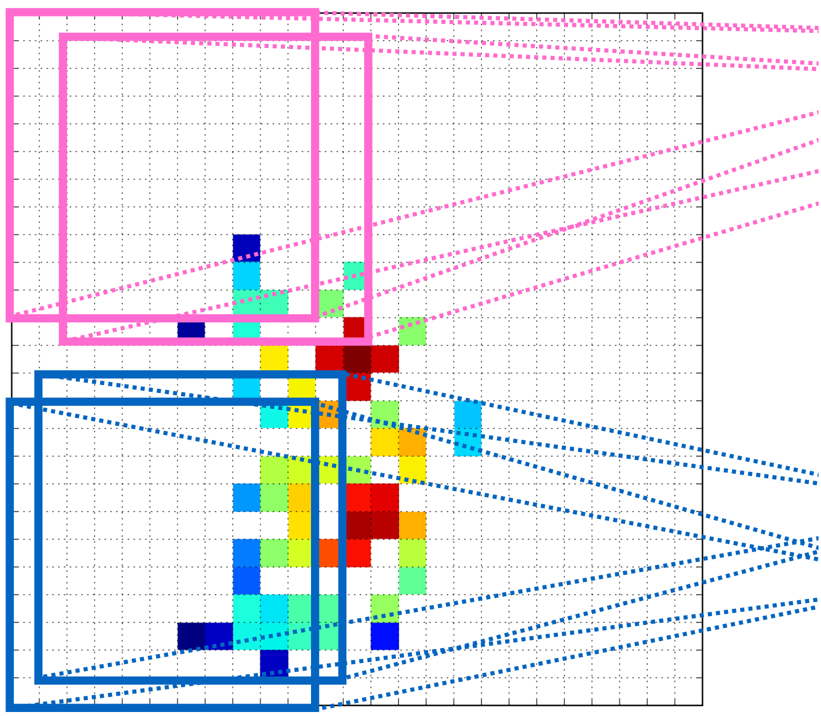
Machine Learning for Jets
2019 - Fermilab edition

Benjamin Nachman

Lawrence Berkeley National Laboratory



BERKELEY LAB

BERKELEY EXPERIMENTAL PARTICLE PHYSICS

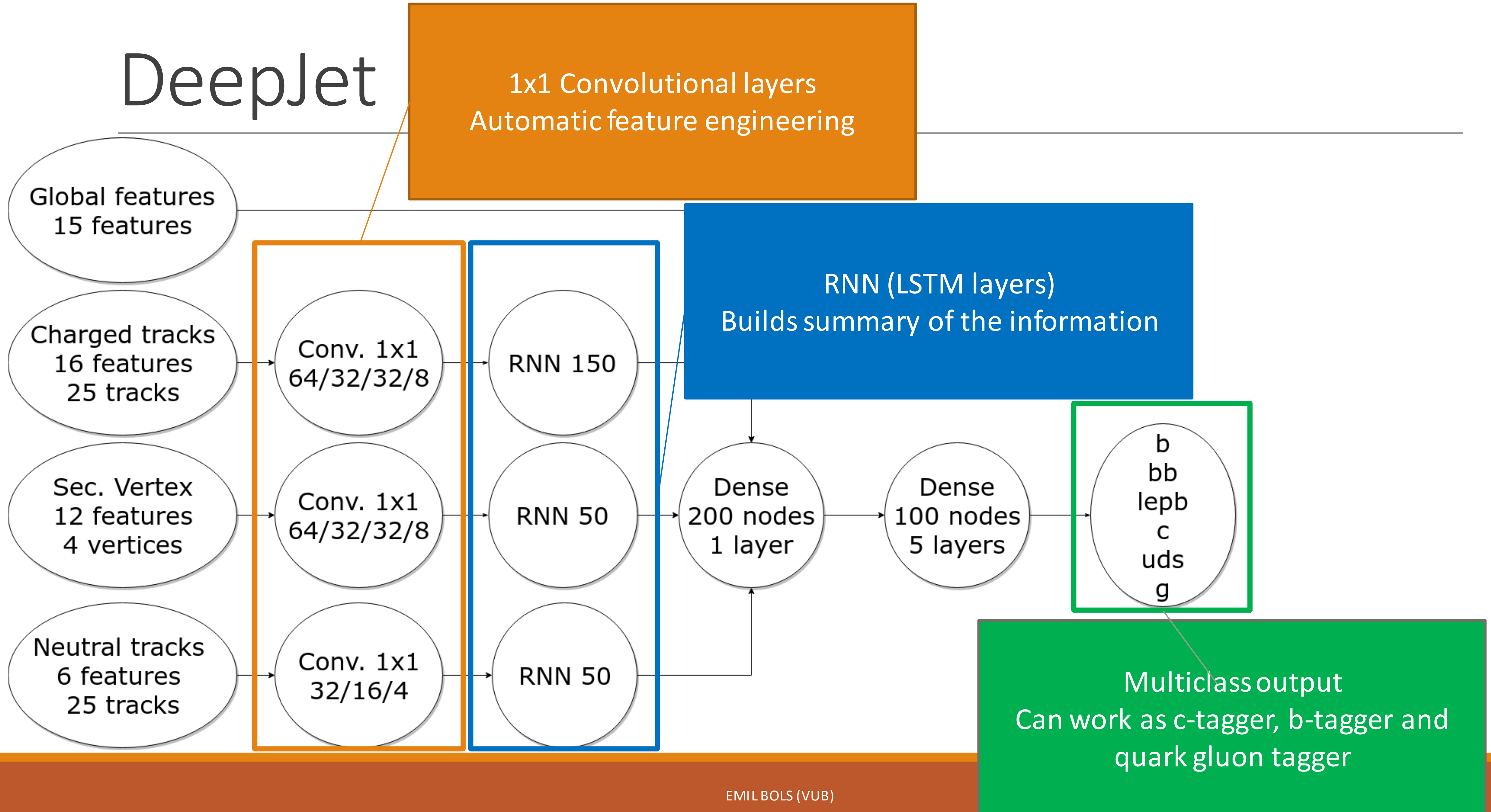
January 23, 2019

What are jets?

ML4Jets

Flavor tagging in CMS

DeepJet

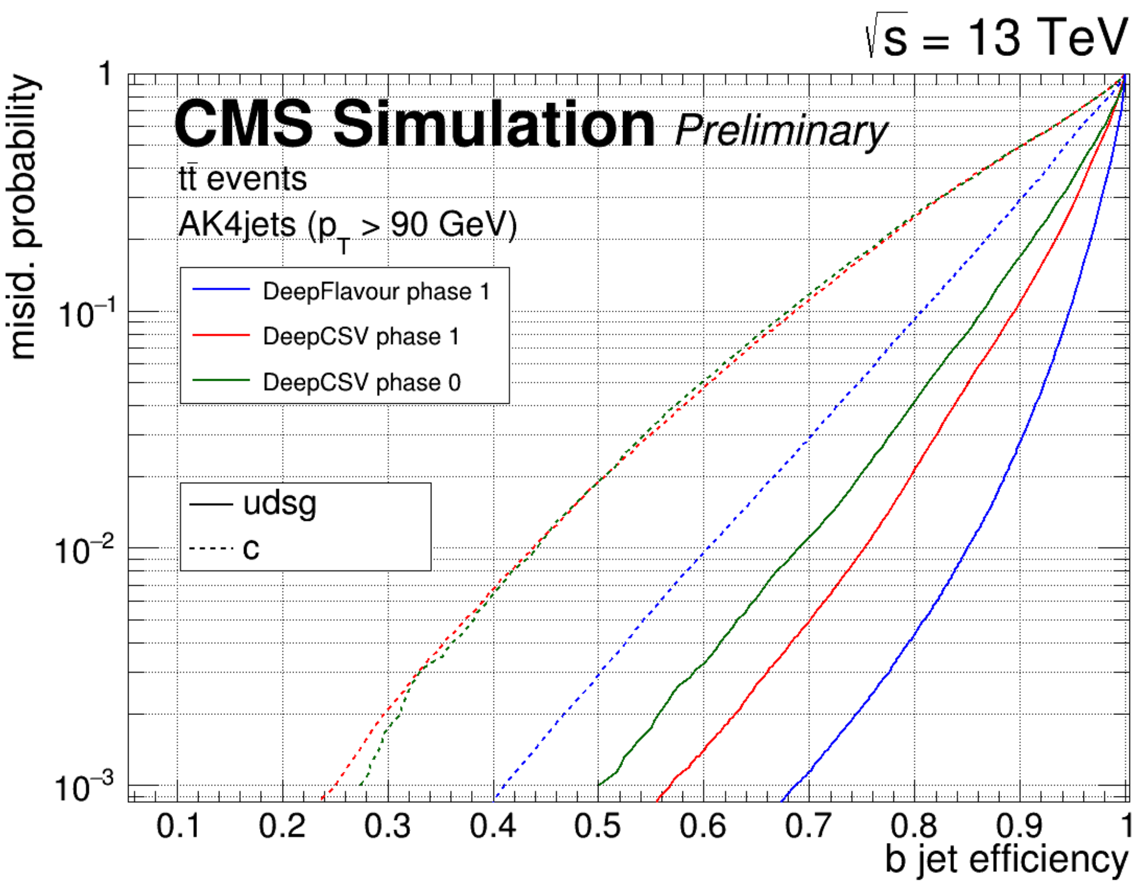


Flavor tagging in CMS

DeepJet

CMS DP-2018/033

- Large improvement in performance in simulation
- However simulation is never perfect
- Will this gain translate into data?

