Deep Learning for Dark Showers

Theo Heimel, Gregor Kasieczka, Tilman Plehn, Jennifer Thompson
(gregor.kasieczka@uni-hamburg.de)

4th LLP Workshop, 2018-10-25

Results based on *QCD or What?* arXiv: 1808.08979

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

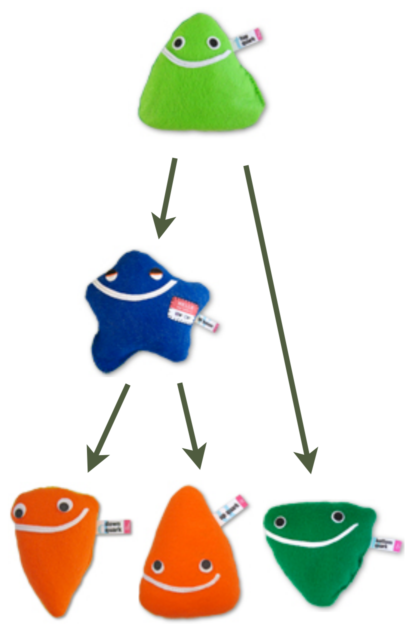
Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

Bundesministerium für Bildung und Forschung

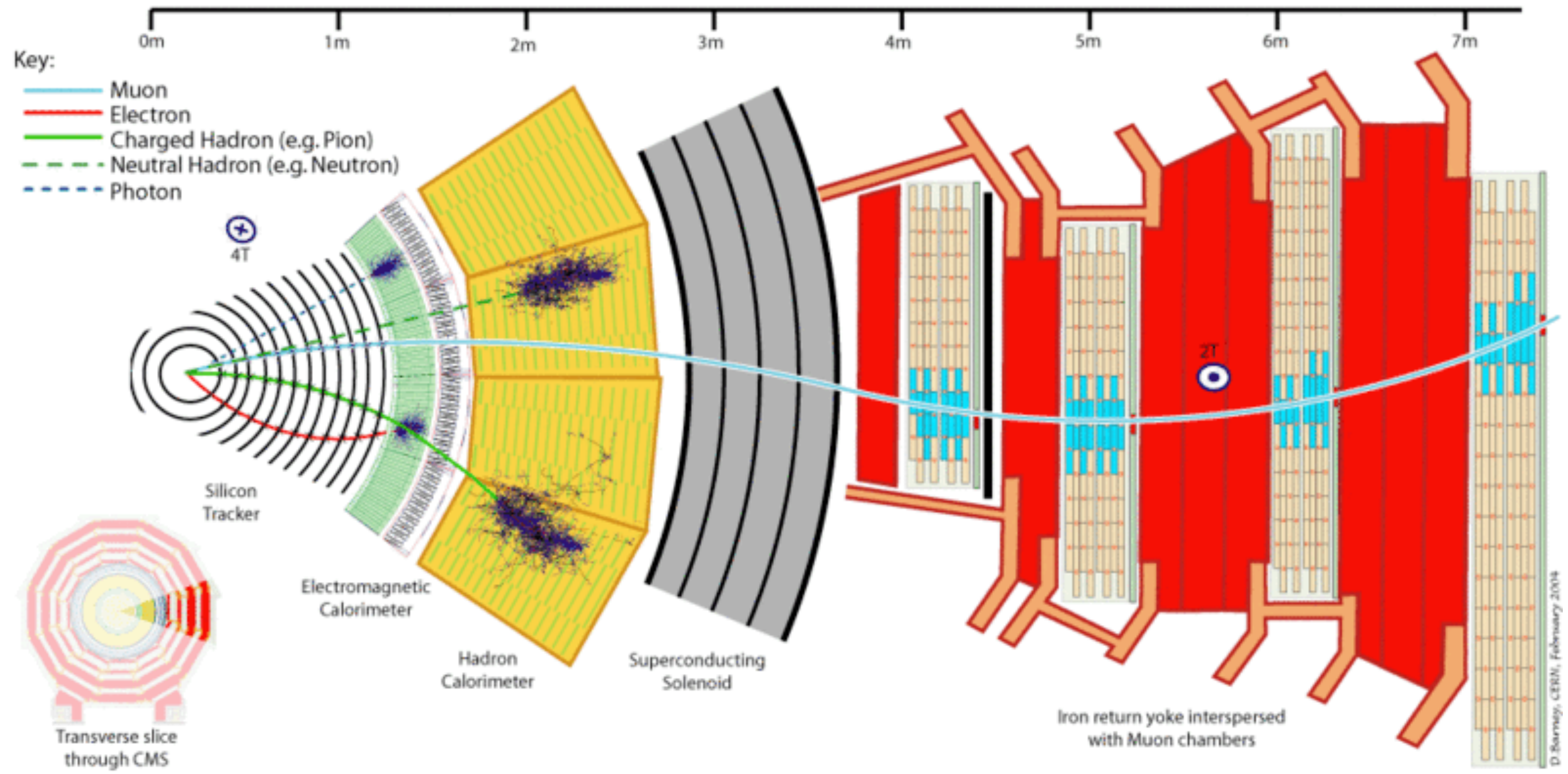
Overview

- *Jet Images & Convolutional Networks*
- *Autoencoders*
- *Adversarial Training*
- *Dark Showers*

Jet Images & Convolutional Networks



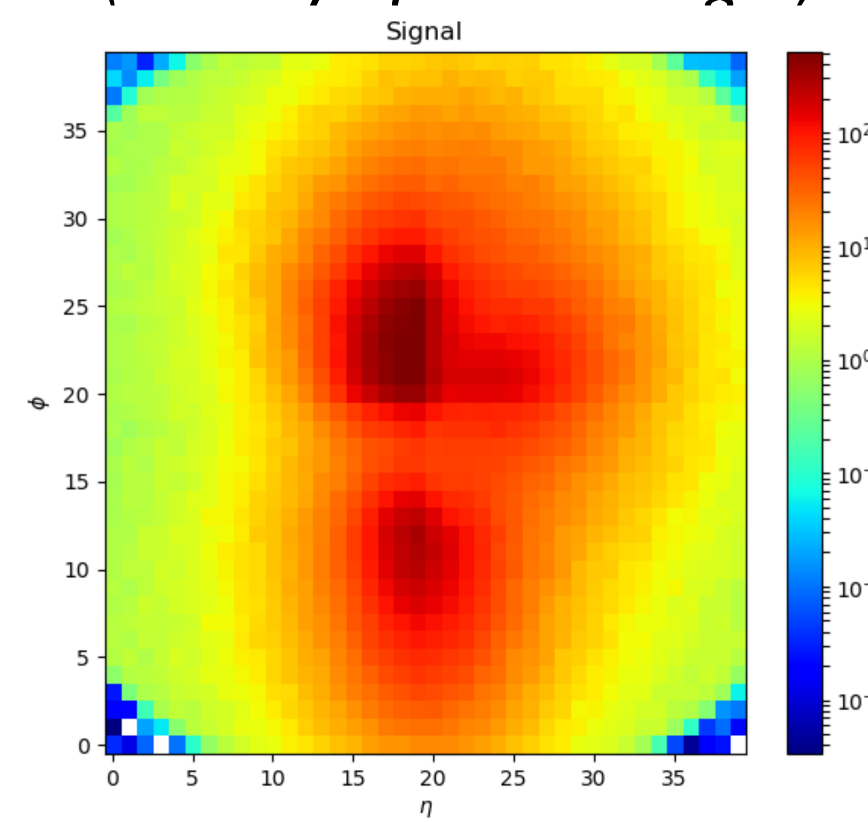
Top



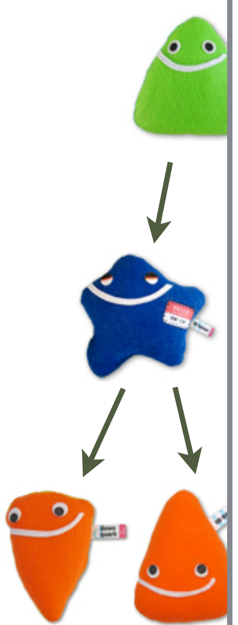
- Reconstruct energy with calorimeter (improve resolution using tracker)
- Cluster energy deposits into jet
- Preprocess:

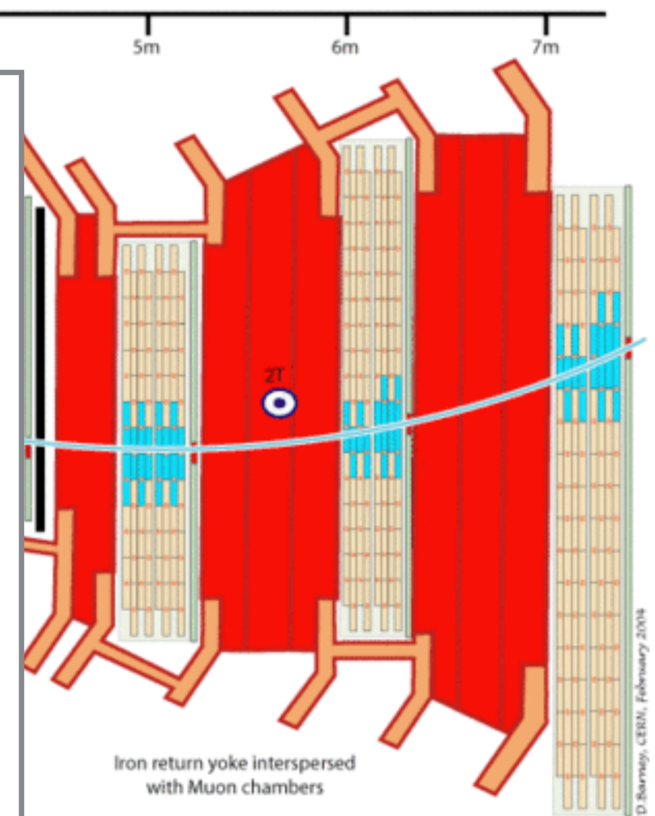
> - center → rotate → flip (twice) → pixelate → crop → normalise
>   - center: centroid is at (0/0)
>   - rotate: principal axis is vertical
>   - flip: in (x<0, y>0)-plane maximum intensity
>   - crop: to nxn images
>   - normalise: intensity of each pixel divided by total intensity

(Overlay of 100k images)

