NYU Center for Data Science

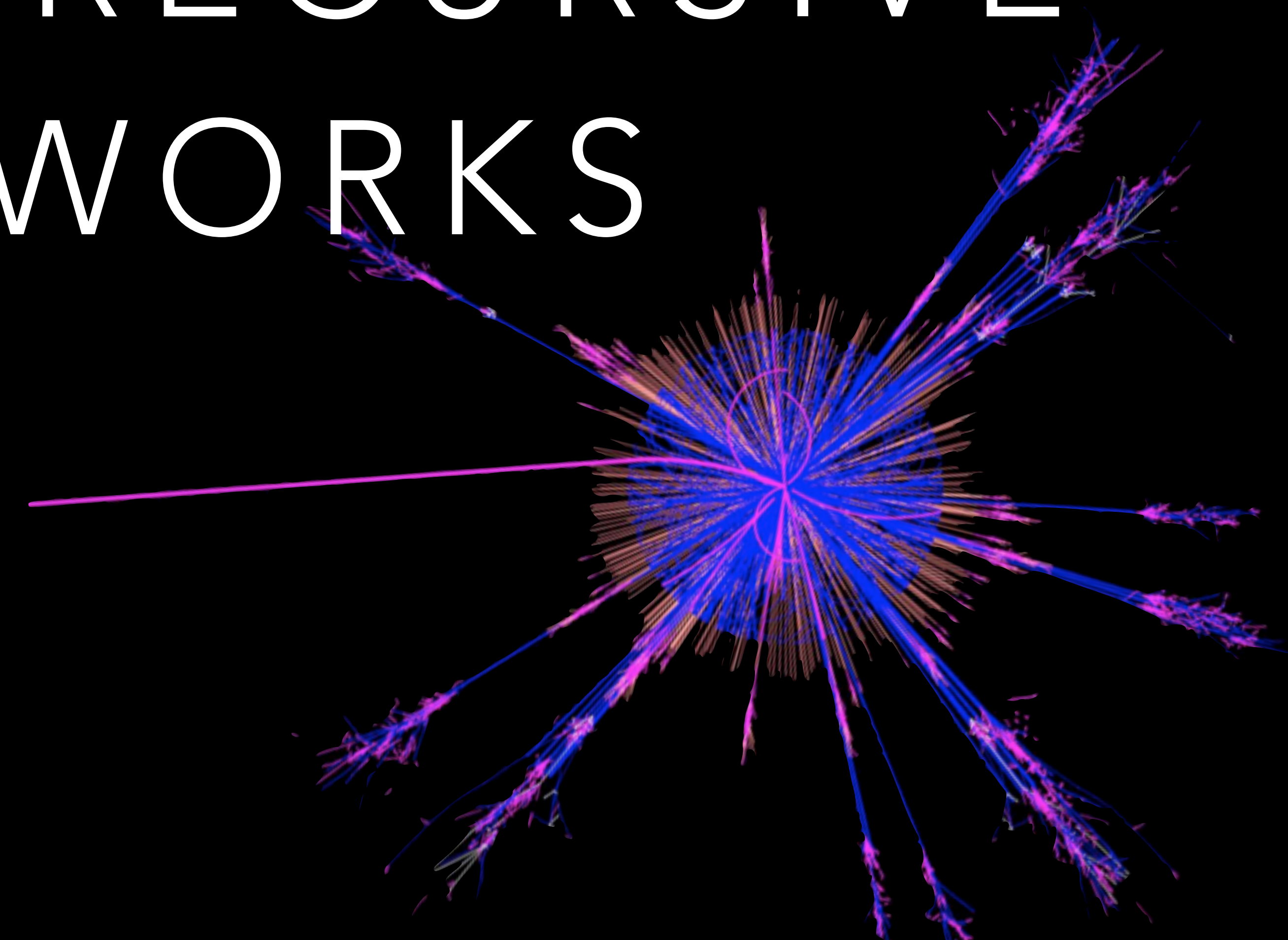
Center for Cosmology and Particle Physics

ARXIV:1702.00748

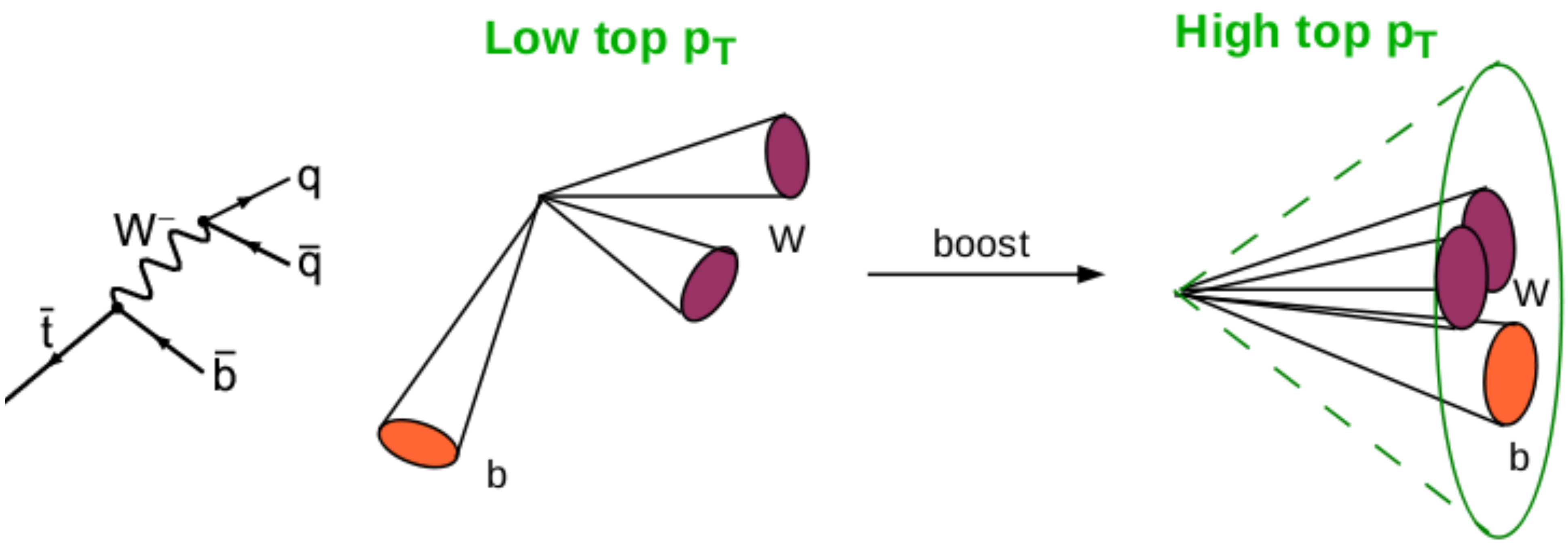
QCD-AWARE RECURSIVE NEURAL NETWORKS

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

with:
Gilles Louppe
Kyunghyun Cho
Joan Bruna
Cyril Becot

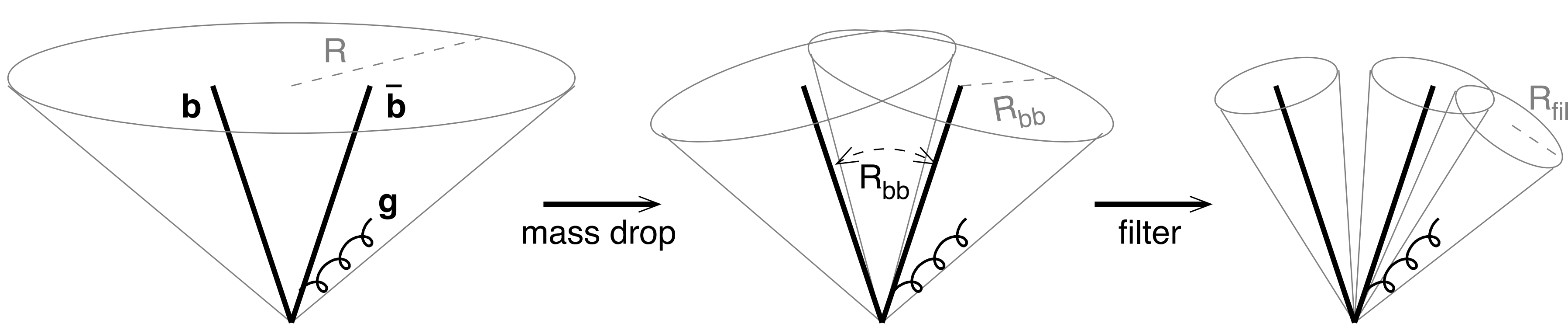


Many scenarios for physics Beyond the Standard Model include highly boosted W, Z, H bosons or top quarks



Identifying these rests on subtle substructure inside jets

- an enormous number of theoretical effort in developing observables and techniques to tag jets like this

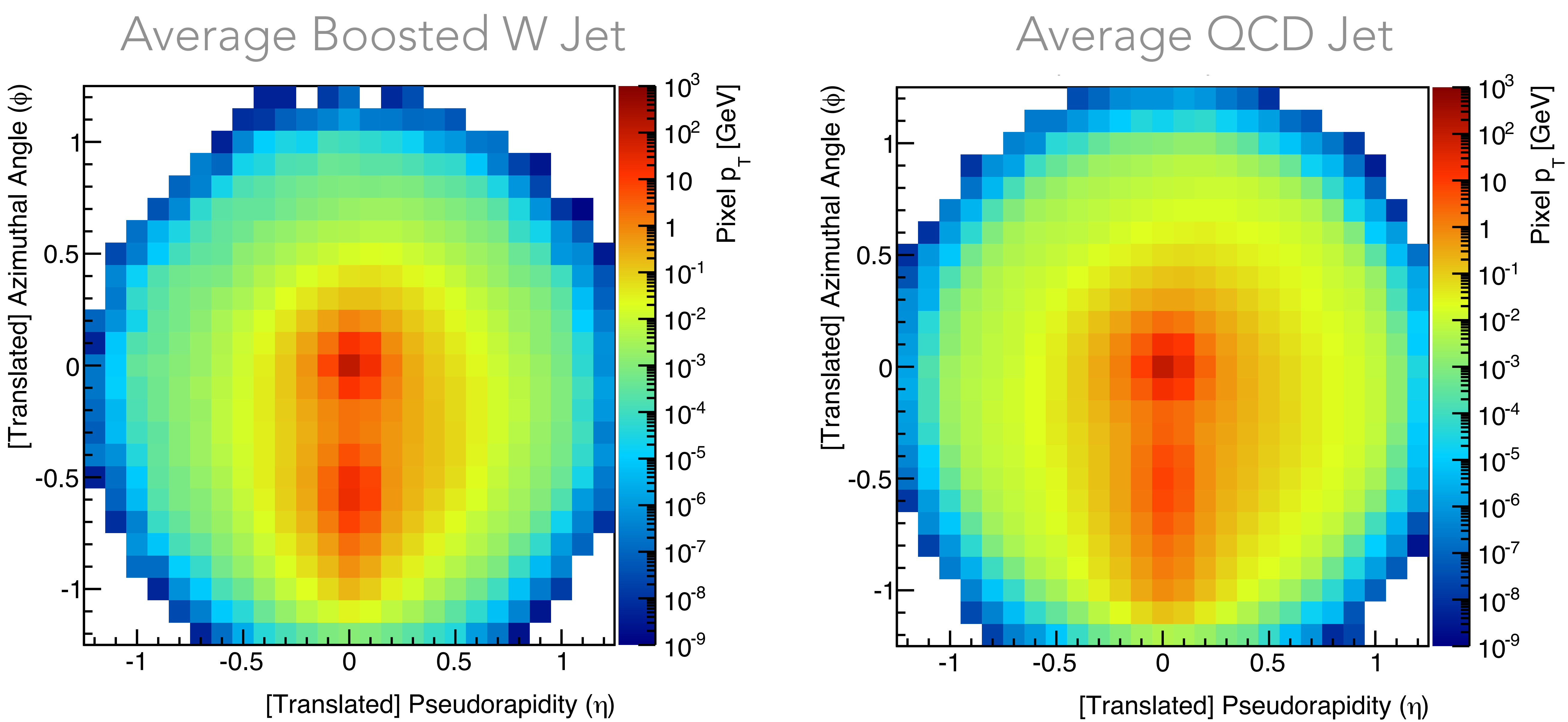


JET IMAGES

Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Barnard, et al arXiv:1609.00607

Recently: deep learning algorithms applied to “jet images”

- based on fast simulation & idealized uniform calorimeter
- preprocessed to recenter (η, φ) & rotated



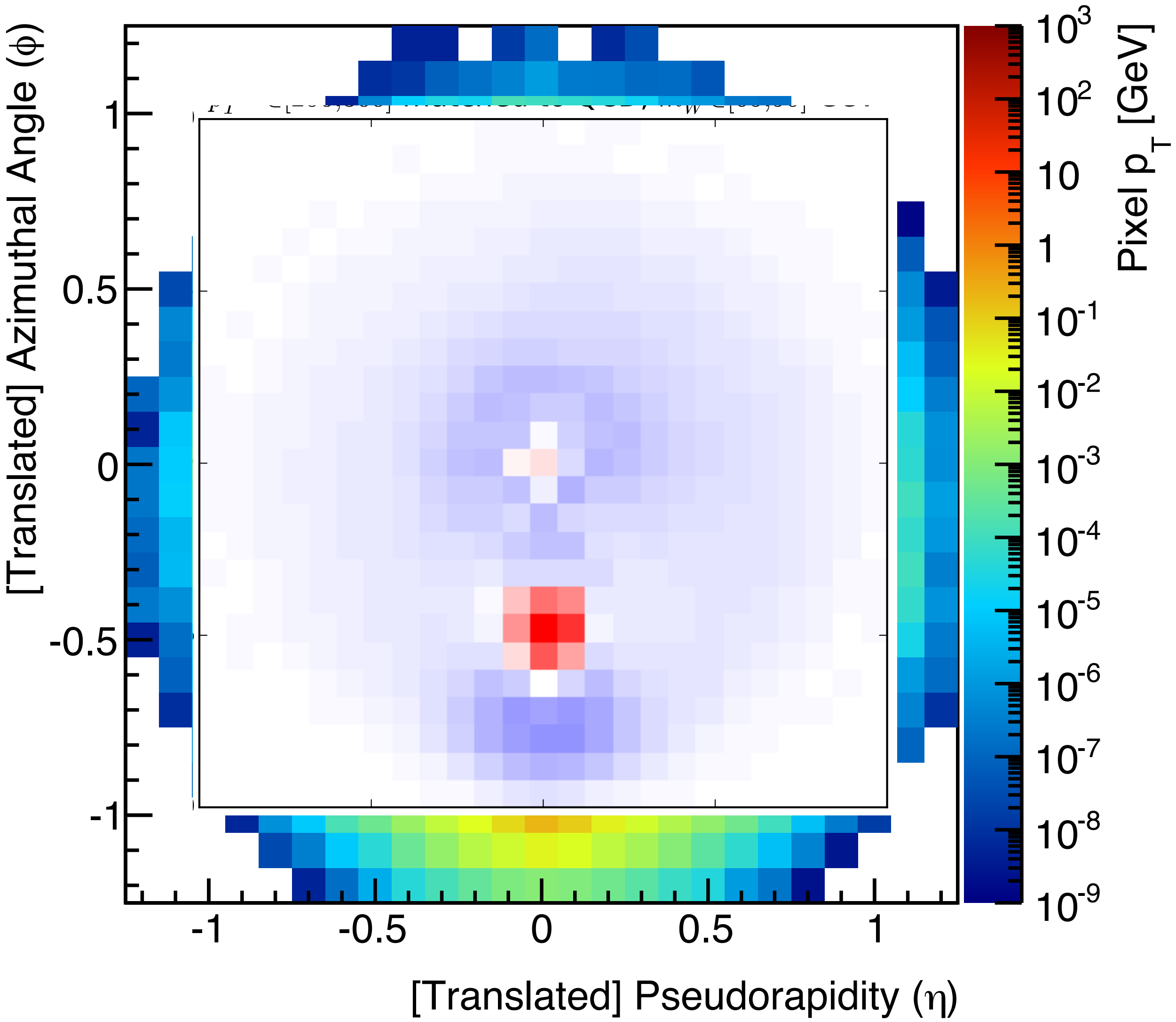
JET IMAGES

Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Barnard, et al arXiv:1609.00607