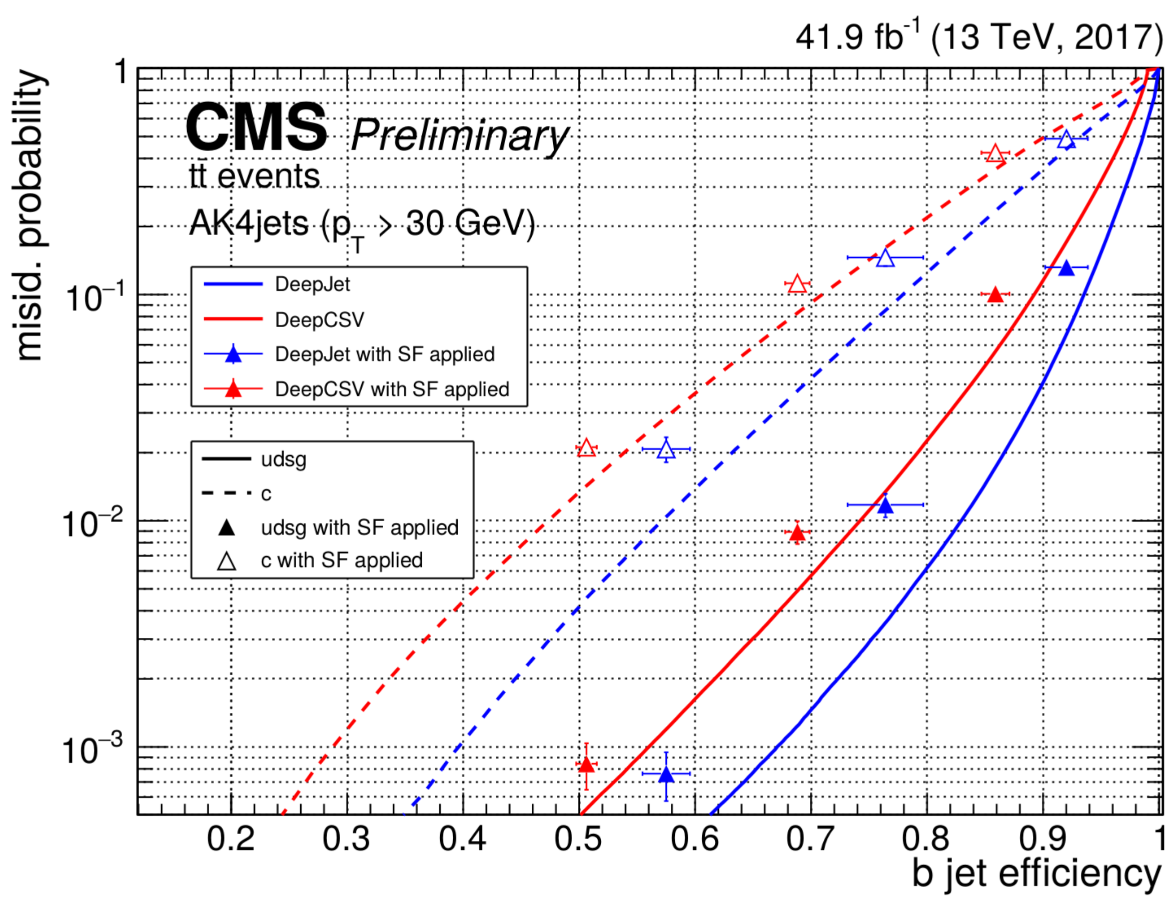


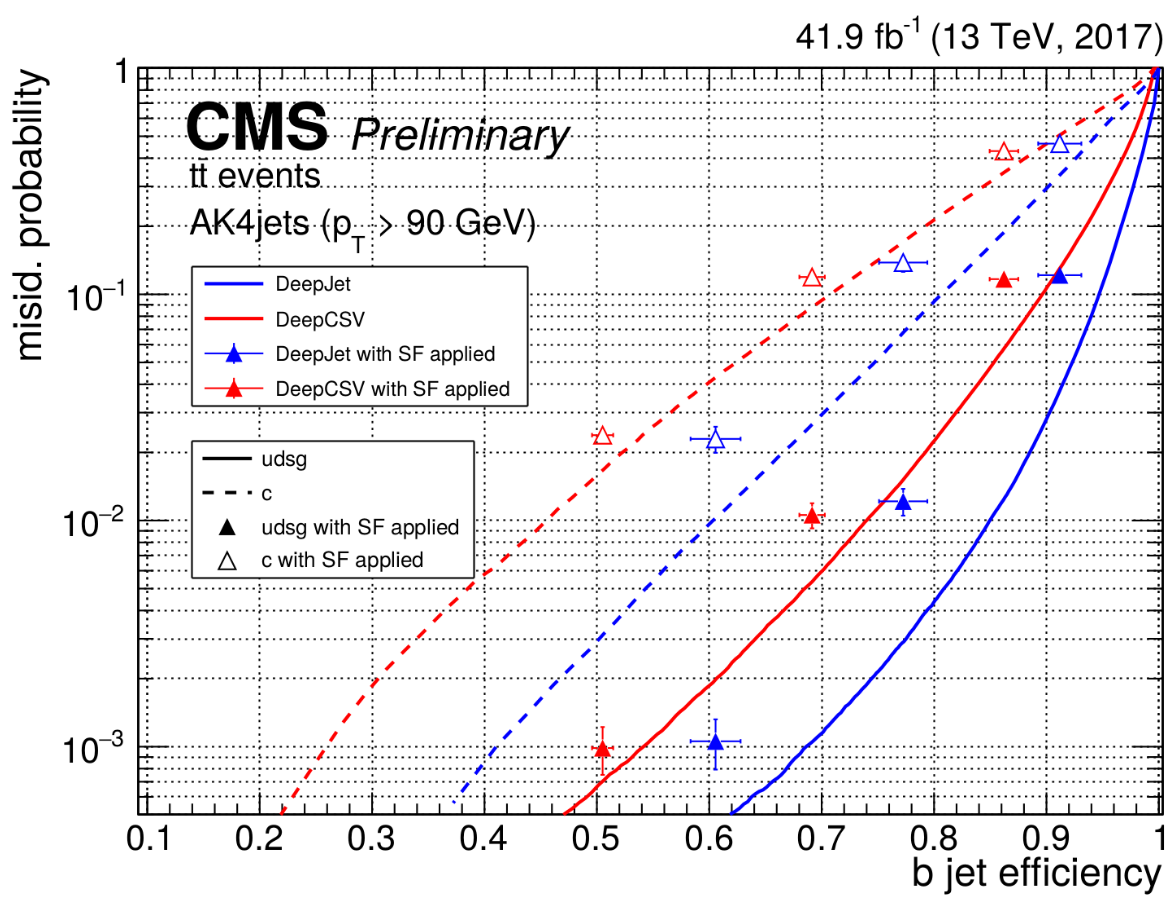
Flavor tagging in CMS

6

DeepJet

CMS DP-2018/058





Deep Sets for Particle Jets

[PTK, Metodiev, Thaler, 1810.05165]

Particle Flow Network (PFN)

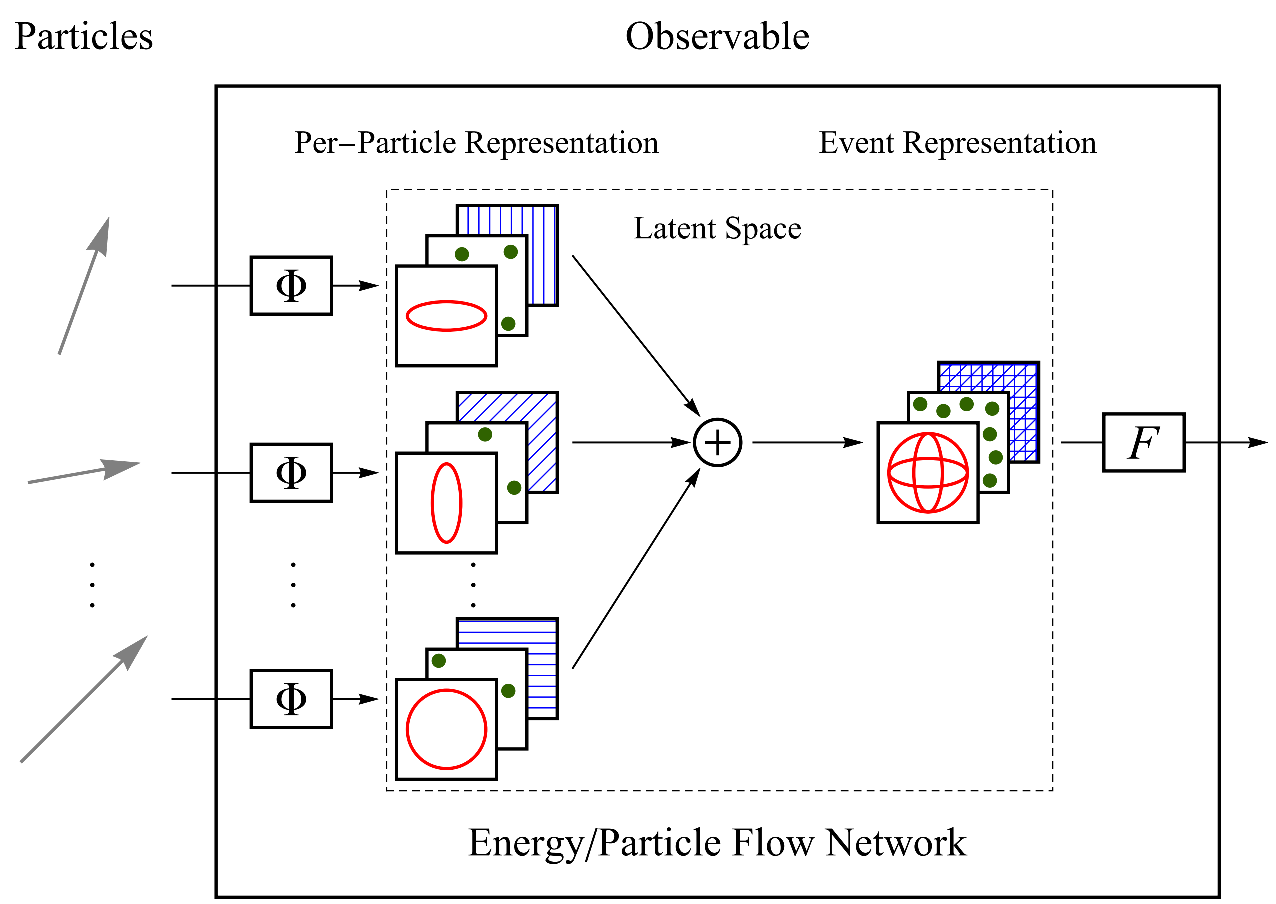
$$\mathrm{PFN}(\{p_1^\mu,\ldots,p_M^\mu\}) = F\left(\sum_{i=1}^{M}\Phi(p_i^\mu)\right)$$

Fully general latent space

Energy Flow Network (EFN)

$$\mathrm{EFN}(\{p_1^\mu,\ldots,p_M^\mu\}) = F\left(\sum_{i=1}^{M} z_i\Phi(\hat{p}_i)\right)$$

IRC-safe latent space



Deep Sets

Classification Performance

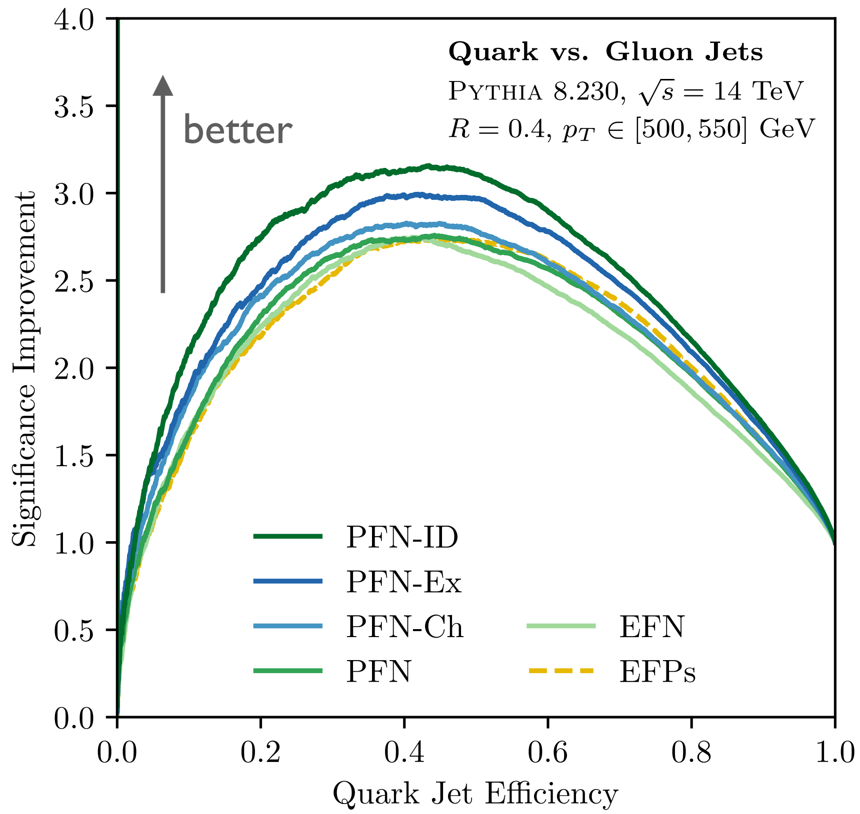
PFN-ID: Full particle flavor info
$(\gamma, \pi^\pm, K^\pm, K_L, p, \bar{p}, n, \bar{n}, e^\pm, \mu^\pm)$

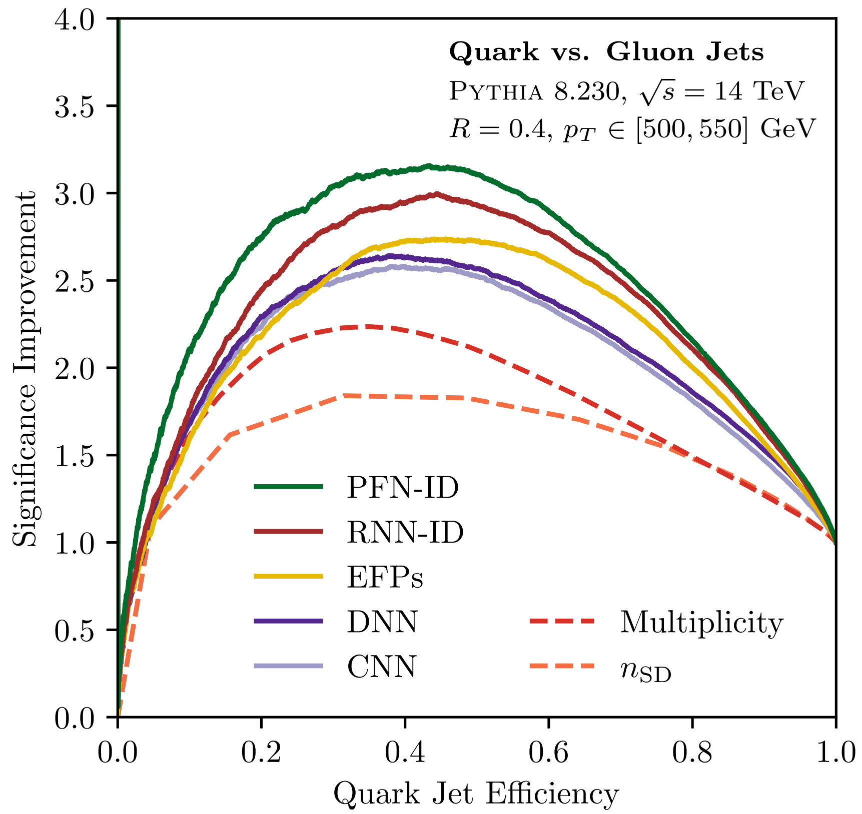
PFN-Ex: Experimentally accessible info
$(\gamma, h^{\pm,0}, e^\pm, \mu^\pm)$

PFN-Ch: Particle charge info
(+,0,−)

