Machine Learning For Searches

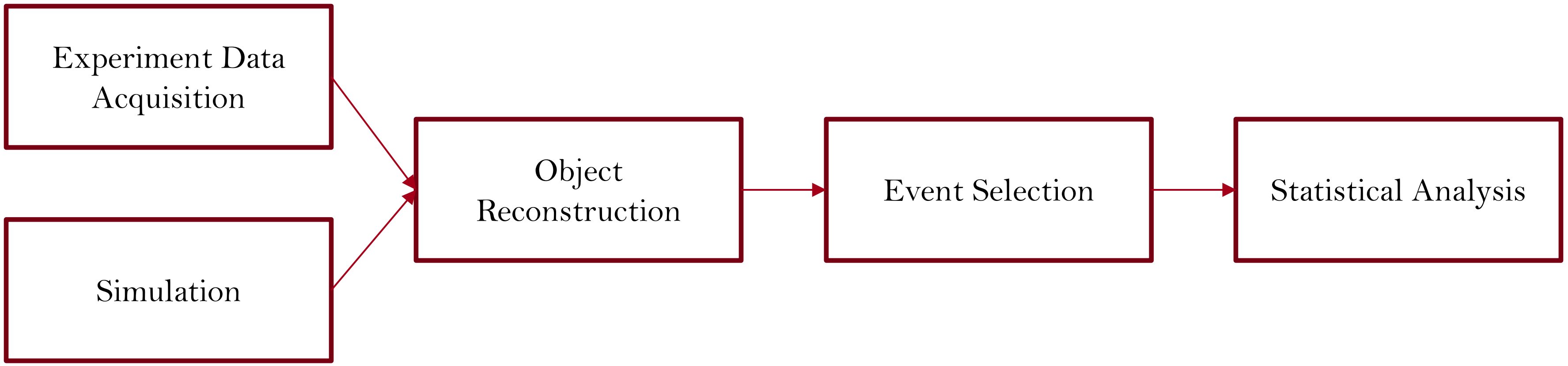
Michael Kagan

LHCP, May 21, 2019

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

The Search Plan



Physics knowledge drives development of experiments, simulations, reconstruction, and analysis pipeline

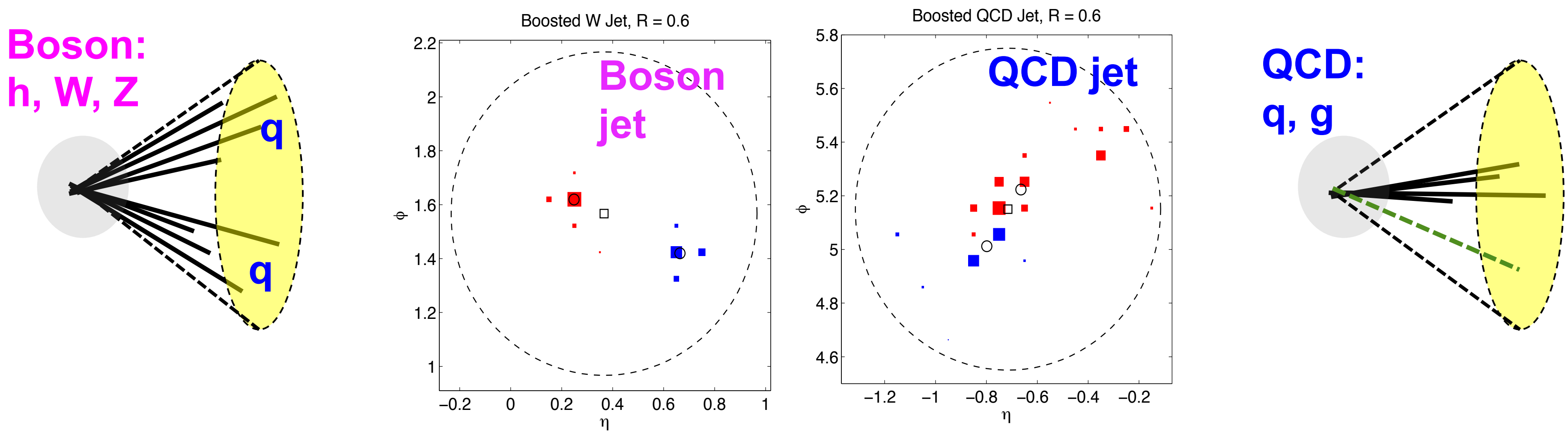
- Underlying theory drives our inference goals
- Mechanistic understanding of structure of events, particles interactions with material
- Compositionality: design detectors and algorithms to identify specific particles, and analyze them together as events

Much of this is intractable

- Don’t know p(shower | electron) or p(electron | shower)
- Can sample distributions with simulators encapsulating physics knowledge

Machine learning to augment and improve the pipeline, preserving our physics knowledge while by providing expressive and flexible models to study our data

Jet Tagging



- Can use internal jet (sub)structure of a jet for classification
- Wealth of domain expertise in feature engineering
- Can Machine Learning perform this classification?

Inductive Bias and Data Representation

Moving **inductive bias** from feature engineering to machine learning (neural network) model design

- Inductive bias ~ knowledge about the problem
- Feature engineering ~ hand crafted variables
- Model design ~ the data representation and the structure of the machine learning model / network

Need a good inductive bias, i.e. physical motivation, for data representation and model structure

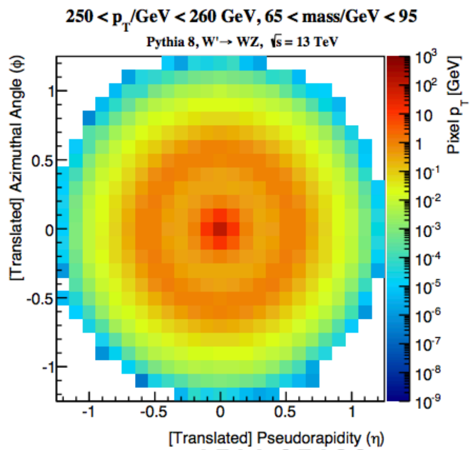
- Better learn to approximate our data
- Easier to extract information about what is learned?

We can represent jets in different ways

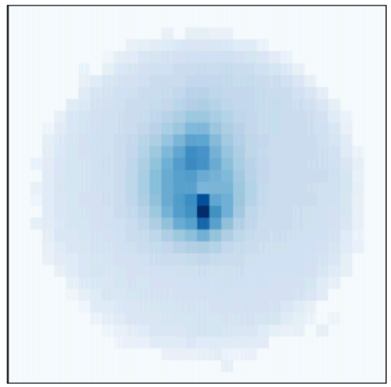
We can utilize different classes of models

Jets as Images

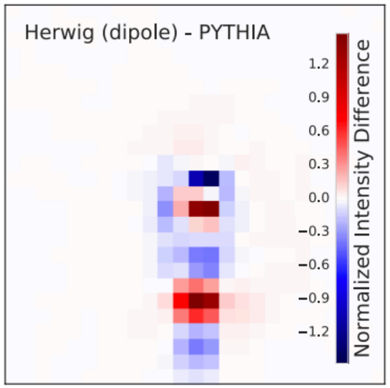
[Based on slides H. Qu]



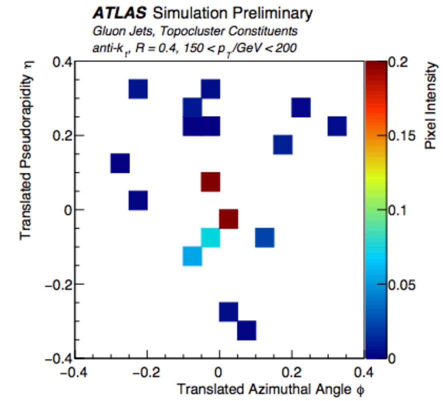
1511.05190



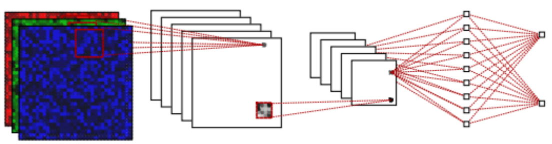
1603.09349



1609.00607



ATL-PHYS-PUB-2017-017



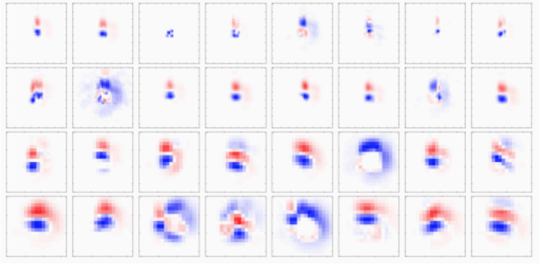
red = transverse momenta of charged particles