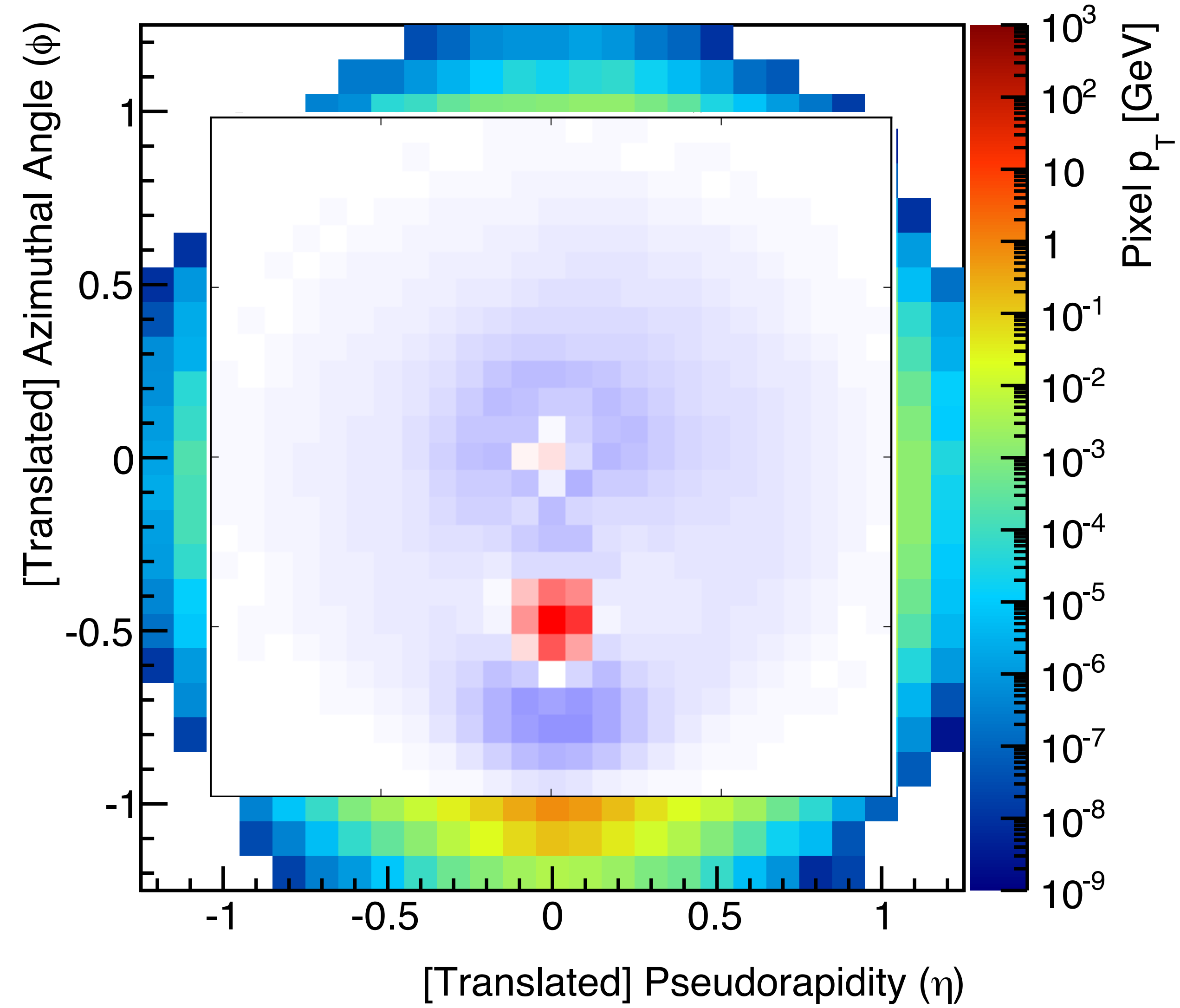
Inspecting the classifier shows parts of image that favor the W→ jj interpretation are consistent with physics intuition

- W-like QCD-like

Average Boosted W Jet



Average QCD Jet

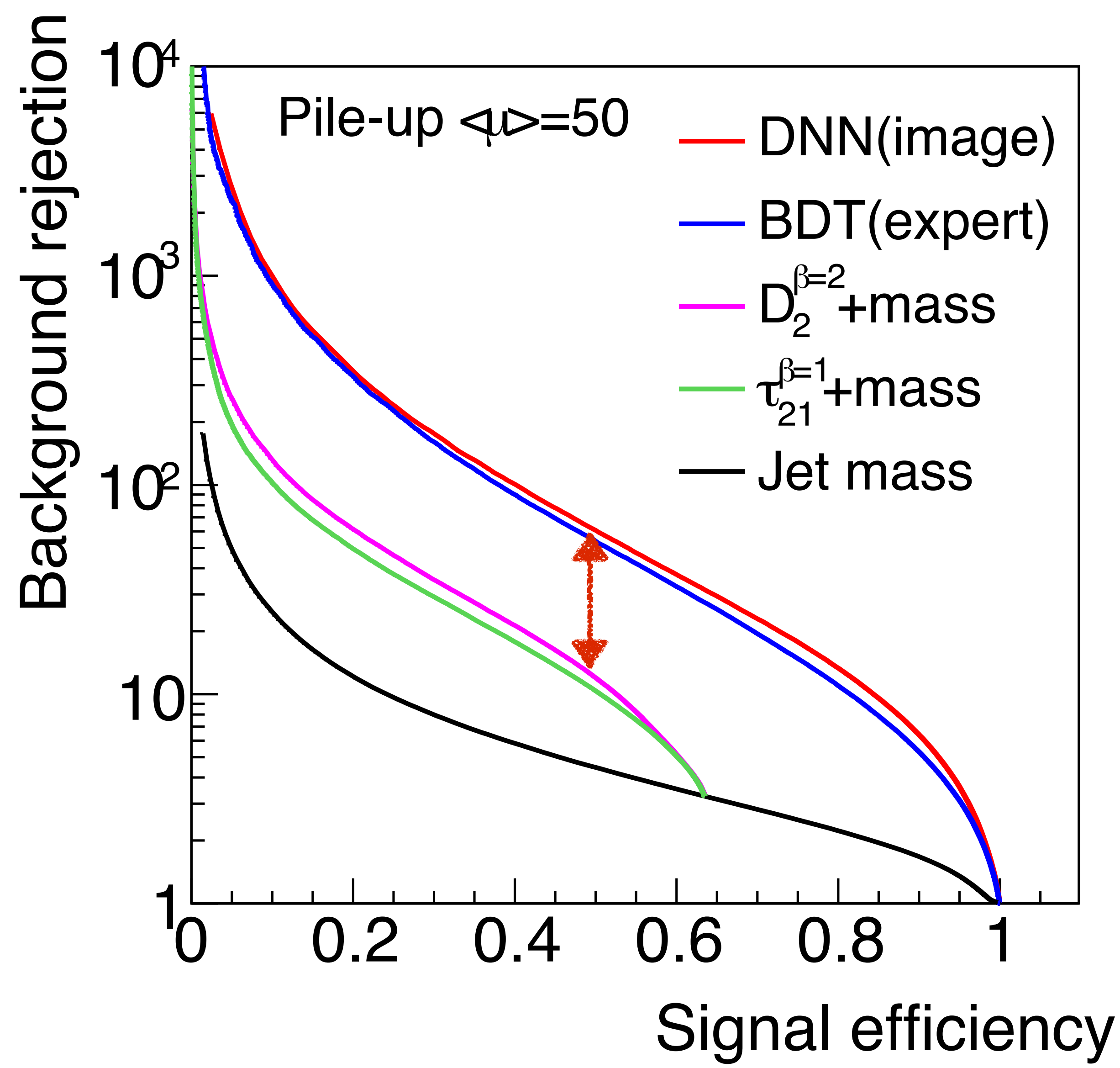


Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Barnard, et al arXiv:1609.00607

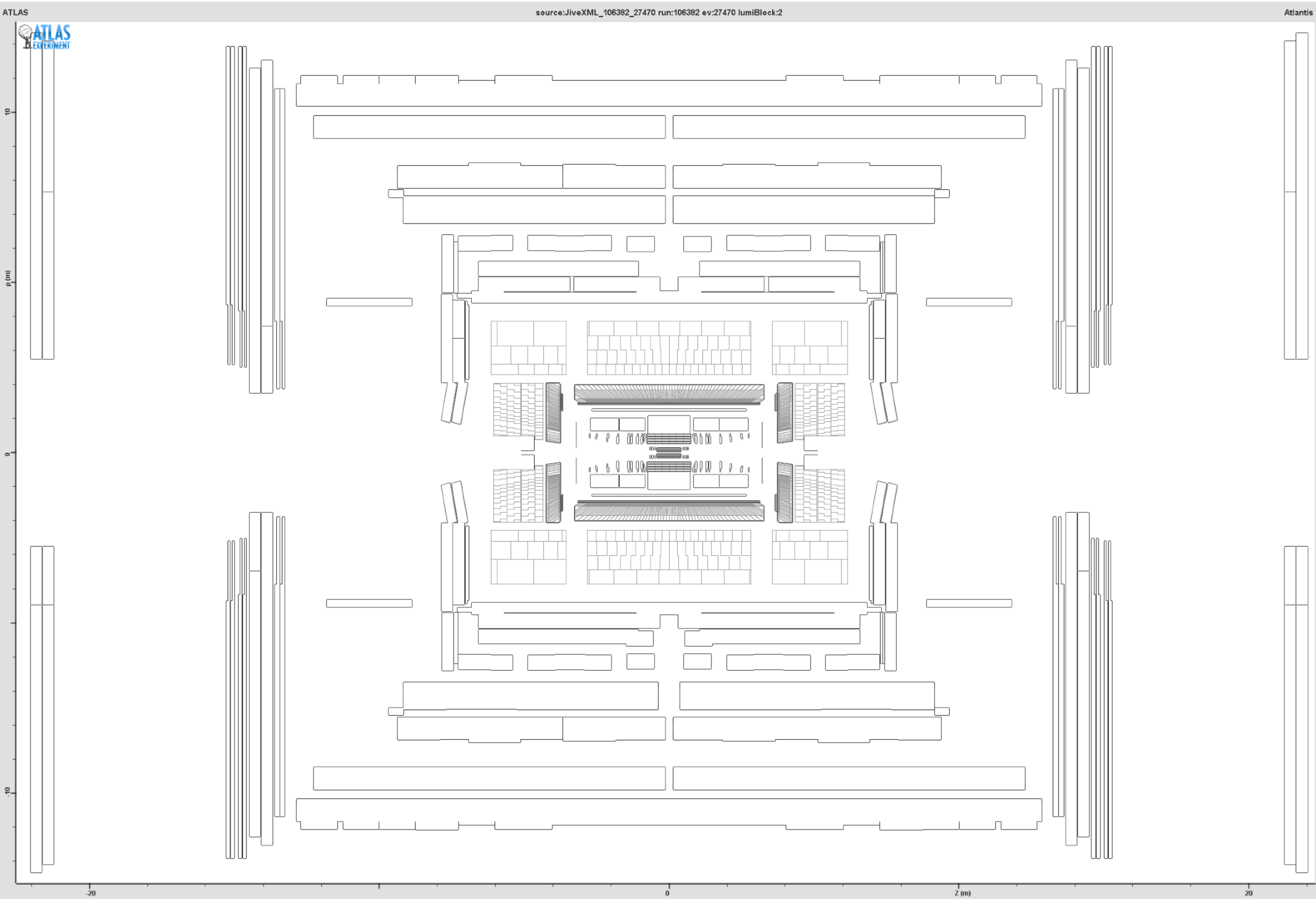
While the DNN shows a significant improvement with respect to the jet mass combined with single theory inspired variable (eg. τ_{21}, D_2), only a small improvement with respect to a BDT using several theory-inspired variables

Other Problems:

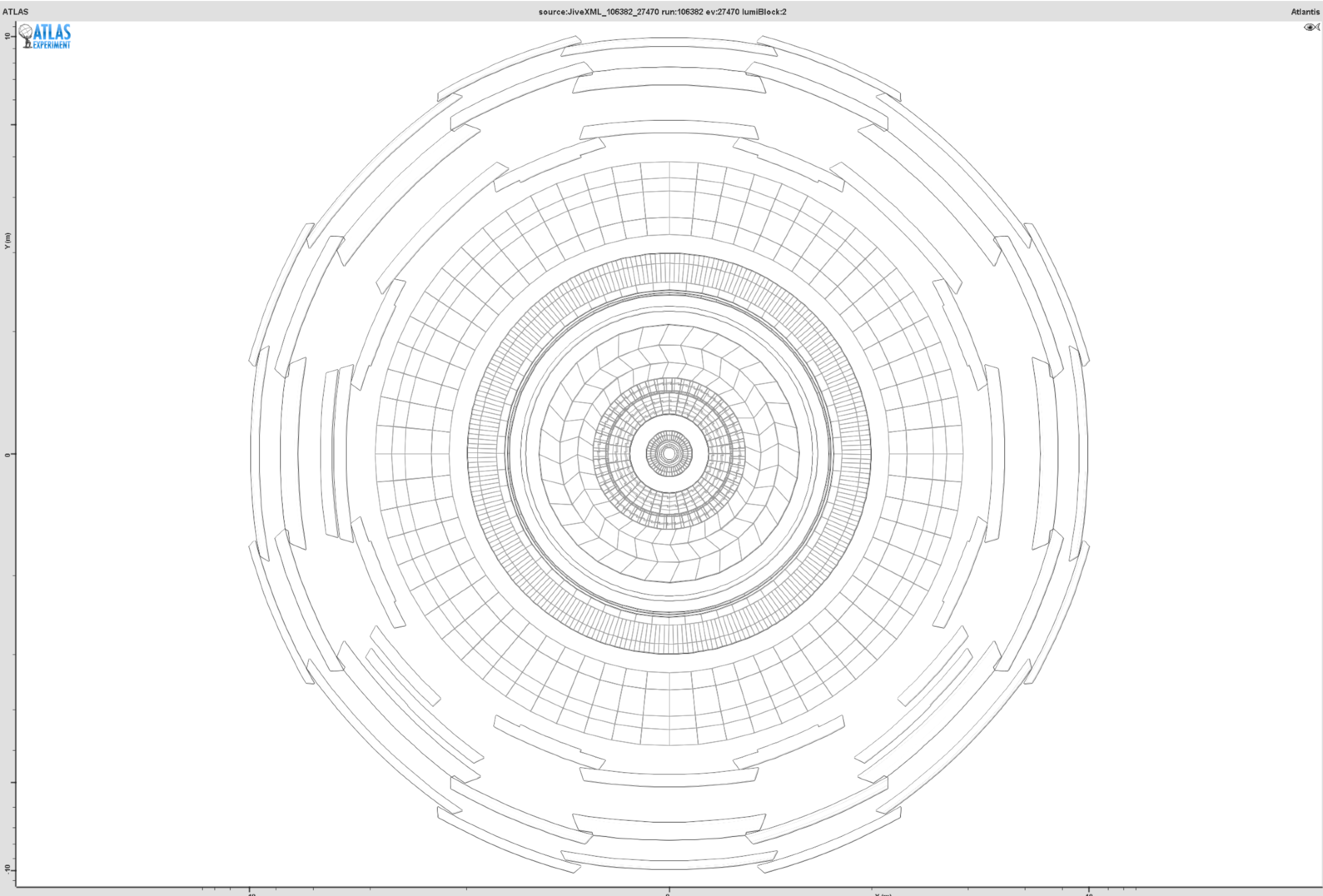
- image-based approach not easily generalized to non-uniform calorimeters
- not easy to extend to tracks, projecting into towers looses information
- theory inspired variables work on set of 4-vectors & have important theoretical properties



NON-UNIFORM GEOMETRY



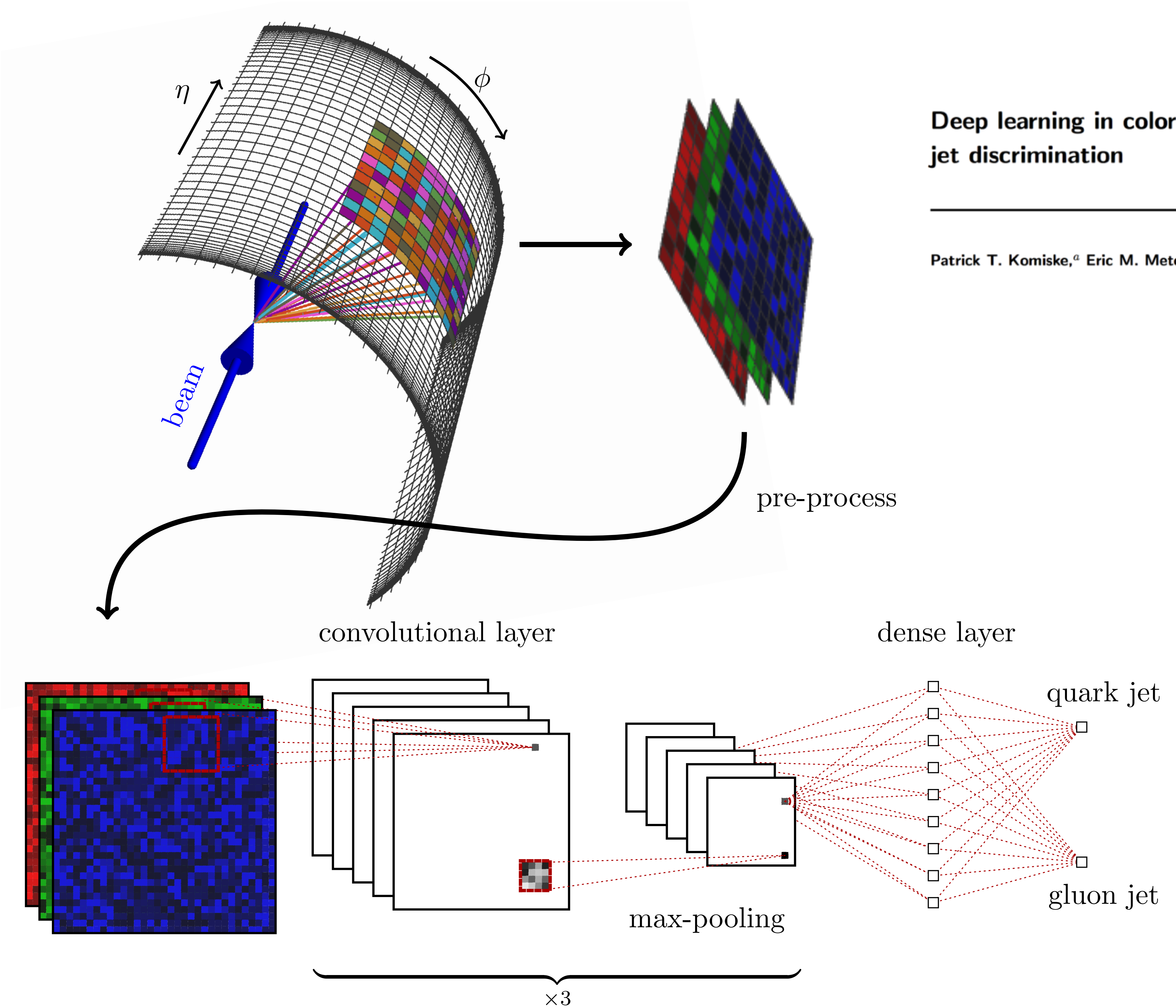
NON-UNIFORM GEOMETRY



JET IMAGES

image: Komiske, Metodiev, Schwartz arxiv:1612.01551
Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Barnard, et al arXiv:1609.00607

"We supplement this construction by adding color to the images, with **red, green and blue** intensities given by the transverse momentum in **charged** particles, transverse momentum in **neutral** particles, and pixel-level charged particle **counts**."