PFN: No particle type info, arbitrary energy dependence

EFN: IRC-safe latent space





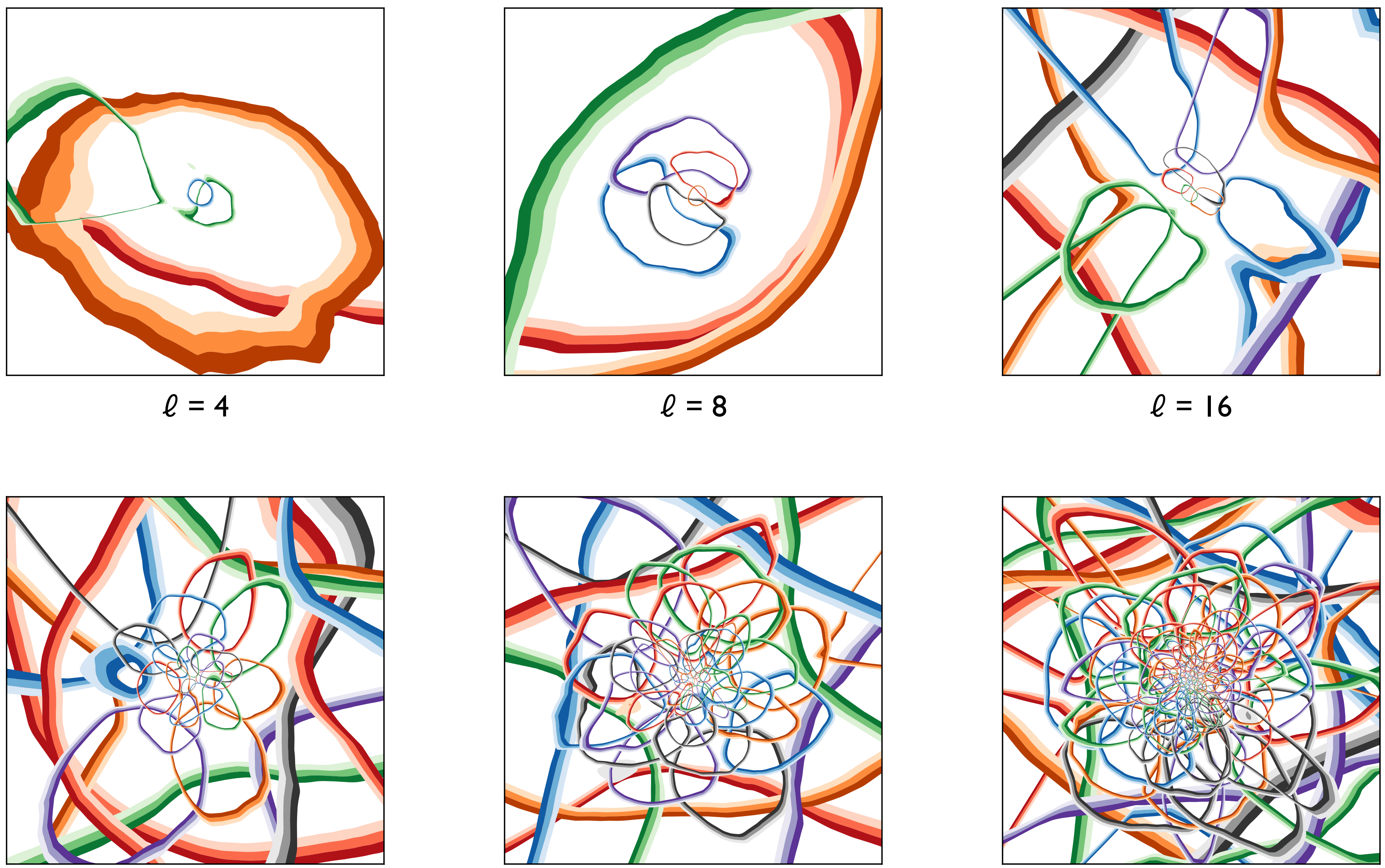
Latent space dimension $\ell = 256$

PFN-ID slightly better than RNN-ID

EFPs are comparable to EFN

Deep Sets

Visualizing Q/G EFN Filters



$\ell = 4$ $\ell = 8$ $\ell = 16$

$\ell = 32$ $\ell = 64$ $\ell = 128$

Deep Sets

Measuring Q/G EFN Filters

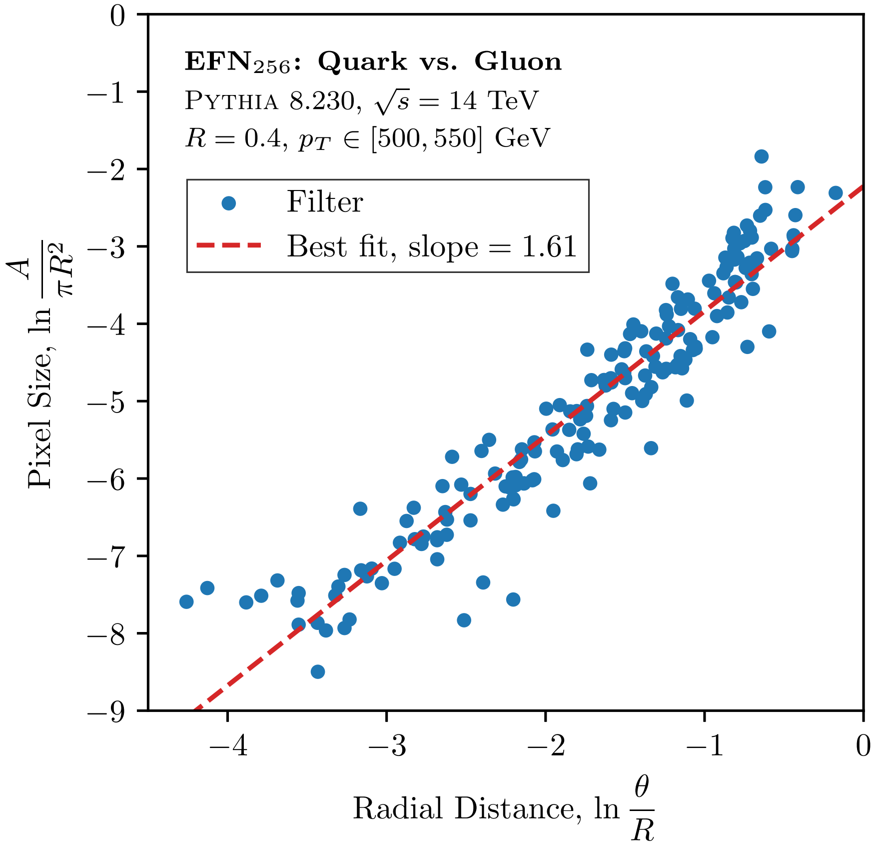
Power-law dependence between filter size and distance from center is observed

Slope of 2 is predicted at leading log

$$\left\|\boxed{\mathrm{d}\ln\frac{\theta}{R}\,\mathrm{d}\varphi}\right\| = \theta^2\,\boxed{\mathrm{d}y\,\mathrm{d}\phi}$$

Area element in rap-phi plane

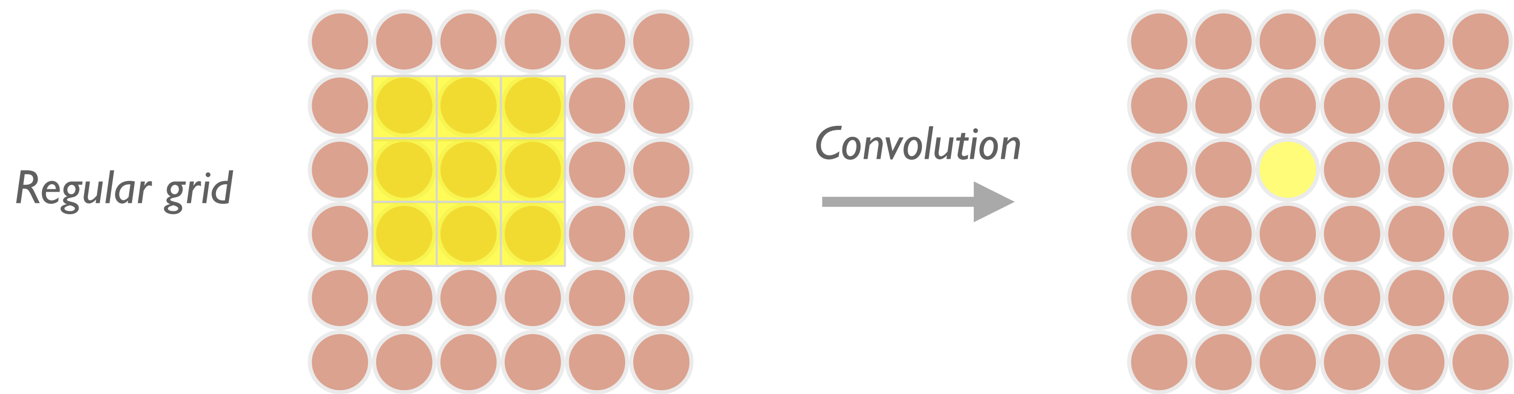
Emission plane area element



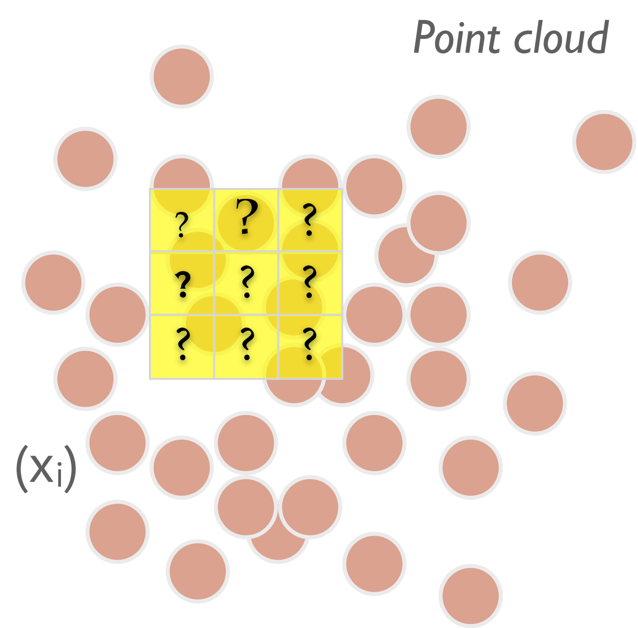
Non-perturbative physics, axis recoil, higher order effects cause deviations from slope of 2

Another particle cloud approach

CONVOLUTION ON REGULAR GRIDS



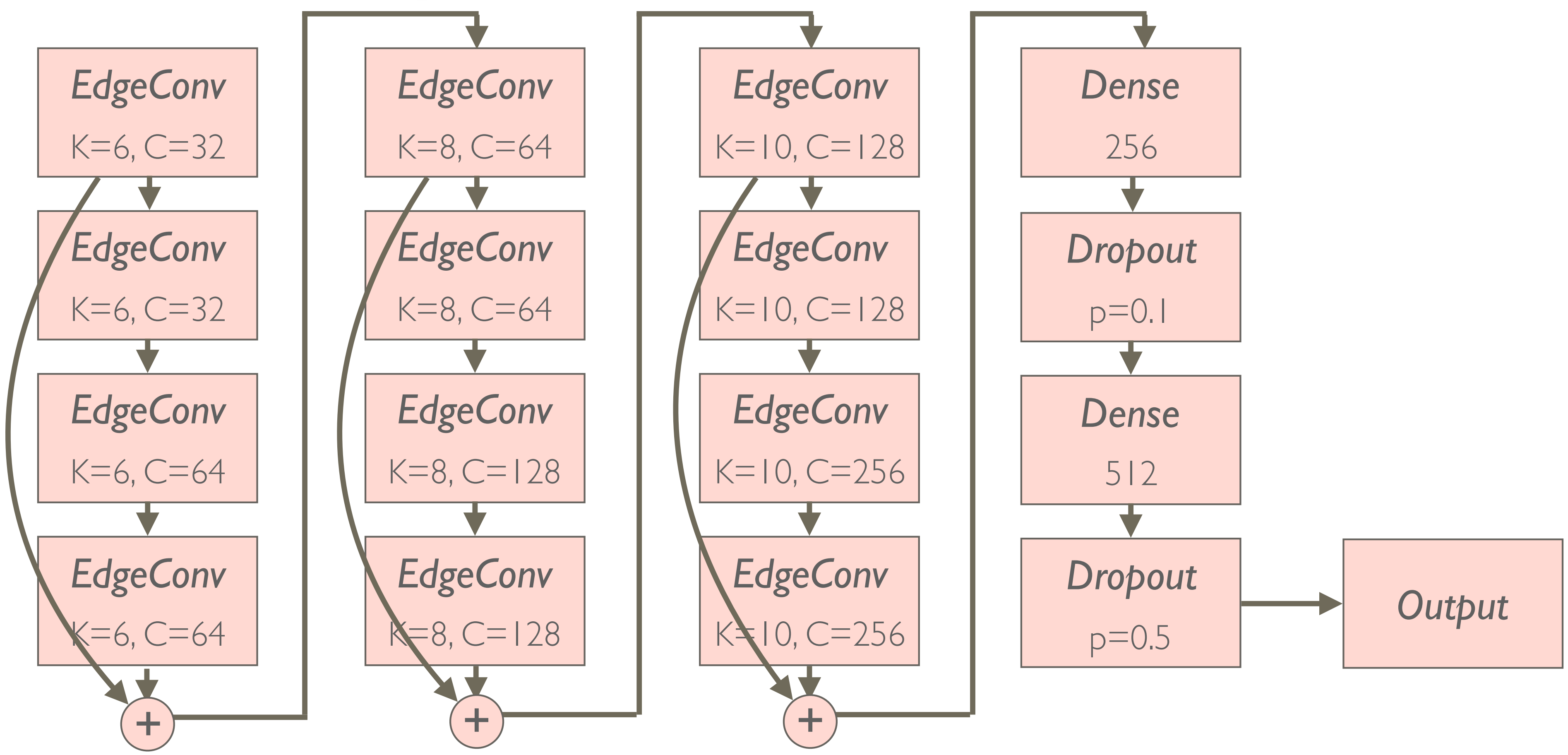
- Conventional convolution only operates on regular grids and cannot be applied on point clouds
  - point clouds are irregular
    - how to define a "local" patch to convolve?
  - point clouds are unordered
    - conventional convolution operation ($\Sigma_i K_i x_i$) is not invariant under permutation of the points (x_i)