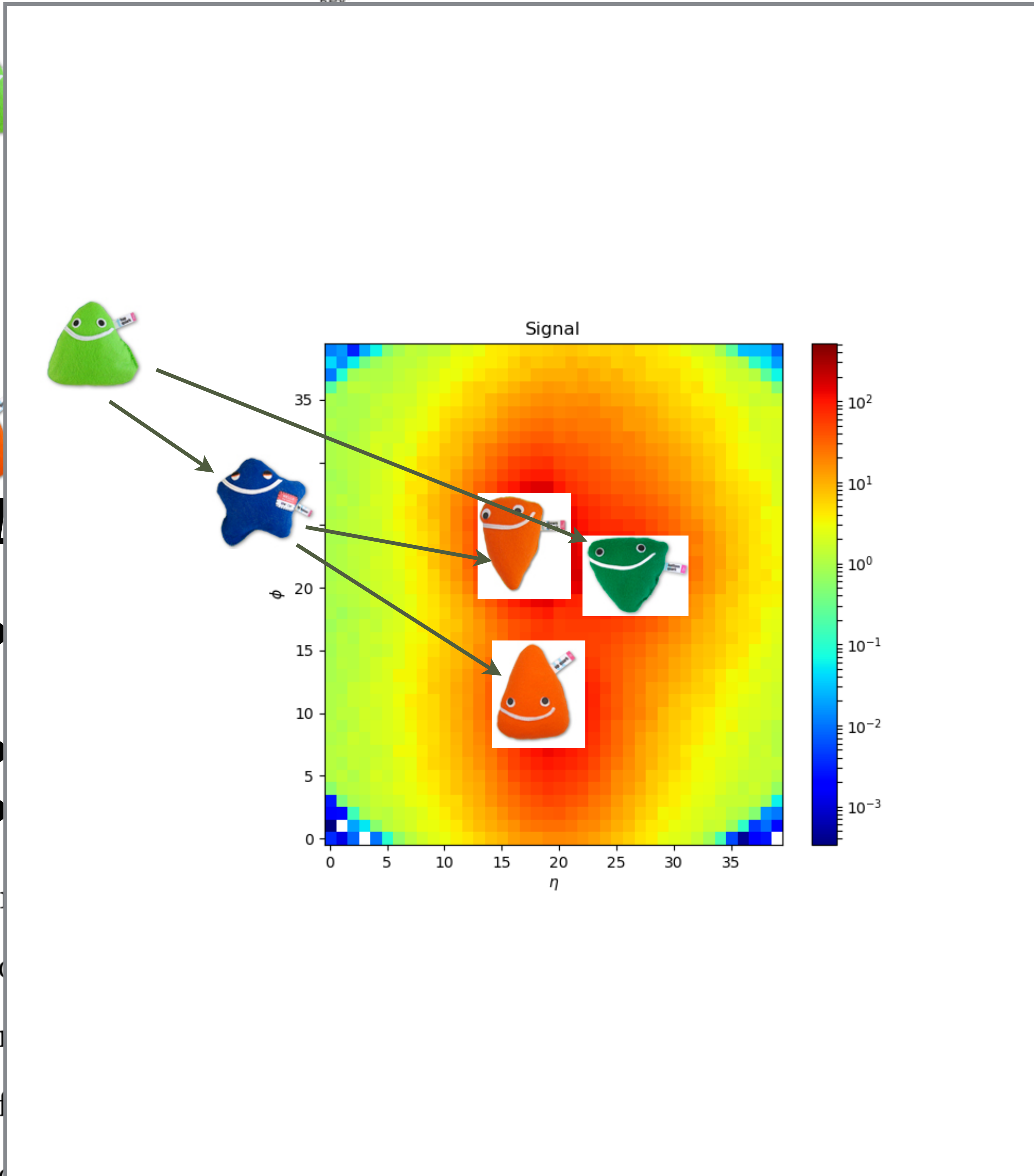


(jet images by Michel Luchmann)



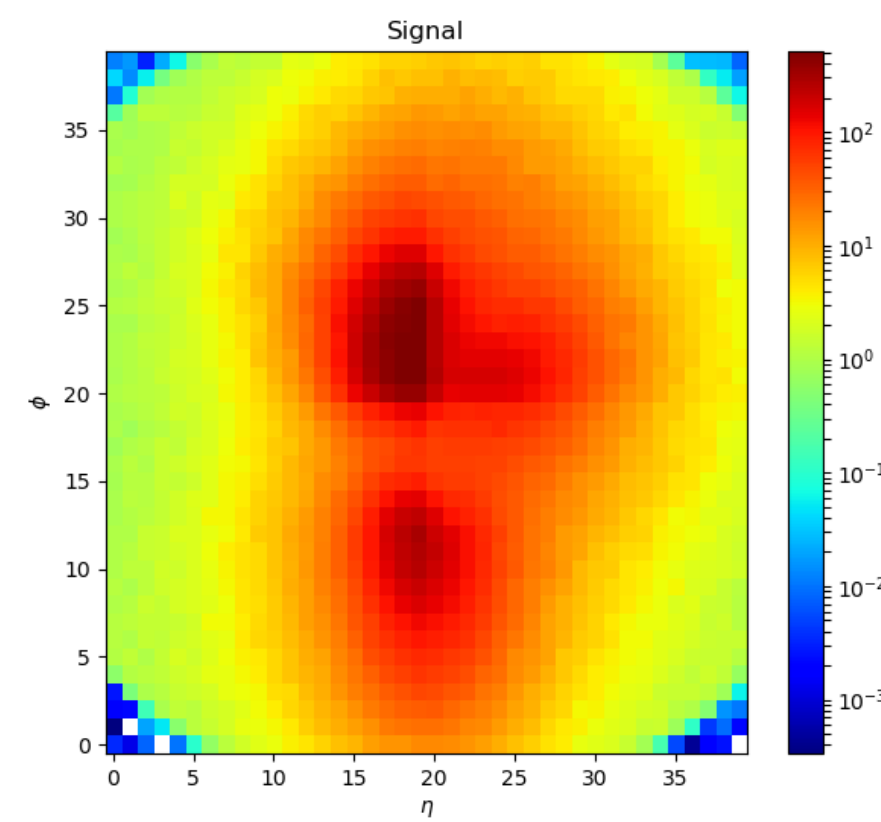


Top



- cente
  - normalise: intensity of each pixel divided by total intensity

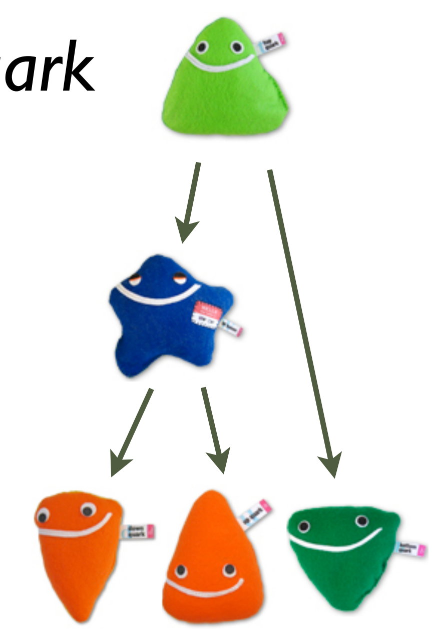
(Overlay of 100k images)

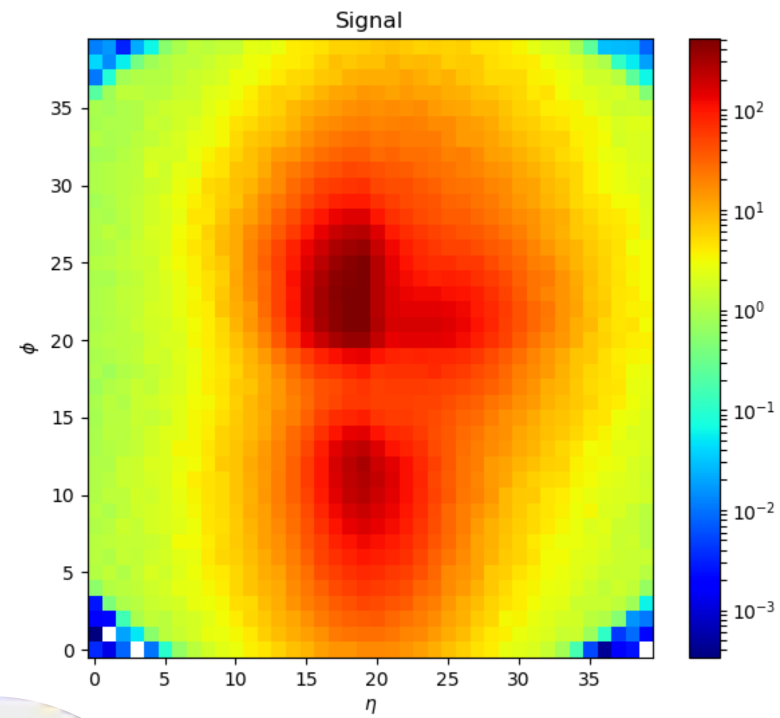


(jet images by Michel Luchmann)

Supervised Classification

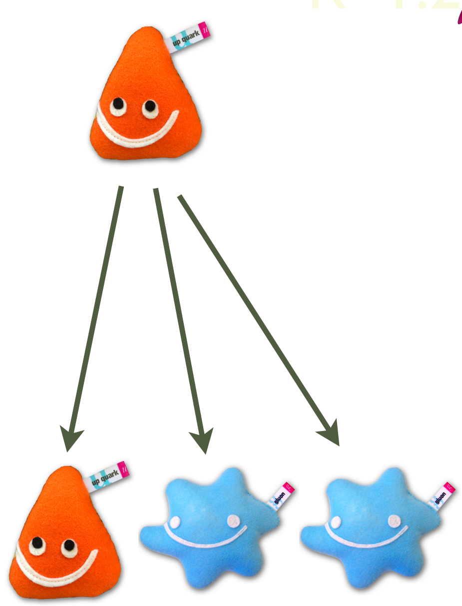
Top Quark Jet

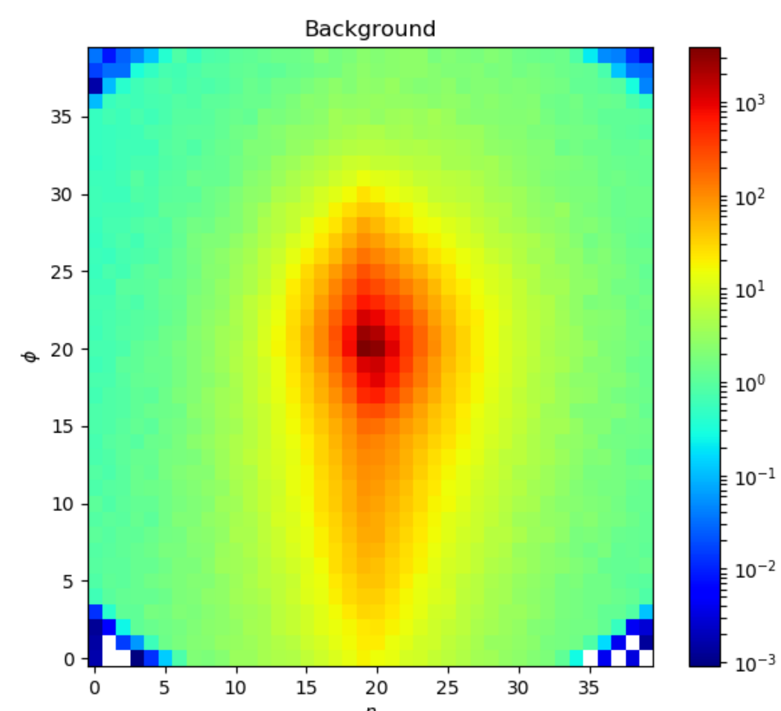




- *Binary classification task*
- *Fully supervised learning (using simulation)*
- *40x40 Pixels, E_T*

