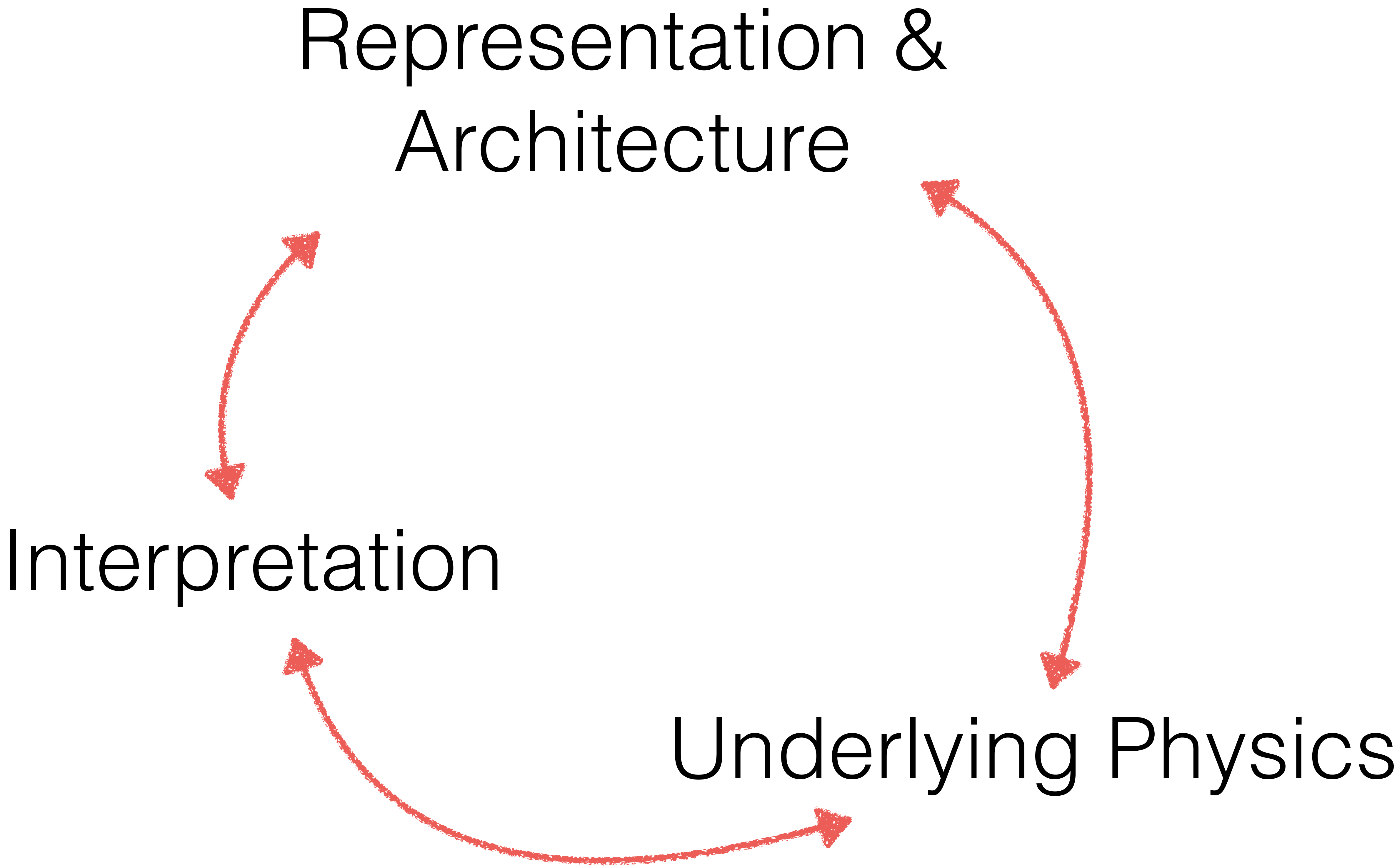
Recursive Neural Networks in Jet Tagging at the LHC

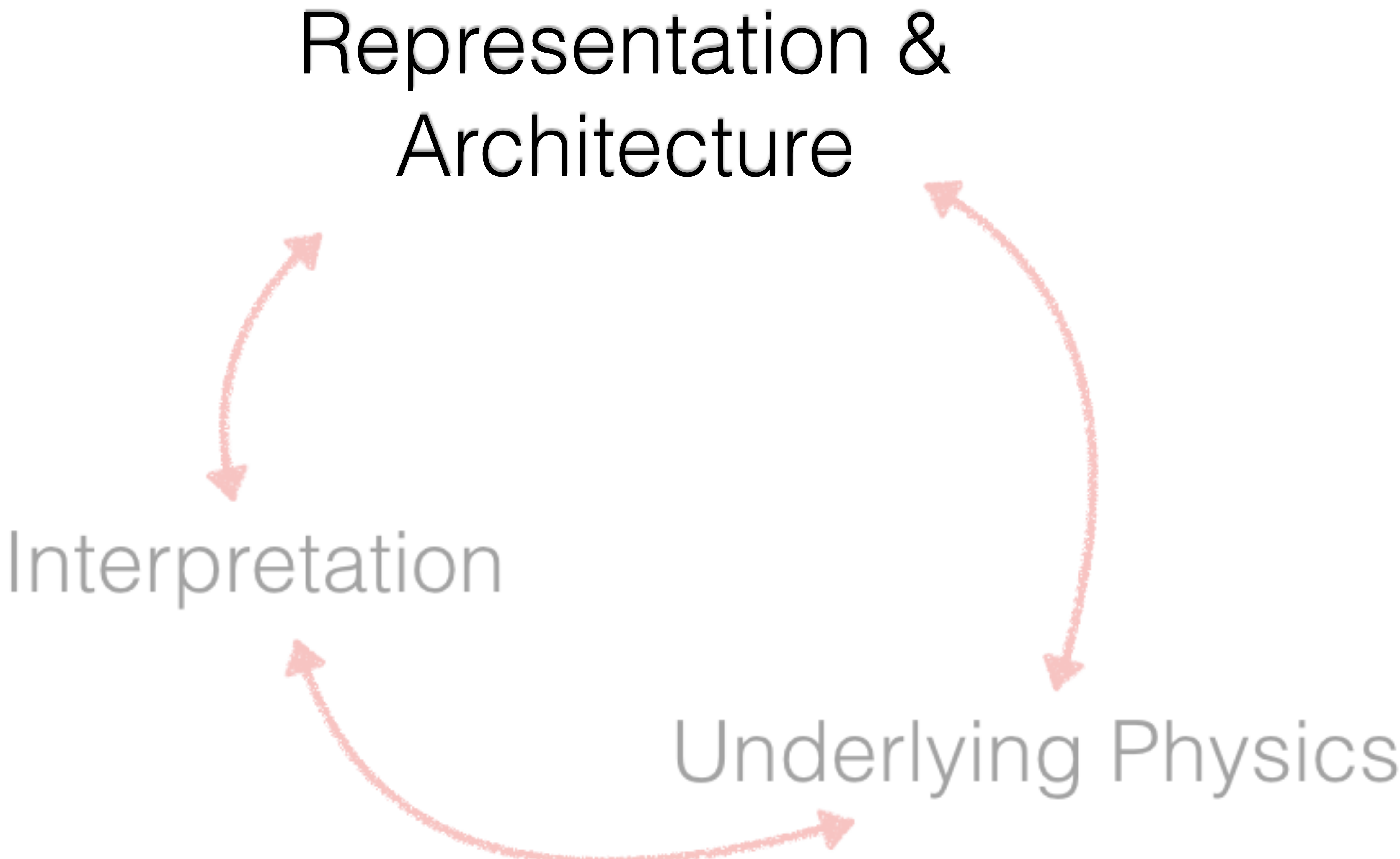
Taoli Cheng

University of Chinese Academy of Sciences

July 19, 2018 @ **BOOST 2018**

partly based on [T. Cheng, arXiv:1711.02633]


Representation &
Architecture
Interpretation
Underlying Physics


Representation &
Architecture
Interpretation
Underlying Physics

Jet Representations ⟷ Analysis Tools

Two key choices when tagging jets

How to represent the jet

- Single expert variable
- A few expert variables
- Many expert variables

- Jet images
- List of particles
- Clustering tree
- *N*-subjettiness basis
- Energy flow polynomials
- Set of particles

→

How to analyze that representation

- Threshold cut
- Multidimensional likelihood
- Boosted decision tree (BDT), shallow neural network (NN)
- Convolutional NN (CNN)
- Recurrent/Recursive NN (RNN)
- Fancy RNN
- Deep neural network (DNN)
- Linear classification
- Energy flow network

See Ben Nachman's intro talk for more

Jet Representations ⟷ Analysis Tools

Two key choices when tagging jets

How to represent the jet

- Single expert variable
- A few expert variables
- Many expert variables

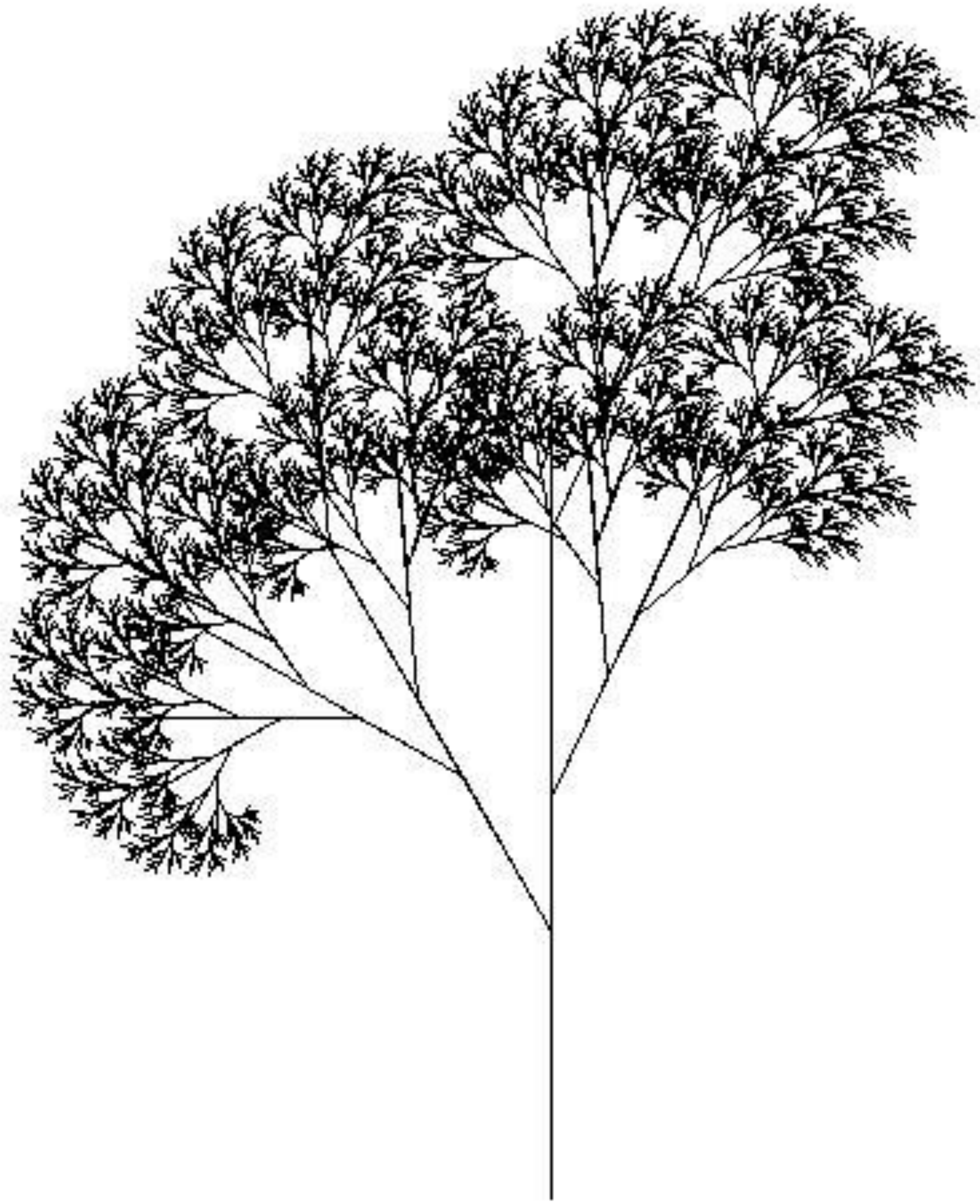
- Jet images
- List of particles
- Clustering tree
- N-subjettiness basis
- Energy flow polynomials
- Set of particles

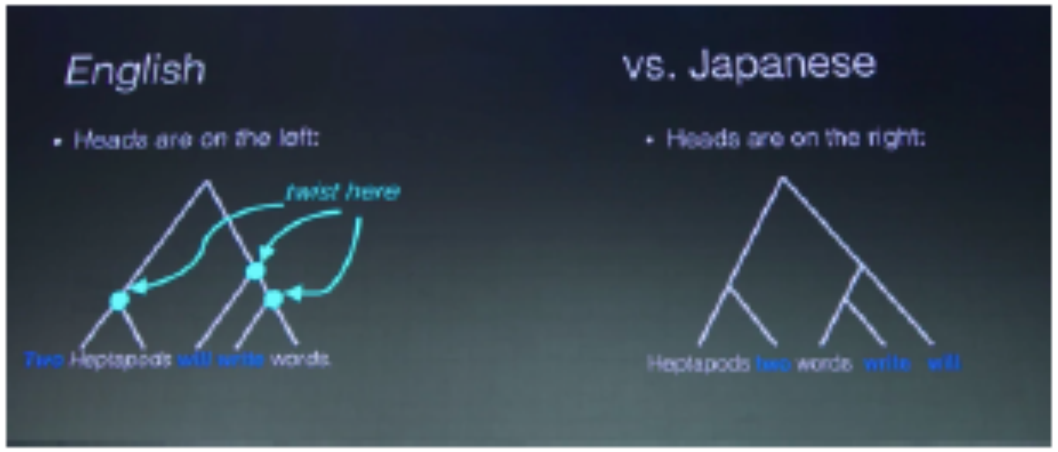
How to analyze that representation

- Threshold cut
- Multidimensional likelihood
- Boosted decision tree (BDT), shallow neural network (NN)
- Convolutional NN (CNN)
- Recurrent/Recursive NN (RNN)
- Fancy RNN
- Deep neural network (DNN)
- Linear classification
- Energy flow network

See Ben Nachman's intro talk for more

WHY RECURSIVE NEURAL NETWORKS(RECNN)?





[by linguist Jessica Coon]

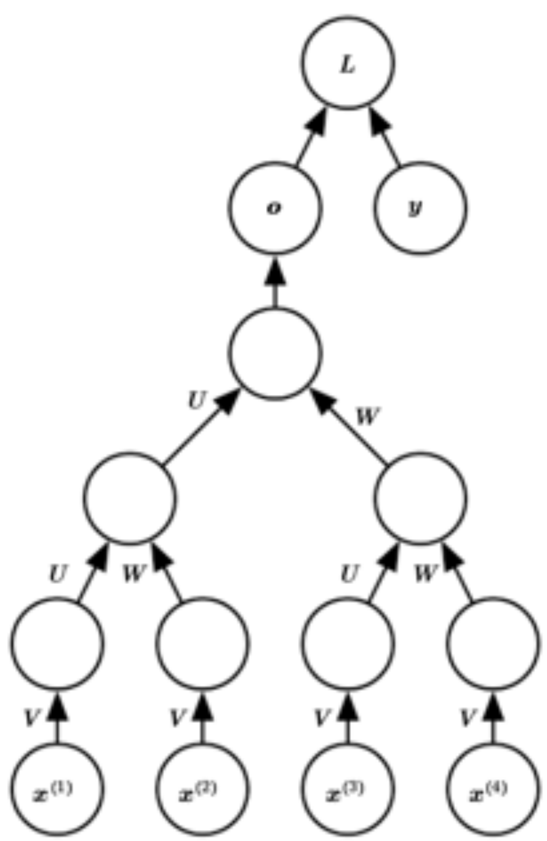
Natural Language Structure

Jet Clustering Tree

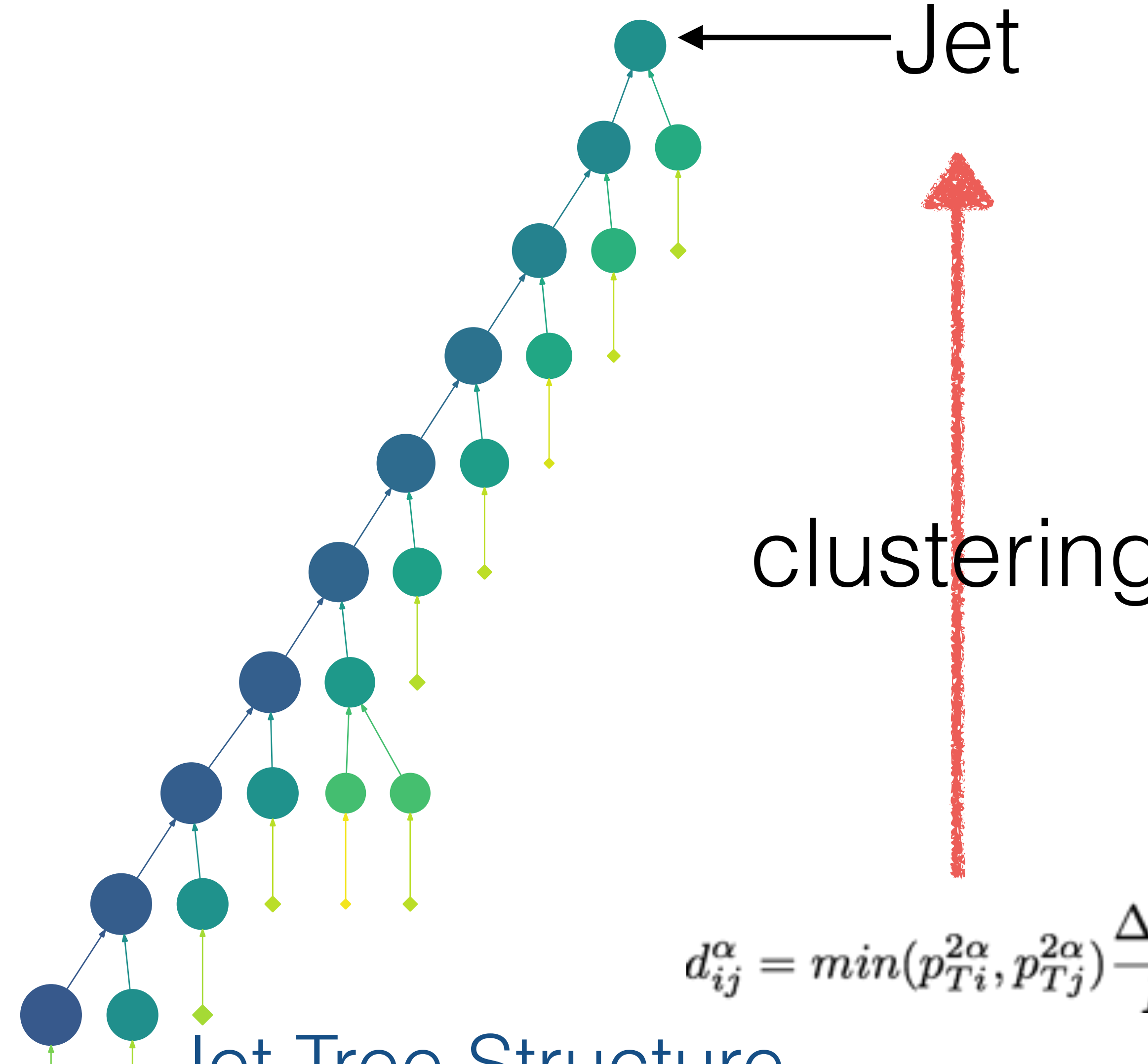
Recursive Neural Networks for Jets

Motivated by:

- problems in image approach: sparsity of jet images (5% - 10% active), fixed image size, (information loss from pixelization)
- natural tree-like structure of sequential jet clustering history
- implementation in event-level



Recursive Neural Nets (RecNN)



$$d_{ij}^{\alpha} = min(p_{Ti}^{2\alpha}, p_{Tj}^{2\alpha}) \frac{\Delta R_{ij}^2}{R^2}$$

Jet Tree Structure

RecNN for Jets

Motivated by:

- problems in image approach: sparsity of jet images (5% - 10% active), fixed image size, (information loss from pixelization)
- natural tree-like structure of sequential jet clustering history
- implementation in event-level

[G. Louppe, K. Cho, C. Becot, K. Cranmer, arXiv: 1702.00748]