Another particle cloud approach

NETWORK ARCHITECTURE

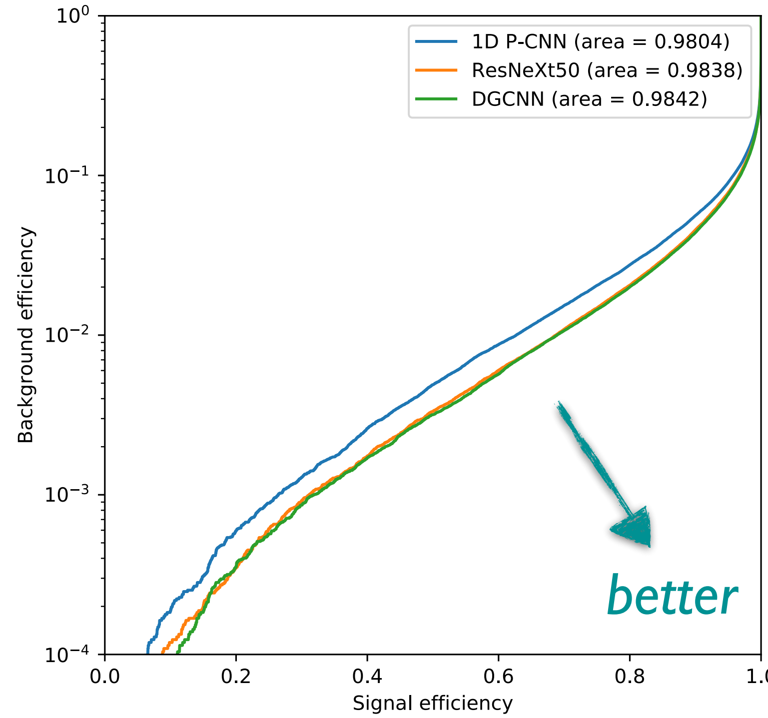
- Implemented with mxnet



- 3 stages: k-nearest neighbors updated at the beginning of each stage
- batch normalization (BN) used after each EdgeConv operation
- residual connection (RC) [1512.03385, 1603.05027] added between EdgeConv layers
- BN and RC helped greatly for stabilizing the training and also improving the performance

PERFORMANCE: TOP TAGGING

- Top tagging:
  - only particle 4-momentum is available
  - train/val/test 1.2M/400k/400k
  - results of more algorithms available at link
- We managed to push the boundary a bit further
  - >20% lower background at signal efficiency of 30%
  - *Is the gain real? Or is it just learning more details of the parton shower model?*
  - but personally I would not really consider the jet tagging problem as "solved"
    - especially facing realistic experimental challenges like pileup, detector effects, and additional information (e.g., tracking, timing, etc.)

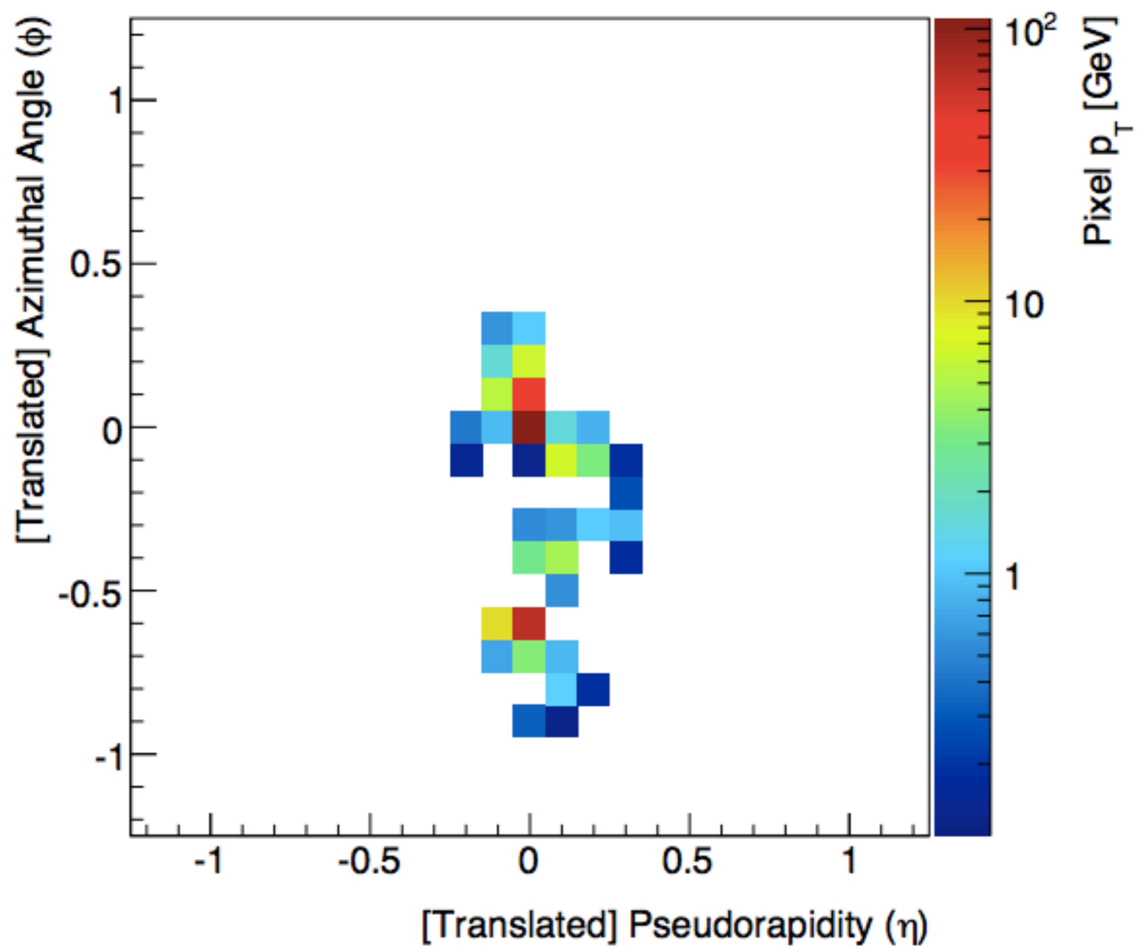


Performance on top-tagging dataset

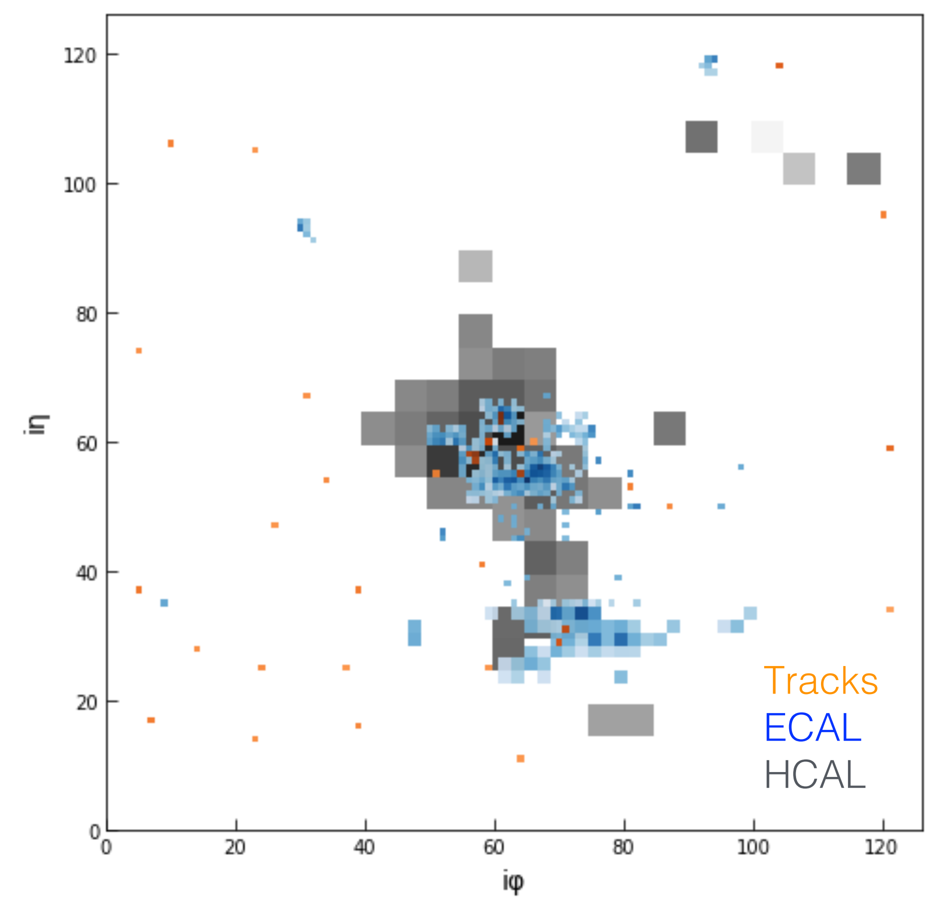
Algorithm	Accuracy	ROC AUC	$1/\varepsilon_{bkg}$ @ ε_{sig}=30%
1D P-CNN	0.930	0.9804	780
2D CNN [ResNeXt50]	0.936	0.9838	1086
DGCNN	**0.937**	**0.9842**	**1160**
PFN-r.r. [arXiv:1810.05165]	*0.932*	*0.9819 ± 0.0001*	*888±17*

Jet ID | Traditional vs E2E Image

Traditional
jet image



E2E
jet image



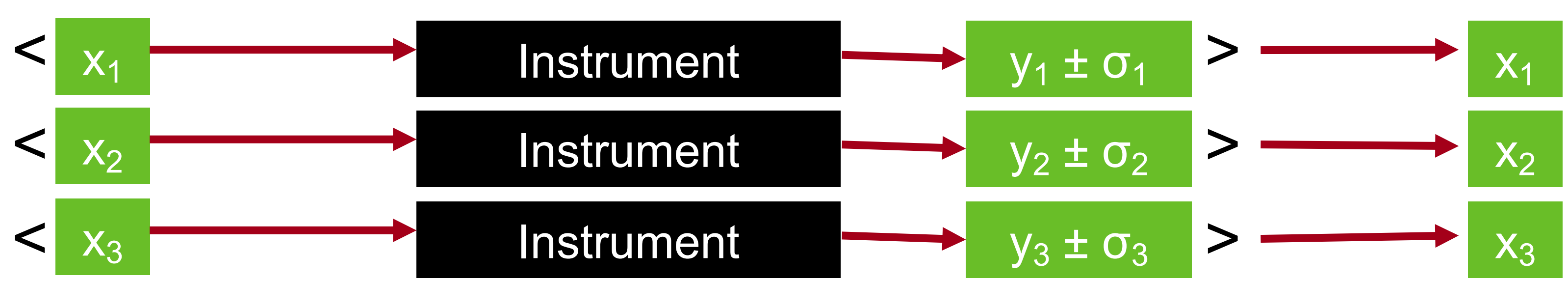
Note: Not the same jet.

Prior-independent regression

Introduction: Numerical Inversion

SLAC

- Calibration with *numerical inversion*:
  - Train on jets with known energies x, take output value with maximum likelihood: $f(x) = \langle p_T^{reco} | p_T^{true} = x \rangle$
  - Invert function for unknown observed values: $p_T^{reco} \rightarrow f^{-1}(p_T^{reco})$



  - *Independent* of the distribution of true energies used in training
    - Processes in LHC have different true energy distributions!
    - Guarantees closure* when conditioning on $p_T^{true}=x$
  - *This is* what we do in ATLAS right now for MCJES and GSC
  - *Formal details: [arXiv:1609.05195]

Aviv Cukierman

Prior-independent regression

Generalized Numerical Inversion

SLAC

- Generalized numerical inversion:
  - $L(x;\theta) = \langle p_T^{reco} | p_T^{true} = x;\theta \rangle$
  - Calibration: $p_T^{reco} \rightarrow C(p_T^{reco};\theta) = L^{-1}(p_T^{reco};\theta)$