QCD Jet

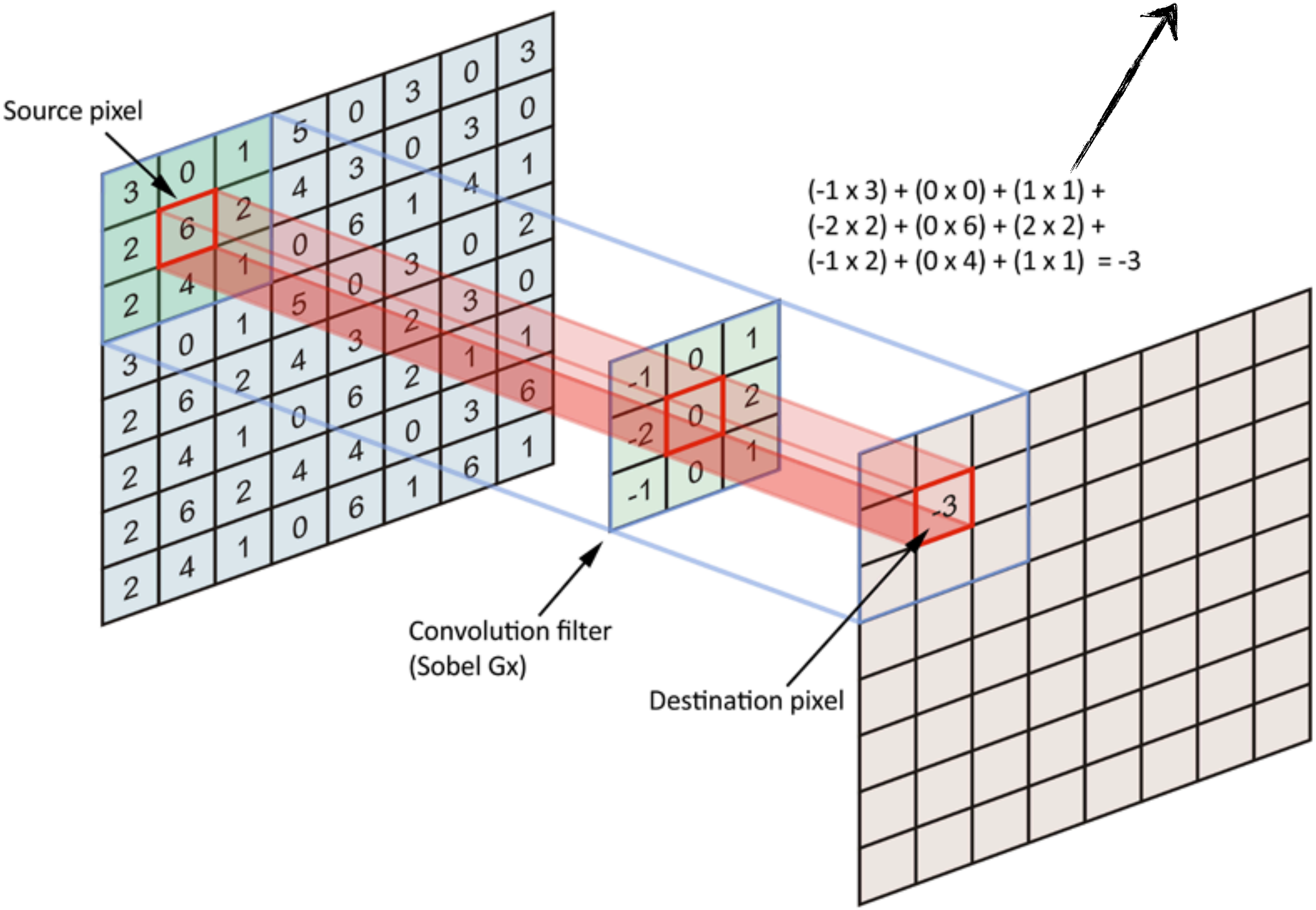




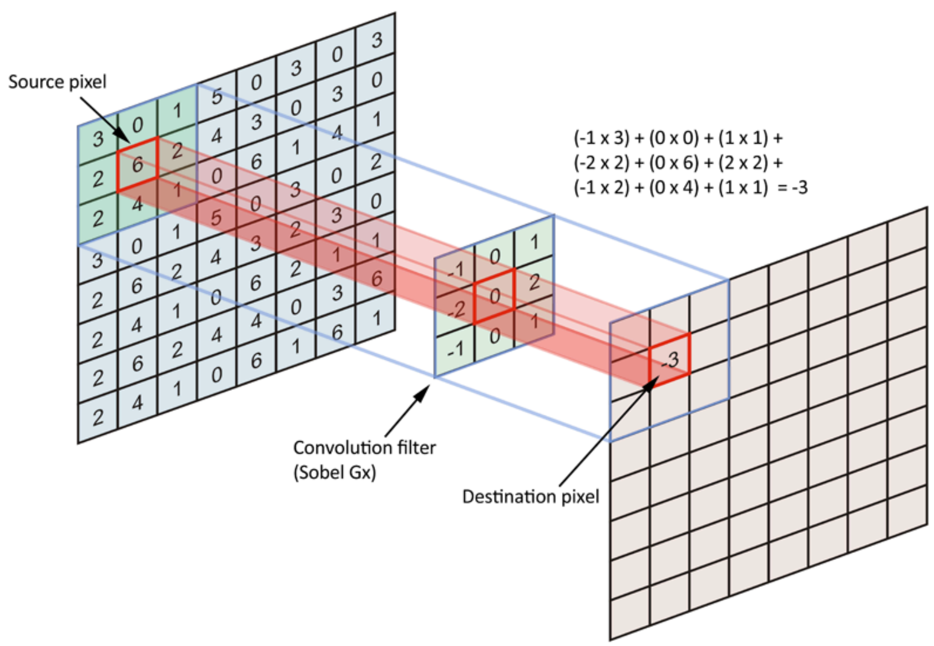
Deep-learning Top Taggers or The End of QCD?
GK, Tilman Plehn, Michael Russell, Torben Schell
JHEP 05 (2017) 006
Pulling Out All the Tops with Computer Vision and Deep Learning, S Macaluso, D Shih,
Origins:
Jet-Images: Computer Vision Inspired Techniques for Jet Tagging
J Cogan, M Kagan, E Strauss, A Schwartzman
arXiv:1407.5675
Jet-Images – Deep Learning Edition
Ld Oliveira, M Kagan, L Mackey, B Nachman, A Schwartzman
JHEP 1607 069

Convolutional Layer

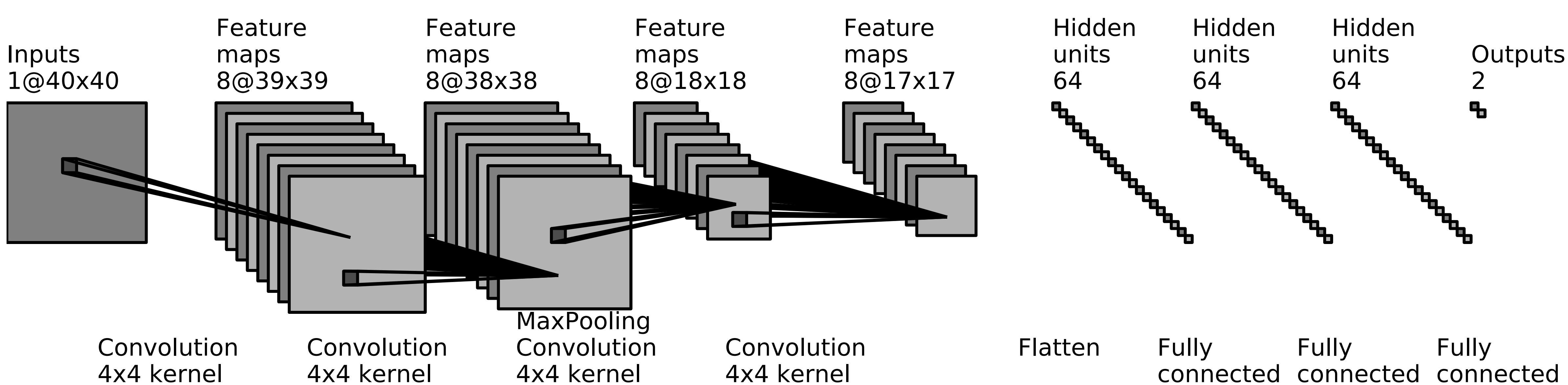
That's the weights we want to train



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



Performance

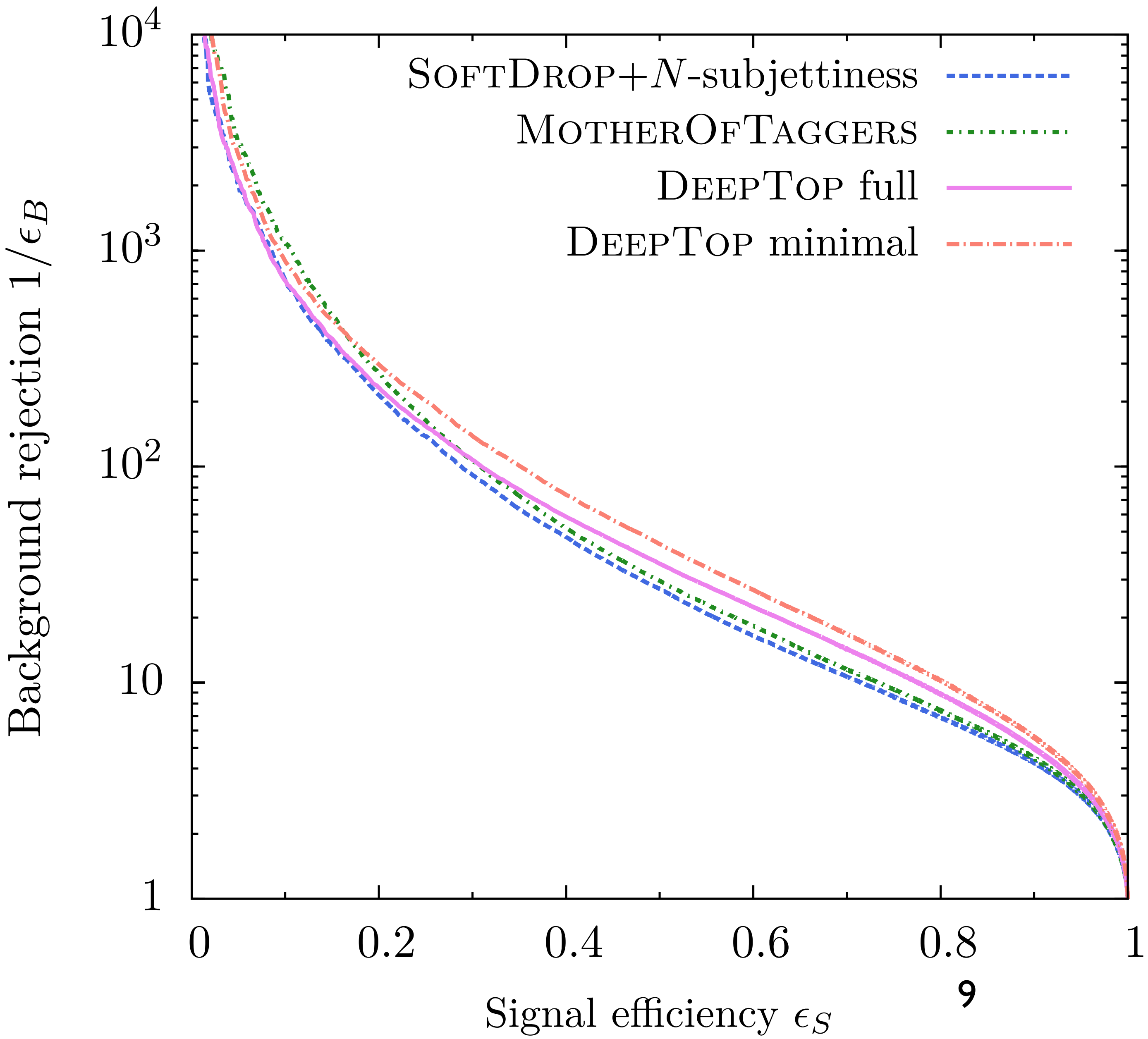
- Train a BDT on a set of standard tagging variables