green = the transverse momenta of neutral particles

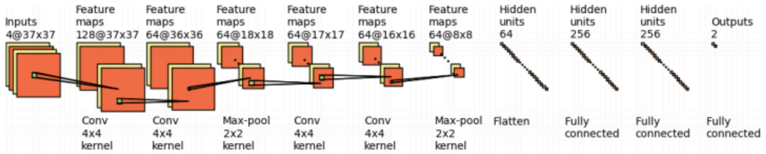
blue = charged particle multiplicity

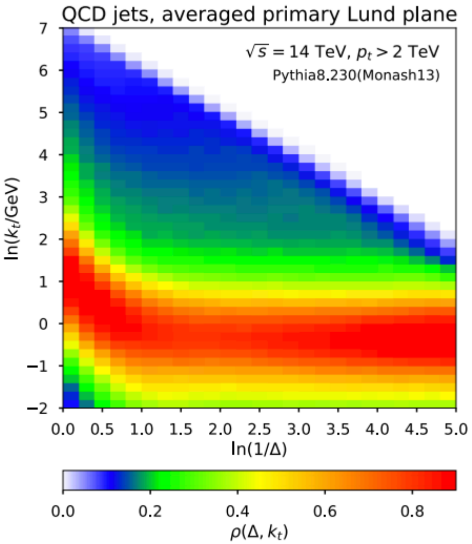
1612.01551

1701.08784



1803.00107



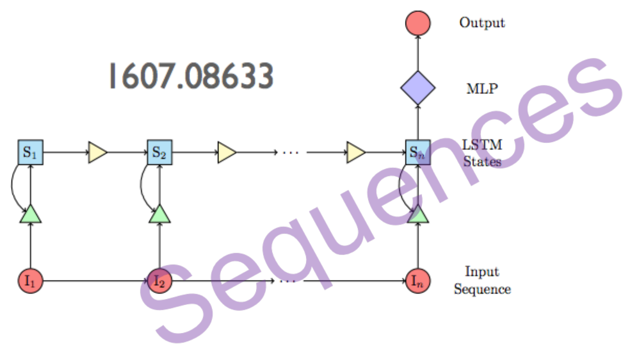


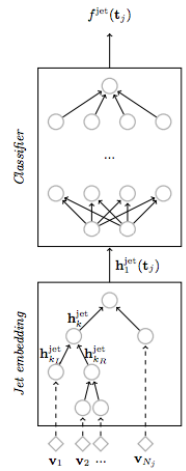
1807.04758

Jets as Collections of Particles

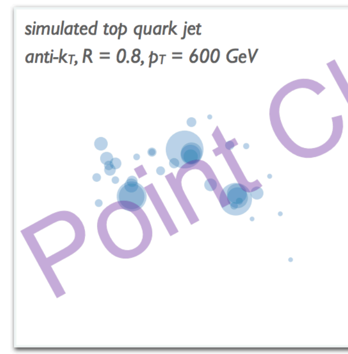
[Based on slides H. Qu]

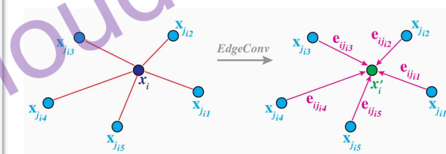
1607.08633





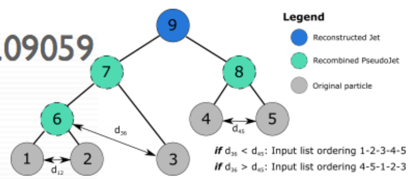
1702.00748

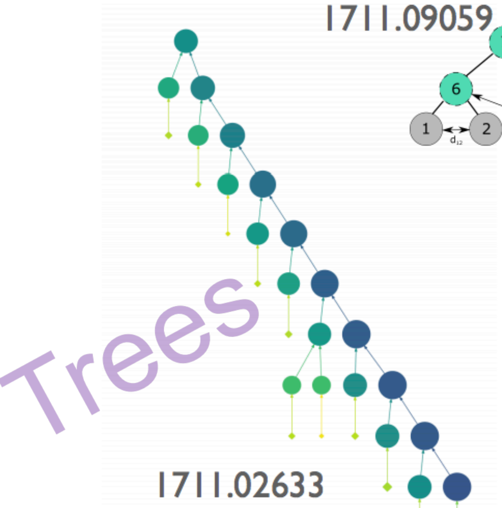




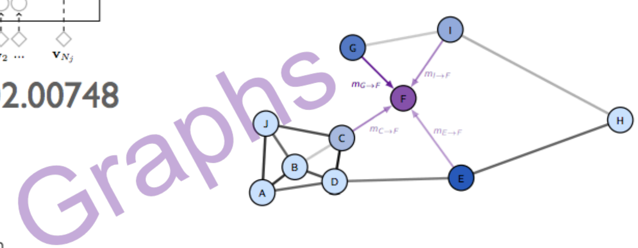
1902.08570

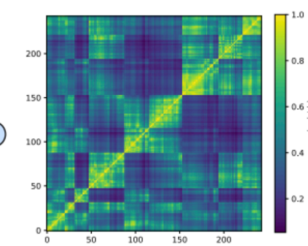
1711.09059



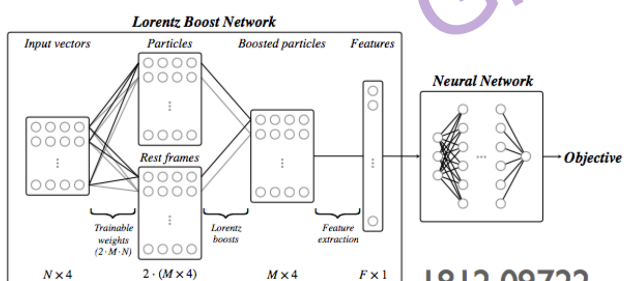


1711.02633

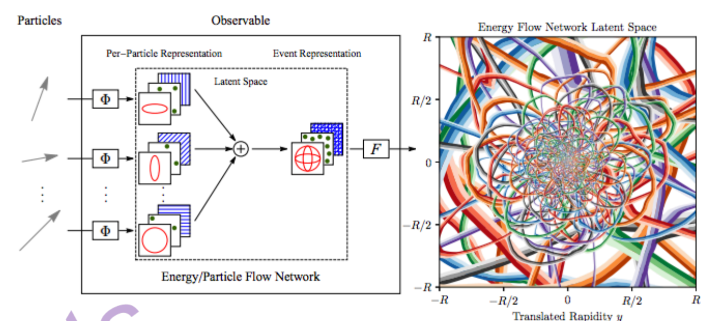




NIPS2017 workshop [http://bit.ly/2AkwYRG]



1812.09722



1810.05165

$$k_{\mu,i} \xrightarrow{\text{CoLa}} \tilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$$

$$\text{with} \quad C = \begin{pmatrix} 1 & 1 & \cdots & 0 & \chi_1 & \cdots & 0 & C_{1,N+2} & \cdots & C_{1,M} \\ \vdots & & \ddots & & & \ddots & & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 1 & 0 & \cdots & \chi_N & C_{N,N+2} & \cdots & C_{N,M} \end{pmatrix} .$$

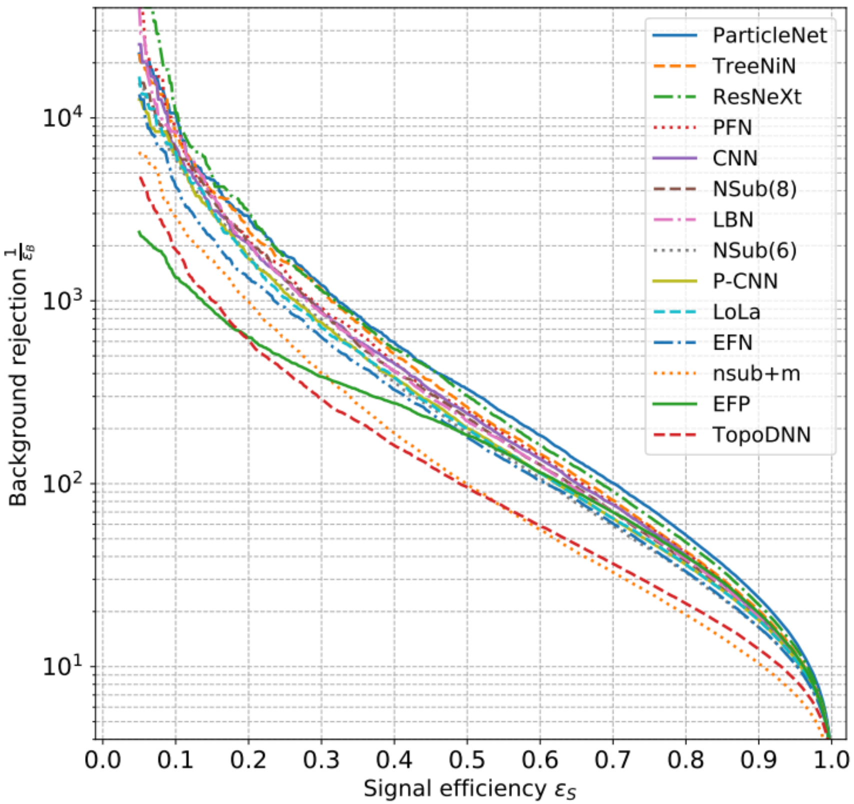
$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ p_T(\tilde{k}_j)\Delta R_{j,\text{jet}} \\ w^{(E)}_{jm}\, E(\tilde{k}_m) \\ w^{(d)}_{jm}\, d^2_{jm} \\ E_T(\tilde{k}_j)E_T(\tilde{k}_m)(\Delta R_{jm})^{0.2} \end{pmatrix} ,$$