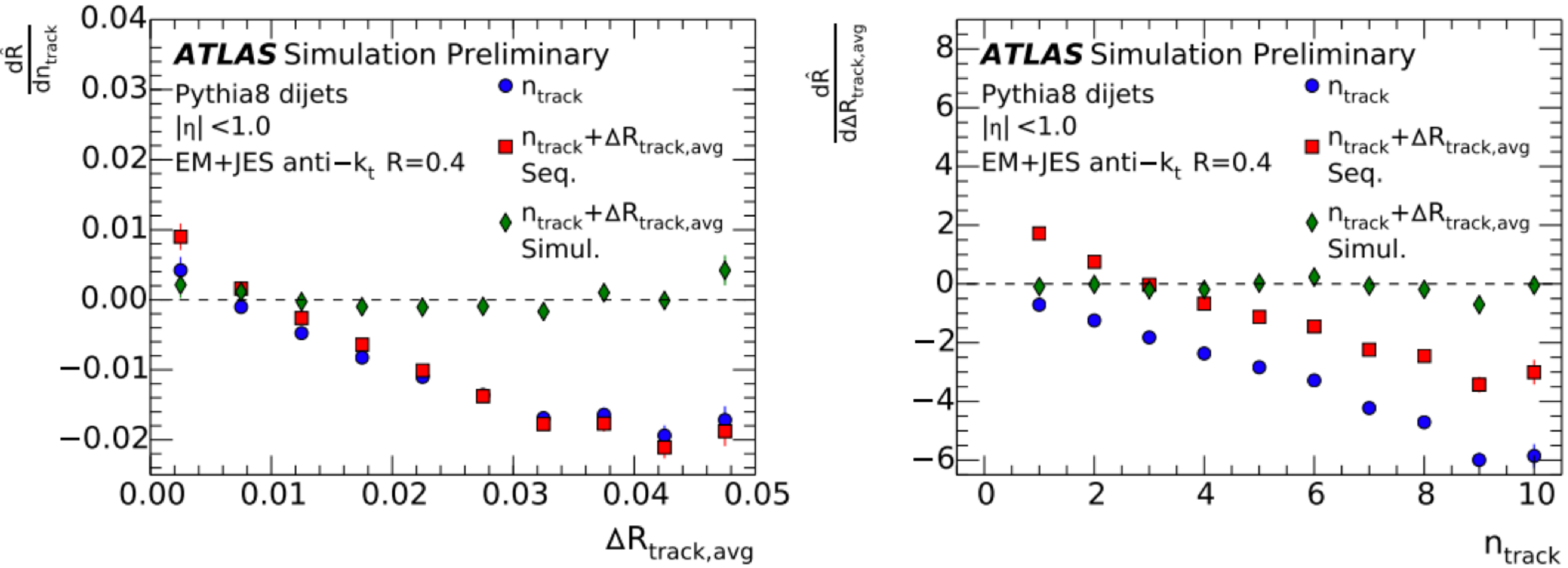
1. Learn a neural network approximation to the function $L(x,\theta) = \langle p_T^{\text{reco}} | p_T^{\text{true}} = x, \theta \rangle$. Note that $L(x,\theta) : \mathbb{R}^{n+1} \to \mathbb{R}$.
2. Learn a neural network $C(L(x,\theta),\theta)$ that tries to predict x given θ and $L(x,\theta)$. This is an approximation to the family of functions $f_\theta^{-1}(x)$. Note that learning the inverse this way is technically simple since L is single-valued.
3. Calibrate with $p_T^{\text{reco}} \mapsto C(p_T^{\text{reco}},\theta)$. The calibration non-closure is given the deviation of $C(p_T^{\text{reco}},\theta|x)$ from x.

  - Advantages:
    - Still independent of underlying p_T distribution
    - Easy to add in other features, unbinned
    - Can take into account correlations between features

8

Generalized Numerical Inversion – Final Results

- Residual dependence
  - Simultaneous is flat at 0 – demonstration of method

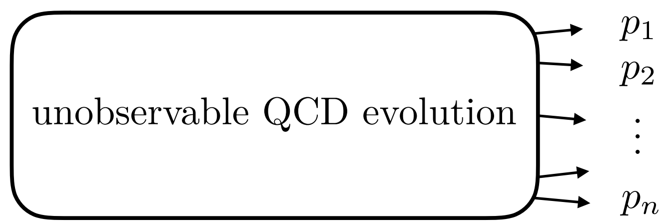


(a) (b)

Physics-inspired generative models

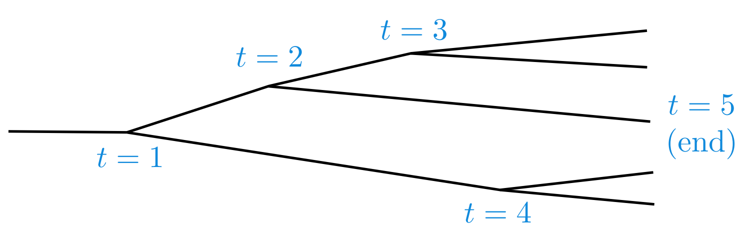
Summary So Far

JUNIPR computes the probability of a jet…



$$\xRightarrow[\text{JUNIPR}]{} \quad P_{\text{jet}}(\{p_1 \ldots p_n\})$$

…as a product over time steps in its clustering tree…



$$\xRightarrow[\text{JUNIPR}]{} \quad P_{\text{jet}}(\{p_1 \ldots p_n\}) = P_{t=1} \cdots P_{t=n}$$

…where each time step is decomposed into 3 parts:

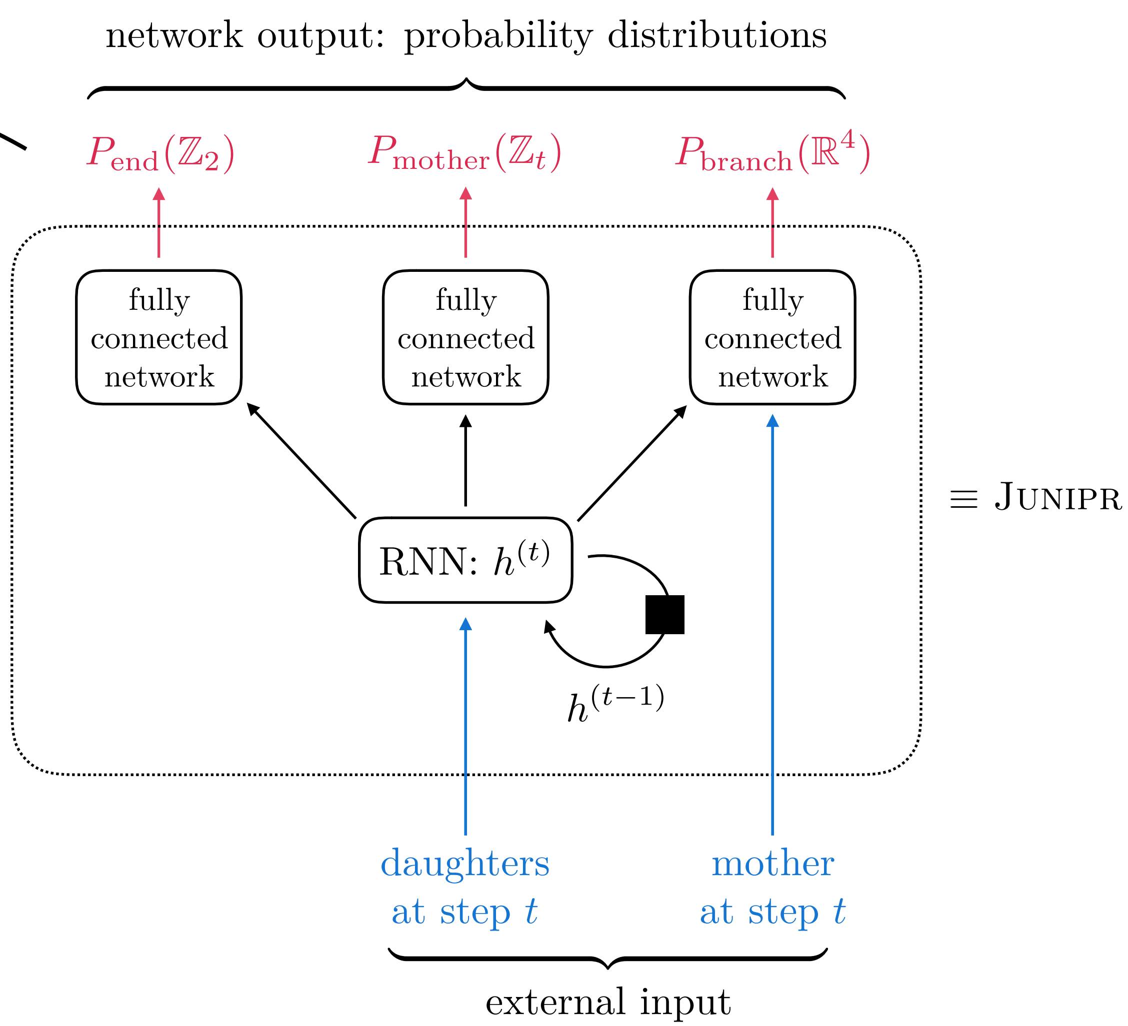
$$\boxed{P_t = P_{\text{end}} \cdot P_{\text{mother}} \cdot P_{\text{branch}}}$$

Anders Andreassen

Physics-inspired generative models

Implementation with RNN

- STEP 2) feed $h^{(t)}$ into neural networks computing probability distributions