SoftDrop + n-subjettiness:

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

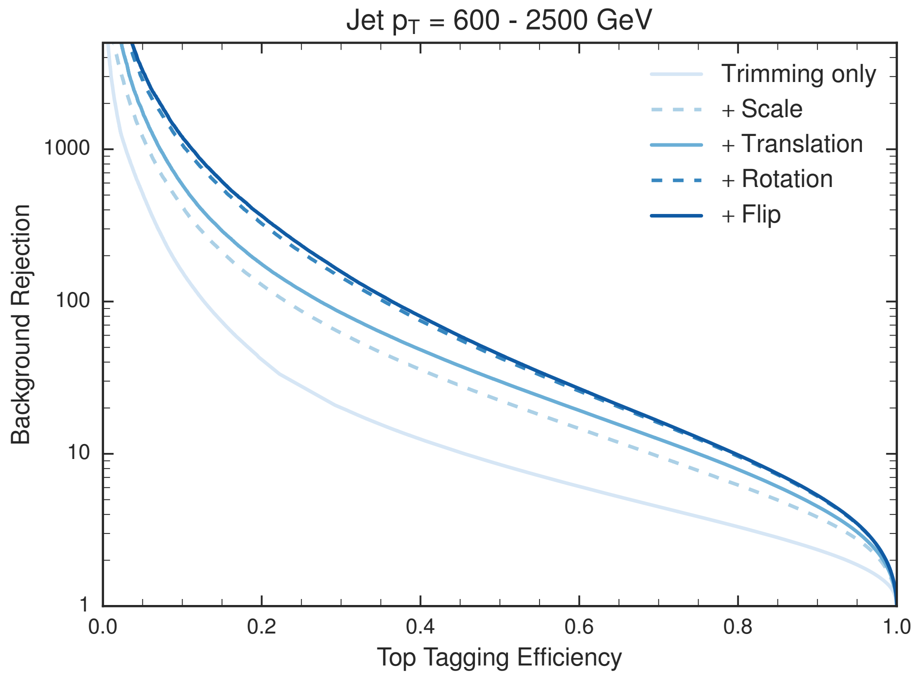
MotherOfTaggers:

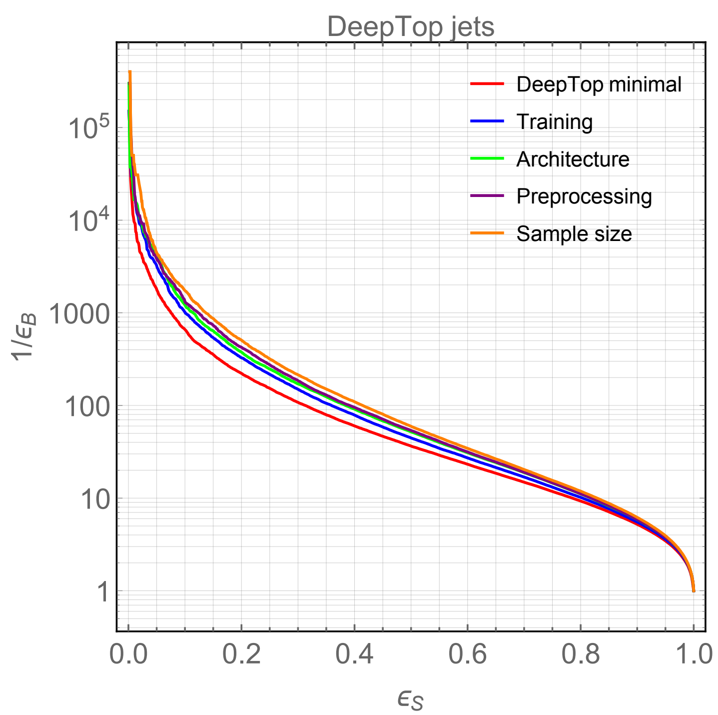
$$\{\ m_{\rm sd}, m_{\rm fat}, m_{\rm rec}, f_{\rm rec}, \Delta R_{\rm opt}, \tau_2, \tau_3, \tau_2^{\rm sd}, \tau_3^{\rm sd}\ \}$$

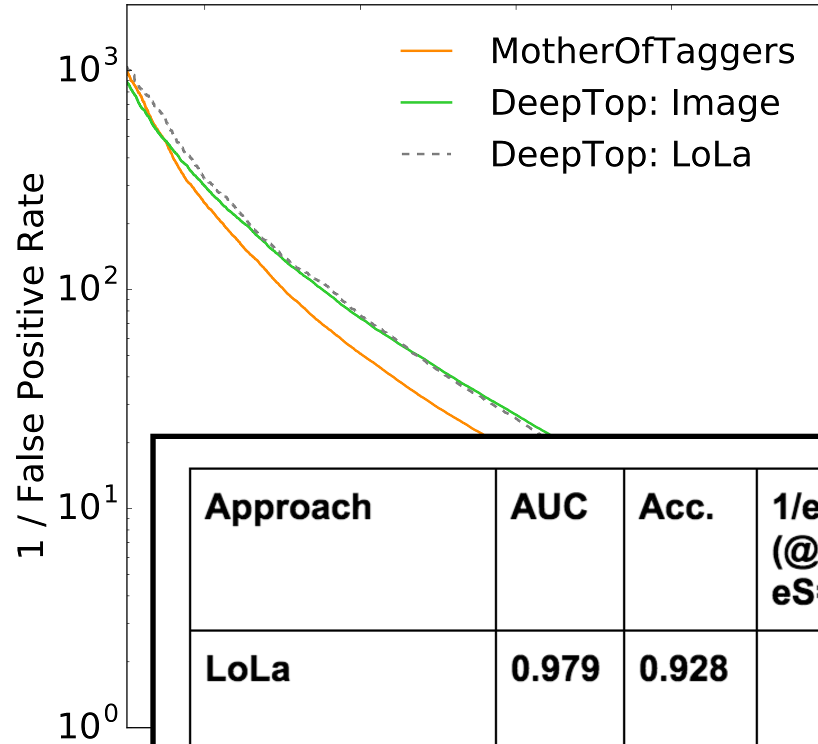


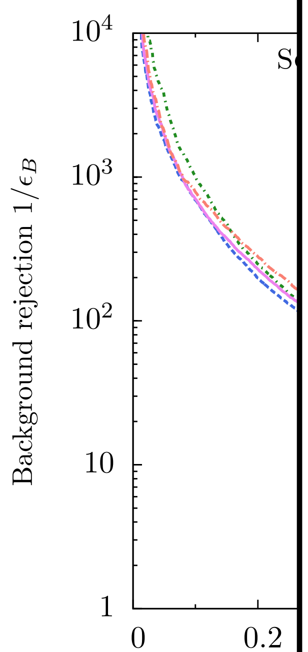
- Advantages:
  - Symmetry / structure
  - Straightforward
- Potential Problems
  - Resolution
  - Sparsity
  - How to encode complex information

Performance Overview





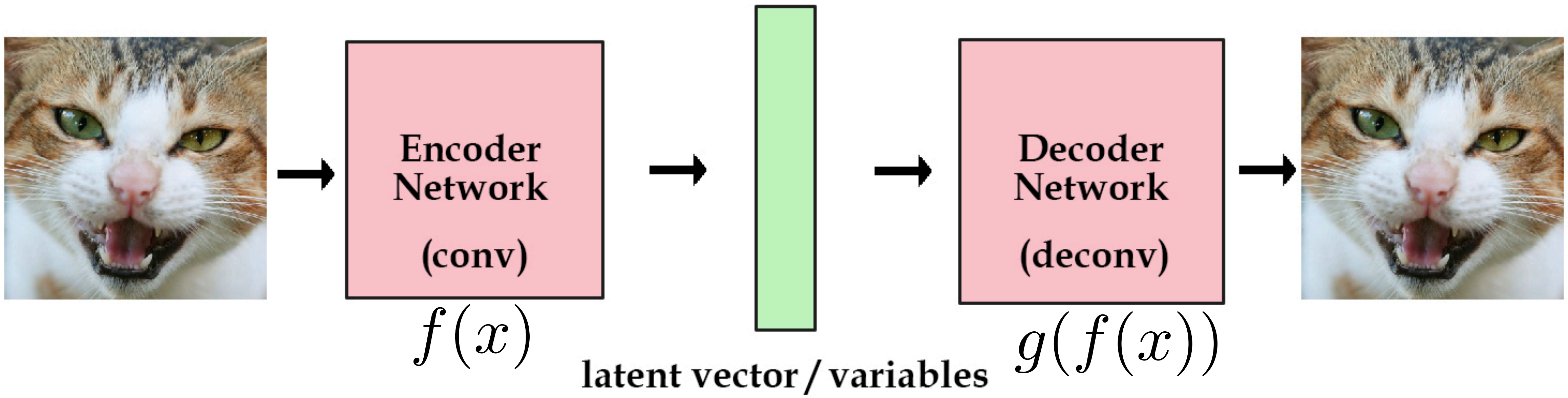




Approach	AUC	Acc.	1/eB (@ eS=0.3)	Contact	Comments
LoLa	**0.979**	**0.928**		**GK / SImon Leiss**	**Preliminary number, based on LoLa**
LBN	**0.981**	**0.931**	**863**	**Marcel Rieger**	**Preliminary number**
CNN	**0.981**	**0.93**	**780**	**David Shih**	**Model from *Pulling Out All the Tops with Computer Vision and Deep Learning (1803.00107)***
P-CNN (1D CNN)	**0.980**	**0.930**	**782**	**Huilin Qu, Loukas Gouskos**	**Preliminary, use kinematic info only (https://indico.physics.lbl.gov/indico/event/546/contributions/1270/)**
6-body N-subjettiness (+mass and pT) NN	**0.979**	**0.922**	**856**	**Karl Nordstrom**	**Based on 1807.04769 (*Reports of My Demise Are Greatly Exaggerated: N-subjettiness Taggers Take On Jet Images*)**
8-body N-subjettiness (+mass and pT) NN	**0.980**	**0.928**	**795**	**Karl Nordstrom**	**Based on 1807.04769 (*Reports of My Demise Are Greatly Exaggerated: N-subjettiness Taggers Take On Jet Images*)**

