Deep Learning In the Wild: Application on ATLAS

Michael Kagan

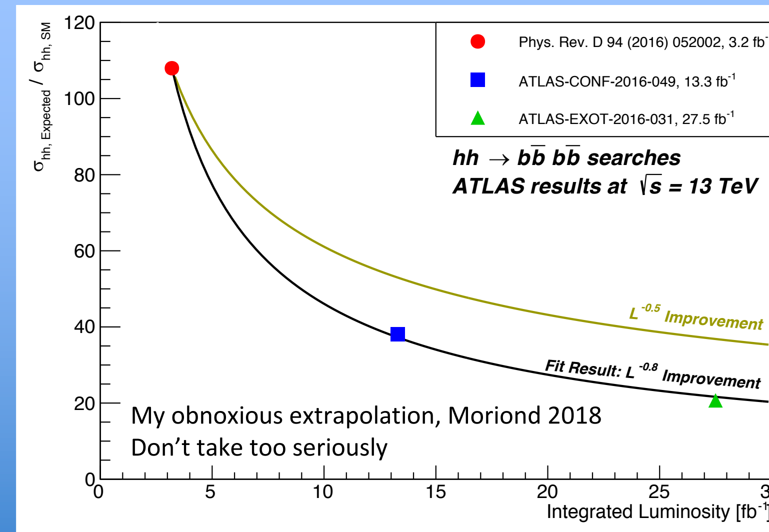
SLAC

Machine Learning at the LHC, KMI, Nagoya University
February 5th , 2020

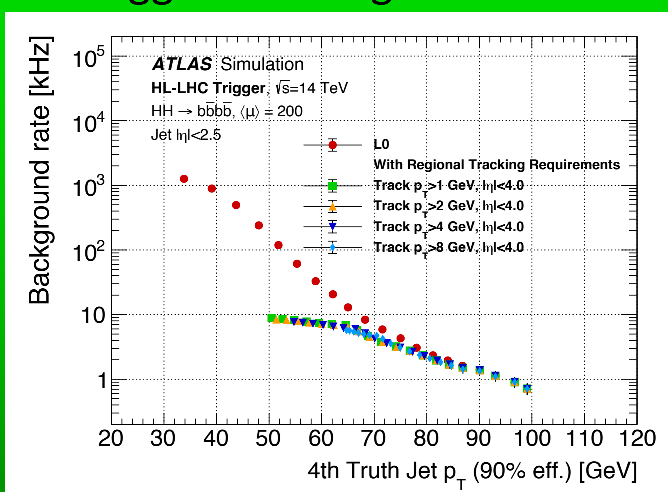
Challenges for ML on ATLAS?

- Successfully physics program at the LHC requires overcoming major hardware, computing, and analysis challenges!
- How will the increasing power of ML play a role?

Analysis Challenges

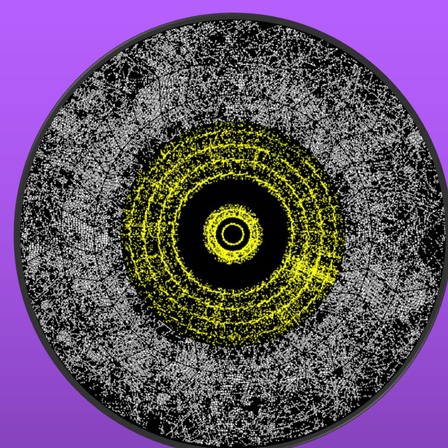


Trigger Challenges

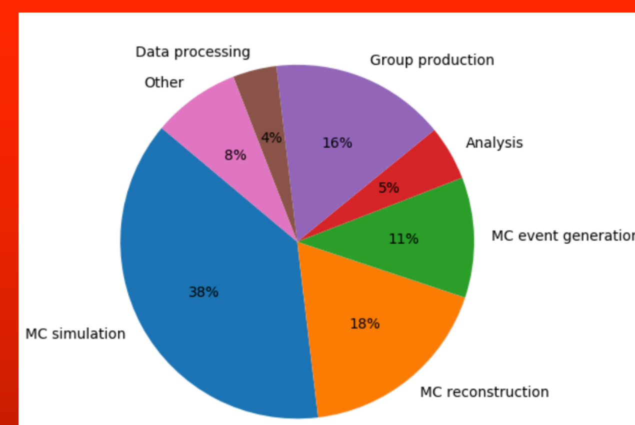


[ATLAS Phase-II TDAQ TDR]

Tracking Challenges



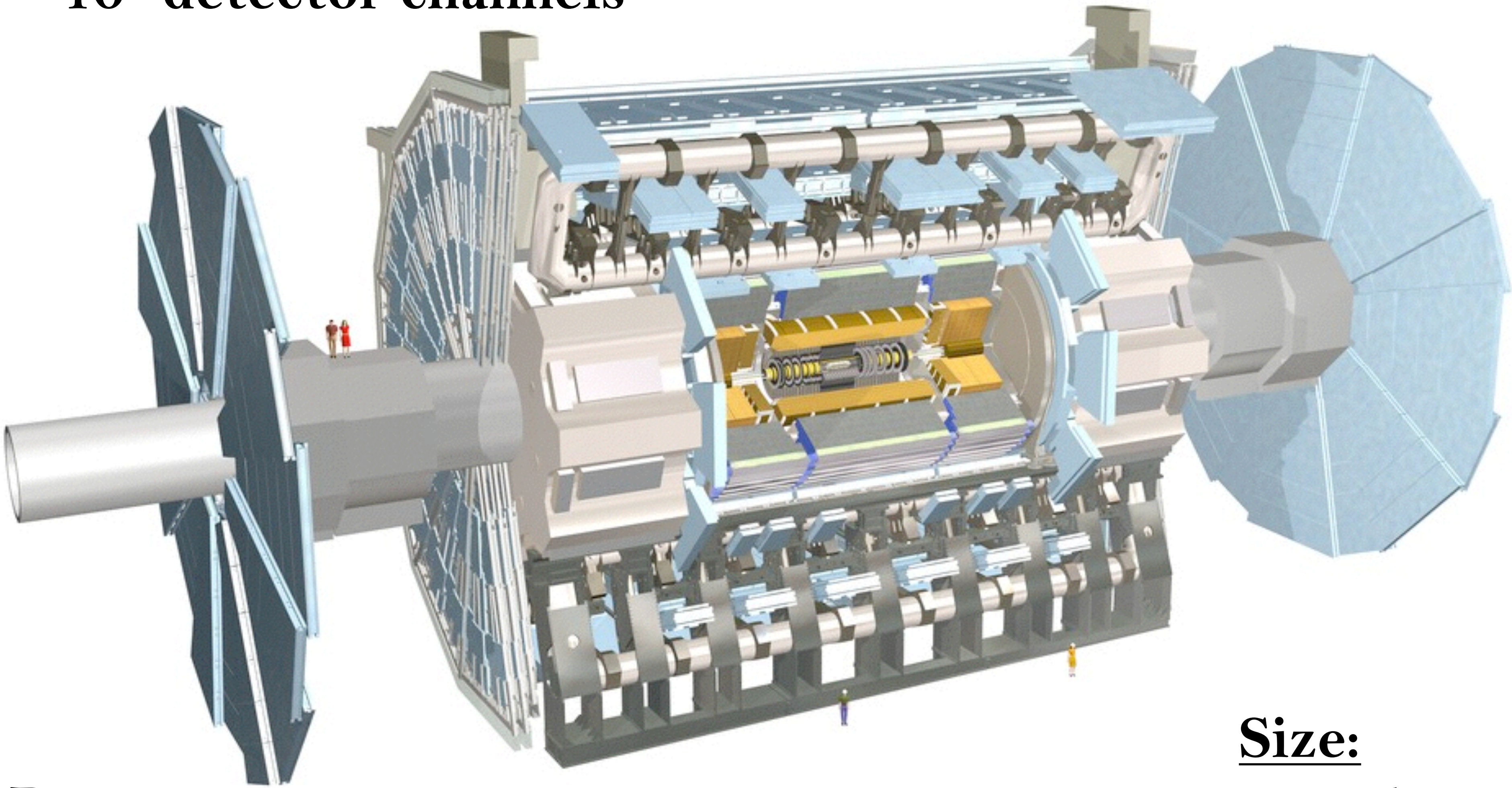
Simulation Challenges



[Image credit: D. Costanza, J. Catmore]

The ATLAS Experiment

~10^8 detector channels



Data:

~300 MB / sec

~3000 TB / year

Weight:

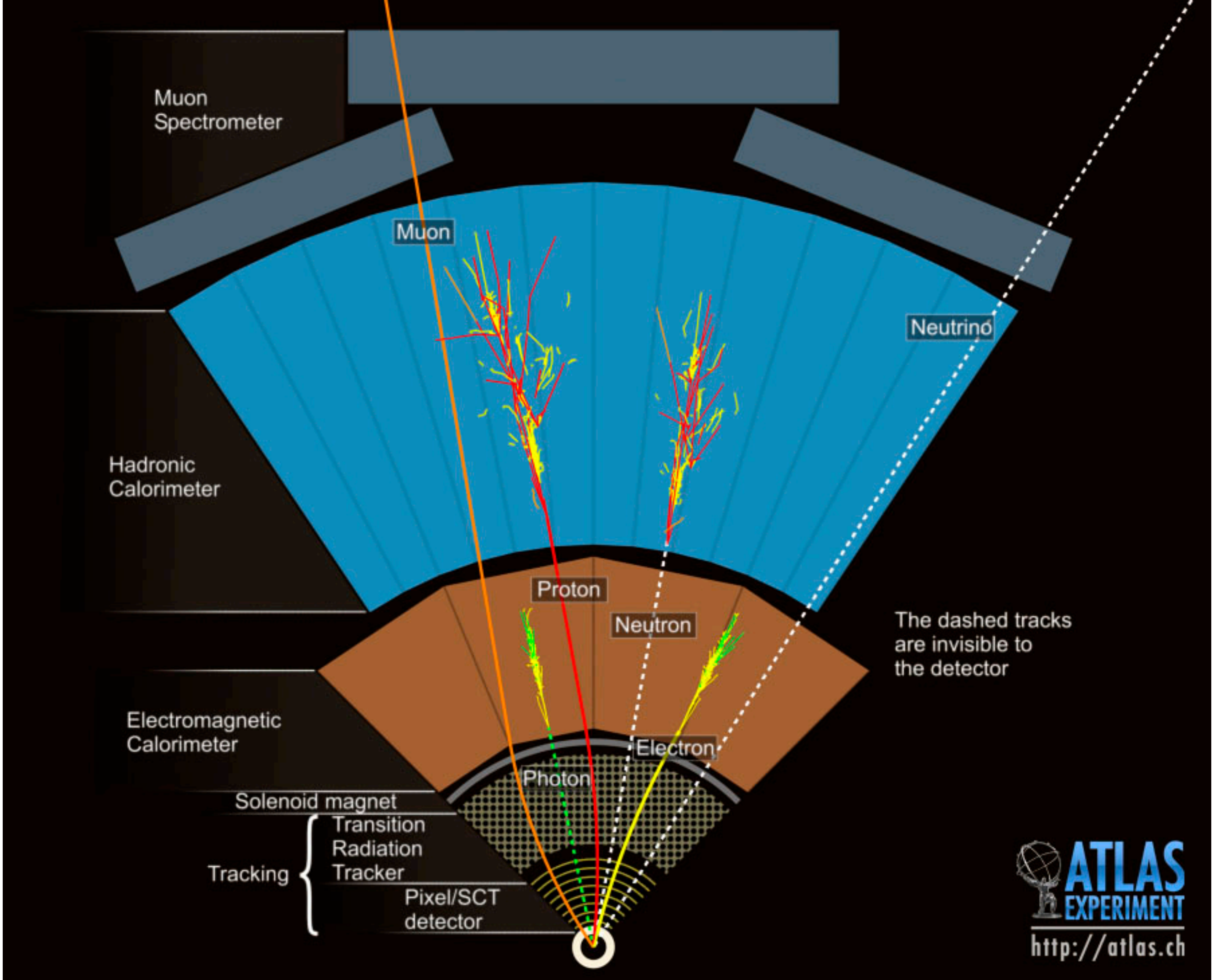
7000 tons

Size:

46 m long,

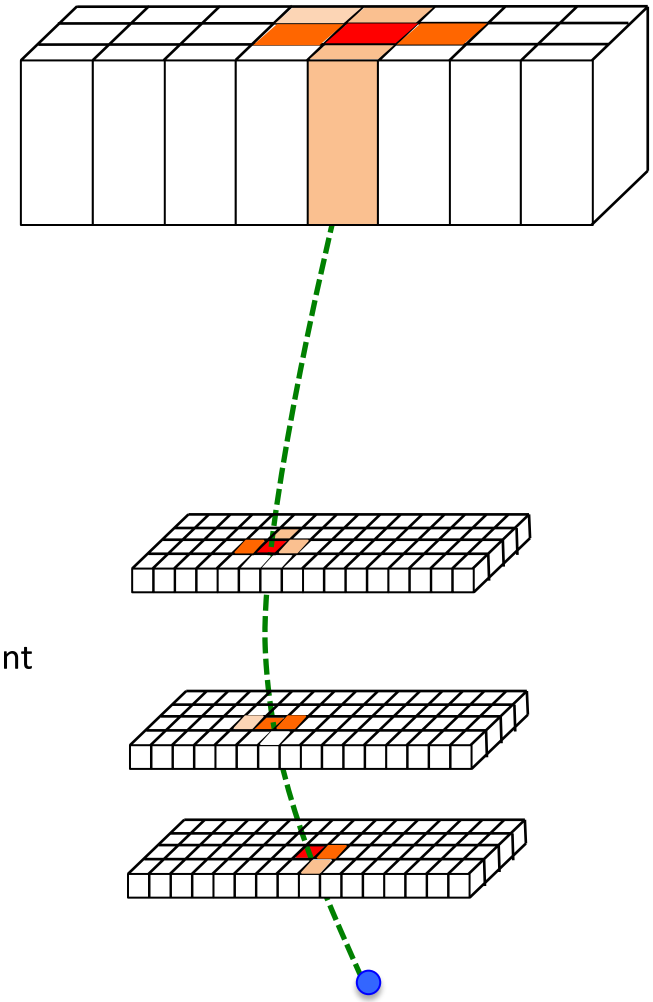
25 m high,

25 m wide


Muon Spectrometer
Hadronic Calorimeter
Electromagnetic Calorimeter
Solenoid magnet
Tracking
Transition Radiation Tracker
Pixel/SCT detector
Muon
Neutrino
Proton
Neutron
Electron
Photon
The dashed tracks are invisible to the detector
ATLAS EXPERIMENT
http://atlas.ch

Reconstructing Particles

<u>Calorimeter:</u>
Stops particle and destructively measure energy / direction



<u>Tracking detector:</u>
Typically Si-pixel detector
Non-destructive space-point measurement

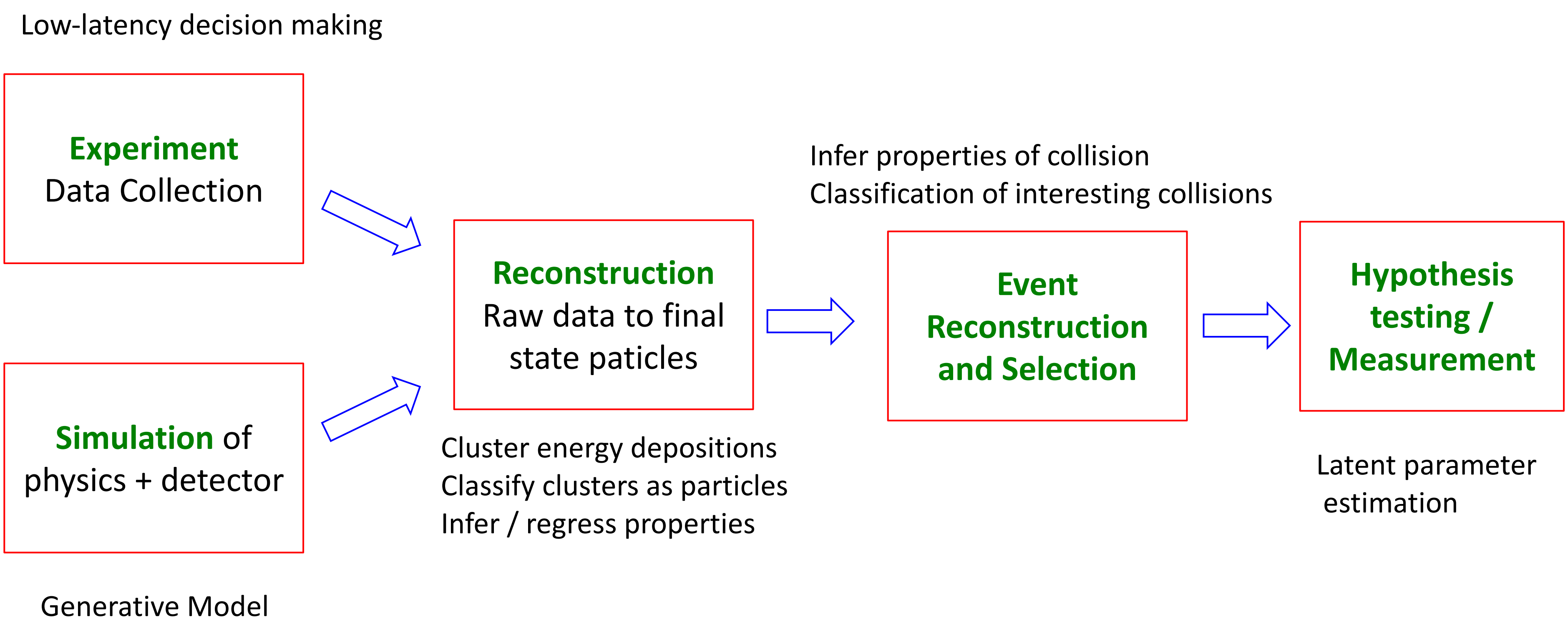
- Particle identification = Classification

$$p(\text{electron} \mid \text{data})$$

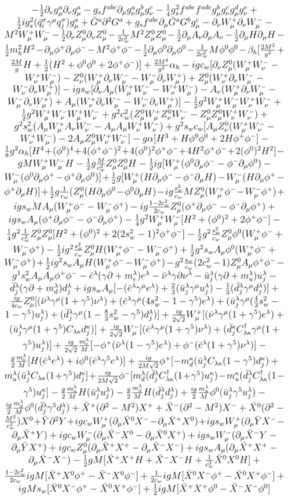
- Energy estimation = Inference, regression

$$p(E_{\text{true}}^{\text{electron}} \mid \text{electron data})$$

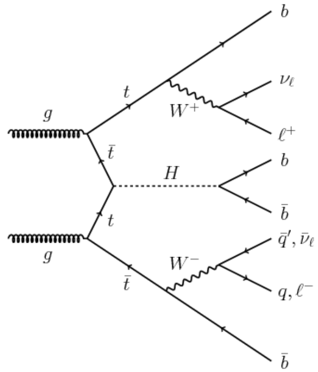
Data Analysis Pipeline



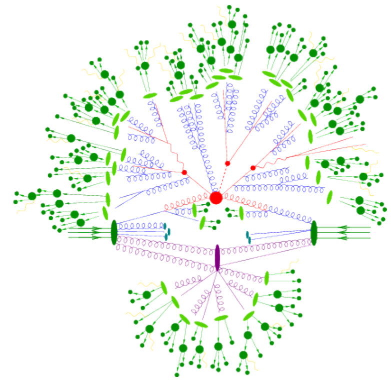
From Theory to Experiment… and Back



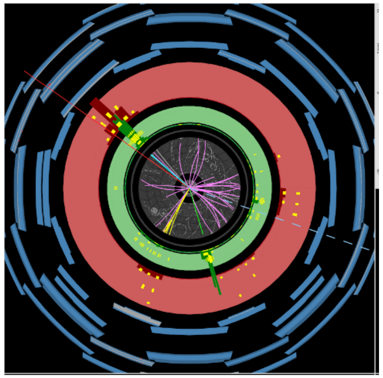
Parameters θ



O(10) particles



O(100) particles



O(10^8) detector elements

Hypothesis testing / Measurement ← **Event Reconstruction and Selection** ← **Reconstruction of particles** ← **Detector data & Simulation**

Slide credit: K. Cranmer

ML Across the ATLAS Analysis Pipeline

- **Reconstruction** - discrimination / regression problems, leading to fastest uptake of new ideas
    - Jets: Tagging, Calibration, Decorrelation
    - Missing Energy – Pileup Subtraction
    - Jet Flavour Tagging
    - Tau Particle ID
    - Pixel Clustering for Tracking

- **Simulation** – problems of density estimation and sampling
    - – Fast Calorimeter Simulation

- **Analysis** - largely well known ML methods for signal vs background discrimination and event reconstruction
    - – Very impactful, but not going to talk about this much

- ML applications growing more sophisticated
    - – Classification → Density estimation → Differentiable Programs

ML Across the ATLAS Analysis Pipeline

- **Reconstruction** - discrimination / regression problems, leading to fastest uptake of new ideas
  - Jets: Tagging, Calibration, Decorrelation

How can existing ML help us do better in tackling known challenges → How can existing ML allows us to approach new challenges → How can we design ML systems to our needs

- ML applications growing more sophisticated
  - Classification → Density estimation → Differentiable Programs

Looming Gap in Theory vs Practice

- Large difference between what is done in phenomenology papers and on the experiment (at least what is public)

- Why????
  - Real detector models have MUCH more complex noise than simplified simulation
    - Method performance doesn't necessarily transfer
    - Even the ones that transfer can be hard to tune

  - Calibration – even our best simulations for training are not perfect
    - After training the algorithm, we still have to calibrate!