QCD-Aware Recursive Neural Networks for Jet Physics

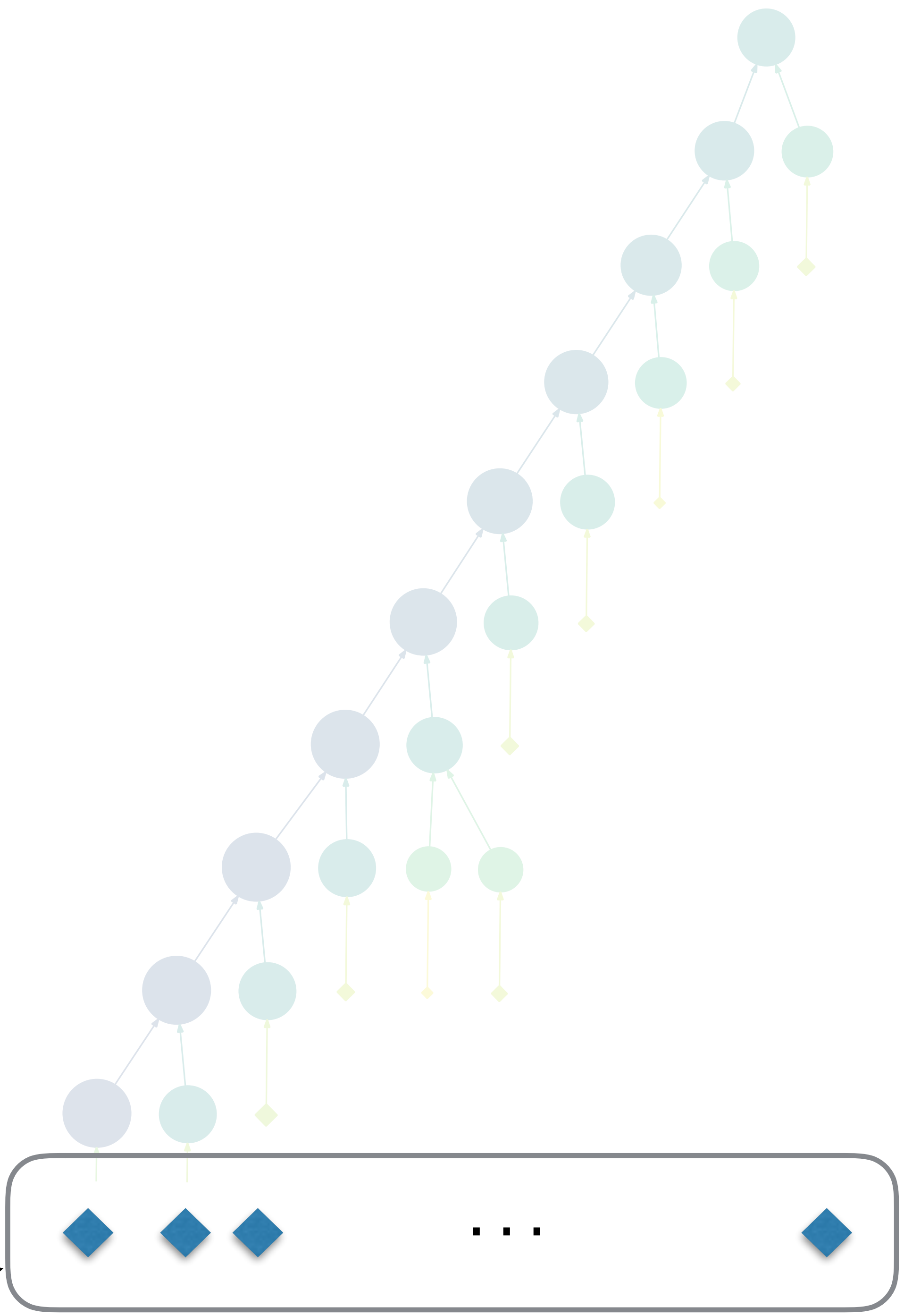
Gilles Louppe,[1] **Kyunghyun Cho**,[1] **Cyril Becot**,[1] and **Kyle Cranmer**[1]

[1] *New York University*

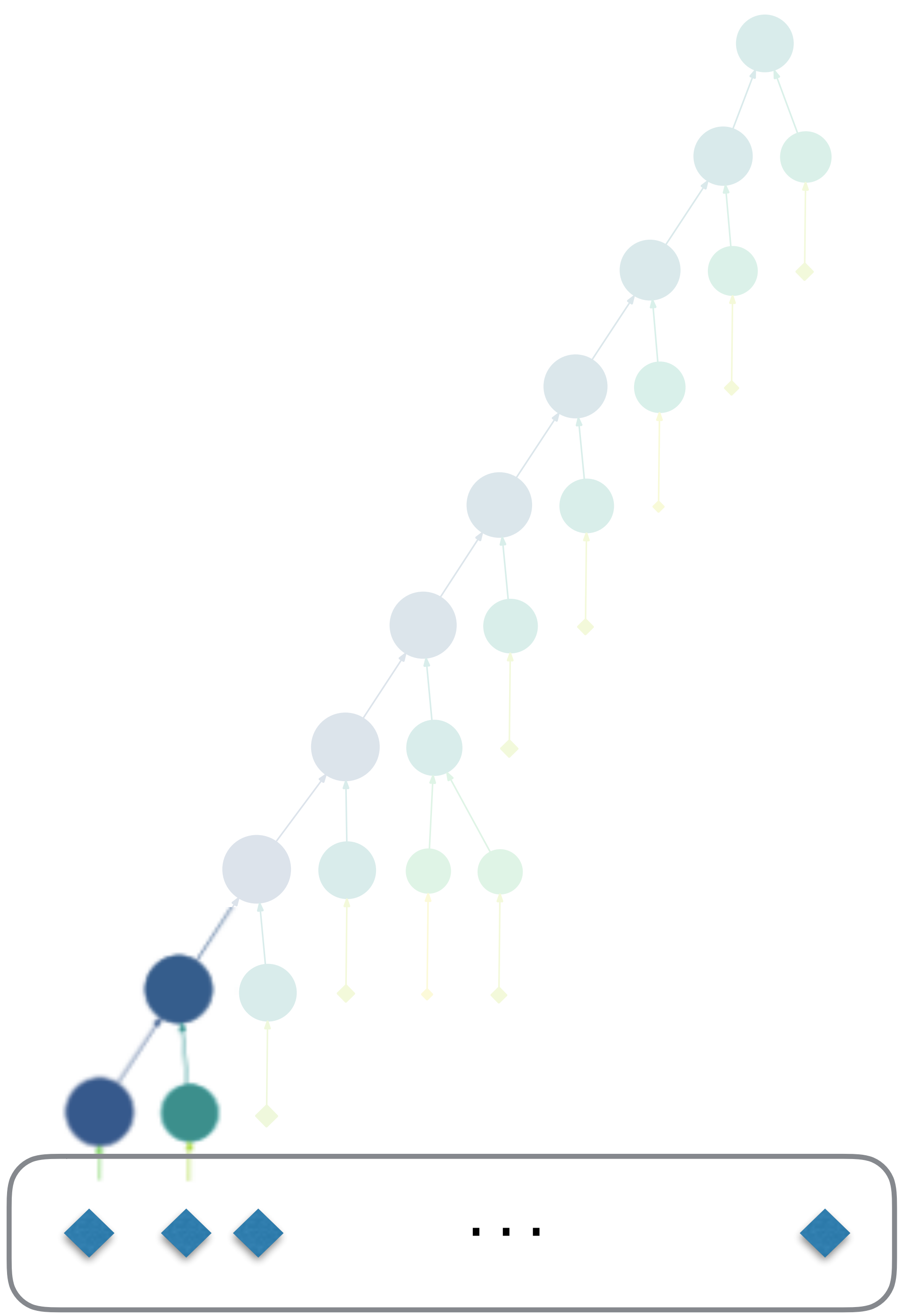
Recent progress in applying machine learning for jet physics has been built upon an analogy between calorimeters and images. In this work, we present a novel class of recursive neural networks built instead upon an analogy between QCD and natural languages. In the analogy, four-momenta are like words and the clustering history of sequential recombination jet algorithms is like the parsing of a sentence. Our approach works directly with the four-momenta of a variable-length set of particles, and the jet-based tree structure varies on an event-by-event basis. Our experiments highlight the flexibility of our method for building task-specific jet embeddings and show that recursive architectures are significantly more accurate and data efficient than previous image-based networks. We extend the analogy from individual jets (sentences) to full events (paragraphs), and show for the first time an event-level classifier operating on all the stable particles produced in an LHC event.

Jet Clustering

$$d_{ij}^{\alpha} = min(p_{Ti}^{2\alpha}, p_{Tj}^{2\alpha})\frac{\Delta R_{ij}^2}{R^2}$$



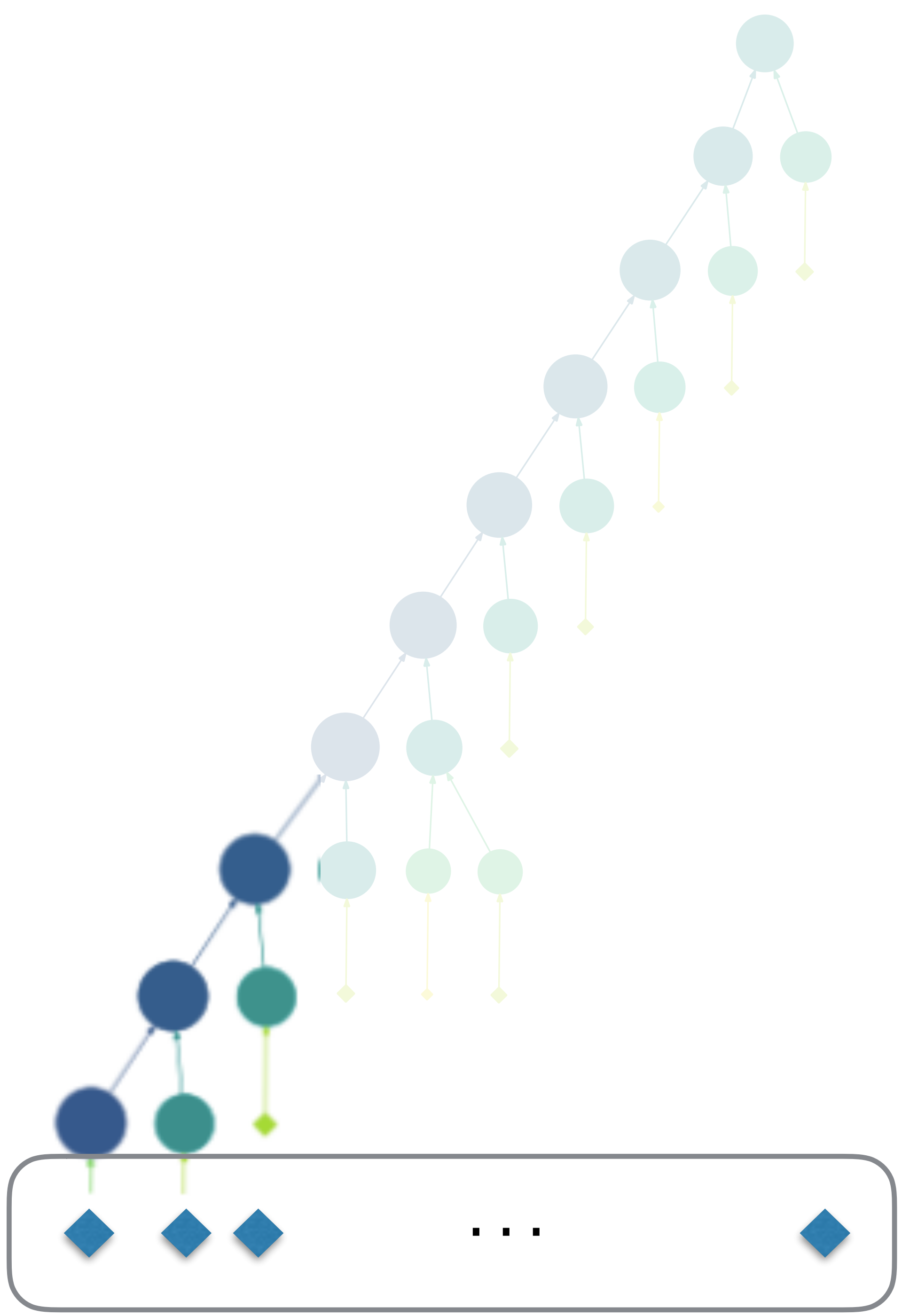
Jet Clustering



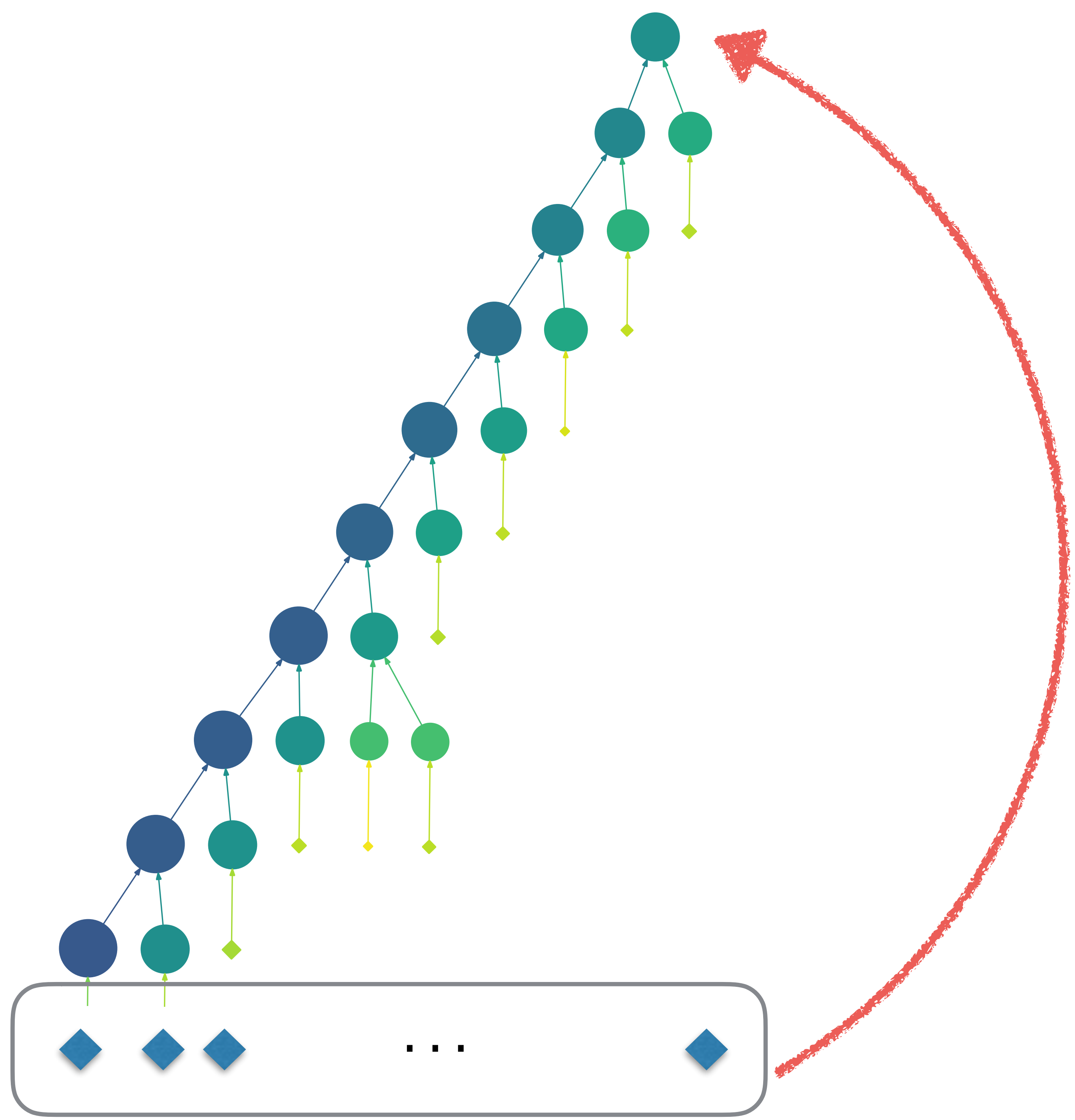
$$d_{ij}^{\alpha} = min(p_{Ti}^{2\alpha}, p_{Tj}^{2\alpha}) \frac{\Delta R_{ij}^2}{R^2}$$

Jet Clustering

$$d_{ij}^{\alpha} = min(p_{Ti}^{2\alpha}, p_{Tj}^{2\alpha}) \frac{\Delta R_{ij}^2}{R^2}$$



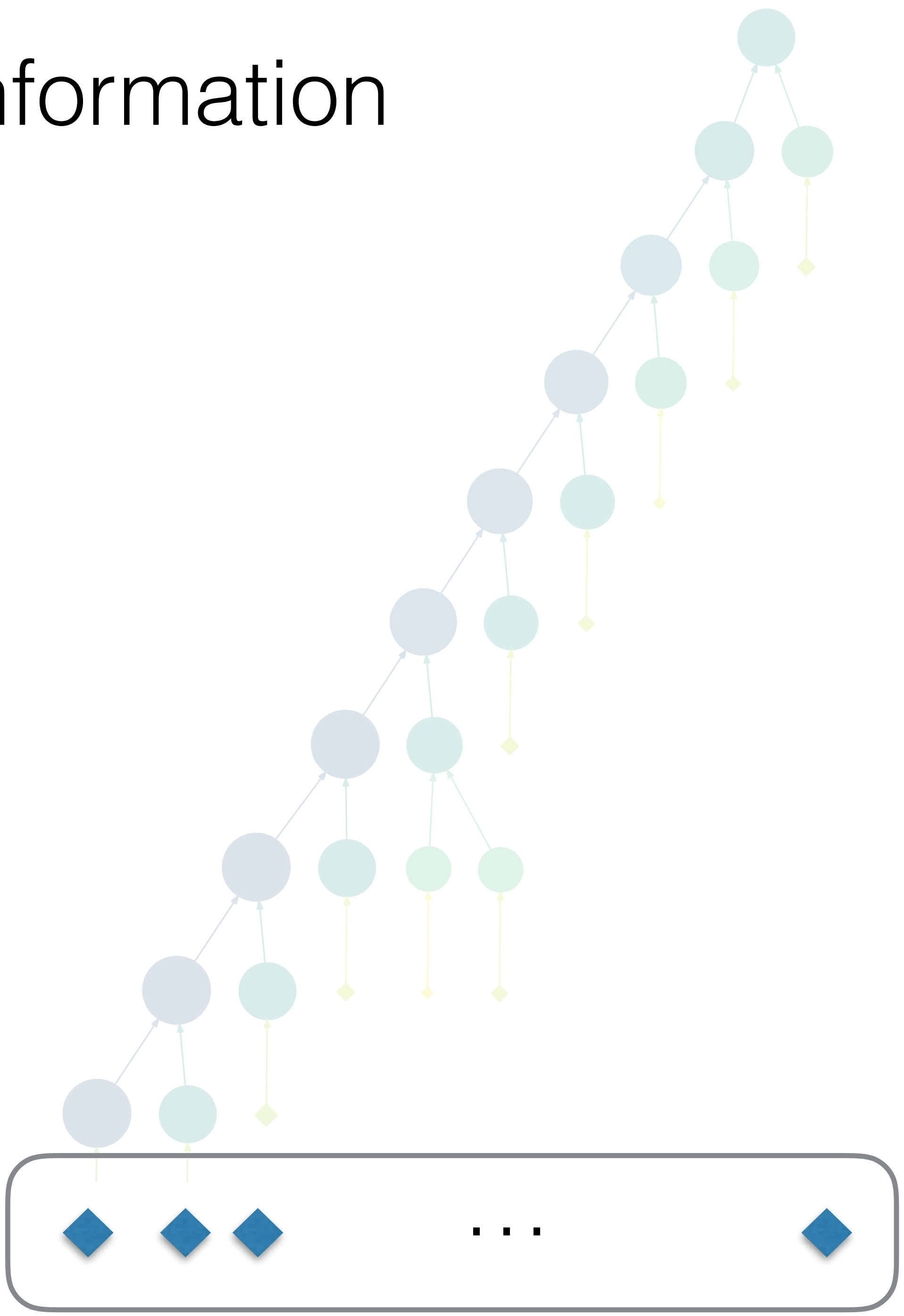
Jet Clustering



$$d_{ij}^{\alpha} = min(p_{Ti}^{2\alpha}, p_{Tj}^{2\alpha})\frac{\Delta R_{ij}^2}{R^2}$$

