Deep Learning

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

Workshop Bad Herrenalb 2018

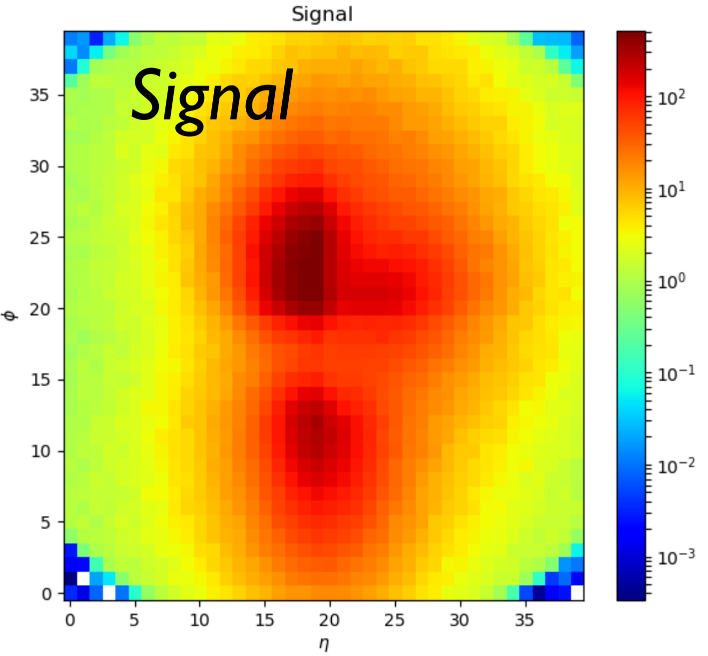
2018-09-12 - 2018-09-14

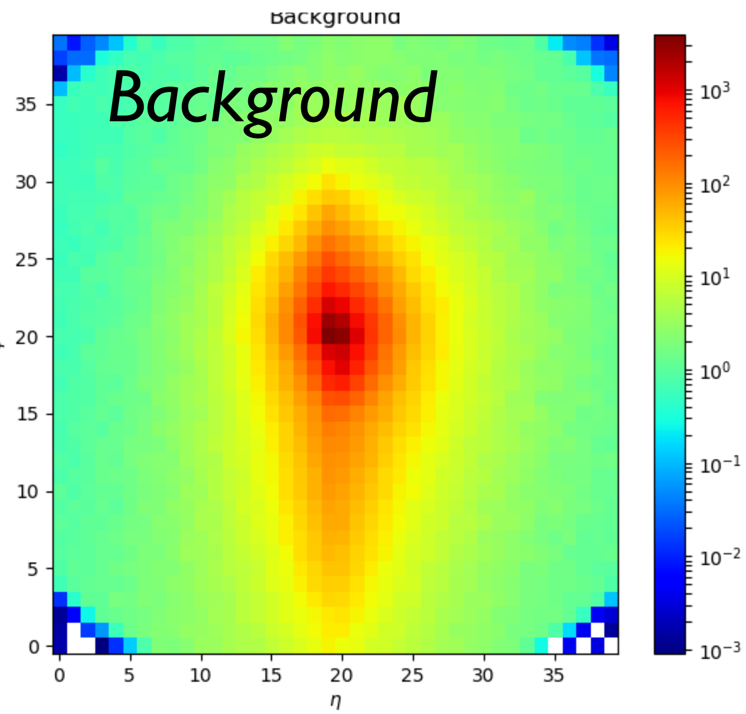
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

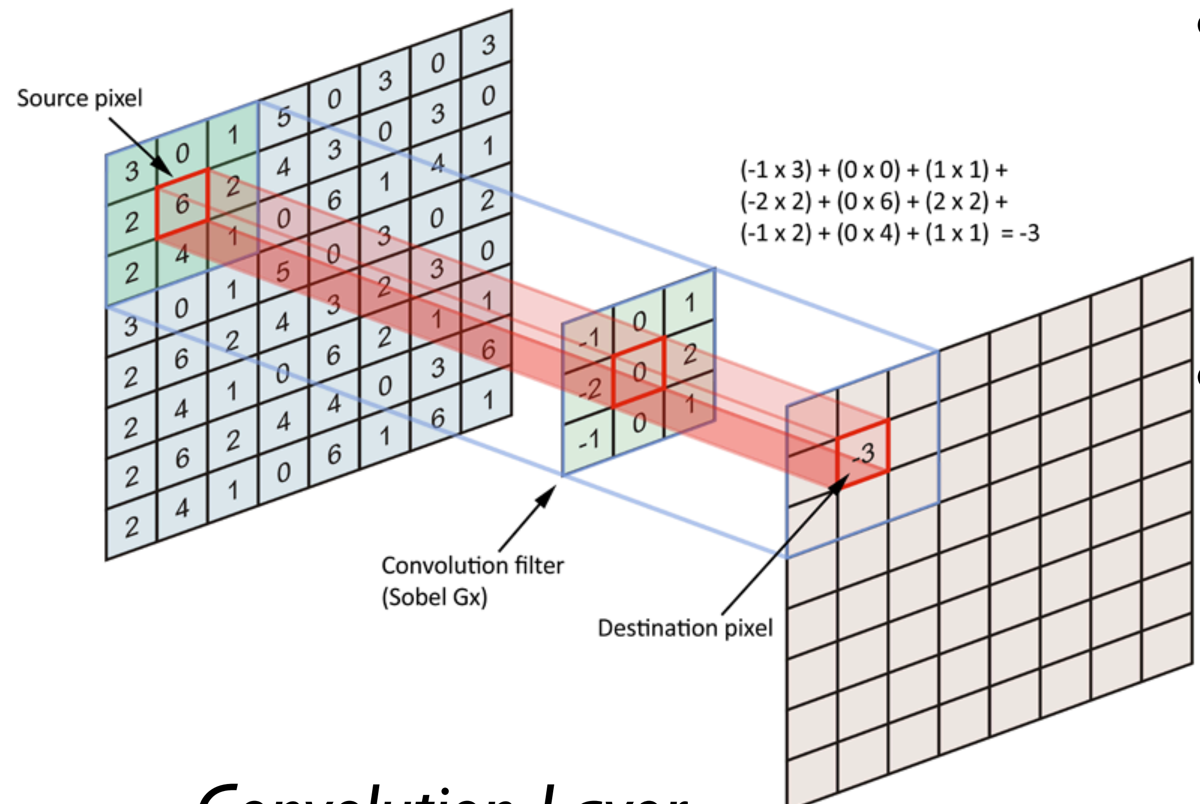
Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

Bundesministerium für Bildung und Forschung

Convolutional Networks

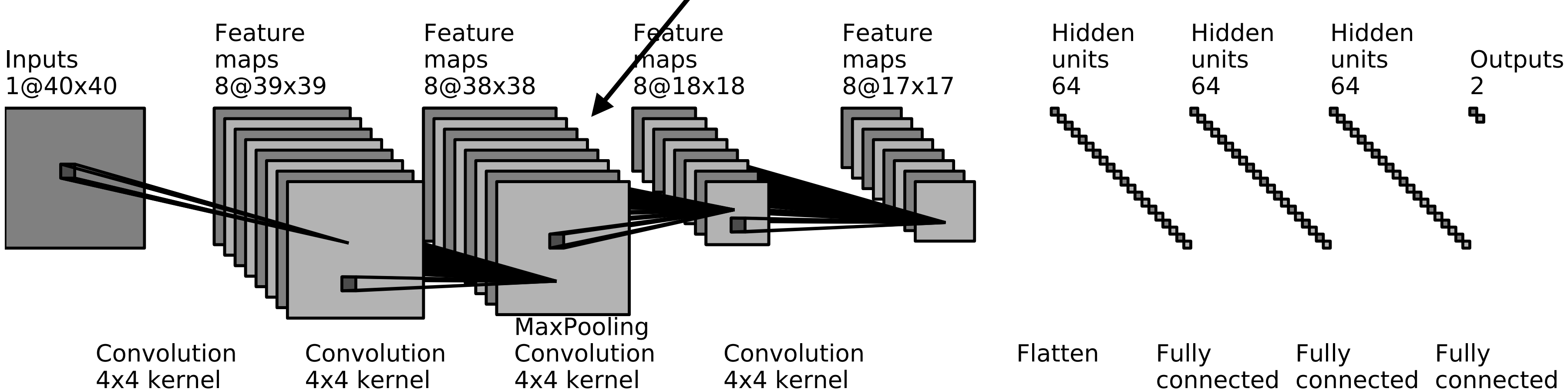






Convolution Layer

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



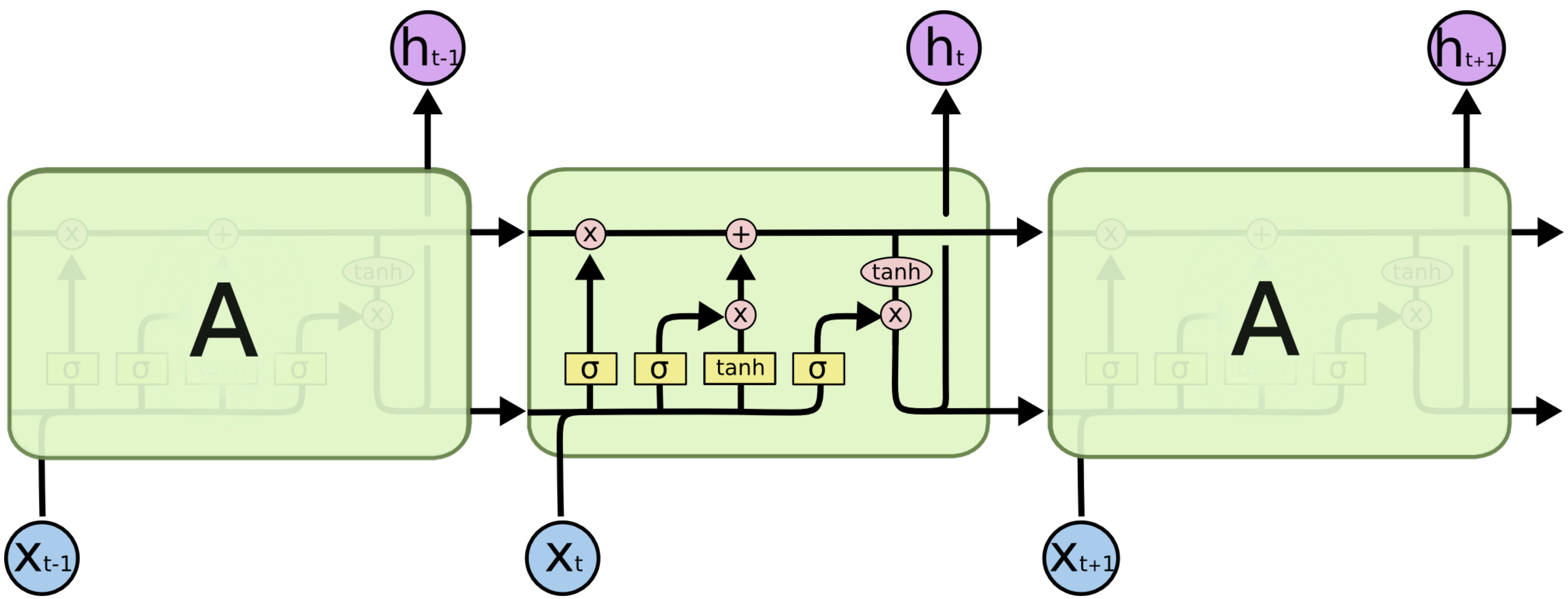
Example Architecture

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

Recurrent Networks

- Can work with 4-vectors (or n-vectors), arbitrary number of inputs, depend on ordering. LSTM or GRU are good starting points
- For concrete application: Possible combination of architectures
  - Or something completely different..



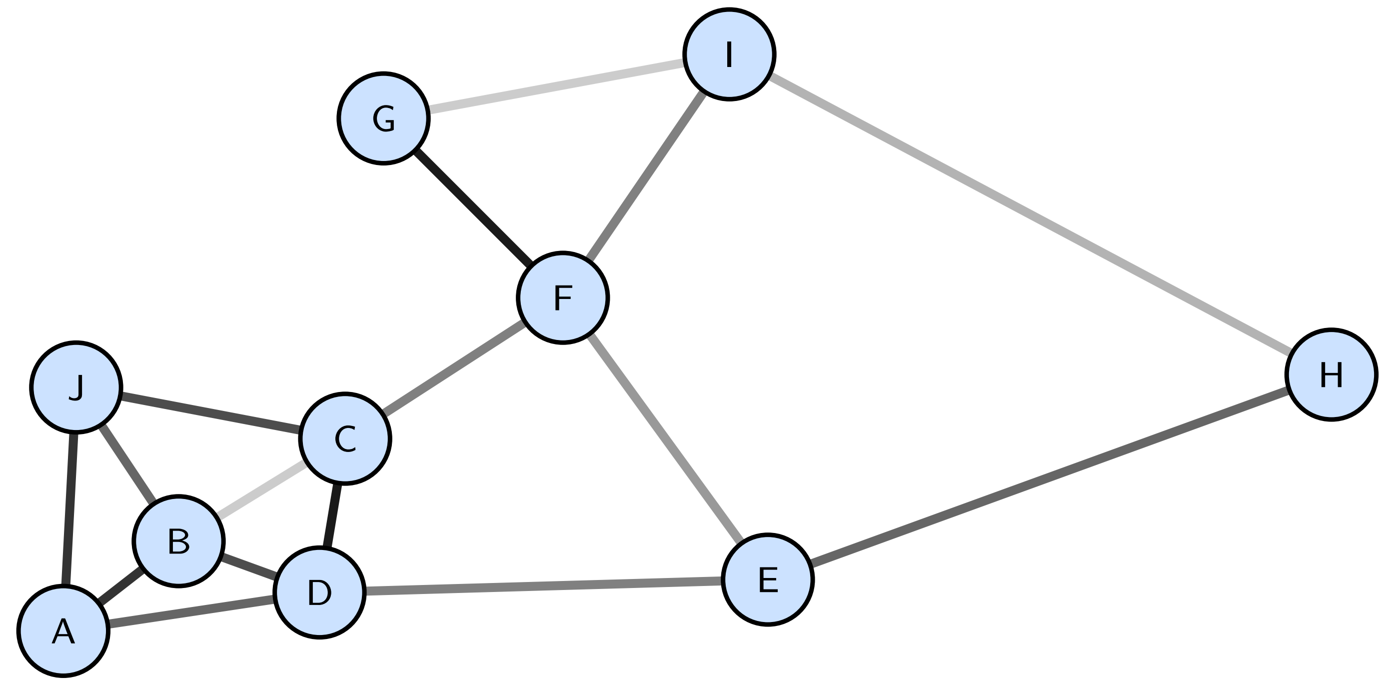
Today

- Other architecture ideas
- Dealing with systematic uncertainties
- Learning from data
- Understanding network decisions