Sets

1707.08966, 1812.09223

And more…

Jet Tagging with ML



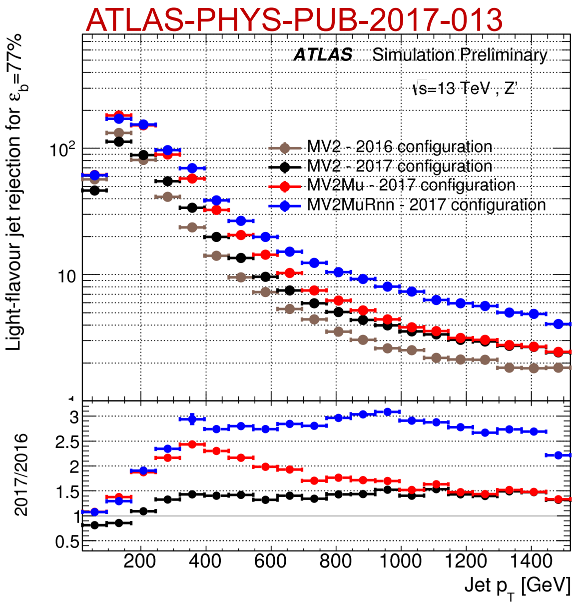
arXiv:1902.09914

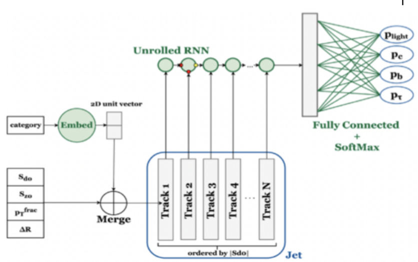
	AUC	Acc	$1/\epsilon_B$ ($\epsilon_S = 0.3$) single	mean	median	#Param
CNN [16]	0.981	0.930	914±14	995±15	975±18	610k
ResNeXt [30]	0.984	0.936	1122±47	1270±28	1286±31	1.46M
TopoDNN [18]	0.972	0.916	295±5	382± 5	378 ± 8	59k
Multi-body N-subjettiness 6 [24]	0.979	0.922	792±18	798±12	808±13	57k
Multi-body N-subjettiness 8 [24]	0.981	0.929	867±15	918±20	926±18	58k
TreeNiN [43]	0.982	0.933	1025±11	1202±23	1188±24	34k
P-CNN	0.980	0.930	732±24	845±13	834±14	348k
ParticleNet [47]	0.985	0.938	1298±46	1412±45	1393±41	498k
LBN [19]	0.981	0.931	836±17	859±67	966±20	705k
LoLa [22]	0.980	0.929	722±17	768±11	765±11	127k
Energy Flow Polynomials [21]	0.980	0.932	384			1k
Energy Flow Network [23]	0.979	0.927	633±31	729±13	726±11	82k
Particle Flow Network [23]	0.982	0.932	891±18	1063±21	1052±29	82k
GoaT	0.985	0.939	1368±140		1549±208	35k

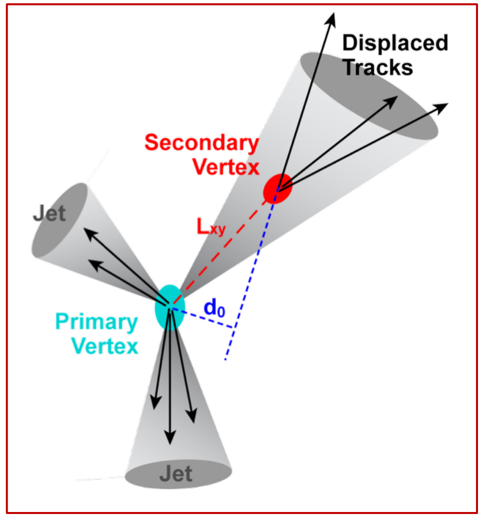
Appear to be reach performance asymptote by several models

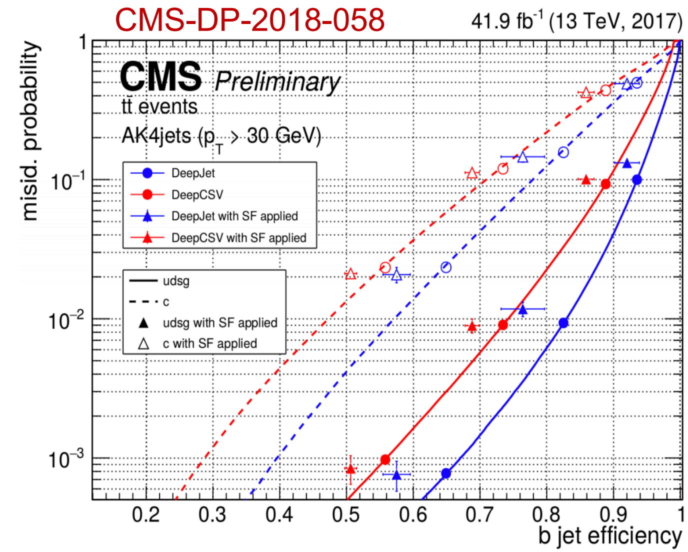
Key for use in experiments: Understanding computational requirements and sensitivity to systematic uncertainties

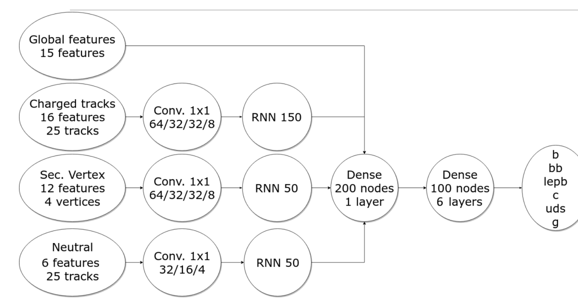
Flavour Tagging: Deep Learning in Experimental Action











Finding jets containing long-lived b-hadrons is key to finding H, Z, Top

- Complex decay topology drives need for powerful algorithms
- (Physics driven) Ordering of set of tracks / vertices to analyze as a sequence
- Sequence based algorithm to account for long range correlations among tracks!

Enforcing Invariance

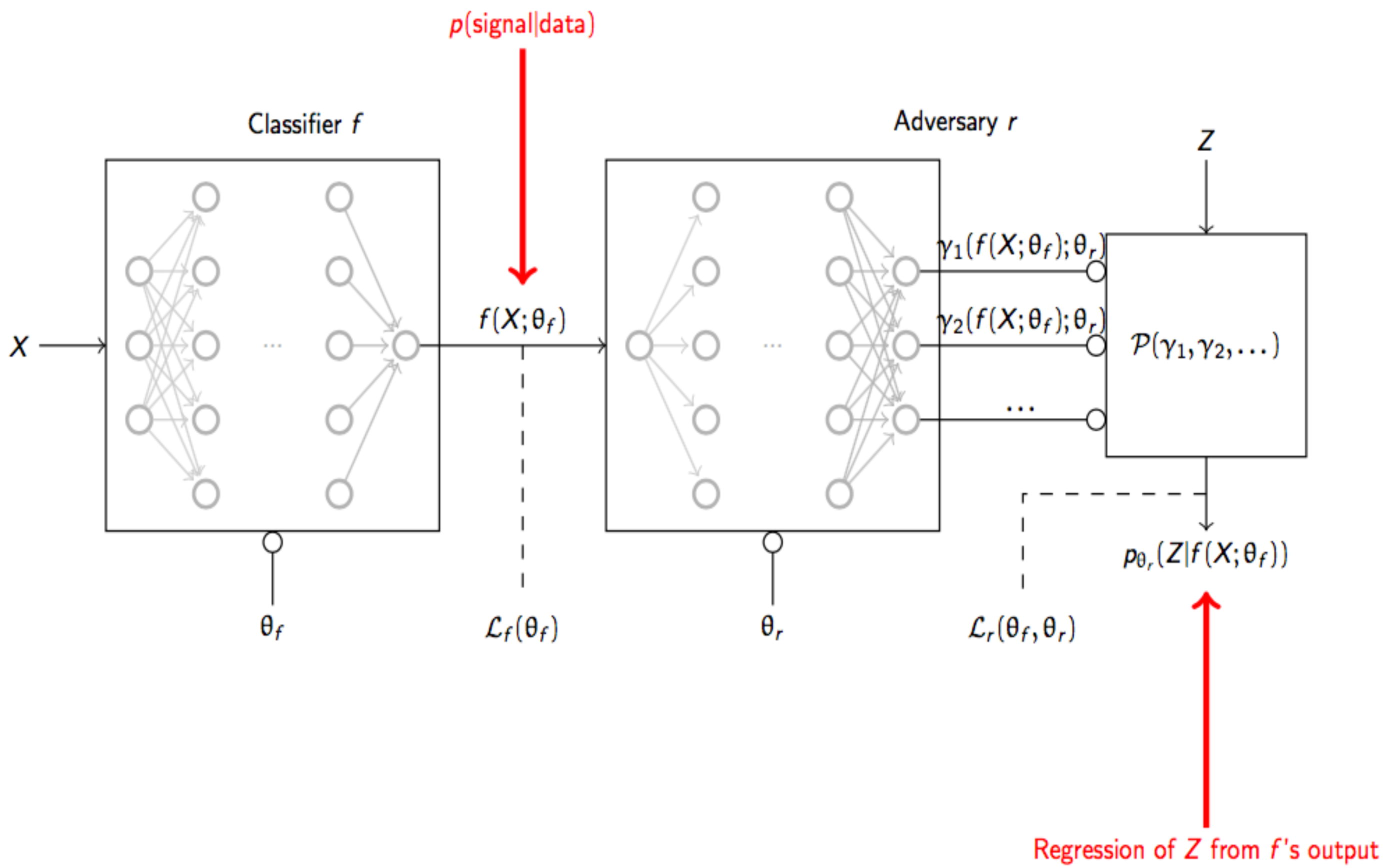
With flexibility comes complexity:

- Hard to control how models learn and 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.)

