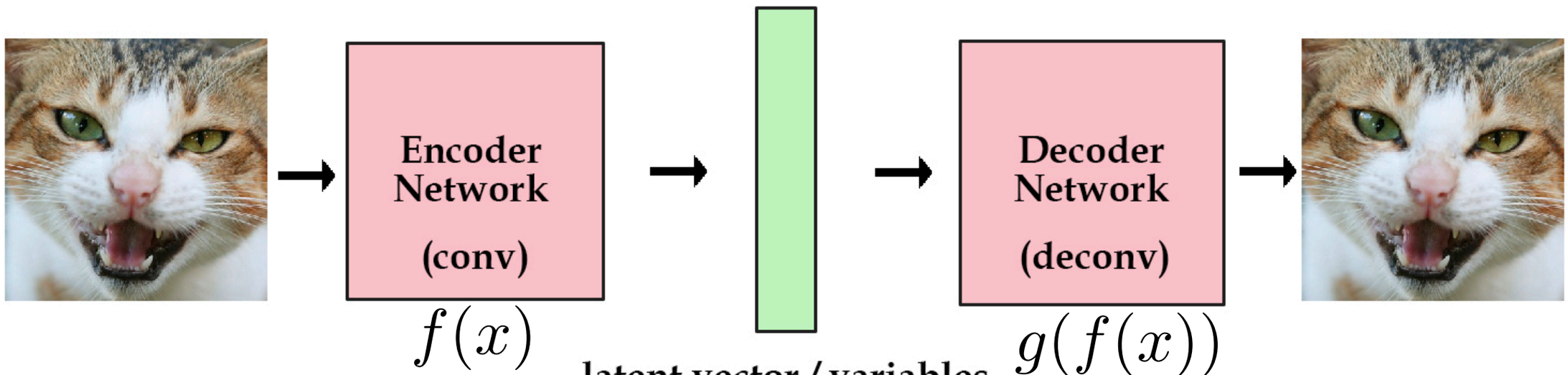




Simulation of cosmic air-showers captured by Cherenkov detector. Use to improve energy reconstruction!

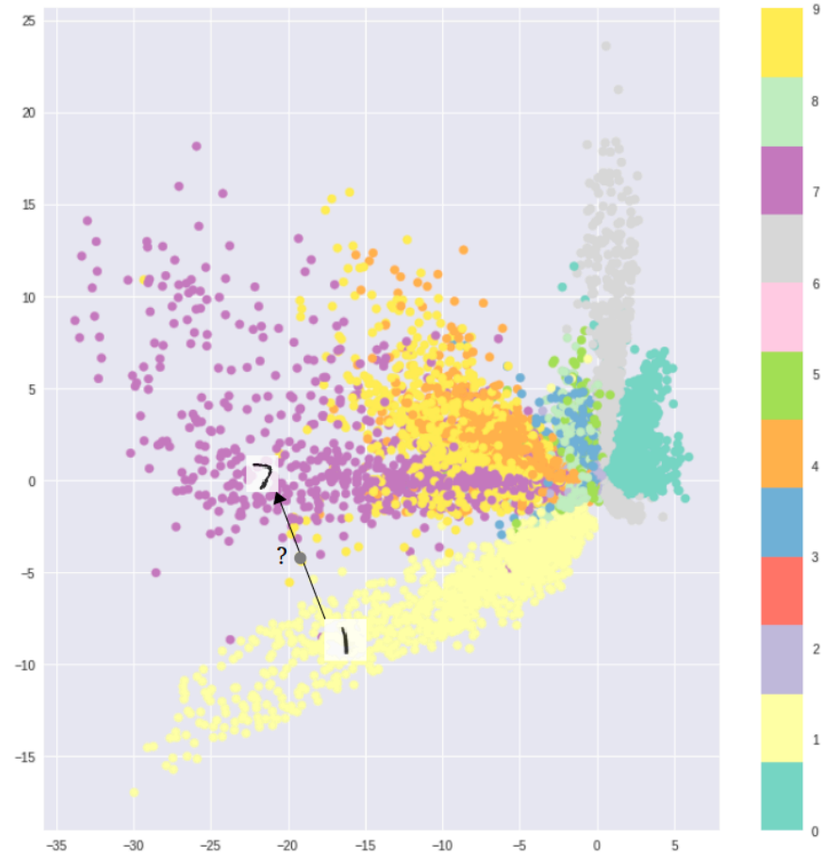
Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks
M Erdmann et al, 1802.03325

Autoencoder



$$L = (\hat{y} - g(f(x)))^2$$

- Self-supervised learning
- *Bottleneck* with compressed representation
- Dimension reduction
- Denoising
- Regularizers



Latent space

kvfrans
deeplearningbook.org