  - Information disconnect
    - Model expertise may be outside experiment
    - Different people build and calibrate algorithm within ATLAS

  - Experiment computational resources may not be well suited to ML
    - Moreover, full data re-processing alone can take months

  - Resistance to change

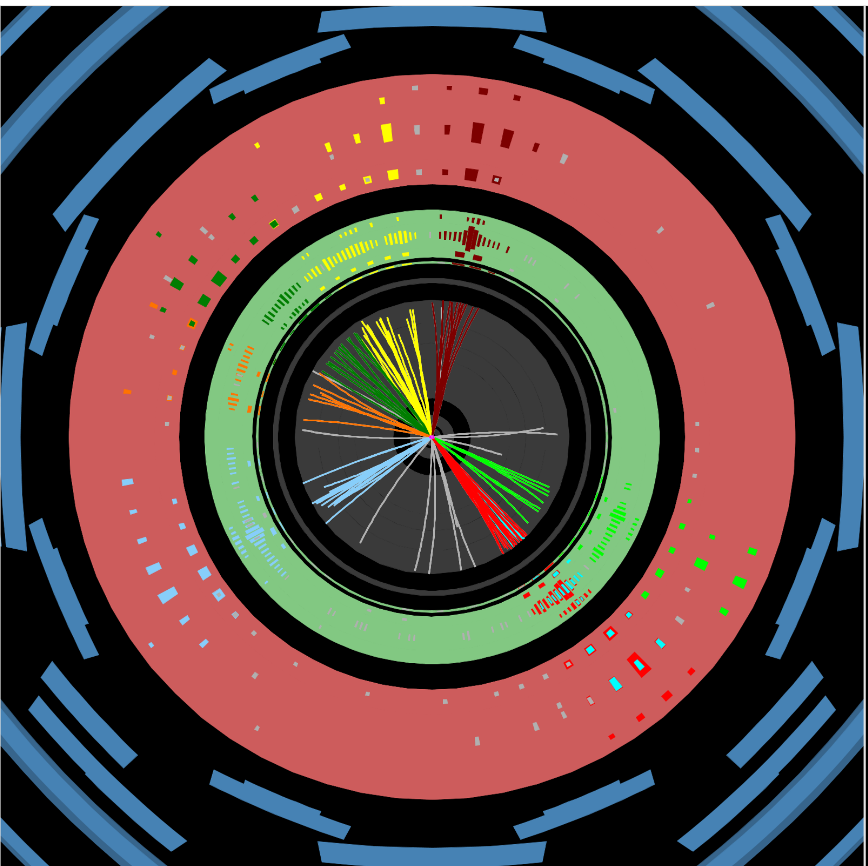
Classification and Regression in Reconstruction

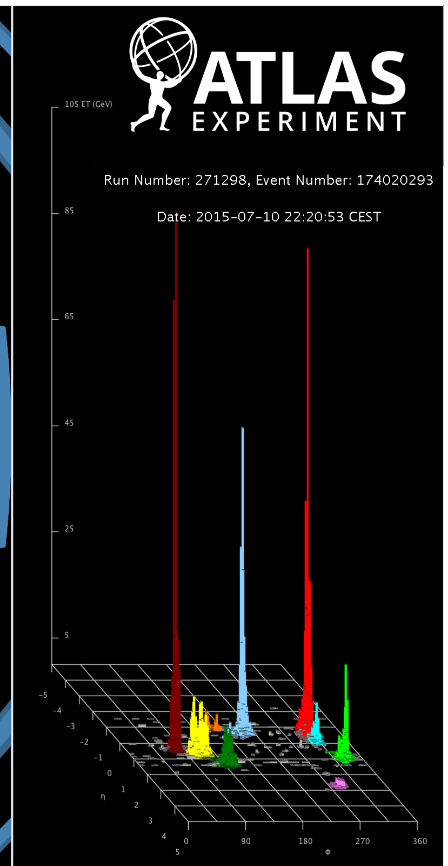
Jets at the LHC

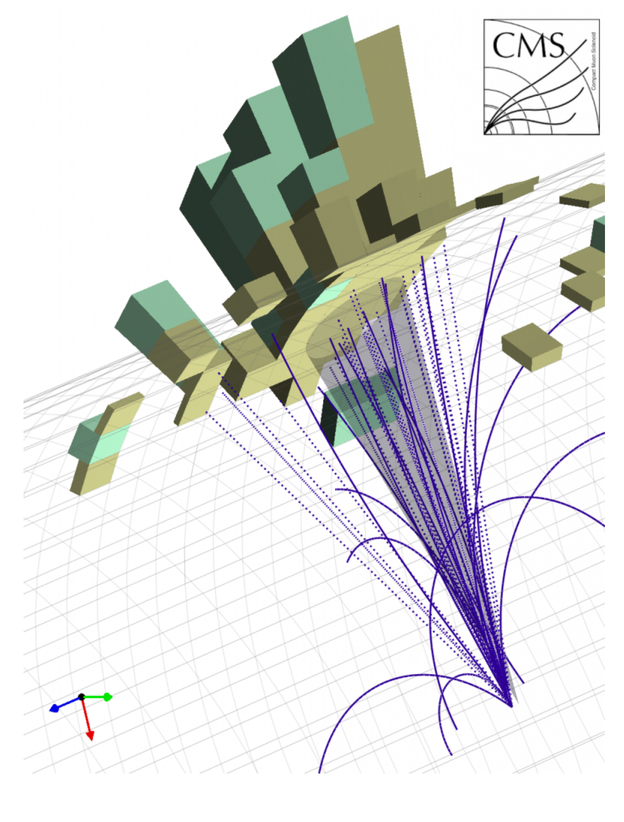
- Jets are formed by clustering energy depositions in calorimeter with the anti-k_T algorithm

- Jet identification = Classification: $p(parent\ particle\,|\,jet\ cluster)$

- Energy estimation = Inference, regression: $p(E_{true}^{jet}\,|\,jet\ cluster)$







Canonical Discrimination Problem: Jet Identification

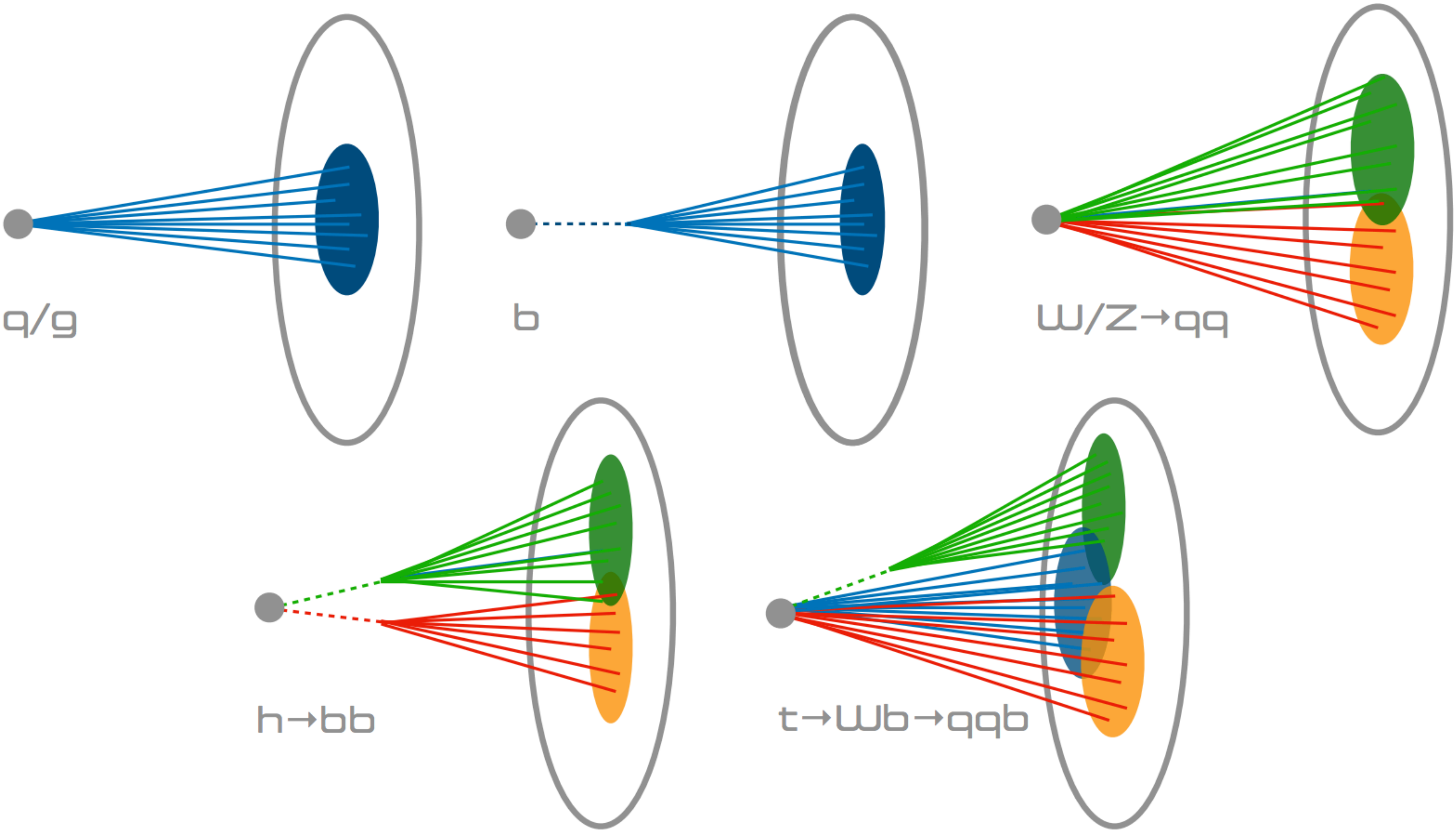
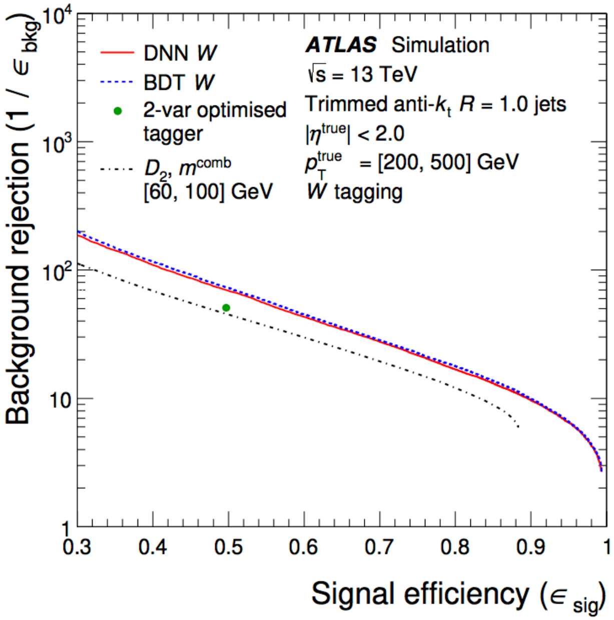
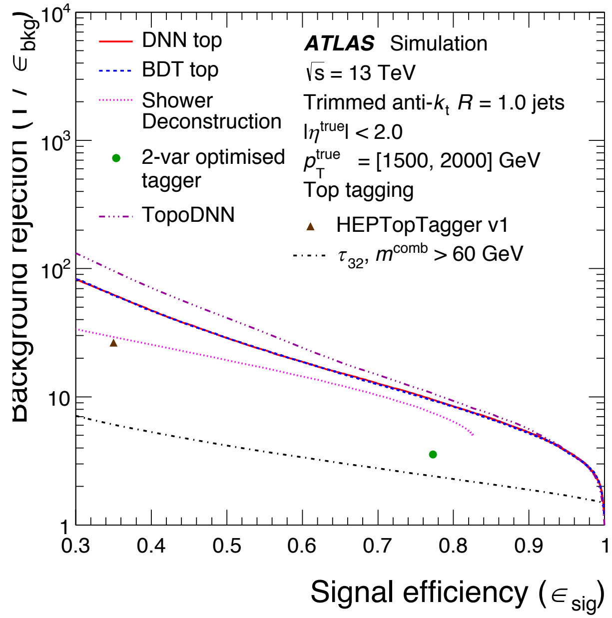


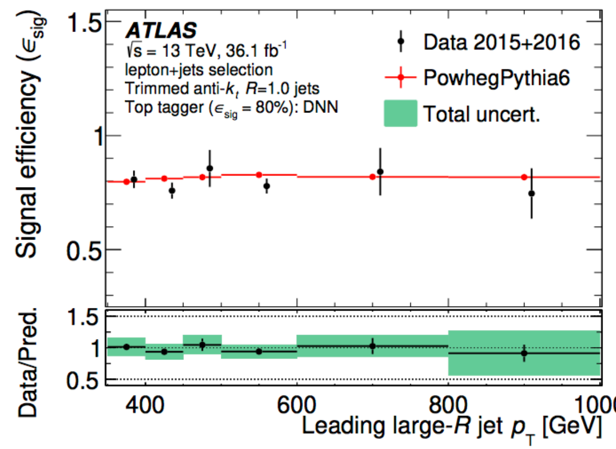
Image Credit: arXiv:1909.12285

Combining Substructure Variables



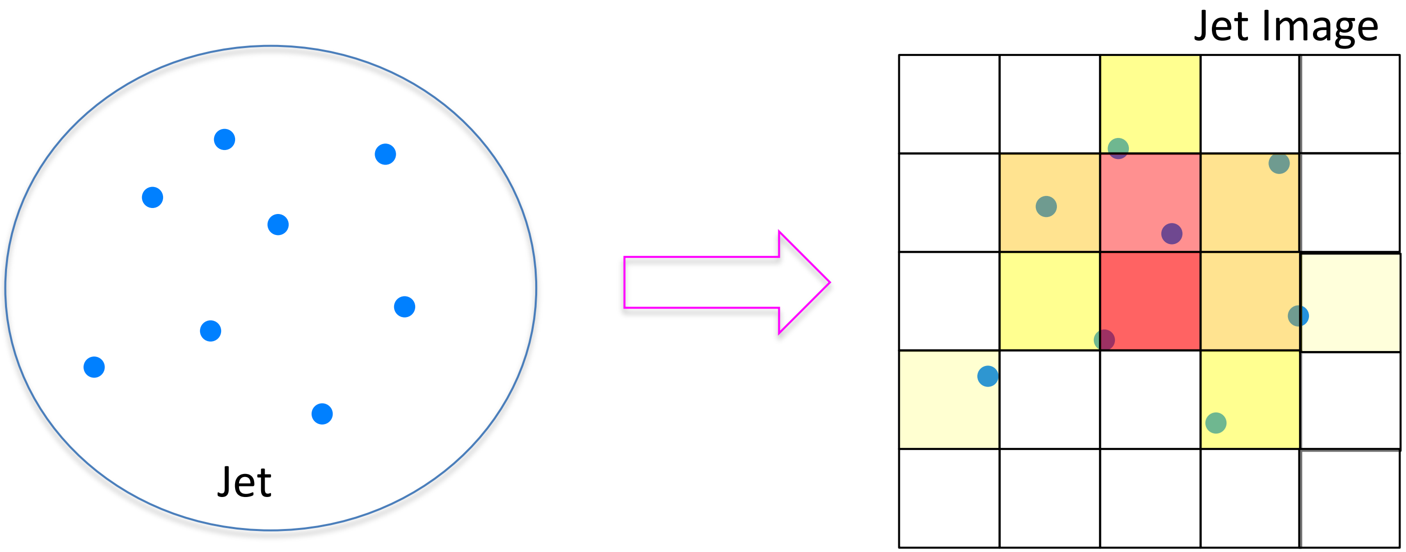


[arXiv:1808.07858](https://arxiv.org/abs/1808.07858)



- Wide array of physics insight has gone into developing jet substructure observables
- Direct application of ML for combining power of multiple partially correlated substructure features
- First calibrations look quite reasonable!

Jets as Images

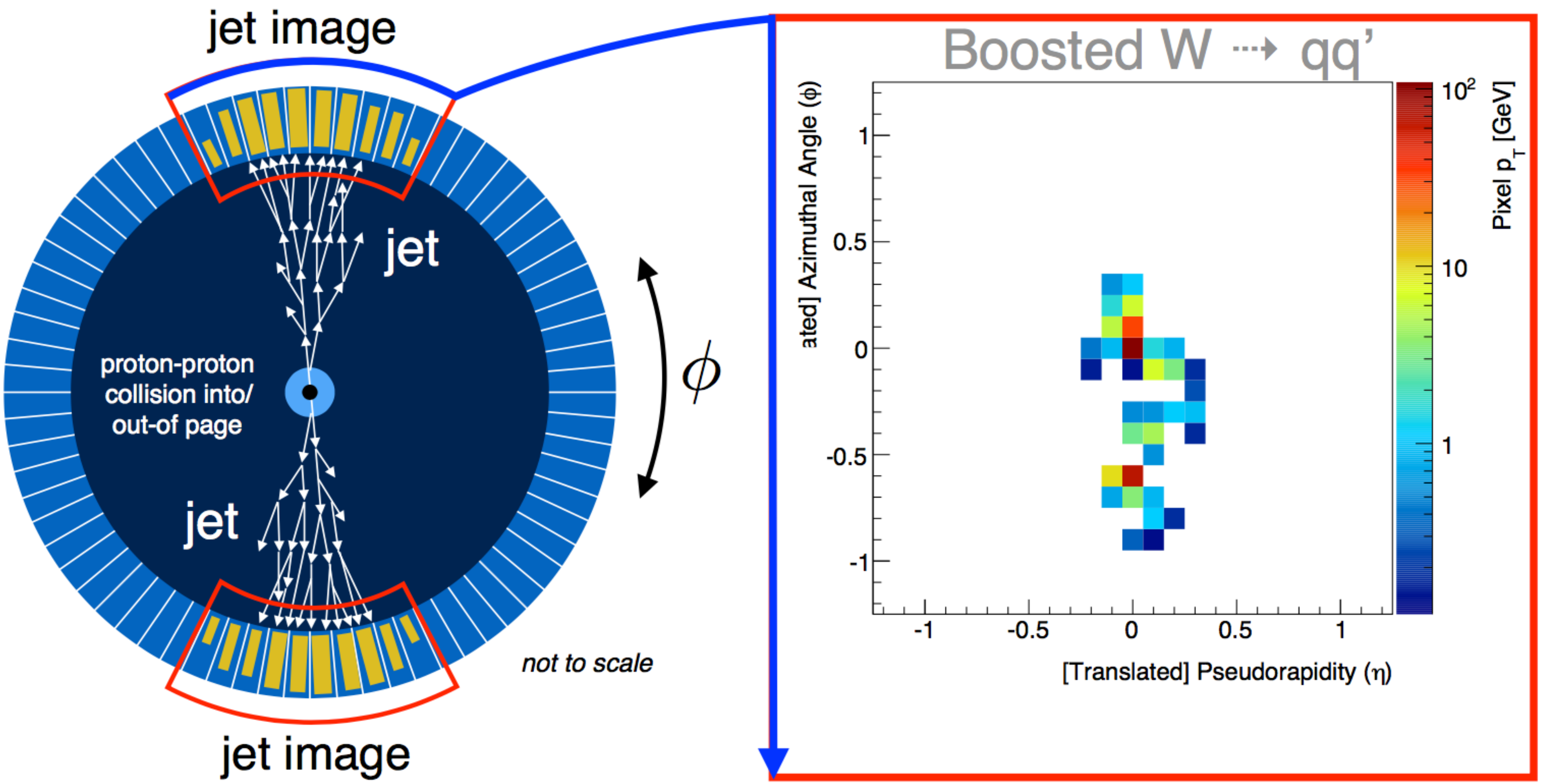


- A jet induces a distribution of energy over $\eta - \phi$
  - Essentially how a jet is seen by calorimeters

- **Jet-image** – fixed size 2D representation of the jet as a distribution of energy

Jet Images

Unrolled slice of detector



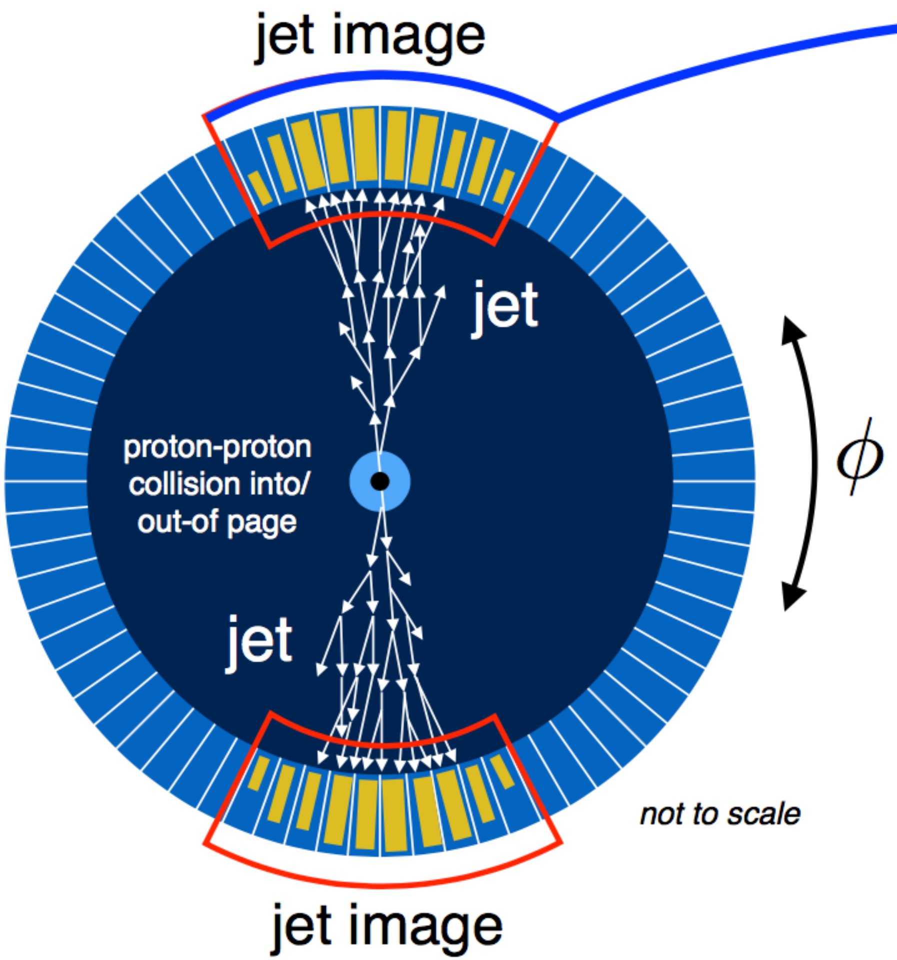
Calorimeter towers as pixels

Energy depositions as intensity

Image Credit: [B. Nachman]

Jet Images

Average of large number of Jet Images



W-jets

QCD-jets

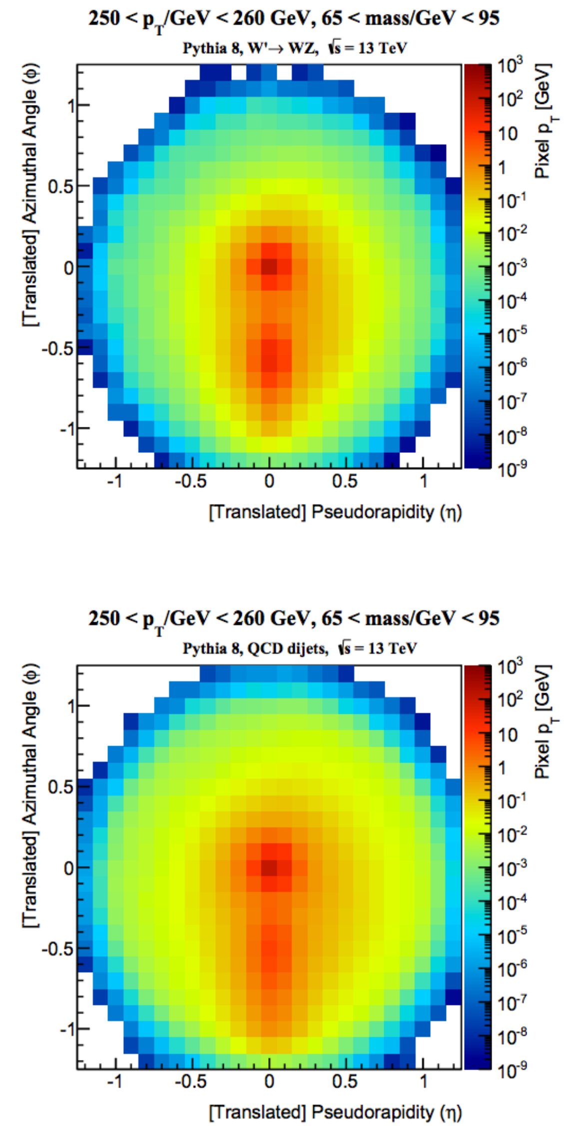
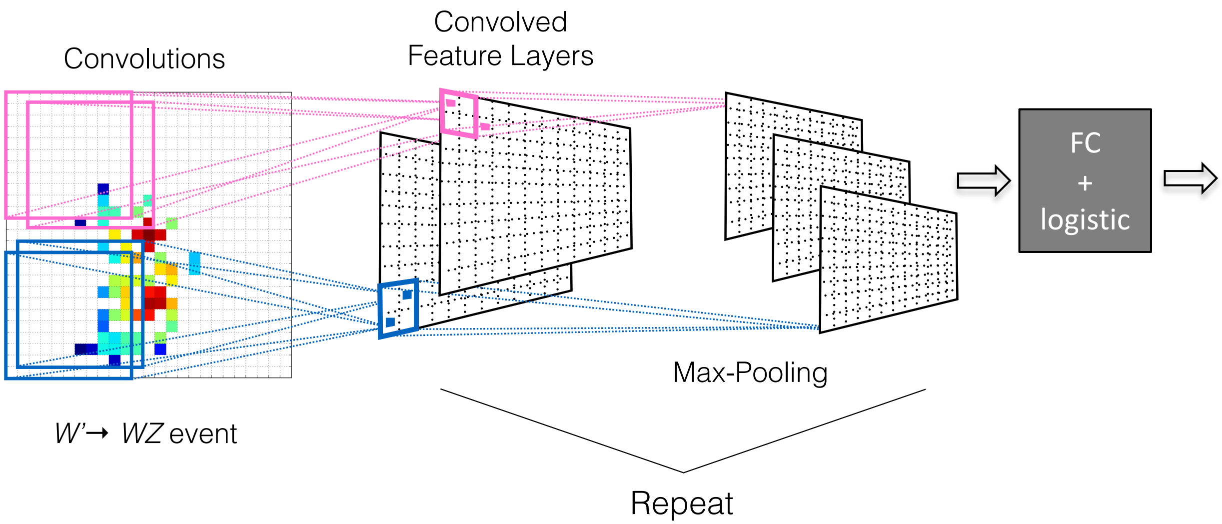
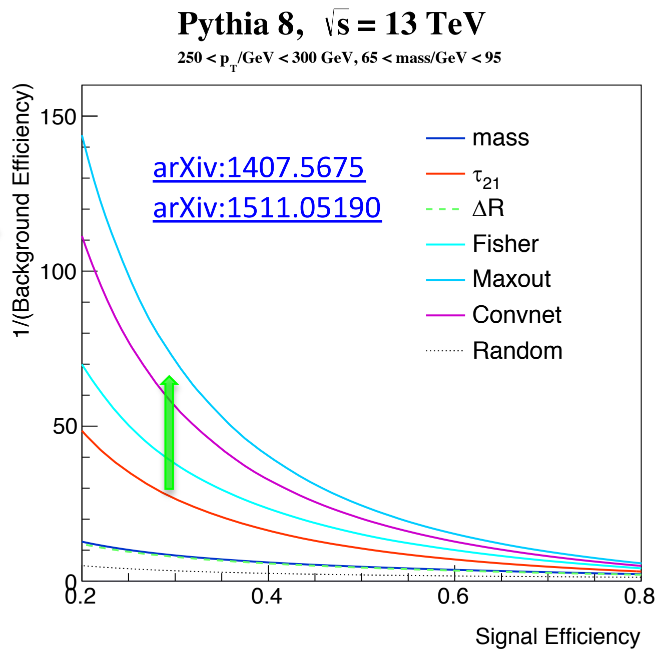