Adversarial Learning

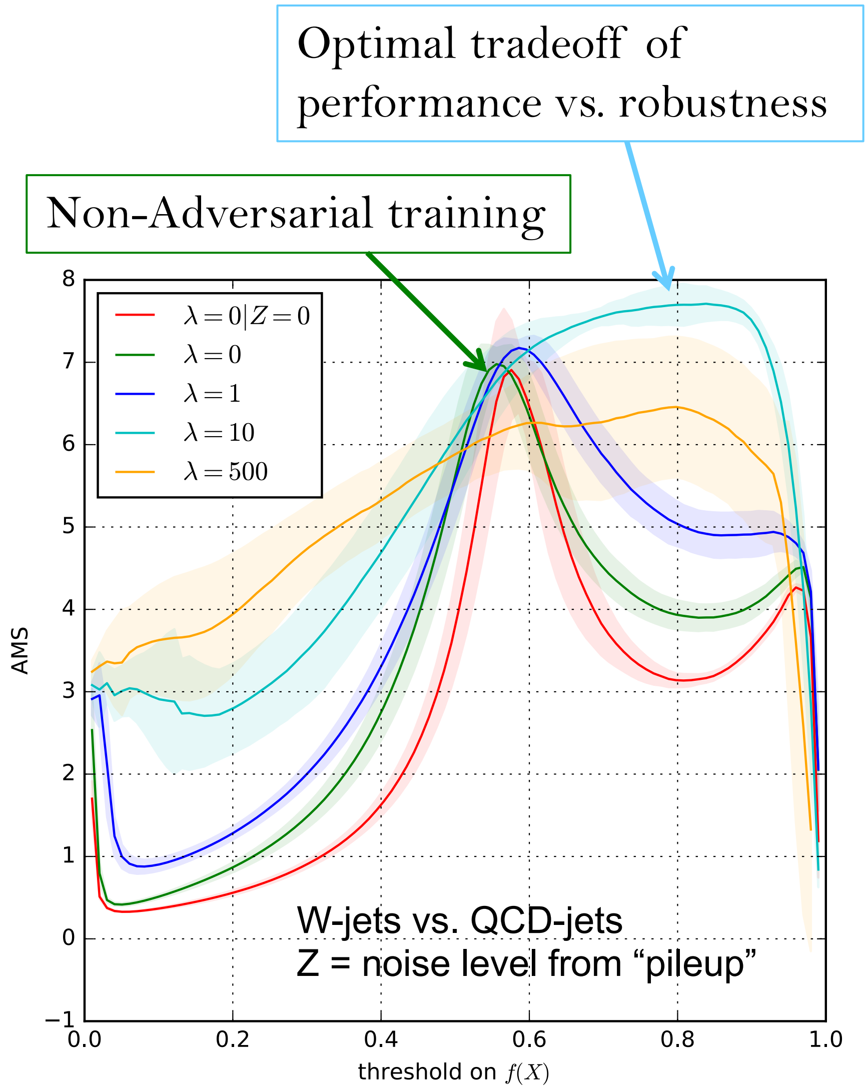
[arXiv:1611.01046]



Adversarial Approach:

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

Learning to Pivot: Physics Example



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

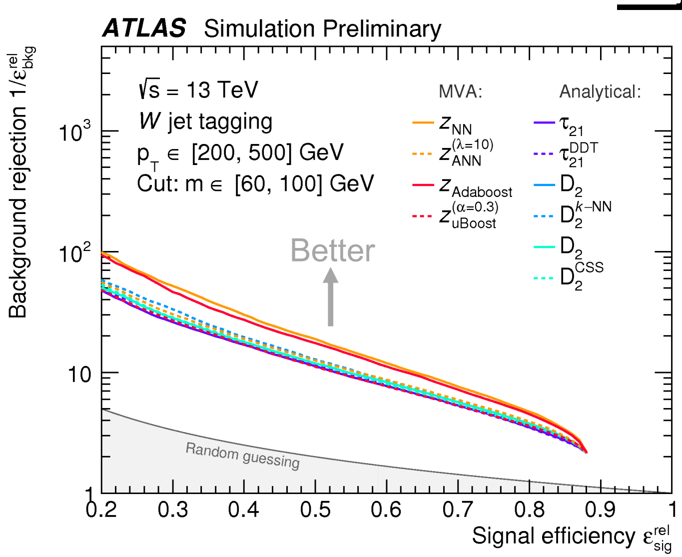
[AMS = Estimate of statistical significance including systematic uncertainty]

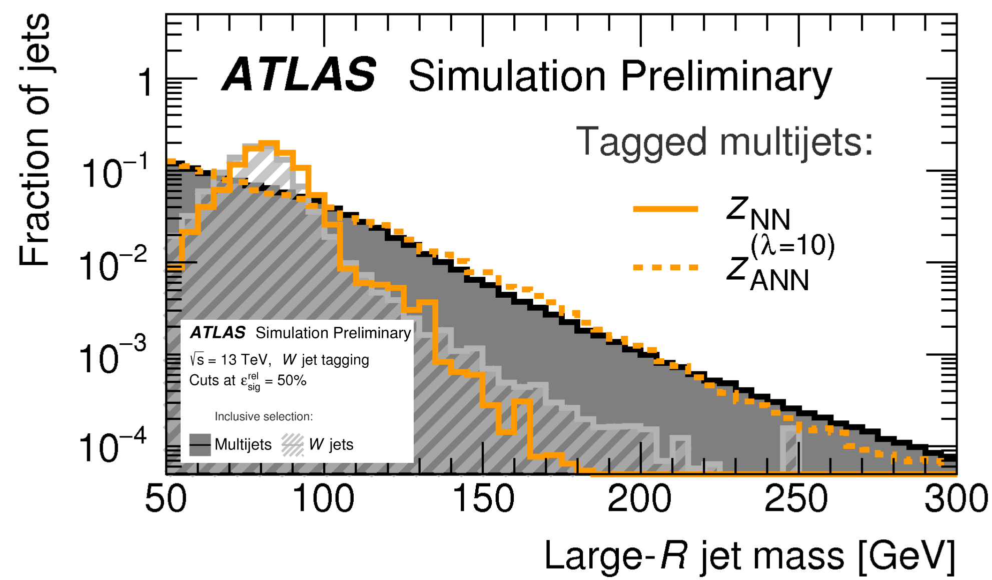
Decorrelating Variables

Same adversarial setup can decorrelate a classifier from a chosen variable (rather than nuisance parameter) [arXiv:1703.03507]

For 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]

Looking for Signals

Machine Learning driven reconstruction techniques allow us to improve the identification of known particle signatures in detectors

Typically combine information from several identified particles to search for signals / perform measurements.

When we know what signal we are looking for

- Can rely on standard MC and data driven techniques

What if we don't know what signal we are looking for?

ML Enhanced Resonance Finding with CWola Hunting

Want to look for resonance but be as agnostic as possible to features, e.g. if

- We don't have a theory yet to predict it
- We don't think MC models its features well

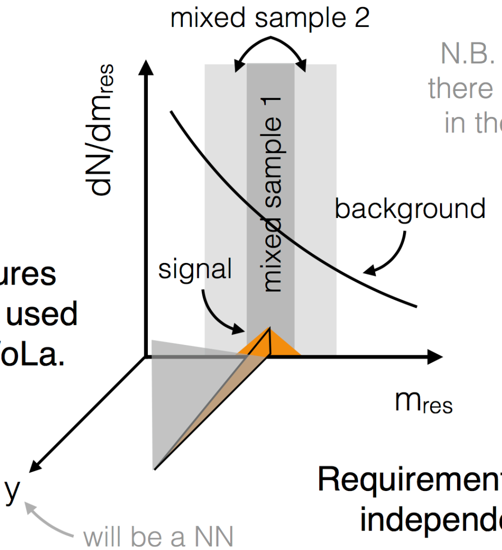
Density ratio trick

$$D(x) = \frac{1}{1+\frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}} = \frac{1}{1+\frac{1}{r(x)}\frac{p(y=0)}{p(y=1)}}$$

- D(x) is discriminator, e.g. ML model
- r(x) is the likelihood ratio

Works even if classes are not pure signal or background, if signal fractions are different!

- Train on data directly in samples with different signal fractions!
- Build mass-independent classifier to not sculpt mass distribution
- Apply varying thresholds on classifier
- Bump Hunt!

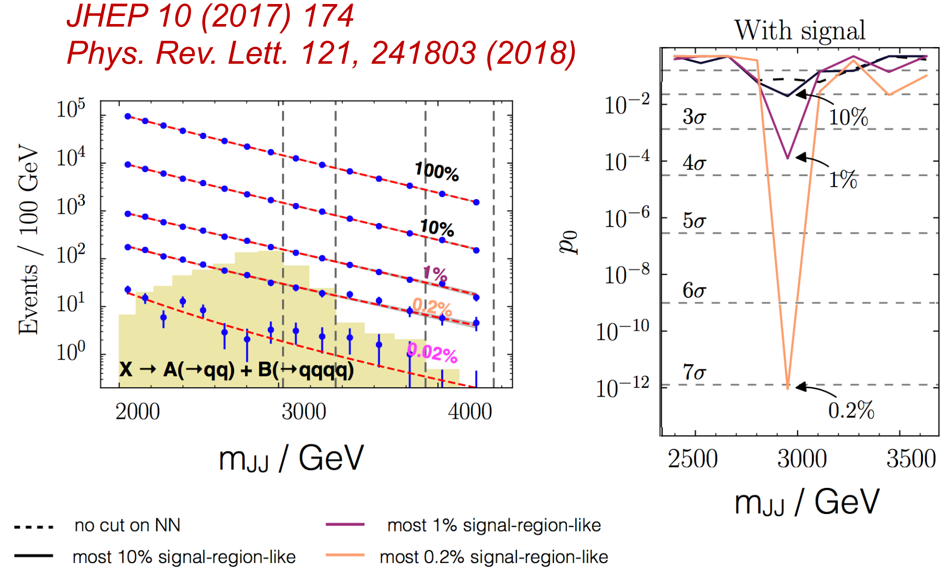


N.B. it is okay that there is some signal in the sidebands!

Other features can then be used to train CWoLa.

Requirement: y is (nearly) independent of m_{res}.