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



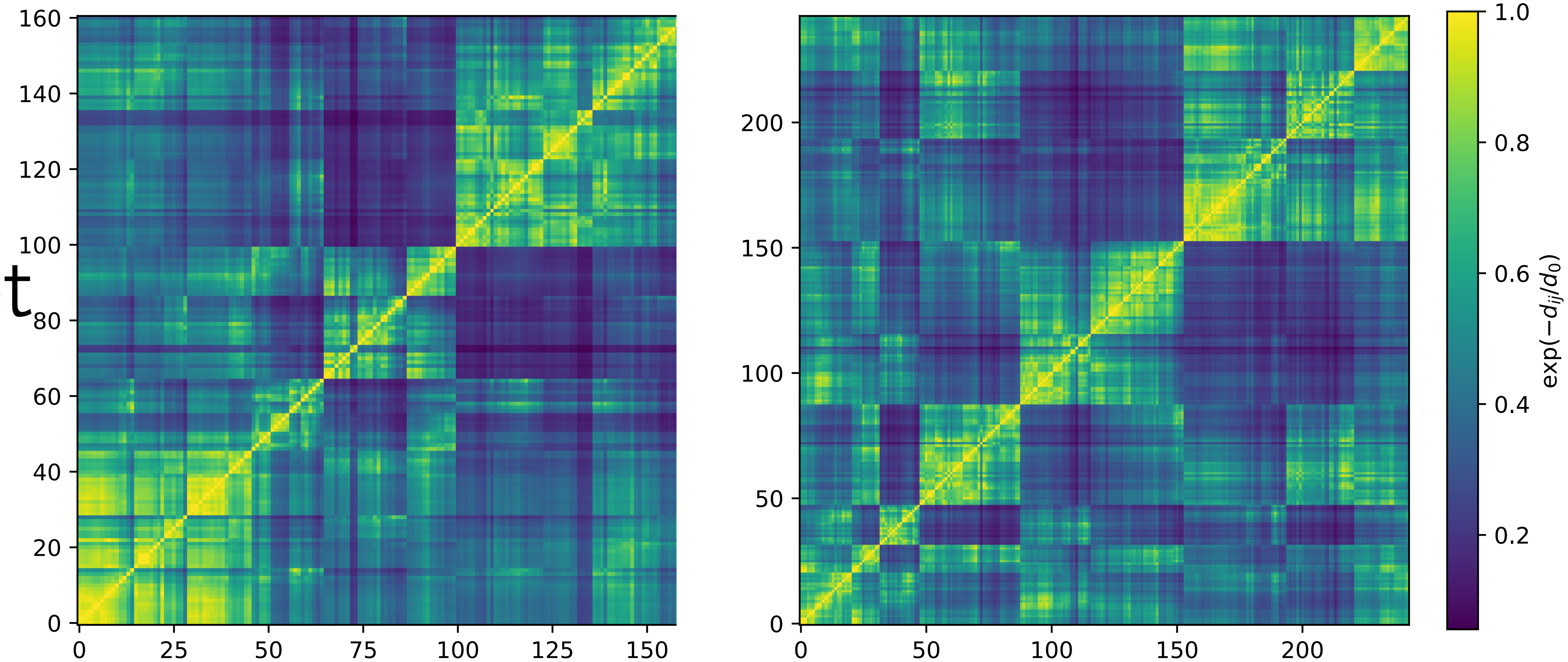
$$\mathbf{h}^{(t+1)} = \mathrm{Gc}(\mathbf{h}^{(t)}) = \rho \left(\sum_{q=1}^{|\mathcal{A}|} A_q \mathbf{h}^{(t)} \theta_q^{(t)} \right)$$

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)

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

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

- Transforms M Lorentz-vectors into M vectors with P components
- Using:
  - Per pseudo-jet variables: $\widetilde{k}_{\mu,i} \rightarrow \widetilde{k}_{0,i}$
    $\widetilde{k}_{\mu,i} \rightarrow \widetilde{k}_{\mu,i}\widetilde{k}_{\nu,i}\eta^{\mu\nu}$
  - Trainable sums: $\widetilde{k}_{\mu,i} \rightarrow \widetilde{k}_{0,j}A_{ij}$
  - Sum of differences: $\widetilde{k}_{\mu,i} \rightarrow \sum_j (\widetilde{k}_i - \widetilde{k}_j)_\mu (\widetilde{k}_i - \widetilde{k}_j)_\nu \eta^{\mu\nu} B_{ij}$

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

Ansatz

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

Metric learned

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

Deep Sets

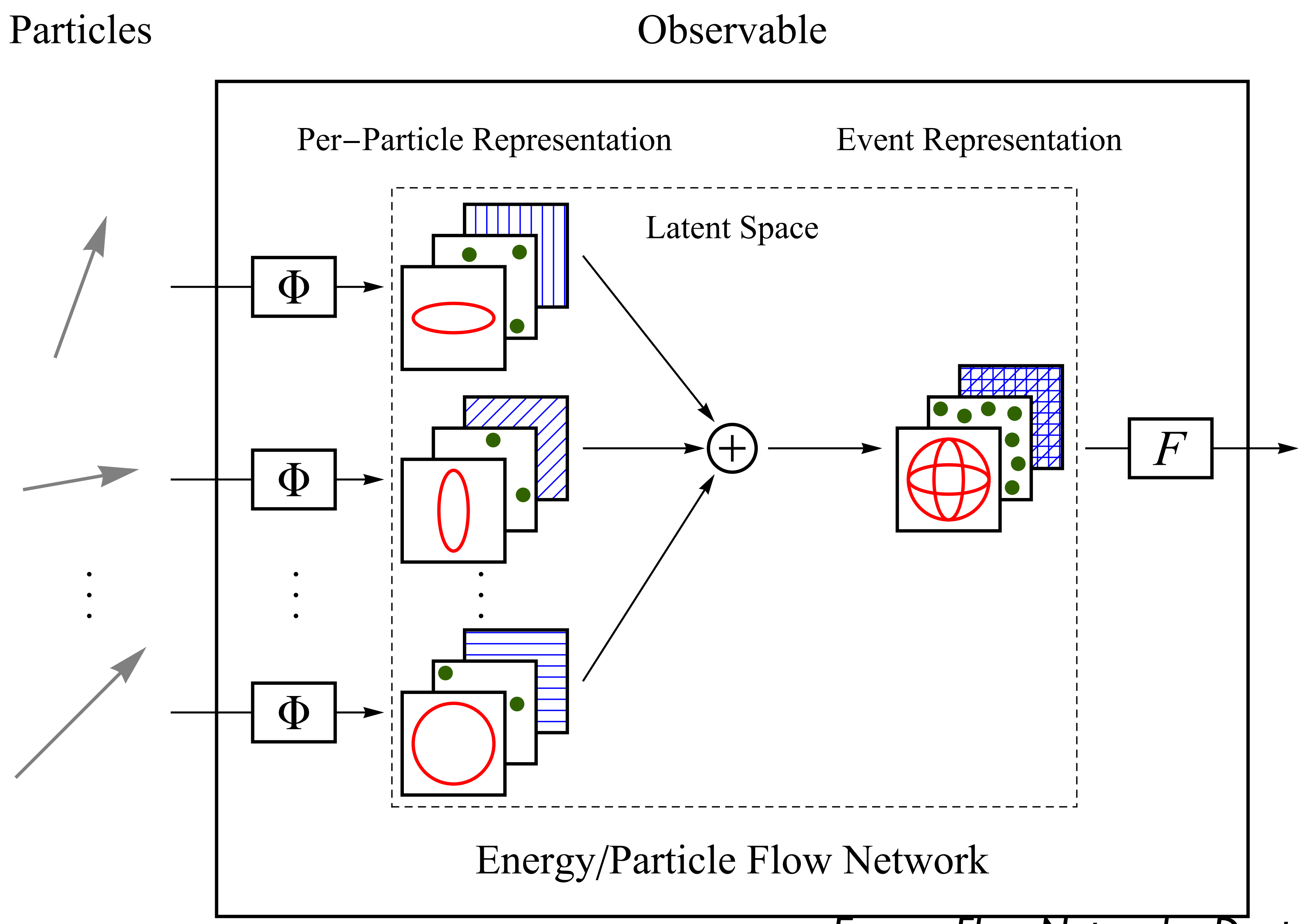
Motivation

Deep Sets Theorem [60]. *Let $\mathfrak{X} \subset \mathbb{R}^d$ be compact, $X \subset 2^{\mathfrak{X}}$ be the space of sets with bounded cardinality of elements in $\mathfrak{X}$, and $Y \subset \mathbb{R}$ be a bounded interval. Consider a continuous function $f : X \to Y$ that is invariant under permutations of its inputs, i.e. $f(x_1, \ldots, x_M) = f(x_{\pi(1)}, \ldots, x_{\pi(M)})$ for all $x_i \in \mathfrak{X}$ and $\pi \in S_M$. Then there exists a sufficiently large integer ℓ and continuous functions $\Phi : \mathfrak{X} \to \mathbb{R}^\ell$, $F : \mathbb{R}^\ell \to Y$ such that the following holds to an arbitrarily good approximation:*[1]

$$f(\{x_1, \ldots, x_M\}) = F\left(\sum_{i=1}^{M} \Phi(x_i)\right). \tag{2.1}$$

…sums are commutative…

Energy Flow Network



Energy Flow Networks: Deep Sets for Particle Jets
Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler
arXiv:1810.05165

Observables and Safety

$$\mathcal{O}\left(\{p_i\}_{i=1}^{M}\right) = F\left(\sum_{i=1}^{M} z_i\,\Phi(\hat{p}_i)\right)$$

IRC Safe

$$\mathcal{O}(\{p_1,\ldots,p_M\}) = F\left(\sum_{i=1}^{M} \Phi(p_i)\right)$$

Not Safe

Observable $\mathcal{O}$		**Map** Φ	**Function** F
Mass	m	p^μ	$F(x^\mu) = \sqrt{x^\mu x_\mu}$
Multiplicity	M	1	$F(x) = x$
Track Mass	m_{track}	$p^\mu \mathbb{I}_{\text{track}}$	$F(x^\mu) = \sqrt{x^\mu x_\mu}$
Track Multiplicity	M_{track}	$\mathbb{I}_{\text{track}}$	$F(x) = x$
Jet Charge [69]	$\mathcal{Q}_\kappa$	$(p_T, Q\,p_T^\kappa)$	$F(x,y) = y/x^\kappa$
Eventropy [71]	$z\ln z$	$(p_T, p_T \ln p_T)$	$F(x,y) = y/x - \ln x$
Momentum Dispersion [90]	p_T^D	(p_T, p_T^2)	$F(x,y) = \sqrt{y/x^2}$
C parameter [91]	C	$(\lvert\vec{p}\rvert, \vec{p}\otimes\vec{p}/\lvert\vec{p}\rvert)$	$F(x,Y) = \frac{3}{2x^2}[(\operatorname{Tr} Y)^2 - \operatorname{Tr} Y^2]$

Learning for Discovery

Cross-entropy and Asimov significance

$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2}$$

- When optimizing a classifier a typical approach is to optimize the accuracy
- For neural networks standard approach for training a binary classifier is the **cross-entropy**
- Accuracy maximizing is equivalent to minimizing the cross-entropy

$$s = W_s \sum_i^{N_{batch}} y_i^{pred} \times y_i^{true}$$

$$b = W_b \sum_i^{N_{batch}} y_i^{pred} \times (1 - y_i^{true})$$

$1/Z_A(s,b)$ becomes a smooth function of y_i^{pred}

- **Can we optimize directly for the Asimov significance, i.e. can we use it as a**
  <u>loss function</u> ?
- **Caveat**: To define the number of signal and background events we need to cut on the discriminator output
  - Makes it non-differentiable ??
  - Differentiability is needed for gradient descent learning

- A single sigmoid output neuron
- Replace the discrete number of signal and background events by a smooth function of the predicted label

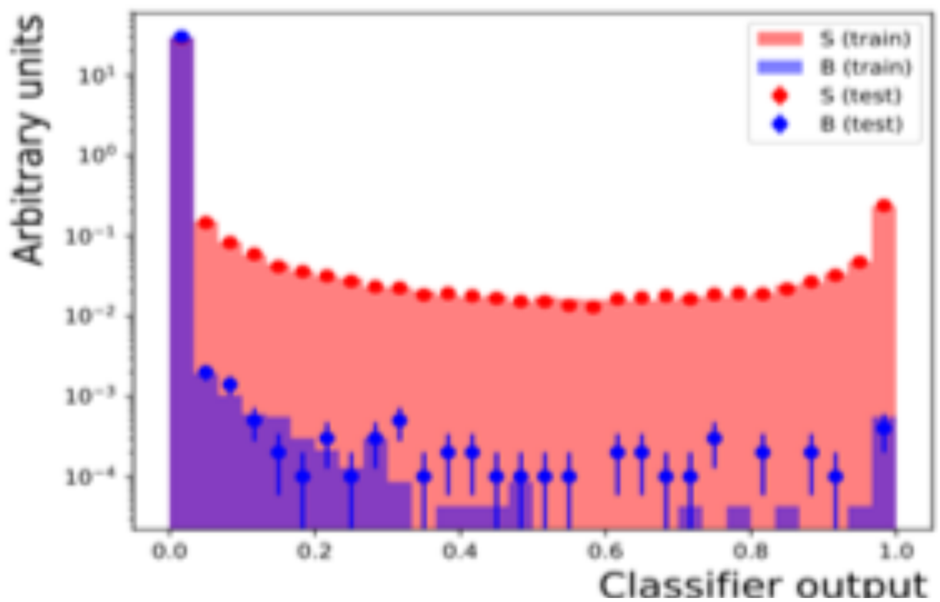
Results

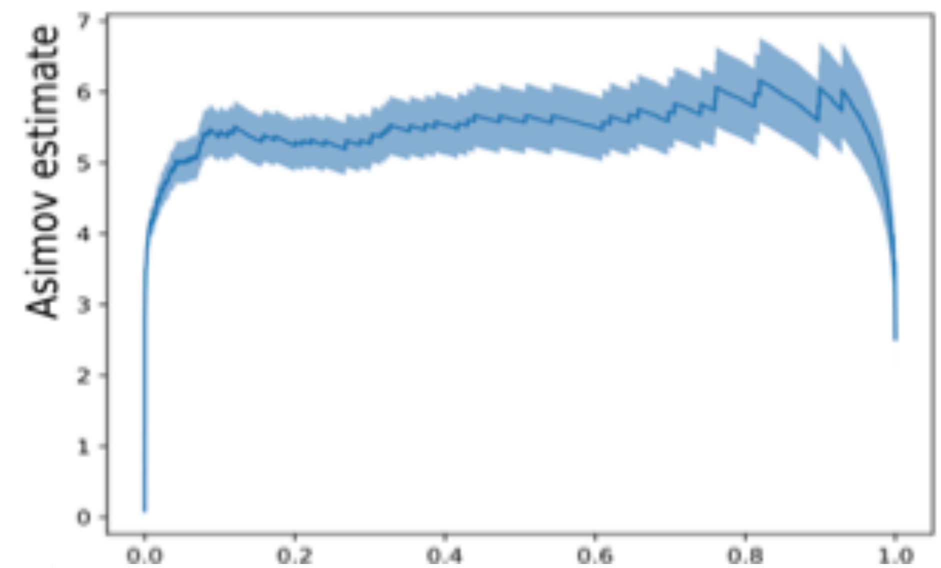
1M events
21 inputs
fully connected network
Large (4096) batch size needed!