Deep learning in color: towards automated quark/gluon jet discrimination

Patrick T. Komiske,[a] Eric M. Metodiev,[a] and Matthew D. Schwartz[b]

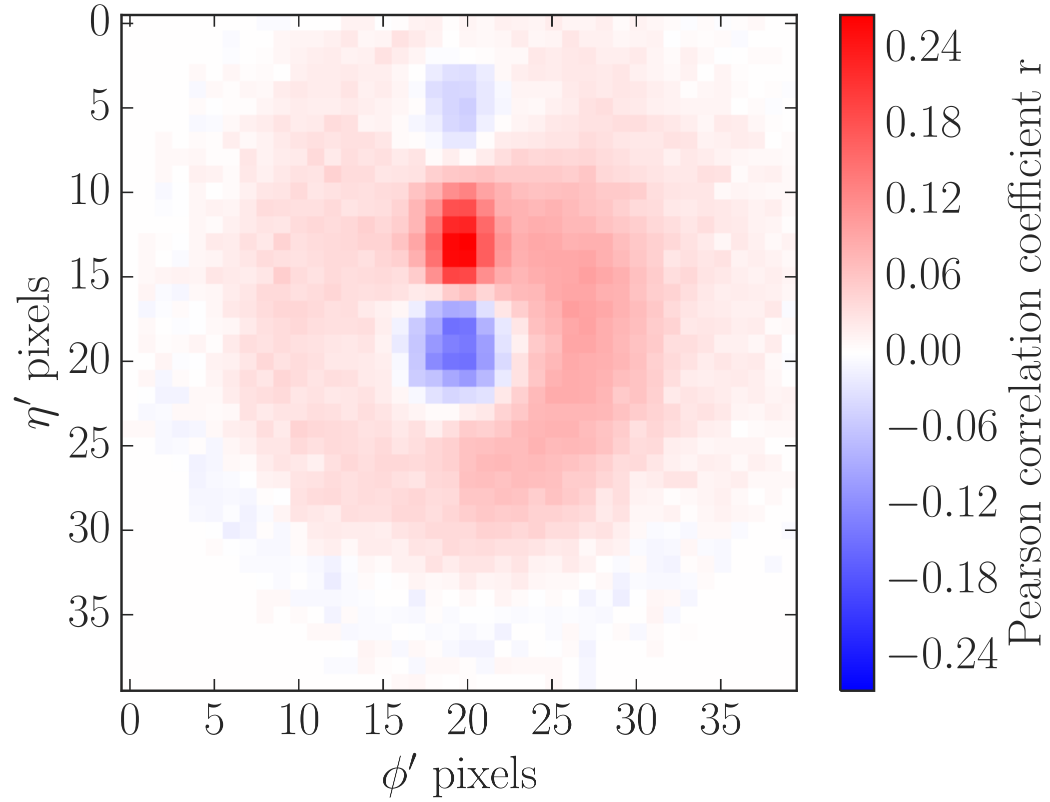
TOP TAGGING

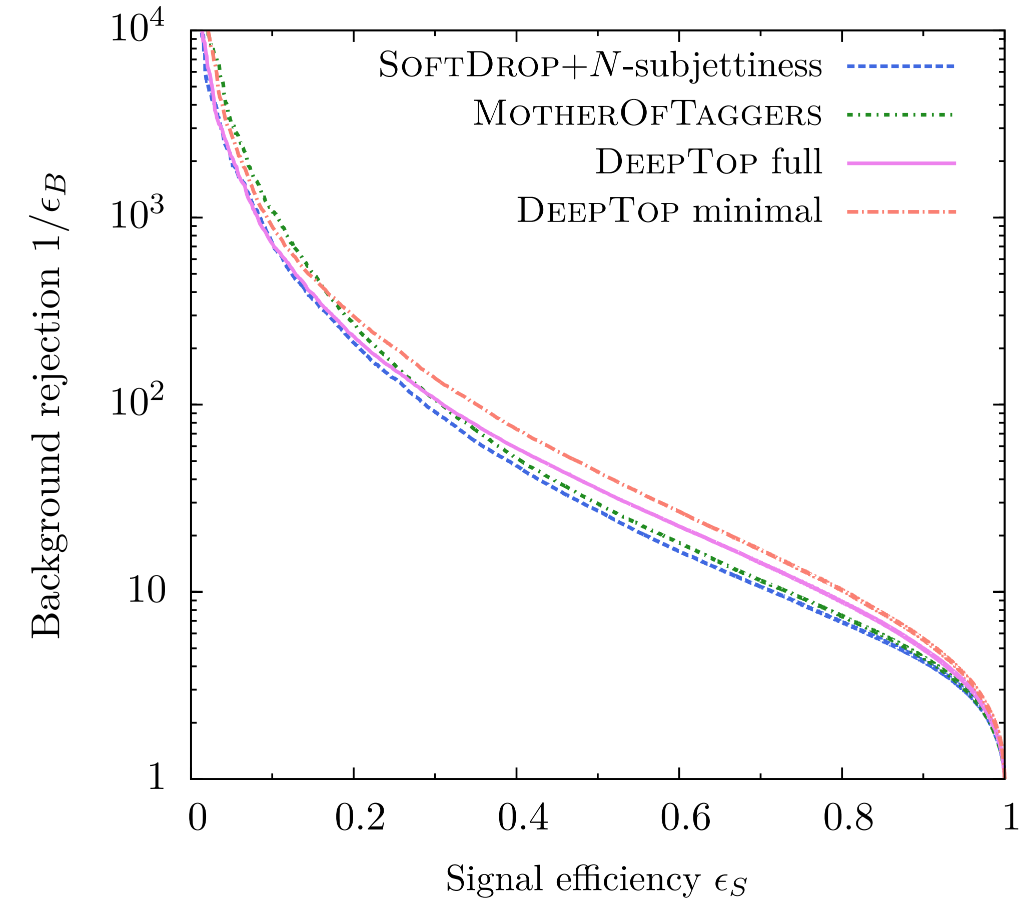
Deep-learning Top Taggers or The End of QCD?

Gregor Kasieczka,[1] Tilman Plehn,[2] Michael Russell,[3] and Torben Schell[2]

Kasieczka, Plehn, et. al.
arXiv:1701.08784
Pearkes, Fedorko, Lister, Gay
arxiv:1704.02124

Similar images ↓ showing which regions correlate with top tag





Again, combining many "expert" / QCD-inspired features ↑ (MotherOfTaggers) does pretty well. Deep network does a little better

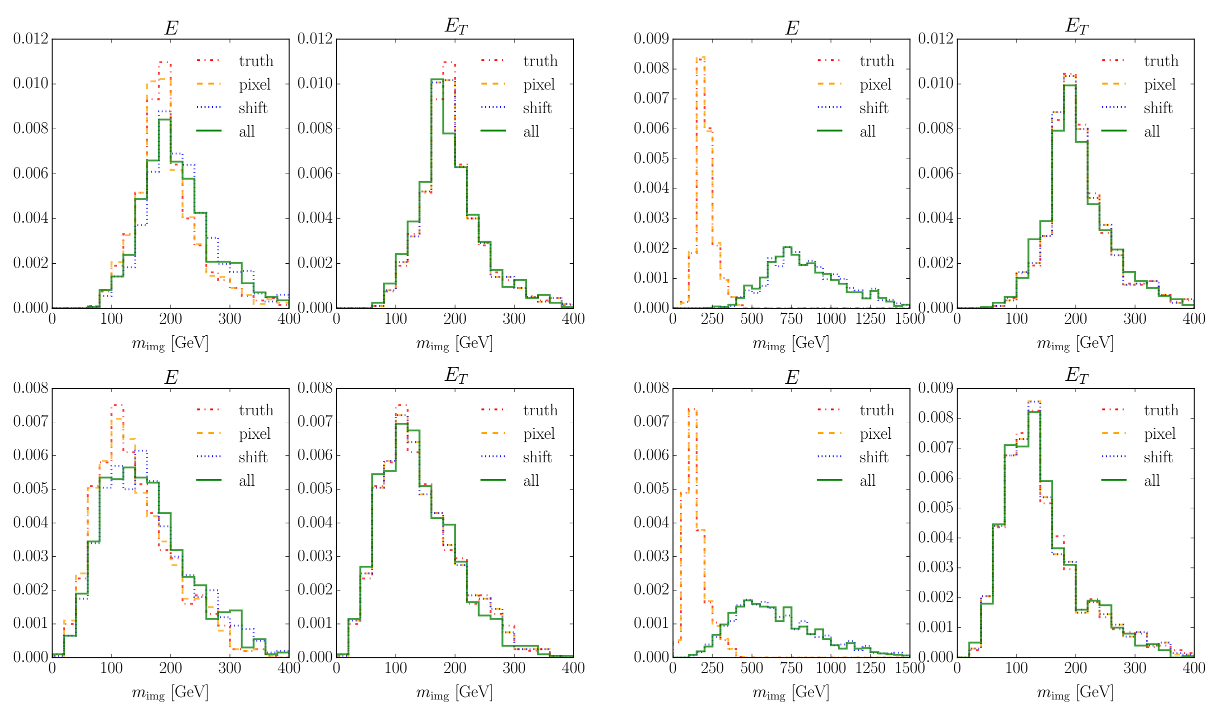


Figure 2. Effect of the preprocessing on the image mass calculated from E-(left) and E_T-images (right) of signal (top) and background(bottom). The right set of plots illustrates the situation for forward jets with $|\eta| > 2$.

← Again, lots of studies to understand how pixilation and pre-processing affects performance

↓ Recent paper using input 4-vectors instead of image

Jet Constituents for Deep Neural Network Based Top Quark Tagging

J. Pearkes, W. Fedorko, A. Lister, C. Gay[1]

[1]*Department of Physics and Astronomy, The University of British Columbia, BC, Canada*

(Dated: April 10, 2017)

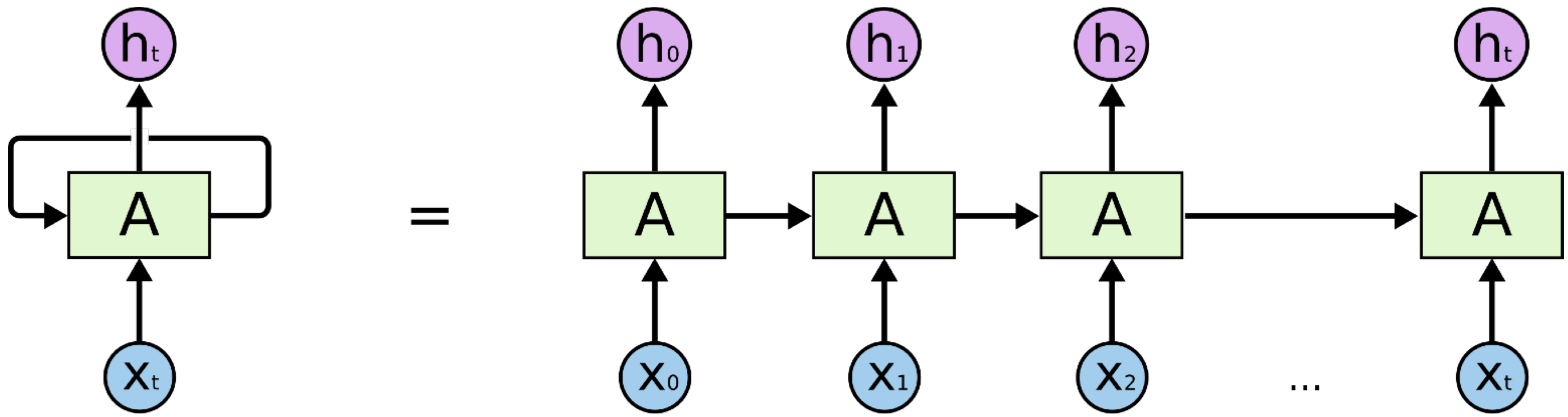
HOW CAN WE IMPROVE?

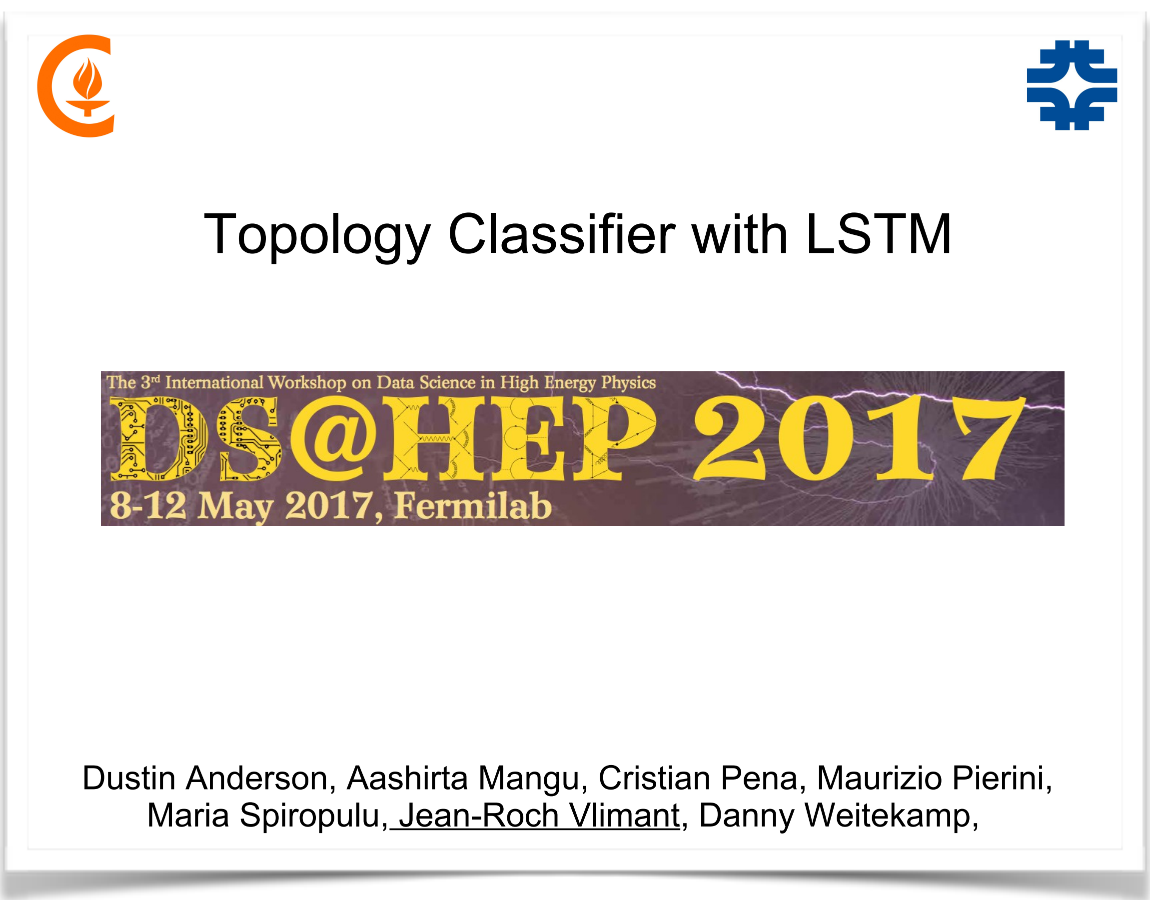
Image based approaches are doing well, but….