Image Credit: [B. Nachman]

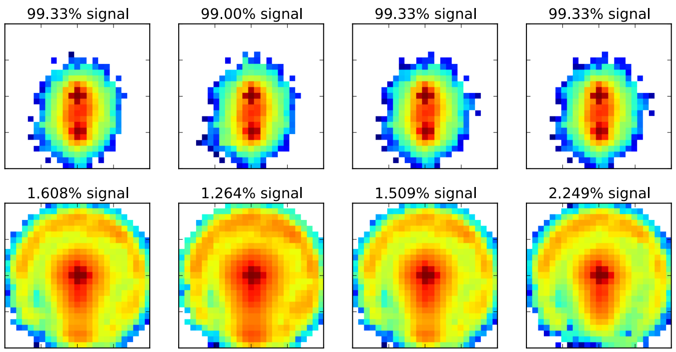
CNNs + Jet Images





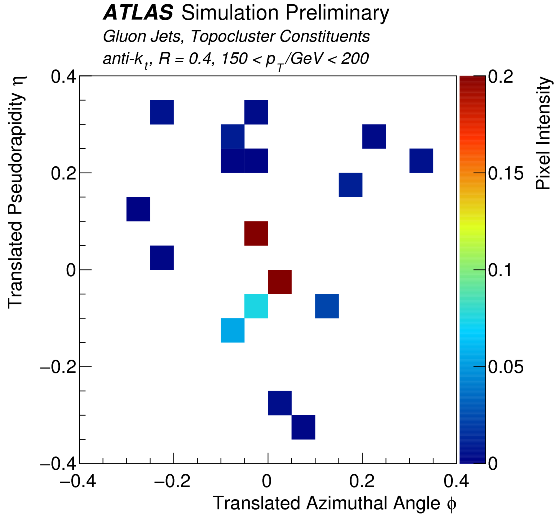
- CNN's applied to Jet images can lead to large performance gains

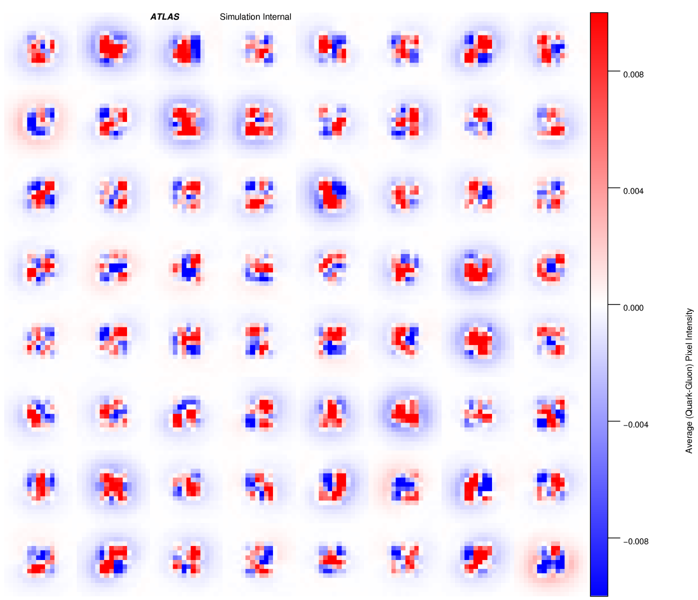
- More on Classification in T. Plehn's talk

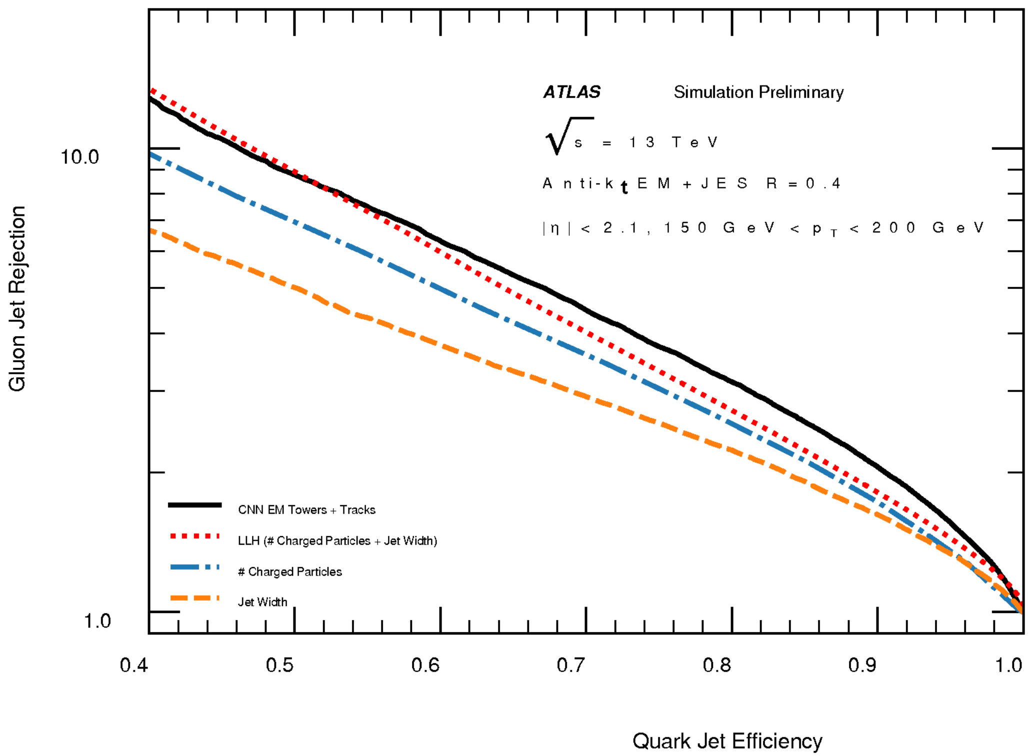


Jet Images on ATLAS: Quarks vs Gluons

ATL-PHYS-PUB-2017-017



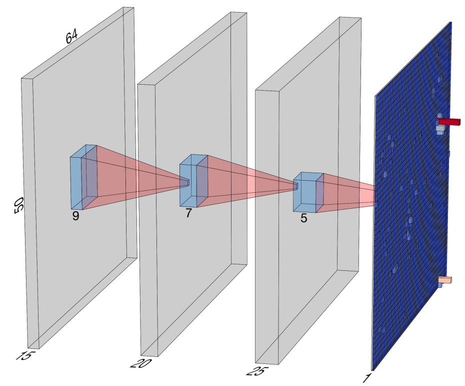


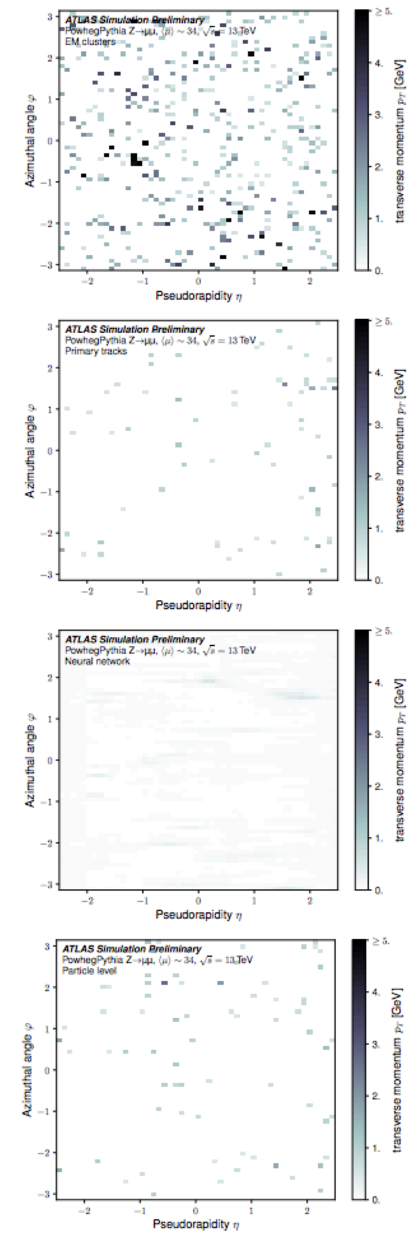


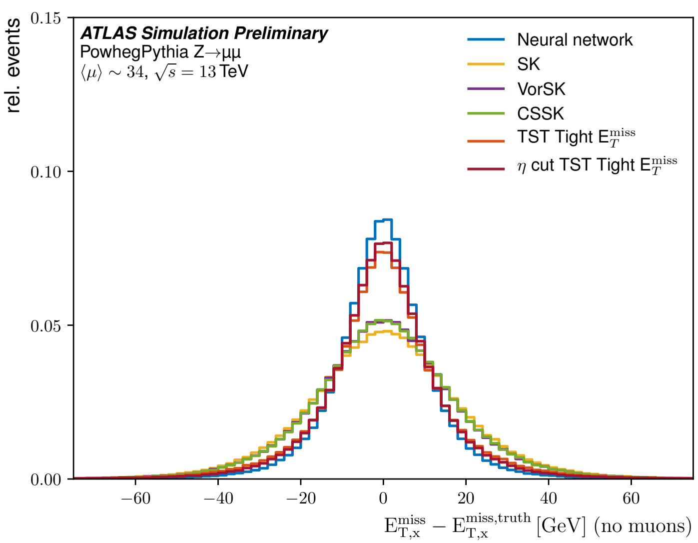
Note: Other experimental results on CMS with ImageTop, a boosted top tagger based on images CMS-PAS-JME-18-002

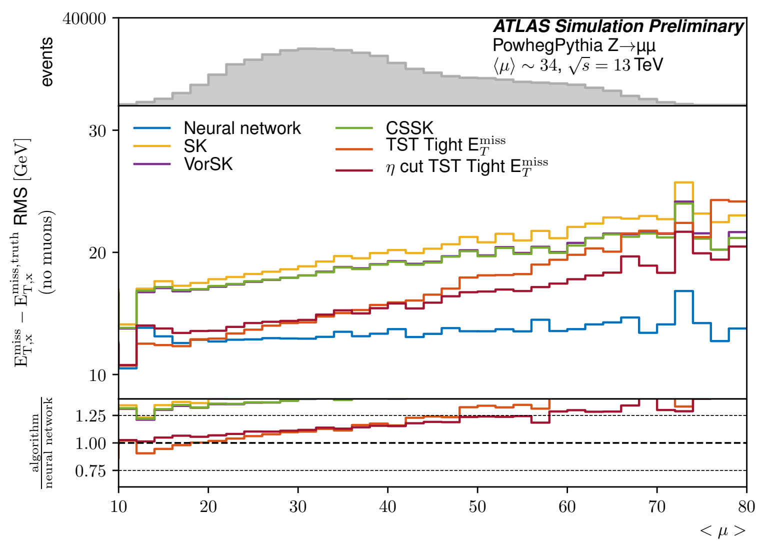
Calo-Images for Missing Energy Pileup Removal

- *Input*: images of calorimeter clusters and tracks
- *Output*: NN regress to predict hard scatter energy in each calorimeter tower
- Gains in resolution
  - NNN doesn't learn accurate cell by cell predictions
  - Considering new ways to define loss



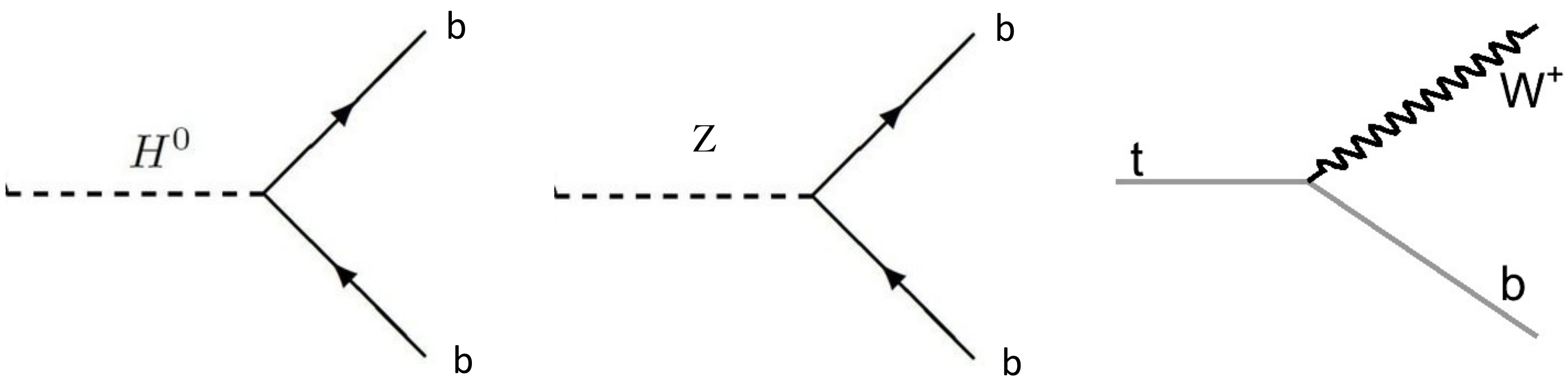




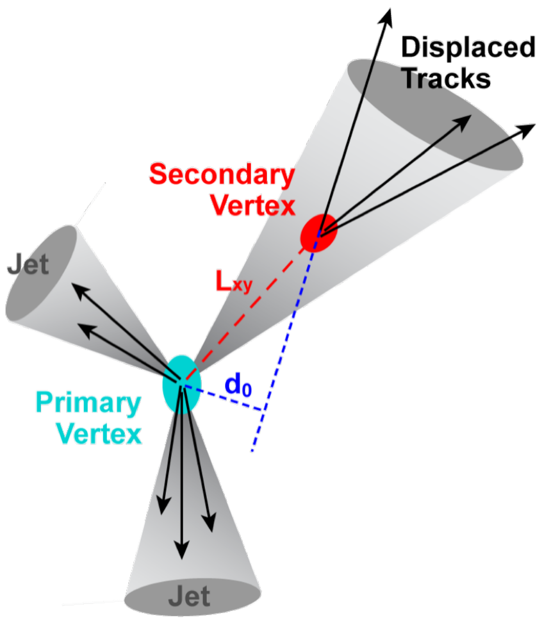


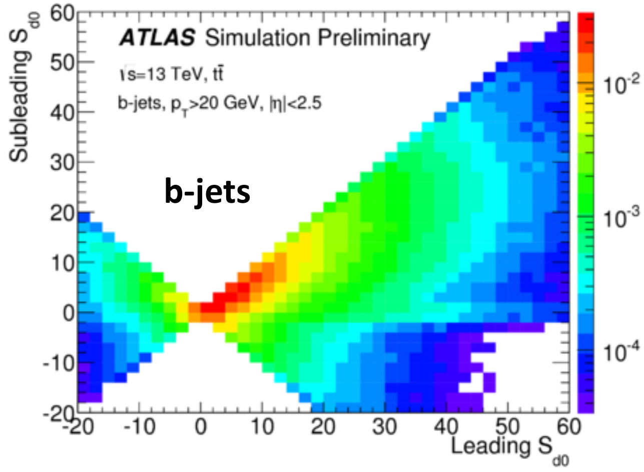
ATL-PHYS-PUB-2019-028

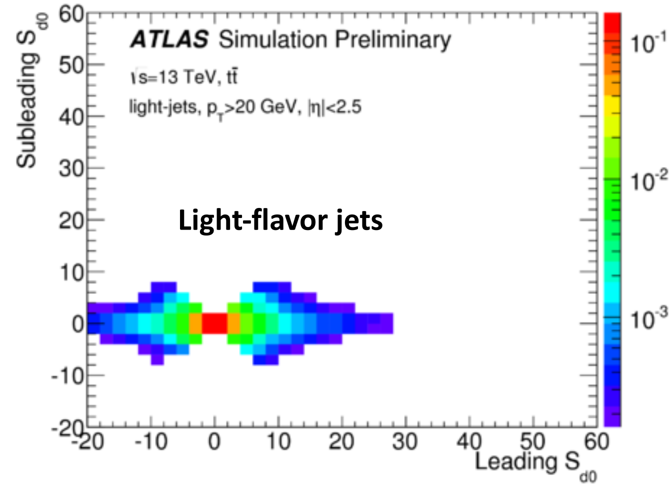
Reconstructing Bottom Quark Jets



Bottom Quark Jet Identification

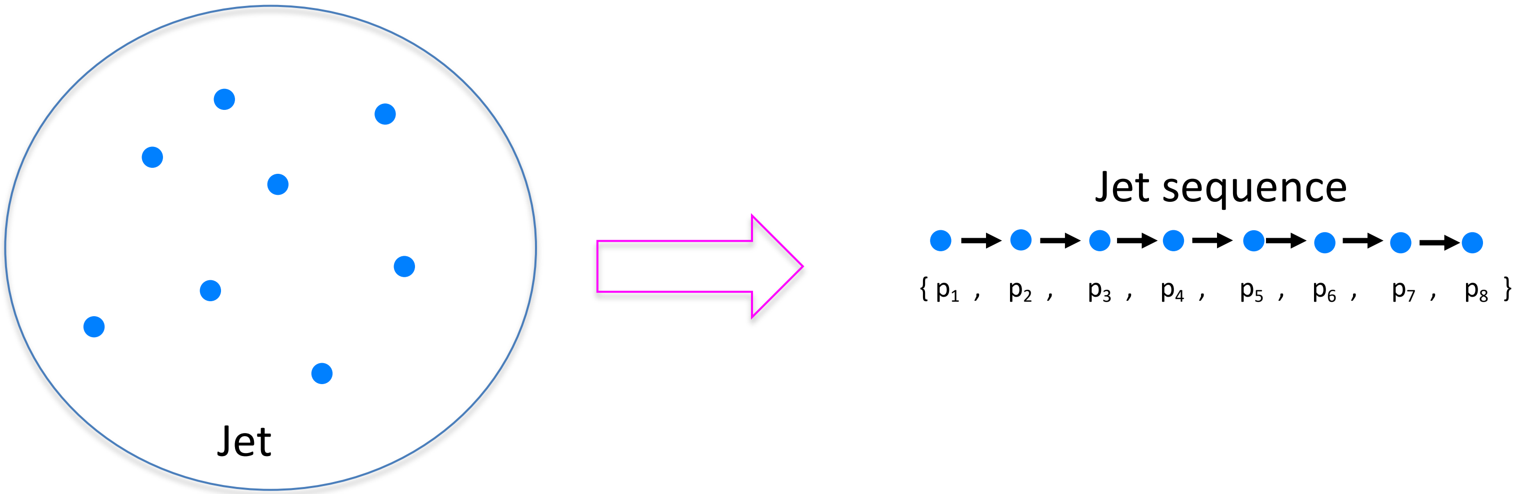






- Goal: Discriminate b-jets from non-b-jets
- **Track Impact Parameter based** taggers:
  $p(jet\ flavor \mid tracks\ in\ jet)$
  – Dimensionality too high for histogram density estimation
  – Often make naïve Bayes assumption that tracks independent!

Jets as Sequences



- Jets are a grouping of a variable number of particles

- With physically motivated ordering: **jet as a sequence**

Recurrent Neural Networks

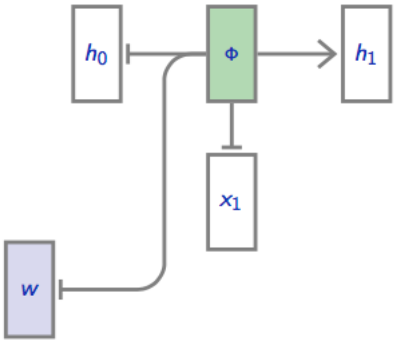


Image credit: [F. Fleuret]