JHEP 10 (2017) 174
Phys. Rev. Lett. 121, 241803 (2018)



Images from B. Nachman

Don't know what to look for? Anomaly Detection Autoencoders

Anomaly detection: find rare events that differ from standard or majority data

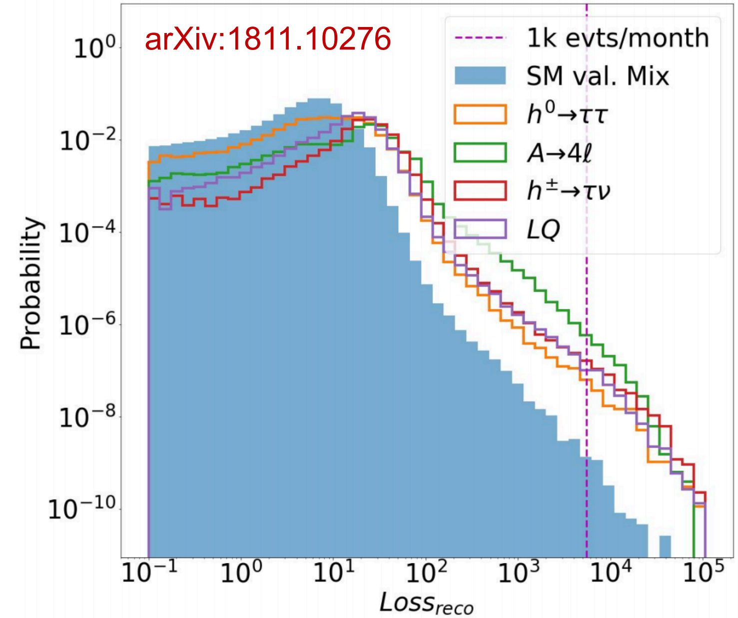
- Define standard: i.e. Standard Model
- Anomaly: BSM events not like the SM

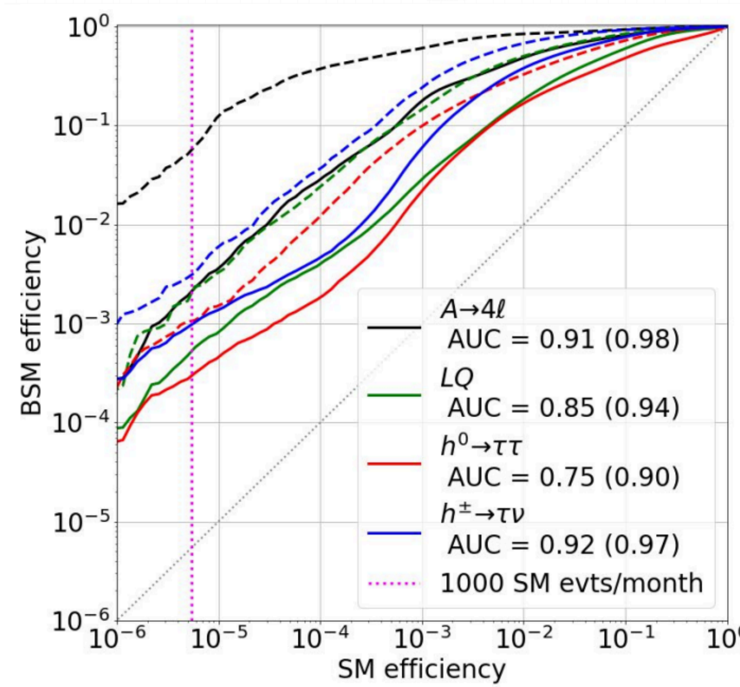
Look for BSM events with small SM probability, $p_{SM}(x)$ … but don't know $p_{SM}(X)$!

(Variational) AutoEncoder

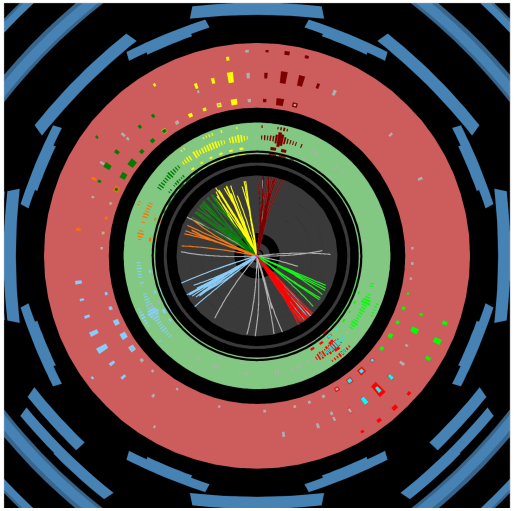
- Latent variable model, latent space z
- Learn encoder p(z|x) and Decoder p(x|z)
- ***Key Idea***: Ability to reconstruct input after encoding into latent space should be diminished for non-standard (i.e. BSM) data

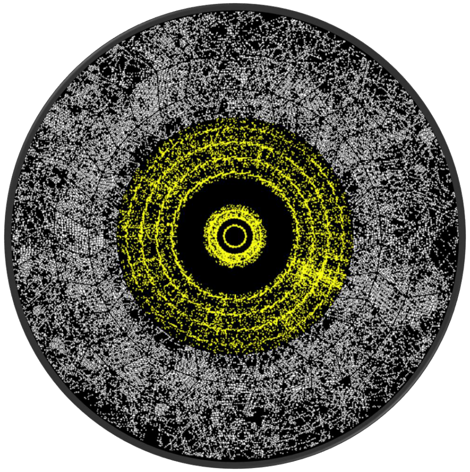
- Growing literature: [1808.08979](), [1808.08992](), [1807.10261](), [1811.10276]()





Resource Constraints





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

- Major challenges in triggering, large scale simulation, and high multiplicity tracking
- New tools and developments in ML may help address some of these challenges

Simulation

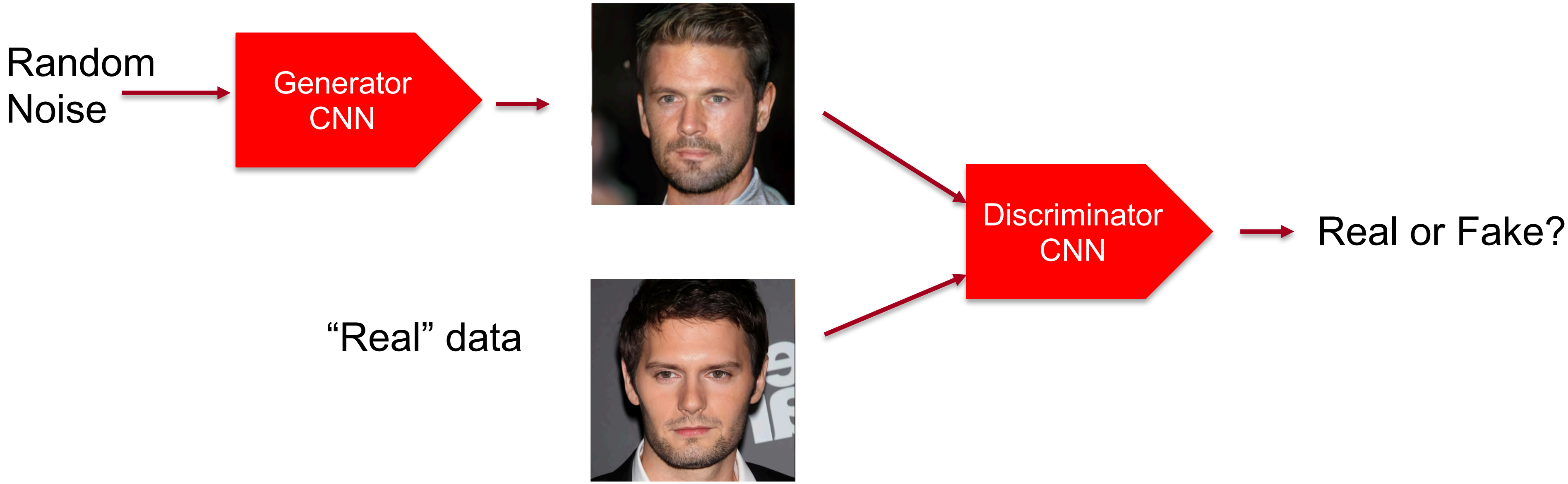
- Accurate but often costly simulation of particle interactions with material, that produces sample and not analytic P(energy deposits | particle)
- *ML approach*: Generative models to learn data distribution, p(x), and produce samples?

Trigger

- High performance algorithms early in trigger to reduce backgrounds for key signals?

Deep Generative Models for Simulation: GANS

arXiv:1406.2661



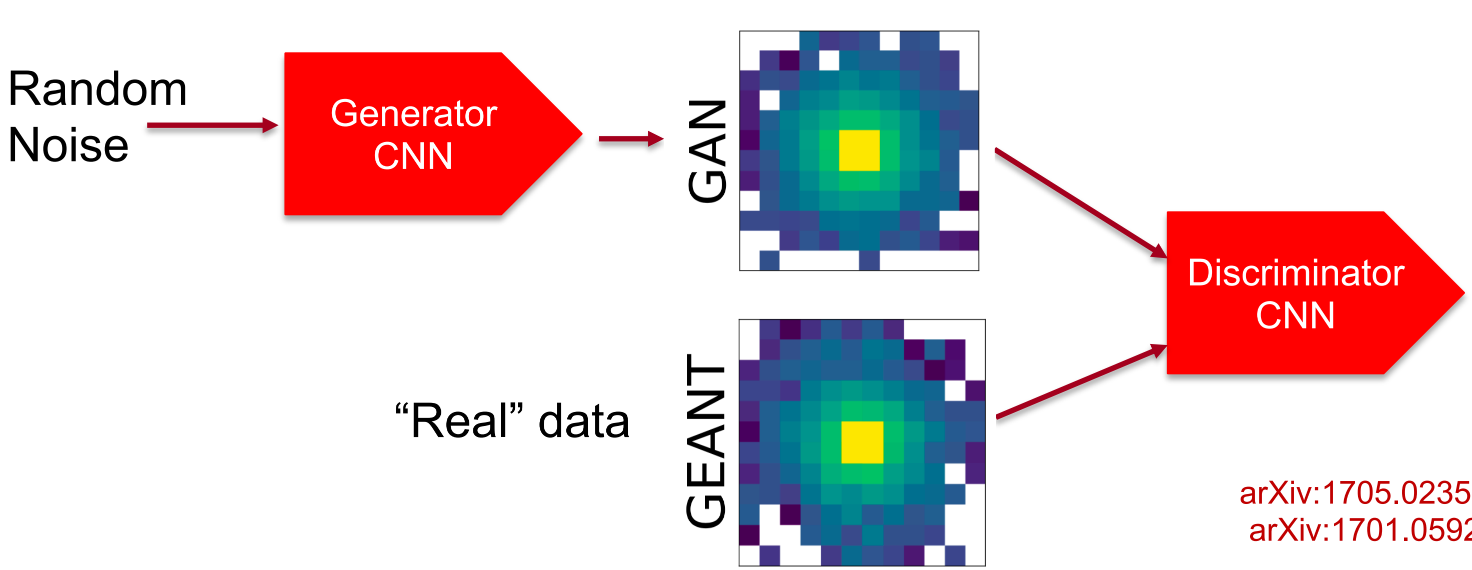
Generative Adversarial Network

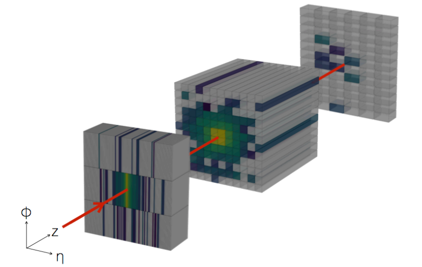
Generator produces images from random noise and tries to trick discriminator into thinking they are real