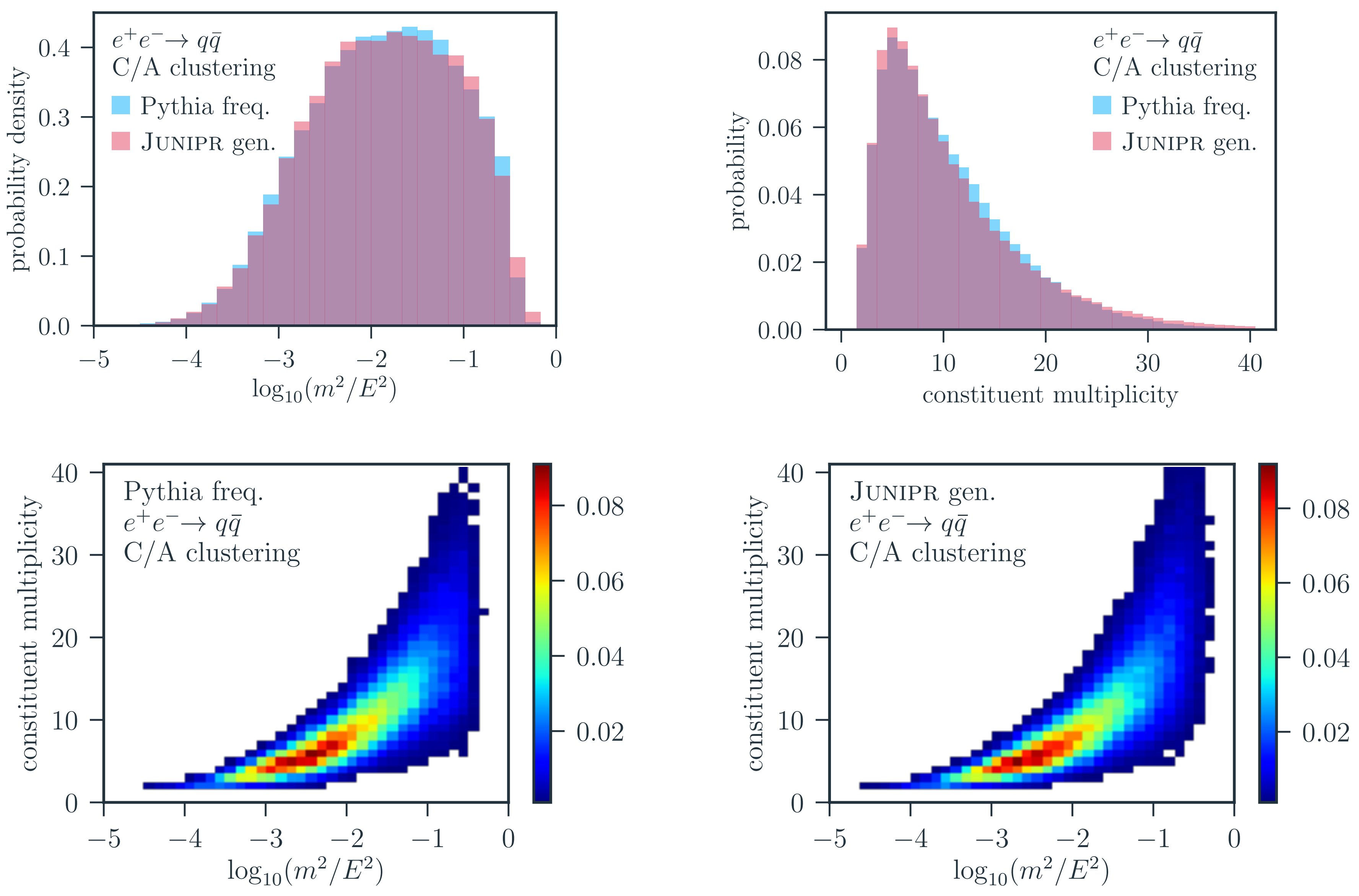


– each component is neural network

– simple building blocks model arbitrarily general functions

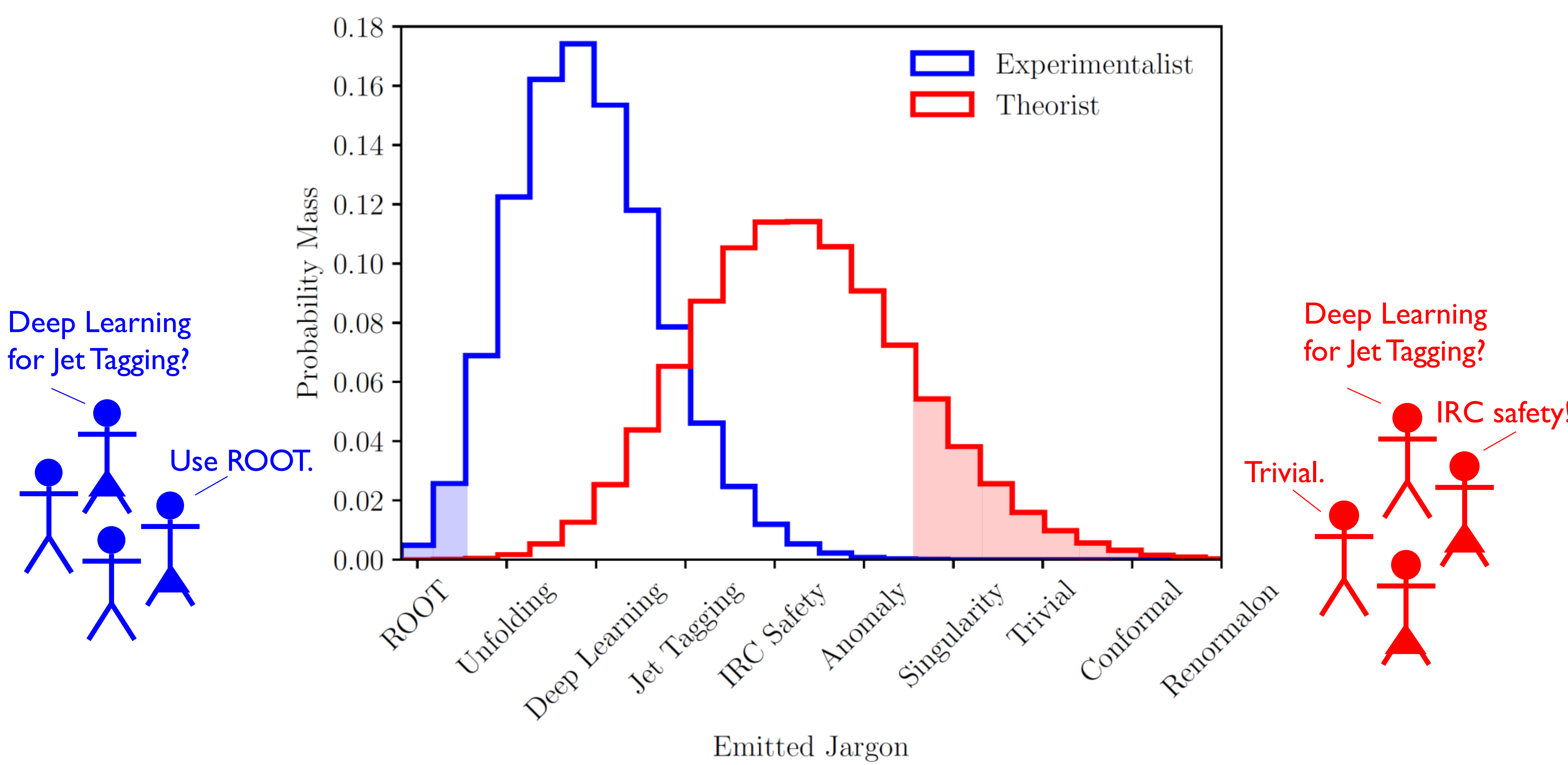
Physics-inspired generative models

(2) Generation



An Example

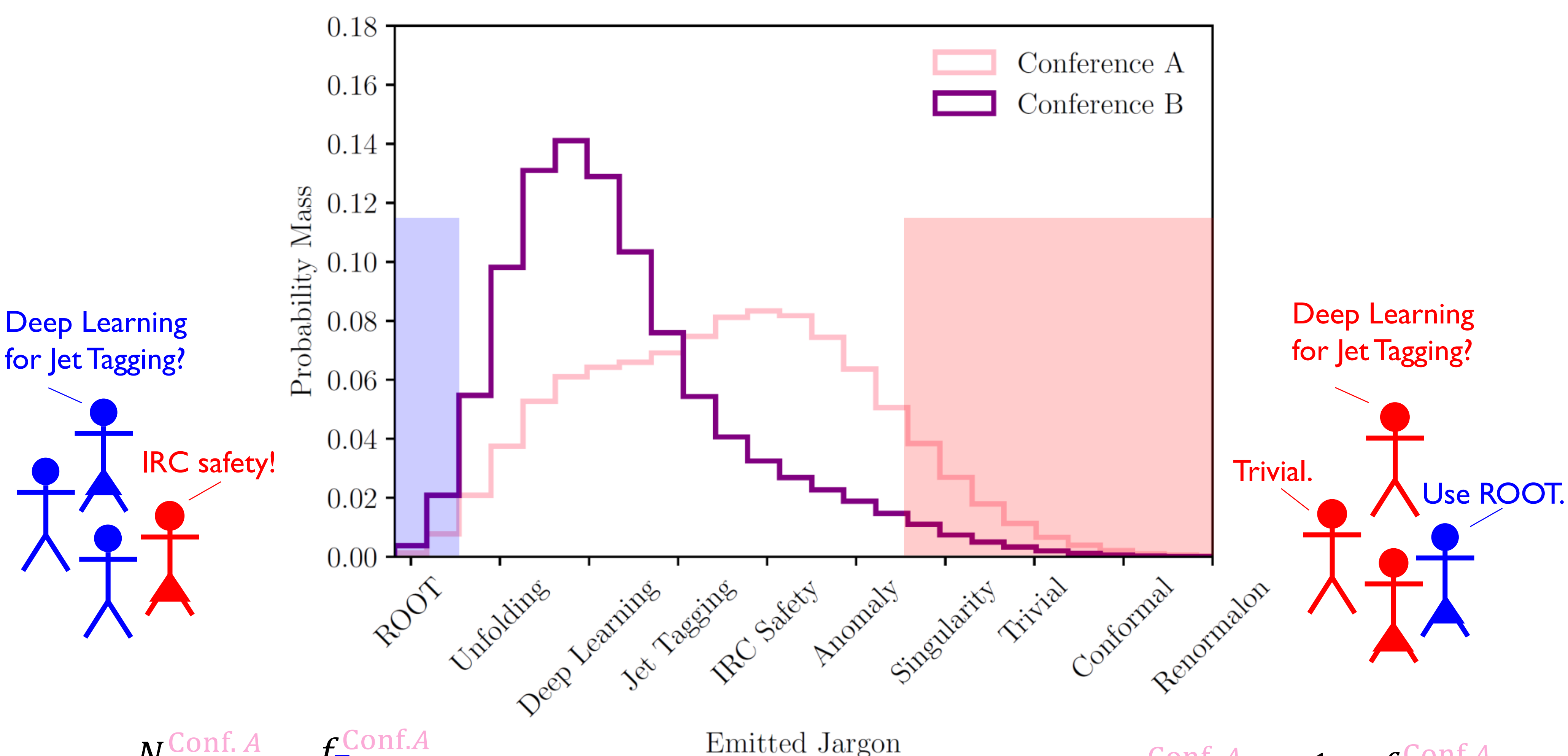
Let’s model physicists as random jargon emitters.



Weak supervision turned on its head

An Example

Listen to the jargon emitted from two different conferences.



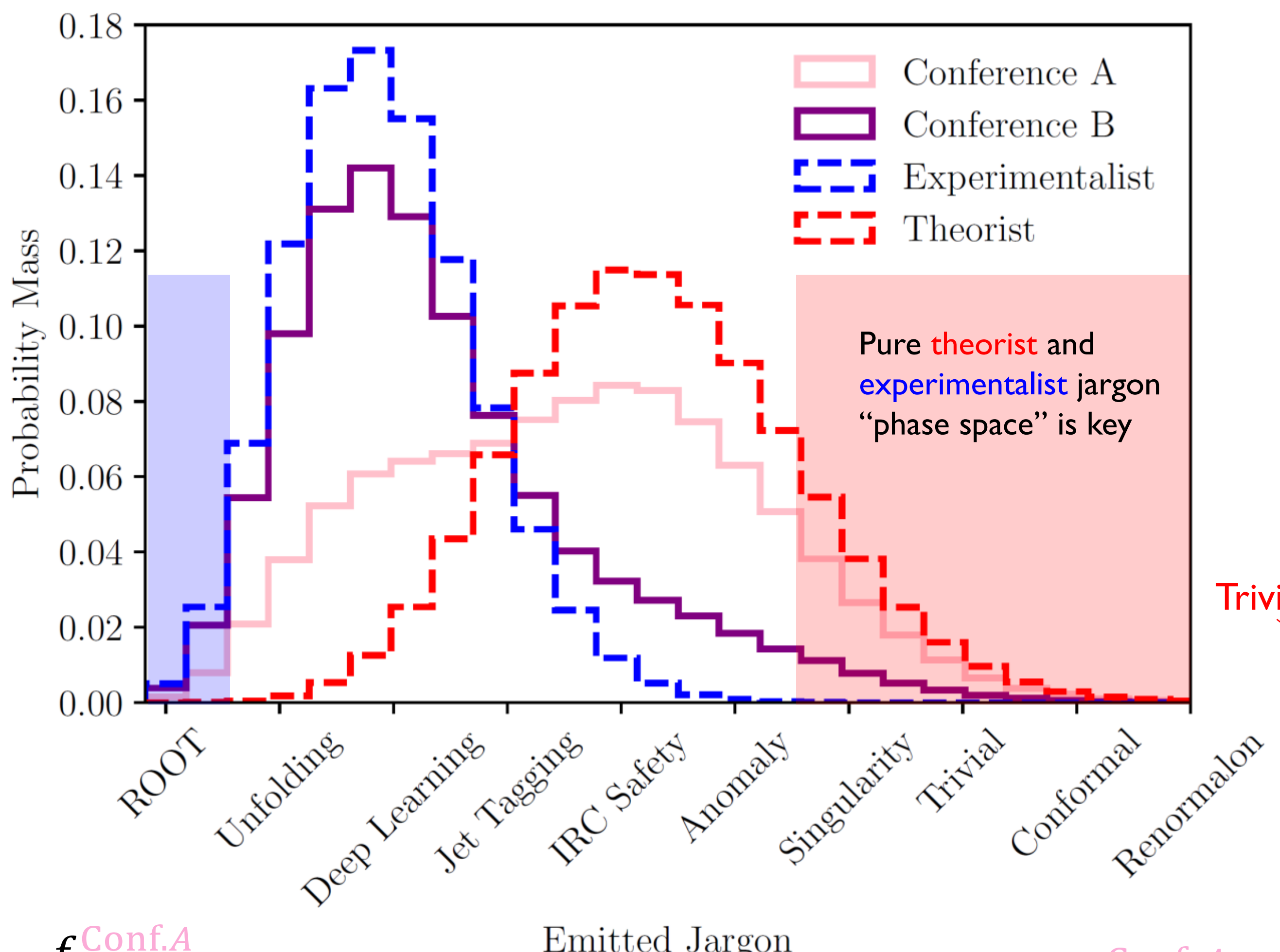
$$\frac{N^{\text{Conf. }A}_{\text{"ROOT"}}}{N^{\text{Conf. }B}_{\text{"ROOT"}}} = \frac{f^{\text{Conf.}A}_{\text{Expt.}}}{f^{\text{Conf.}B}_{\text{Expt.}}}$$

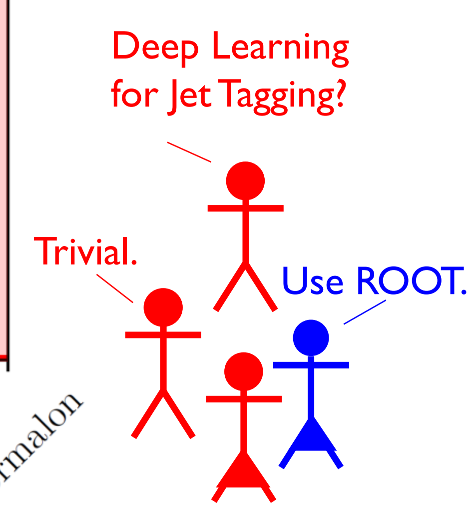
$$\frac{N^{\text{Conf. }A}_{\text{"Trivial"}}}{N^{\text{Conf. }B}_{\text{"Trivial"}}} = \frac{1 - f^{\text{Conf.}A}_{\text{Expt.}}}{1 - f^{\text{Conf.}B}_{\text{Expt.}}}$$

Weak supervision turned on its head

An Example

Disentangle theorist and experimentalist vocabularies from the jargon at conferences.





$$\frac{N^{\text{Conf. }A}_{\text{"ROOT"}}}{N^{\text{Conf. }B}_{\text{"ROOT"}}} = \frac{f^{\text{Conf.}A}_{\text{Expt.}}}{f^{\text{Conf.}B}_{\text{Expt.}}}$$

$$\frac{N^{\text{Conf. }A}_{\text{"Trivial"}}}{N^{\text{Conf. }B}_{\text{"Trivial"}}} = \frac{1 - f^{\text{Conf.}A}_{\text{Expt.}}}{1 - f^{\text{Conf.}B}_{\text{Expt.}}}$$