low level	high level
$\vec{p}_l$	m_T
$\vec{p}_{jet(1,2,3)}$	m_{T2}^W
n_{jet}	$\not{E}_T$
$n_{b\,jet}$	H_T

Asimov loss training

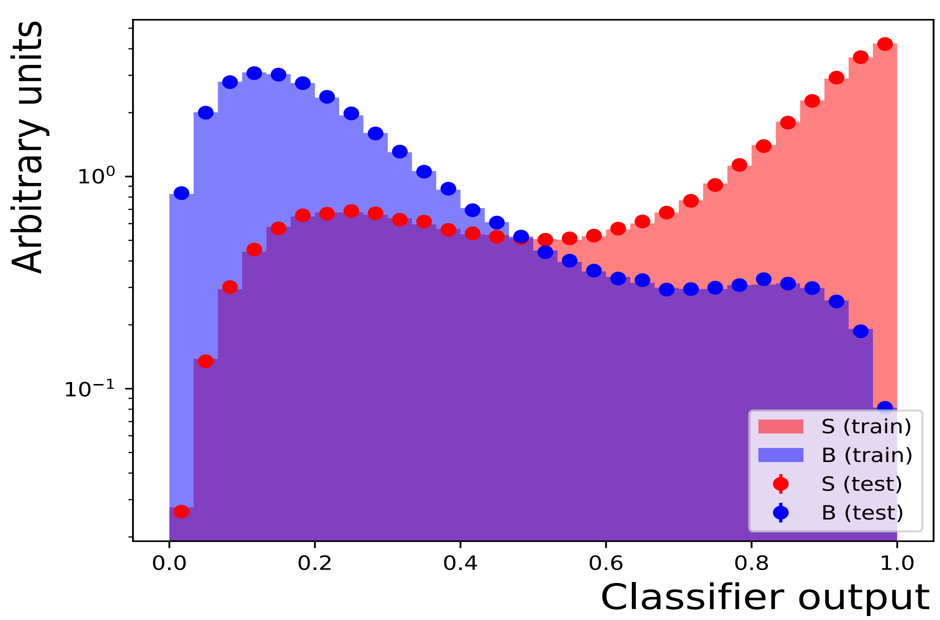
- best $\mathbf{Z_A = 6.2 \pm 0.6}$
- **Acc = 59%**
- AUC=0.80
- Tries to find a background free region

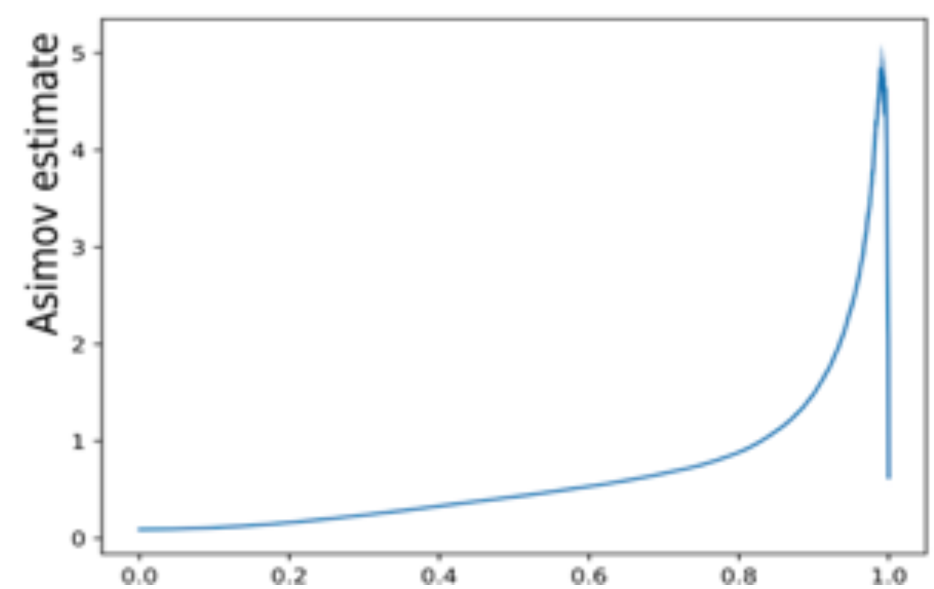




Cross-entropy training + purity cut

- best $\mathbf{Z_A = 4.8 \pm 0.3}$
- **Acc = 92%**
- AUC=0.87





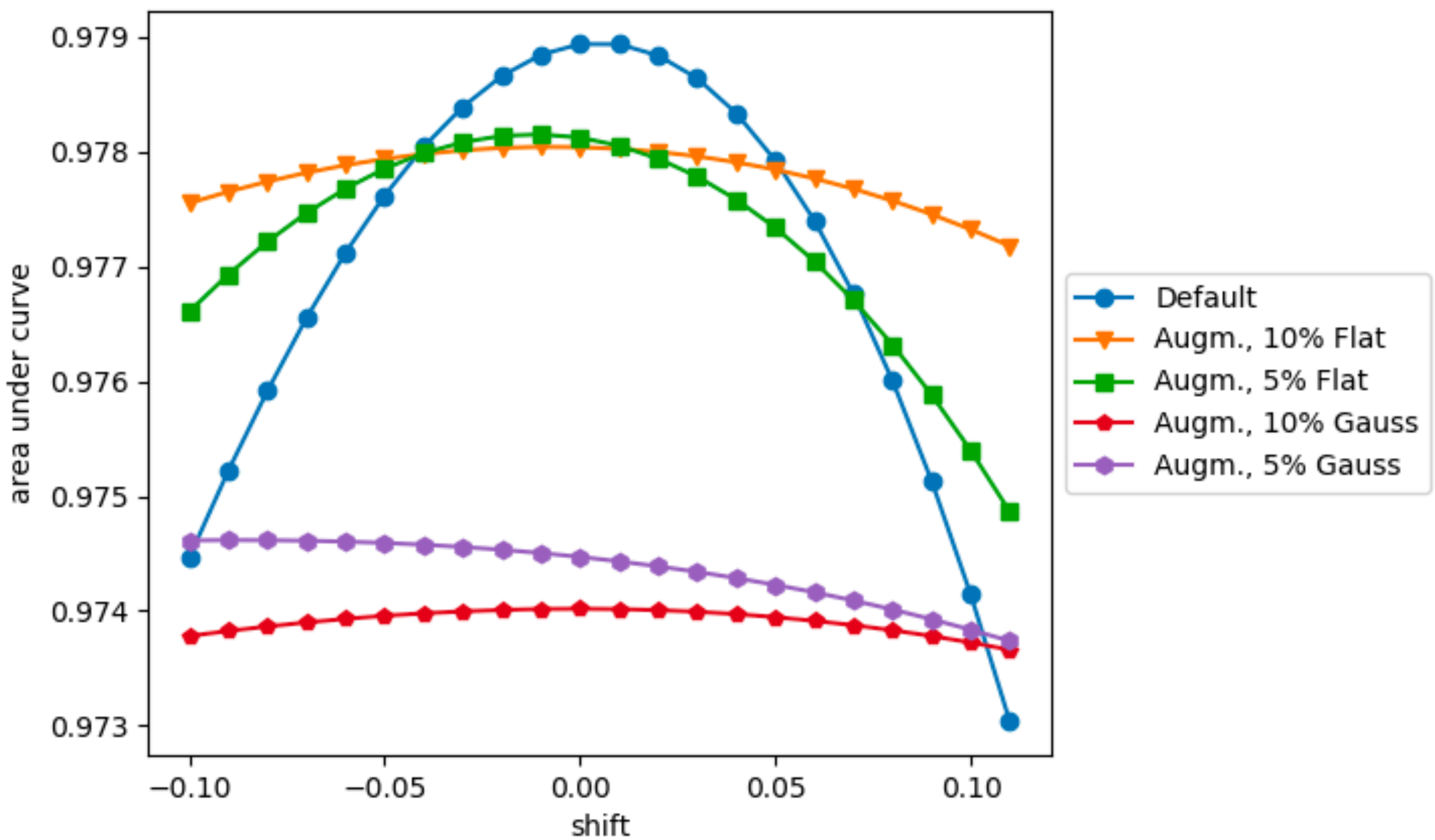
Systematic uncertainty 50%, Differences in Z_A shrink for small systematic uncert.

https://arxiv.org/abs/1806.00322

Uncertainties

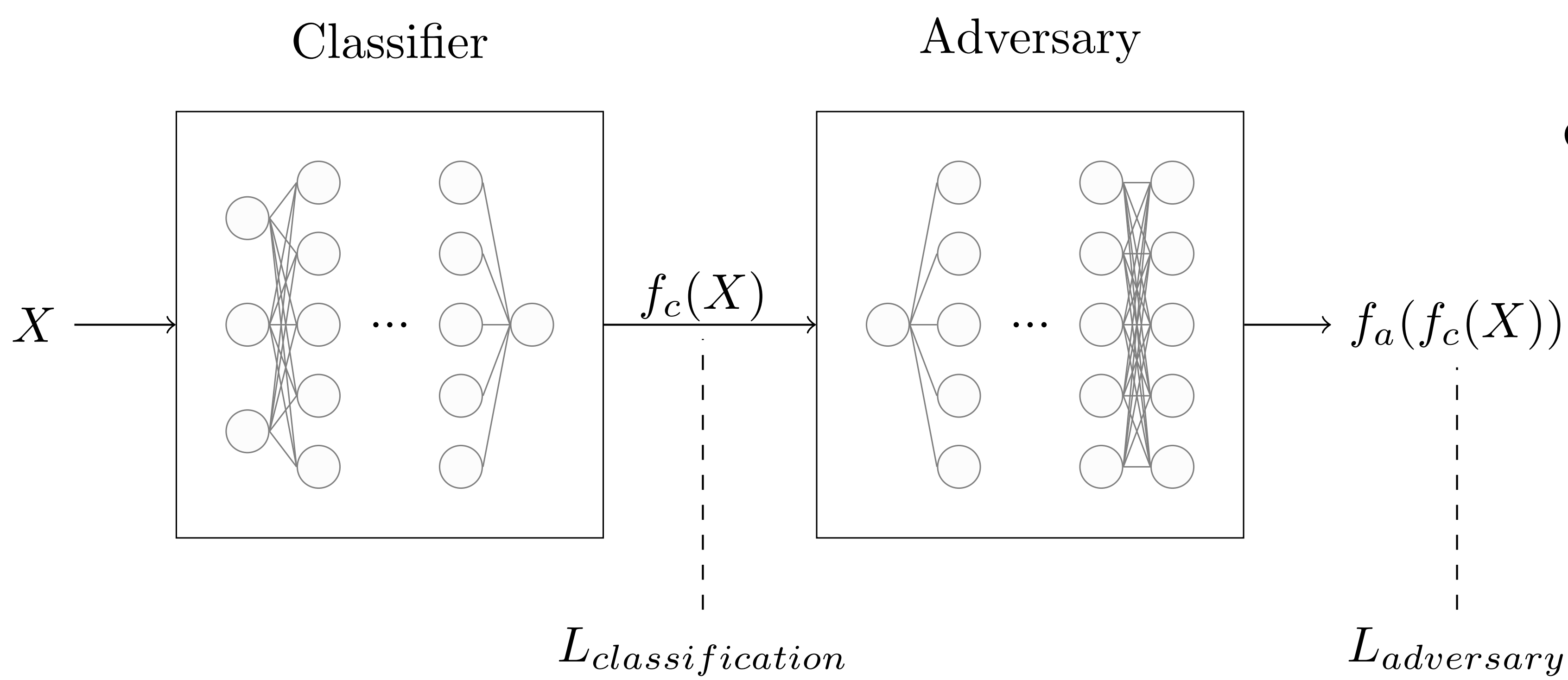
Data Augmentation

- Test network response under global rescaling of all 4-vector inputs (simimilar to jet energy scale)
- Re-train network using shifted samples as well.
  - So the network sees multiple (shifted) copies of the event = *data augmentation*
- Trade off performance and stability
- Now looking into multiple simultaenous uncertainties
  - resolution
  - pile up
  - lost particles
  - …
- Can adversarial training help further?