Recursive Neural Networks for Jets

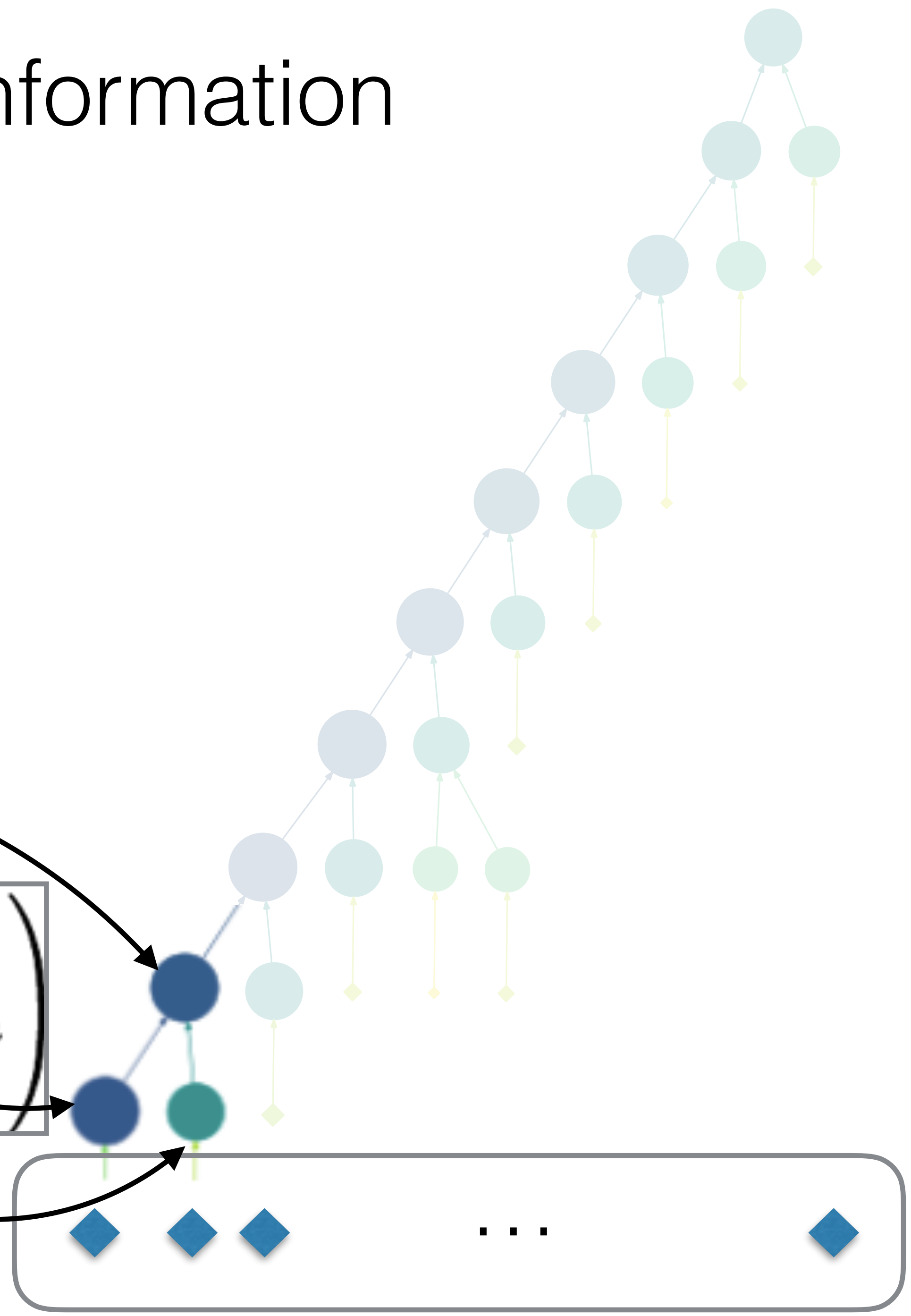
embedding jet information



Recursive Neural Networks for Jets

embedding jet information

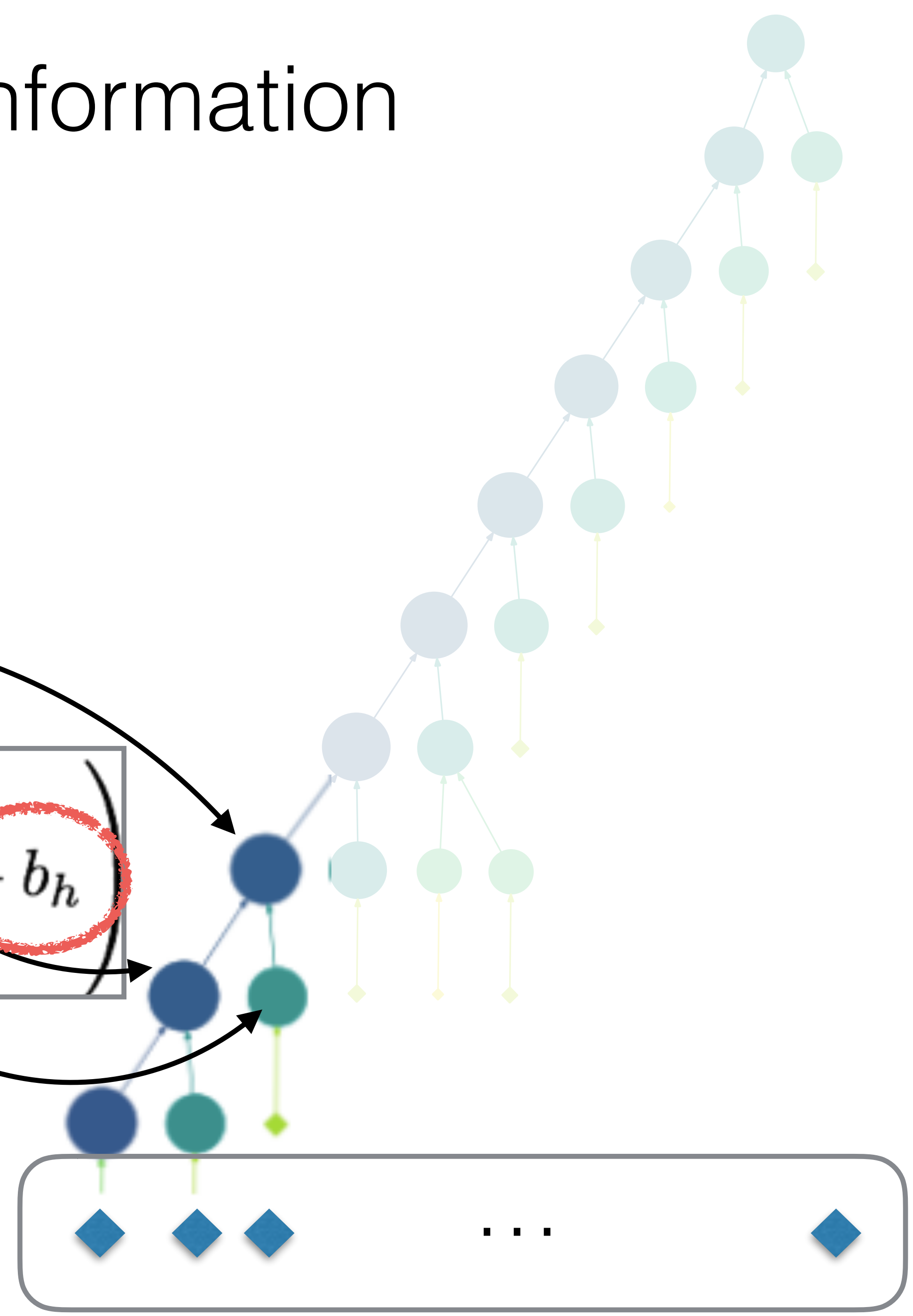
$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \\ \mathbf{u}_k \end{bmatrix} + b_h \right)$$



Recursive Neural Networks for Jets

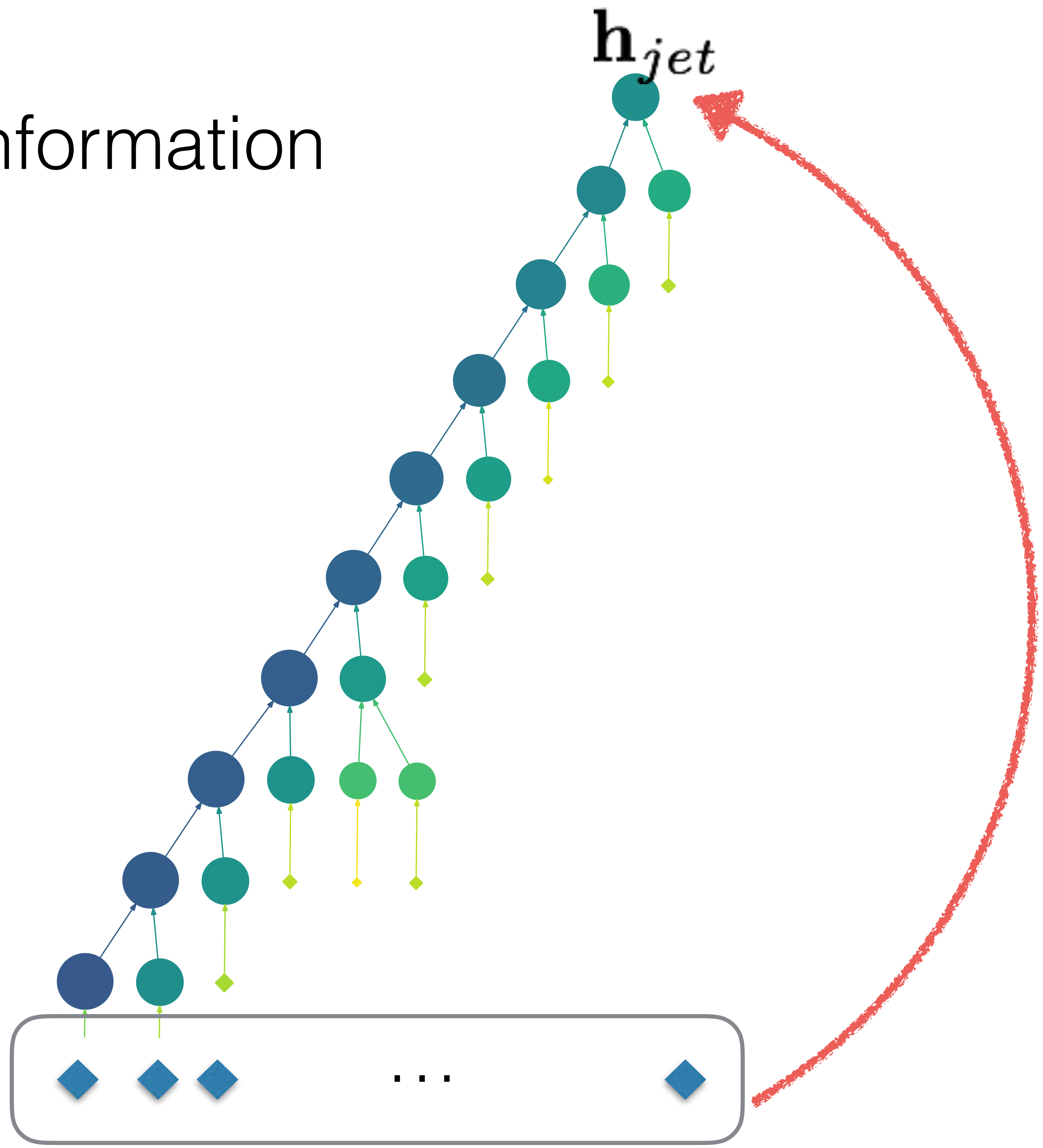
embedding jet information

$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \\ \mathbf{u}_k \end{bmatrix} + b_h \right)$$



Recursive Neural Networks for Jets

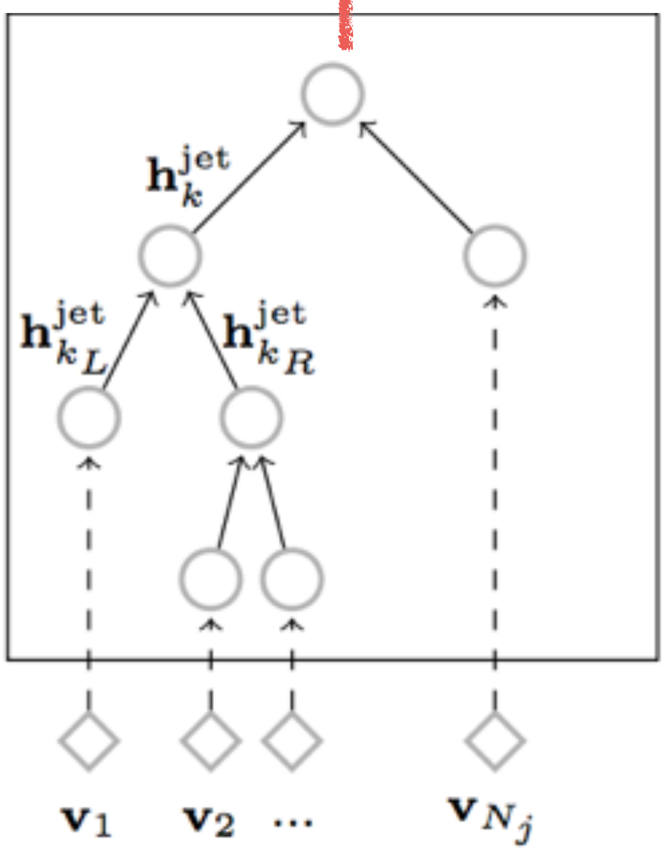
embedding jet information



RecNN & Jet Embedding

$\mathbf{h}^{\mathrm{Jet}}$

Classifier



(taken from G. Louppe, K. Cho, C. Becot, K. Cranmer, arXiv: 1702.00748)

$R^f \to R^q$

$$\boxed{\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}_{k_L}^{\mathrm{jet}} \\ \mathbf{h}_{k_R}^{\mathrm{jet}} \\ \mathbf{u}_k \end{bmatrix} + b_h\right)} \quad \mathbf{u}_k = \sigma(W_u \mathbf{x}_k + b_u)$$

$(p_T, \eta, \phi, \ldots)_i$ and particle flow identification

- One hot vector $((i_{\text{neutral hadron}}, i_{\text{photon}}, i_+, i_-), i = 0 \text{ or } 1)$
- pt-weighted charge $Q_k^{\mathrm{rec}} = \dfrac{Q_{k_L}^{\mathrm{rec}}(p_T^{k_L})^\kappa + Q_{k_R}^{\mathrm{rec}}(p_T^{k_R})^\kappa}{(p_T^k)^\kappa}$

* with recursively defined pt-weighed charge, we can include the particle flow information in one variable which is well defined for all the nodes

Workflow

Pythia8

↓

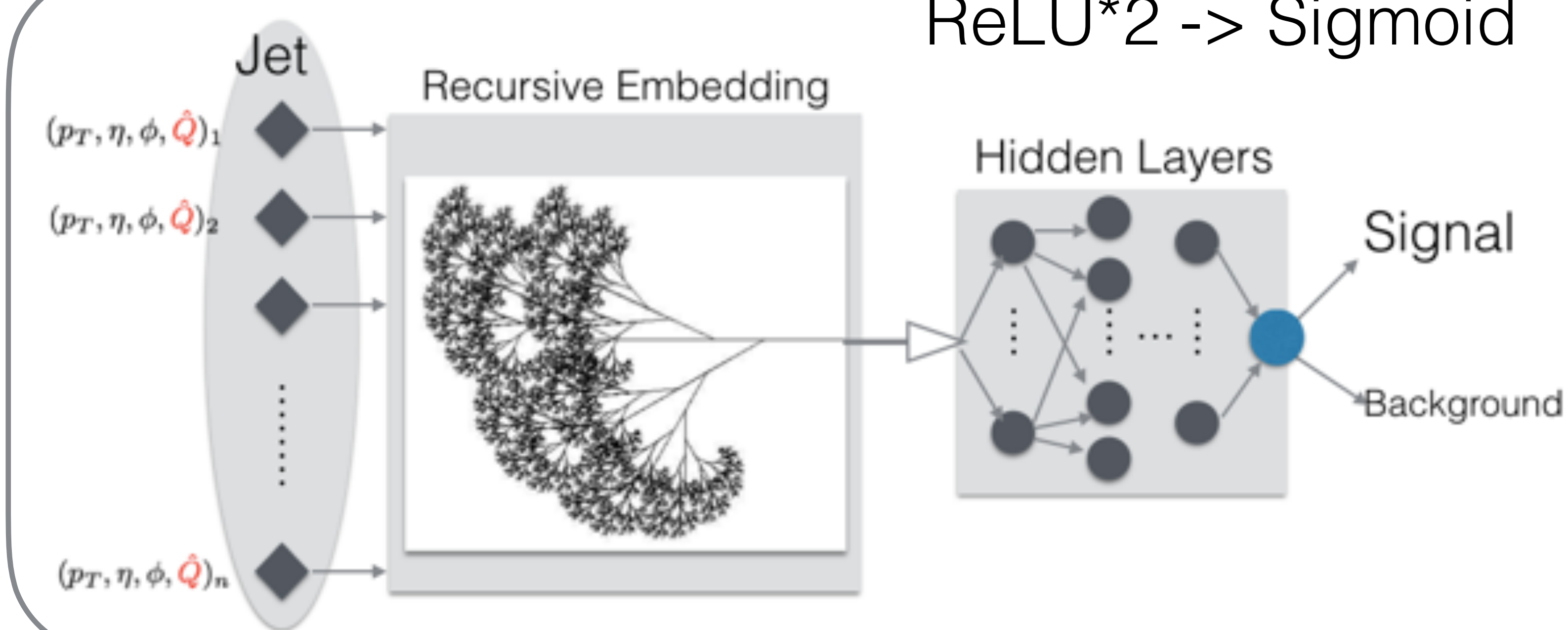
Delphes

↓

Data Preprocessing →

RecNN

ReLU*2 -> Sigmoid



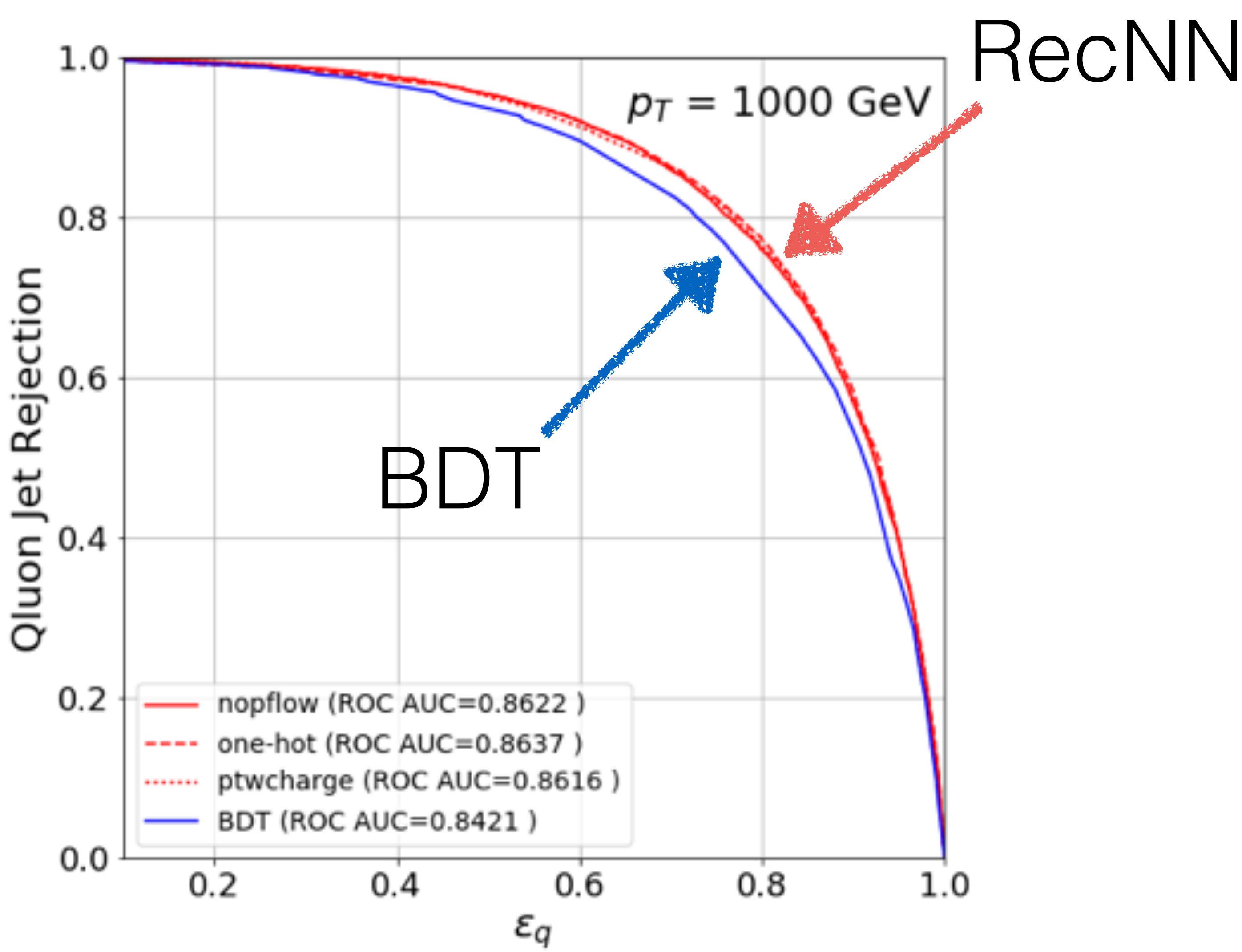
- Measure: Receiver Operating Characteristic (ROC), Area Under the Curve (AUC) of ROC, background rejection rate @ $\epsilon_s = 50\%$
- Particle Flow Identification: one-hot vectors, or pt weighted charge