Recurrent Neural Networks

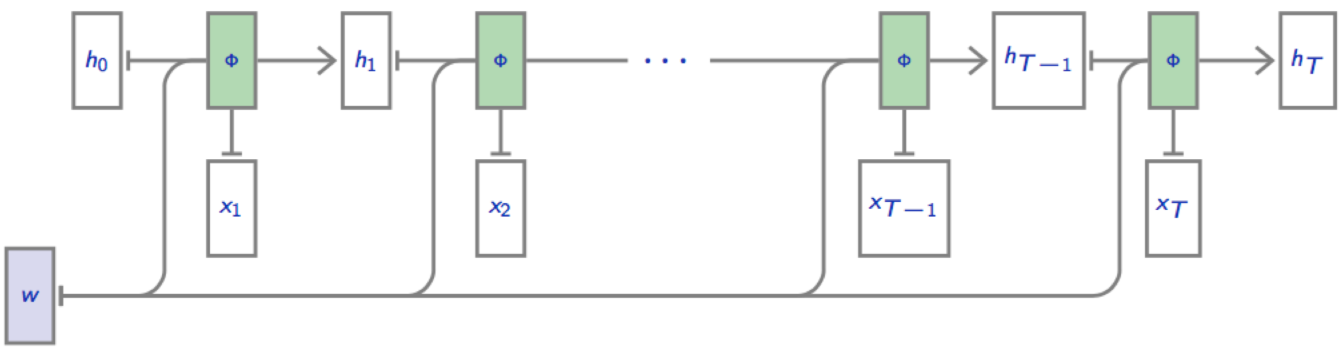


Image credit: F. Fleuret

Recurrent Neural Networks

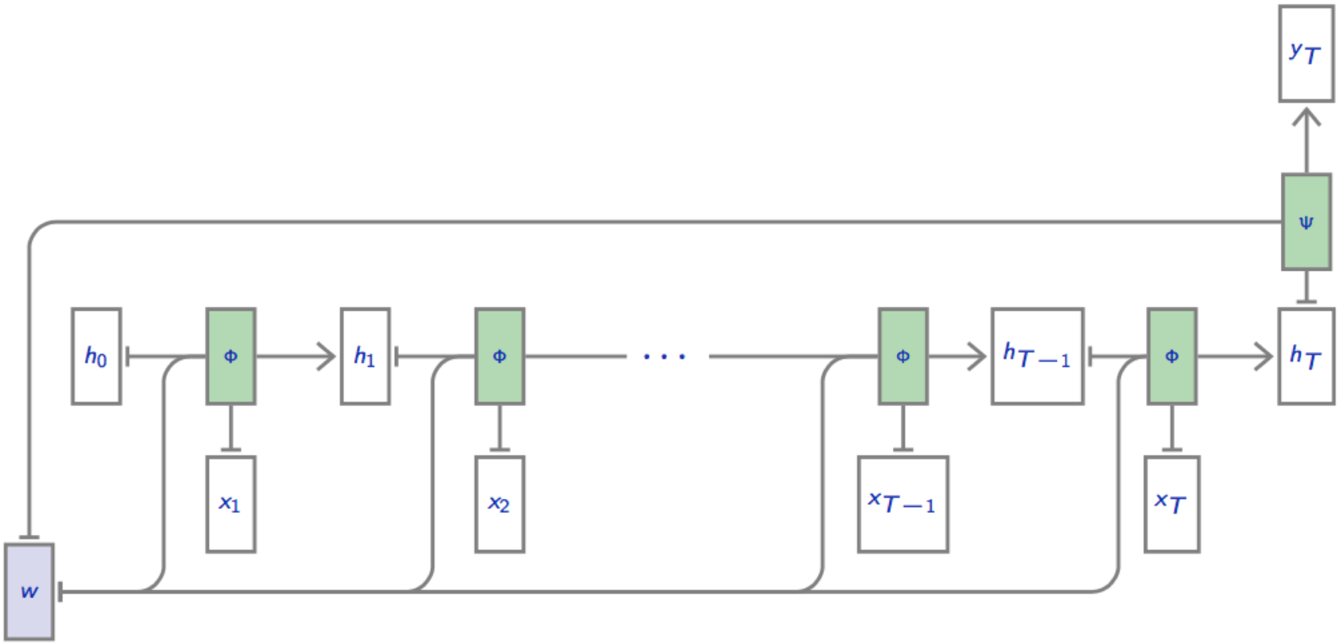
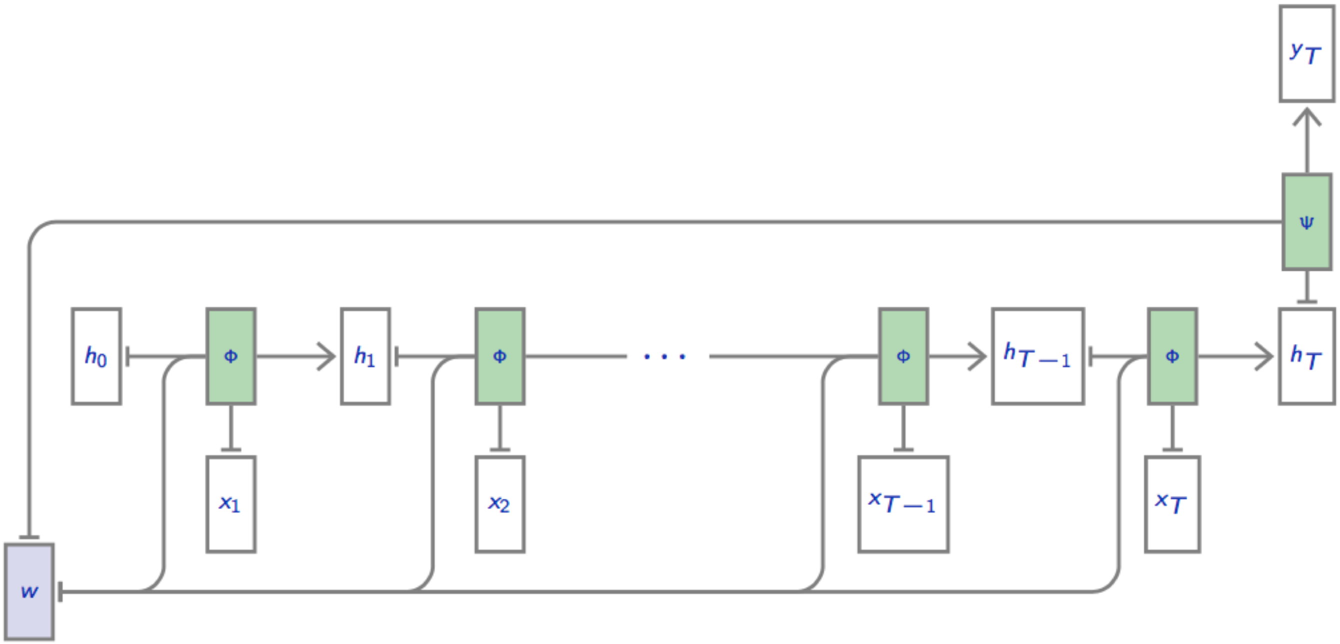


Image credit: [F. Fleuret]

Recurrent Neural Networks

[0.98] → Positive Sentiment



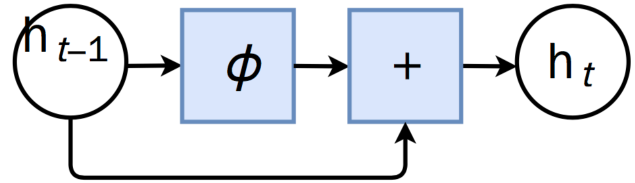
Sentiment Analysis

The movie was great

Image credit: [F. Fleuret]

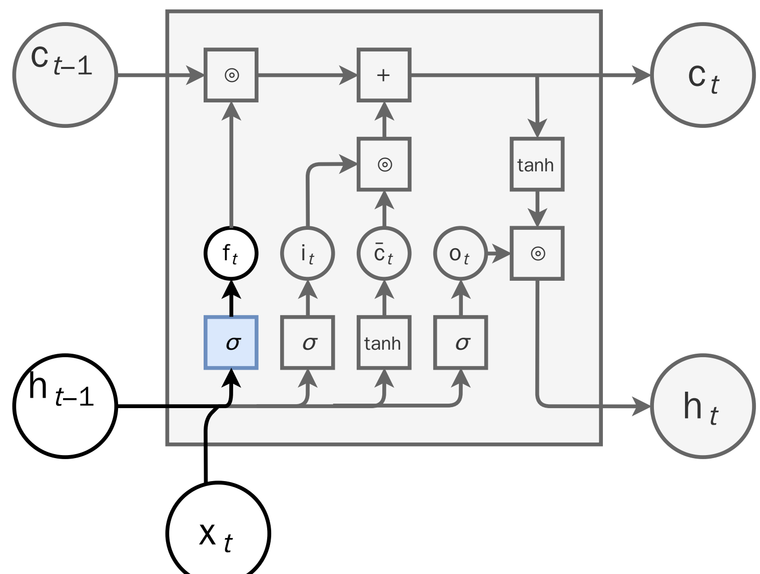
Long Short Term Memory (LSTM)

- Gating:



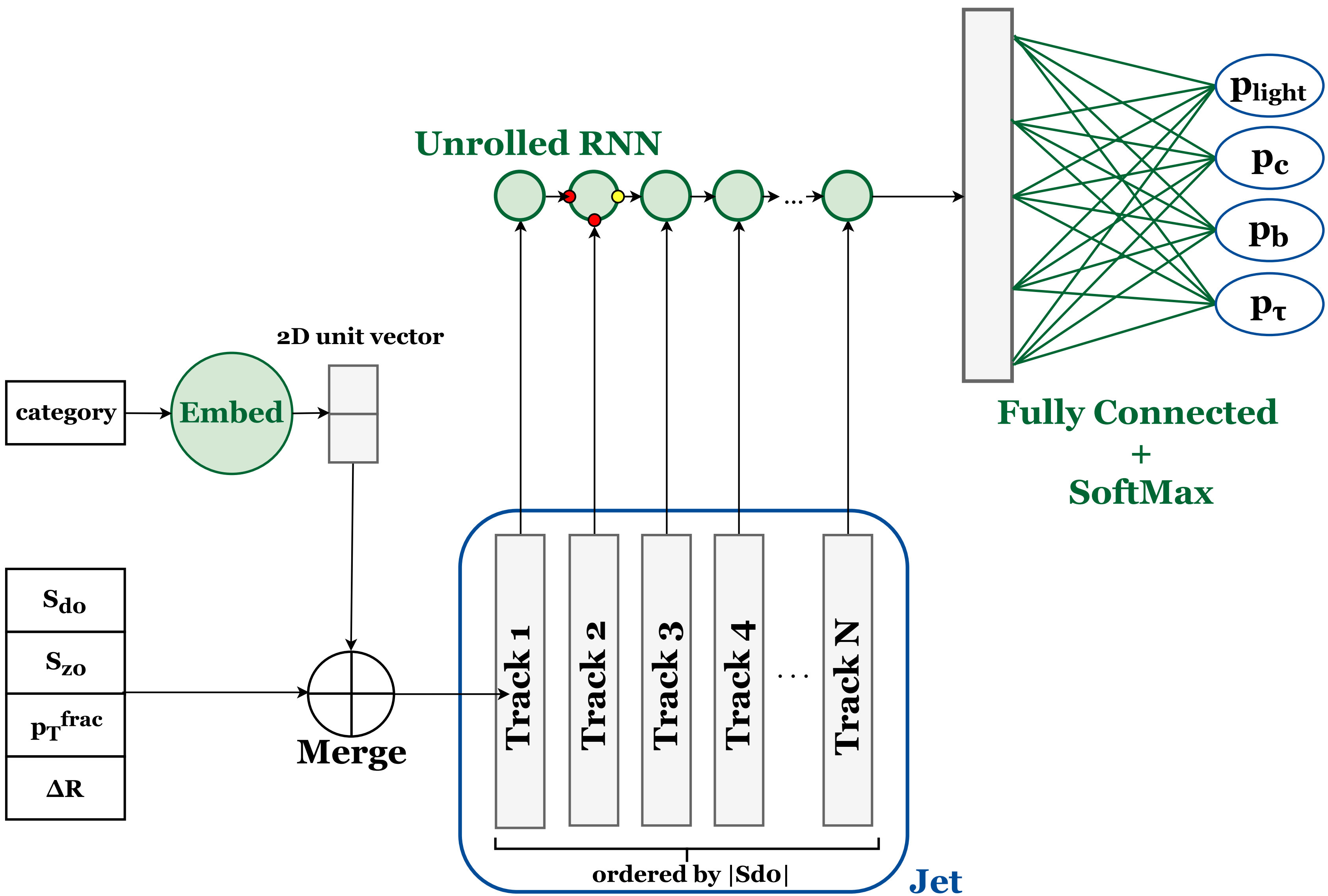
  - network can grow very deep, in time → vanishing gradients.
  - *Critical component*: add pass-through (additive paths) so recurrent state does not go repeatedly through squashing non-linearity.

- LSTM:
  - Add internal state separate from output state
  - Add input, output, and forget gating



Credit: Gilles Louppe

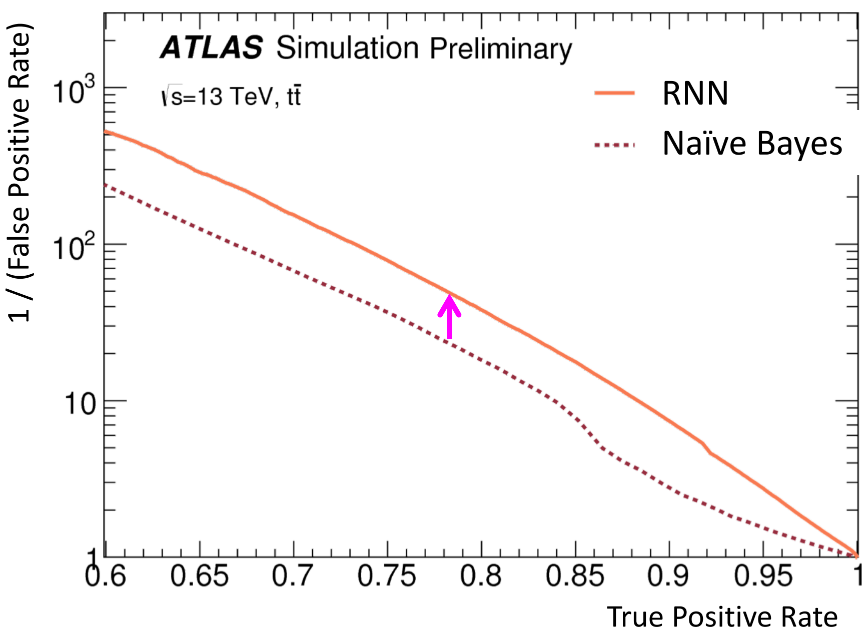
Jets and Sequence Processing

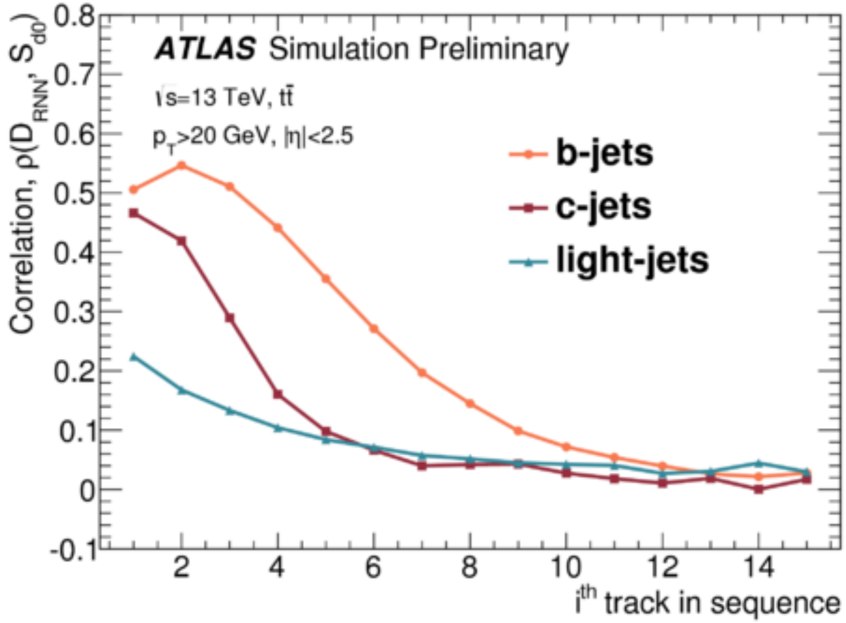


ATL-PHYS-PUB-2017-003

RNN b-tagging

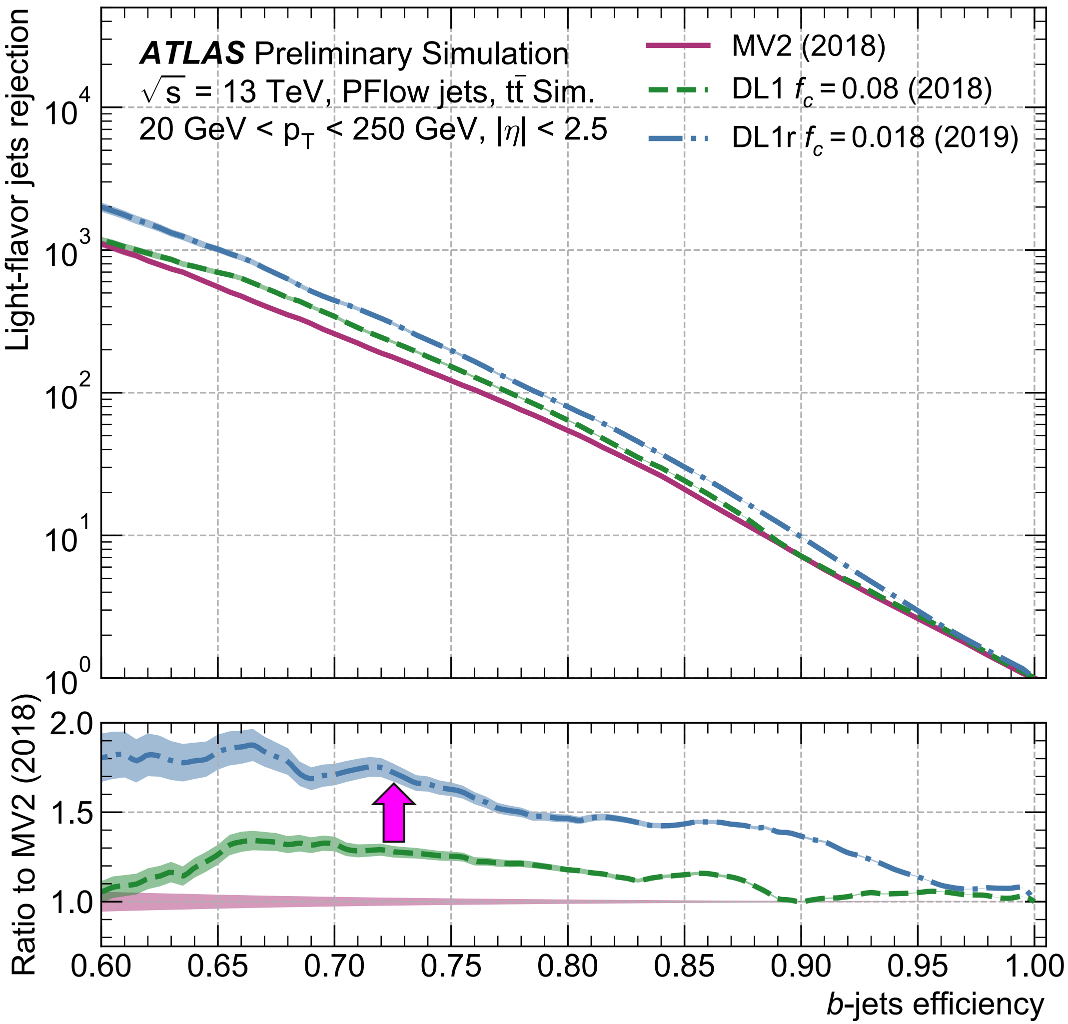
- Order tracks by impact parameter
- RNN can learn inter-track dependencies