Plot by Sven Bollweg

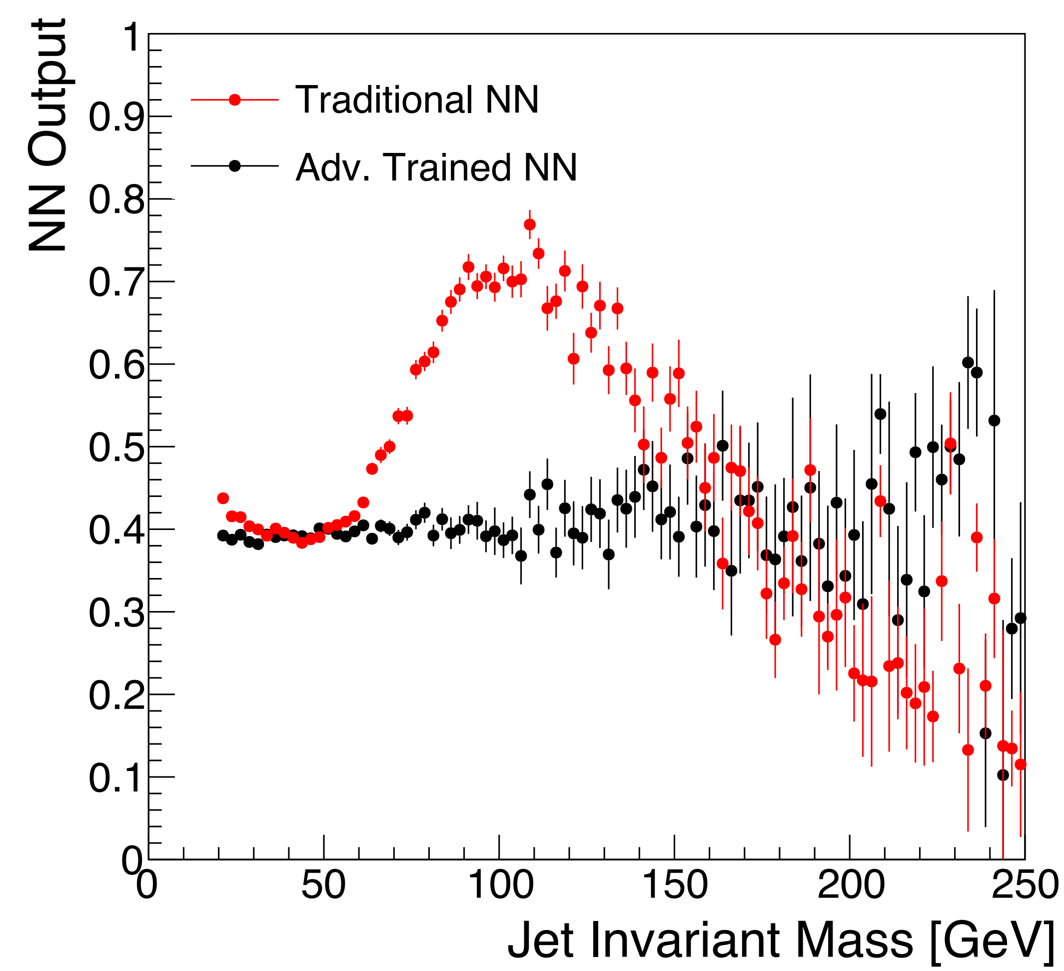
Removing Correlations



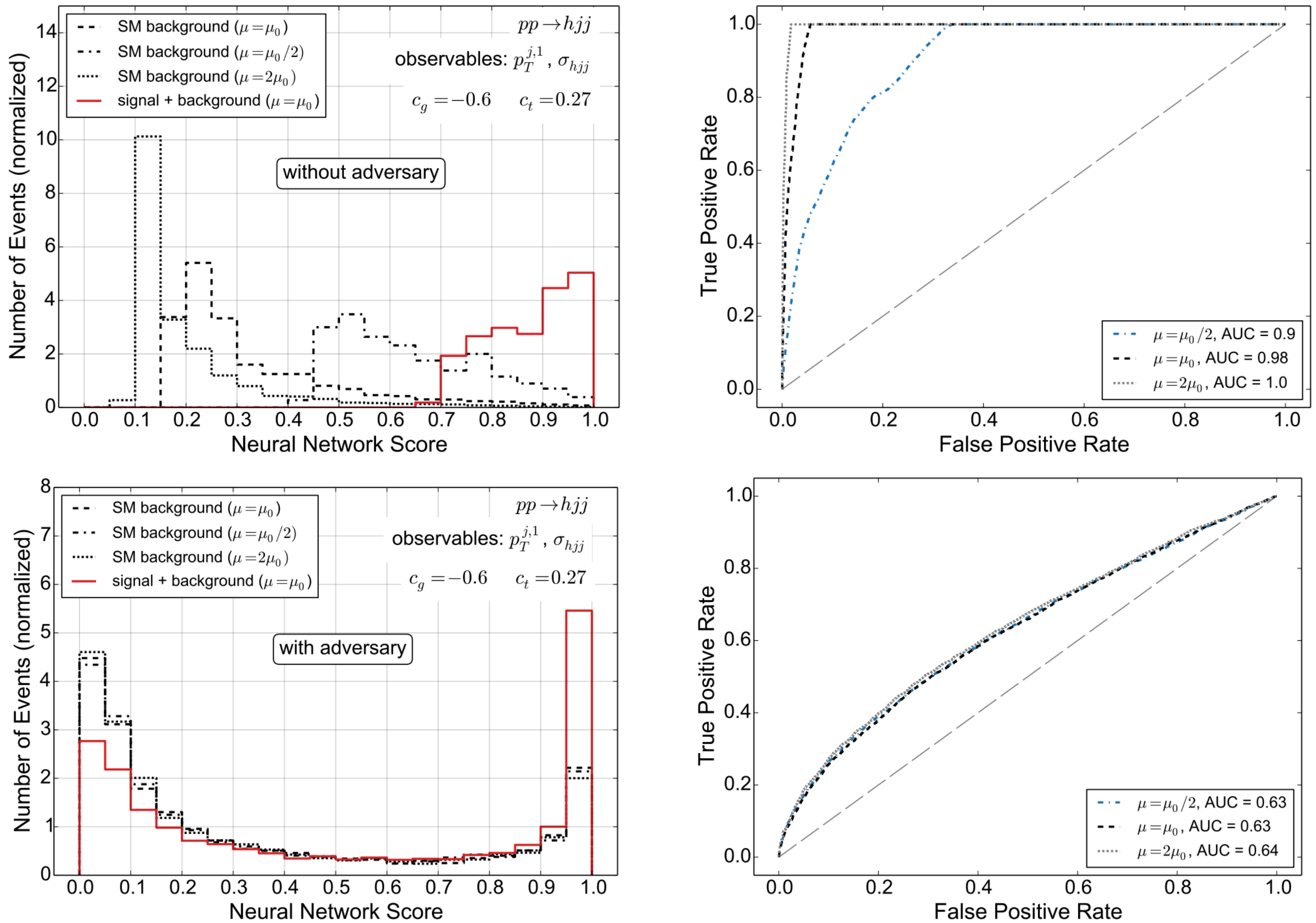
Decorrelated Jet Substructure Tagging using Adversarial Neural Networks
C Shimmin, P Sadowski, P Baldi, E Weik, D Whiteson, E Goul, A Søgaard 1703.03507

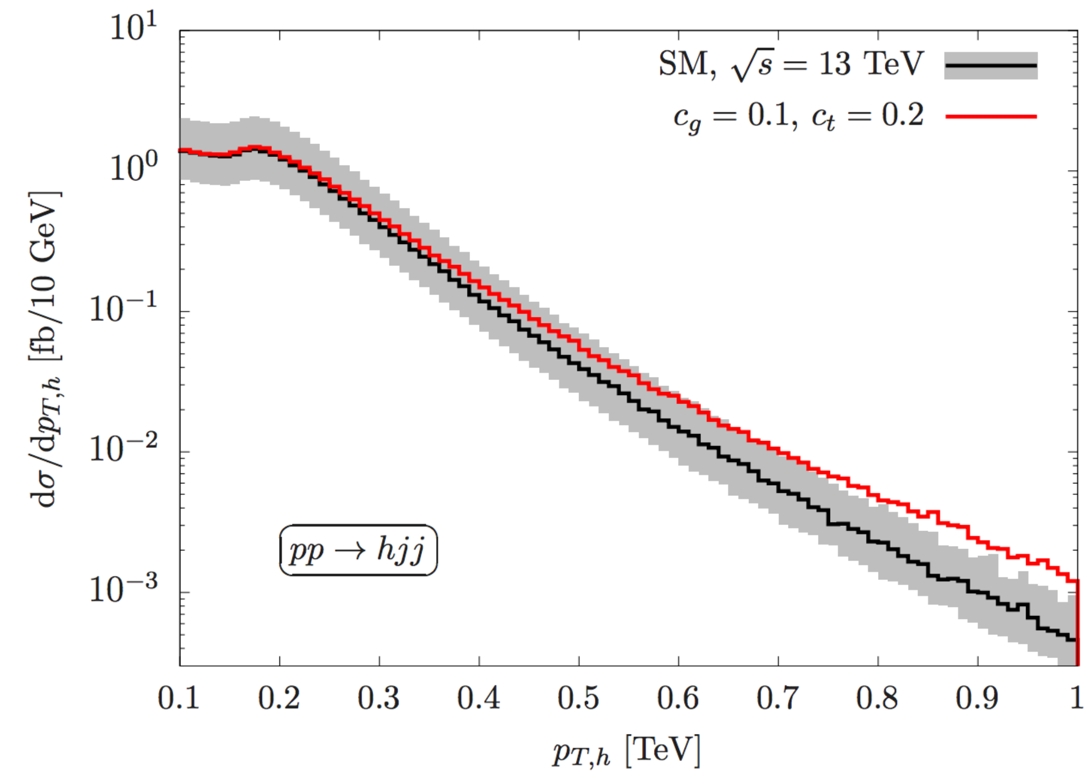
$$L_{\text{tagger}} = L_{\text{classification}} - \lambda L_{\text{adversary}}$$

- Classifier:
  - Distinguish Z' from QCD
- Adversary:
  - Infer jet mass
- **Trade-off discrimination power and stability**



Adversaries



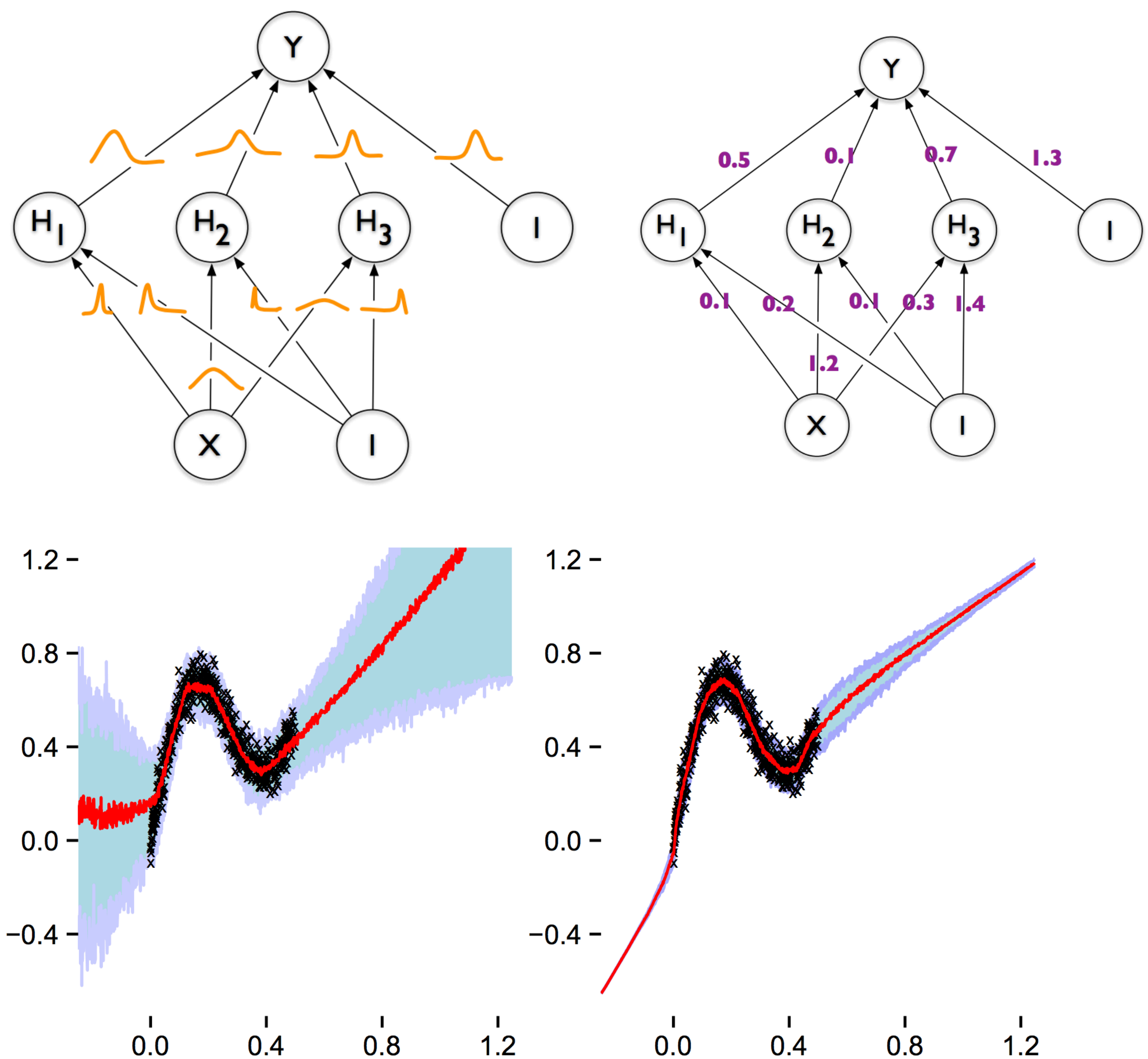


$$\mathcal{L}_{\mathrm{d}6} = c_g \mathcal{O}_g + c_t \mathcal{O}_t = \frac{c_g\, g_s^2}{16\pi^2 v}\, h\, G^{a\,\mu\nu} G^a_{\mu\nu} + c_t\, h\, \bar{t} t$$

- Goal: distinguish SM Higgs boson from new physics in EFT approach (using Hjj events)
- Problem: SM scale uncertainty can look similar to signal
- Solution: Train network to be invariant to MC scale choice, again using adversarial approach

Machine Learning Uncertainties with Adversarial Neural Networks
C Englert, P Galler, P Harris, M Spannowsky, 1807.08763

Bayesian Networks

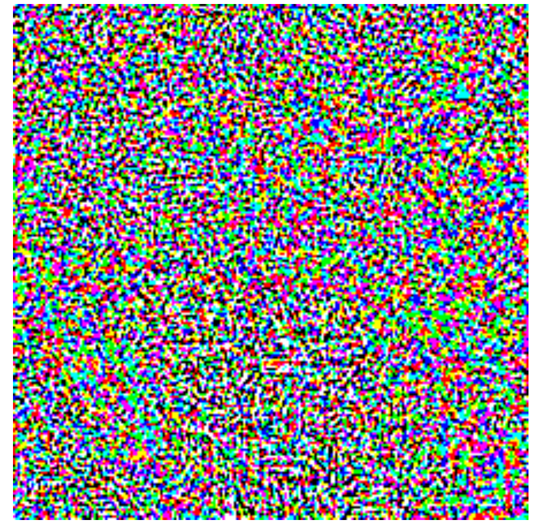


- So far discussed handling uncertainties on the inputs
- How can we with training data not fully covering the phase space?