Quark/Gluon Discrimination

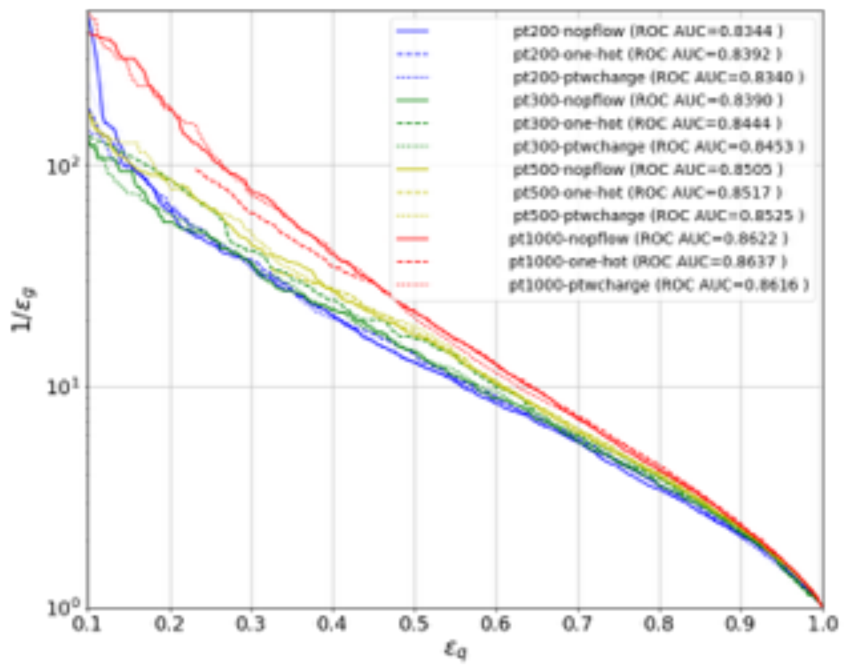
Baseline: BDT (jet mass m/p_T, jet girth $\sum_{i \in \mathrm{Jet}} \frac{p_T^i}{p_T^J} r_i$, charged particle count $\#_{\mathbf{charged}}$)

For RecNN,

- no particle flow identification
- one-hot vectors
- pt-weighted charge instead

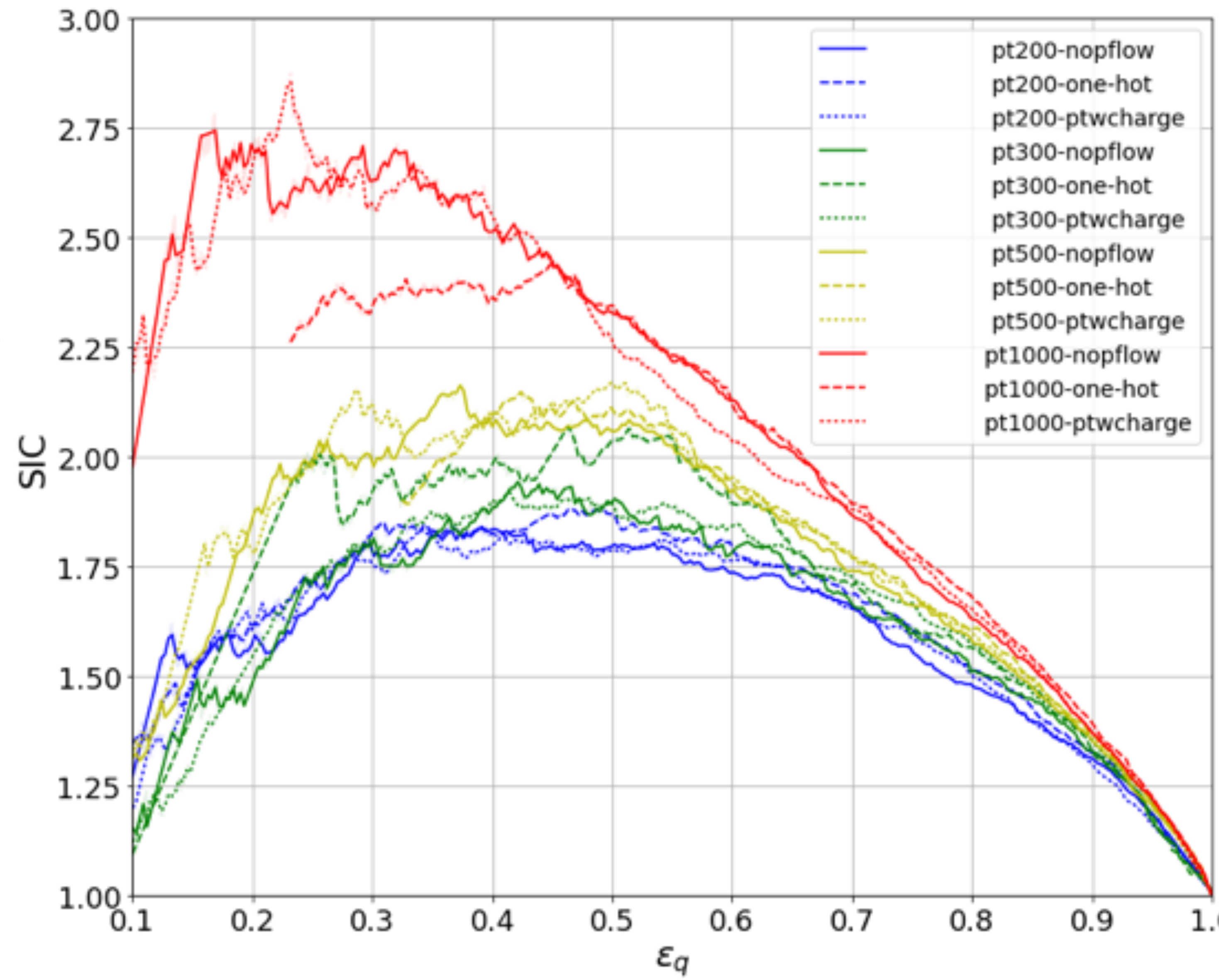


Quark/Gluon Discrimination



$$\sigma \equiv \frac{S}{\sqrt{B}} \to \frac{\epsilon_S S}{\sqrt{\epsilon_B B}} = \left(\frac{\epsilon_S}{\sqrt{\epsilon_B}}\right)\sigma \to \mathrm{SI} = \frac{\epsilon_S}{\sqrt{\epsilon_B}}$$

Significance Improvement



Jet pts: 200, 300, 500, 1000 GeV

Variants

$$\mathbf{h}_k = \sigma \left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \\ \mathbf{u}_k \end{bmatrix} + b_h \right)$$

Variants	AUC	$R_{\epsilon=50\%}$
Baseline	0.8344	12.9
R=0.7	0.8210	12.4
$W_h \to R^{q\times 2q}$	0.8268	12.3
$W_h \to R^{q\times 2q}$ with one-hot	0.8313	13.7
$\mathbf{x}$=(p_T, η, ϕ)	0.8291	11.8
$\mathbf{x}$=(η, ϕ)	0.8249	11.9
$\mathbf{x}$=(p_T)	0.8264	11.6
only one-hot	0.8255	11.9
$\mathbf{x}$=($Q^{\text{rec}}_{\kappa=50\%}$)	0.8234	11.3

Variants in input information

- particle flow identification doesn't help significantly
- the discriminating information for q/g tagging is RecNN mainly reside in the tree structure itself

Variants

$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \end{bmatrix} + b_h\right)$$

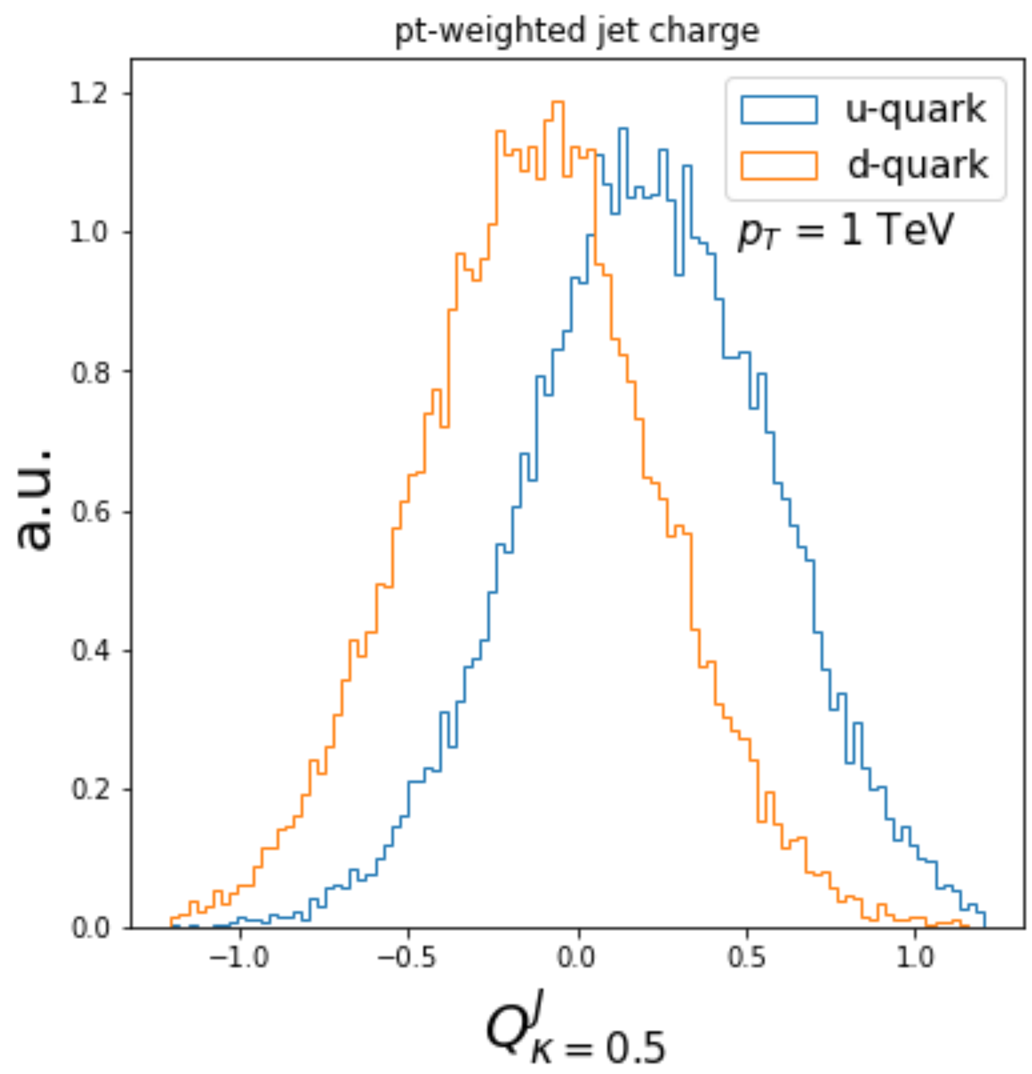
Variants in input information

Variants	AUC	$R_{\epsilon=50\%}$
Baseline	0.8344	12.9
R=0.7	0.8210	12.4
$W_h \to R^{q\times 2q}$	0.8268	12.3
$W_h \to R^{q\times 2q}$ with one-hot	0.8313	13.7
$\mathbf{x}=(p_T,\ \eta,\ \phi)$	0.8291	11.8
$\mathbf{x}=(\eta,\ \phi)$	0.8249	11.9
$\mathbf{x}=(p_T)$	0.8264	11.6
only one-hot	0.8255	11.9
$\mathbf{x}=(Q^{\text{rec}}_{\kappa=50\%})$	0.8234	11.3

- particle flow identification doesn’t help significantly
- the discriminating information for q/g tagging is RecNN mainly reside in the tree structure itself

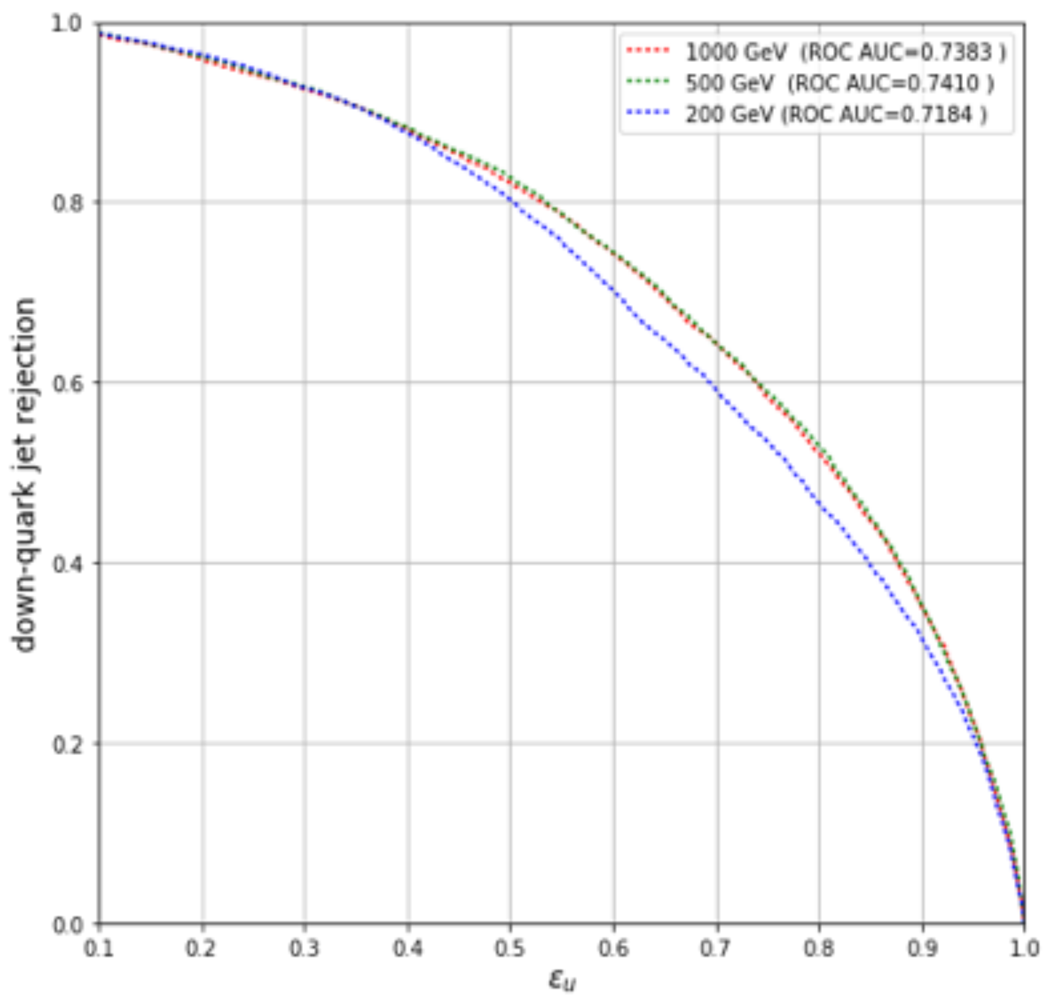
Jet Charge

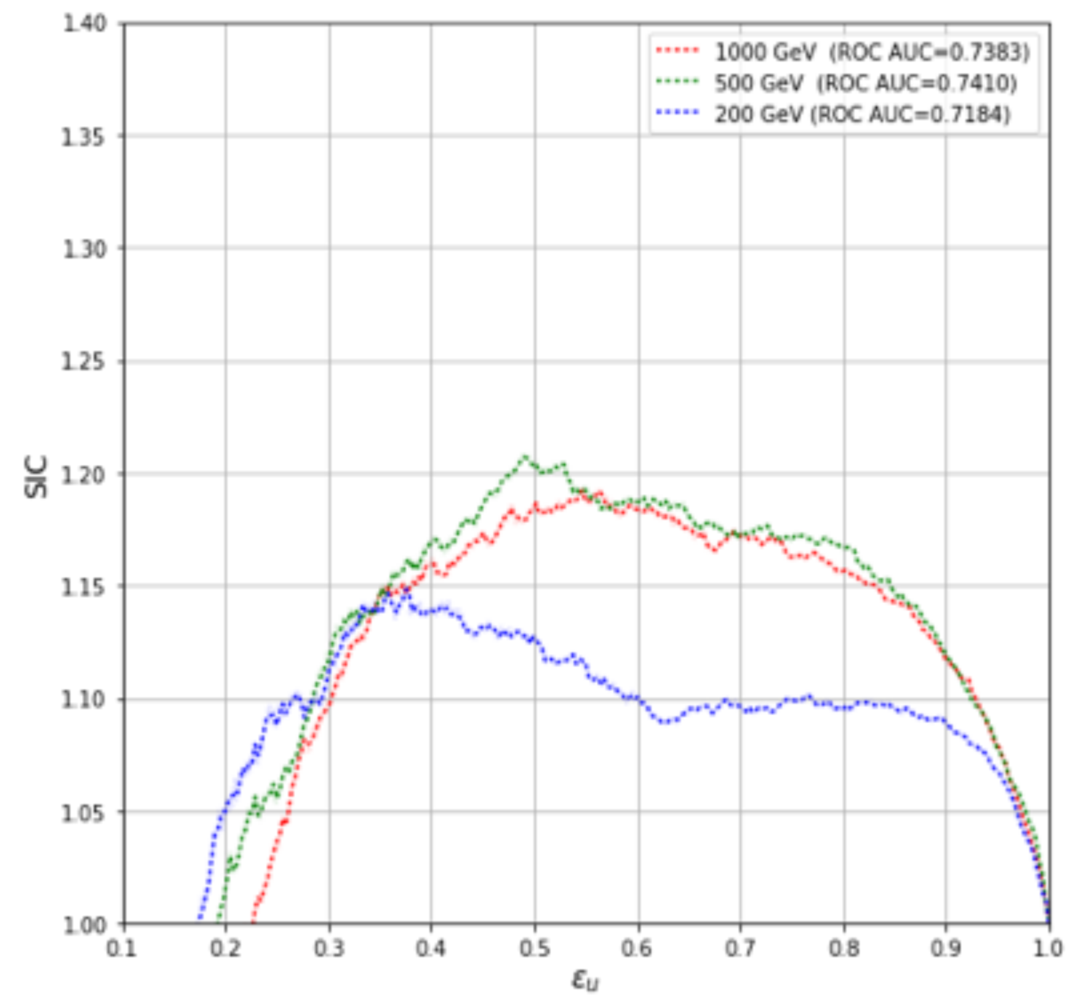
pt weighted jet charge $Q^J_\kappa = \sum_{i \in J} (\frac{p^i_T}{p^J_T})^\kappa q_i$