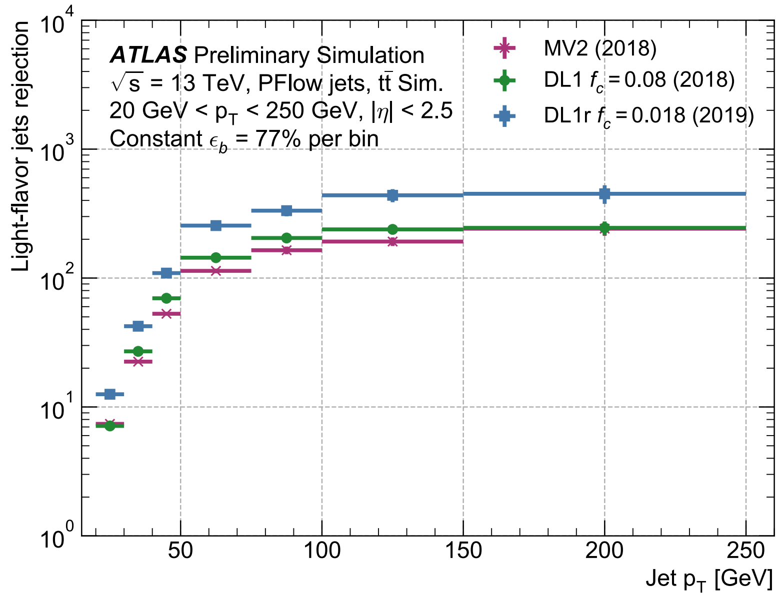


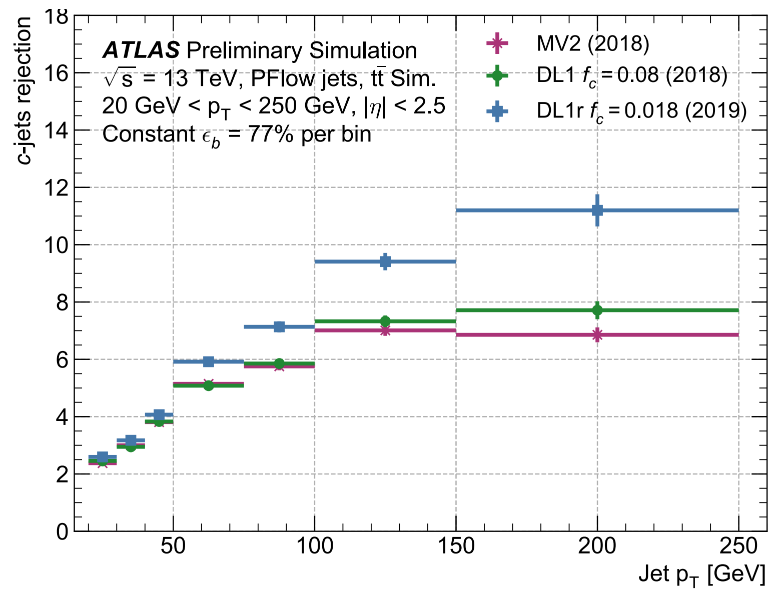


ATL-PHYS-PUB-2017-003

Combining With Other Algorithms



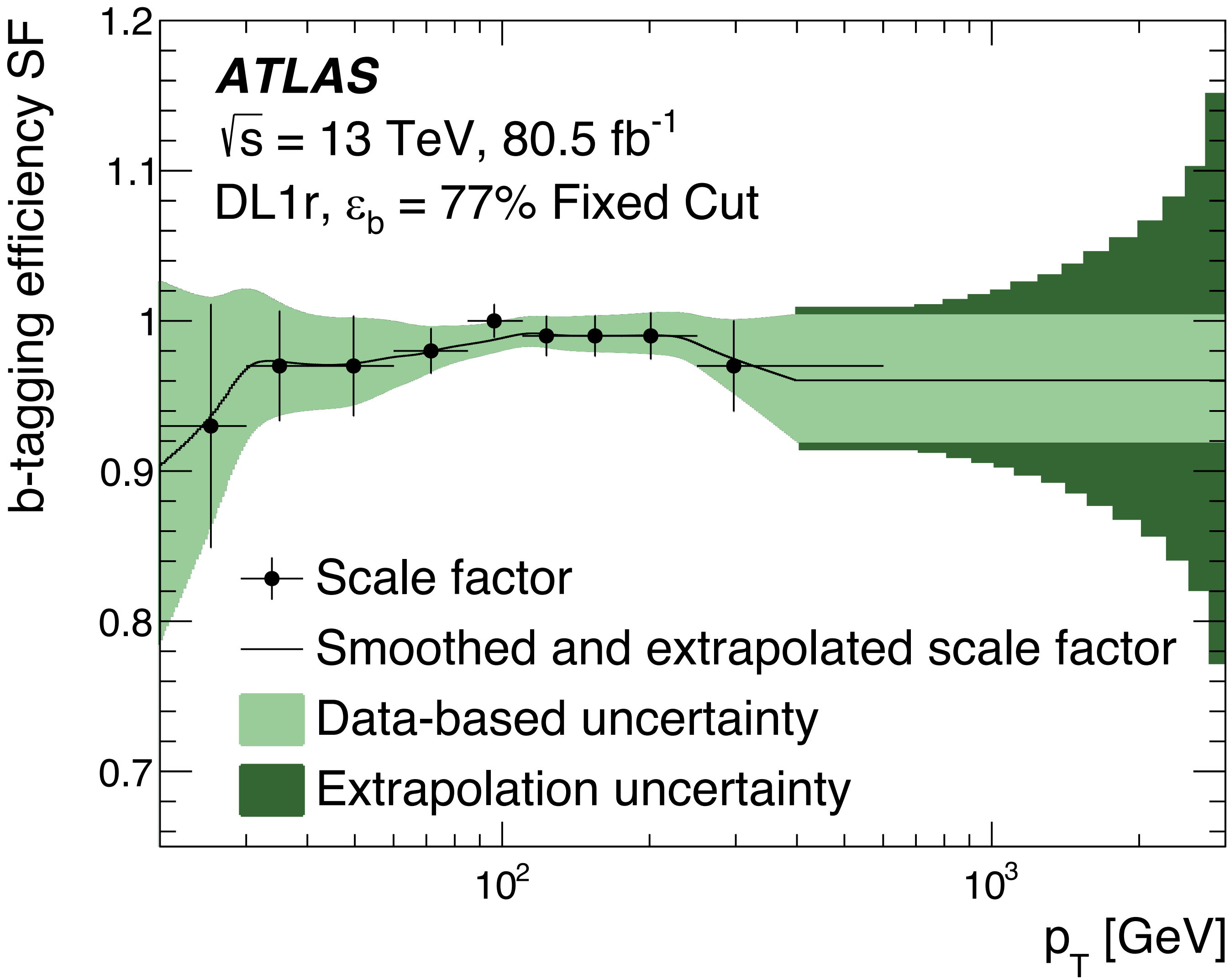




ATLAS-PHY-PLOTS-FTAG-2019-005

1/(false-positive rate) at fixed true-positive rate vs jet p_T

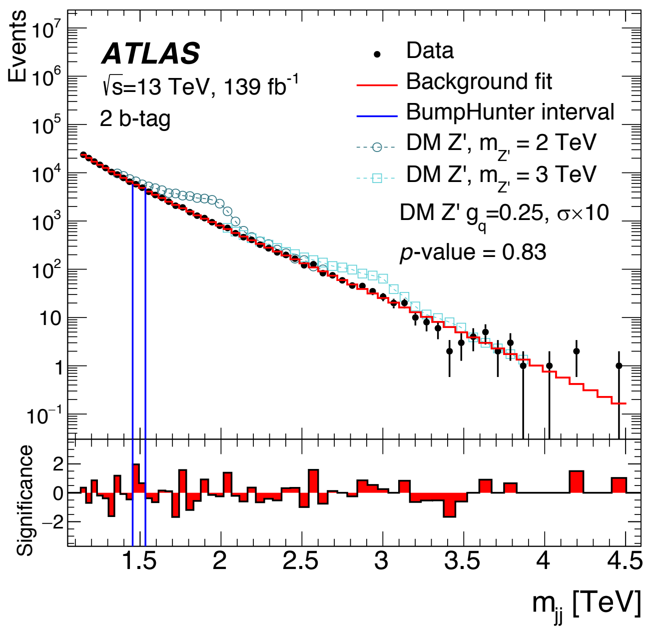
Calibration

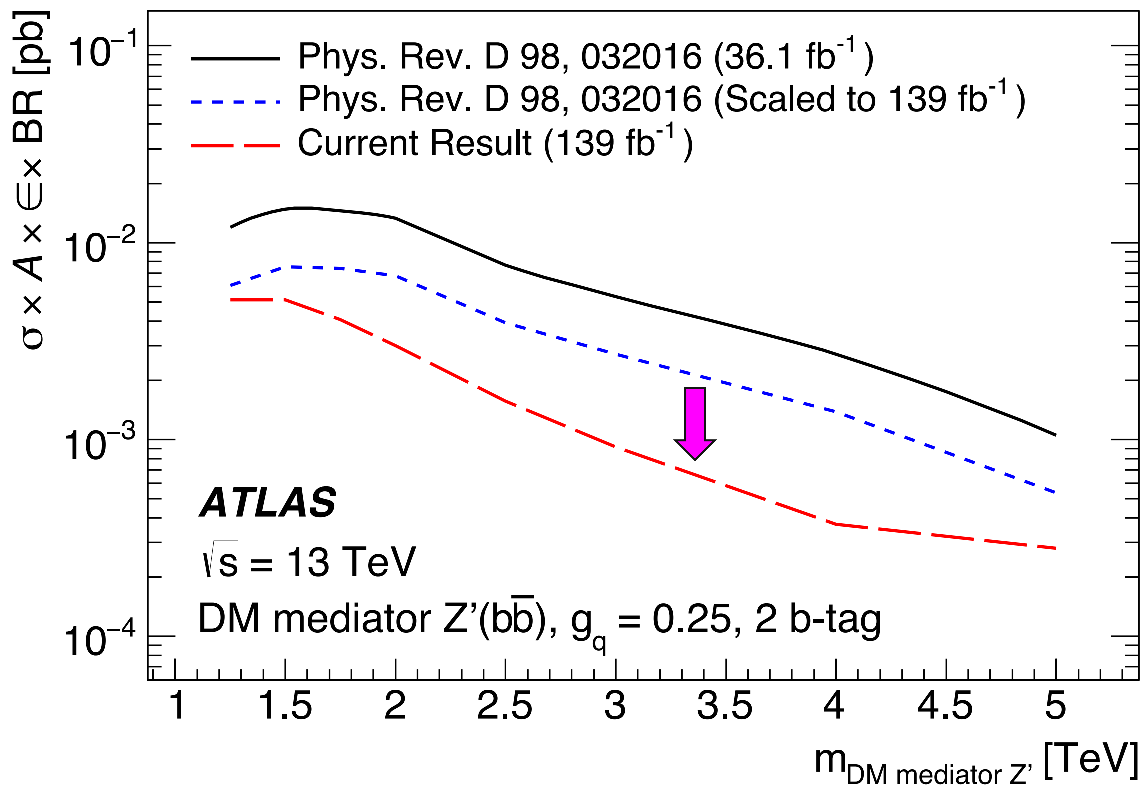


*This is actually a slightly older version of the algorithm than previous slide

arXiv:1910.08447

Real World Impact: Dijet Resonances



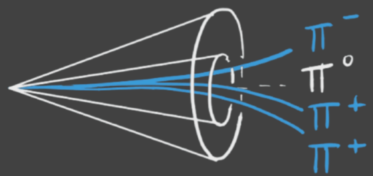


arXiv:1910.08447

Tau RNN

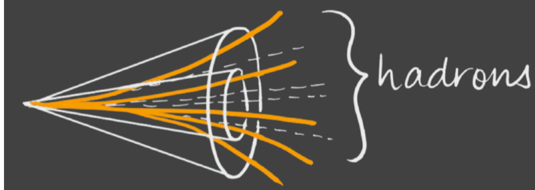
[Images from Mariel Pettee]

Hadronic Tau Decay:

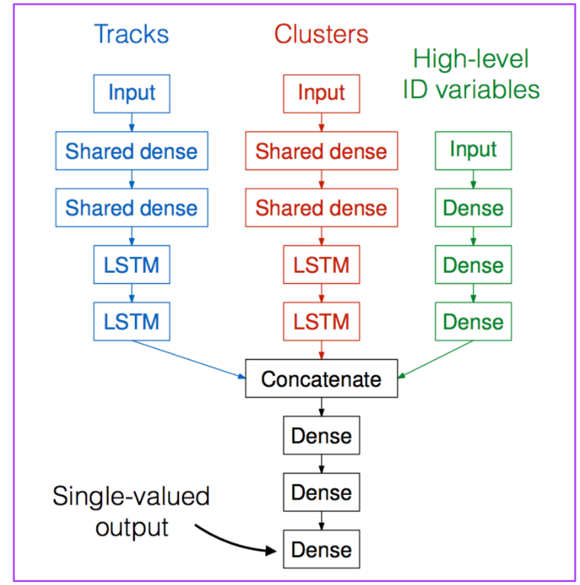


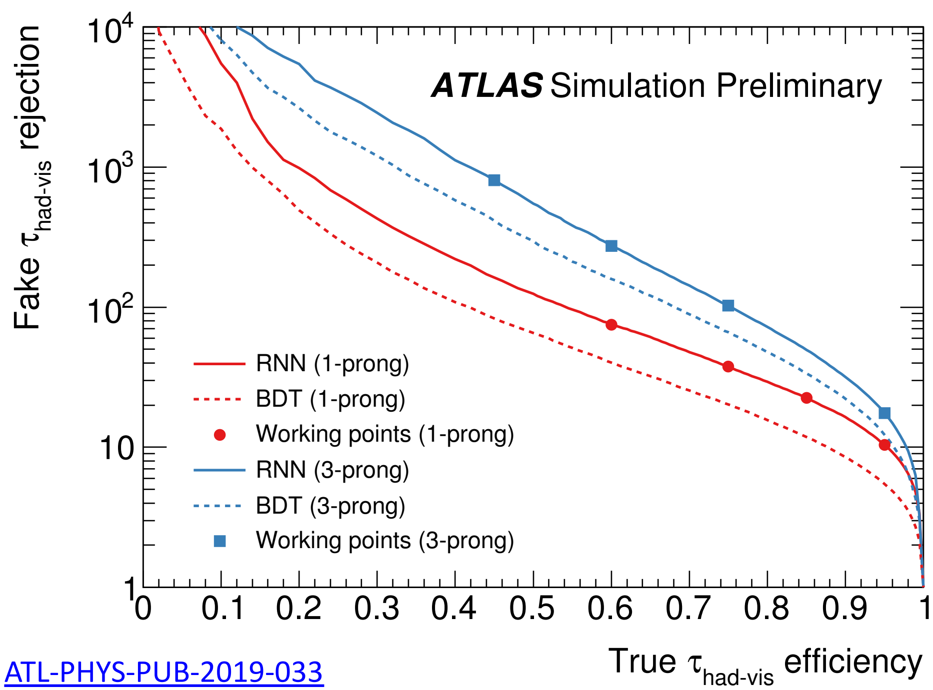
- Typically 1 or 3 tracks
- Highly collimated
- No surrounding tracks

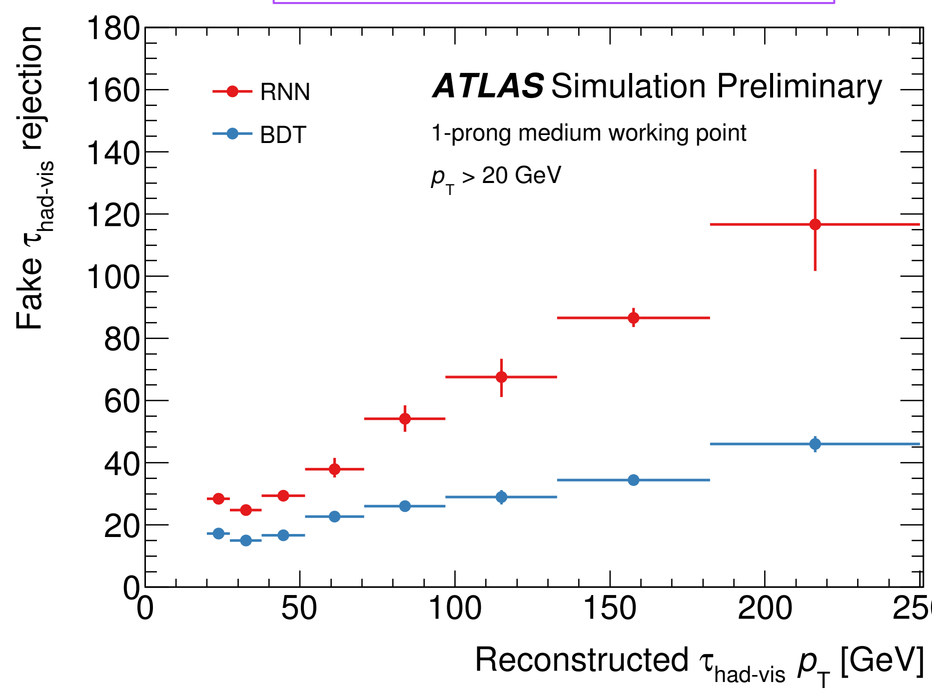
Jet Background:



- Many tracks
- Less collimated
- Tracks in isolation region





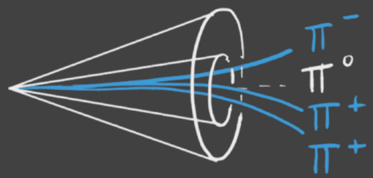


ATL-PHYS-PUB-2019-033

Tau RNN

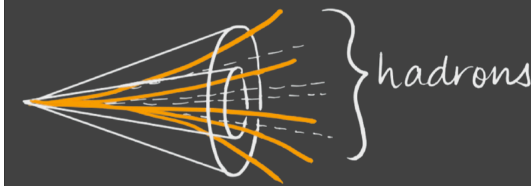
[Images from Mariel Pettee]

Hadronic Tau Decay:

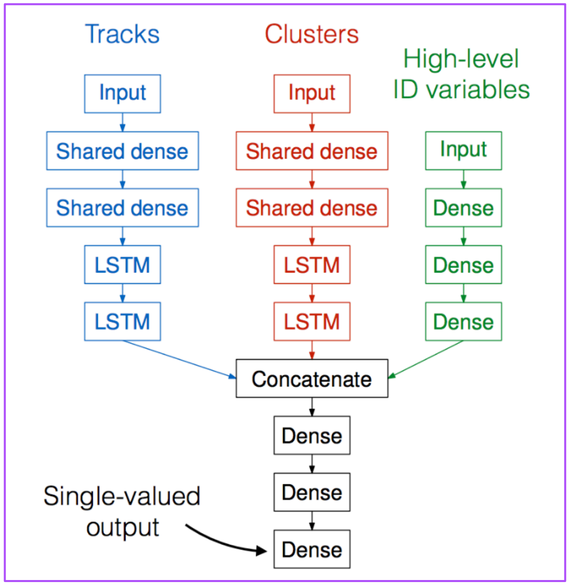


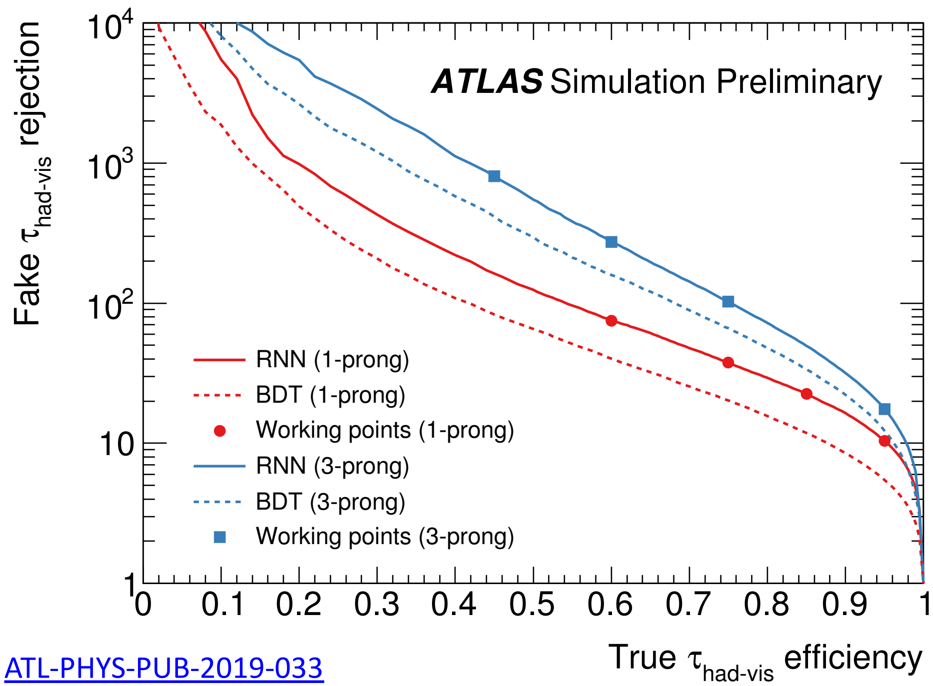
- Typically 1 or 3 tracks
- Highly collimated
- No surrounding tracks

Jet Background:

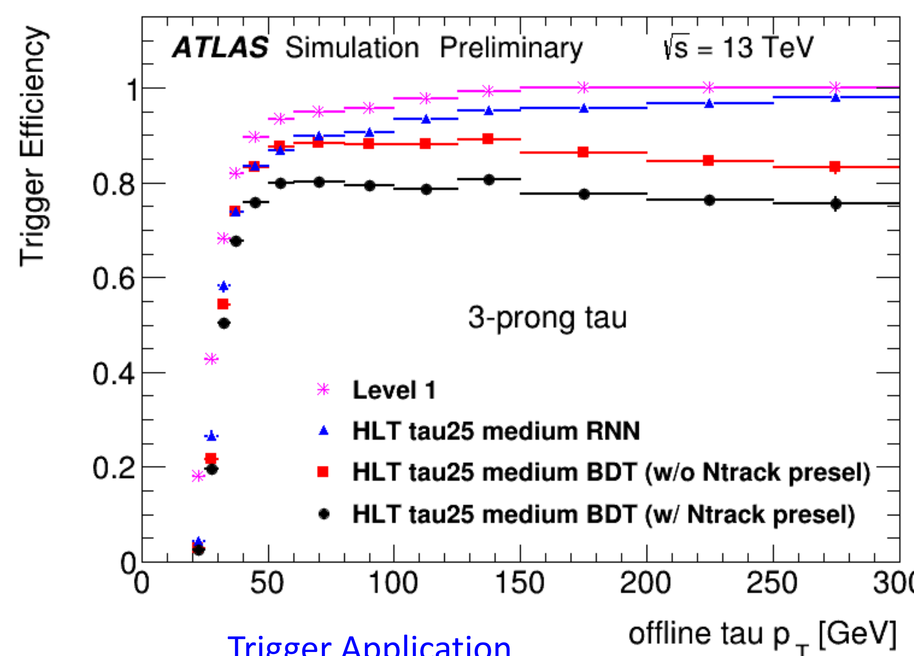


- Many tracks
- Less collimated
- Tracks in isolation region





ATL-PHYS-PUB-2019-033



Trigger Application

Density Estimation and Generative Modeling

Why Generative Modeling

- Discriminative models: $f(x) \approx \bar{y} = E_{p(y|x)}[y]$

- How do we model uncertainty on predictions, i.e. learn a posterior on likelihood?

- Generative models aim to estimate density p(x) or conditional p(x | y)
  - Explicitly: can compute the value $p(\cdot)$
  - Implicitly: can draw samples from $p(\cdot)$

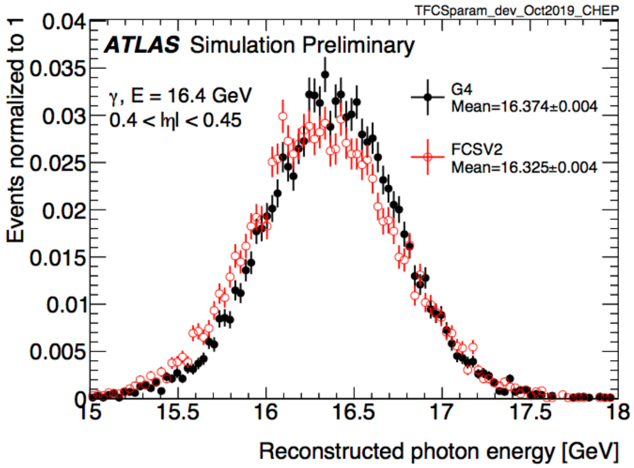
  - More on Generative Models in A. Butter's Talk

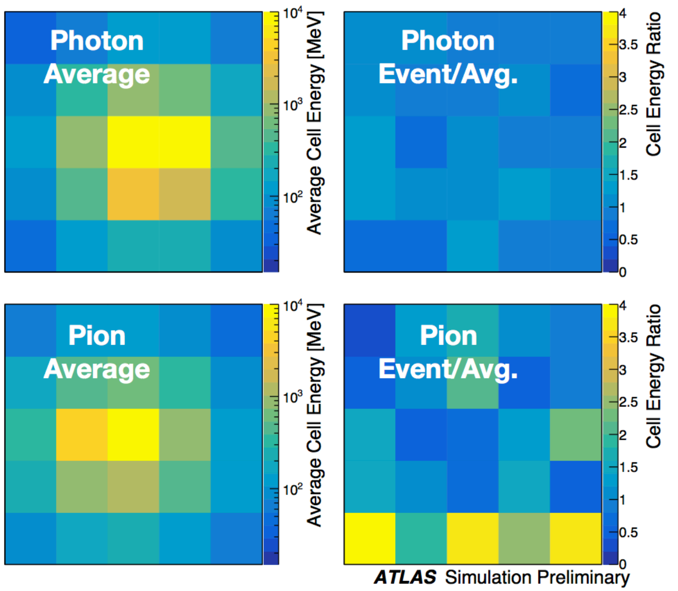
Fast Simulation

- Increased pileup at HL-LHC will push boundaries of our computational capabilities for simulation

- Full Simulation
  - Accurate but costly to sample

- Fast Simulation
  - Sample from parametric average shower model
  - Doesn't account for correlations in shower shape fluctuations





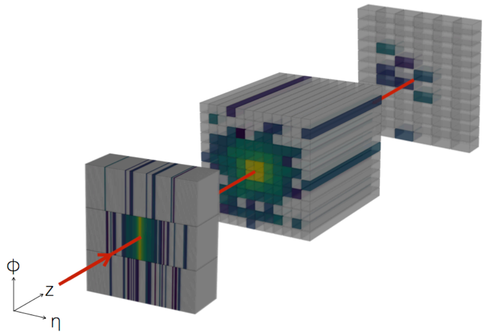
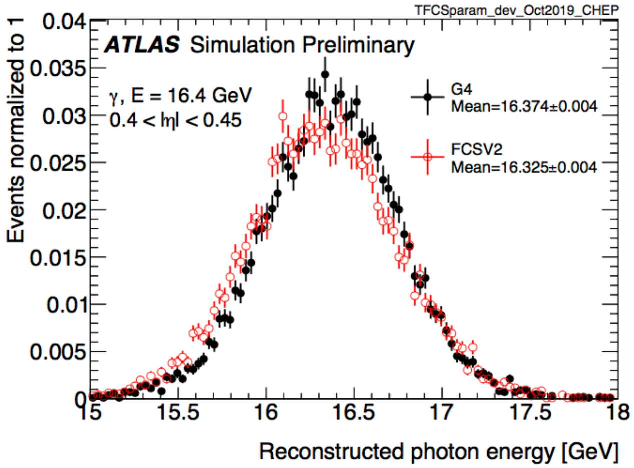


Image: arXiv:1701.05927

Fast Simulation

- Increased pileup at HL-LHC will push boundaries of our computational capabilities for simulation



- Full Simulation
  - Accurate but costly to sample
- Fast Simulation
  - Sample from parametric average shower model

- New approaches:
  - *Correct average:* Model correlated fluctuations on top of average
  - *Full ML approach*: Learn generative mode of distribution of showers, p(x), and produce samples

Correcting the Average

- Method: Fit parametric approximation to correlated noise distribution

- *Sklar's Theorem*: given a random vector (X_1 , … , X_n), the joint cumulative distribution function

$$H(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_1 \leq x_1)$$

can be expressed using marginals $F_i(x_i) = P(X_1 \leq x_1)$ as

$$H(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n))$$

where C(·) is the copula

Correcting the Average

- Method: Fit parametric approximation to correlated noise distribution with a *Gaussian Copula*

Gaussian Copula

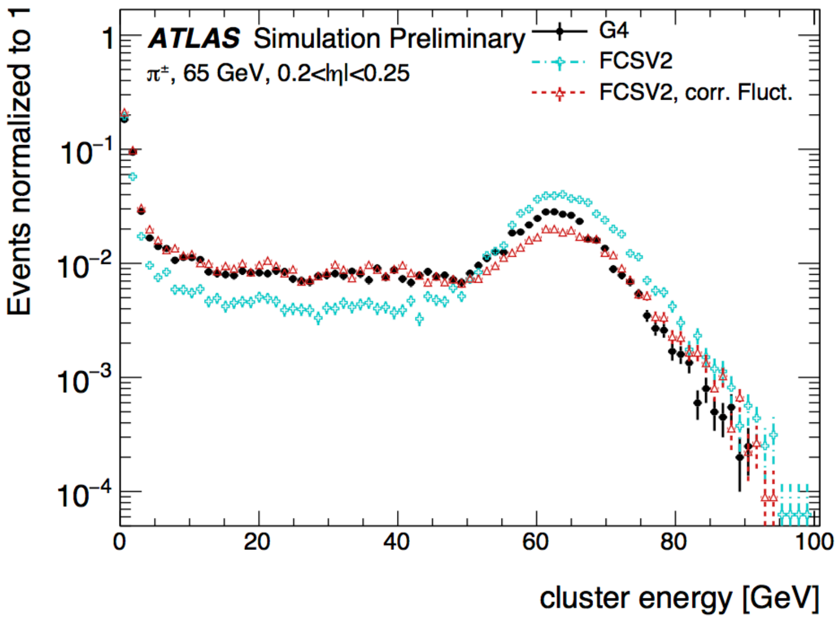
$$C_R^{\text{Gauss}}(u) = \Phi_R\left(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)\right)$$