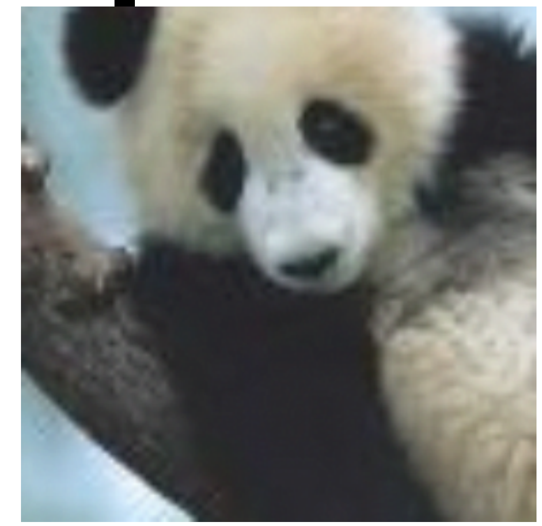
Weight Uncertainty in Neural Networks
C Blundell et al, ICML Proc's 2015

Adverserial Examples

$+ .007 \times$



$=$



$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda" 57.7% confidence	"nematode" 8.2% confidence	"gibbon" 99.3 % confidence

- Attempt to trick an image classifying network
    - A small amount of specifically crafted noise can greatly alter the prediction of a network
- Recent work shows even single pixel changes can matter

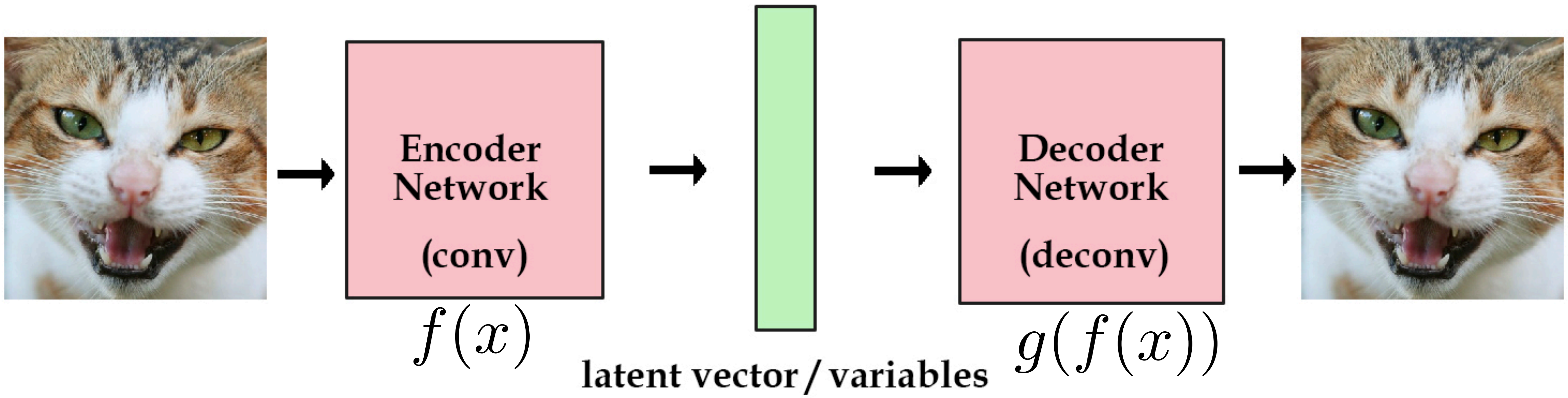
$$\boldsymbol{w}^\top \tilde{\boldsymbol{x}} = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{\eta}$$

weights *input* *perturbation*

Explaining and Harnessing Adverserial Examples
IJ Goodfellow, J Shlens, C Szegedy
ICLR Proc. 2015
One pixel attack for fooling deep neural networks
J Su, D Vasconcellos Vargas, S Kouichi
1710.08864

Less Simulation

Autoencoder

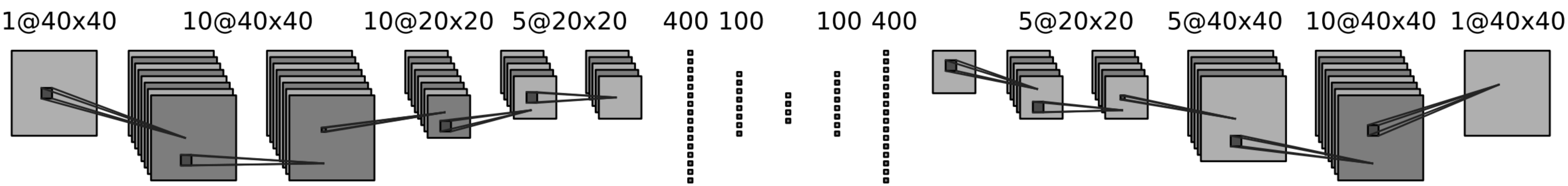


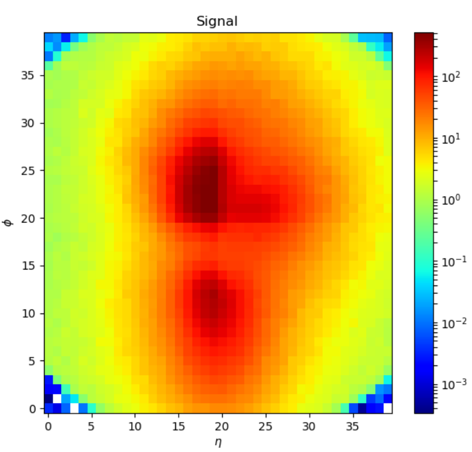
$$L = (\hat{y} - g(f(x)))^2$$

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

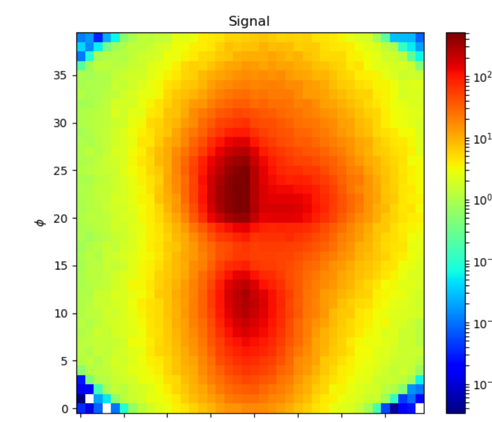
kvfrans
deeplearningbook.org

Autoencoder for Physics

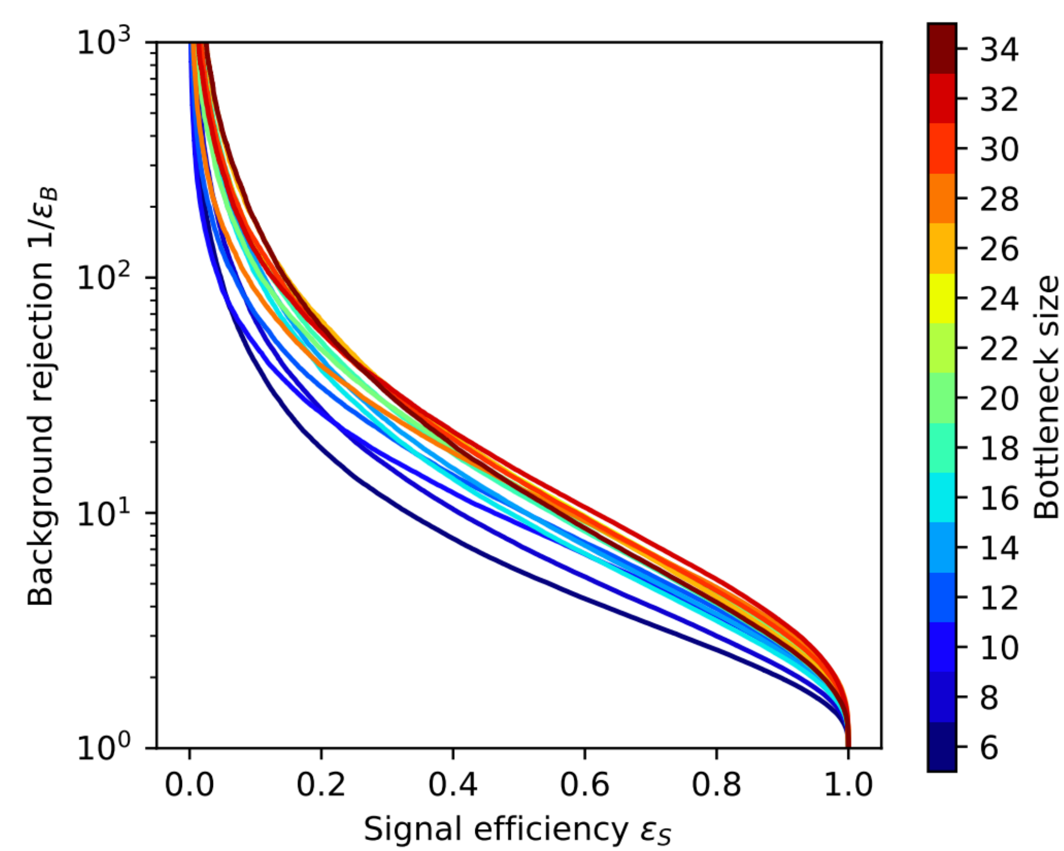




$$L_{\text{auto}} = \sum_{1600\ \text{pixels}} \left(k_T^{\text{norm,in}} - k_T^{\text{auto}} \right)^2$$



- 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

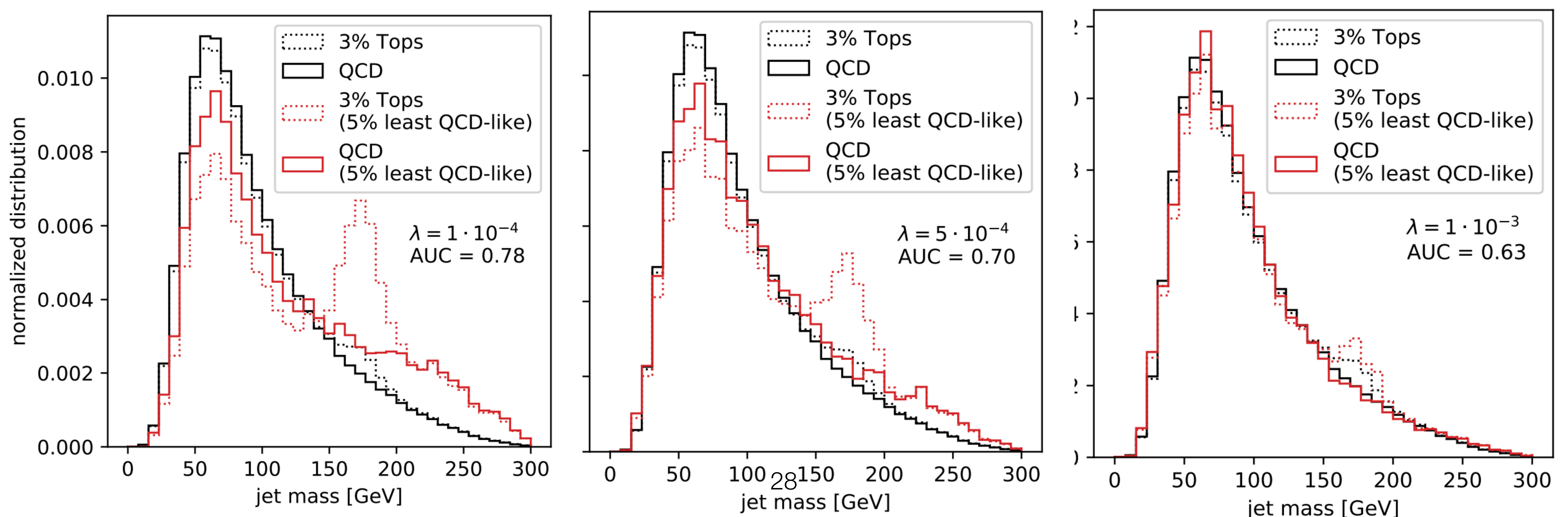
Mass Sculpting

- Autoencoder alone will also learn mass distribution
- Counteract with adversary:

$$L_{\text{adv}}(M) = \left[\widetilde{M}\left(\left|k_{T,i}^{\text{adv}} - k_{T,i}^{\text{auto}}\right|\right) - M\right]^2$$

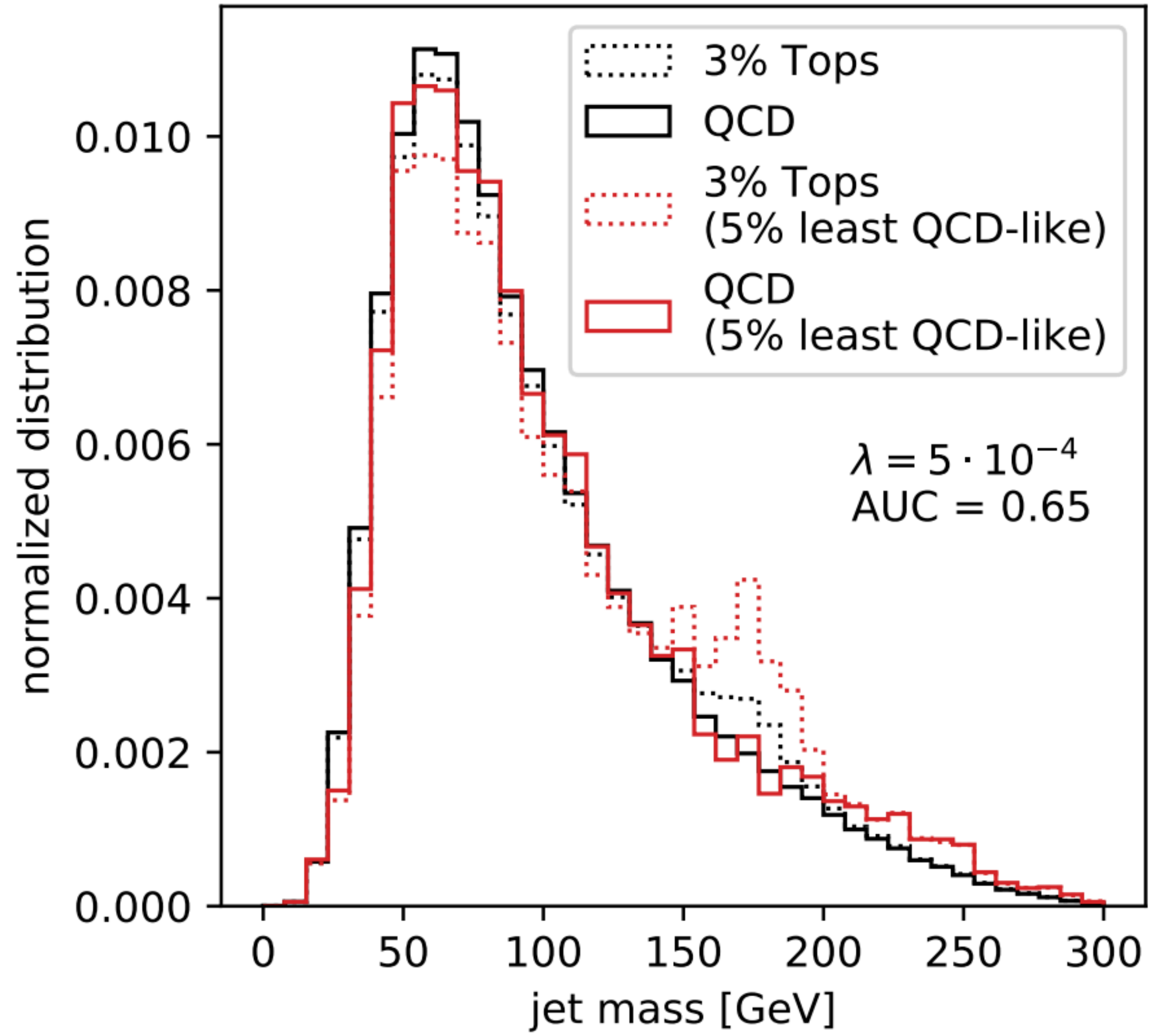
$$L = L_{\text{auto}} - \lambda\, L_{\text{adv}}(M)$$