Jet Charge

u/d discrimination





RecNNs with pt-weighted charge

* one-hot implementation doesn't work here

For Different Tasks

W Tagging

Top tagging

q/g discrimination

jet charge discrimination

Leads to comparative study

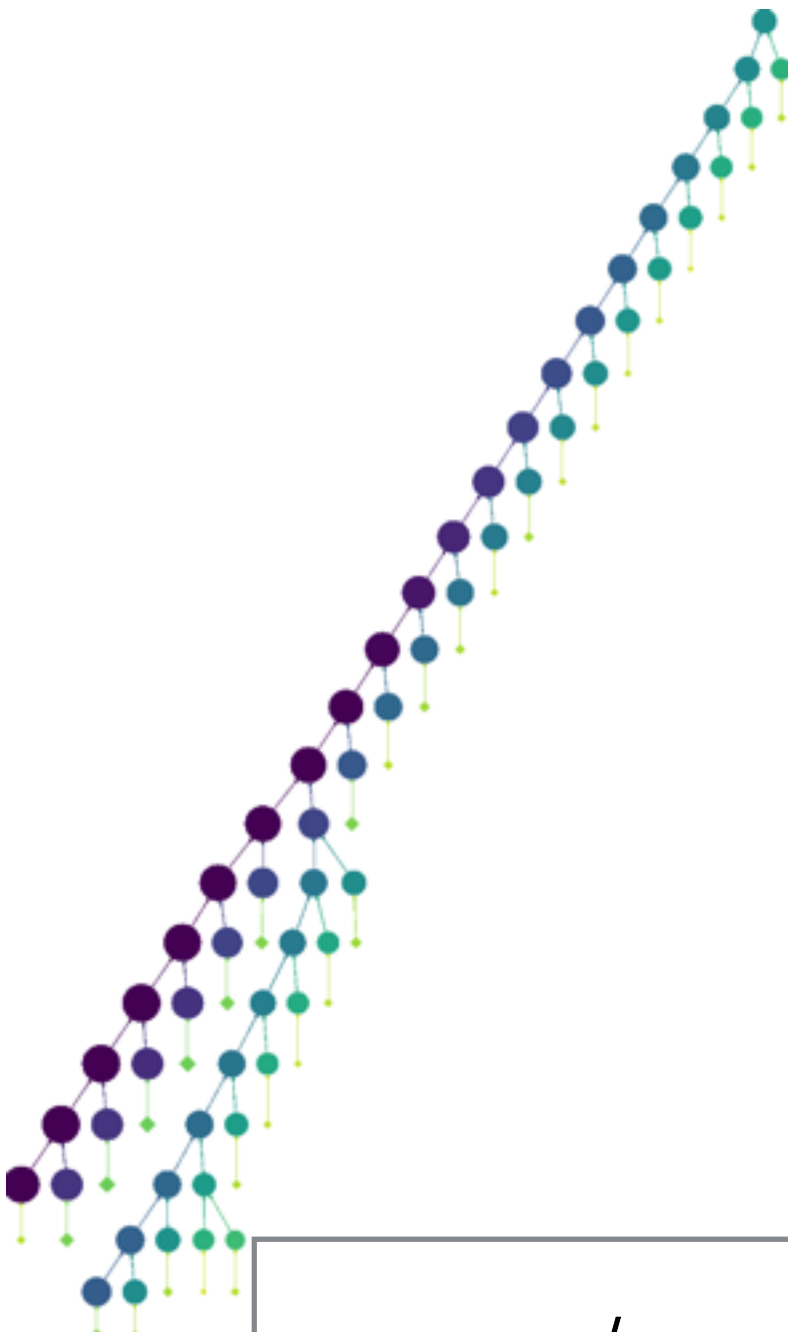
Representation &
Architecture

Interpretation

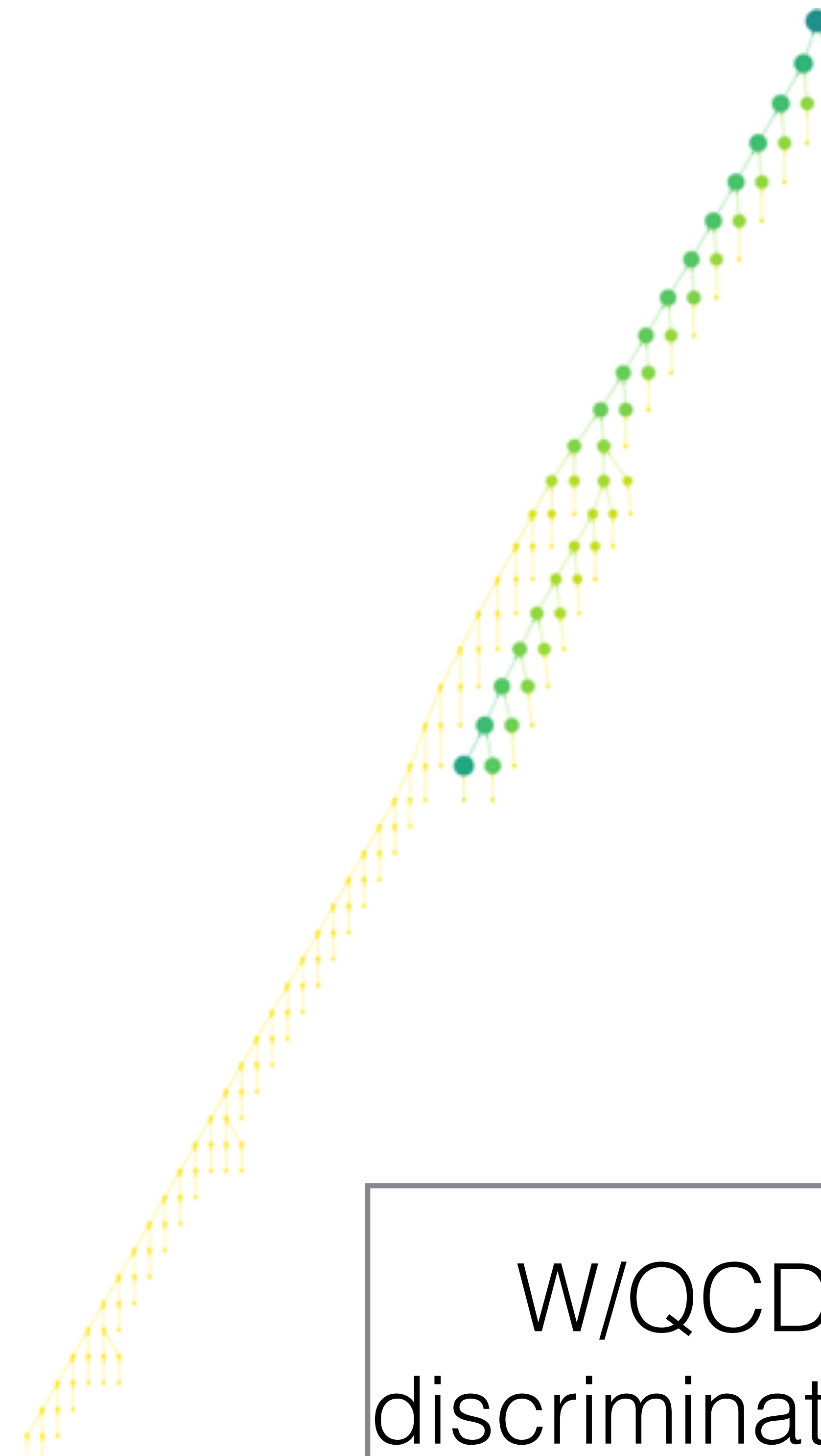
*what has been learnt?
*how it was learnt?
*anything else can be learnt?

Underlying Physics

Visualisation



q/g discrimination



W/QCD discrimination

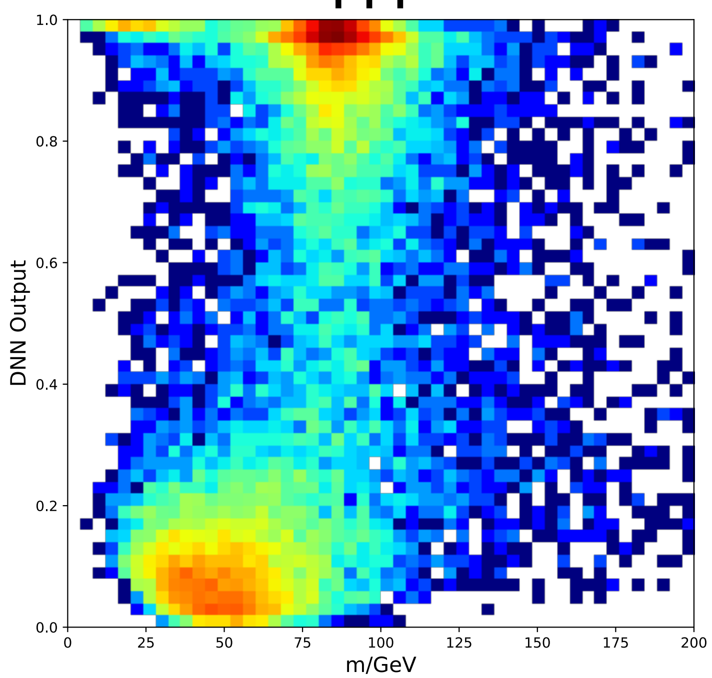
Preliminary Results

(in collaboration with Gilles Louppe)