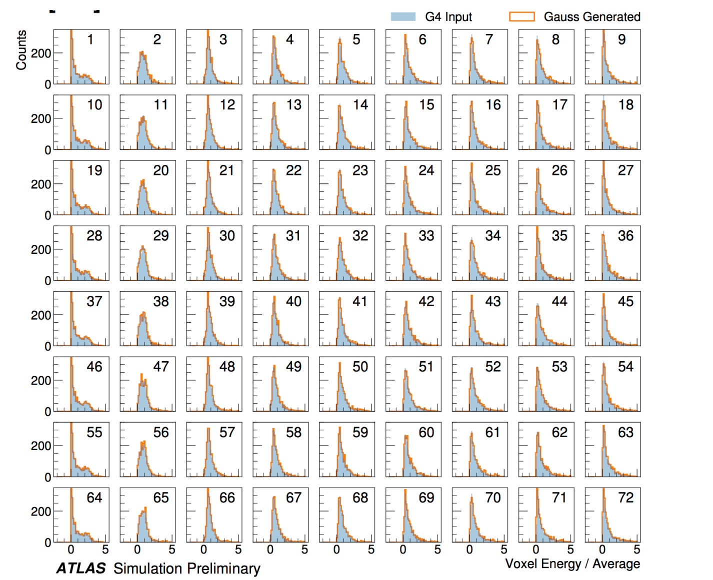
Gaussian Copula density

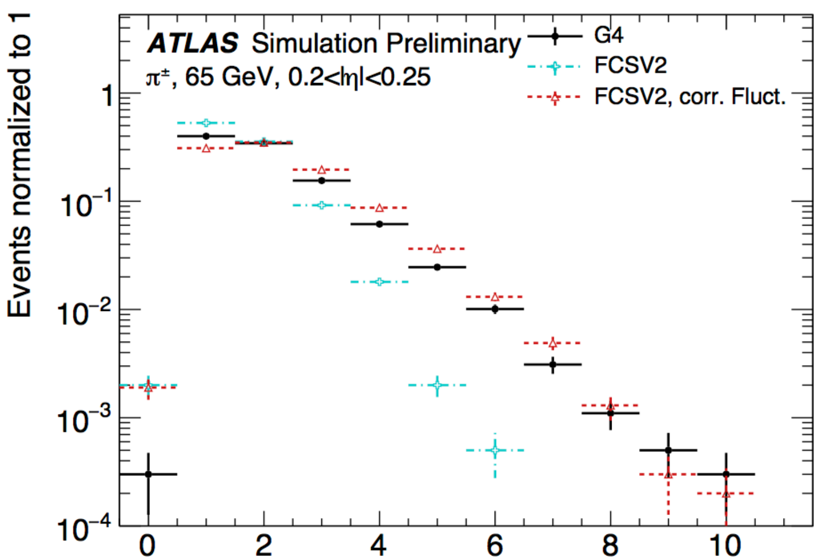
$$c_R^{\text{Gauss}}(u) = \frac{1}{\sqrt{\det R}} \exp\left(-\frac{1}{2}\begin{pmatrix}\Phi^{-1}(u_1)\\ \vdots \\ \Phi^{-1}(u_d)\end{pmatrix}^T \cdot \left(R^{-1} - I\right) \cdot \begin{pmatrix}\Phi^{-1}(u_1)\\ \vdots \\ \Phi^{-1}(u_d)\end{pmatrix}\right)$$

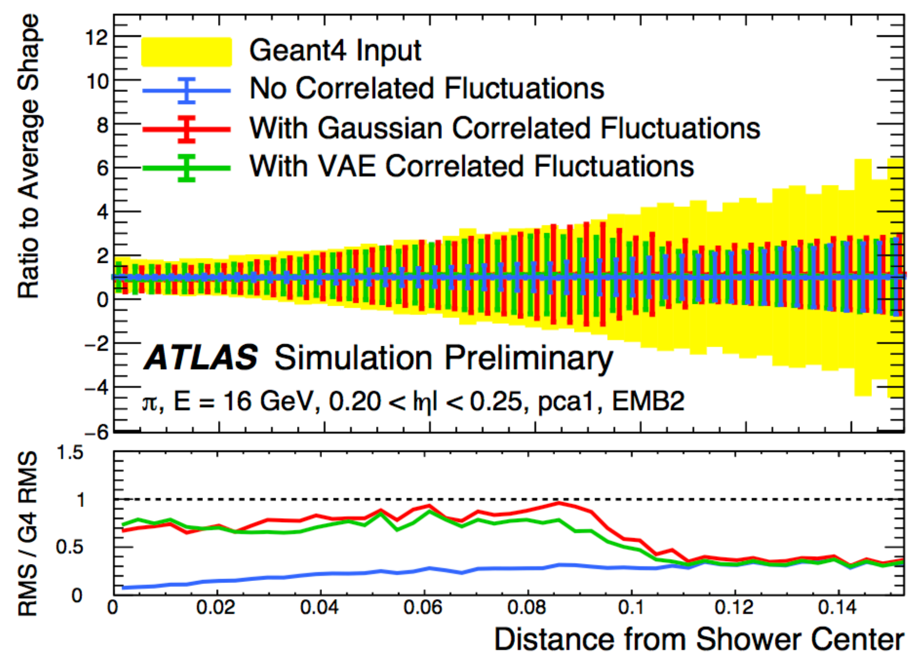
1. CDF transform inputs x_i to uniform u_i
2. Fit copula to sample of correlated uniform variables
3. Sample Copula to get u_i and invert CDF to get x_i

Correcting the Average with Copula





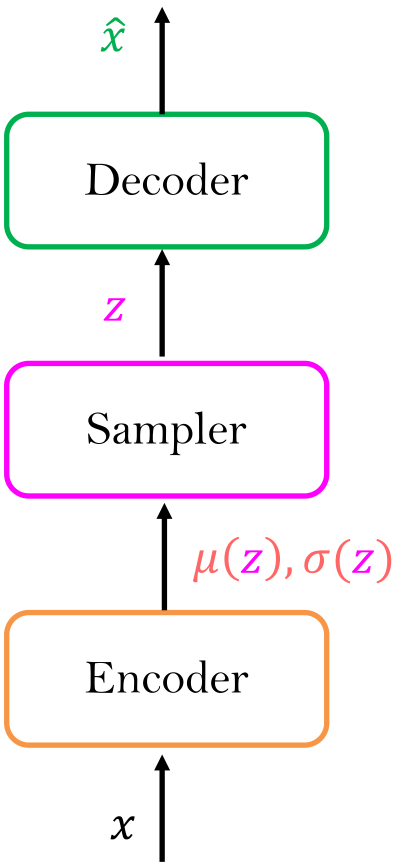




ATLAS-PHYS-PLOTS-SIM-2019-009

Deep Generative Models

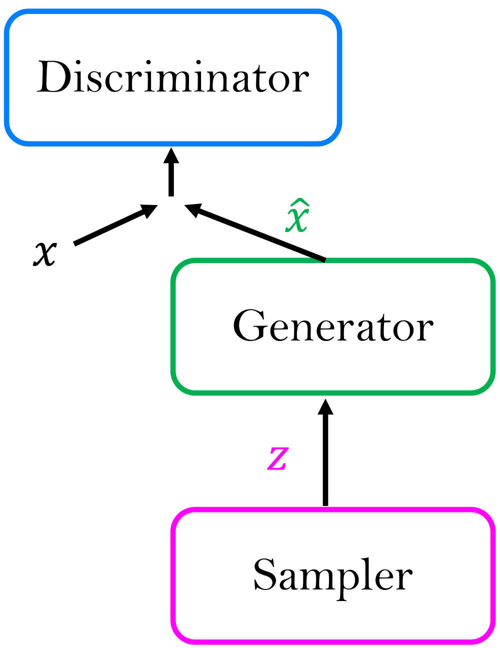
Variational AutoEncoder



$$\mathcal{L}(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}[q_\phi(z|x) \parallel \mathrm{p(z)}]$$

arXiv:1312.6114
arXiv:1401.4082

Generative Adversarial Network

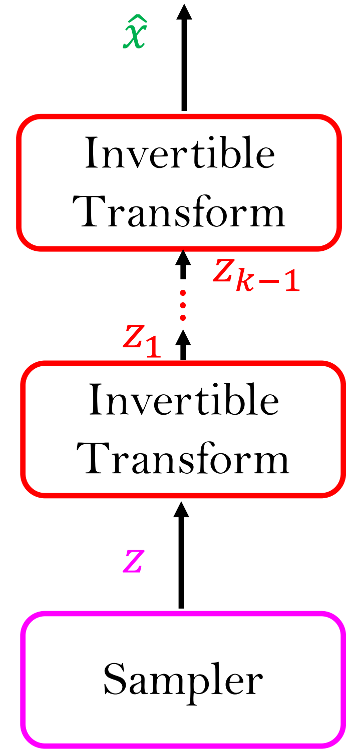


$$\min_G \max_D V(G, D)$$

$$V(G, D) = \mathbb{E}_{p_{data}(x)}[\log D(x)] + \mathbb{E}_{p_z(z)}[\log(1 - D(G(z)))]$$

arXiv:1406.2661

Normalizing Flow



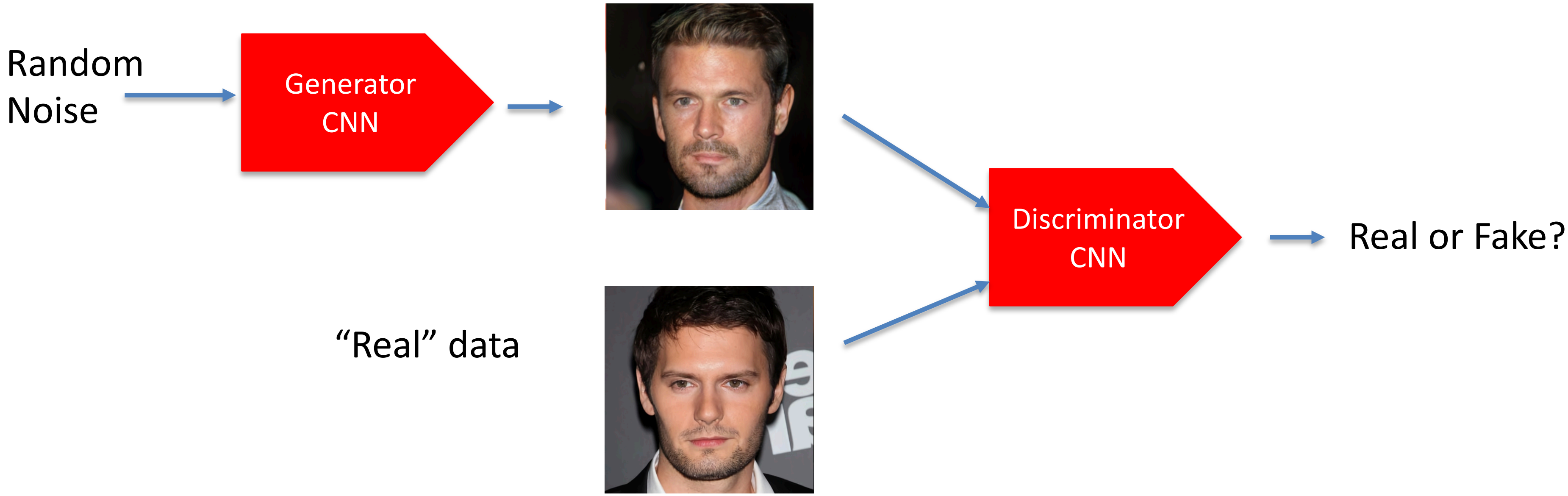
$$z_1 = f_1(z) \quad \rightarrow \quad p(z_1) = p(z) \left| det \frac{df_1}{dz} \right|^{-1}$$

$$x = f_k \circ f_{k-1} \circ \cdots \circ f_1 \, (z)$$

$$\log p(x) = \log p(z) - \sum_{i=1}^{k} \log \left| det \frac{df_i}{dz_i} \right|$$

arXiv:1410.8516
arXiv:1505.05770

Generative Adversarial Networks (GAN) [arXiv:1406.2661]



- Generator produces images from random noise and tries to trick discriminator into thinking they are real

- Classifier tries to tell the difference between real and fake images

Images: arXiv:1710.10196

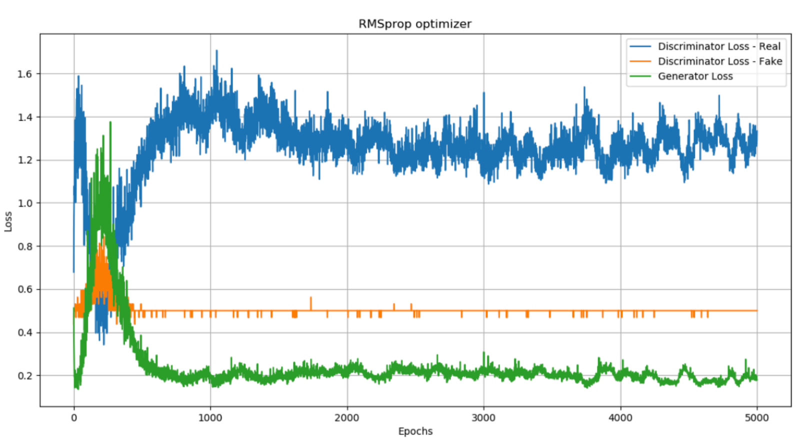
GAN Loss and Training

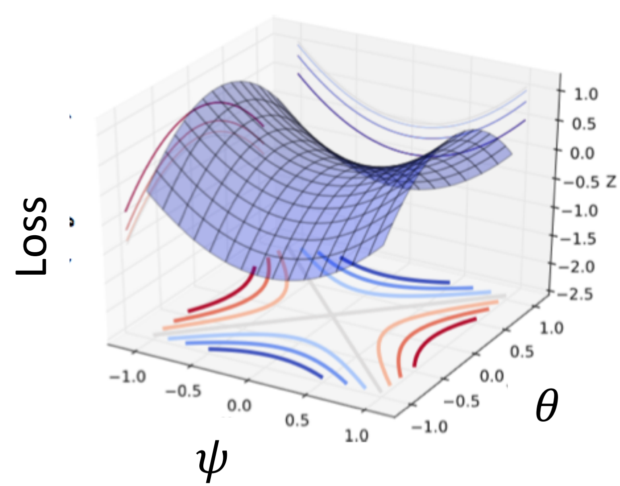
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z))]$$

- Two-player minimax game between Generator (G) and Discriminator (D) networks

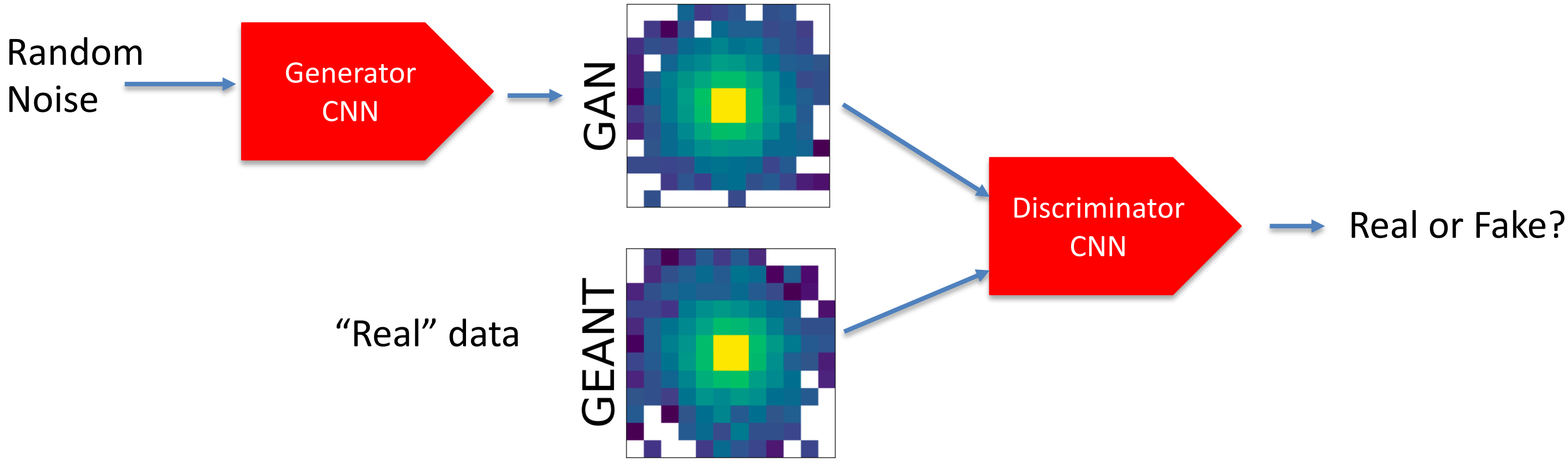
- Training involves carful and often unstable iteration between updating G parameters (θ) and D parameters (ψ)

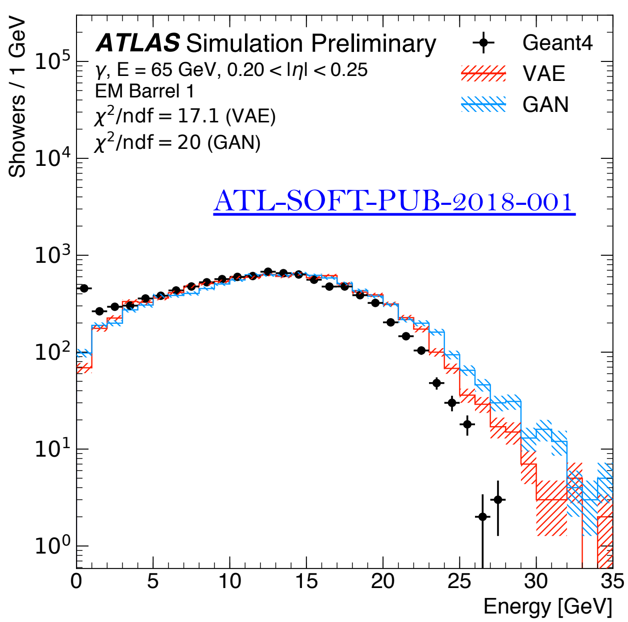
- If perfectly trained, generator converges to implicit model of data density: $\mathbf{G(z) = x \sim p_{data}(x)}$

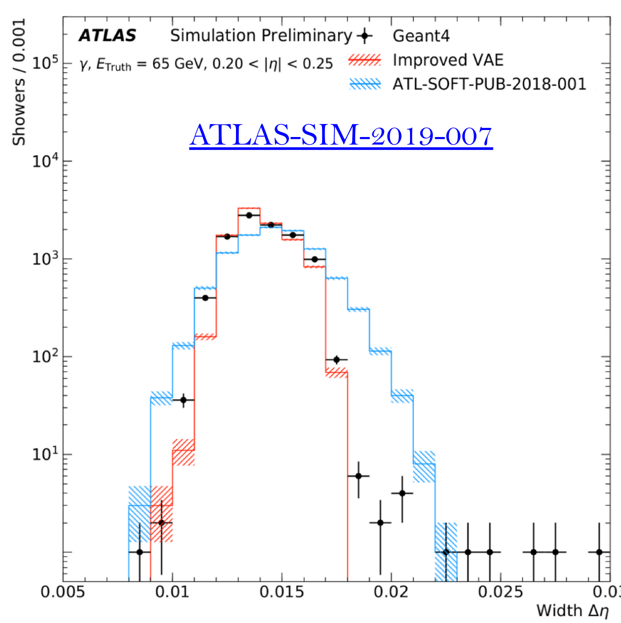


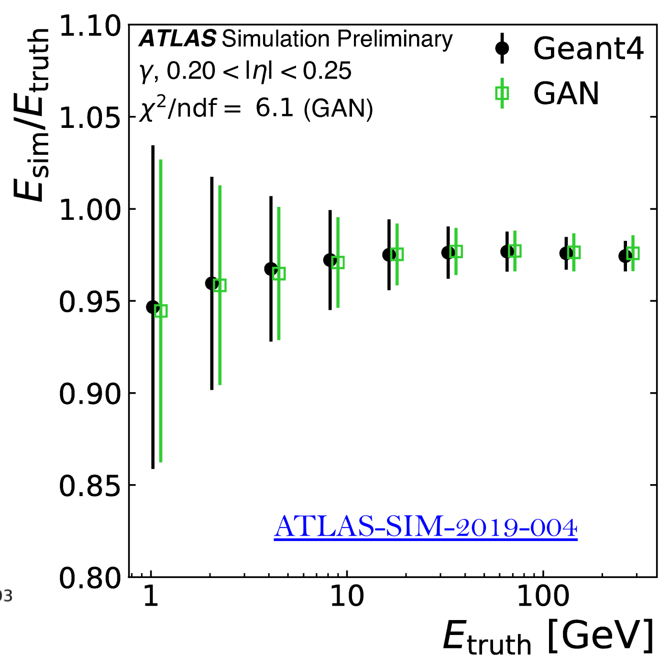


GANs for Calorimeter Energy Depositions









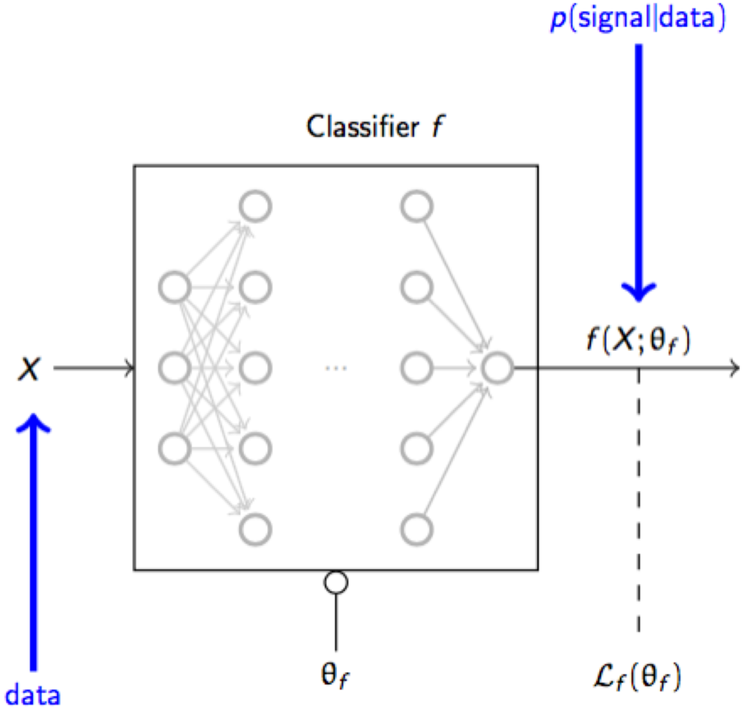
Making use of Generative Modeling Tools

Adversarial Learning for Enforcing Invariance

- With flexibility come complexity:
  - Hard to control how models learn / utilize information
  - Potentially unwanted sensitivity to poorly modeled aspects of simulation
  - Potentially unwanted sculpting of key physics distributions like mass

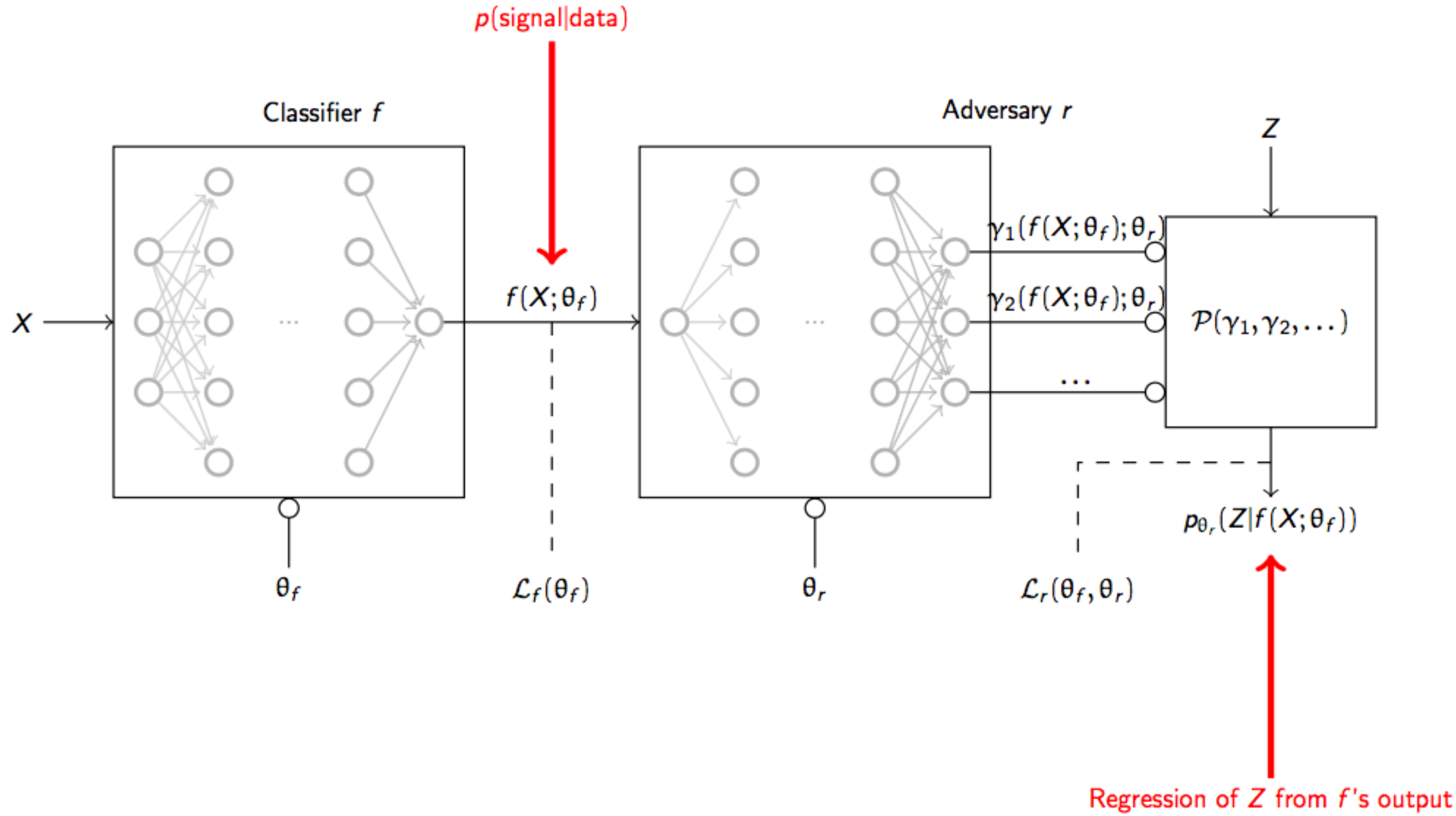
- ***Idea***: Augment training of classifier to enforce invariance to changes in a variable Z (nuisance parameter for systematic uncertainty, kinematic variables, etc.)
  - Several ways to do this, see D. Shih's Talk