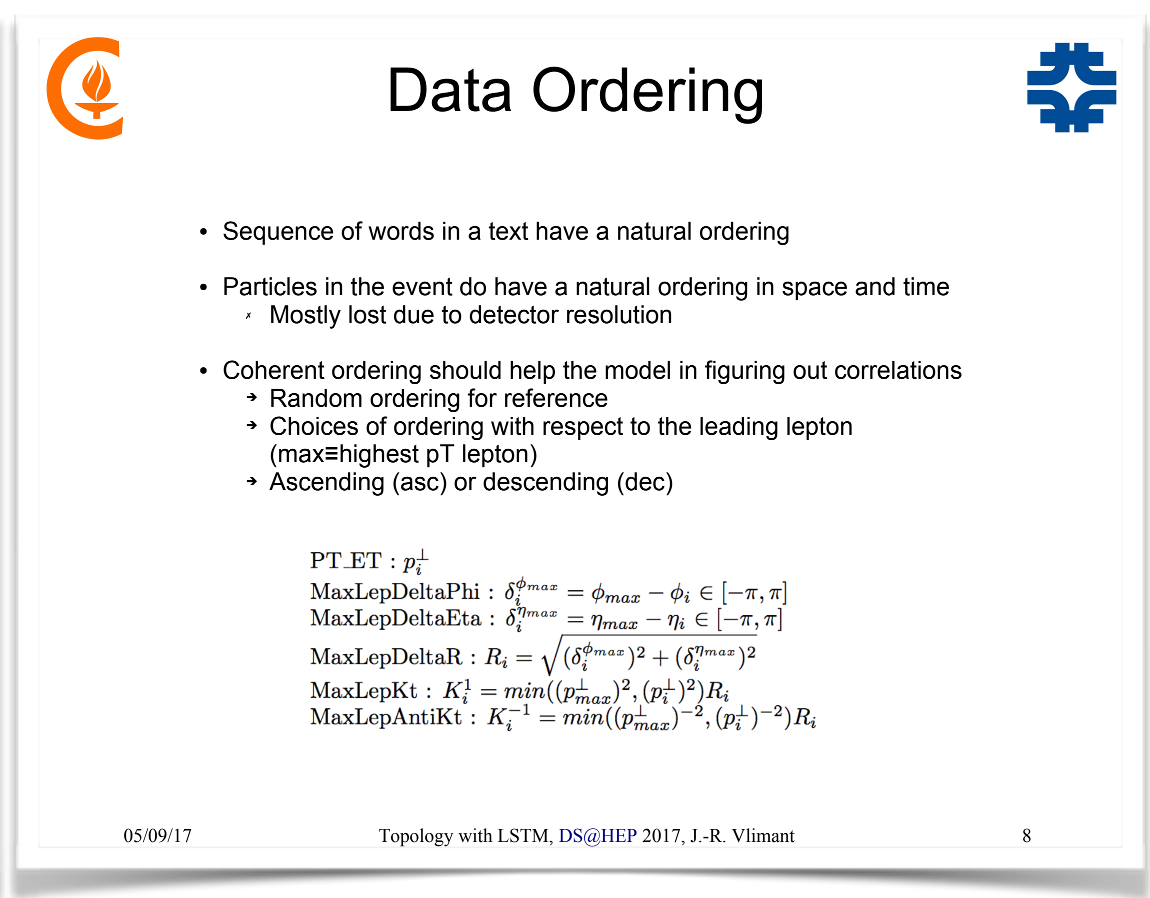
- would be nice to be able to work with a variable length set of 4-momenta
  - avoid discretization (eg. use tracks, particle flow, clusters as input)
  - avoid pre-processing into a regular-grid (eg. non-uniform calorimeters)
  - avoid representing empty pixels (sparse input)
- would be nice if classifier had nice theoretical properties
  - infrared & collinear safety, robustness to pileup, etc.
- would be nice to be more data efficient, most image-based networks use a LOT of training data.

Recurrent Neural Network (acting on a variable-length sequence)

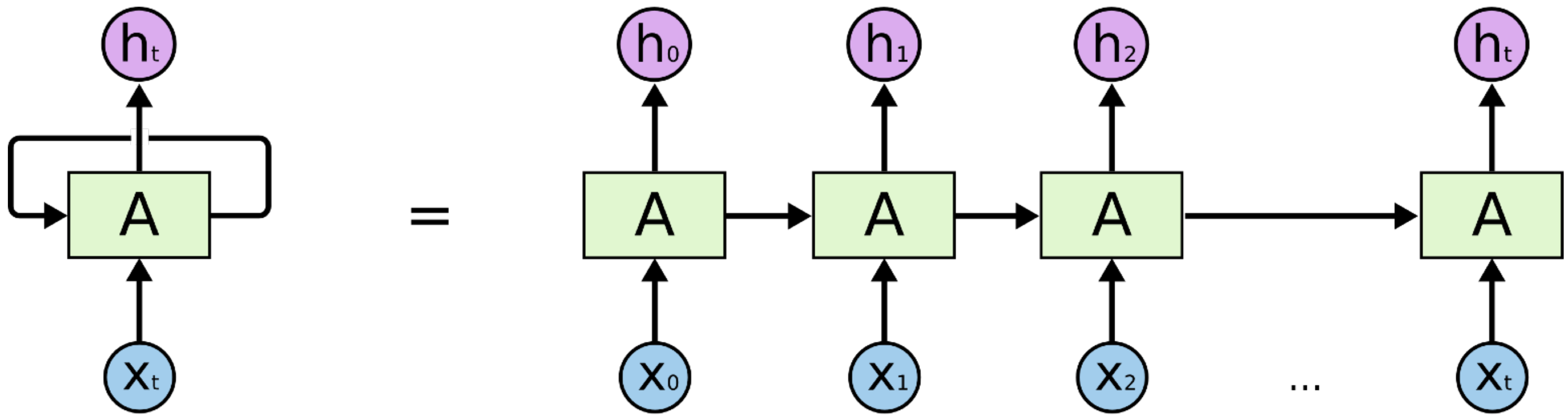
see eg. Guest, Collado, et al in arxiv:1607.08633.



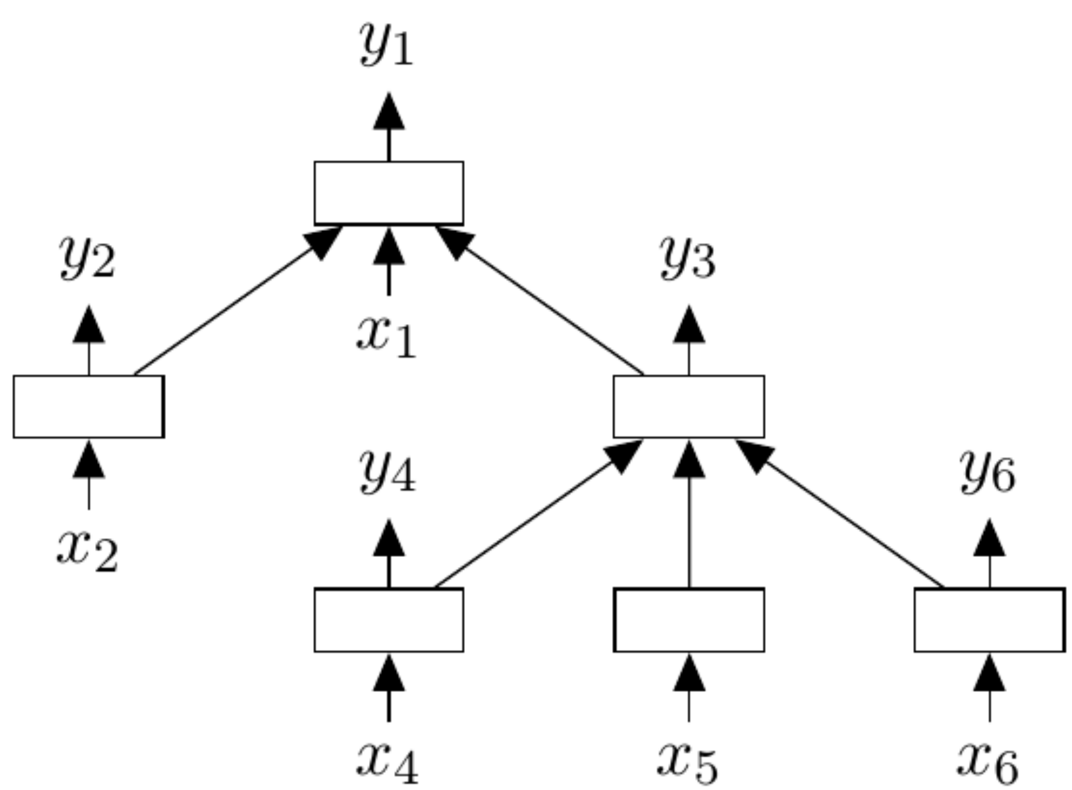




Recurrent Neural Network (acting on a variable-length sequence)
see eg. Guest, Collado, et al in arxiv:1607.08633.

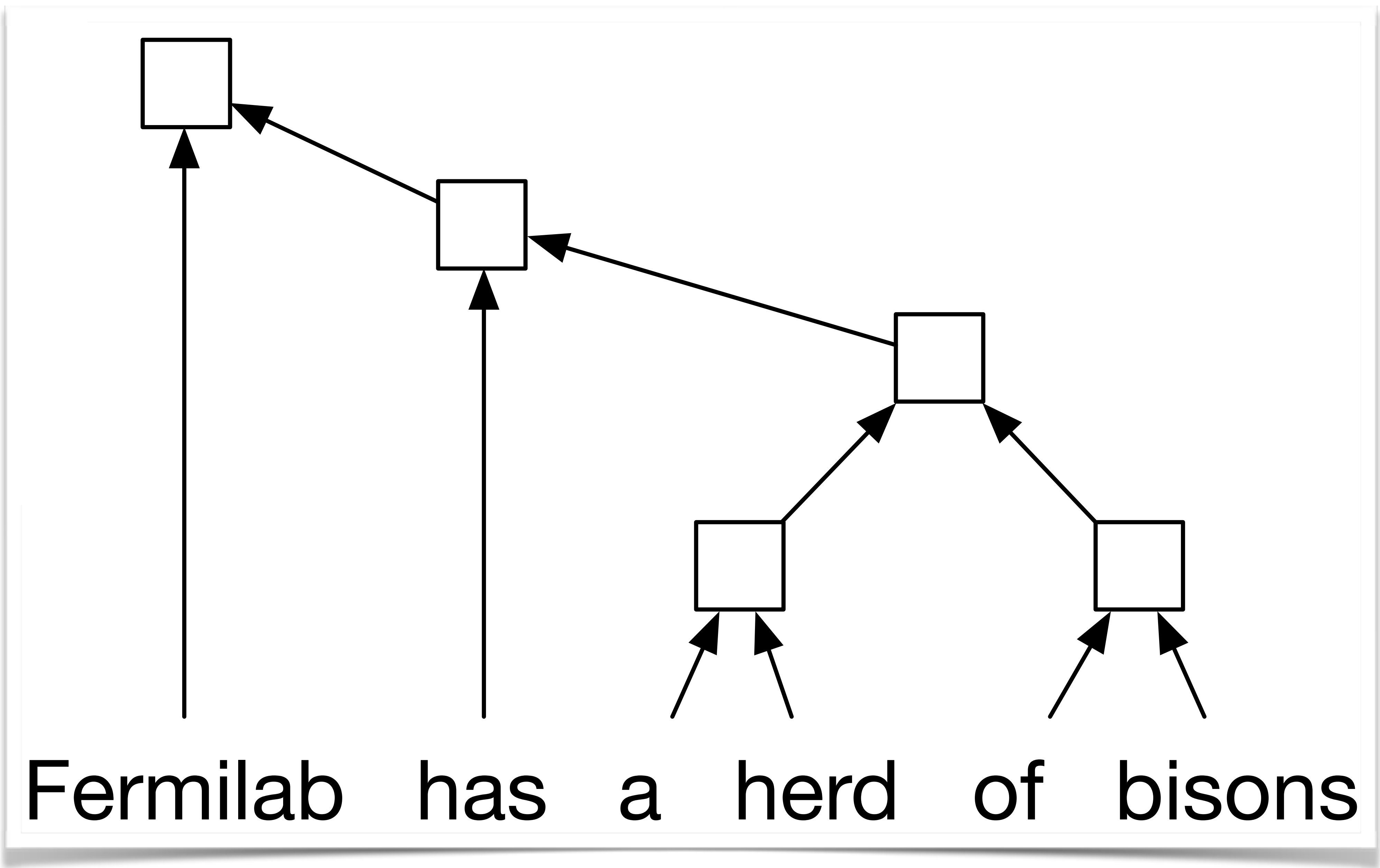


Generalization: Recursive Neural Network



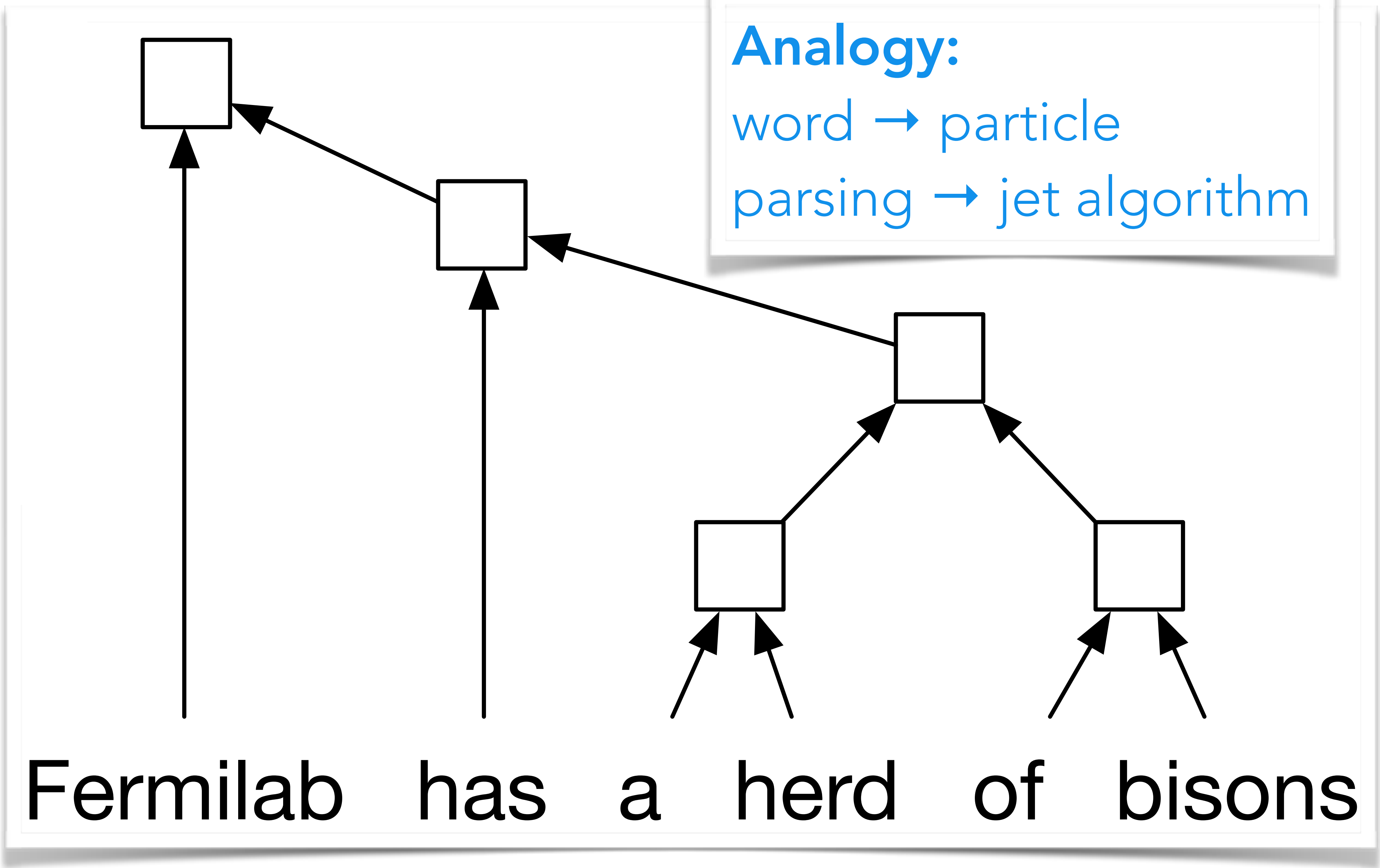
Recursive Neural Networks showing great performance for Natural Language Processing tasks

- neural network's topology given by parsing of sentence!