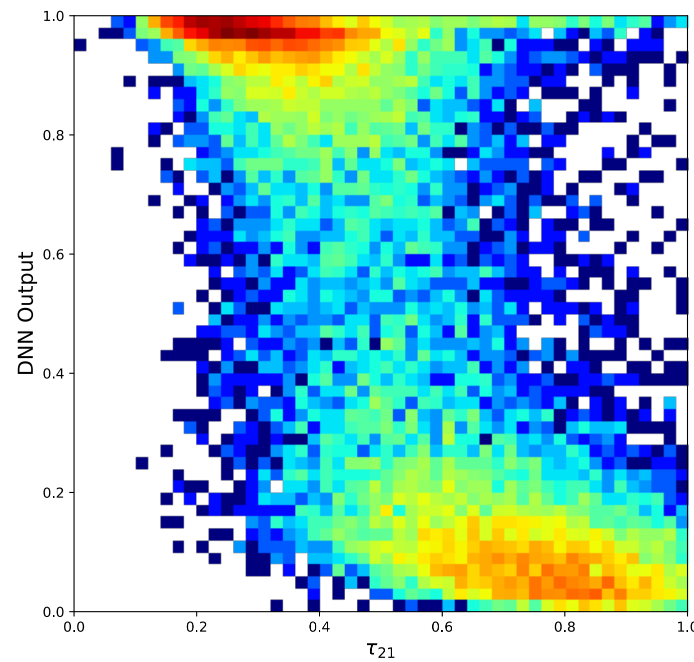
Joint Probability Distribution

W/QCD

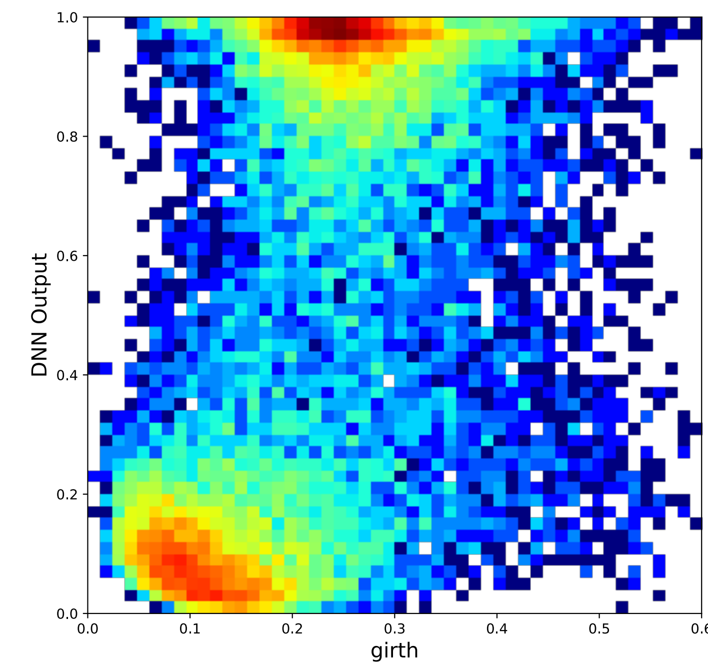
m

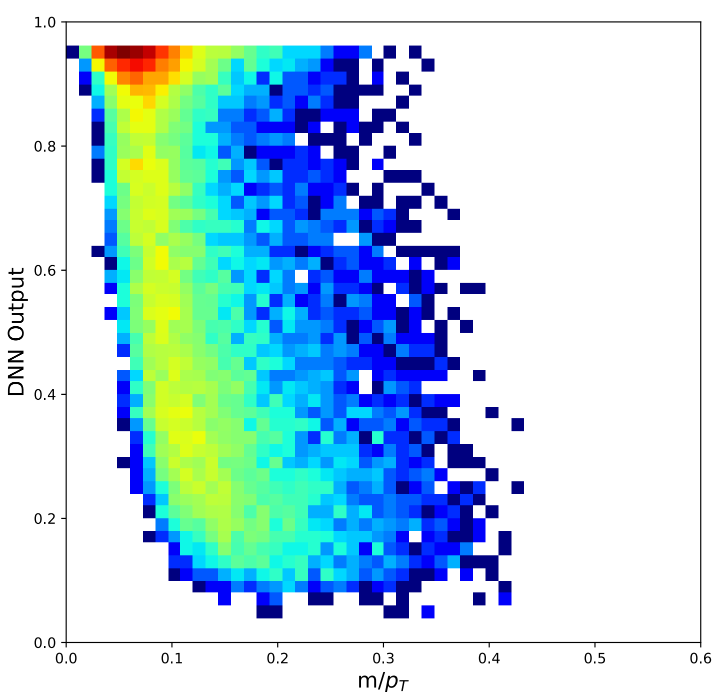


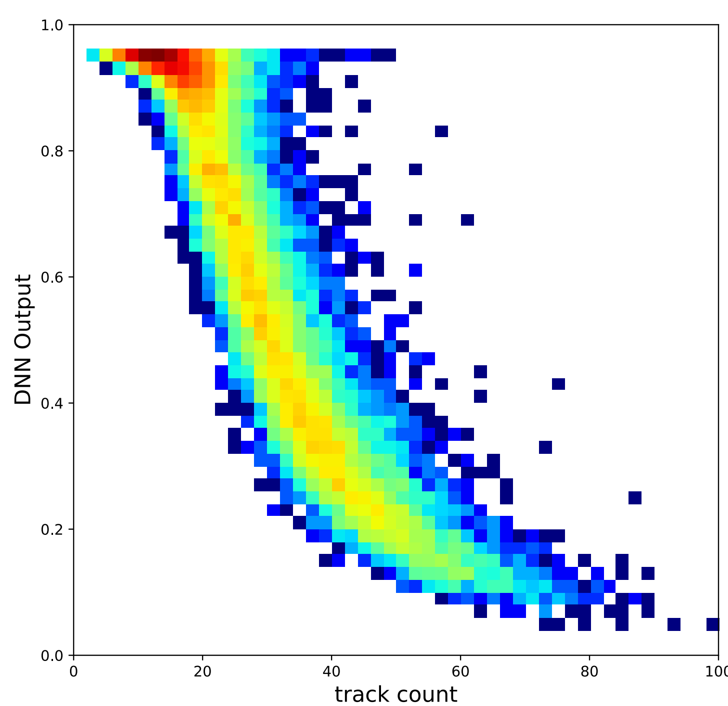
τ_{21}

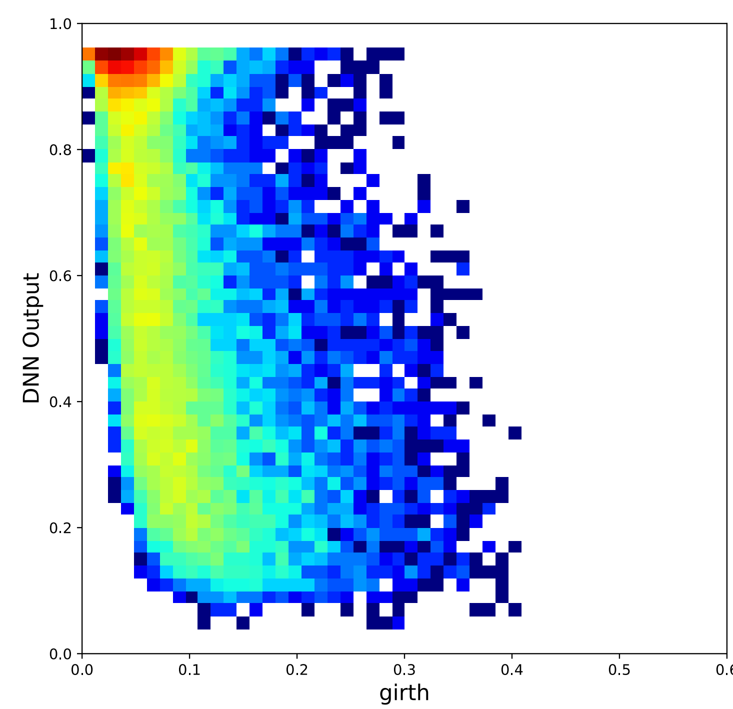


girth









q/g

m

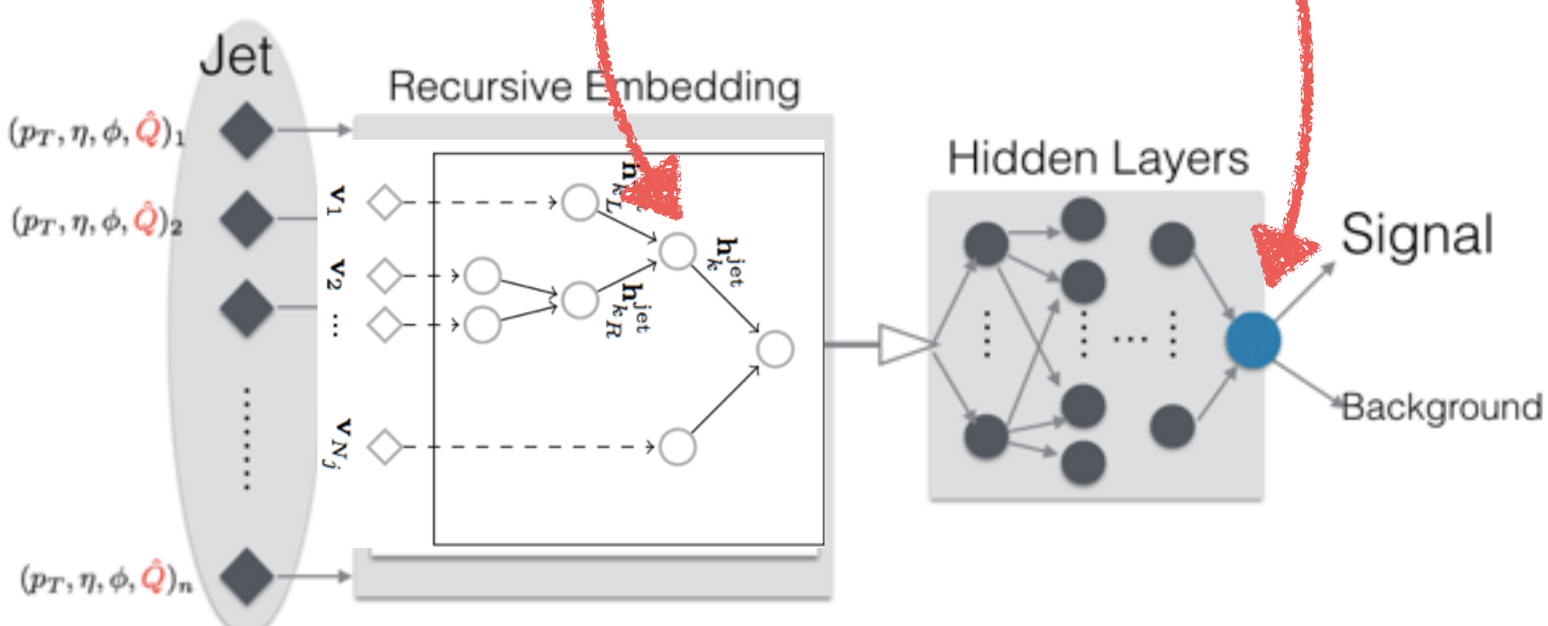
track count

girth

Sensitivity

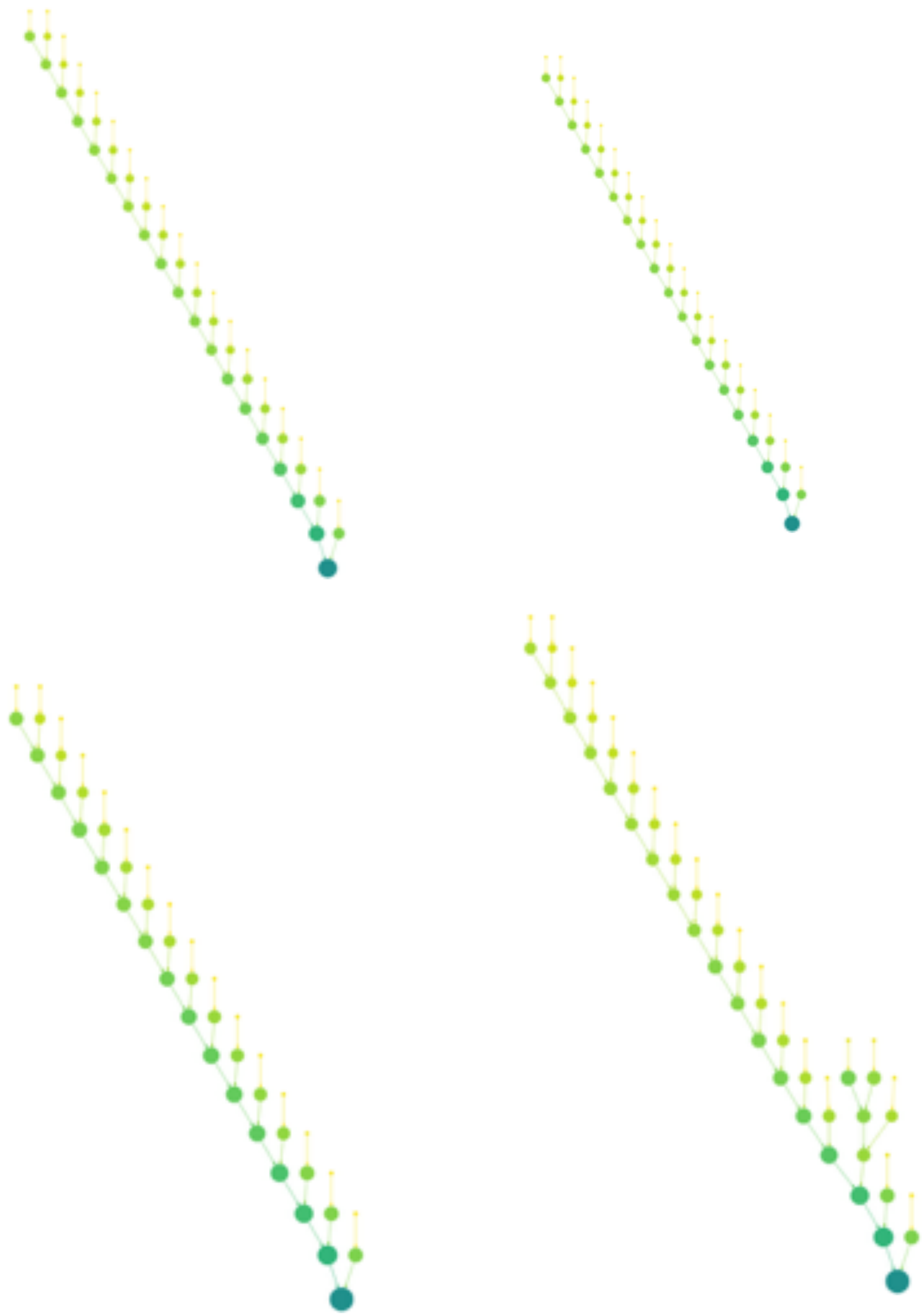
Sensitivity indicated by gradients

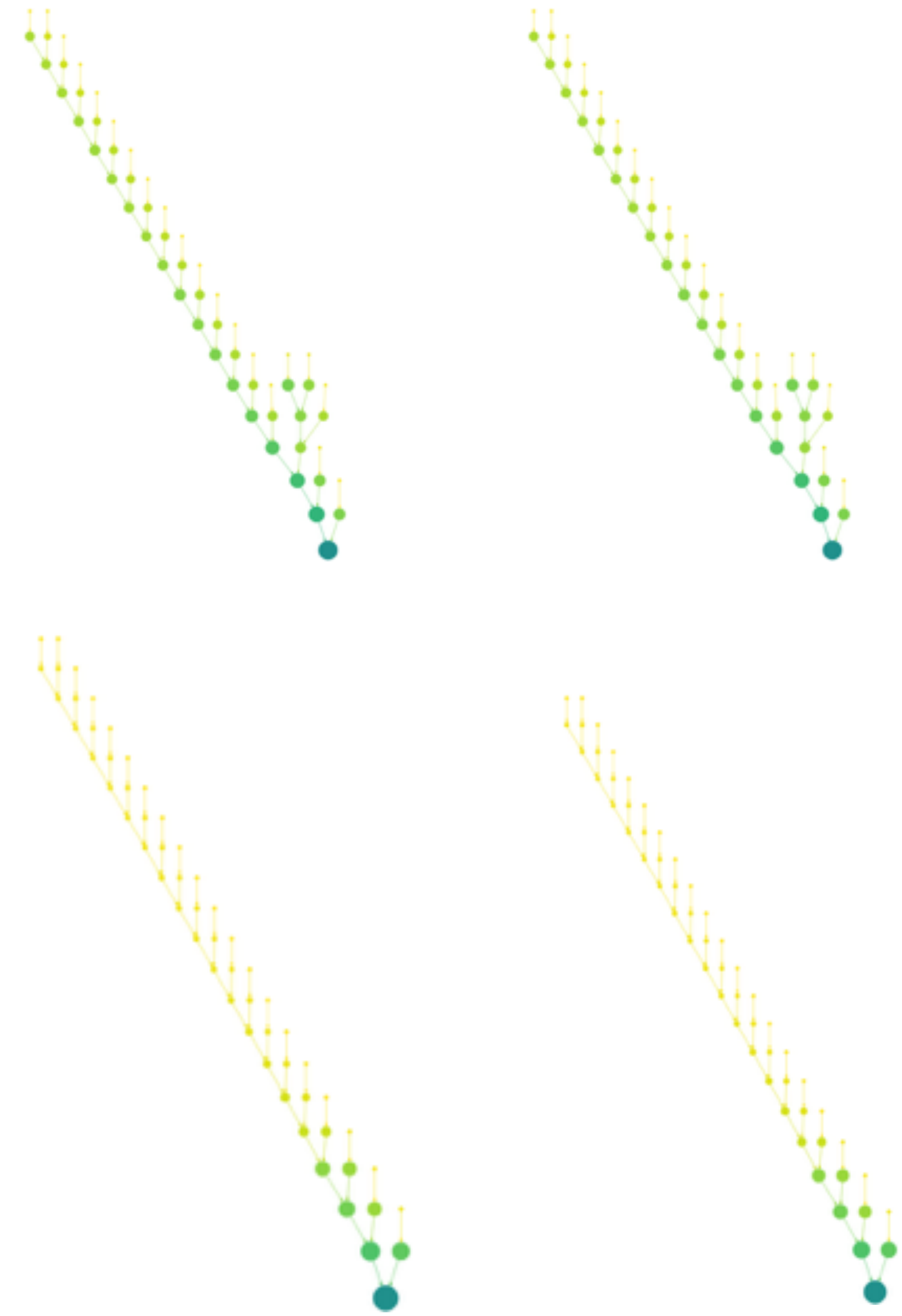
$$\left\| \frac{df}{dh} \right\|$$



W/QCD

Sensitivity

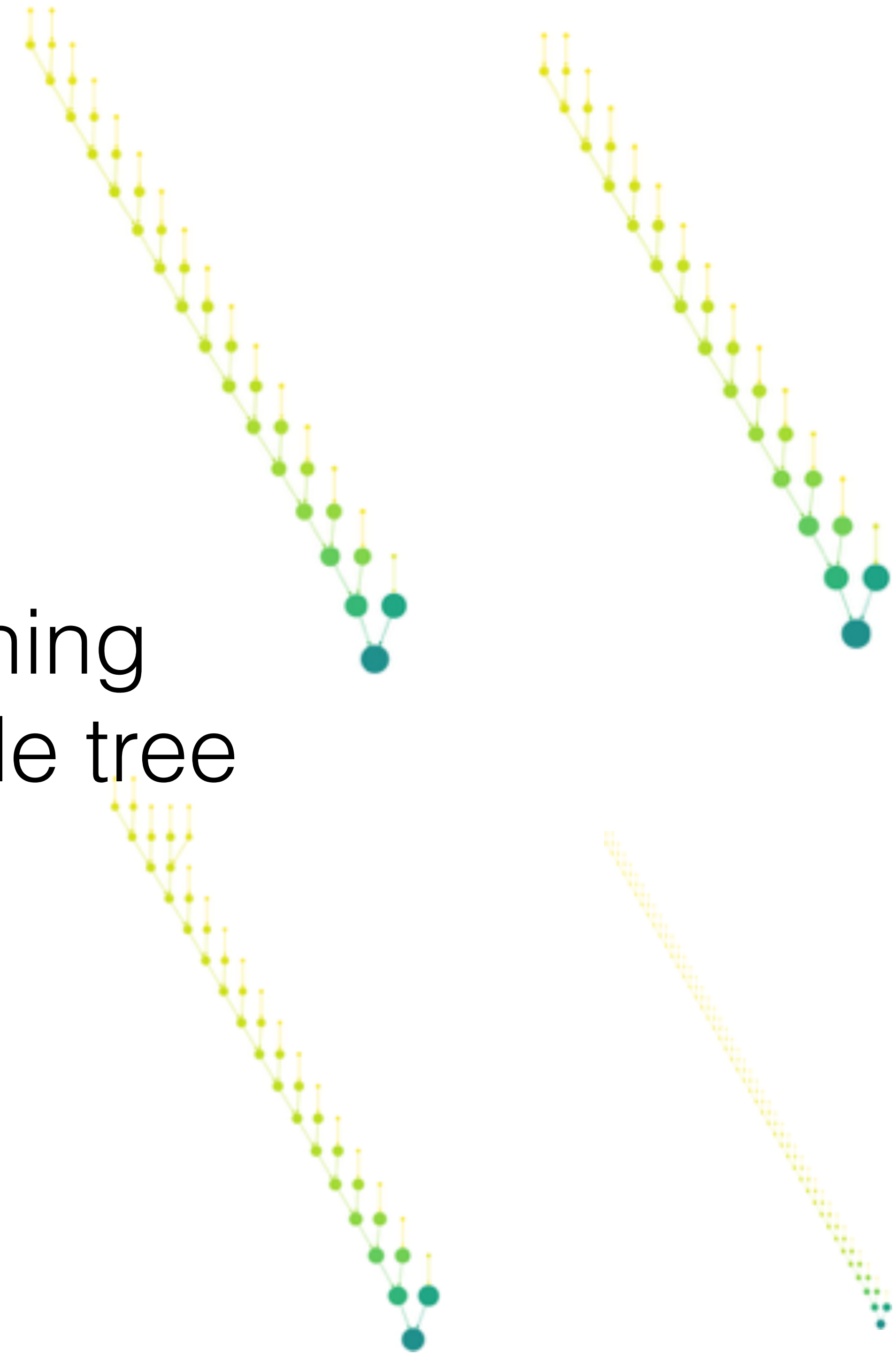




QCD Jets (best prediction samples) W Jets

Sensitivity

q/g

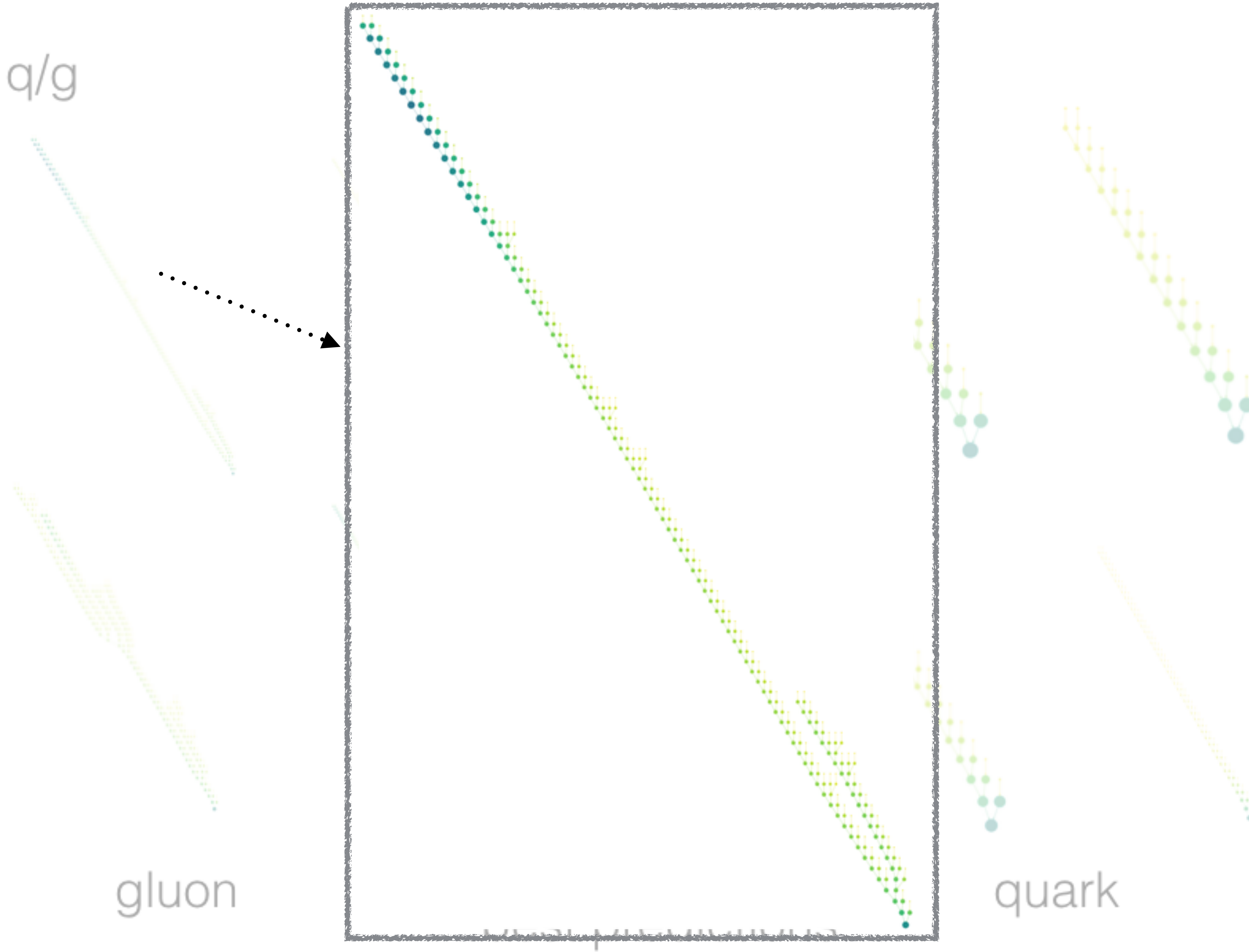


sensitivity running
through the whole tree

gluon

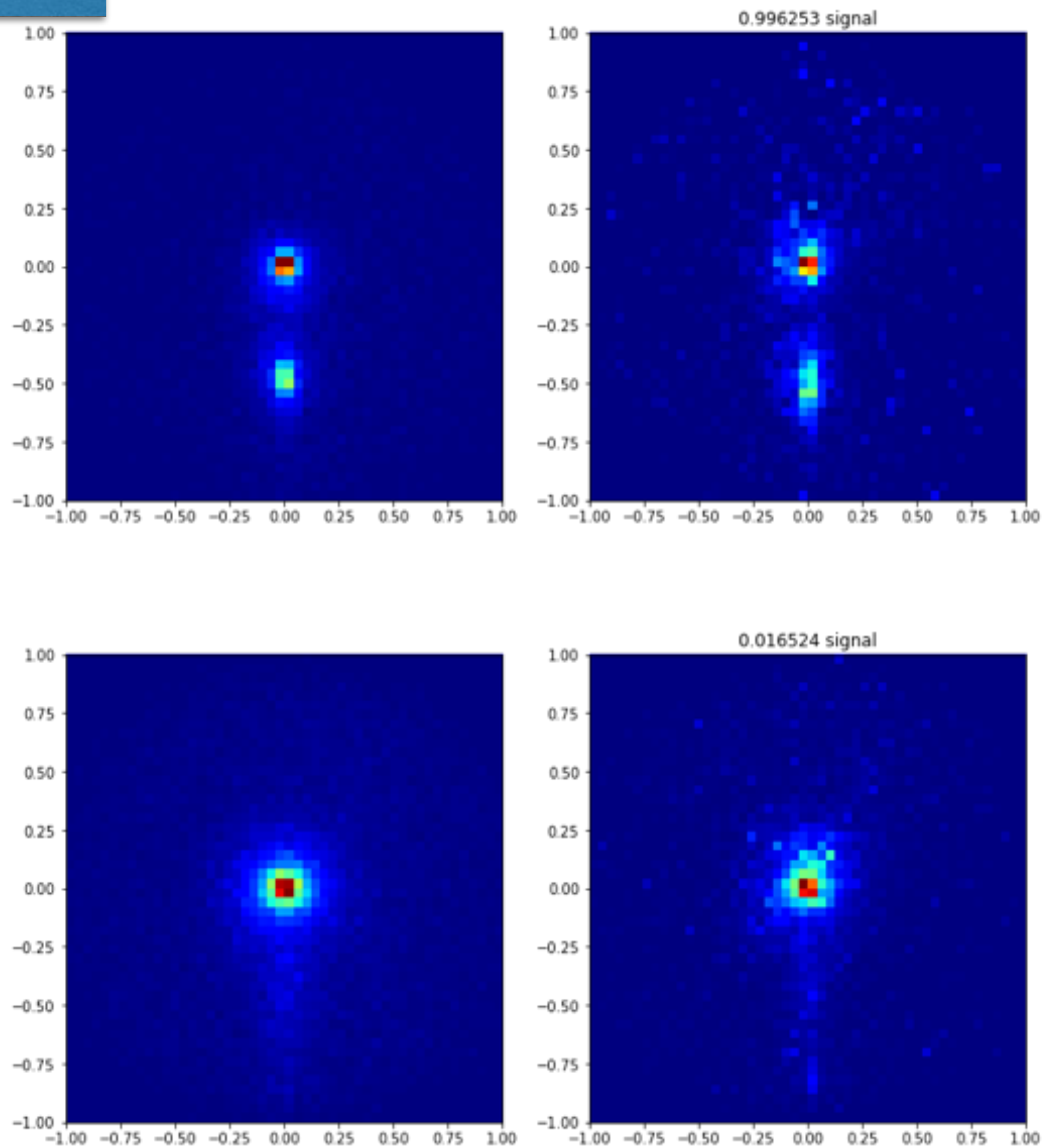
quark

(best prediction samples)