- Tune mass dependency with Lagrange multiplier



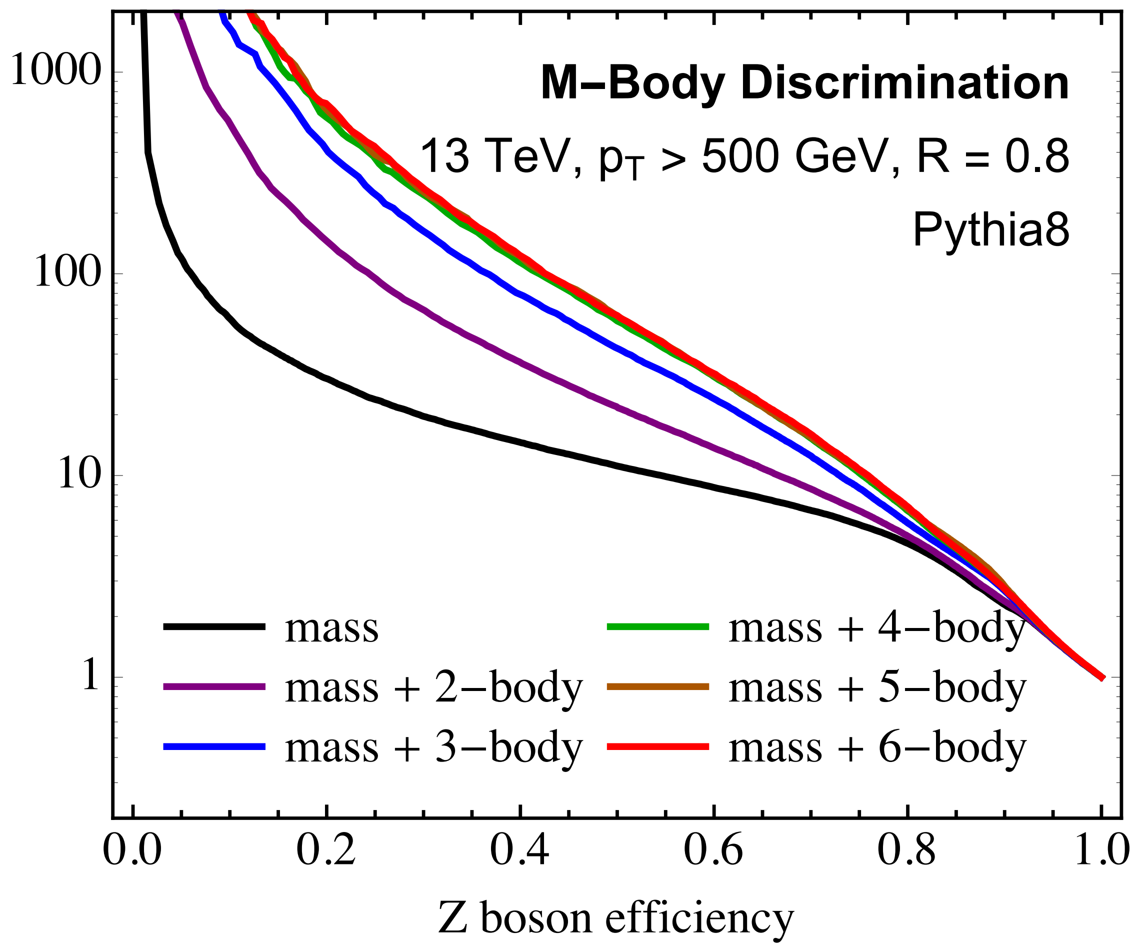
Signal contamination

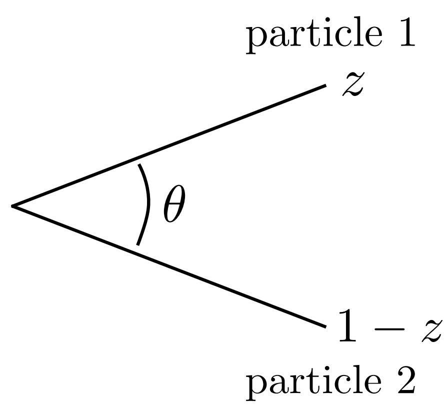
- Procedure works also when signal is present in training data
- This means a search for exotic new physics with unknown shower patterns (dark showers) could be done using data-only training

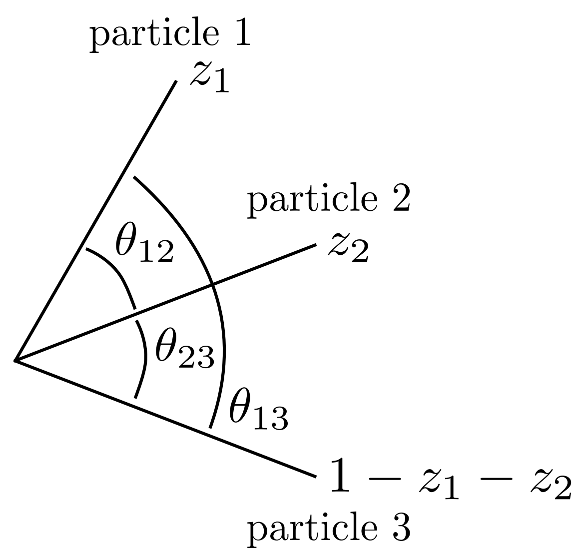


Understanding

How much information is in a jet?







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

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

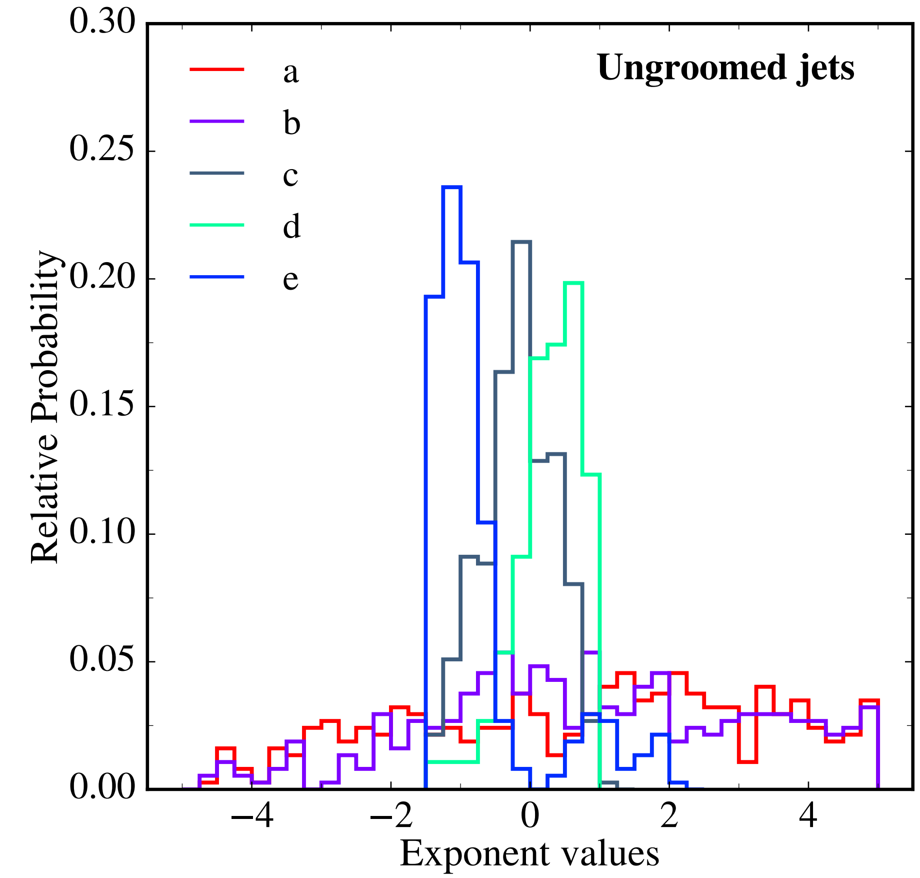
- Study W tagging
- Use n-subjettiness with different exponents to build basis
- Train ANNs with different numbers of variables