Adversarial Networks



- Classifier built to solve problem at hand

Adversarial Networks

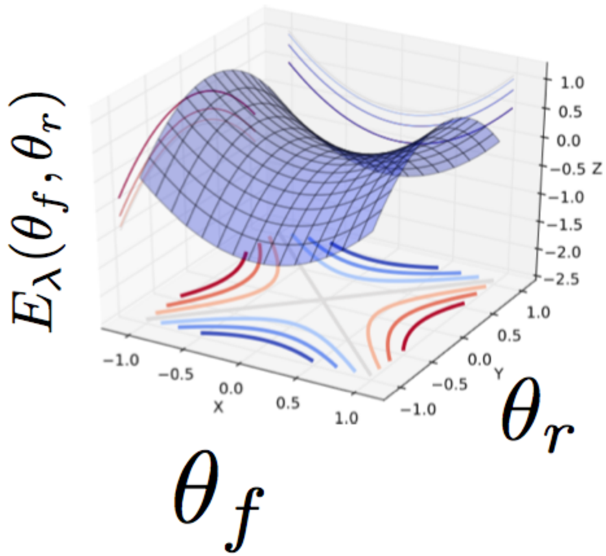


- Loss that encodes performance of a classifier and adversary
- Classifier penalized when adversary does well at predicting Z

Adversarial Networks

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

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



- Hyper-parameter λ controls trade-off
  - Large λ enforces f(…) to be pivotal, e.g. robust to nuisance
  - Small λ allows f(…) to be more optimal without Z variation

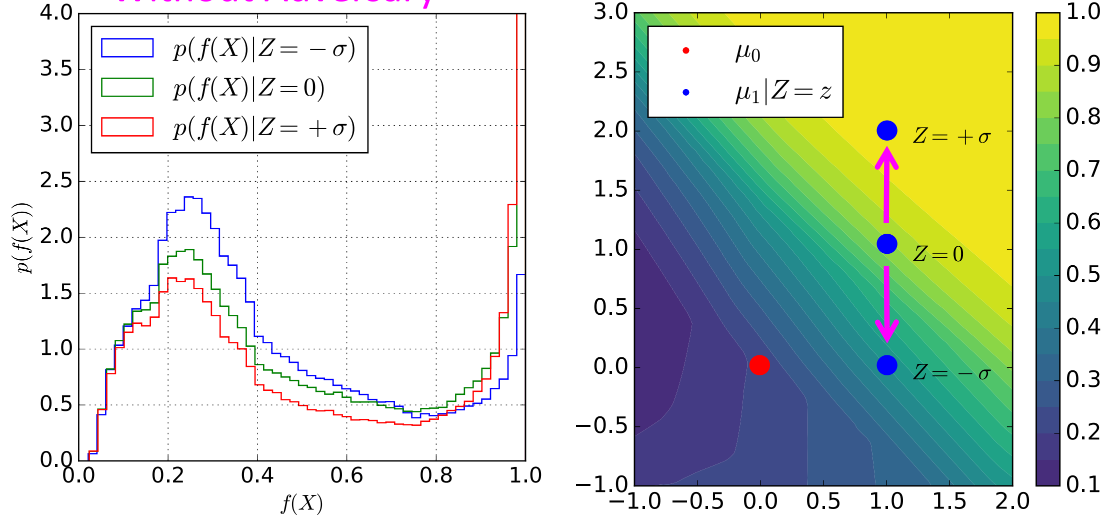
Learning to Pivot: Toy Example

- 2D example

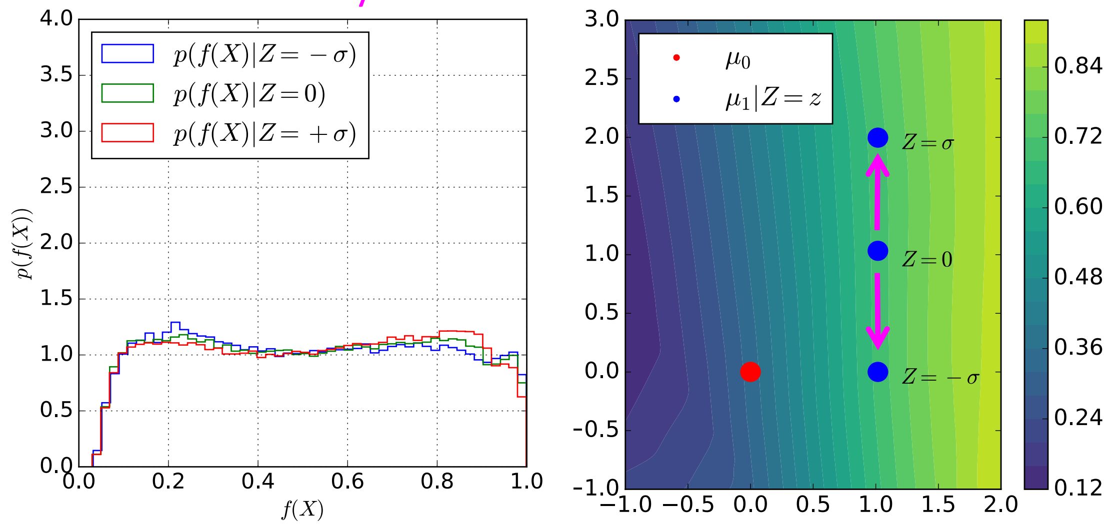
$$x \sim \mathcal{N}\left((0,0), \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}\right) \quad \text{when } Y = 0,$$

$$x \sim \mathcal{N}\left((1, 1+Z), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \quad \text{when } Y = 1.$$

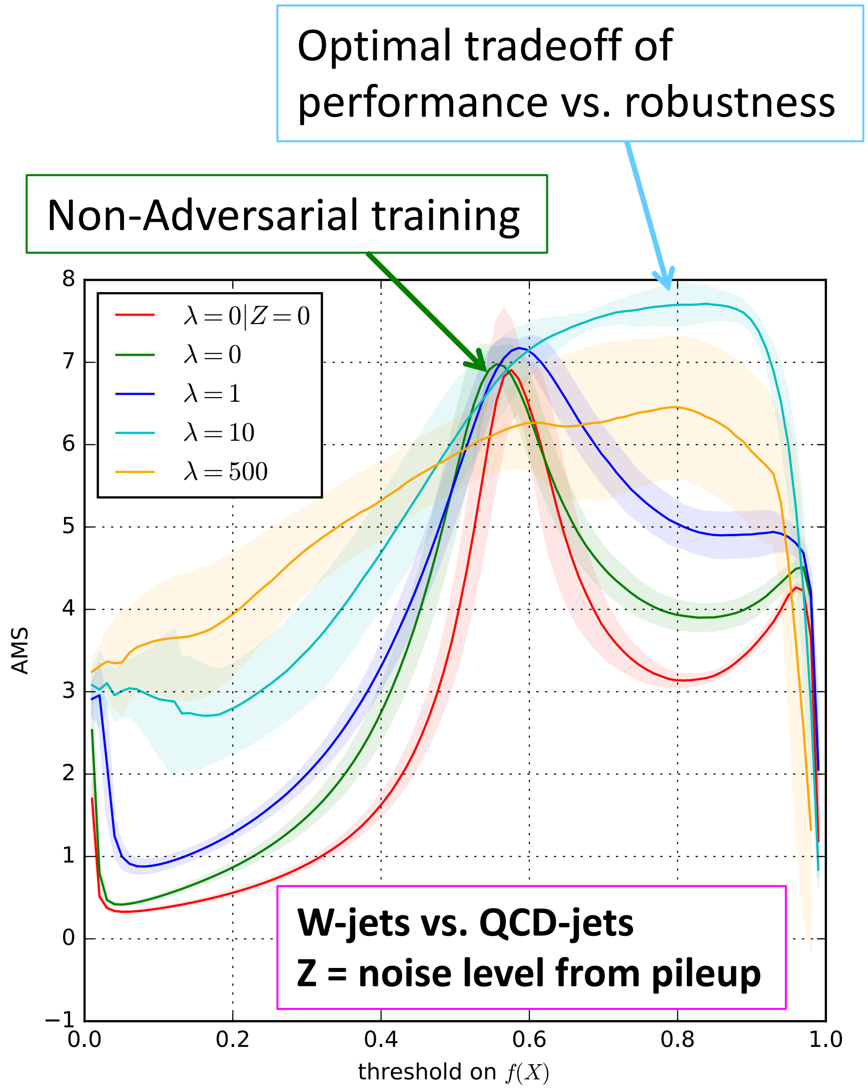
Without Adversary



With Adversary



Learning to Pivot: Physics Example



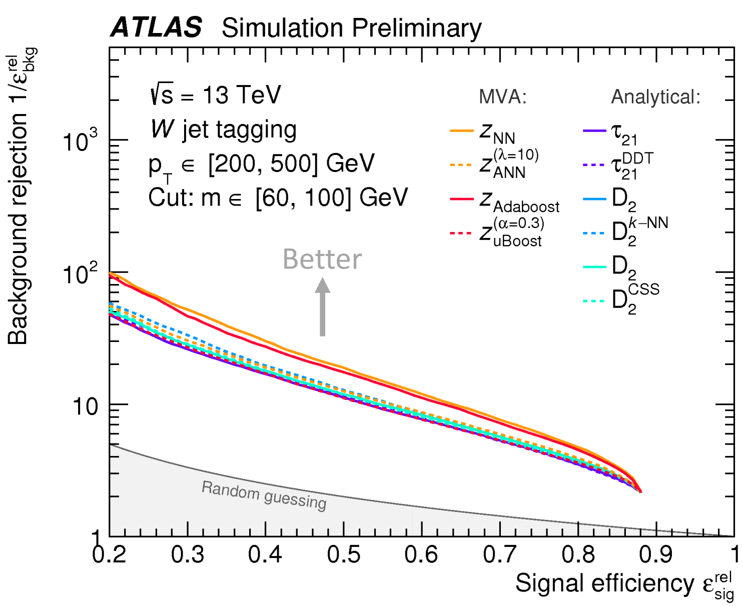
[AMS = Estimate of statistical significance including systematic uncertainty]

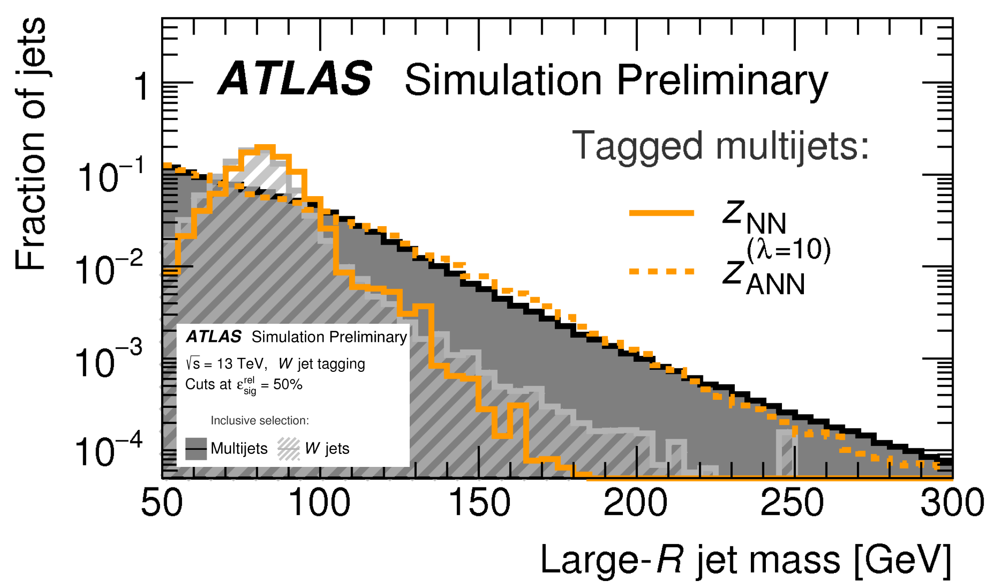
- λ=0, Z=0
  - Standard training with no systematics during training, evaluate systematics after training
- λ=0
  - Training samples include events with systematic variations, but no adversary used
- λ=10
  - Trading accuracy for robustness results in net gain in terms of statistical significance

Decorrelating Variables

- Same adversarial setup can decorrelate a classifier from a chosen kinematic variable [arXiv:1703.03507]
- Example: decorrelate classifier from jet mass, so as not to sculpt jet mass distribution with classifier cut

W-jets vs. QCD Jets





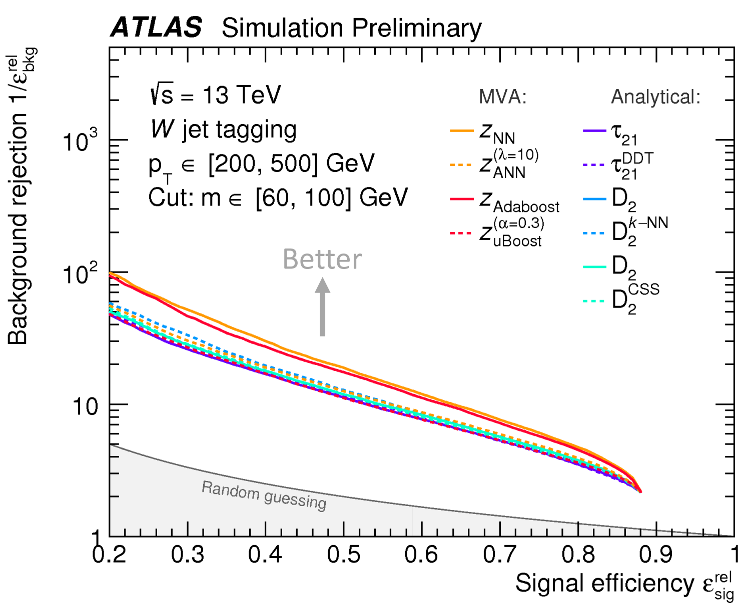
ATL-PHYS-PUB-2018-014

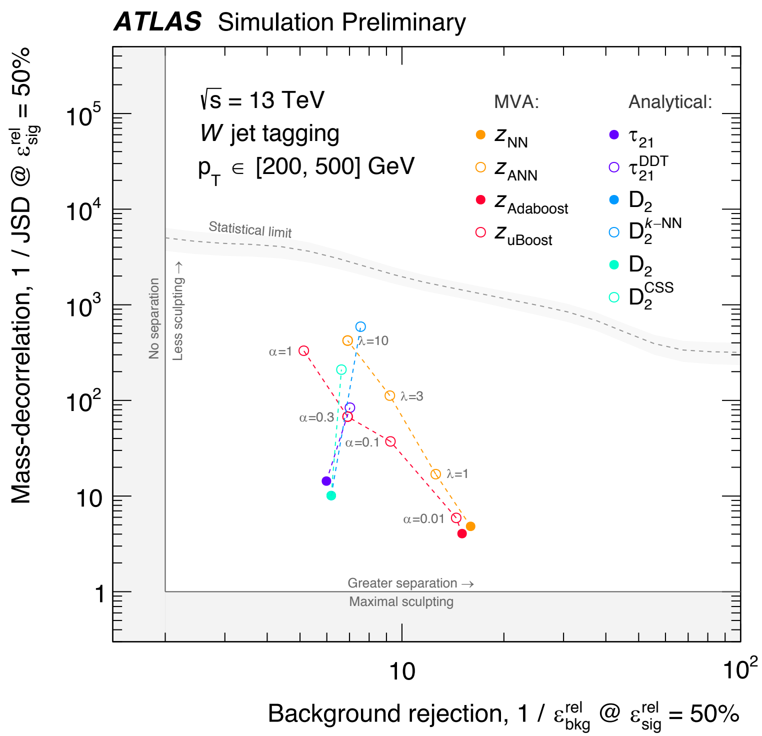
Decorrelating Variables

- Same adversarial setup can decorrelate a classifier from a chosen kinematic variable [arXiv:1703.03507]

- Example: decorrelate classifier from jet mass, so as not to sculpt jet mass distribution with classifier cut

W-jets vs. QCD Jets

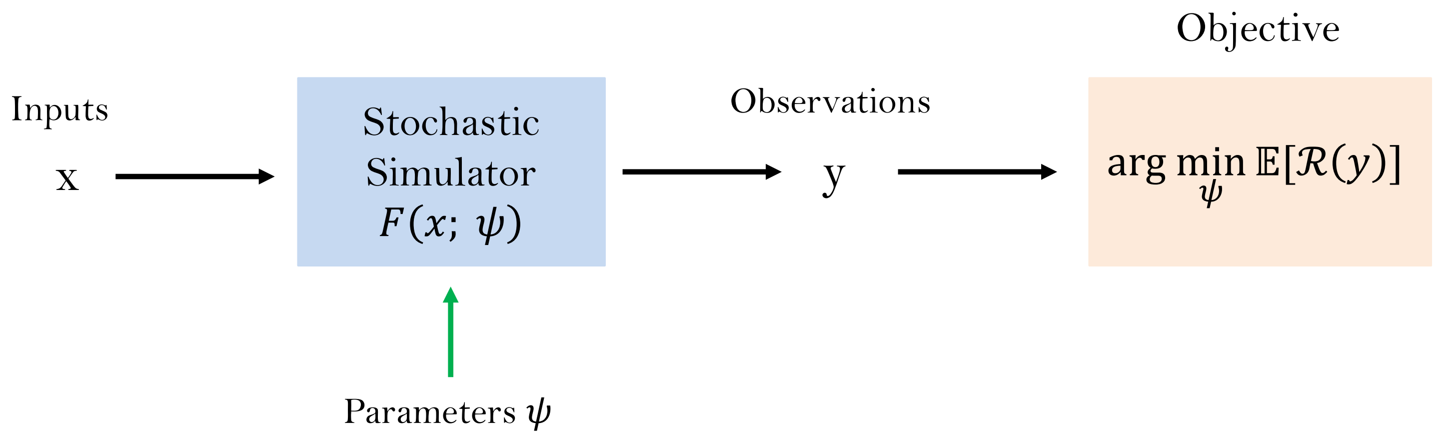




ATL-PHYS-PUB-2018-014

Non-ATLAS, but related, use of Generative Modeling Tools

Black Box Optimization



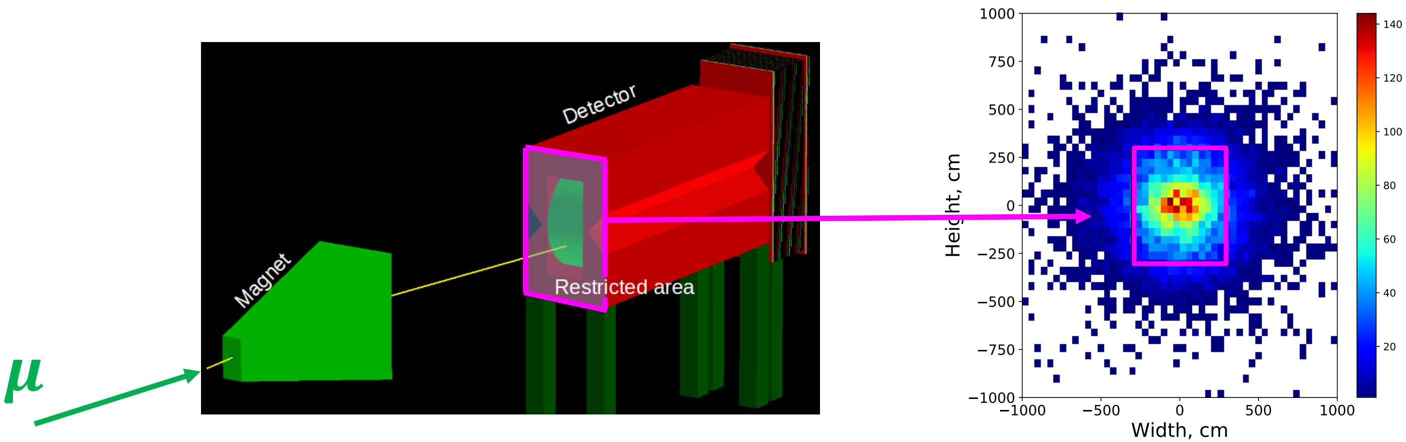
- Goal: Optimize simulator parameters to minimize objective

Black Box Optimization

Under Review
S. Shirbikov, V. Belavin, M. K. ,
A. Baydin, A. Ustyuzhanin

Inputs x → Stochastic Simulator $F(x;\, \psi)$ → Observations y → Objective $\arg\min_{\psi} \mathbb{E}[\mathcal{R}(y)]$