q/g
gluon
quark

Maximum Response Samples

W/QCD



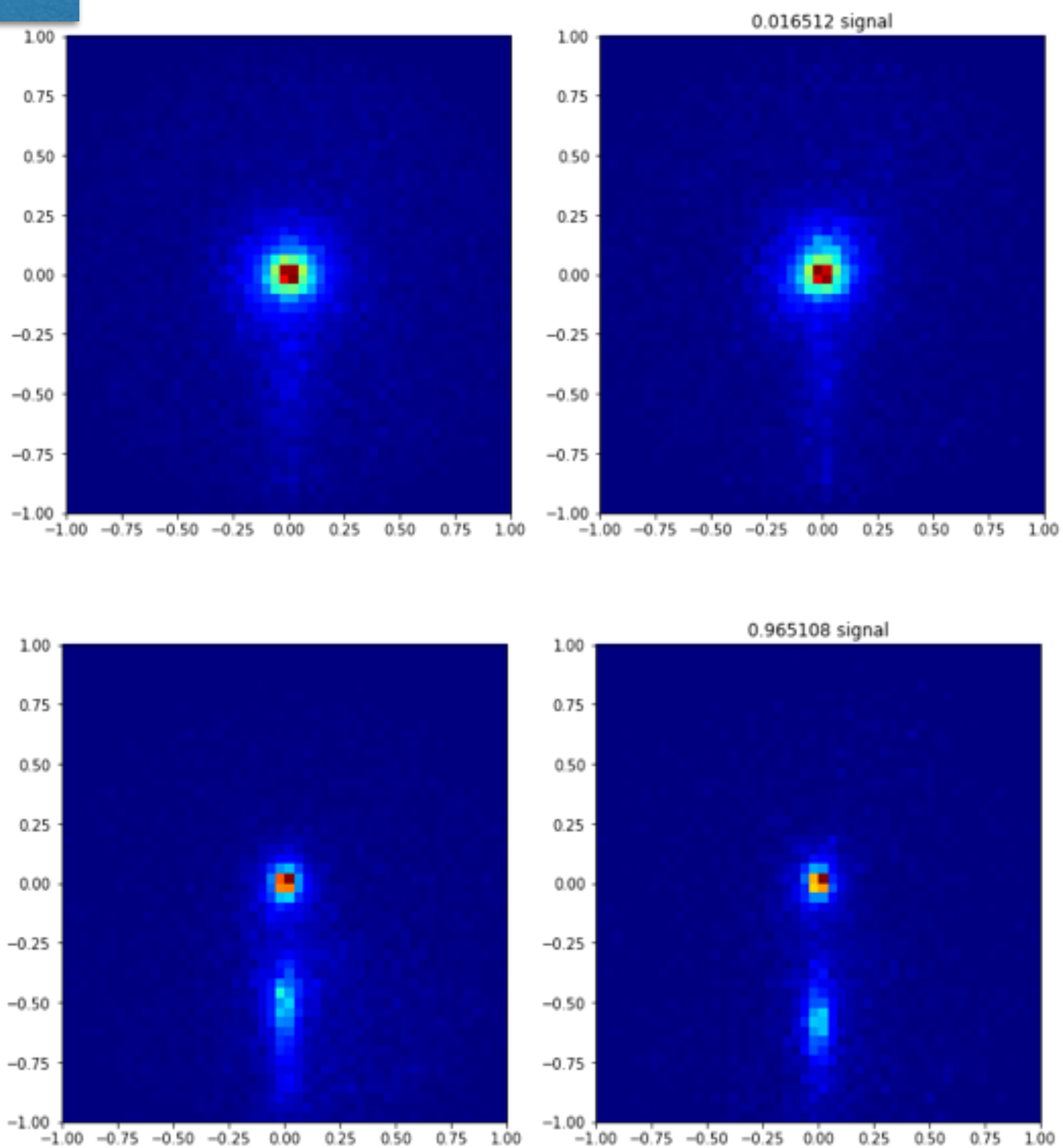
Output

jet images

sensitivity

W/QCD

Maximum Response Samples

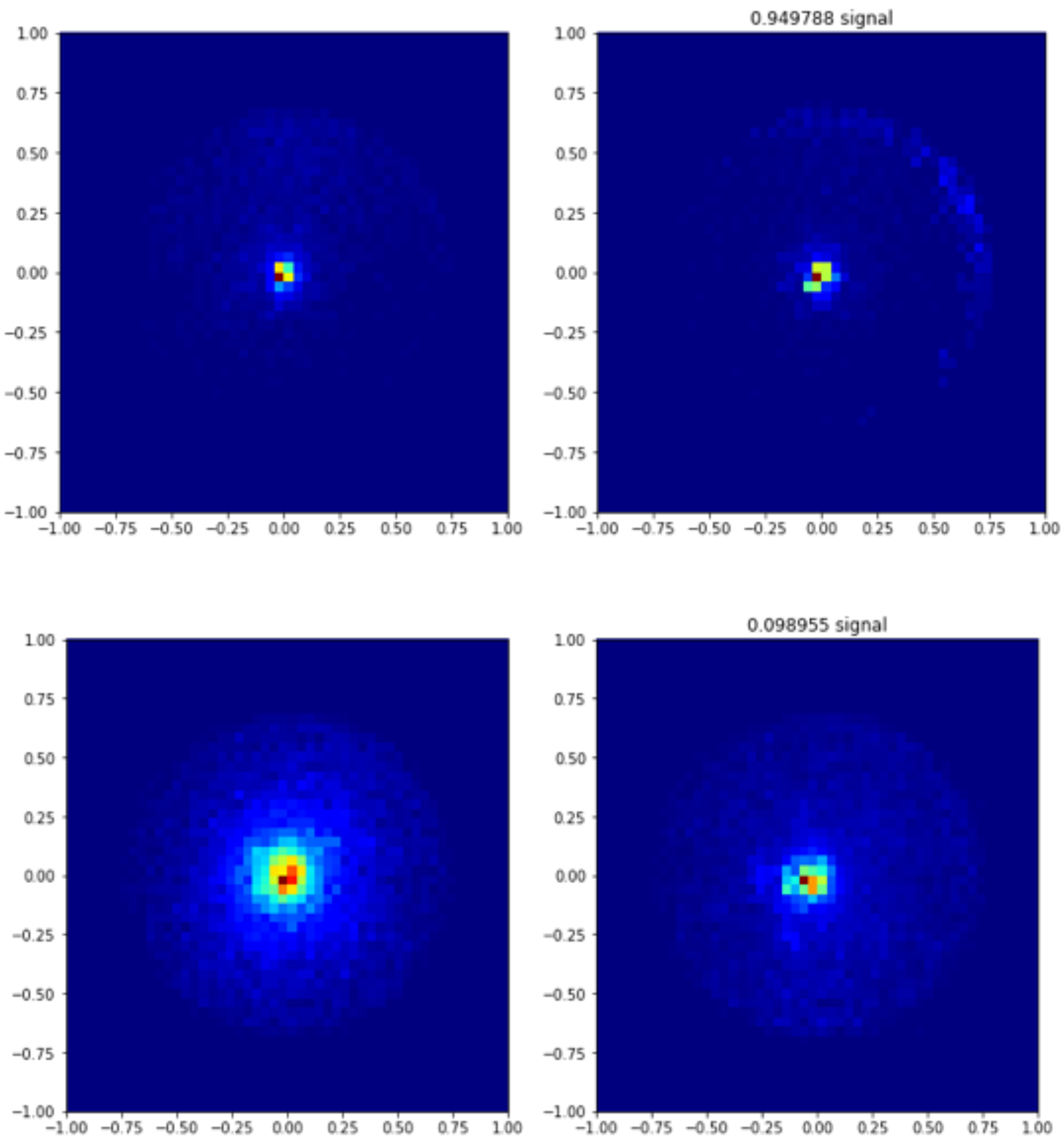


jet images

sensitivity

ReLU2

Maximum Response Samples



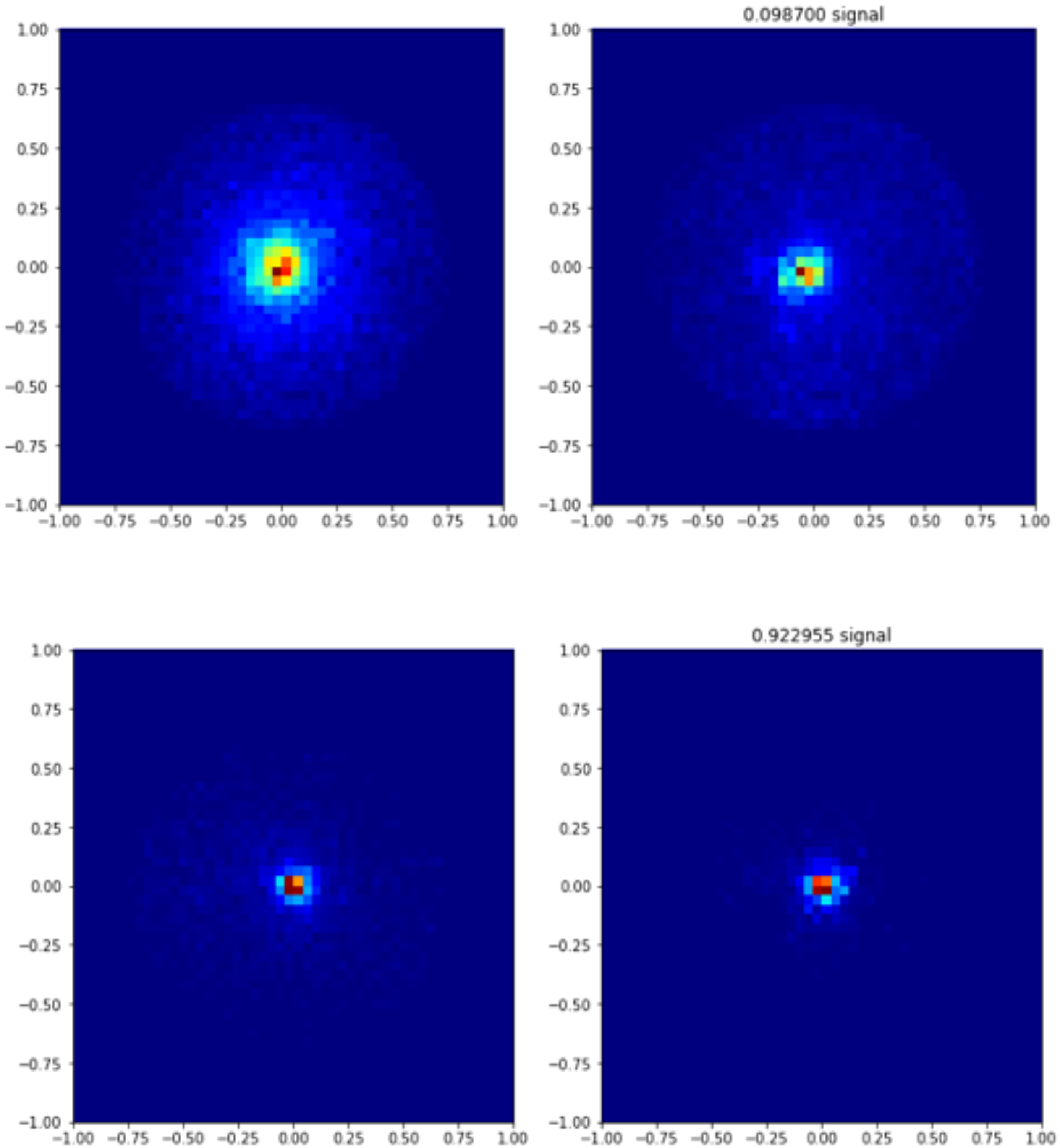
Output

jet images

sensitivity

q/g

Maximum Response Samples



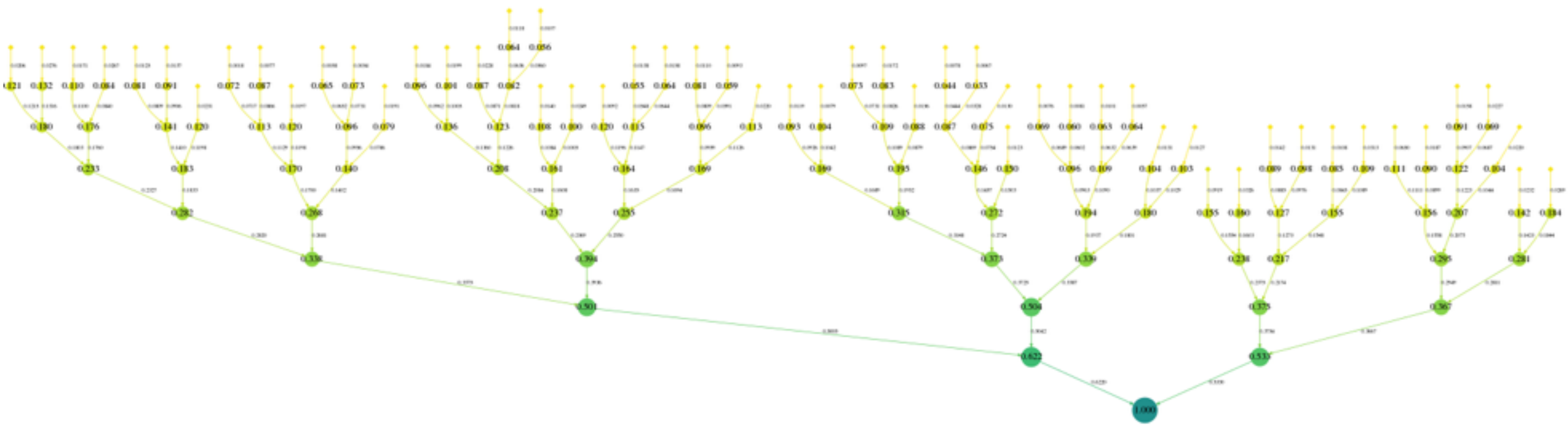
ReLU2

jet images

sensitivity

q/g

¿Jet Grooming?

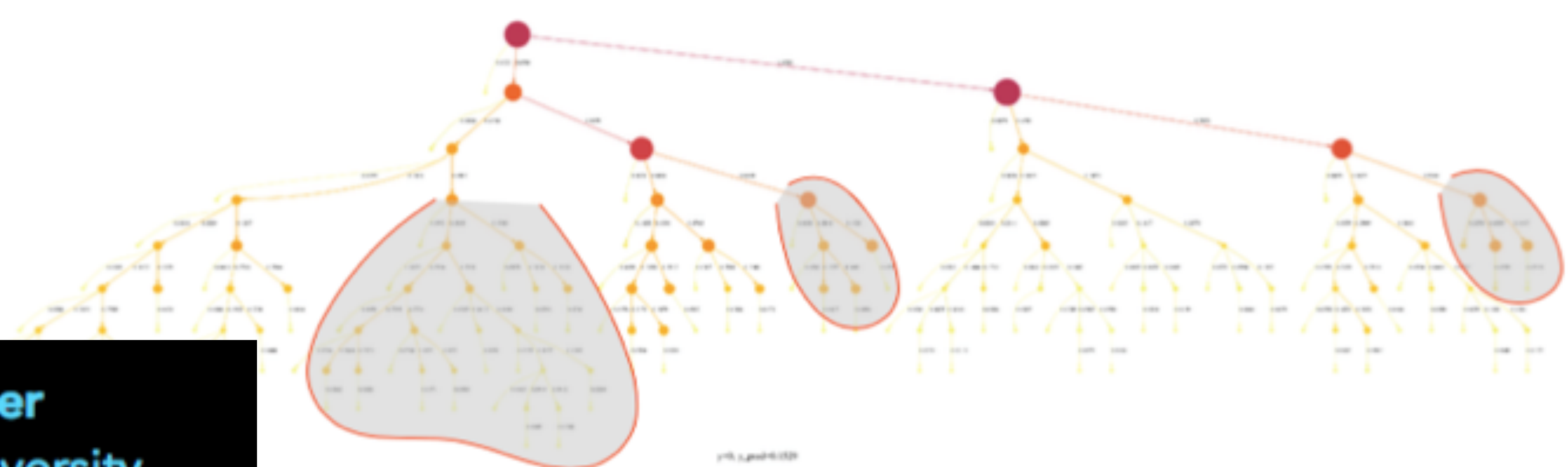


Jet Grooming

"LEARN TO PIVOT" → "LEARN TO GROOM"

We can use the same adversarial strategy to be robust to variations in pileup and underlying event.

- combined with GRU/LSTM gating, the network should learn to ignore parts of the jet that are not robust to these variations
- eg. network will learn a jet grooming/pruning/trimming/… strategy.
- Compare traditional grooming with weights assigned to constituents.

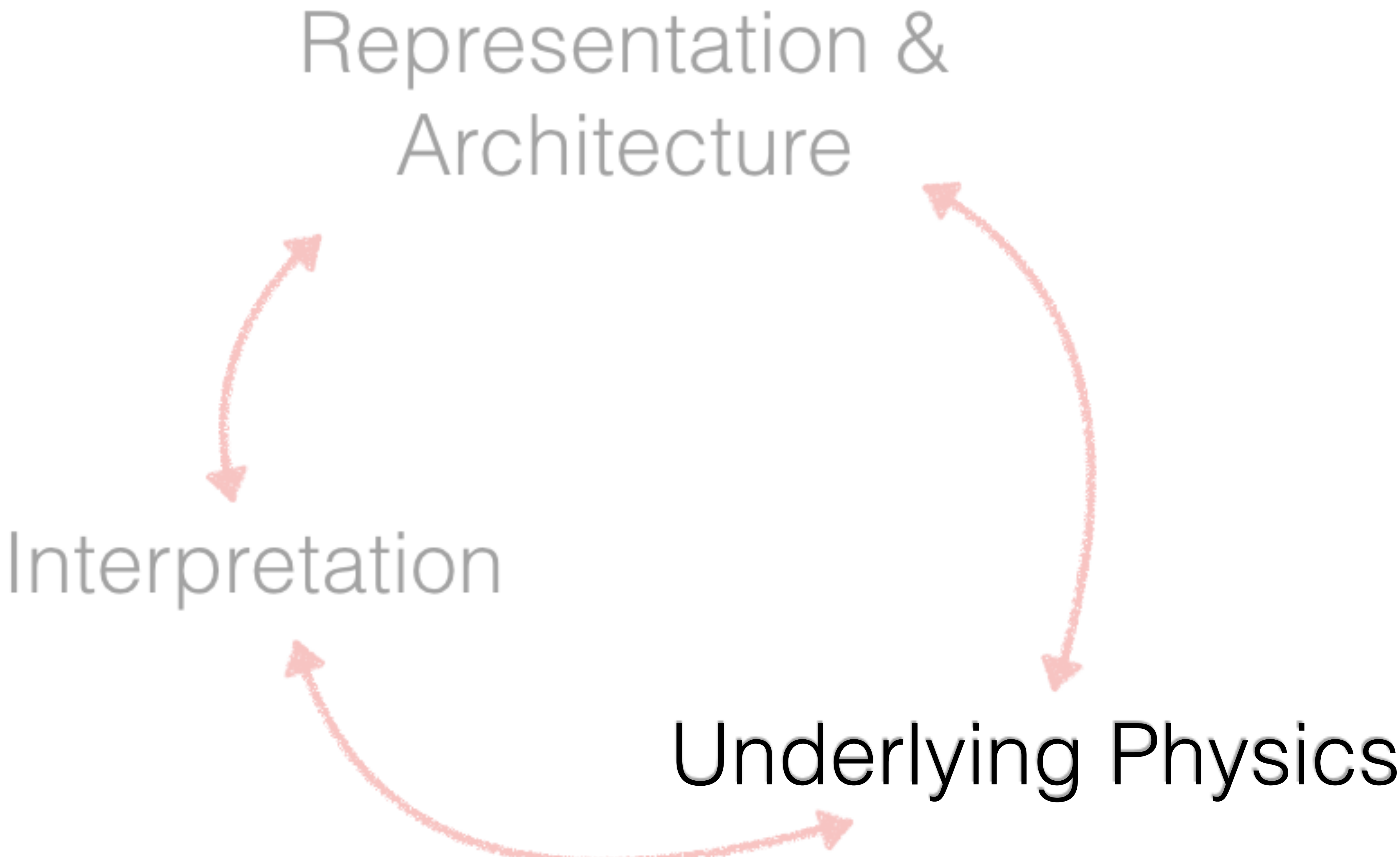


*Work in progress with Dipsikha Debnath

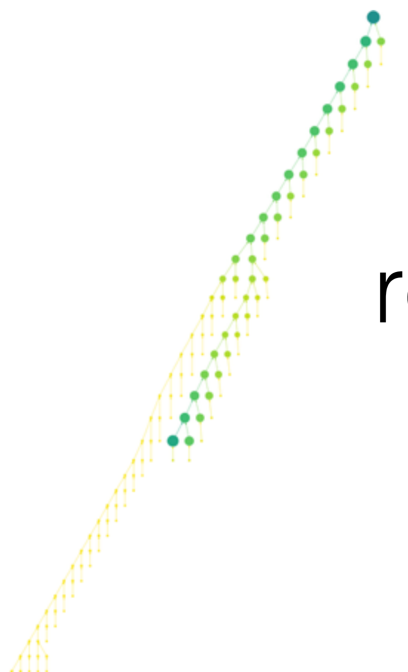
74

@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab

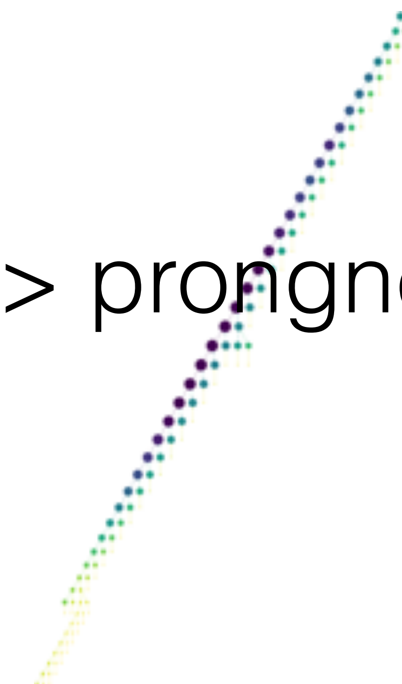
[K. Cranmer's talk at ML4Jets]


Representation &
Architecture
Interpretation
Underlying Physics

Sensitivity

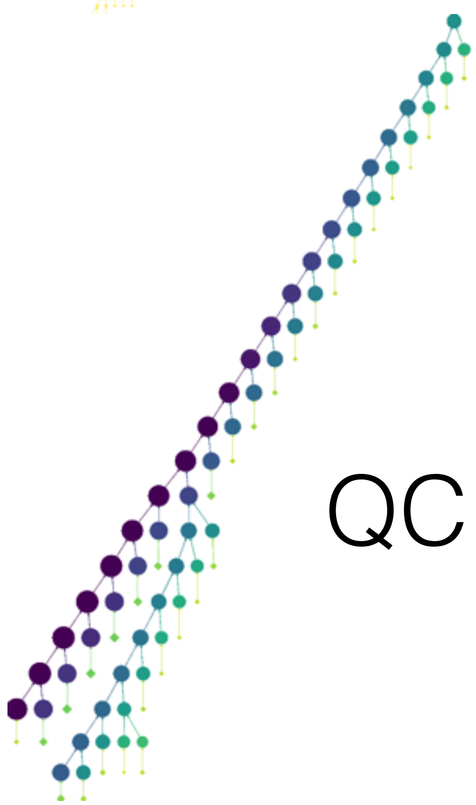


W Tagging



Top Tagging

resonance decay -> prongness



q/g Tagging

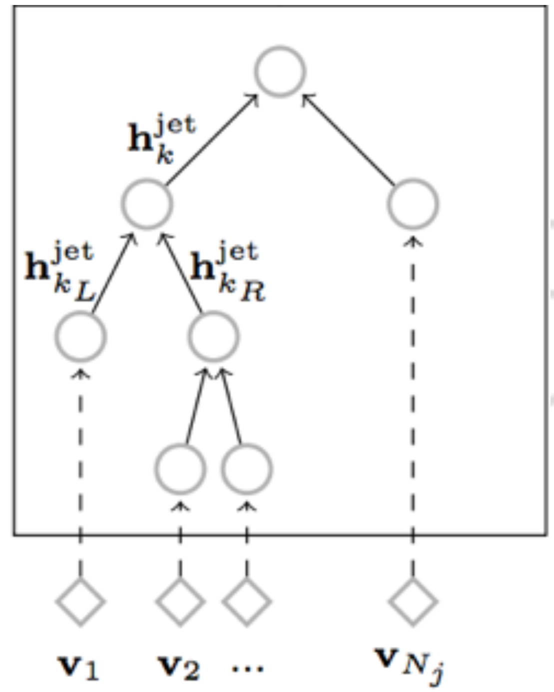
u/d Tagging

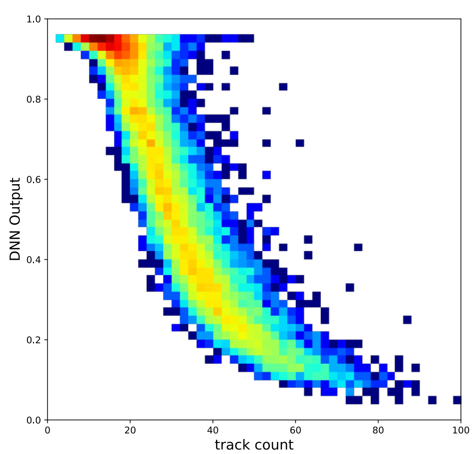
QCD Radiation

Electroweak

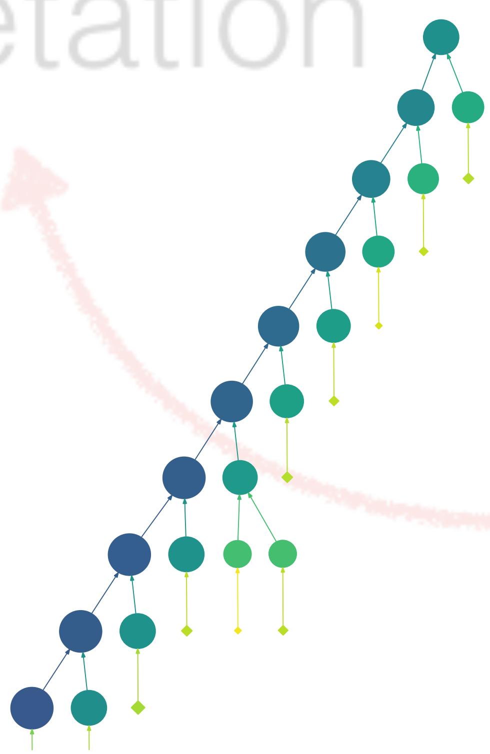
(work in progress)

Repr on &
Ar re





tation



More than a classifier

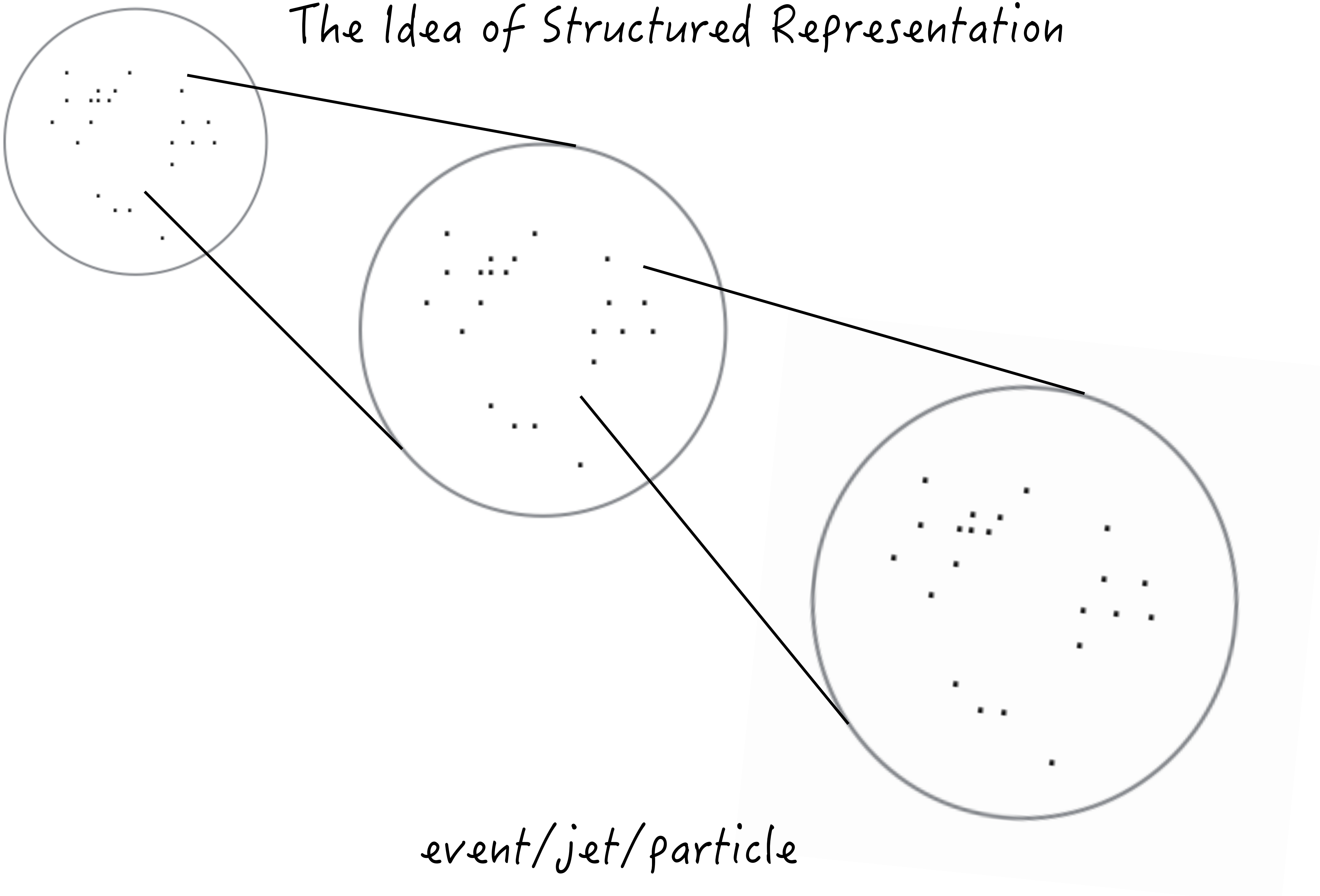
Underlying Physics

Wrap-up

* Jet Clustering inspired RecNN framework for (not only) jet tagging
* Effective, Compact, Transferability
* Physics-friendly
* Nice interpretability
* Not only a classifier (physics intuition, practical use, and natural structure)

Thank you!