Autoencoders

Autoencoder



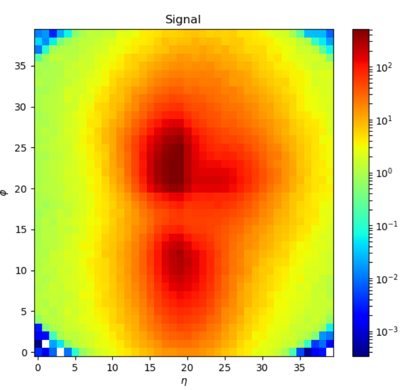
$$L = (\hat{y} - g(f(x)))^2$$

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

kvfrans
deeplearningbook.org

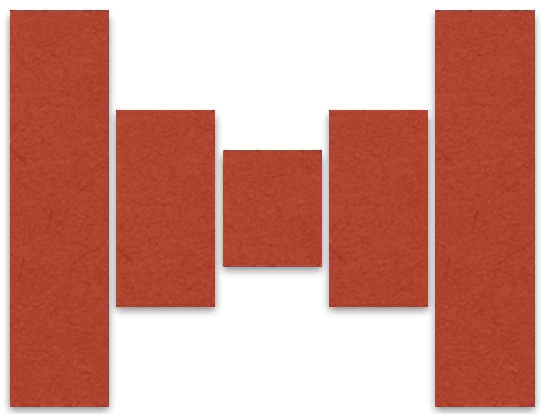
Autoencoder for Physics

Input

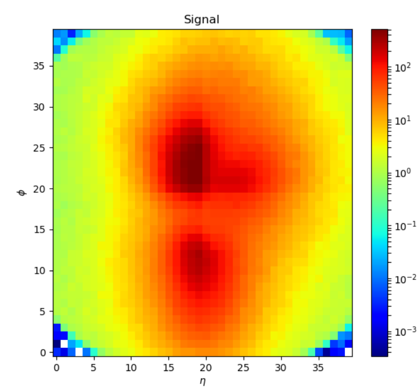


X_{ij}

Autoencoder



Output

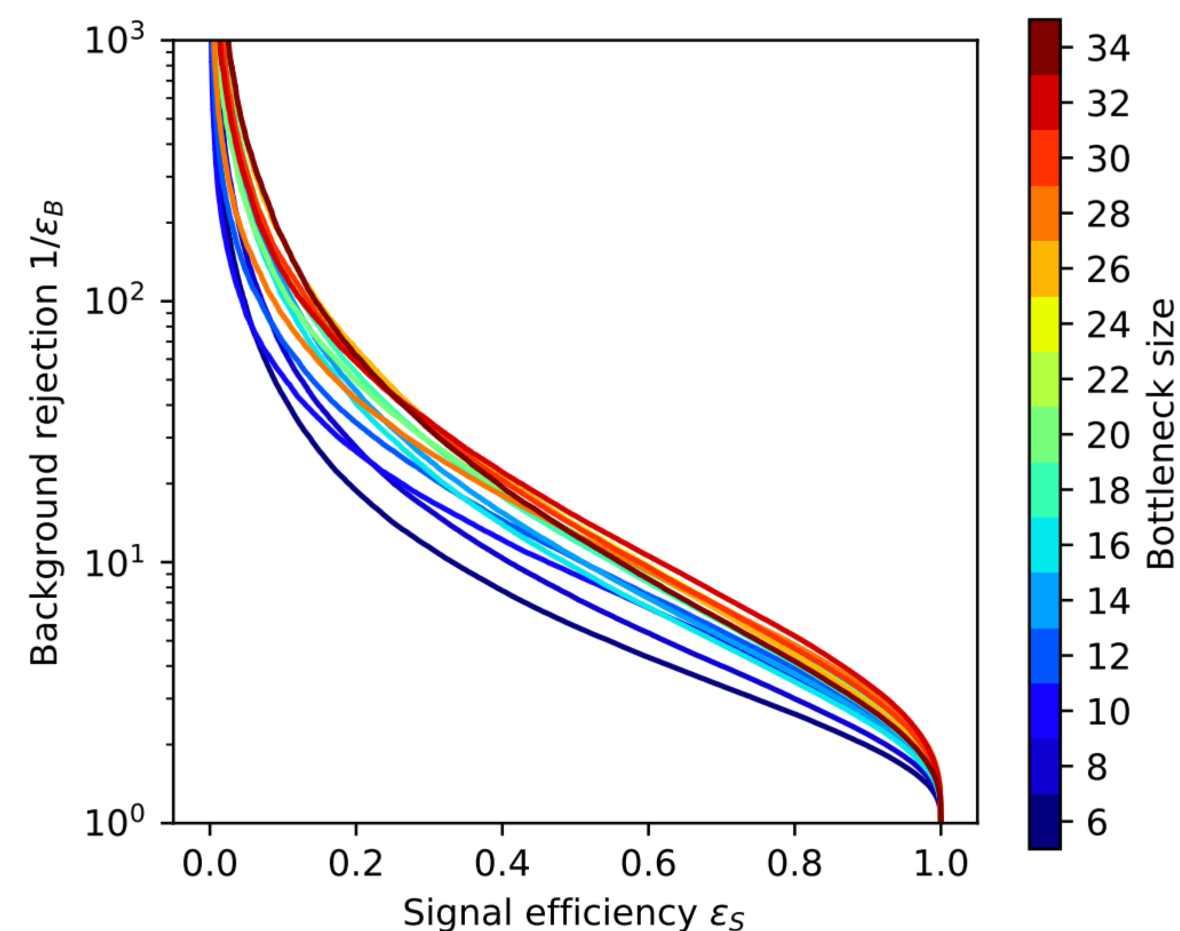


$\widetilde{X}_{ij}$

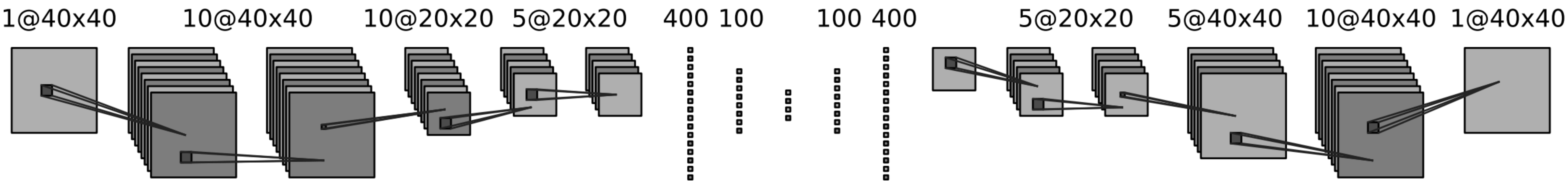
$$L_{\text{Auto}} = \sum_{\text{Pixels } ij} \left(X_{ij} - \widetilde{X}_{ij} \right)$$

- Can we find new physics without knowing what to look for?
- Train on **pure** QCD light quark/gluon jets and apply to top tagging
- Top quarks identified as anomaly

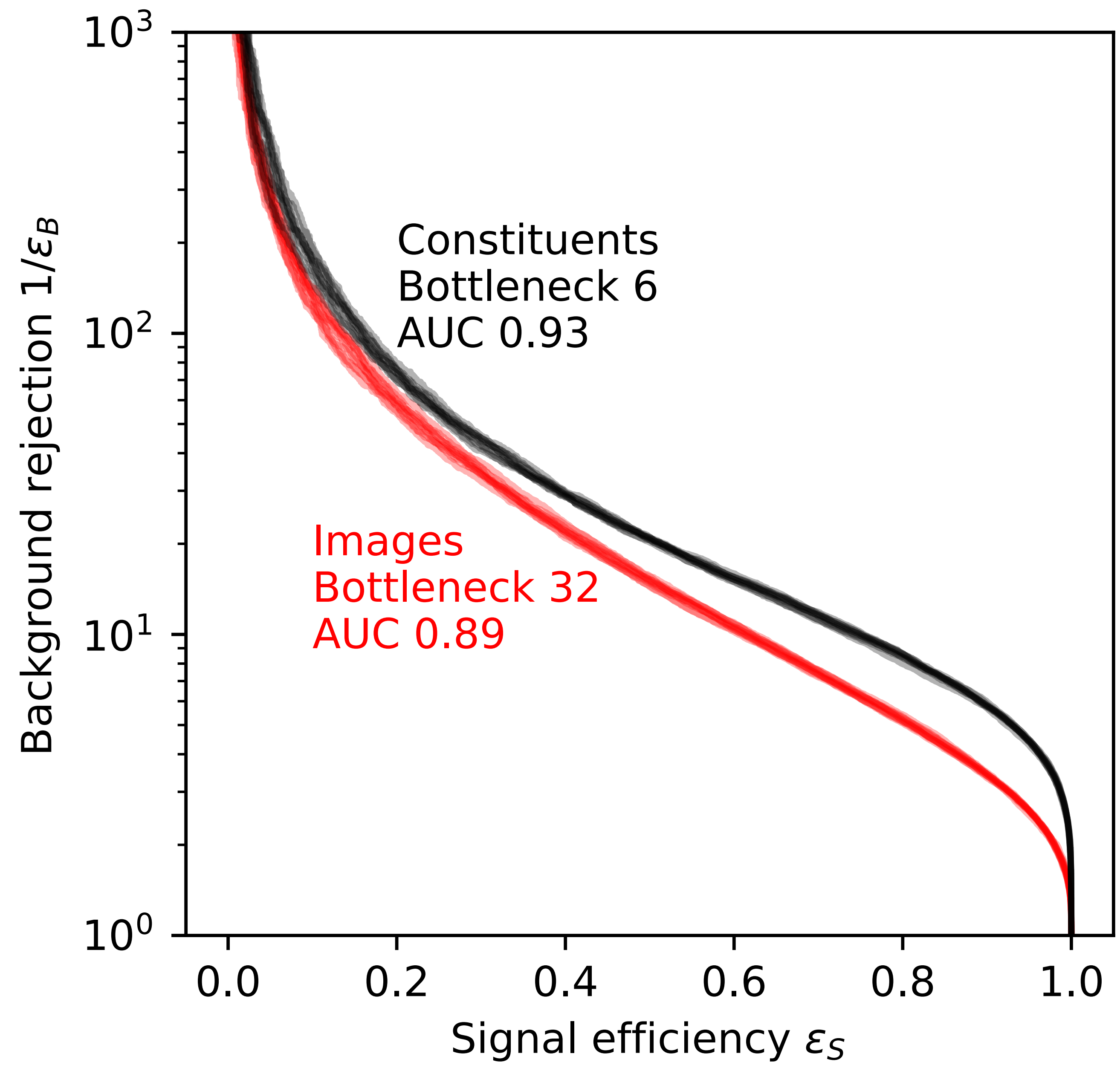
QCD or What?
T Heimel, GK, T Plehn, JM Thompson, 1808.08979
Searching for New Physics with Deep Autoencoders
M Farina, Y Nakai, D Shih, 1808.08992