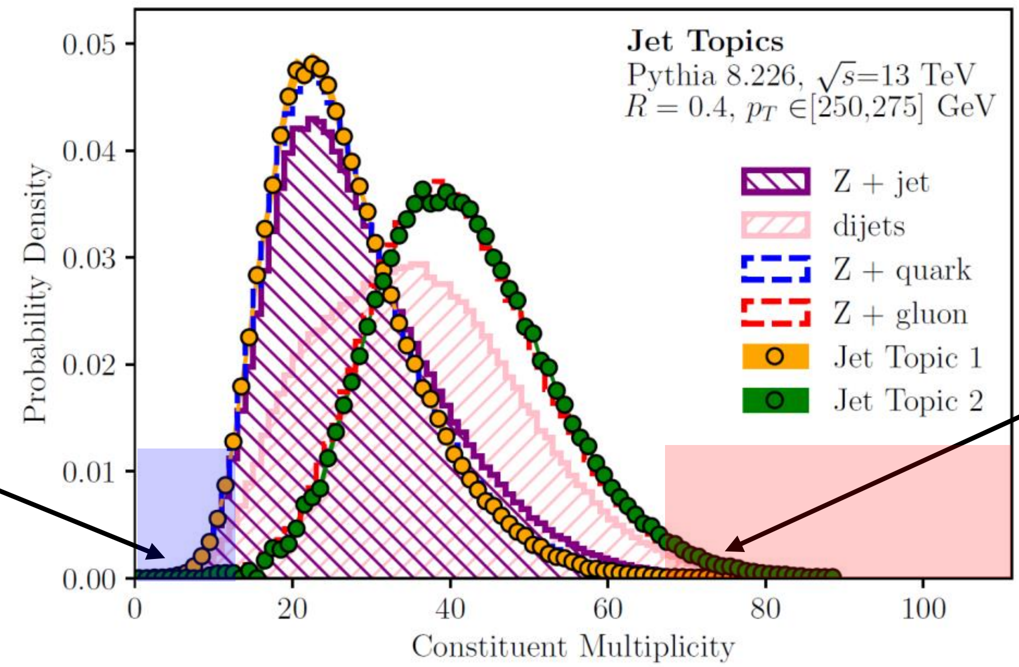
Weak supervision turned on its head

24

Demixing the mixtures

$$p_A(\boldsymbol{x}) = f_A^q\, p_{\text{quark}}(\boldsymbol{x}) + \left(1 - f_A^q\right) p_{\text{gluon}}(\boldsymbol{x})$$
$$p_B(\boldsymbol{x}) = f_B^q\, p_{\text{quark}}(\boldsymbol{x}) + \left(1 - f_B^q\right) p_{\text{gluon}}(\boldsymbol{x})$$

$$\kappa_{\text{BA}} \equiv \min_{\boldsymbol{x}} \frac{p_B(\boldsymbol{x})}{p_A(\boldsymbol{x})} = \frac{f_B^q}{f_A^q}$$



$$\kappa_{\text{AB}} \equiv \min_{\boldsymbol{x}} \frac{p_A(\boldsymbol{x})}{p_B(\boldsymbol{x})} = \frac{1-f_A^q}{1-f_B^q}$$

With **reducibility factors** κ_{AB} and κ_{BA}, solve for the quark and gluon fractions and distributions:

$$f_A^q = \frac{1-\kappa_{\text{AB}}}{1-\kappa_{\text{AB}}\kappa_{\text{BA}}} \qquad f_{\text{B}}^q = \frac{\kappa_{\text{BA}}(1-\kappa_{\text{AB}})}{1-\kappa_{\text{AB}}\kappa_{\text{BA}}}$$

$$p_{\text{quark}}(\boldsymbol{x}) = \frac{p_A(\boldsymbol{x}) - \kappa_{\text{AB}}\, p_B(\boldsymbol{x})}{1-\kappa_{\text{AB}}} \qquad p_{\text{gluon}}(\boldsymbol{x}) = \frac{p_B(\boldsymbol{x}) - \kappa_{\text{BA}}\, p_A(\boldsymbol{x})}{1-\kappa_{\text{BA}}}$$

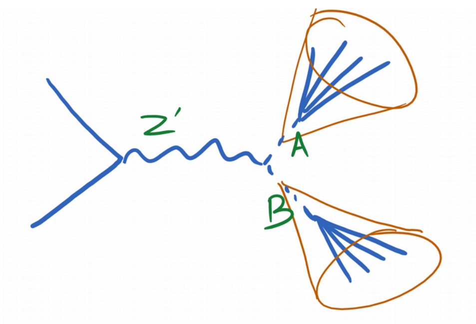
Weak supervision for new physics?

Jack Collins

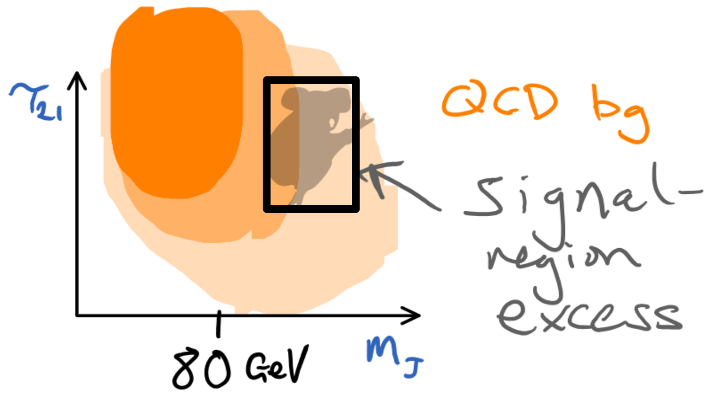
Case Study: Dijet Resonances

CWoLa Hunting Proposal

1) Propose *generic* signal resonance hypothesis



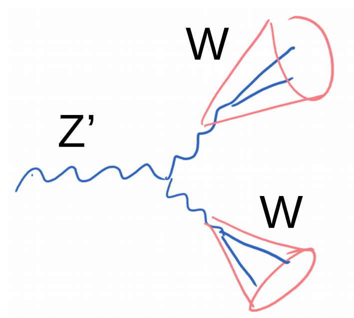
2) Let Machine look for signal region overdensity in some substructure phase space region



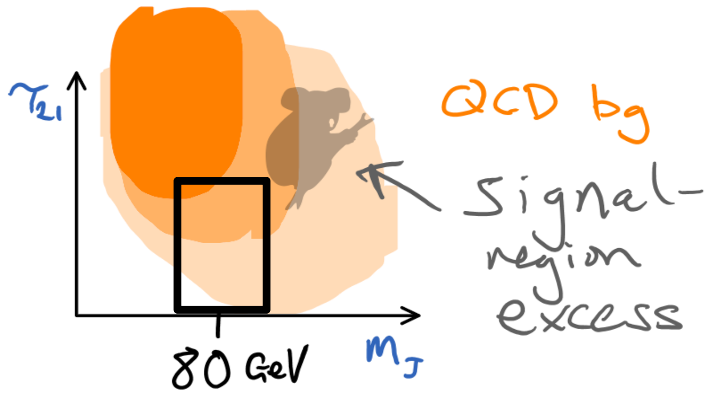
3) Perform bump-hunt in distribution of selected events

How do existing di-SM searches use substructure information?

1) Propose *specific* signal resonance hypothesis



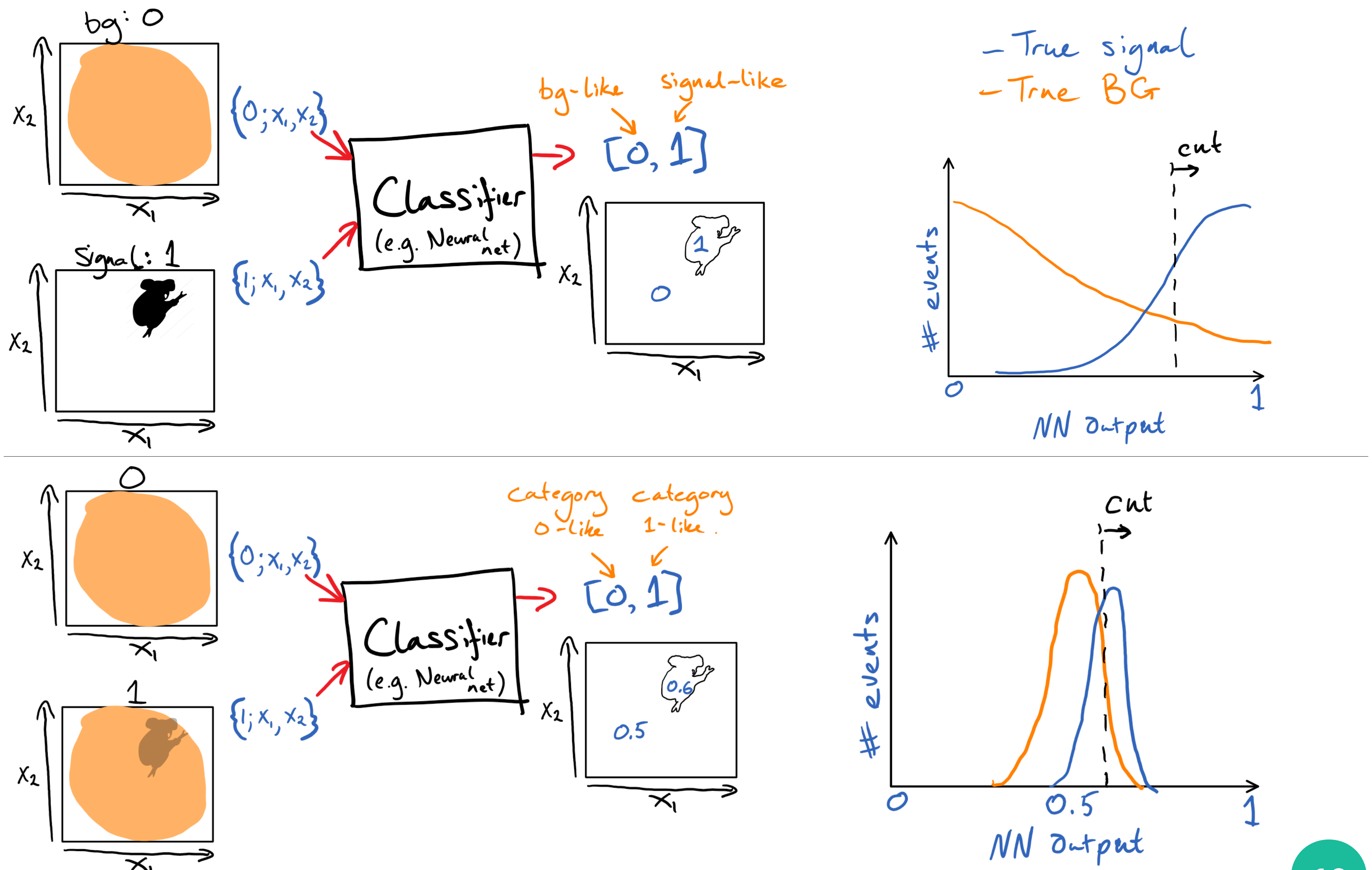
2) Apply pre-defined model-specific substructure cut using a couple of expert variables or supervised NN



3) Perform bump-hunt in distribution of selected events

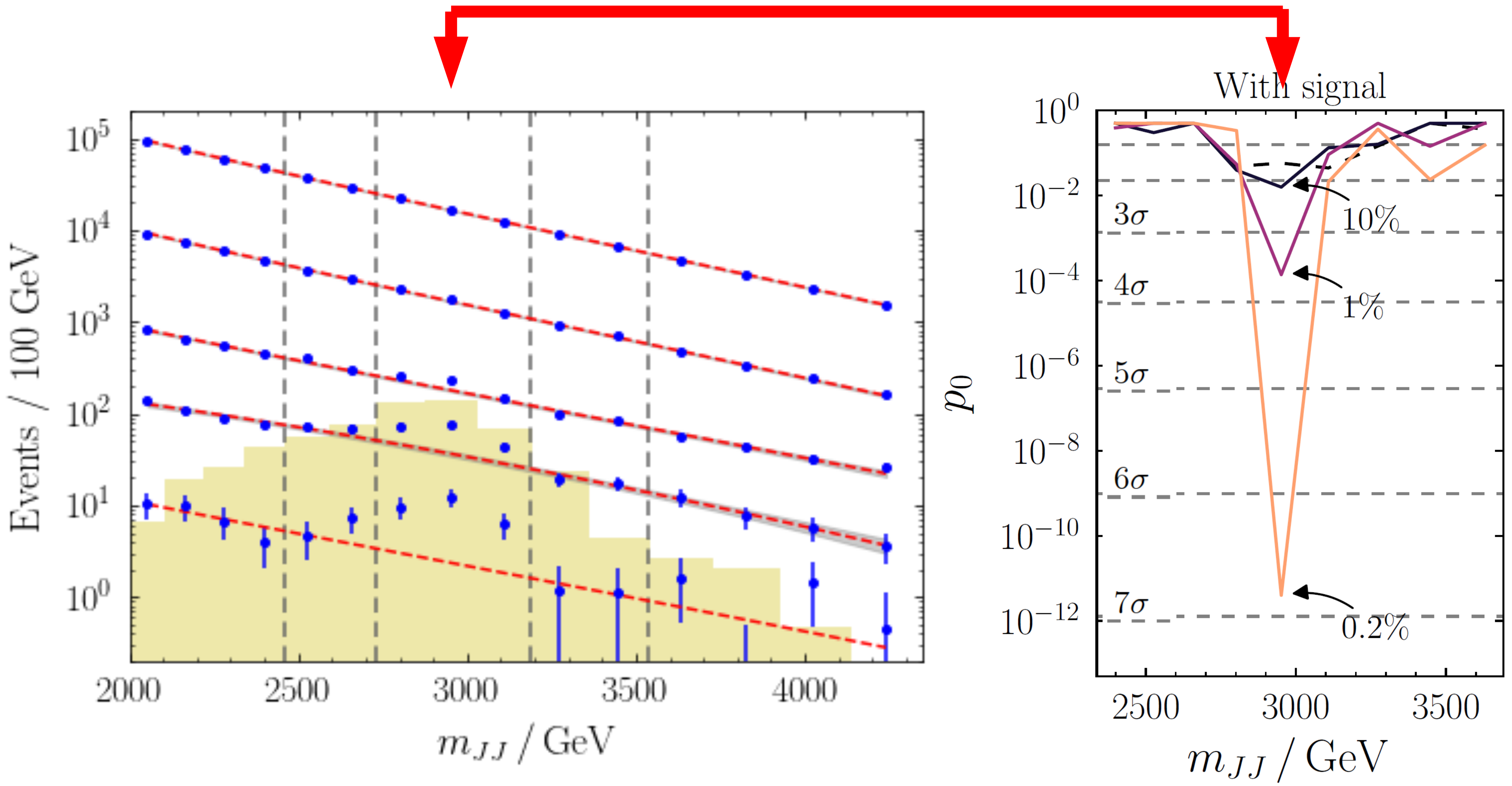
Weak supervision for new physics?