Recursive Neural Networks showing great performance for Natural Language Processing tasks

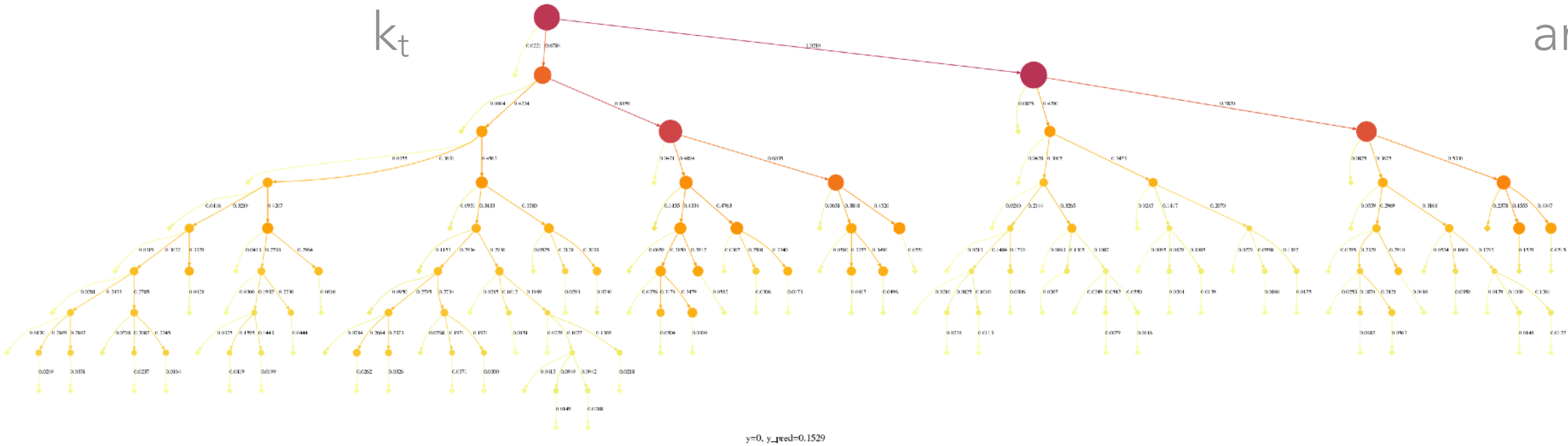
- neural network's topology given by parsing of sentence!



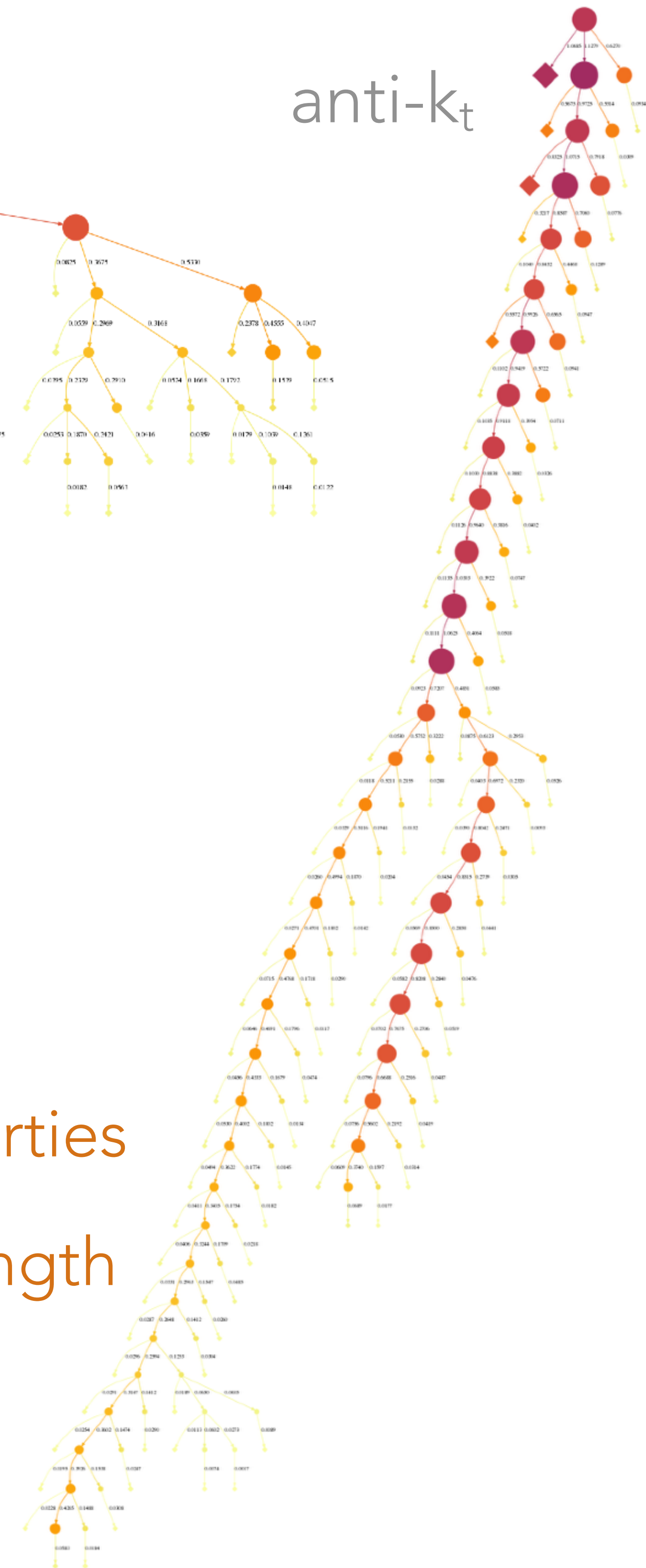
Analogy:
word → particle
parsing → jet algorithm

QCD-INSPIRED RECURSIVE NEURAL NETWORKS

k_t



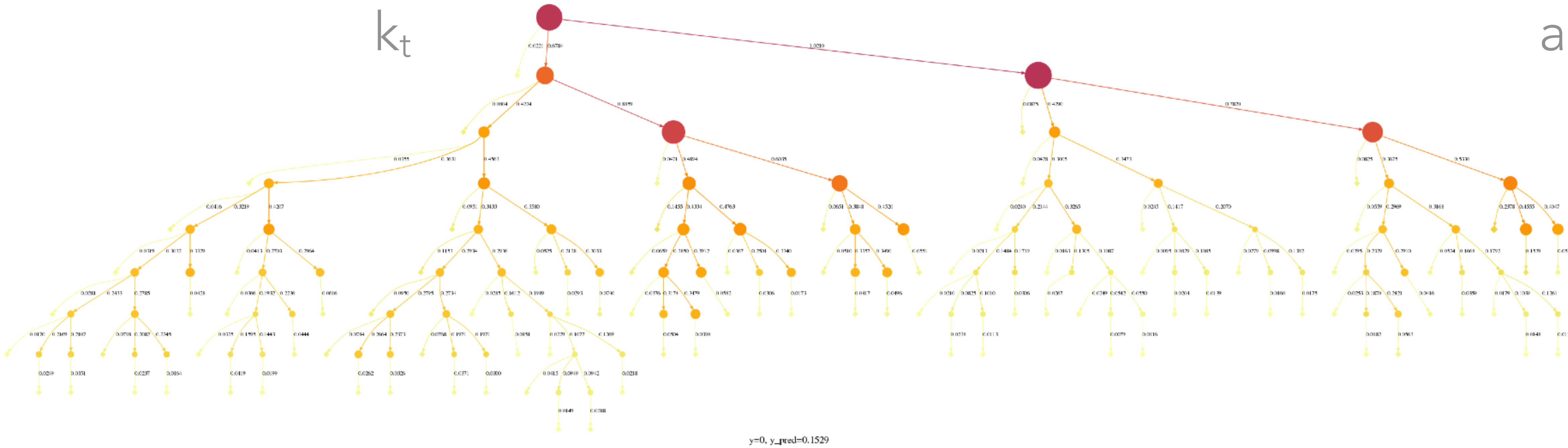
anti-k_t

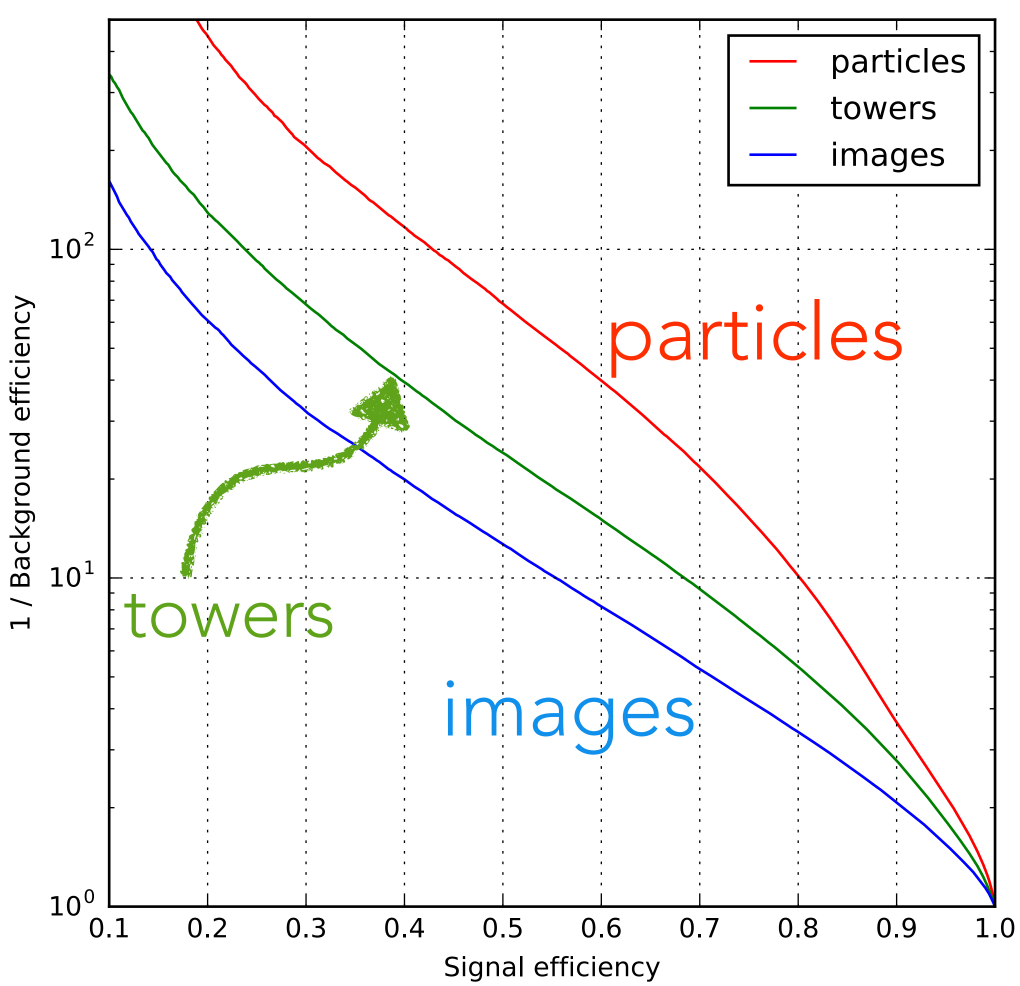


Work with Gilles Louppe, Kyunghyun Cho, Cyril Becot (arXiv:1702.00748)

- Use sequential recombination jet algorithms to provide network topology (on a per-jet basis)
- path towards ML models with good physics properties
- Top node of recursive network provides a fixed-length **embedding** of a jet that can be fed to a classifier

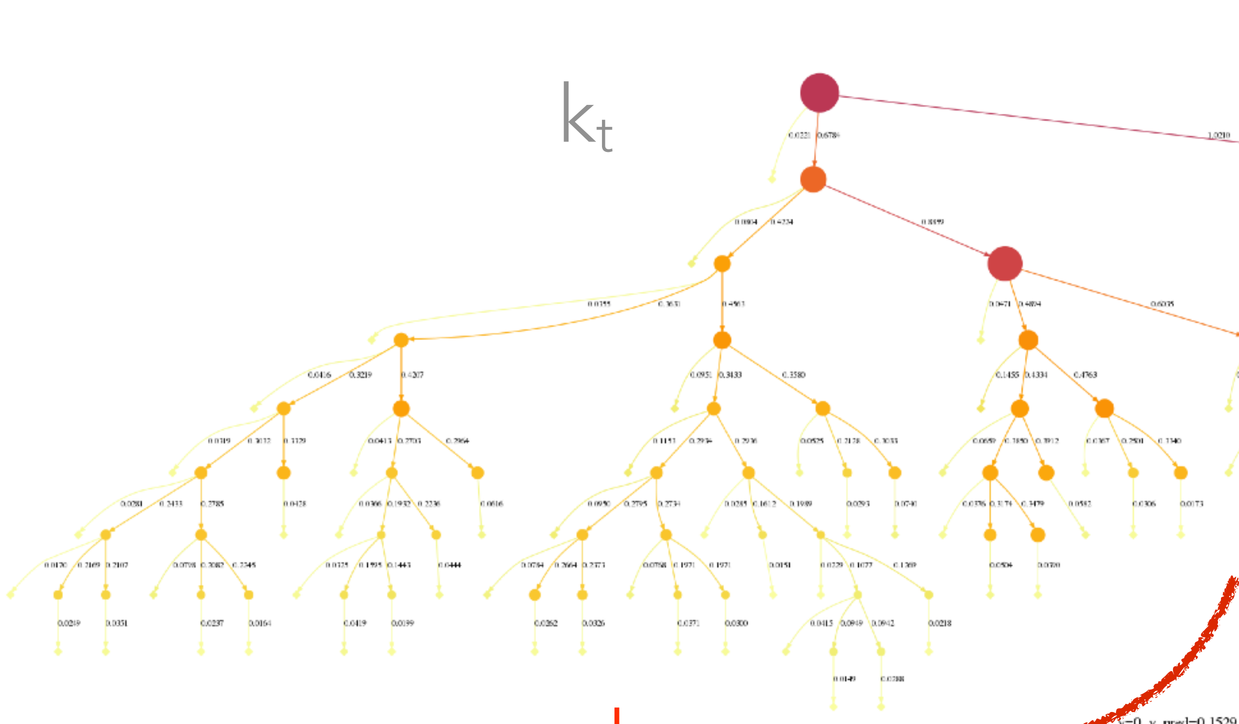
QCD-INSPIRED RECURSIVE NEURAL NETWORKS



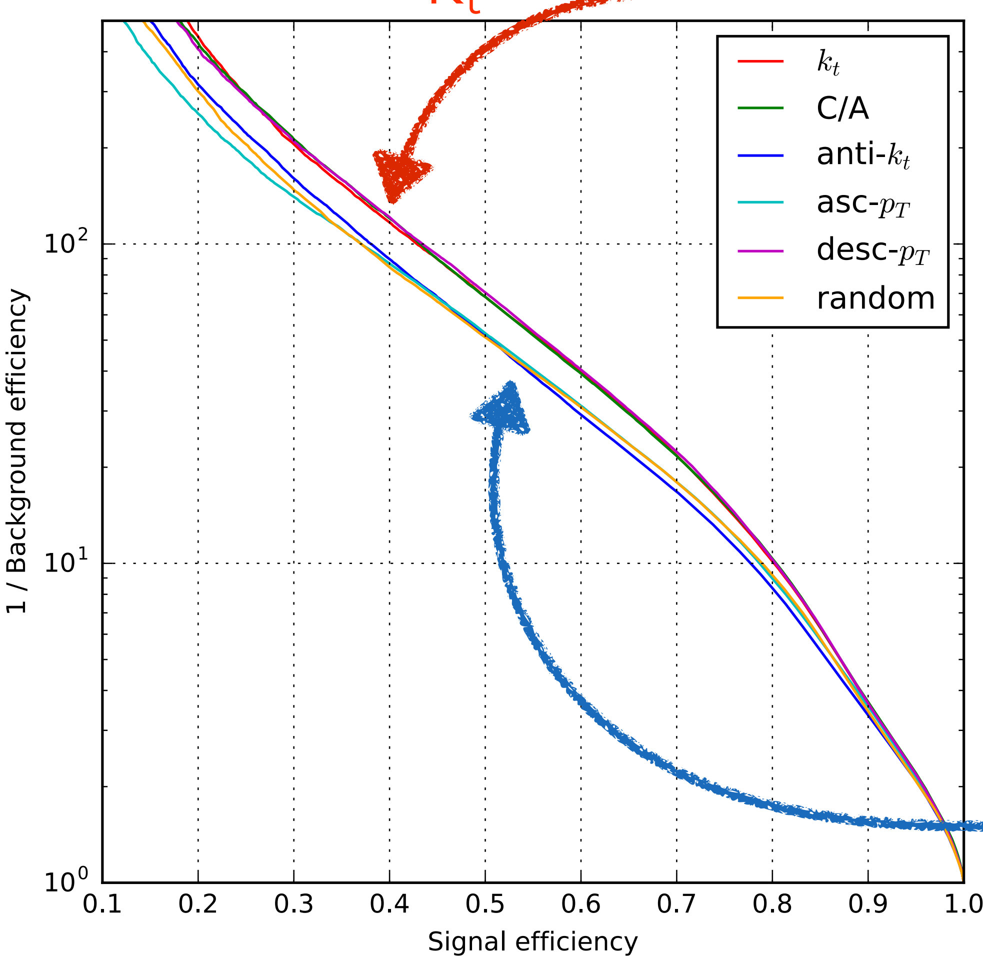


- W-jet tagging example using data from Dawe, et al arXiv:1609.00607
- down-sampling by projecting into images looses information
- RNN needs much less data to train!