Architecture



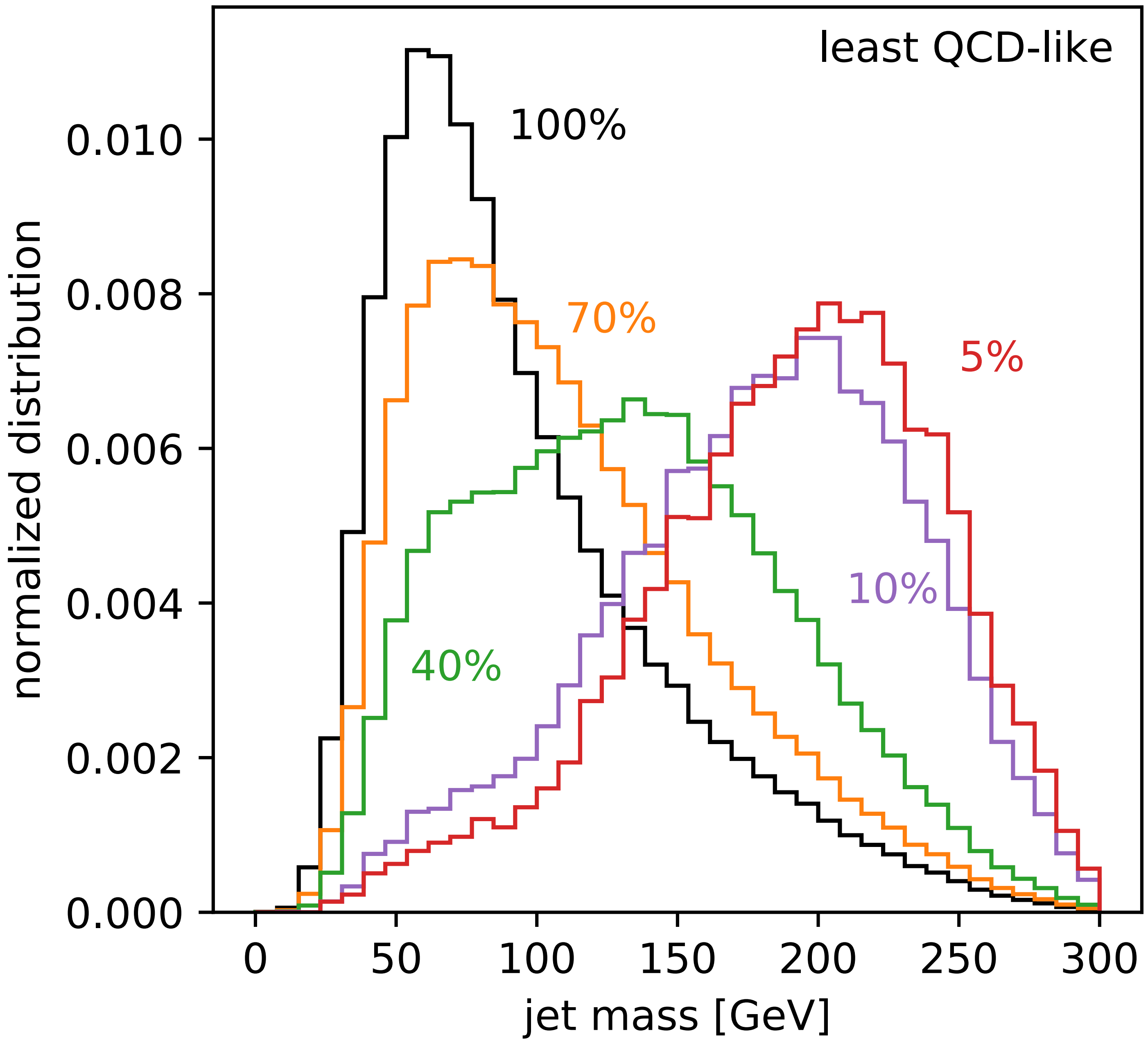
Convolutional network



Autoencoder will also work with other network architectures. Tested physics inspired, constituent based LoLa architecture.*

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

What about mass?

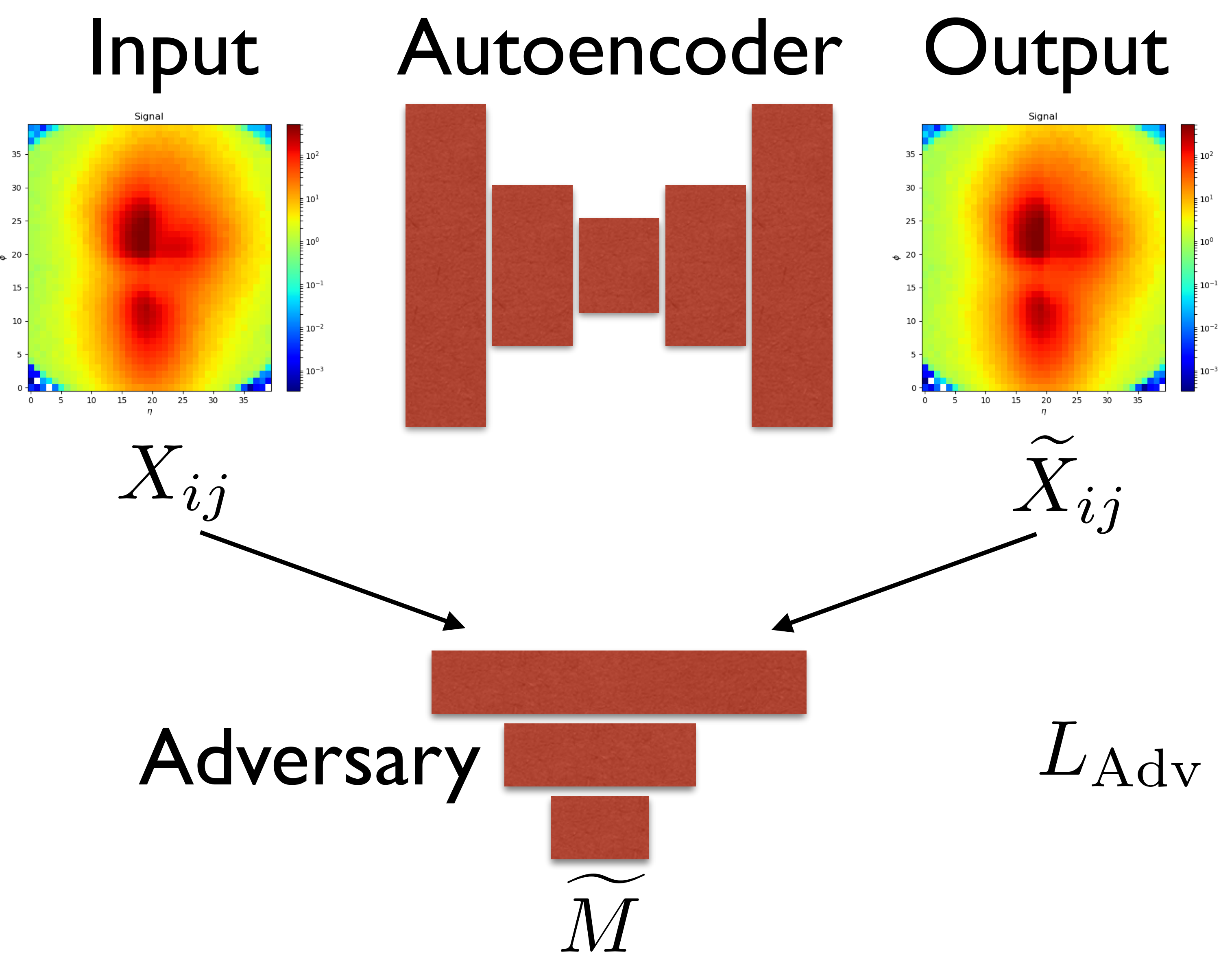


- Without additional constraints the autoencoder also learns the kinematics of the training sample

- How to avoid?

Adversarial Training

Combined Setup



$$L_{\text{Auto}} = \sum_{\text{Pixels } ij} \left(X_{ij} - \widetilde{X}_{ij} \right)$$

$$L_{\text{Adv}} = \text{CCE}\left(M, \widetilde{M}(X_{ij} - \widetilde{X}_{ij}) \right)$$

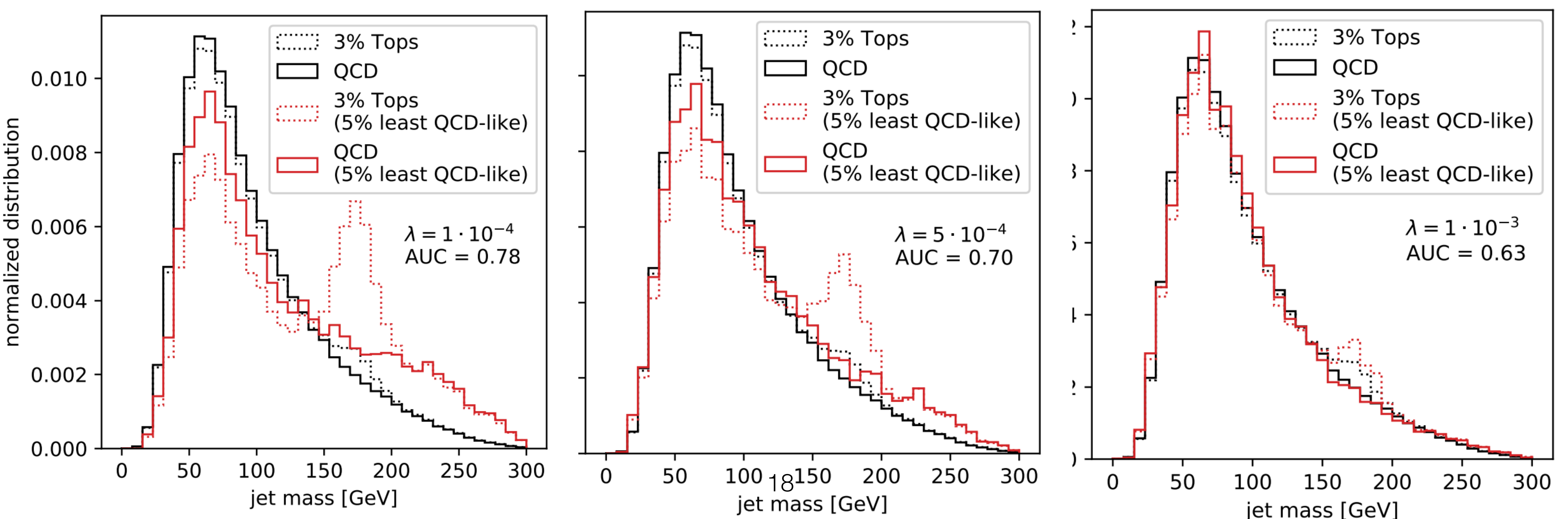
$$L = L_{\text{Auto}} - \lambda L_{\text{Adv}}$$

Mass Sculpting

- Counteract with adversary:

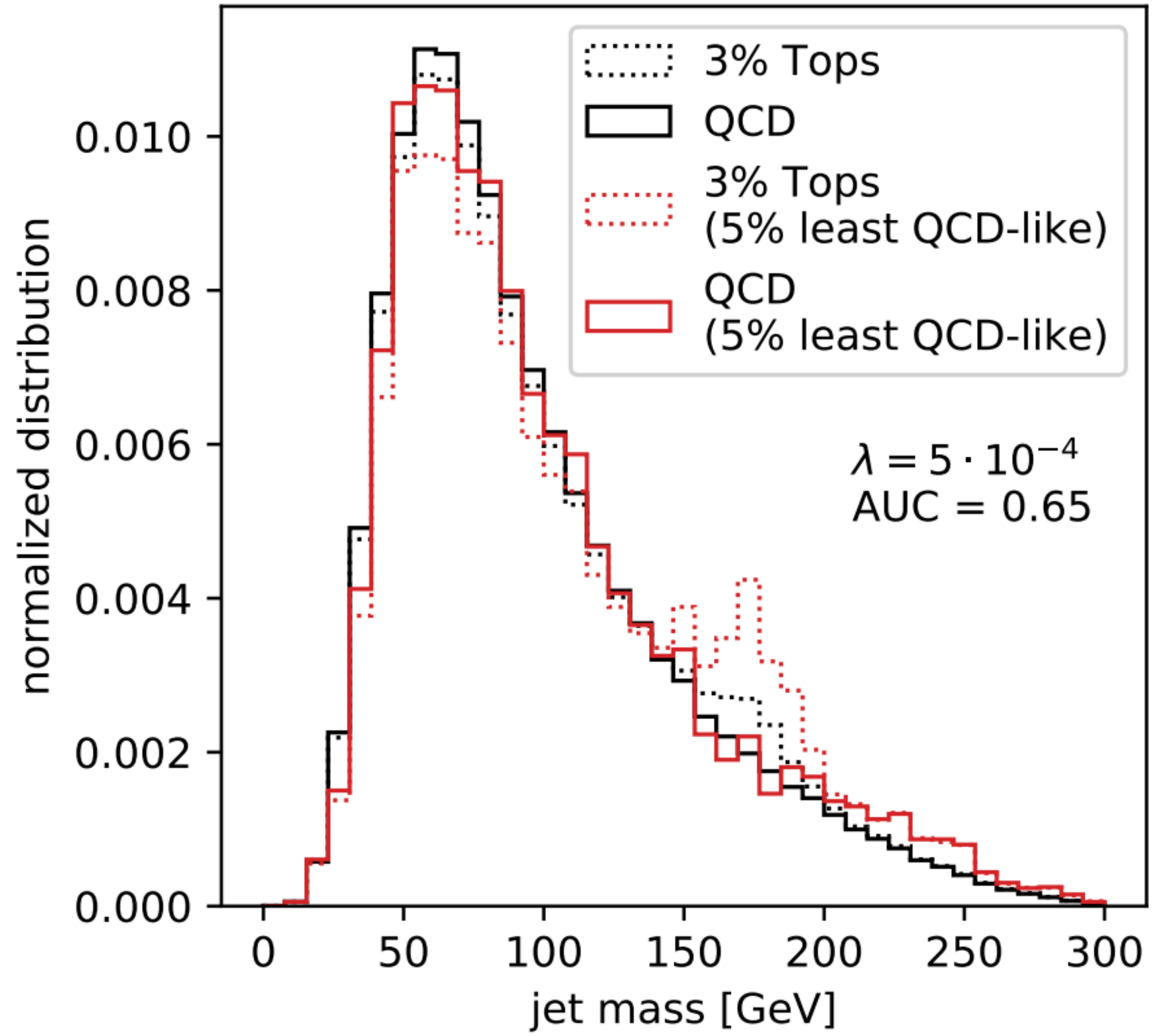
$$L = L_{\mathrm{Auto}} - \lambda L_{\mathrm{Adv}}$$

- Tune mass dependency with Lagrange multiplier



Signal contamination

- Procedure works also when signal is present in training data
- This means a search for exotic new physics with unknown shower patterns (dark showers) could be done using data-only training



Dark Showers

Recap

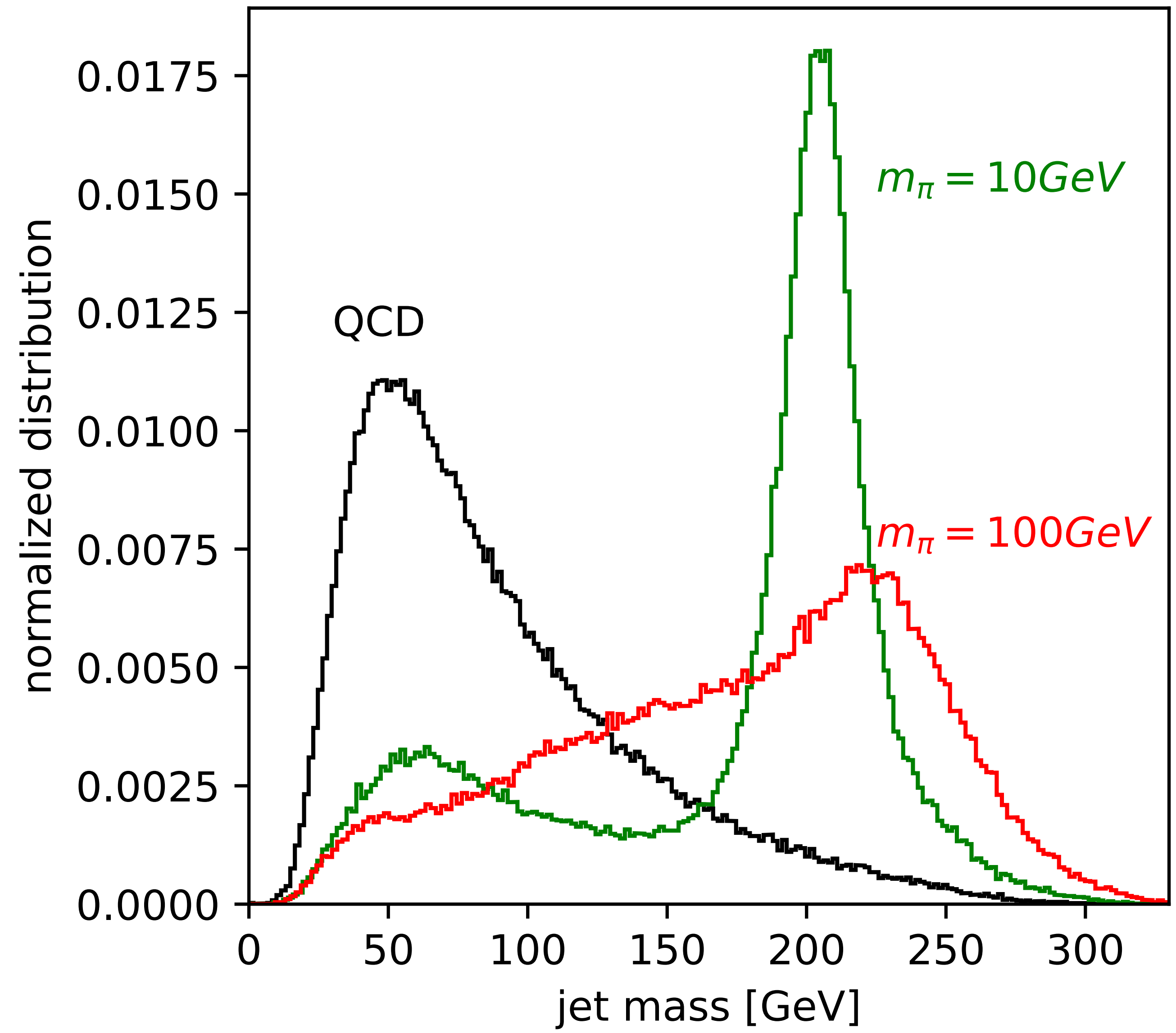
- We now have a tool that can identify anomalous jets..
  - ..purely **trained on data** in an unsupervised way
  - ..**decorrelated** from arbitrary variables (like mass)
- Potential usecase:
  - **Dark shower jets**

Dark multi-jet shower

Model

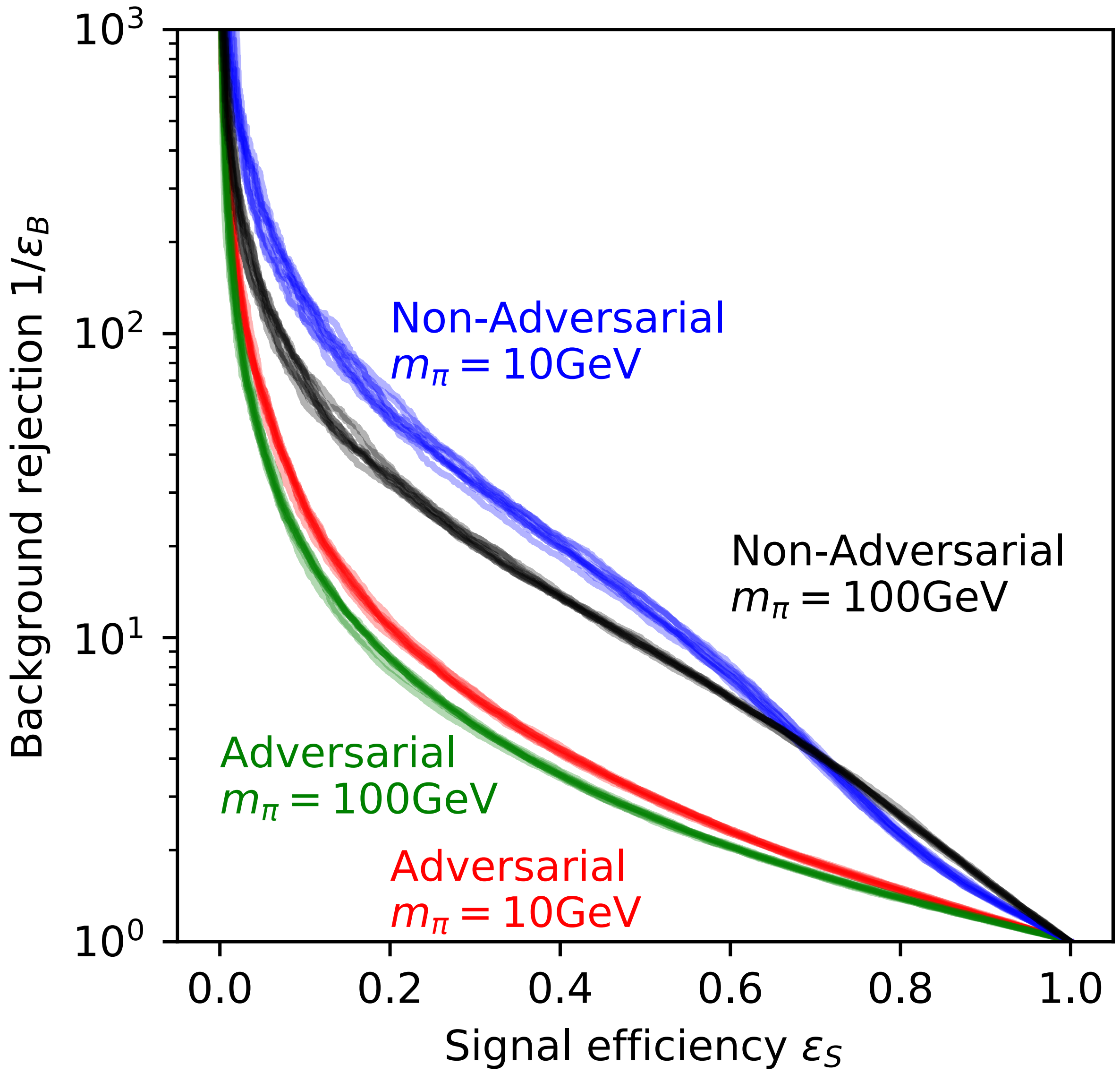
$$pp \to q_v \bar{q}_v \to q\bar{q} + \not{E}_T$$