Classifier tries to tell the difference between real and fake images

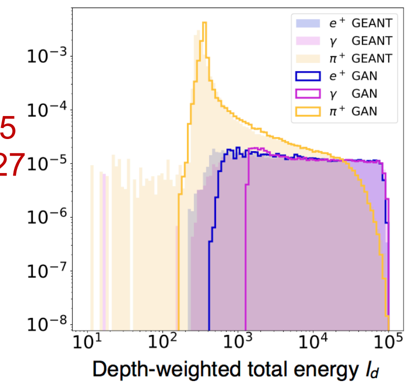
Images: arXiv:1710.10196

GANs / VAEs Generating Jet-images, and 3D calo-clusters

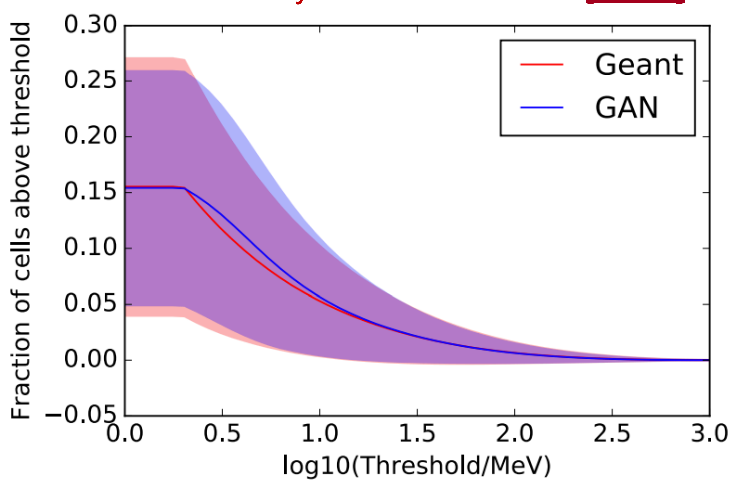


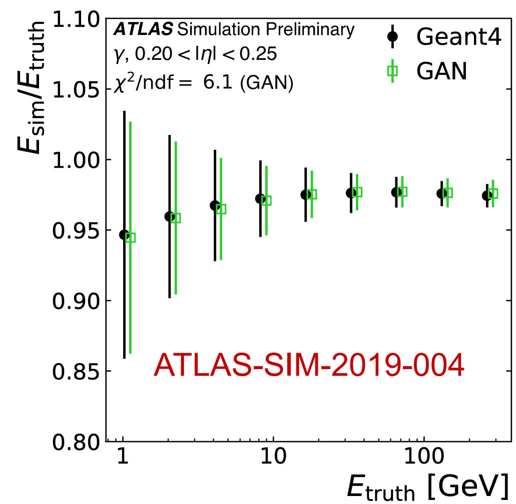


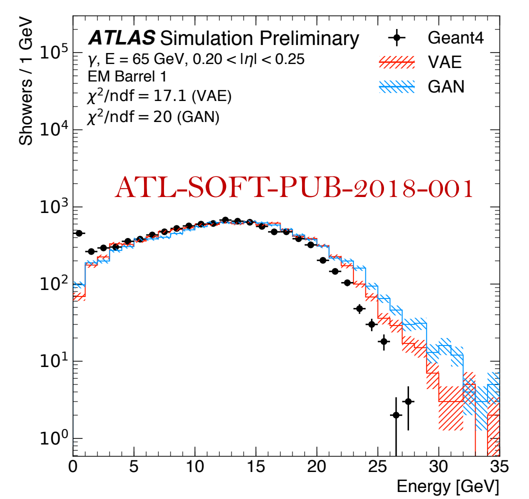
arXiv:1705.02355
arXiv:1701.05927



A. Maevskiy on behalf of LHCb [slides]



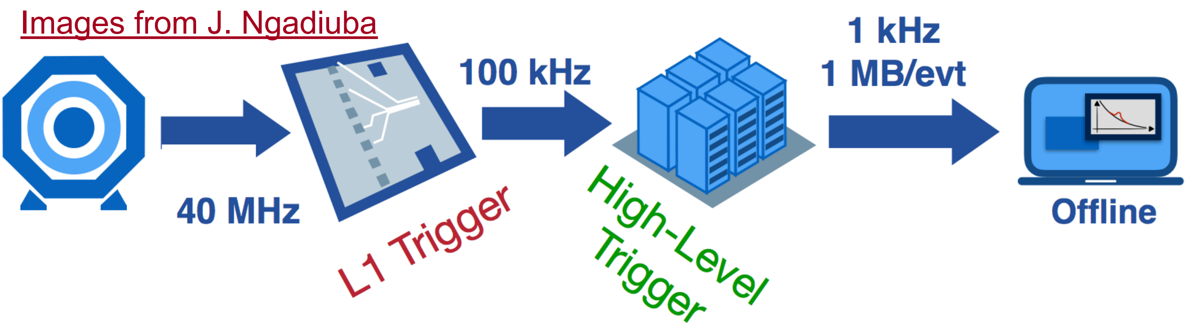




Fast Data Acquisition with ML on FPGA with HLS4ML

arXiv:1804.06913

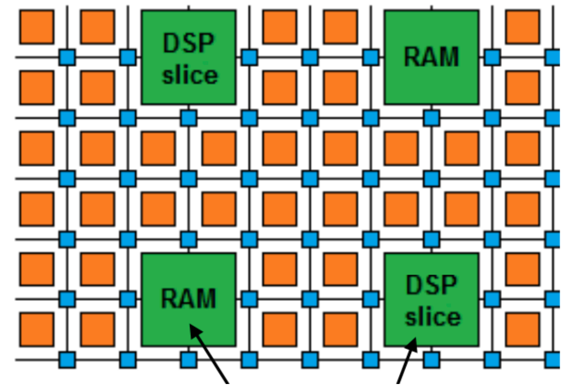
Images from J. Ngadiuba



Absorbs 100s Tb/s
Trigger decision to be made in O(μs)
Latencies require all-FPGA design

Computing farm for detailed analysis of the full event
Latency O(100 ms)

FPGA Diagram

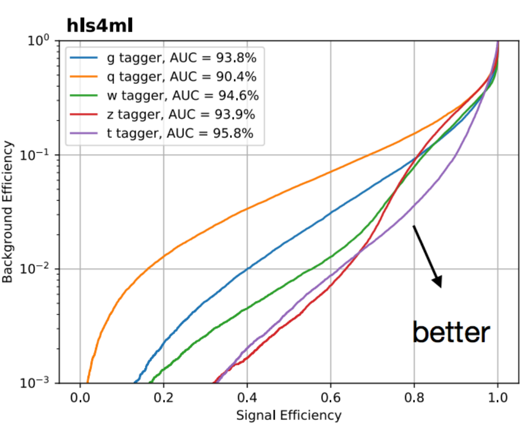


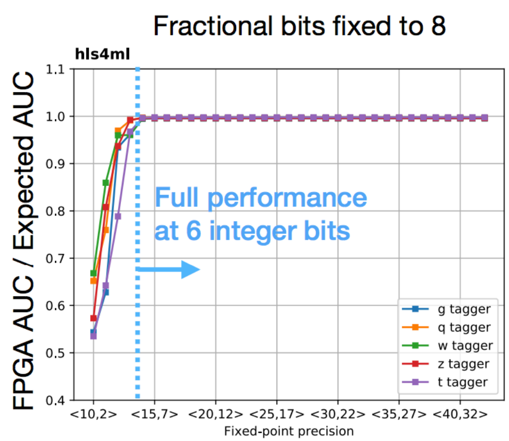
FPGAs are high speed, low power, and highly parallelizable

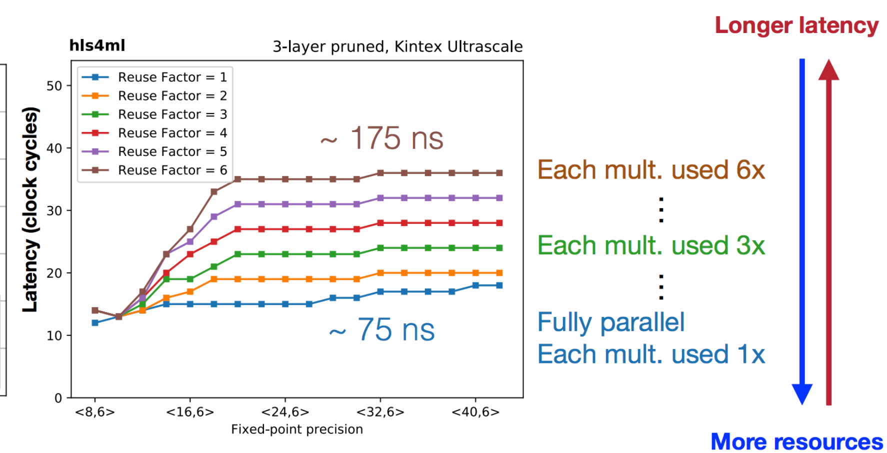
Dedicated SW needed to efficiently and effectively port ML algorithms to FPGA