QCD-INSPIRED RECURSIVE NEURAL NETWORKS

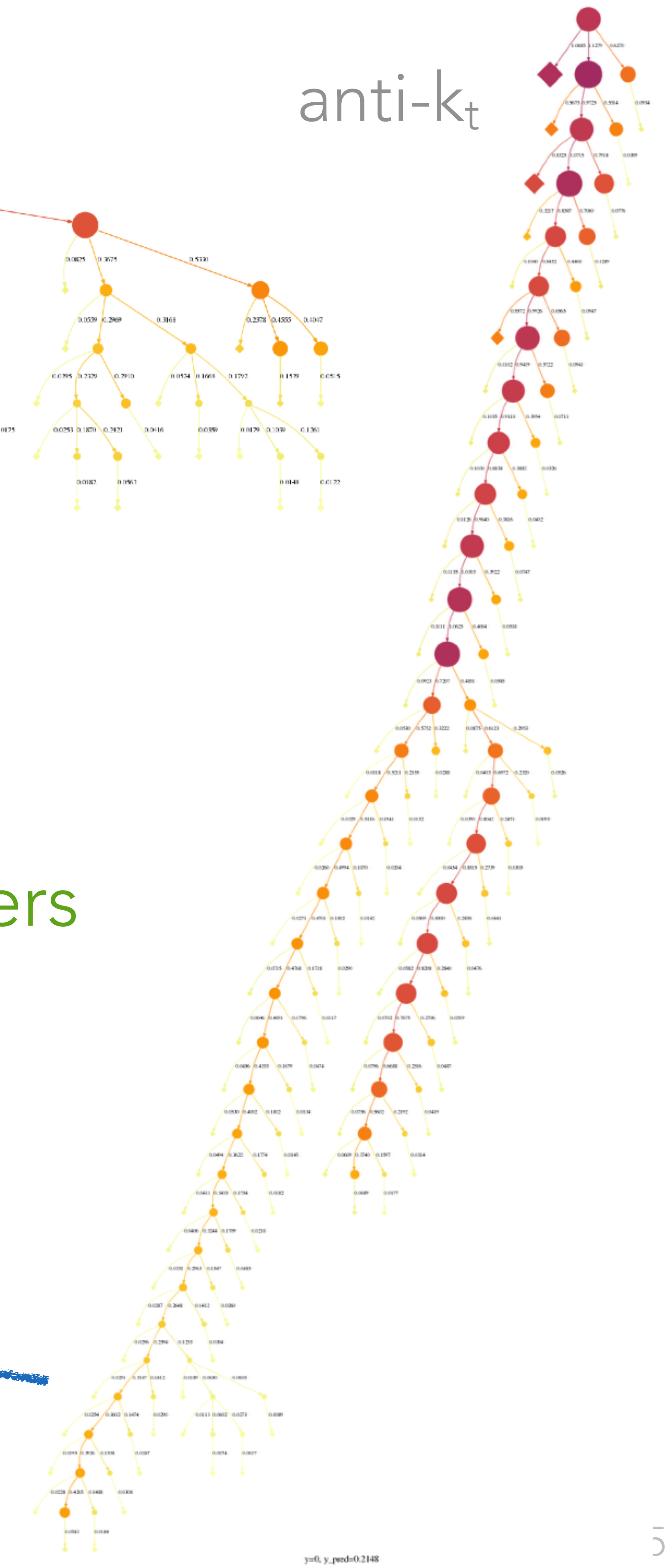


k_t



- choice of jet algorithm matters
- GRU "gating" improves performance

anti-k_t



JET-LEVEL CLASSIFICATION RESULTS

TABLE I. Summary of jet classification performance for several approaches applied either to particle-level inputs or towers from a DELPHES simulation.

Input	Architecture	ROC AUC	$R_{\epsilon=50\%}$
Projected into images			
towers	MaxOut	**0.8418**	–
towers	k_t	0.8321 ± 0.0025	**12.7 ± 0.4**
towers	k_t (gated)	0.8277 ± 0.0028	12.4 ± 0.3
Without image preprocessing			
towers	τ_{21}	0.7644	6.79
towers	mass + τ_{21}	0.8212	11.31
towers	k_t	0.8807 ± 0.0010	24.1 ± 0.6
towers	C/A	0.8831 ± 0.0010	24.2 ± 0.7
towers	anti-k_t	0.8737 ± 0.0017	22.3 ± 0.8
towers	asc-p_T	0.8835 ± 0.0009	**26.2 ± 0.7**
towers	desc-p_T	**0.8838 ± 0.0010**	25.1 ± 0.6
towers	random	0.8704 ± 0.0011	20.4 ± 0.3
particles	k_t	0.9185 ± 0.0006	68.3 ± 1.8
particles	C/A	**0.9192 ± 0.0008**	68.3 ± 3.6
particles	anti-k_t	0.9096 ± 0.0013	51.7 ± 3.5
particles	asc-p_T	0.9130 ± 0.0031	52.5 ± 7.3
particles	desc-p_T	0.9189 ± 0.0009	**70.4 ± 3.6**
particles	random	0.9121 ± 0.0008	51.1 ± 2.0
With gating (see Appendix A)			
towers	k_t	0.8822 ± 0.0006	25.4 ± 0.4
towers	C/A	0.8861 ± 0.0014	26.2 ± 0.8
towers	anti-k_t	0.8804 ± 0.0010	24.4 ± 0.4
towers	asc-p_T	0.8849 ± 0.0012	27.2 ± 0.8
towers	desc-p_T	**0.8864 ± 0.0007**	**27.5 ± 0.6**
towers	random	0.8751 ± 0.0029	22.8 ± 1.2
particles	k_t	0.9195 ± 0.0009	74.3 ± 2.4
particles	C/A	**0.9222 ± 0.0007**	81.8 ± 3.1
particles	anti-k_t	0.9156 ± 0.0012	68.3 ± 3.2
particles	asc-p_T	0.9137 ± 0.0046	54.8 ± 11.7
particles	desc-p_T	0.9212 ± 0.0005	**83.3 ± 3.1**
particles	random	0.9106 ± 0.0035	50.7 ± 6.7

When working on images:

- recursive network has similar performance to previous approaches

Improved performance when working with calo towers without image pre-processing

- loss of information depends on details of calorimeter, pixelation, etc.

Working on truth-level particles led to a significant improvement

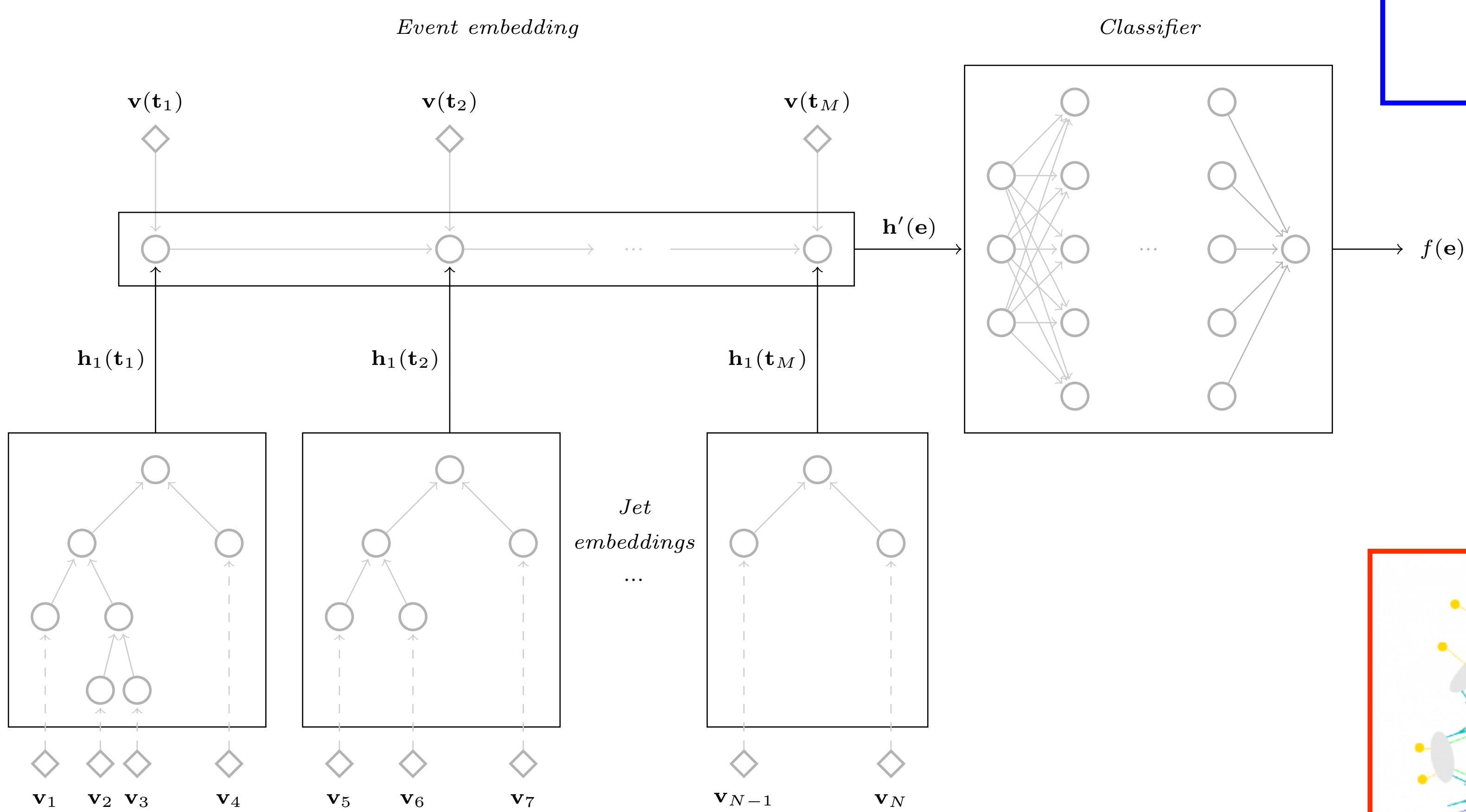
- generically expect information from tracking, particle flow, etc. to be somewhere between towers and truth particle-level

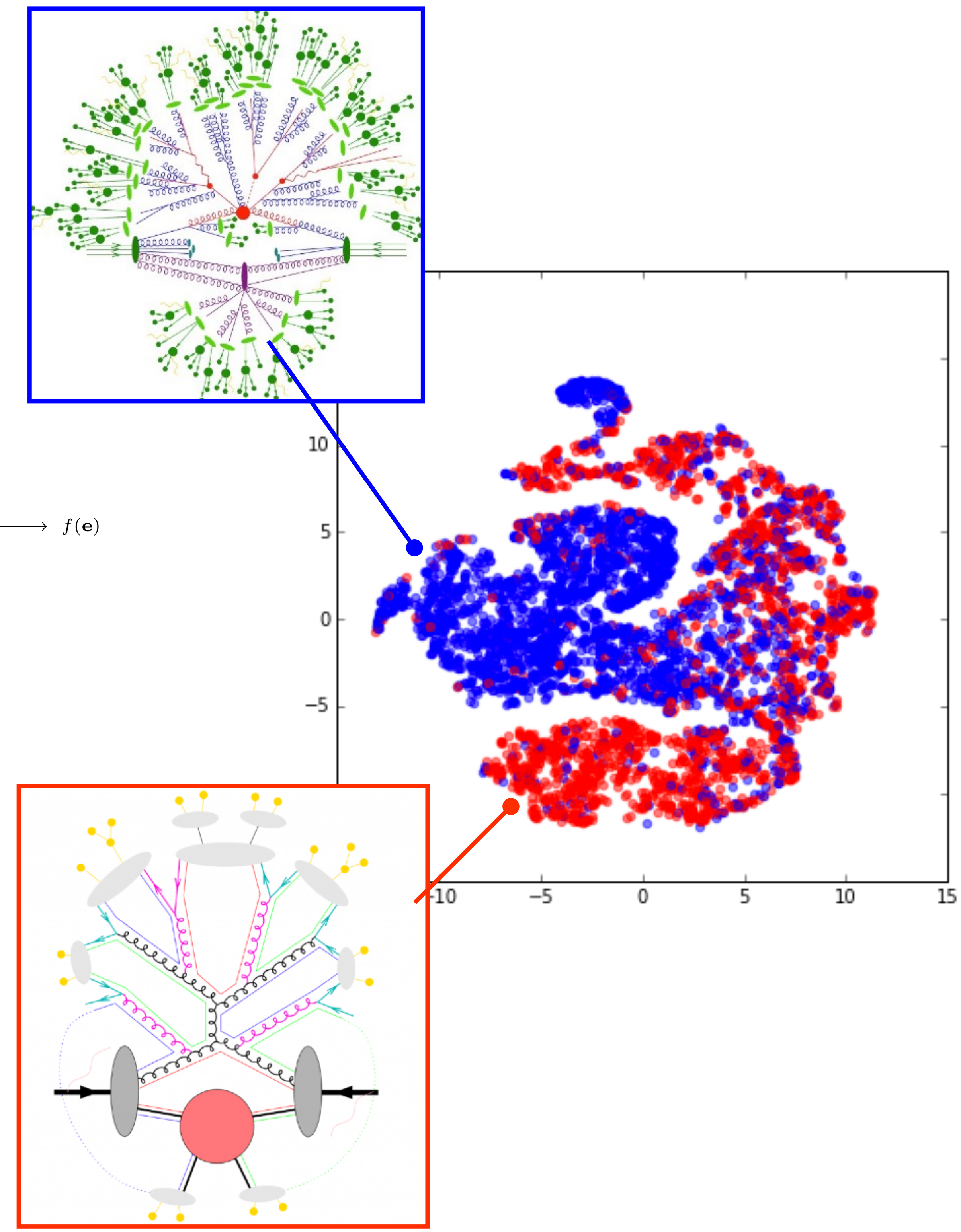
From Jets to Events

WE WILL USE ALL THE PARTICLES IN THE EVENT AS INPUT TO THE CLASSIFIER!

Jointly optimize jet embedding → event embedding → classifier

It scales!





We considered pp → W'(700) → W(→ J) Z(→J)

- compared only jet-level 4-momentum $v(t_j)$ to adding jet-embedding h_j
  - adding jet embedding is much better (provides jet tagging info)
- compared RNN that works on jet-level embeddings to an RNN that simply processes all particles in the event
  - jet clustering & jet embeddings help a lot

TABLE III. Summary of event classification performance. Best results are achieved through nested recurrence over the jets and over their constituents, as motivated by QCD.

Input	ROC AUC	$R_{\epsilon=80\%}$
Hardest jet		
$\mathbf{v}(\mathbf{t}_j)$	0.8909 ± 0.0007	5.6 ± 0.0
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(k_t)}$	**0.9602 ± 0.0004**	**26.7 ± 0.7**
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(\text{desc}-p_T)}$	0.9594 ± 0.0010	25.6 ± 1.4
2 hardest jets		
$\mathbf{v}(\mathbf{t}_j)$	0.9606 ± 0.0011	21.1 ± 1.1
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(k_t)}$	0.9866 ± 0.0007	156.9 ± 14.8
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(\text{desc}-p_T)}$	**0.9875 ± 0.0006**	**174.5 ± 14.0**
5 hardest jets		
$\mathbf{v}(\mathbf{t}_j)$	0.9576 ± 0.0019	20.3 ± 0.9
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(k_t)}$	0.9867 ± 0.0004	152.8 ± 10.4
$\mathbf{v}(\mathbf{t}_j)$, $\mathbf{h}_j^{\text{jet}(\text{desc}-p_T)}$	**0.9872 ± 0.0003**	**167.8 ± 9.5**
No jet clustering, desc-p_T on $\mathbf{v}_i$		
$i = 1$	0.6501 ± 0.0023	1.7 ± 0.0
$i = 1, \ldots, 50$	**0.8925 ± 0.0079**	**5.6 ± 0.5**
$i = 1, \ldots, 100$	0.8781 ± 0.0180	4.9 ± 0.6
$i = 1, \ldots, 200$	0.8846 ± 0.0091	5.2 ± 0.5
$i = 1, \ldots, 400$	0.8780 ± 0.0132	4.9 ± 0.5

MISC / OTHER THINGS WE TRIED

Average scores reported include uncertainty estimates that come from training 30 models with distinct initial random seeds.

We tried a "stereo" embedding that used both kt and anti-kt, but no significant gain in performance

- want to optimize over the space of sequential recombination jet algorithms… but that's not differentiable in this setup.

We transferred activations learned in one topology to another and saw significant loss in performance.

- not surprising, but demonstrates activations aren't generic

We extended representation of particles from 4-momentum only to also include charge & EM/Had info from Delphes particle flow block.

- At level of Delphes simulation, not much difference, but important point is can extend to "particle embedding". Path towards end-to-end learning.

Theoretical considerations, Systematics, & Jet Grooming

One of the primary concerns in the literature constructing jet-tagging observables is that they are theoretically well-behaved. For instance, physicists want observables to be **infrared and collinear safe**.

We compared nominal results to perturbed samples where we applied collinear splits or added soft radiation.

- QCD-inspired networks are more stable (have less variance) than networks based on simple p_T ordering.
- Does this outweigh small gain in nominal performance?

TABLE II. 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 $-5 < \eta < 5$.