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

New Variables

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

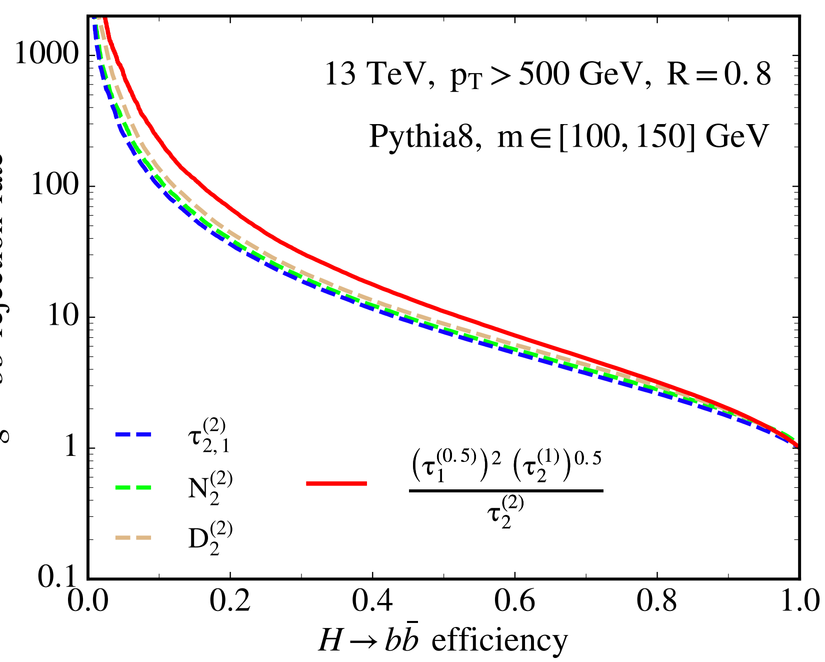
Parametrise phase space

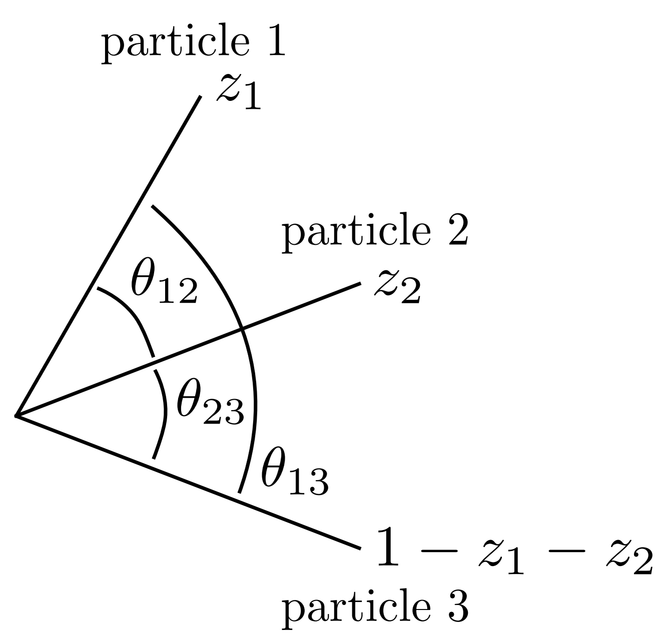


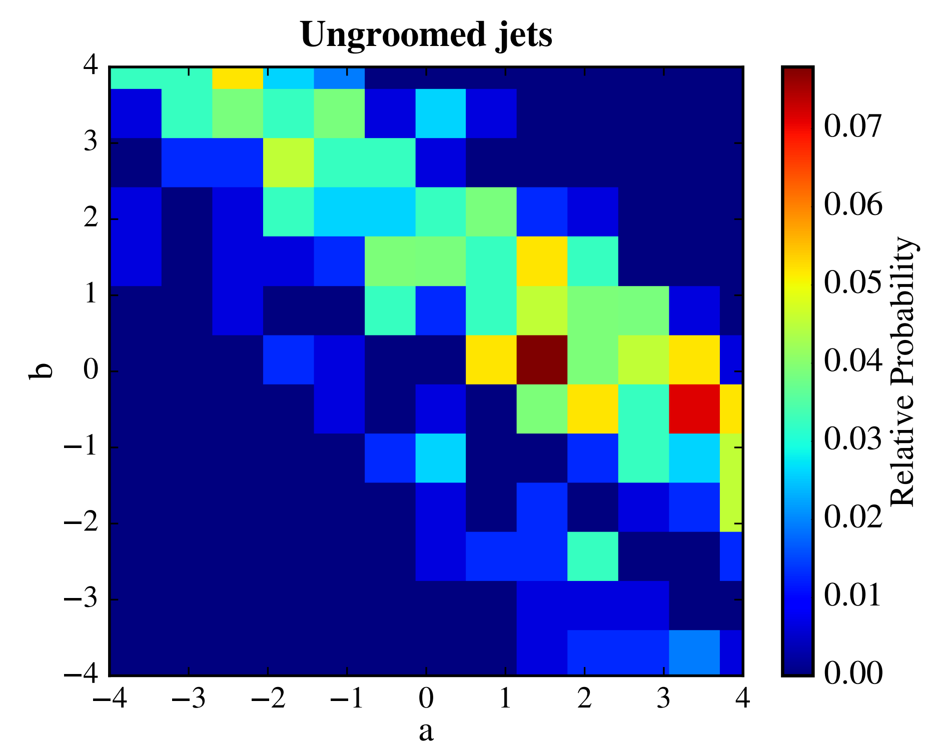
Find optimal exponents using MC

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

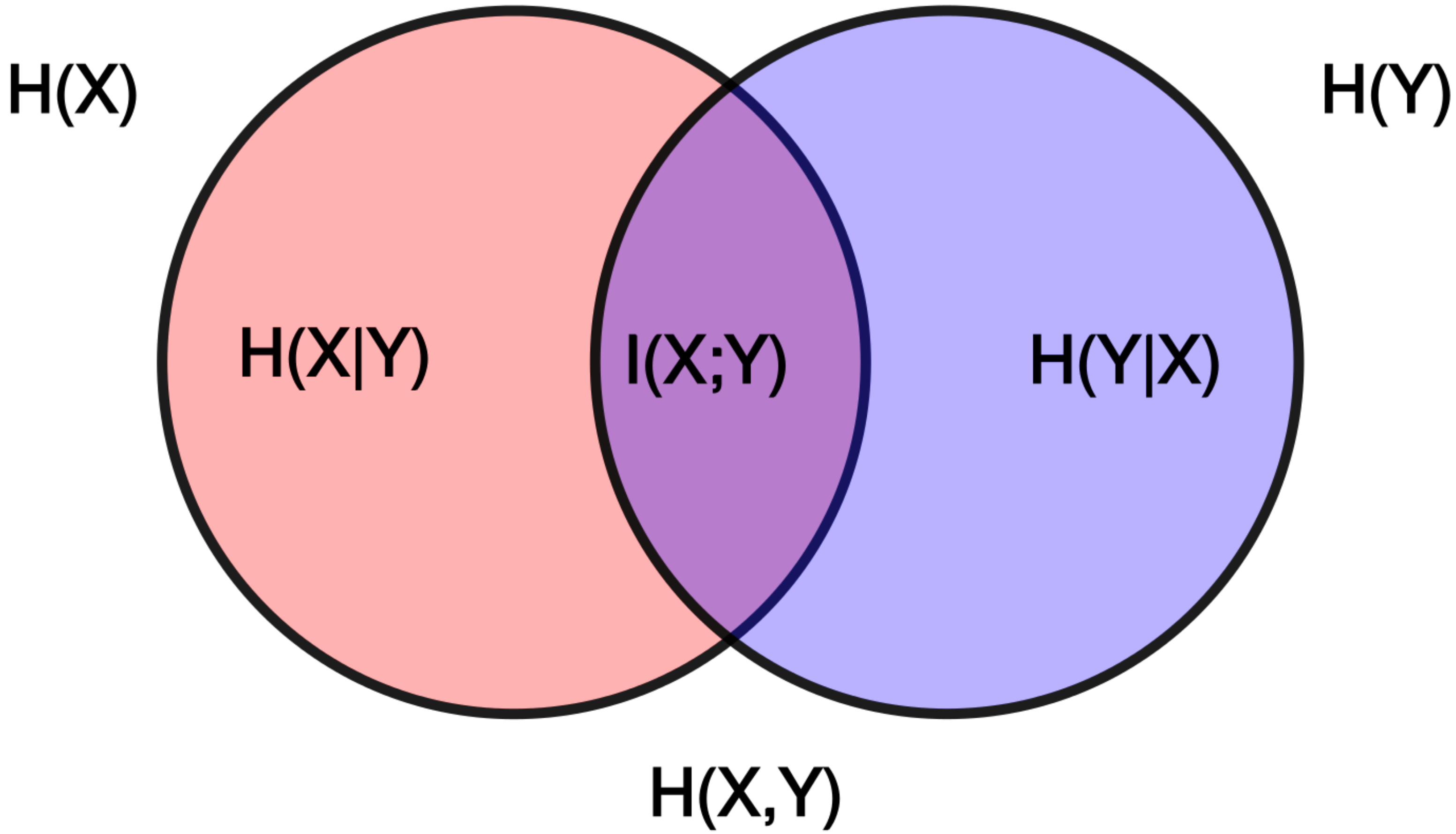
Final variable







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



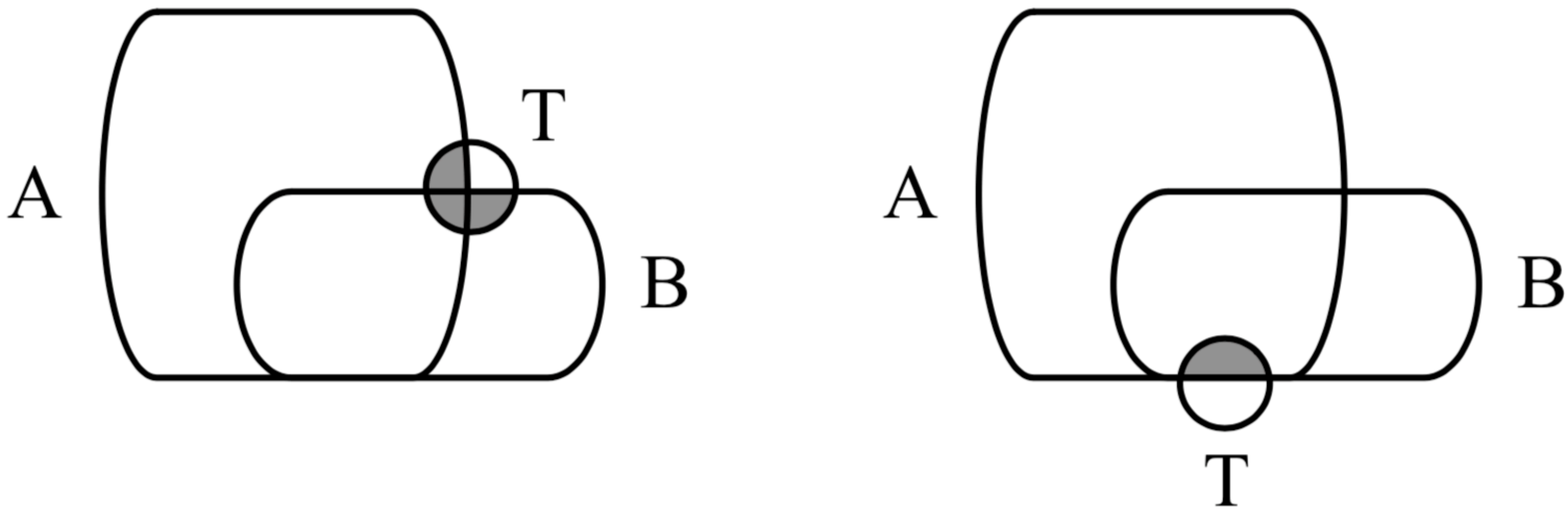
Mutual Information

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$ *discrete*

$$I(X;Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) dx\,dy$$ *continuous*

$$I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

Mutual Information in Physics

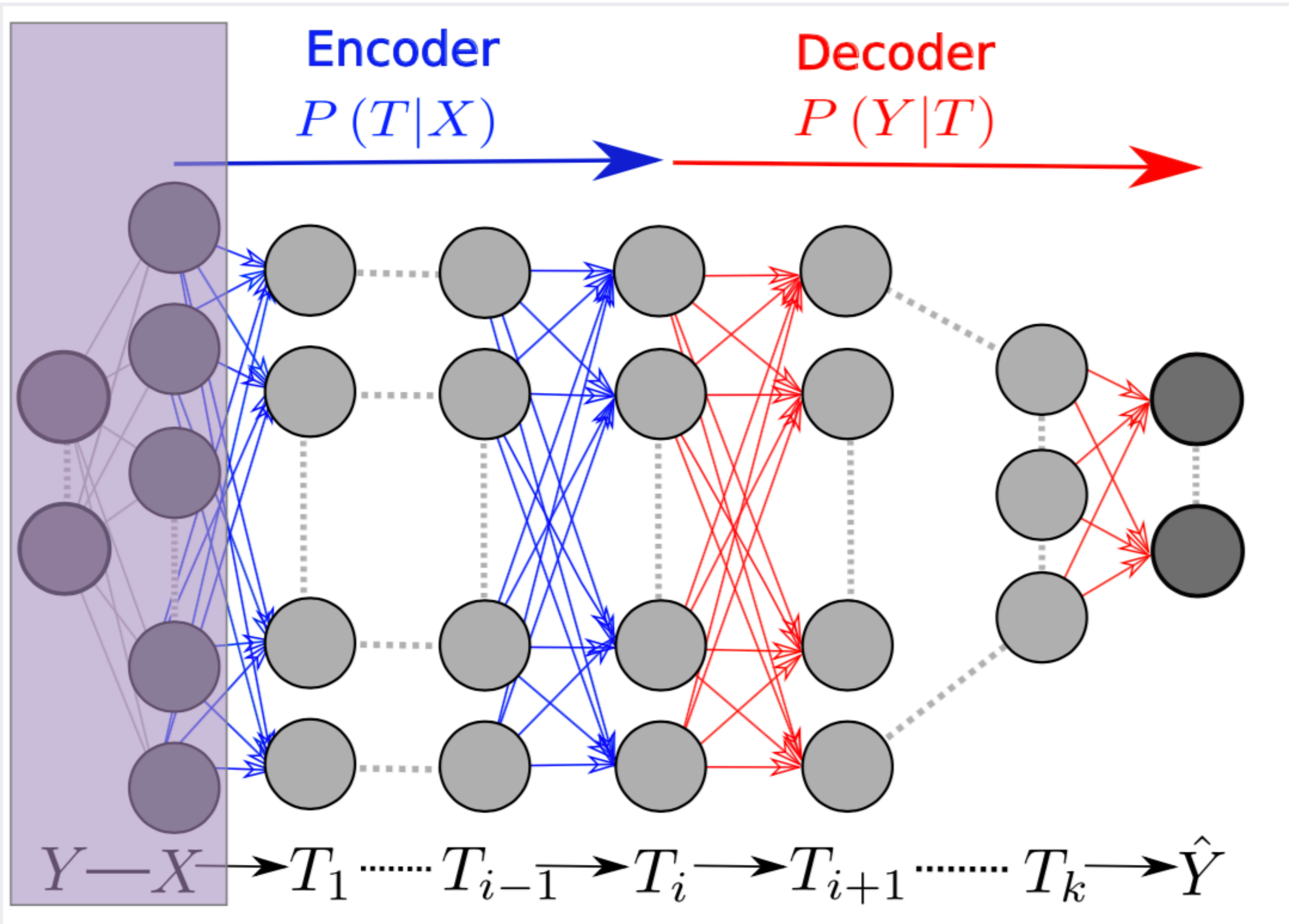


Usually care about mutual information of a variable with Truth

Related to ROC curve

Gaining (Mutual) Information about Quark/Gluon Discrimination
AJ Larkoski, J Thaler, WJ Waalewijn
JHEP11(2014)129

Information Plane



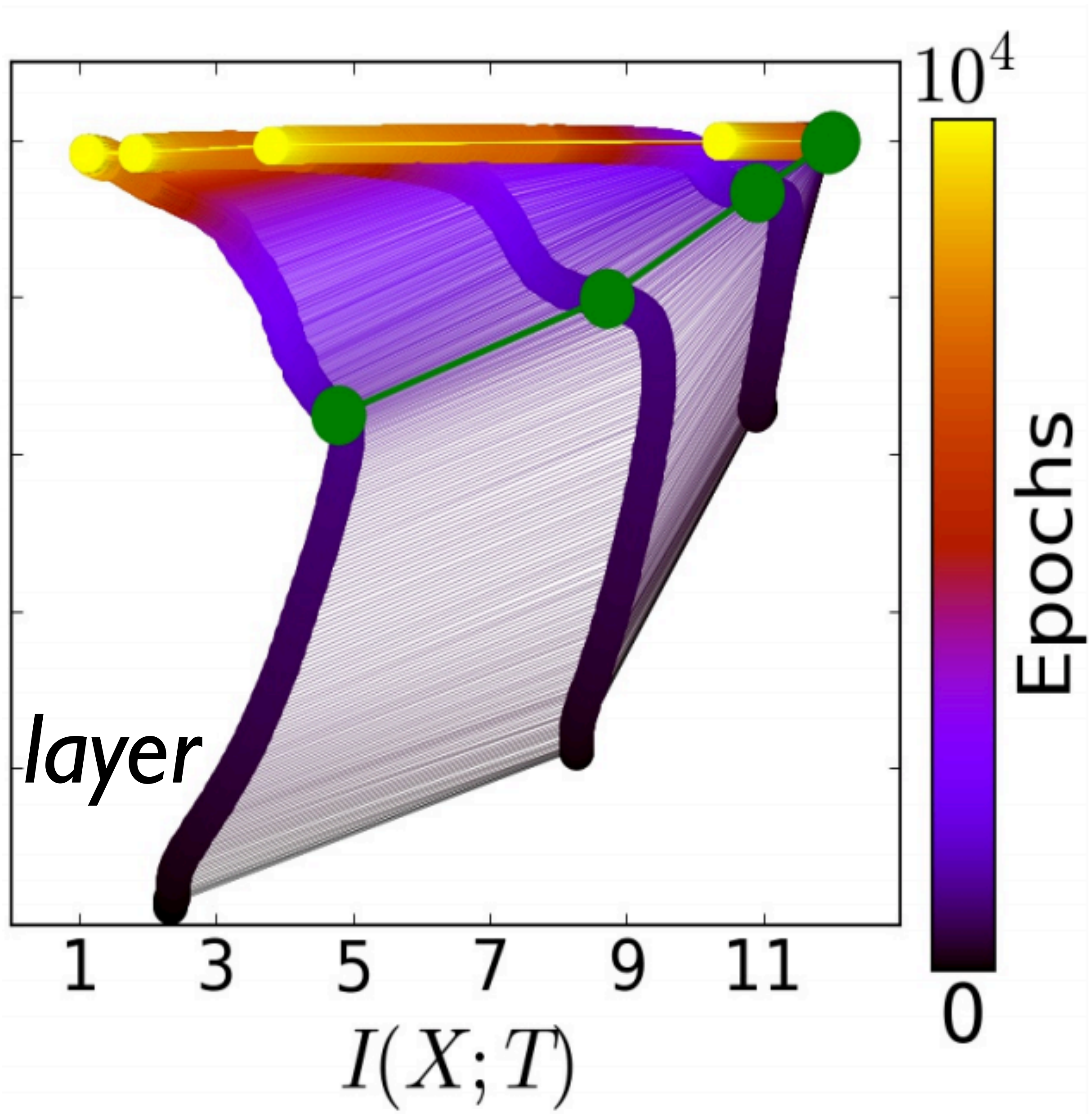
- True Value: Y
- Input: X
- Representation: T