Tuning resource usage, data precision, and model pruning needed to hit timing needs

Example: Boosted jet tagging







Conclusion

The structure of analysis pipeline is grounded in our detail physics domain knowledge

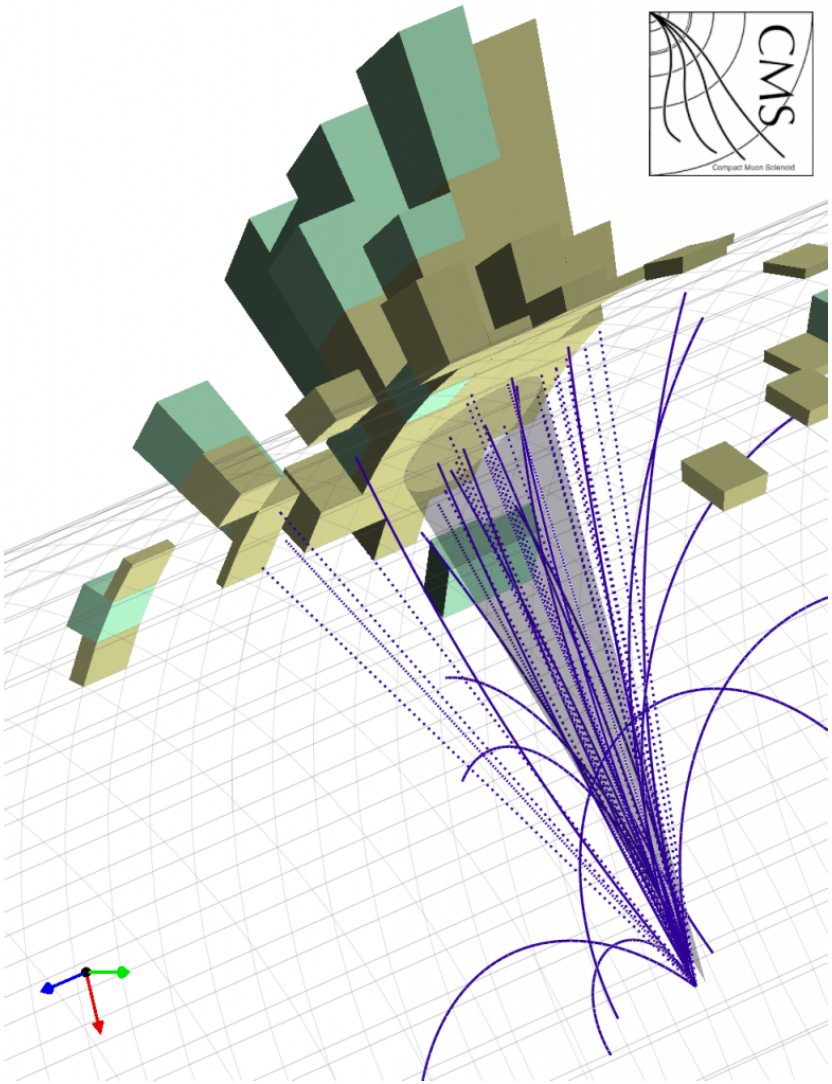
We can maintain our physics knowledge embedded in this pipeline while utilizing ML to help solve some of the intractable challenges

ML methods have shown strong performance improvements in reconstruction, and techniques to deal with key experimental challenges such as computational feasibility and systematic uncertainty mitigation are under study

New ideas in data driven search strategies, fast simulation, and triggering with ML may help expand the scope of our searches!

Backup

Reconstructing and Tagging Particles



- **Jet**: stream of particles produced by high energy quarks and gluons
  - Clustering algorithms used to find them

Jet identification = Classification

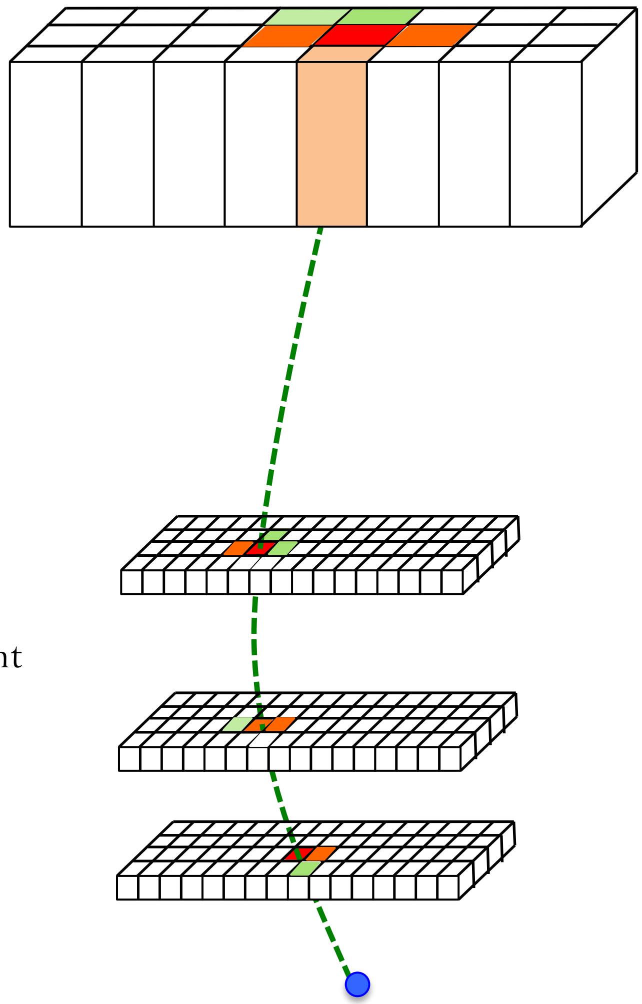
$$p(parent\ particle \mid jet\ cluster)$$

Energy estimation = Inference, Regression

$$p(E_{true}^{jet} \mid jet\ cluster)$$

Reconstructing and Tagging 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

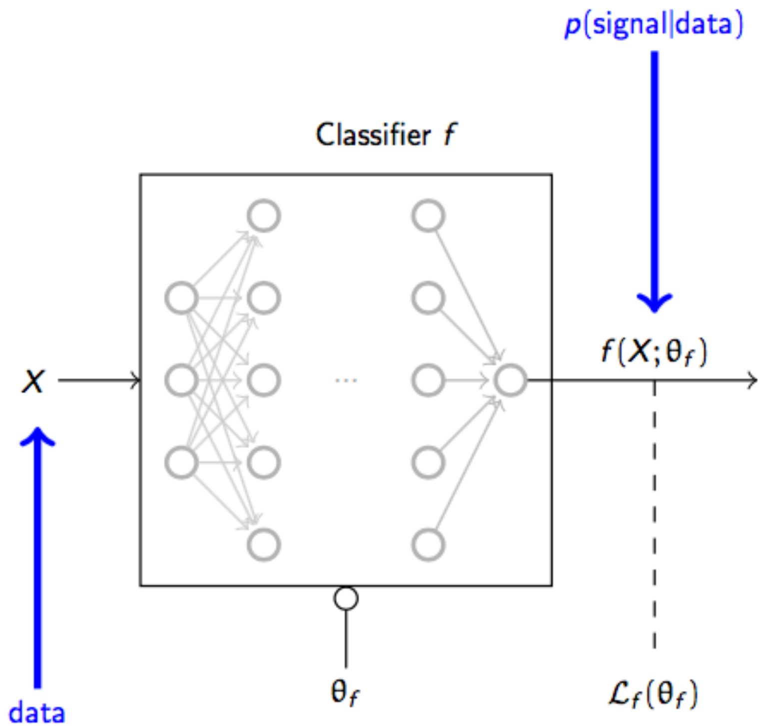
$$p(\text{electron} \mid \text{data})$$

Energy estimation = Inference, Regression

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

Adversarial Networks

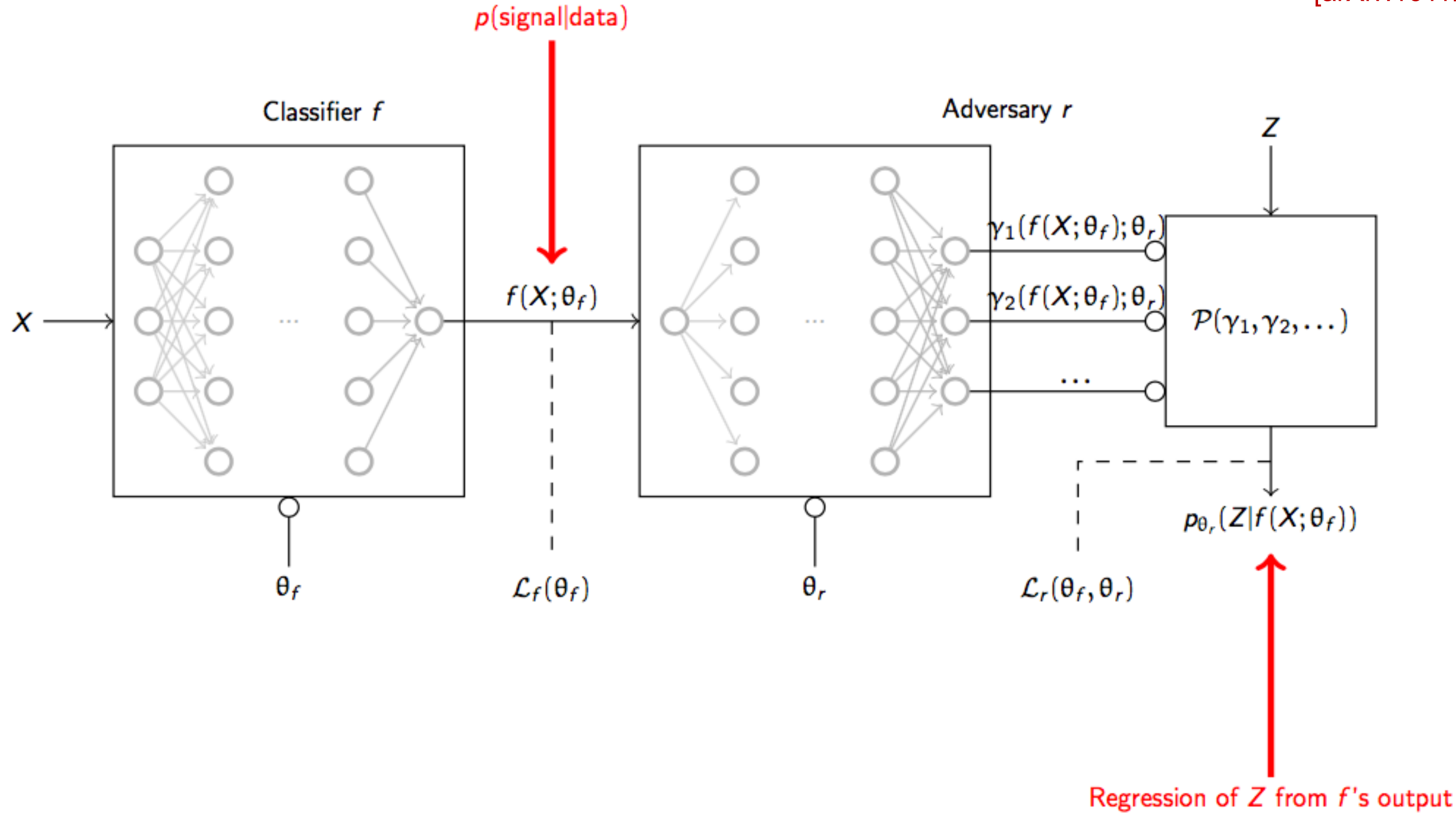
[arXiv:1611.01046]