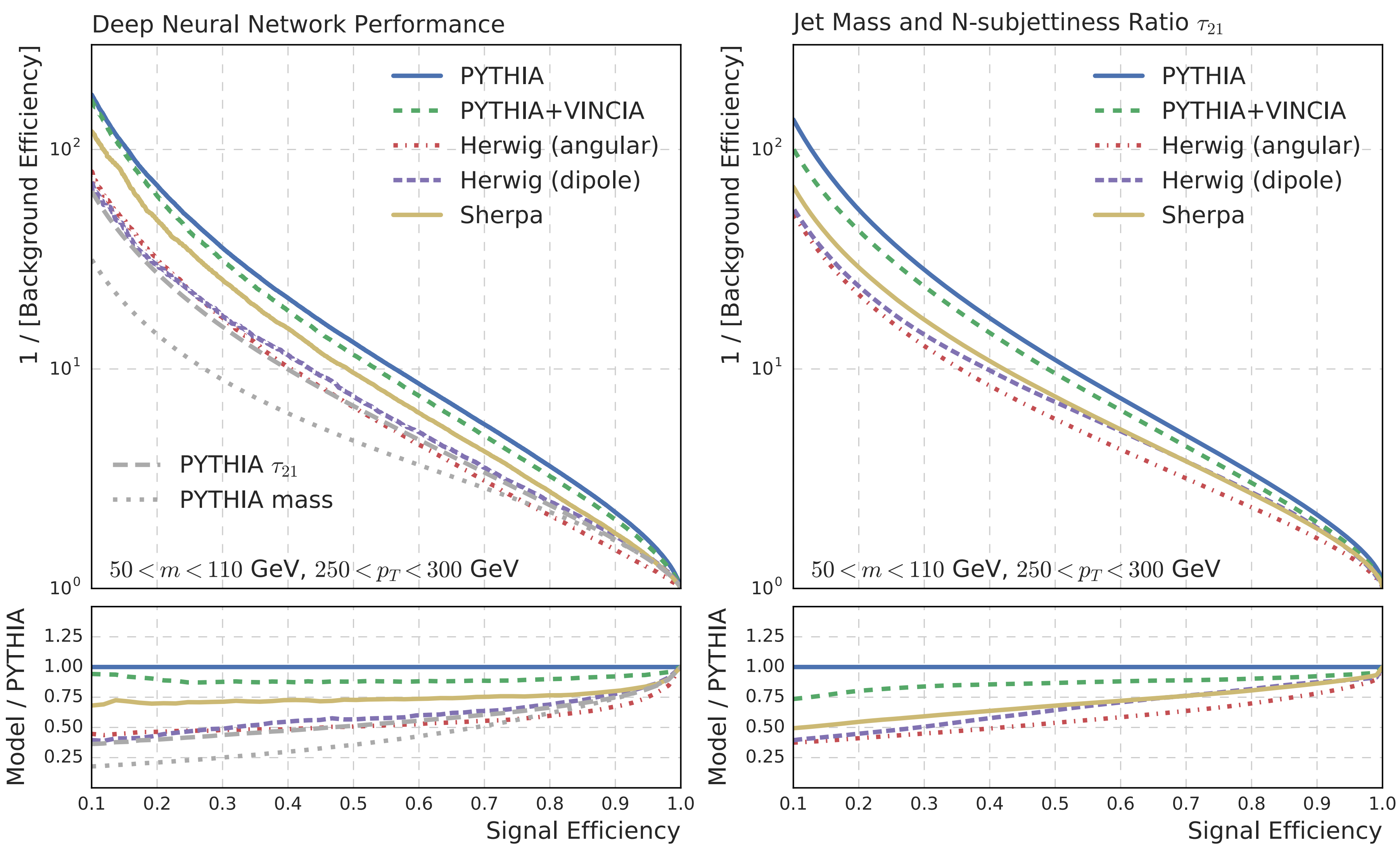
Scenario	Architecture	ROC AUC	$R_{\epsilon=50\%}$
nominal	k_t	0.9185 ± 0.0006	68.3 ± 1.8
nominal	desc-p_T	0.9189 ± 0.0009	70.4 ± 3.6
collinear1	k_t	0.9183 ± 0.0006	68.7 ± 2.0
collinear1	desc-p_T	0.9188 ± 0.0010	70.7 ± 4.0
collinear10	k_t	0.9174 ± 0.0006	67.5 ± 2.6
collinear10	desc-p_T	0.9178 ± 0.0011	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

SHOWER UNCERTAINTIES

G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046
Barnard, et al arXiv:1609.00607
Dery, et al arXiv:1702.00414

We should keep in mind that the there is uncertainty in the showers due to different generators. Two approaches:

- weakly supervised approach (see arXiv:1702.00414) uses real data, but requires signal examples in data with known proportion
- "learning to pivot" modify training to be robust to the "known unknowns" of the simulation

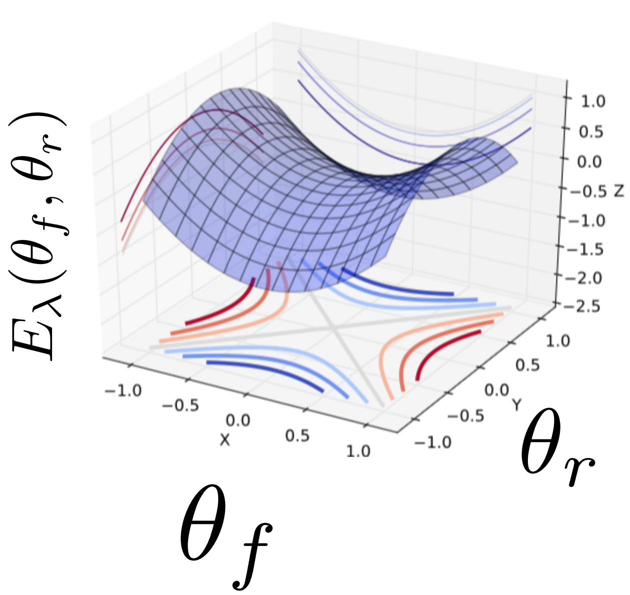


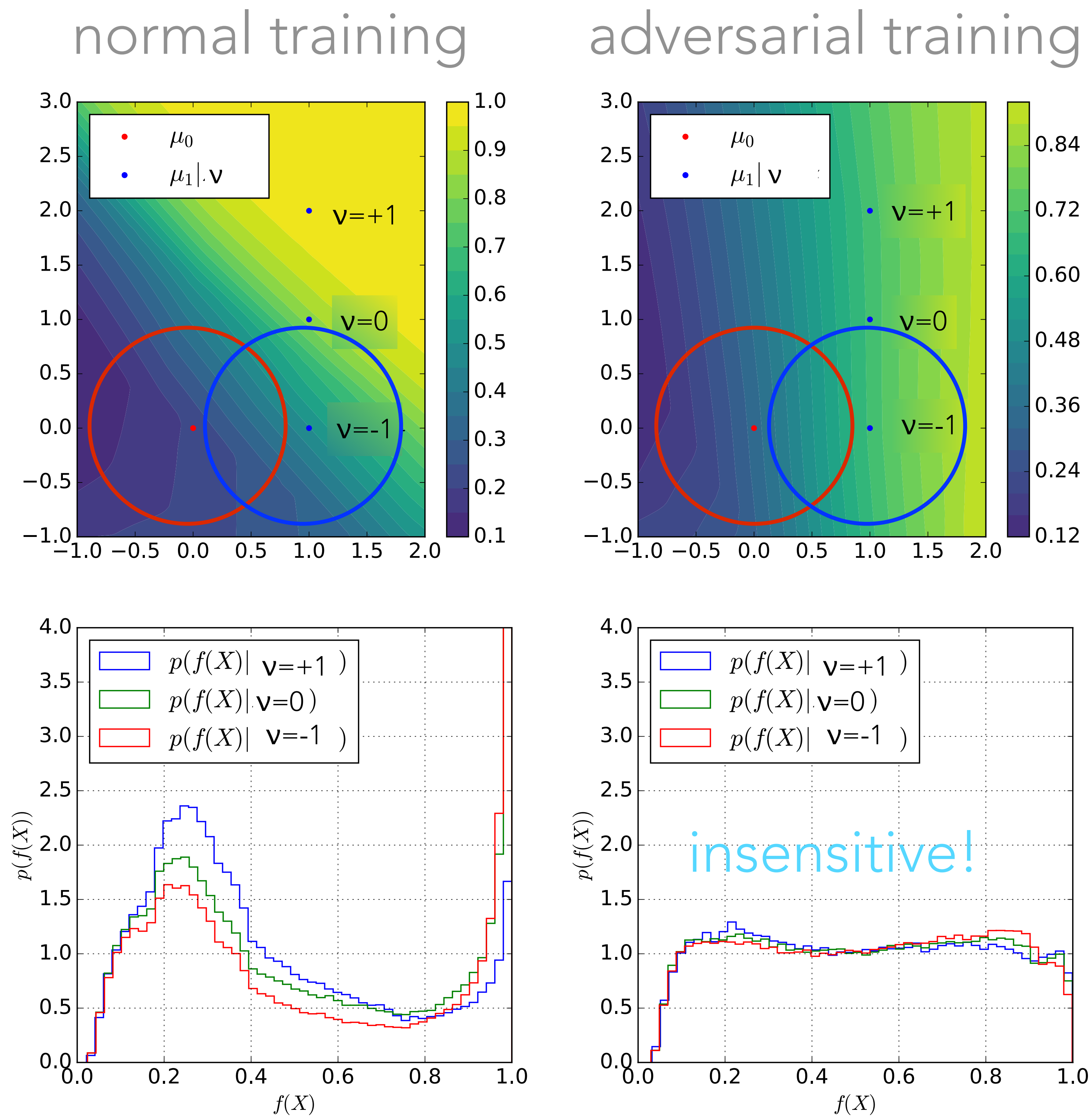
Typically classifier **f(x)** trained to minimize loss $\mathbf{L_f}$.

- want classifier output to be insensitive to systematics (nuisance parameter $\boldsymbol{\nu}$)
- introduce an **adversary r** that tries to predict ν based on f.
- setup as a minimax game:

$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} E(\theta_f, \theta_r).$$

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)$$





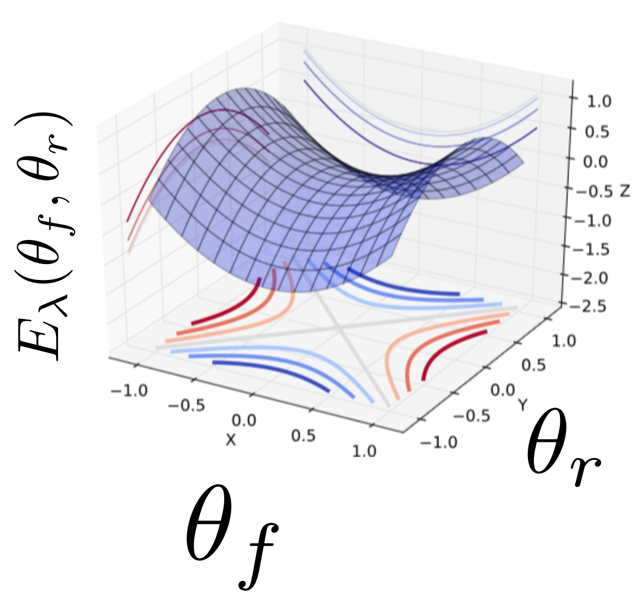
G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

Typically classifier **f(x)** trained to minimize loss **L_f**.

- want classifier output to be insensitive to systematics (nuisance parameter **ν**)
- introduce an **adversary r** that tries to predict ν based on f.
- setup as a minimax game:

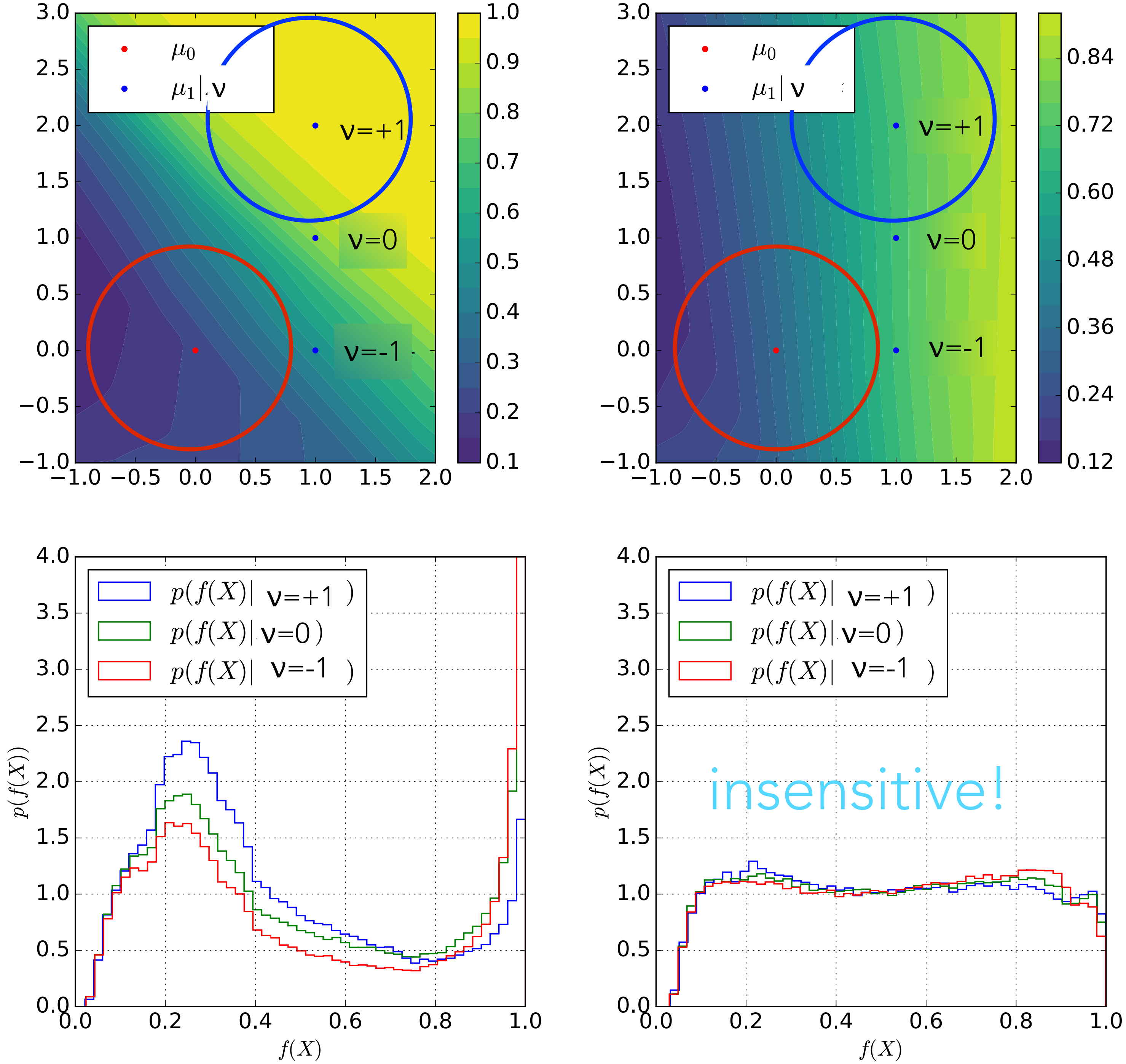
$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f, \theta_r).$$

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)$$



normal training

adversarial training



G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

Technique allows us to tune **λ**, the tradeoff between classification power and robustness to systematic uncertainty