Parameters ψ → Stochastic Simulator



- Goal: Optimize simulator parameters to minimize objective

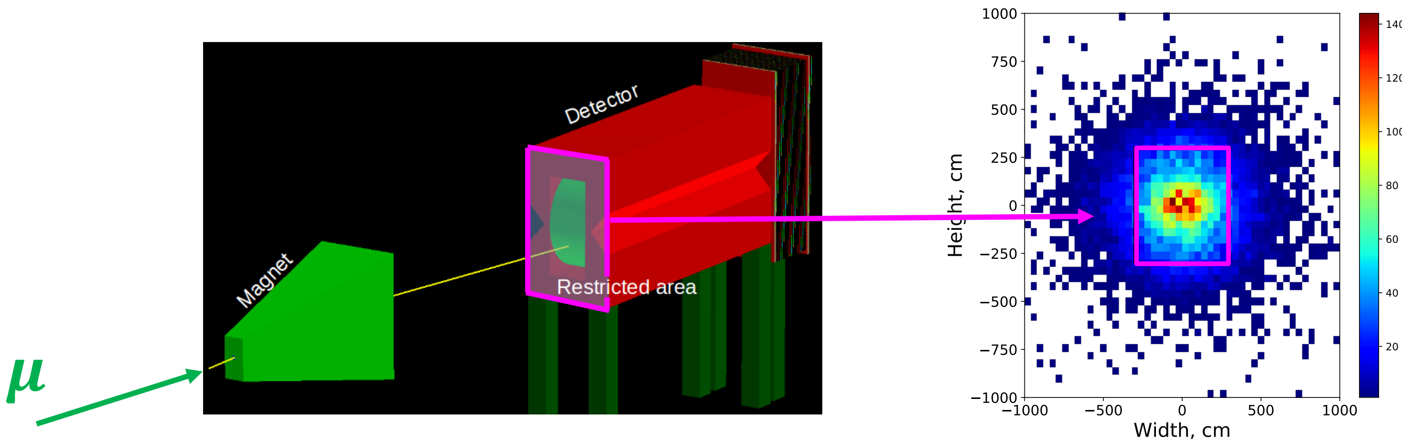
Black Box Optimization

Objective

Inputs

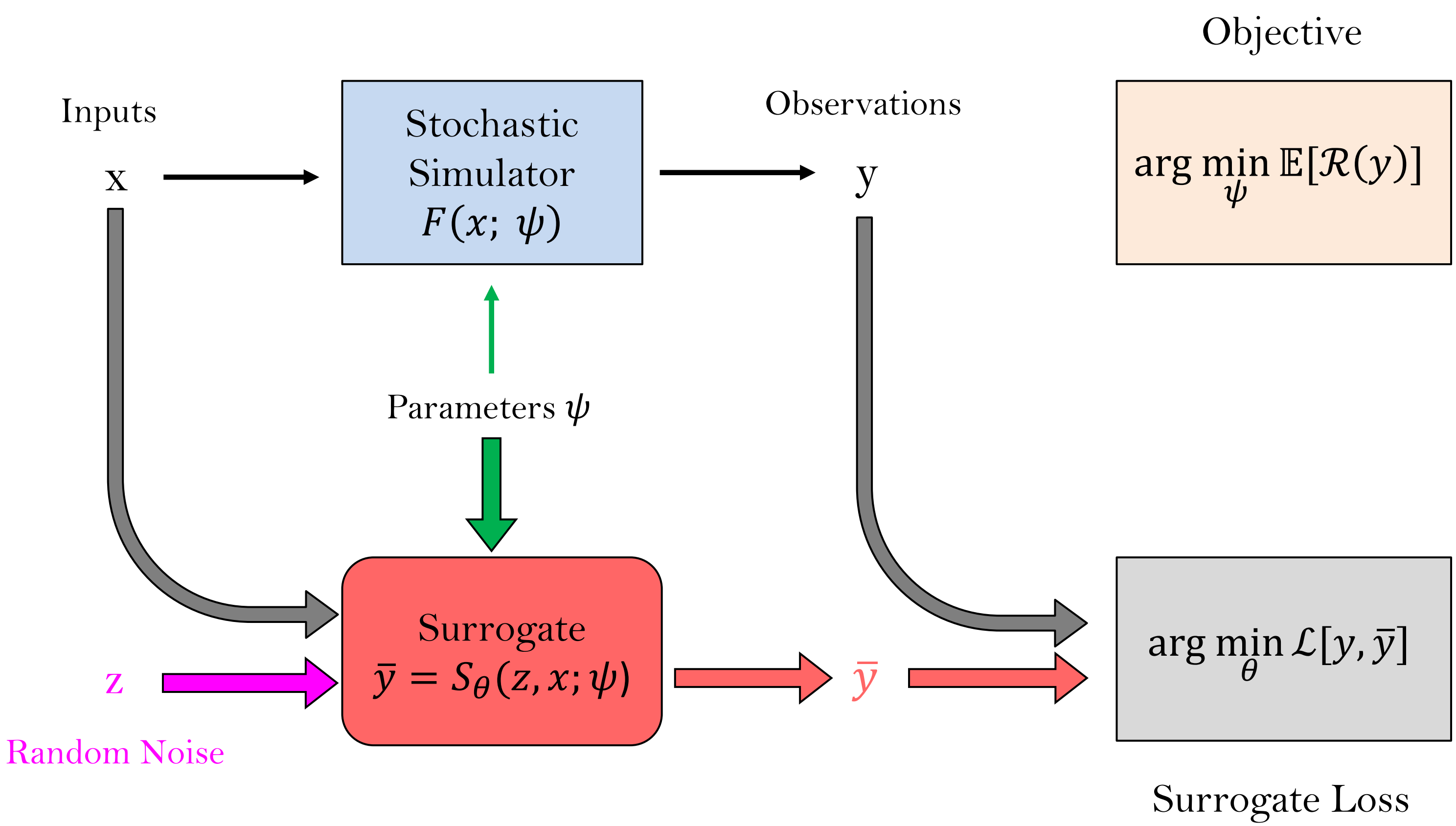
x → Stochastic Simulator $F(x;\ \psi)$ → Observations y → $\arg\min_{\psi} \mathbb{E}[\mathcal{R}(y)]$

Parameters ψ



- Goal: Optimize simulator parameters to minimize objective
- Can we approximate the simulator directly?

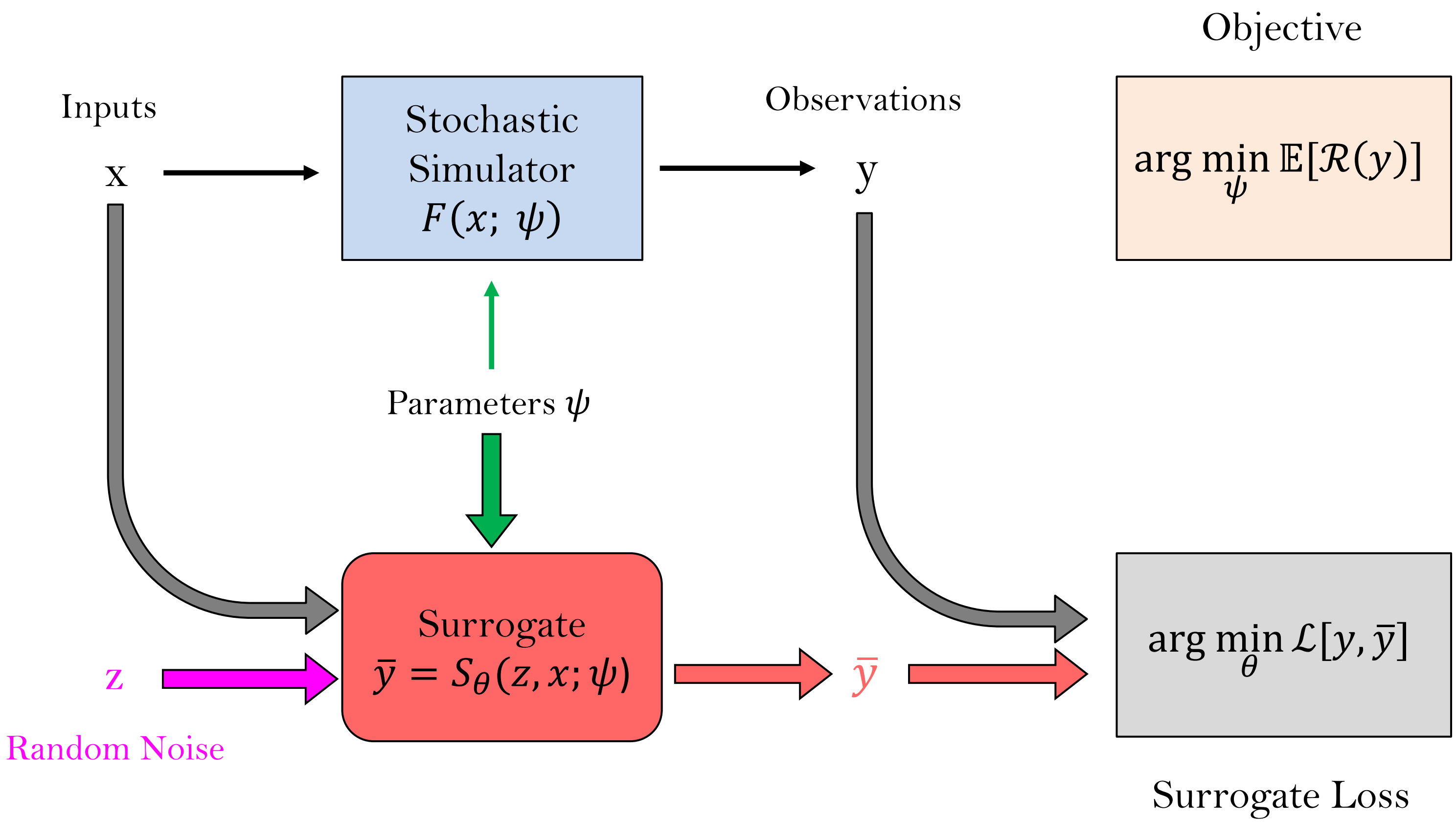
Differentiable Surrogates



- Train parameterized generative surrogate model S, i.e. GAN or flow, to approximate $F(x;\ \psi)$
  - Noise input to surrogate can account for stochastic nature of F

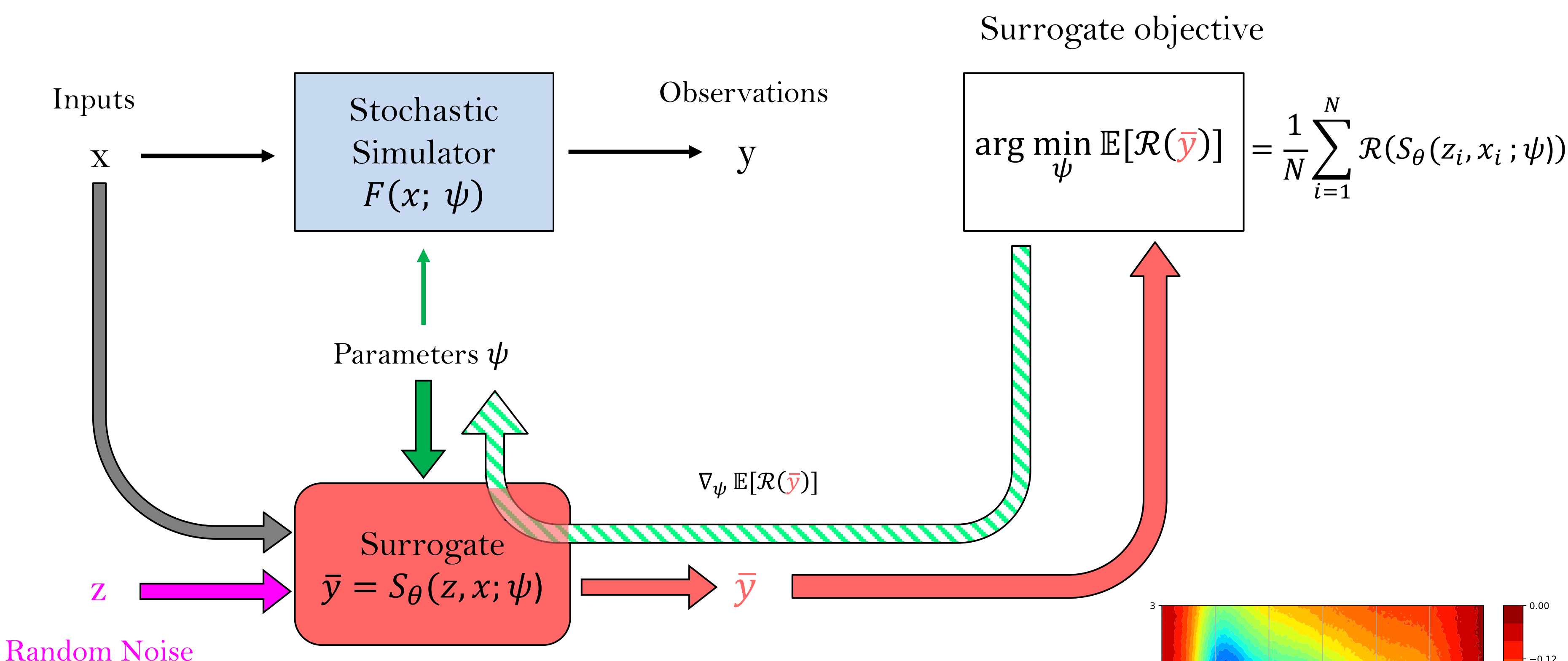
Differentiable Surrogates

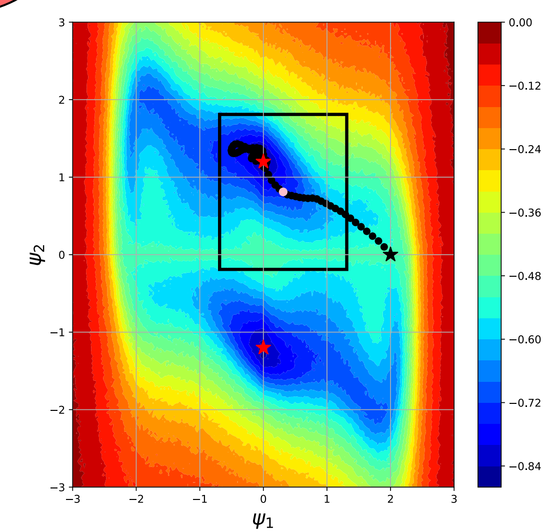
Under Review
S. Shirbikov, V. Belavin, M. K. ,
A. Baydin, A. Ustyuzhanin



- Surrogate samples $\bar{y} = S_\theta(z, x; \psi)$ are the output of a NN
  → $S_\theta(z, x; \psi)$ is a continuous function of its inputs
  → Can differentiate the samples w.r.t. the parameters!

Differentiable Surrogates

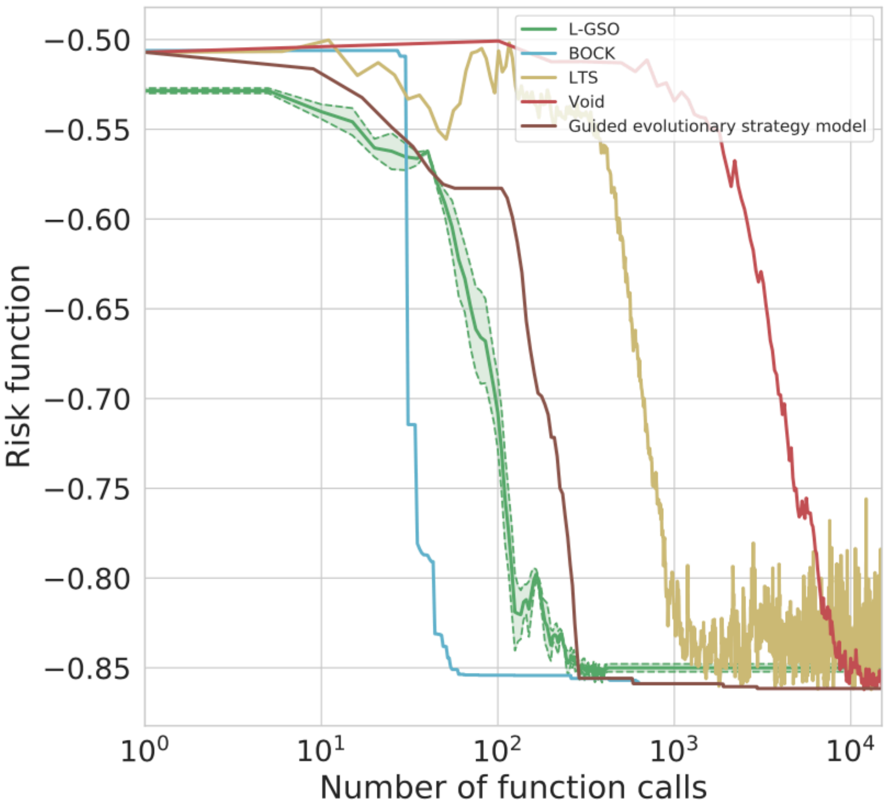




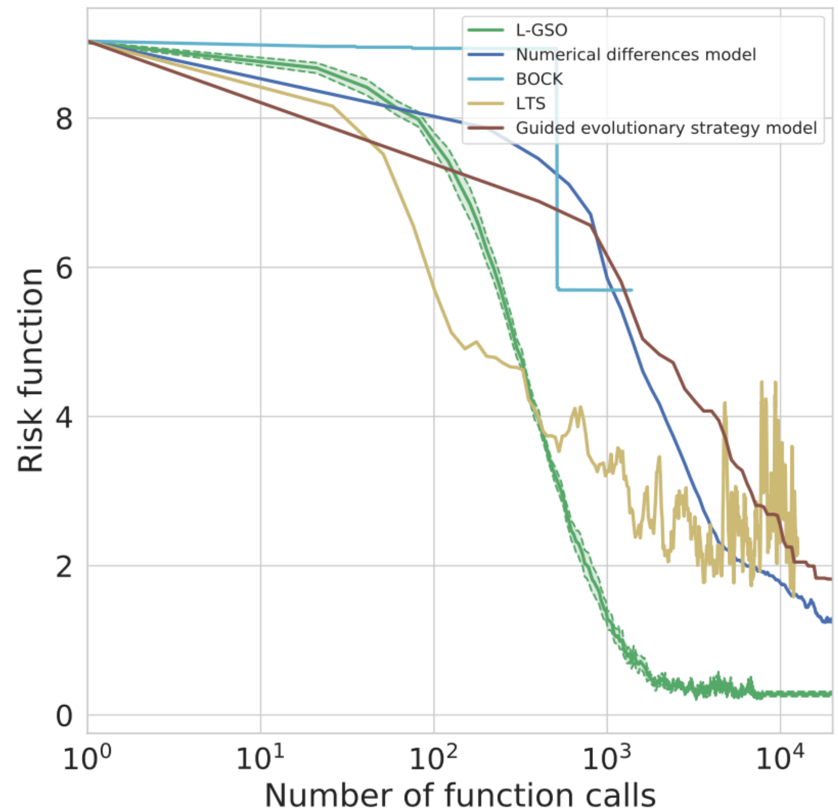
- Optimize objective with gradient descent using trained surrogate to produce differentiable samples

Optimization on Toy Examples

Three-Hump Camel function



Rosenbrock function



$$\psi^* = \arg\min_{\psi} \mathbb{E}[\mathcal{R}(y)] = \mathbb{E}[\sigma(y-10) - \sigma(y)], \text{s.t.}$$

$$y \sim \mathcal{N}(y; \mu_i, 1), i \in \{1,2\}$$

$$\mu_i \sim \mathcal{N}(x_i h(\psi), 1), \quad \mathrm{P}(i=1) = \frac{\psi_1}{||\psi||_2} = 1 - \mathrm{P}(i=2)$$

$$x_1 \sim \mathrm{U}[-2,0], \quad x_2 \sim \mathrm{U}[2,5],$$

$$h(\psi) = 2\psi_1^2 - 1.05\psi_1^4 + \psi_1^6/6 + \psi_1\psi_2 + \psi_2^2$$

$$\psi^* = \arg\min_{\psi} \mathbb{E}[\mathcal{R}(y)] = \arg\min_{\psi} \mathbb{E}[y], \quad \text{s.t.}$$

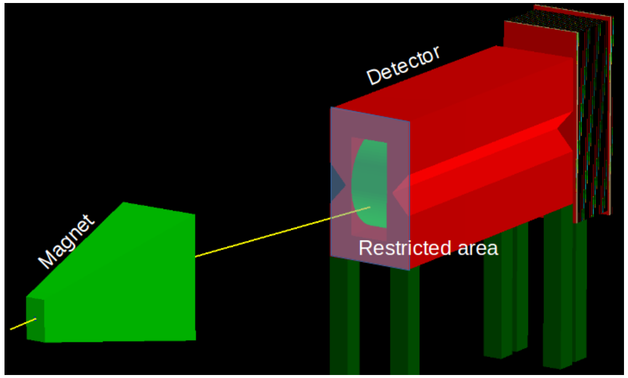
$$y \sim \mathcal{N}\left(y; \sum_{i=1}^{n-1} \left[(\psi_i' - \psi_{i+1}')^2 + (1-\psi_i')^2\right] + x, 1\right),$$

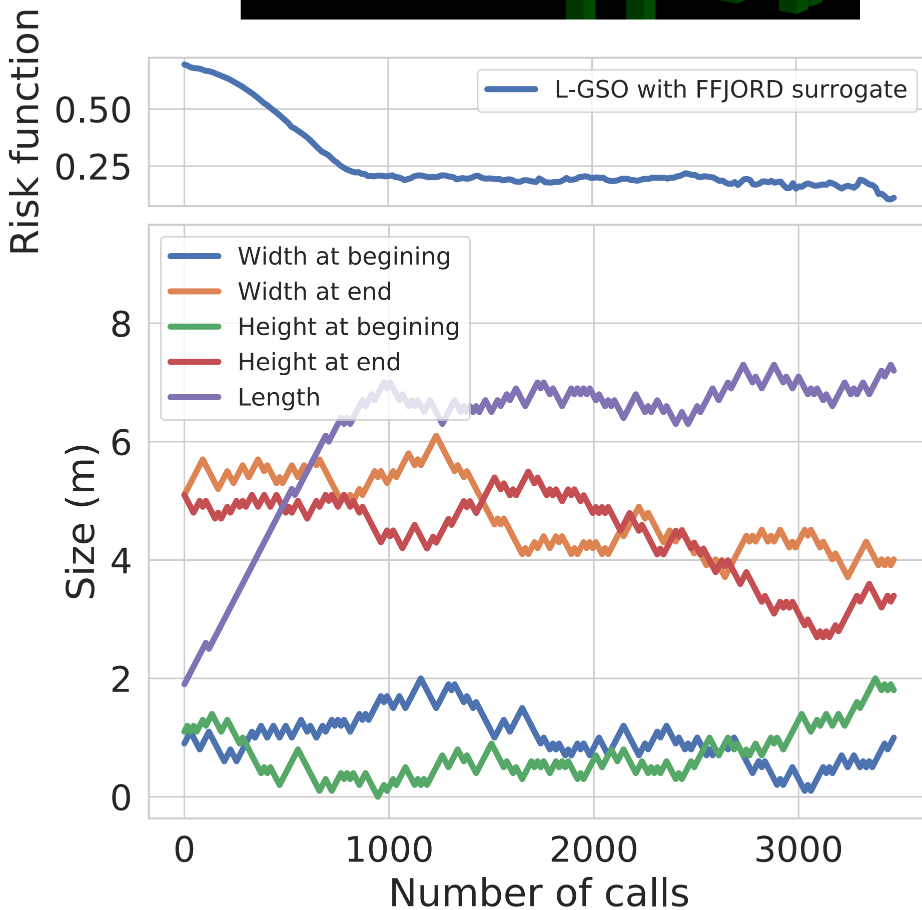
$$\psi' = A \cdot (\psi[\text{mask}]), \quad x \sim \mathcal{N}(x; \mu, 1), \quad \mu \sim \mathrm{U}[-10, 10]$$

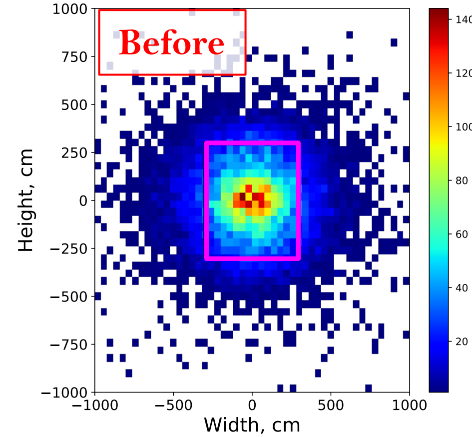
100 Dim parameter space projected on 10 Dim submanifold

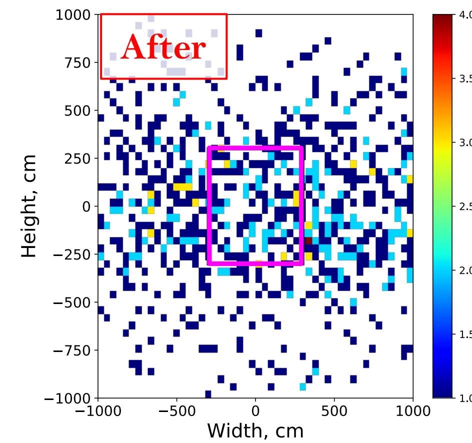
Optimization In Physics Example

Under Review
S. Shirbikov, V. Belavin, M. K. ,
A. Baydin, A. Ustyuzhanin









Conclusion

- Analysis pipeline is grounded in our detail physics domain knowledge
- Maintain our physics knowledge embedded in this pipeline while using ML to help solve some of the intractable challenges we face on ATLAS
- ML methods have shown strong performance improvements in reconstruction and analysis methods
- Techniques to deal with key challenges such as simulation computational cost and systematic uncertainty mitigation are under study

How can existing ML help us do better in tackling known challenges → How can existing ML allows us to approach new challenges → How can we design ML systems to our needs

backup