$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq \ldots \geq I(T_k;Y) \geq I(\hat{Y};Y)$$

$$I(X;Y) = I(\psi(X);\phi(Y))$$

On the Information Bottleneck Theory of Deep Learning, Saxe et al, ICLR Proc 2018
Opening the Black Box of Deep Neural Networks via Information
Ravid Shwartz-Ziv, Naftali Tishby
1703.00810

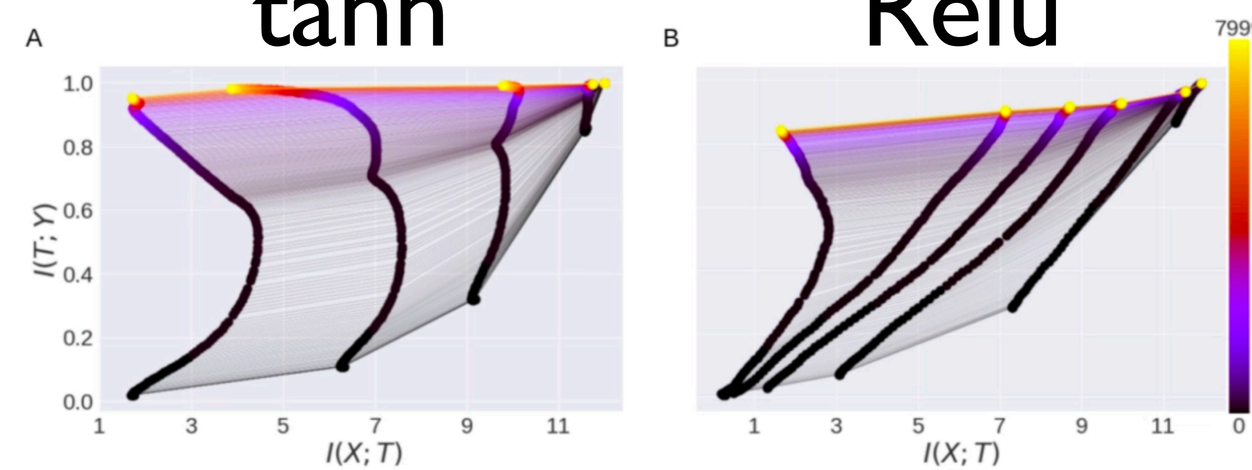
Information Plane



last layer

Two phases?

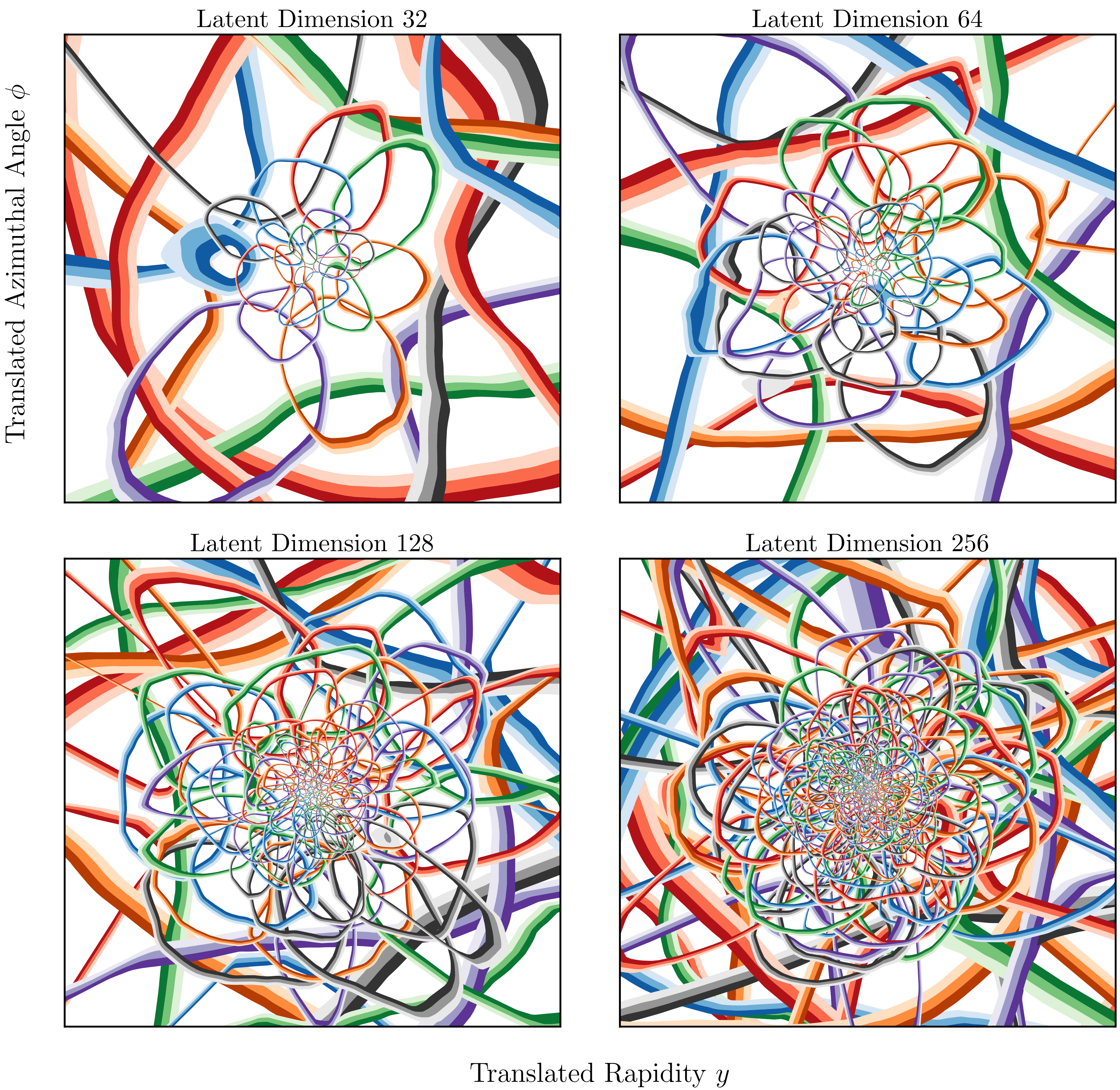
tanh Relu

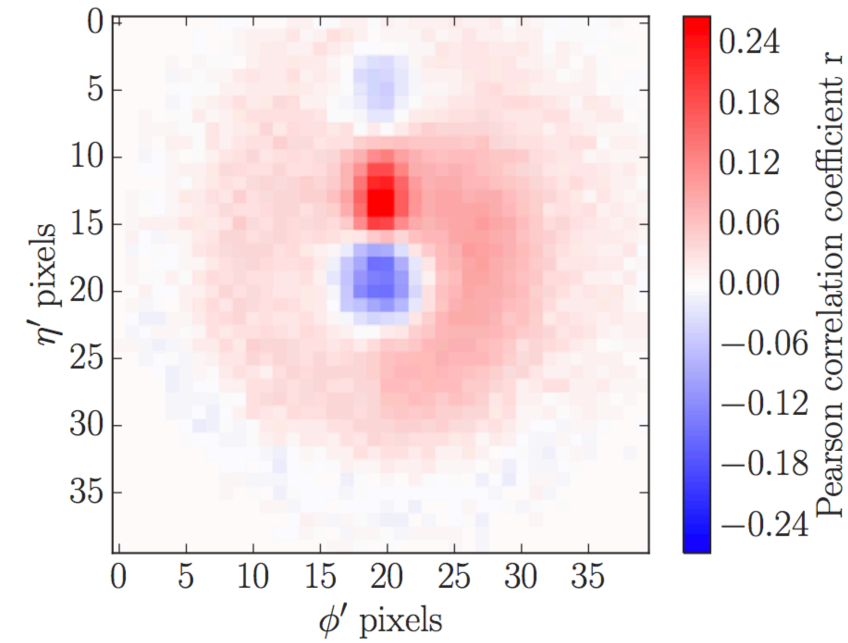


Compression = Generalisation?

On the Information Bottleneck Theory of Deep Learning, Saxe et al, ICLR Proc 2018

Opening the Black Box of Deep Neural Networks via Information
Ravid Shwartz-Ziv, Naftali Tishby
1703.00810


Latent Dimension 32
Latent Dimension 64
Latent Dimension 128
Latent Dimension 256
Translated Azimuthal Angle ϕ
Translated Rapidity y


η' pixels
ϕ' pixels
0
5
10
15
20
25
30
35
0.24
0.18
0.12
0.06
0.00
−0.06
−0.12
−0.18
−0.24
Pearson correlation coefficient r

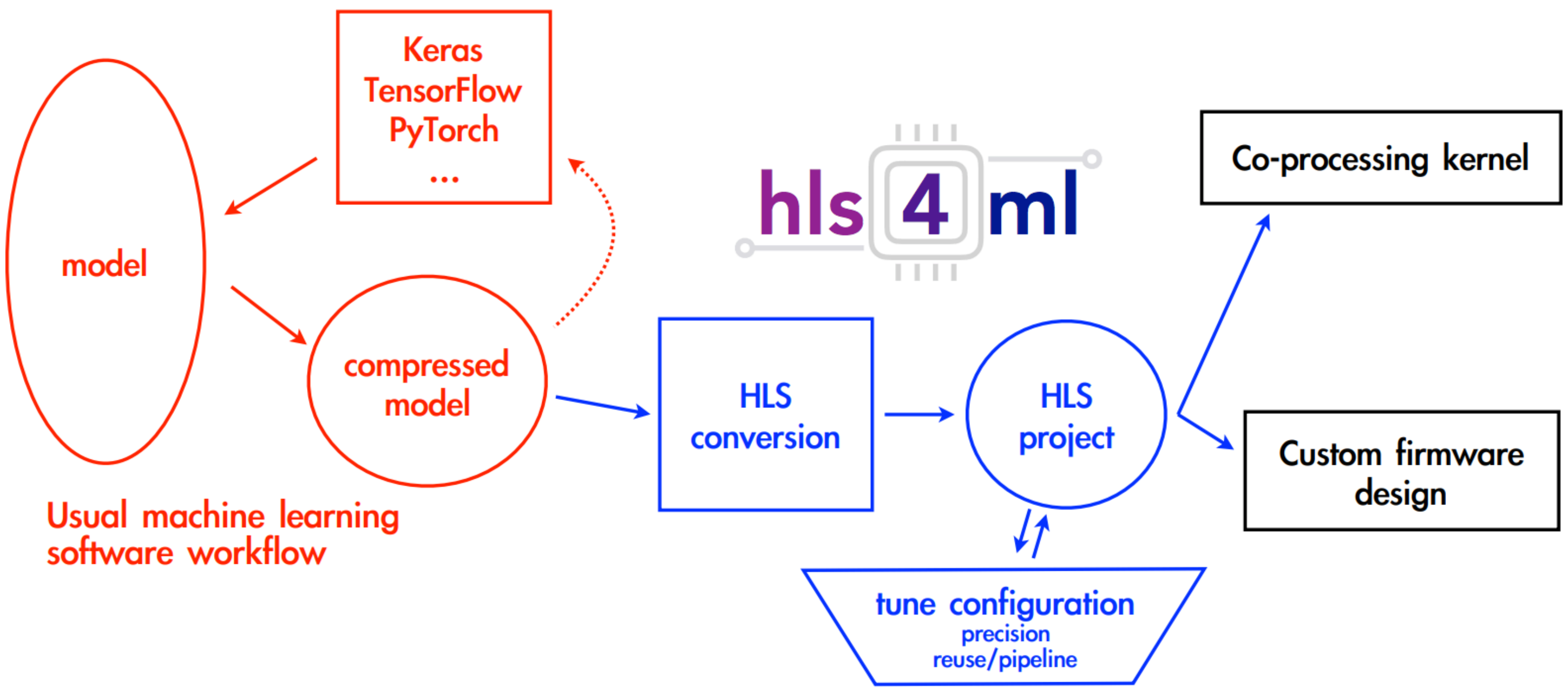
Deep Dream

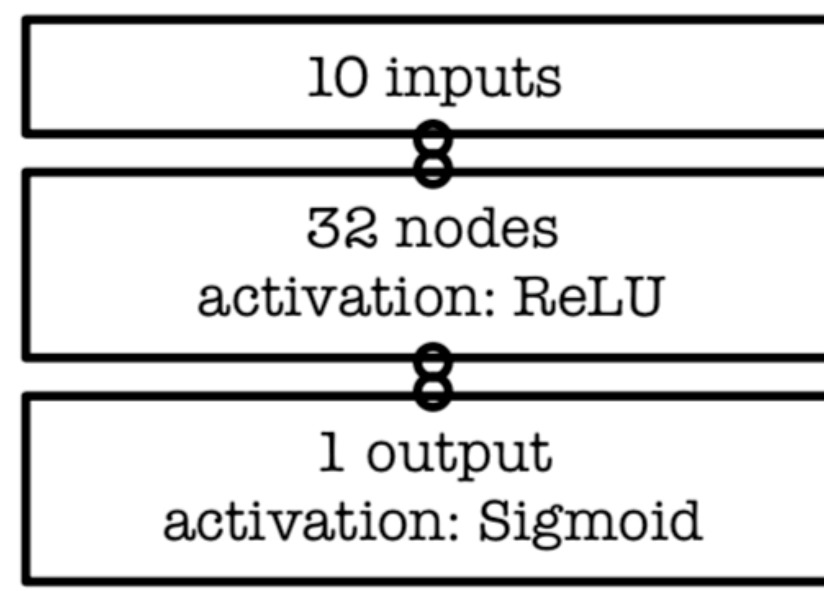
DeepDream: Slighly modify image to increase classification score. Highlight the features the network learned

FPGA DNN Triggers

Observables

m_{mMDT}
$N_2^{\beta=1,2}$
$M_2^{\beta=1,2}$
$C_1^{\beta=0,1,2}$
$C_2^{\beta=1,2}$
$D_2^{\beta=1,2}$
$D_2^{(\alpha,\beta)=(1,1),(1,2)}$
$\sum z \log z$
Multiplicity





Fast inference of deep neural networks in FPGAs for particle physics
J Duarte et al
1804.06913

- Framework to translate NNs to FPGAs for fast (L1 trigger) execution
- Latency of 75-150 ns

The End.