An example:

background: 1000 QCD jets

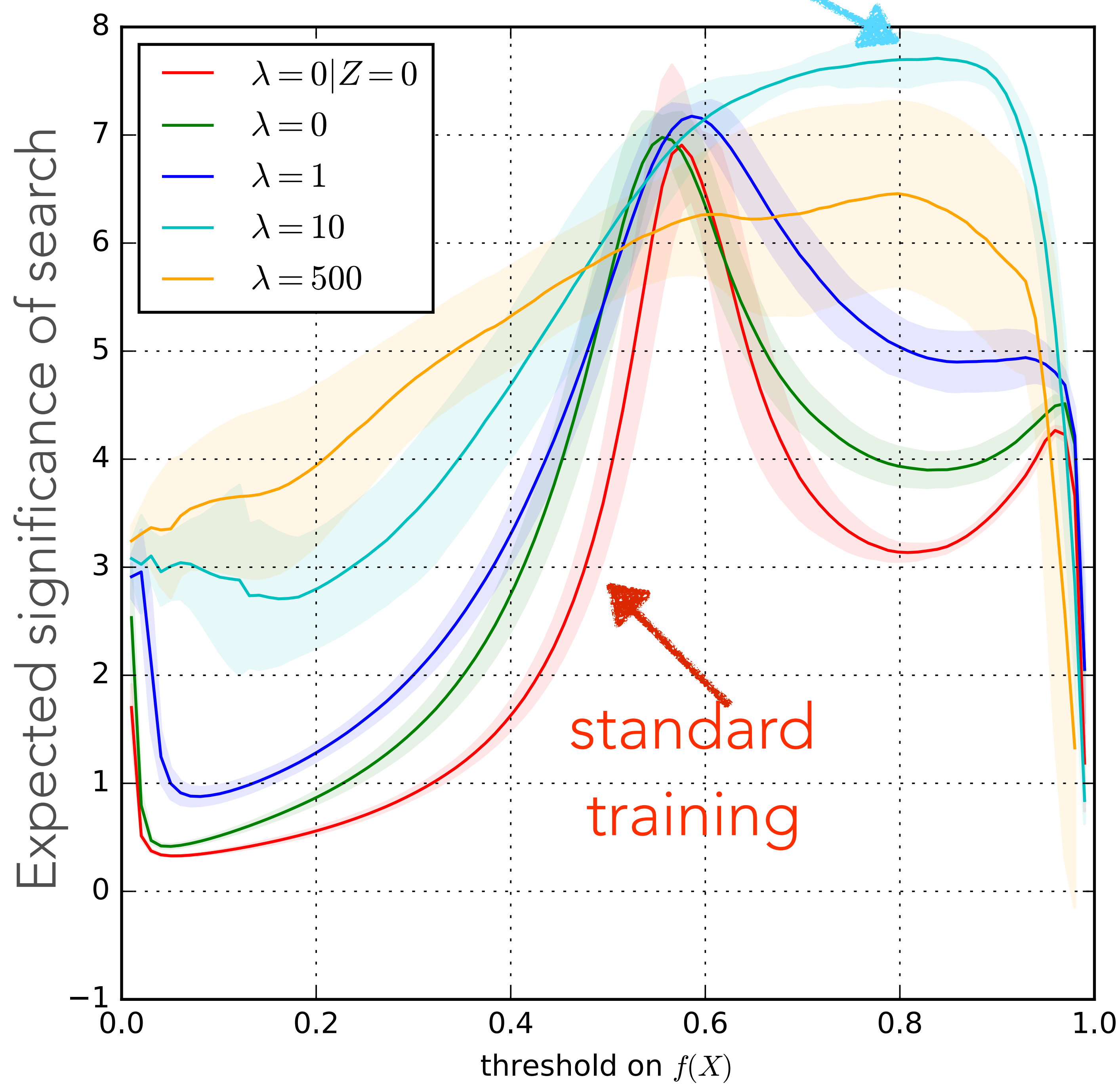
signal: 100 boosted W's

Train W vs. QCD classifier

Pileup as source of uncertainty

Simple cut-and-count analysis with background uncertainty.

optimal tradeoff of classification vs. & robustness



APPLICATION OF "LEARNING TO PIVOT"

Decorrelated Jet Substructure Tagging using Adversarial Neural Networks

Chase Shimmin
Department of Physics and Astronomy, UC Irvine, Irvine, CA 92627 and
Department of Physics, Yale University, New Haven, CT

Peter Sadowski and Pierre Baldi
Department of Computer Science, UC Irvine, Irvine, CA 92627

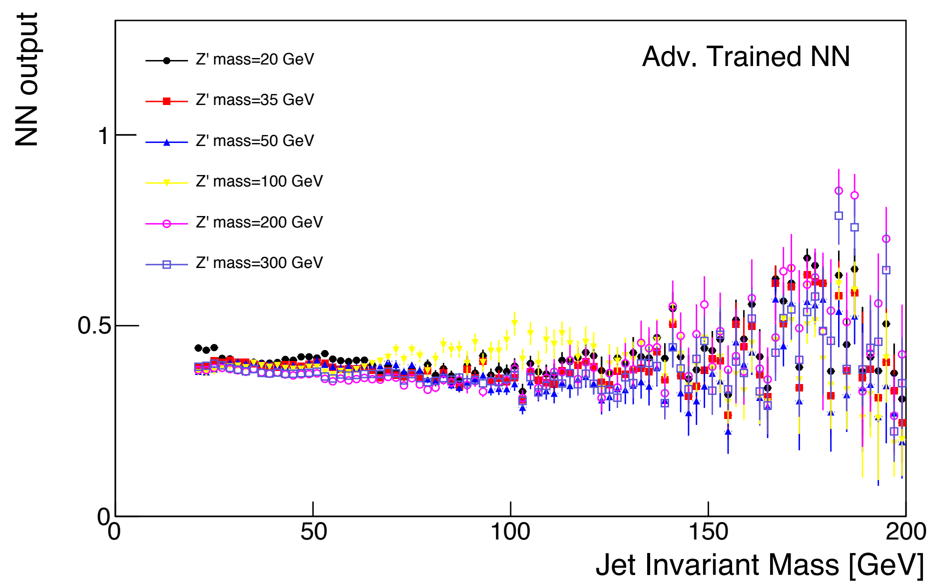
Edison Weik and Daniel Whiteson
Department of Physics and Astronomy, UC Irvine, Irvine, CA 92627

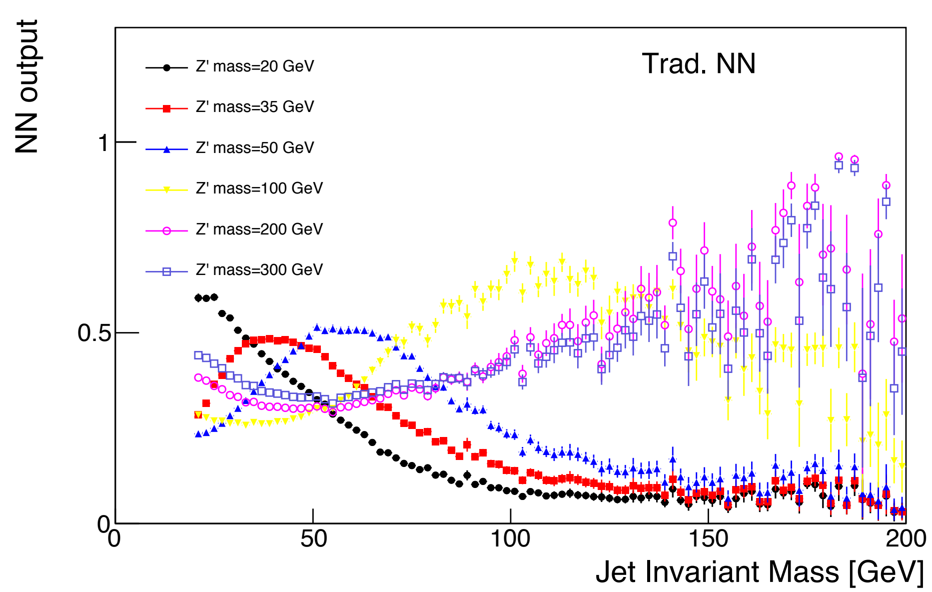
Edward Goul
Department of Physics, MIT, Cambridge, MA 02139

Andreas Søgaard
Department of Physics and Astronomy, University of Edinburgh, Edinburgh UK

(Dated: March 13, 2017)

Our adversarial technique has been applied to find jet tagger that is decorrelated with jet mass (which would be used as a discriminating variable in a fit)





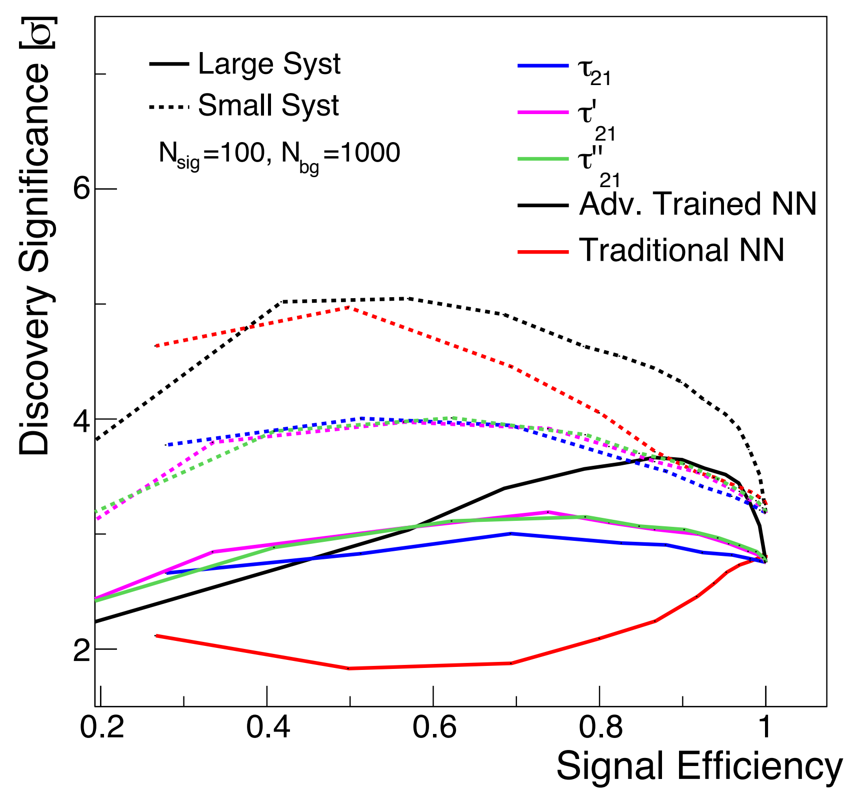
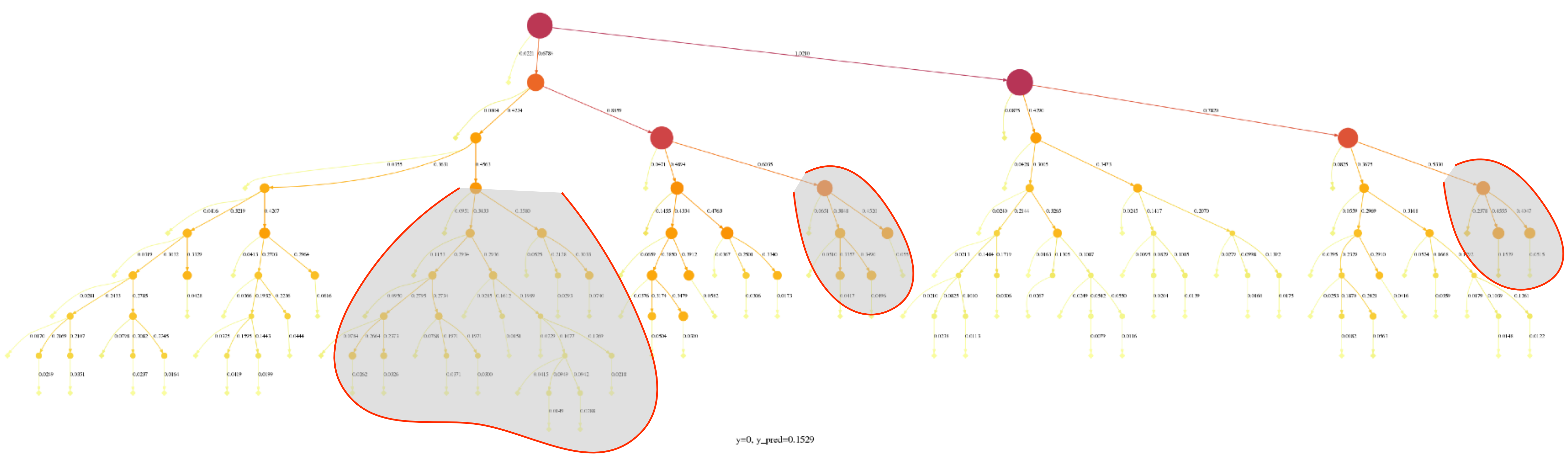


FIG. 9. Statistical significance of a hypothetical signal for varying thresholds on the outputs of networks trained to optimize classification compared to adversarial networks trained to optimize classification while minimizing impact on jet mass. Shown are two scenarios, in which the uncertainty on the background level is negligible or large, both with $N_{\rm sig} = 100, N_{\rm bg} = 1000$.

Work in progress

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 Gilles Louppe

GRAPH CONVOLUTIONAL NEURAL NETWORKS

So far the compositional structure we are iterating over is fixed by the jet algorithm.

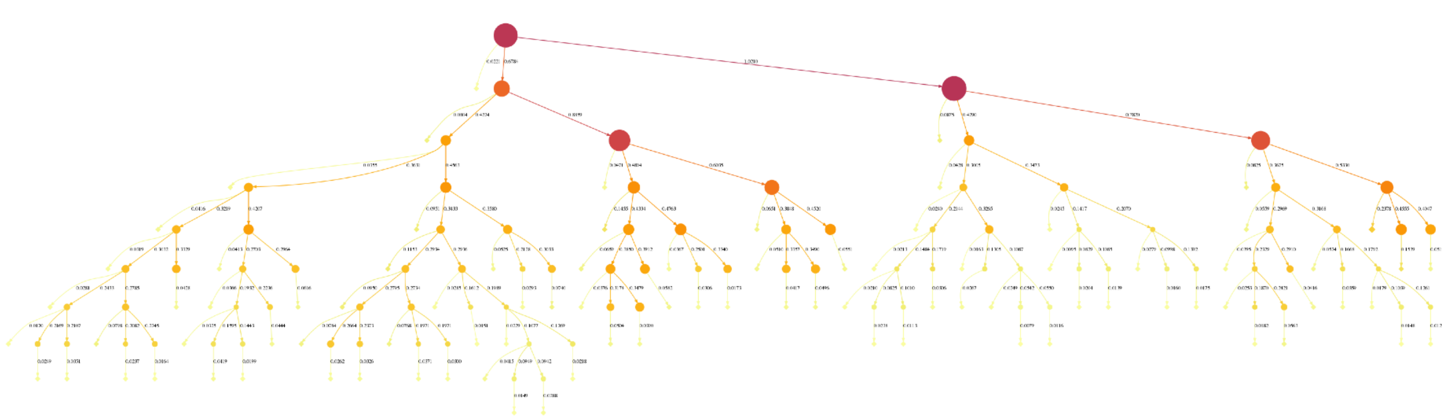
- Hyperparameter α interpolating kt → anti-kt

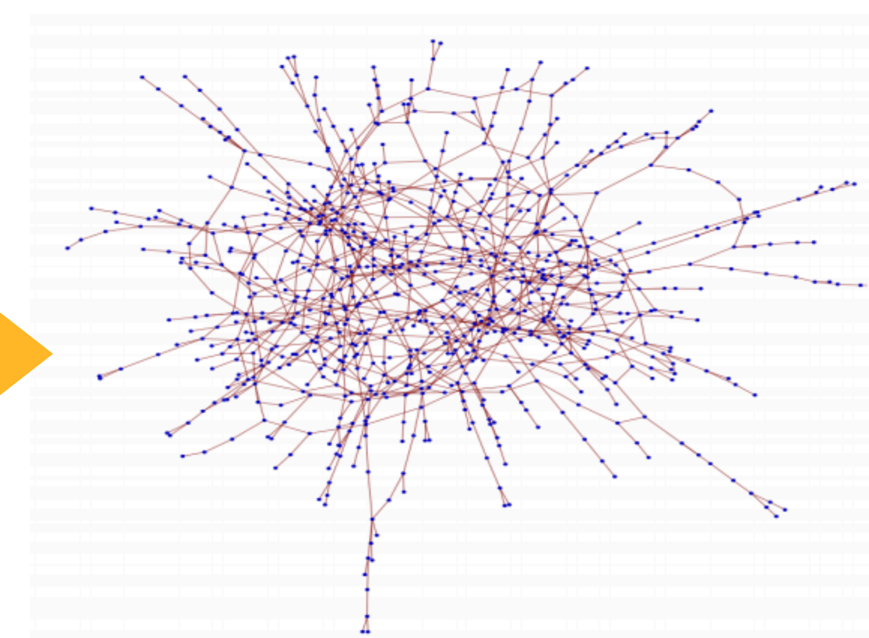
$$d_{ii'}^{\alpha} = \min(p_{ti}^{2\alpha}, p_{ti'}^{2\alpha}) \frac{\Delta R_{ii'}^2}{R^2}$$

Would like to optimize α, but that leads to discontinuous change in jet clustering history.

Instead, consider a graph over particles with adjacency matrix given by $d^{\alpha}_{ii'}$

- Defines a graph convolutional neural network, we can propagate gradients wrt α!
- potentially promote constant α to a non-linear function of hidden state $\alpha(h_t)$





Spectral Networks and Deep Locally Connected Networks on Graphs

Joan Bruna
New York University
bruna@cims.nyu.edu

Wojciech Zaremba
New York University
woj.zaremba@gmail.com

Arthur Szlam
The City College of New York
aszlam@ccny.cuny.edu

Yann LeCun
New York University
yann@cs.nyu.edu

*Work in progress with Gilles Louppe, Joan Bruna, Gaspar Rochette

CONCLUSIONS

Jet physics is a very active area of machine learning research

- previously it has been dominated by an image-based analogy (using fixed input representation that requires pre-processing)
- we operate on a variable length set of 4-momenta and use a QCD-inspired network topology. The network topology matters.
- QCD-inspired appears more IRC-robust.
  - To do: “learn to pivot” → “learn to groom"
- requires much less data to train (we used ~100x less data)
- we can extend ↑ to “event embedding” & use all the particles in an event as input! Intermediate jets representation helps. Also extend ↓ to “particle embedding”
- Code: https://github.com/glouppe/recnn (would like to translate to PyTorch)

Many more ideas for hybrids of QCD & machine learning!