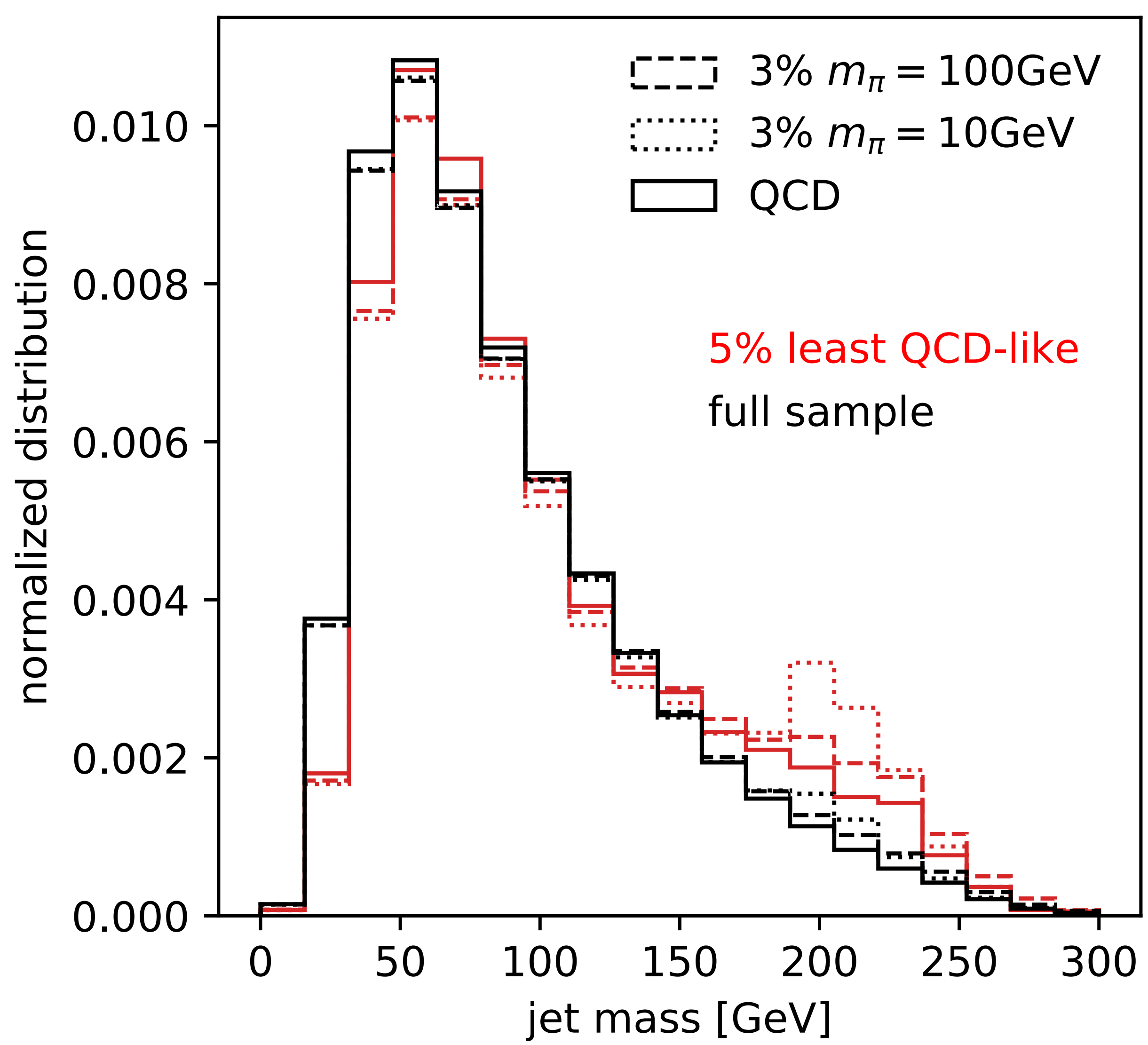
- Heavy quark q_v pair-produced
- Decay to SM partner + dark boson b
- Hadronise into dark mesons π (stable or not)
- Assume:
  - Dark SU(3)$_C$, α=0.1
  - $m_\pi = 2m_b$
  - m_q = 200 GeV



Visible Effects of Invisible Hidden Valley Radiation
L Carloni, T Sjostrand, JHEP 1009 (2010)
Discerning Secluded Sector gauge structures
L Carloni, J Rathsman, T Sjostrand, JHEP 1104 (2011)

Results





- Identify dark showers vs QCD
- Sensitivity will depend on model parameters

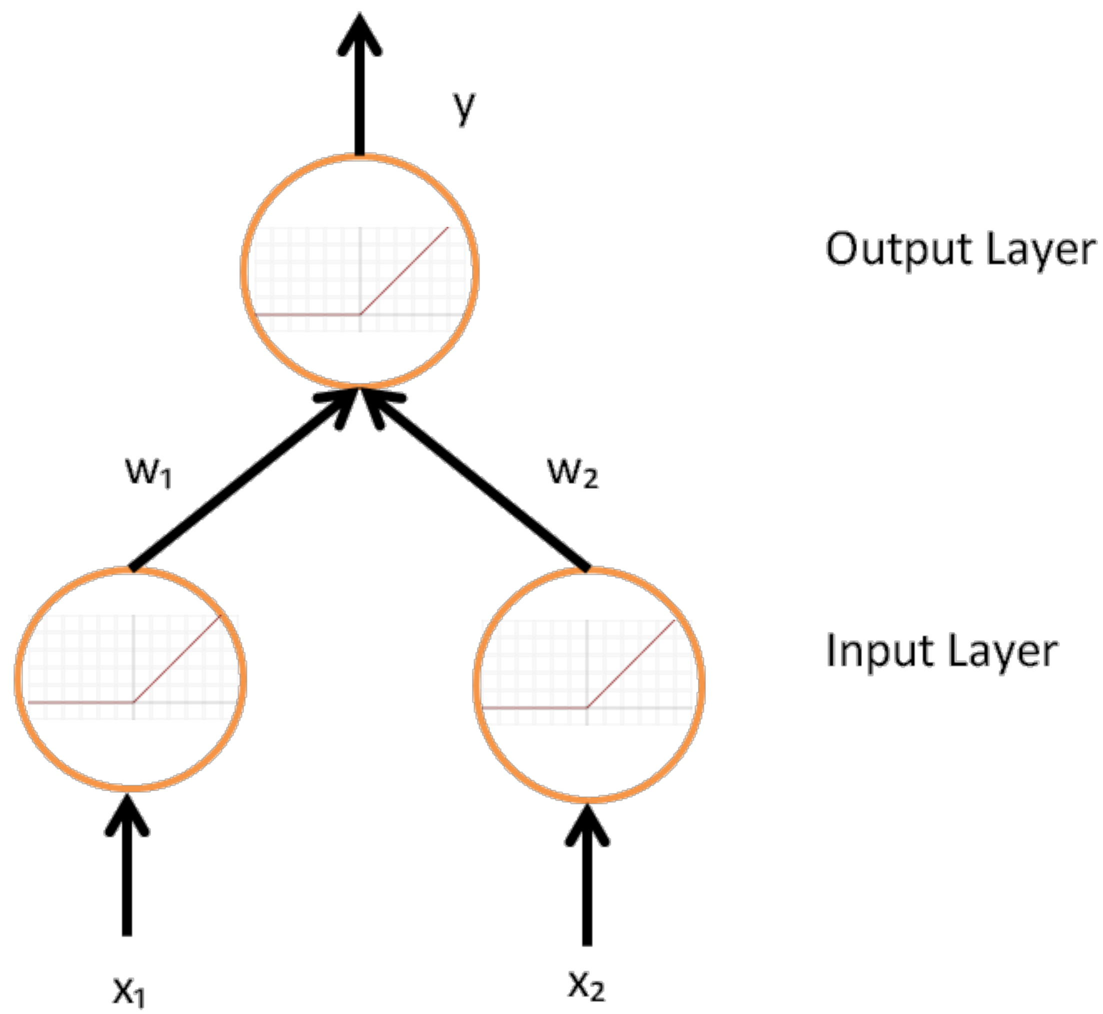
Conclusions

- Propose a new method based on unsupervised deep networks find non-SM physics as anomaly
- Can be trained from data and made independent of mass
- Explained for images, but can work with any neural network architecture
- Anti-QCD tagger: Orthogonal approach to dedicated searches

Thank you!

Backup

A Very Simple Network



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

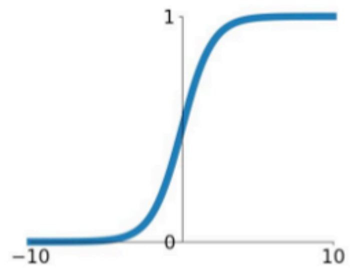
$$f(x) = \Theta(x) \cdot x$$

Activation Functions

Activation Functions

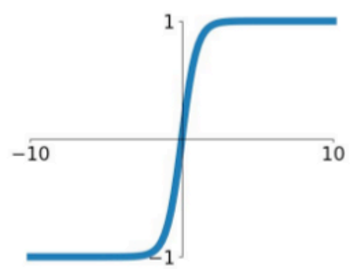
Sigmoid

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



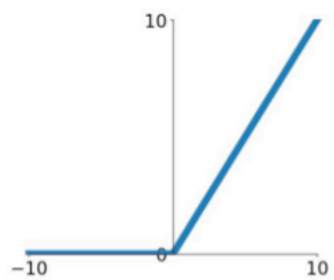
tanh

$\tanh(x)$



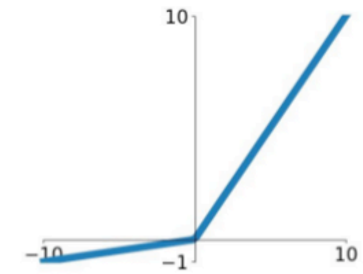
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

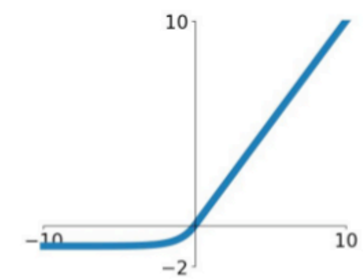


Maxout

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

ELU

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



Softmax

(for final classification layer)

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

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

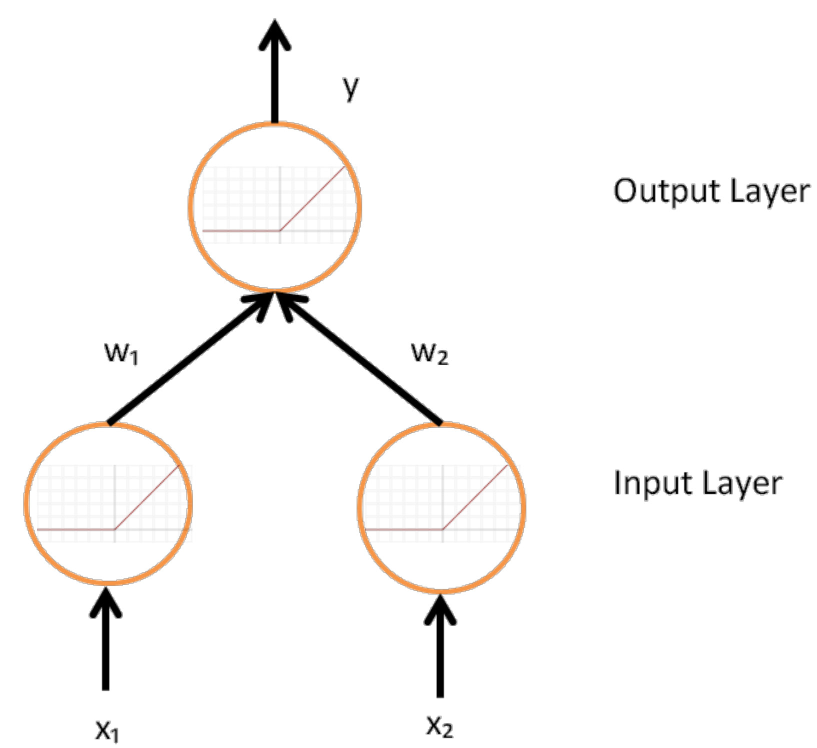
How do networks learn?

- *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)

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