The Idea of Structured Representation



event/jet/particle

Backup

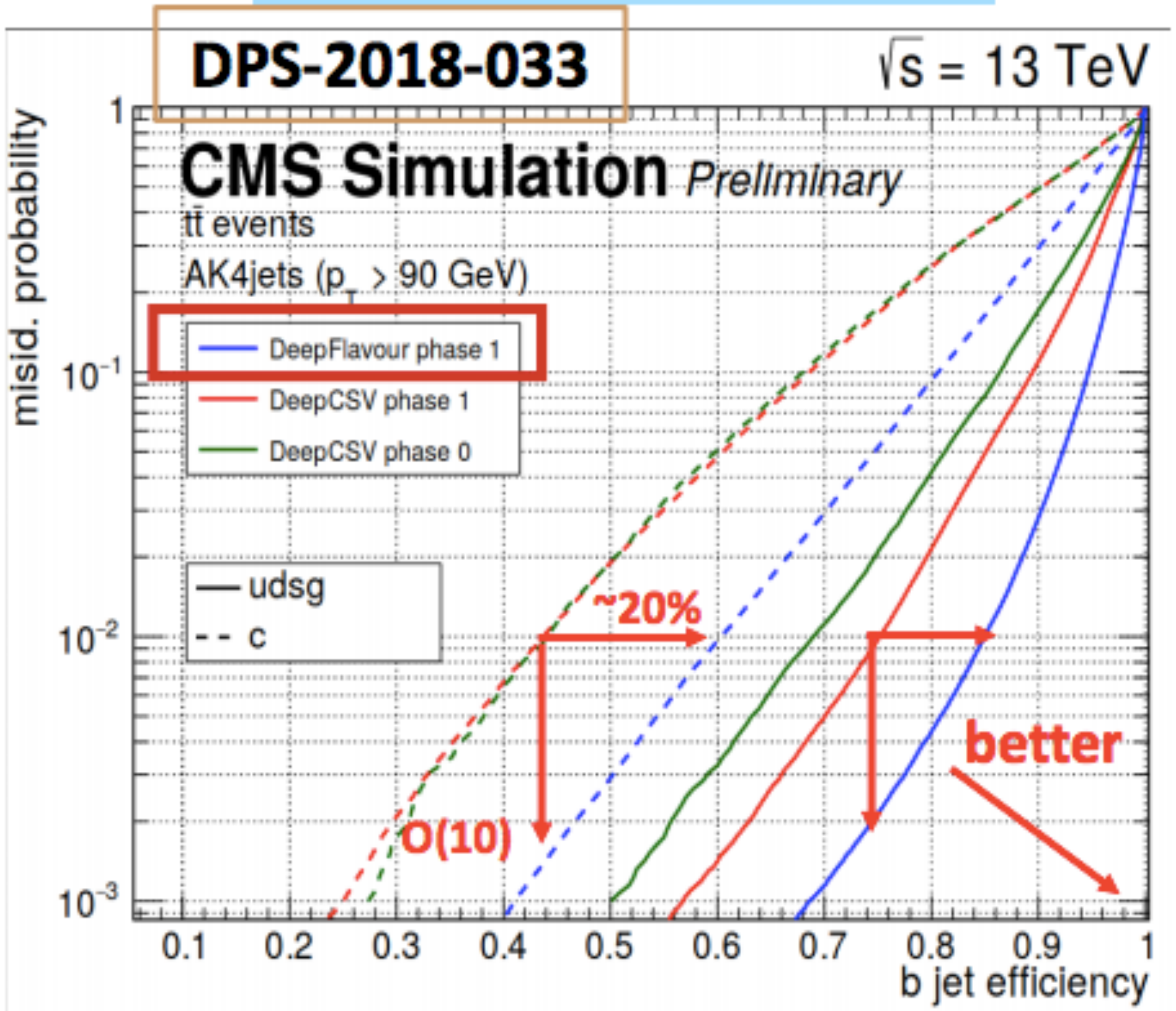
Quark/Gluon Discrimination

	Gluon Jet Efficiency (%) at 50 % Quark Jet Acceptance	200 GeV	1000 GeV
Pythia	BDT of all jet variables	*5.2**	*5.2**
	Deep CNN without Color	*4.8**	*4.0**
	Deep CNN with Color	*4.6**	*3.4**
	RecNN without pflow	6.4	4.5
Delphes	BDT	9.5	6.2
	RecNN without pflow	7.8	4.6
	RecNN with categorical pflow	7.1	4.5
	RecNN with pt-weighted charge	7.8	4.9
Full Sim.	DNN@CMS	~*10.0*†	–

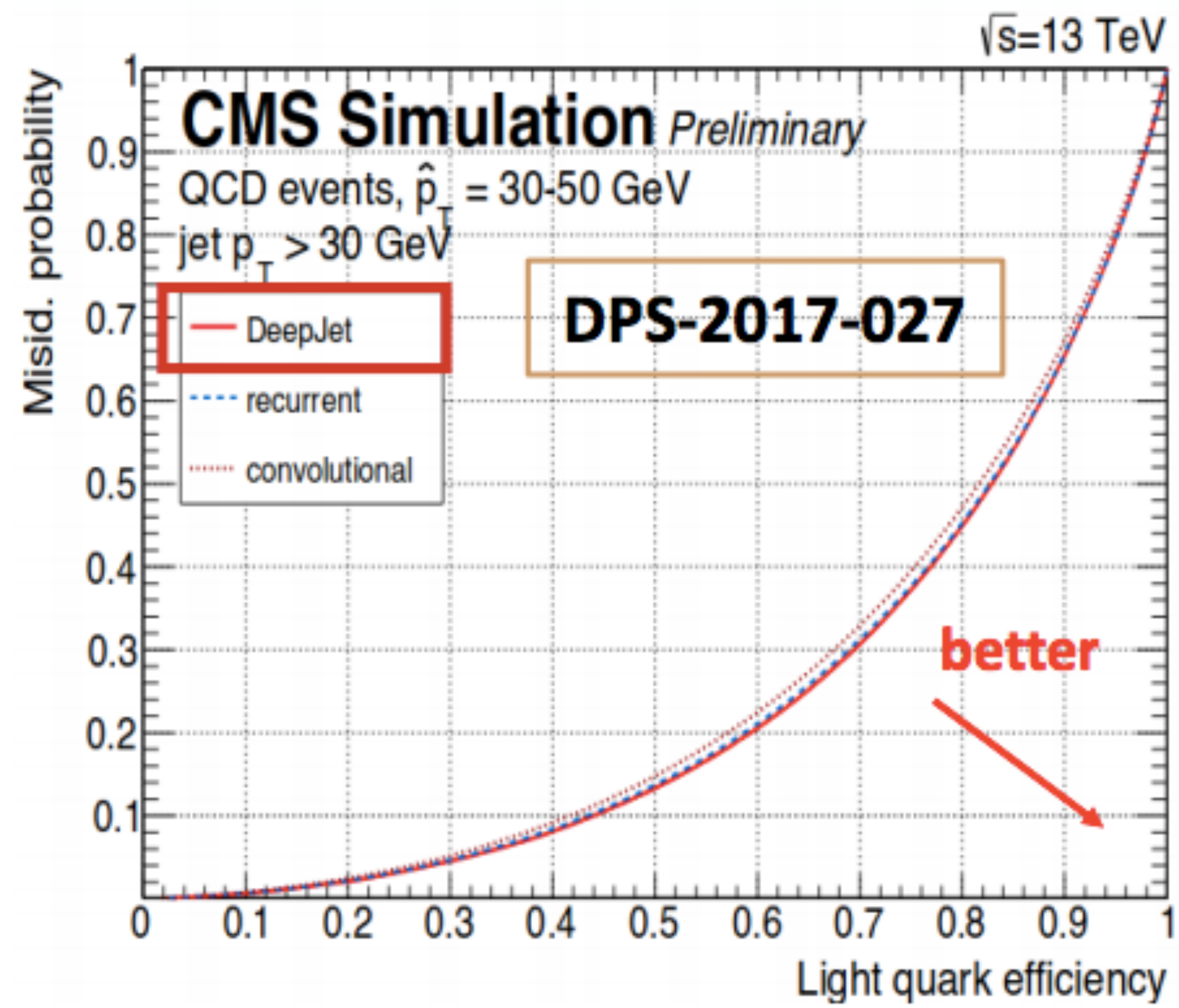
(* data taken from P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, arXiv:1612.01551;
† data taken from CMS Collaboration Collaboration, New Developments for Jet Substructure Reconstruction in CMS)

Performance of slim jet tagging

b vs. udsg / b vs. c



Quark – gluon separation



→ Significant gain in performance even more significant at higher pT

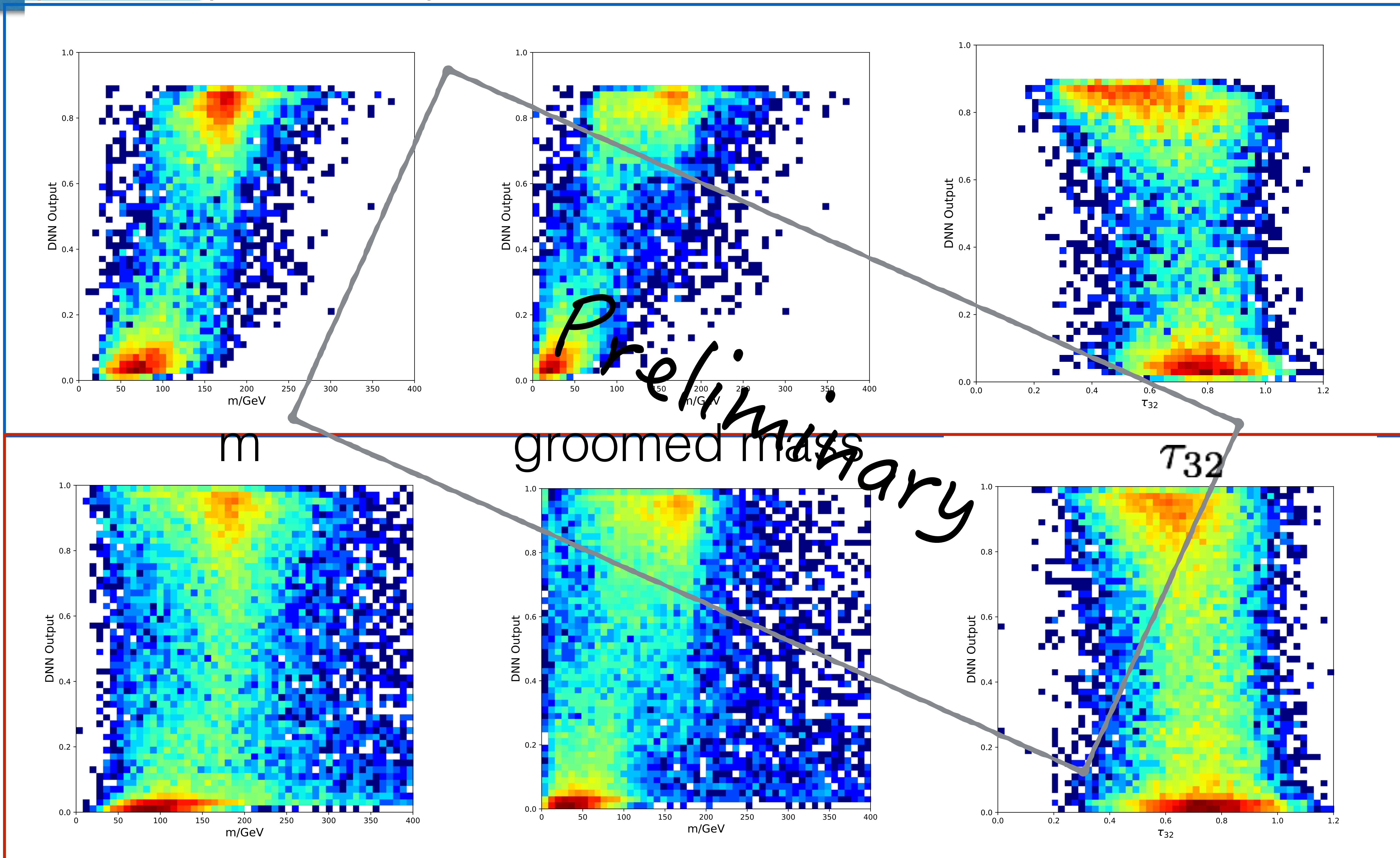
→ Large part of the performance loss of previous [non particle-based] taggers was due to track preselection

→ Generator level light quarks/gluons that did not split to heavy flavour

→ Similar performance to simpler & dedicated architectures

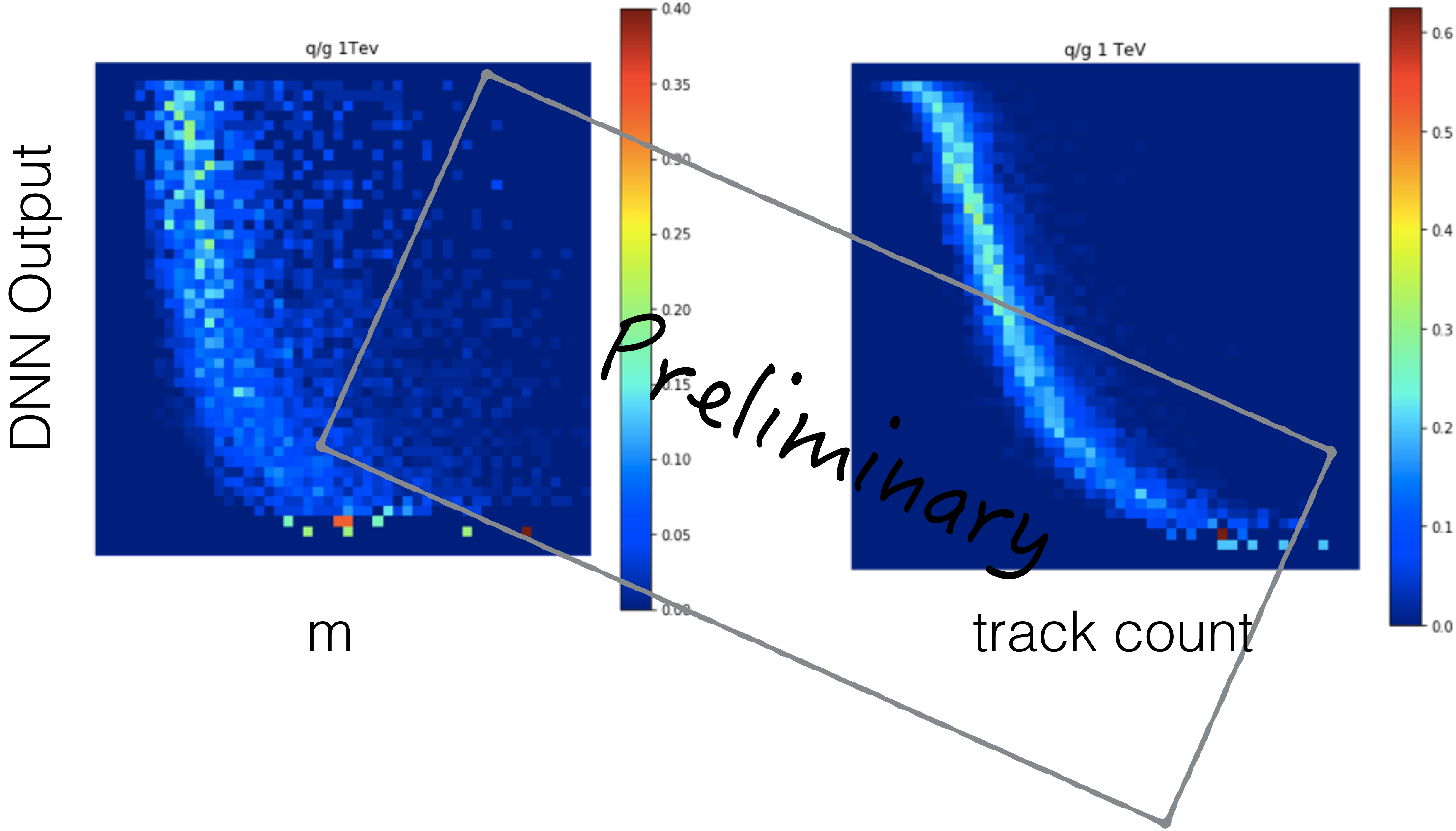
Joint Probability Distribution

Top/QCD(500 GeV)



Top/QCD(1 TeV)

Conditional Probability Distribution



IRC Safety

Table 2. Performance of pre-trained RNN classifiers (without gating) applied to nominal and modified particle inputs. The *collinear1* (*collinear10*) scenarios correspond to applying collinear splits to one (ten) random particles within the jet. The *collinear1-max* (*collinear10-max*) scenarios correspond to applying collinear splits to the highest p_T (ten highest p_T) particles in the jet. The *soft* scenario corresponds to adding 200 particles with $p_T = 10^{-5}$ GeV uniformly in $0 < \phi < 2\pi$ and $-5 < \eta < 5$.

Scenario	Architecture	ROC AUC	$R_{\epsilon=50\%}$
nominal	k_t	0.9185 ± 0.0006	68.3 ± 1.8
nominal	desc-p_T	0.9189 ± 0.0009	70.4 ± 3.6
collinear1	k_t	0.9183 ± 0.0006	68.7 ± 2.0
collinear1	desc-p_T	0.9188 ± 0.0010	70.7 ± 4.0
collinear10	k_t	0.9174 ± 0.0006	67.5 ± 2.6
collinear10	desc-p_T	0.9178 ± 0.0011	67.9 ± 4.3
collinear1-max	k_t	0.9184 ± 0.0006	68.5 ± 2.8
collinear1-max	desc-p_T	0.9191 ± 0.0010	72.4 ± 4.3
collinear10-max	k_t	0.9159 ± 0.0009	65.7 ± 2.7
collinear10-max	desc-p_T	0.9140 ± 0.0016	63.5 ± 5.2
soft	k_t	0.9179 ± 0.0006	68.2 ± 2.3
soft	desc-p_T	0.9188 ± 0.0009	70.2 ± 3.7

[G. Louppe, K. Cho, C. Becot and K. Cranmer, arXiv:1702.00748]