Fully Supervised learning vs Weak Supervision

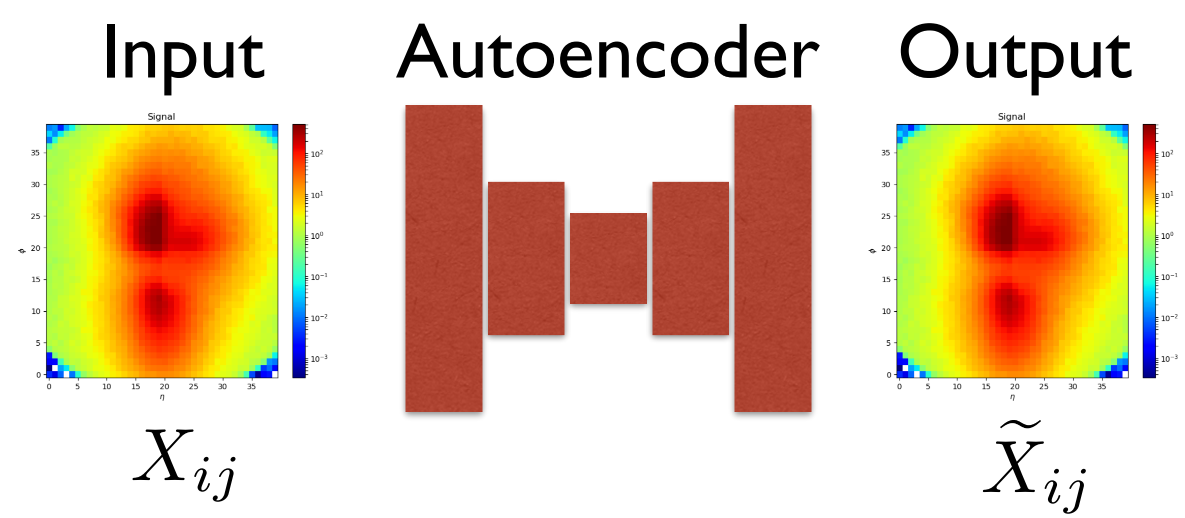


Weak supervision for new physics?

Mass Scan

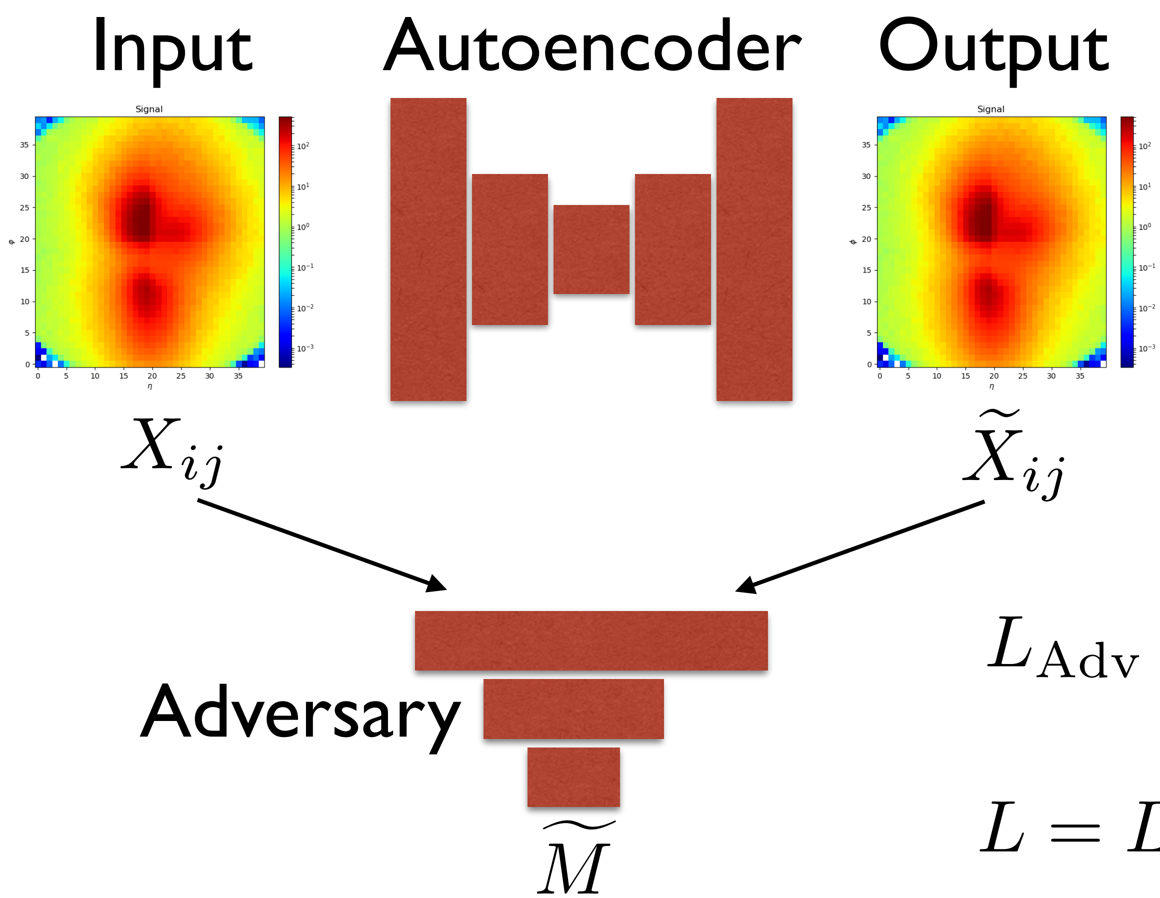


Autoencoder for Physics



- Train on pure QCD light quark/gluon jets
  - Use top tagging reference sample https://goo.gl/XGYju3
  - Train **only on QCD** events
- New physics identified as anomaly
  - Tail of the loss function

Combined Setup



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

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

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

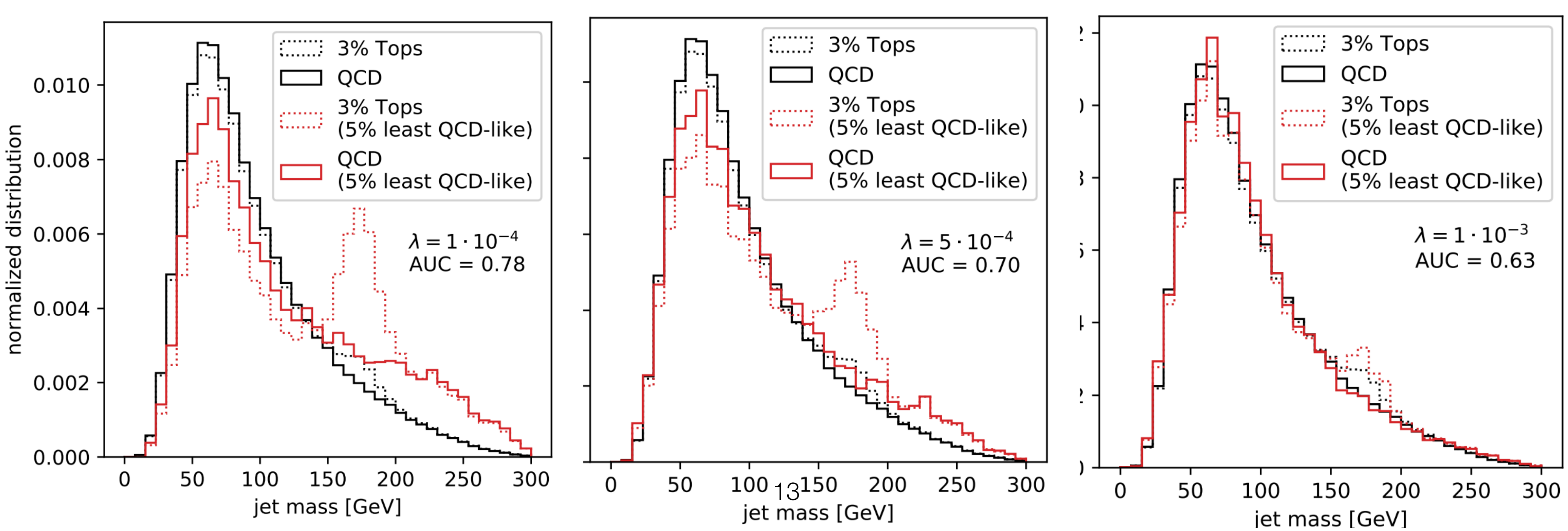
Decorrelated Jet Substructure Tagging using Adversarial Neural Networks
C Shimmin, P Sadowski, P Baldi, E Weik, D Whiteson, E Goul, A Søgaard 1703.03507

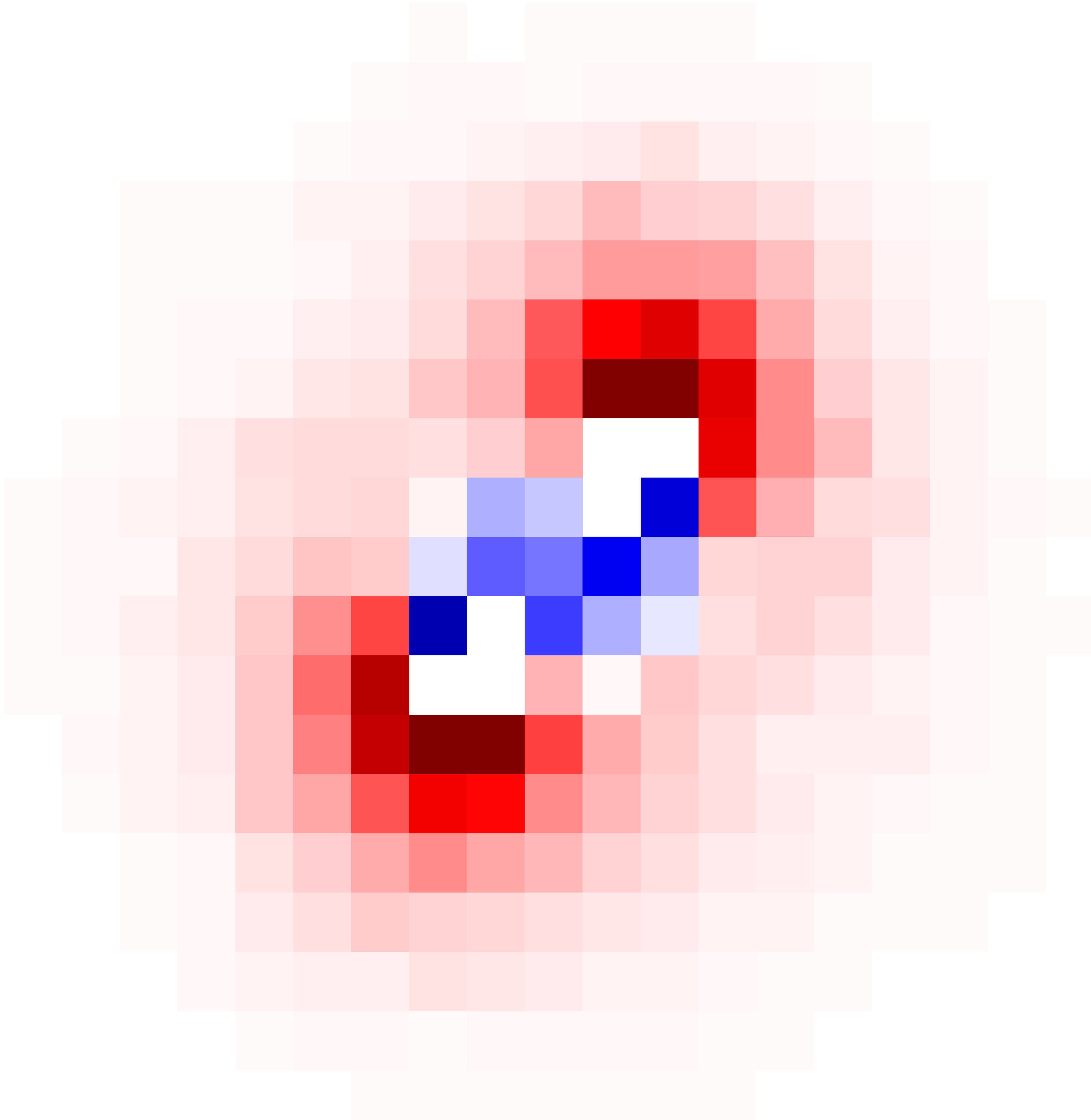
Mass Sculpting

- Counteract with adversary:

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

- Tune mass dependency with Lagrange multiplier
  - Defines control regions in data





Fin.