Classifier built to solve problem at hand

Adversarial Networks

[arXiv:1611.01046]



Systematic uncertainty encoded as nuisance parameters, Z

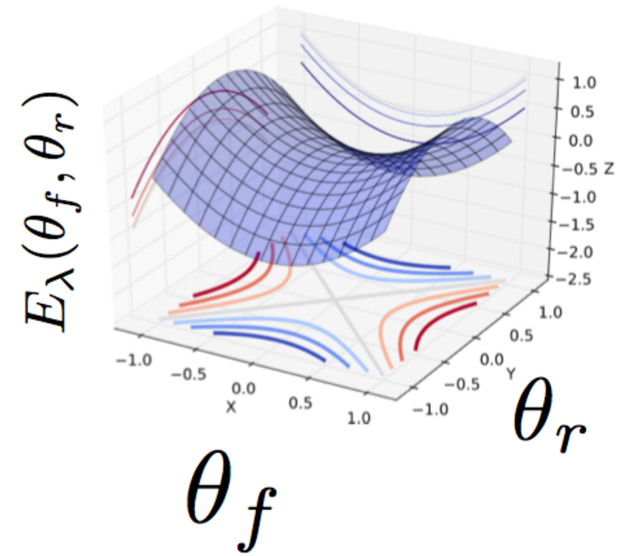
Adversary to predict the value of Z given classifier output

Adversarial Networks

[arXiv:1611.01046]

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

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



Loss encodes performance of classifier and adversary

- Classifier penalized when adversary does well at predicting Z

Hyper-parameter λ controls trade-off

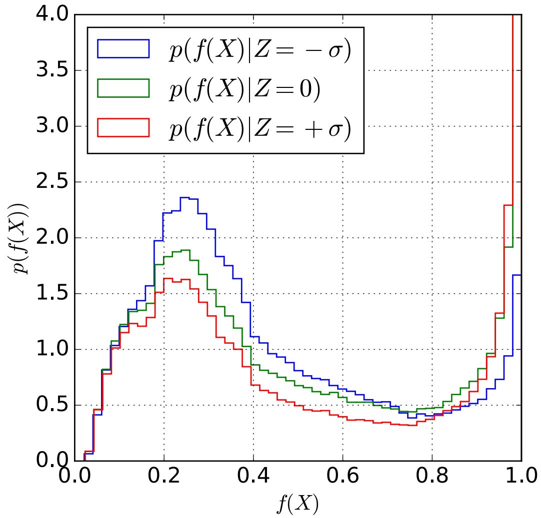
- Large λ enforces f(…) to be pivotal, e.g. robust to nuisance
- Small λ allows f(…) to be more optimal

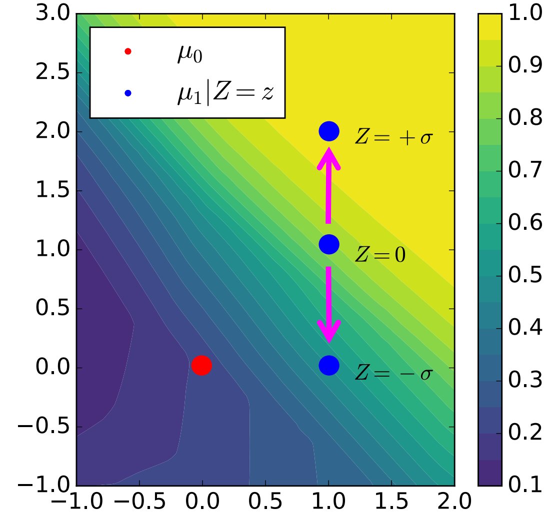
Learning to Pivot: Toy Example

2D example

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

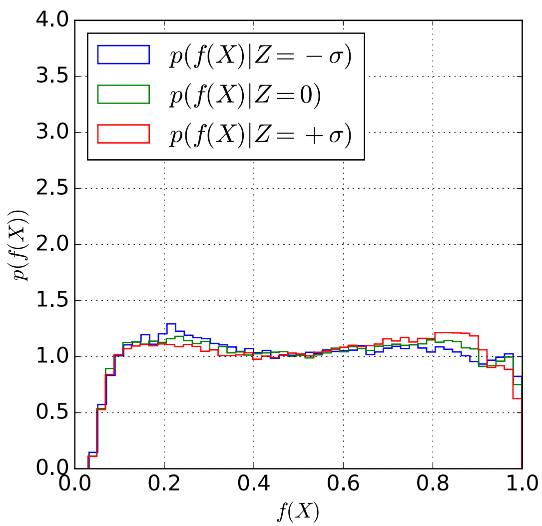
$$x \sim \mathcal{N}\left((1,1+Z), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \quad \text{when } Y = 1.$$

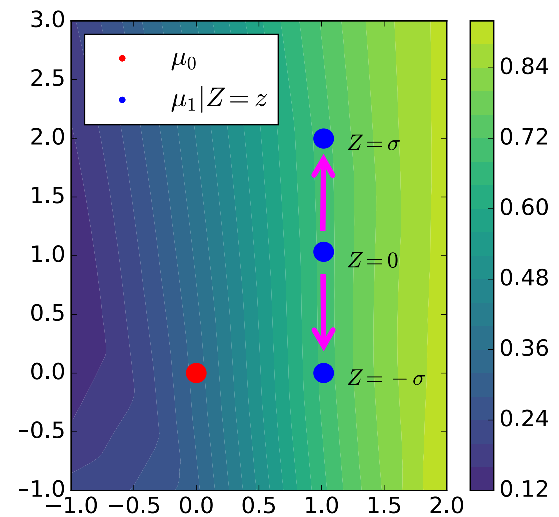




Without adversary (top)

large variations in network output with nuisance parameter

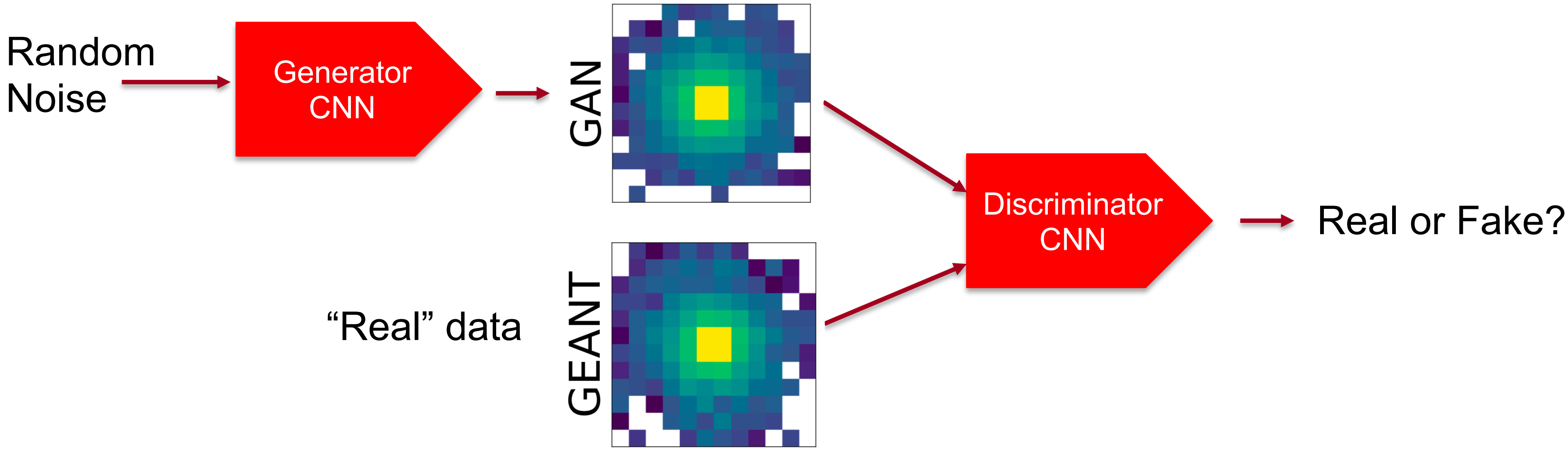




With adversary (bottom)

performance is independent!

Deep Generative Models for Simulation



Quickly growing literature

- 1701.05927, 1705.02355, 1807.01954
- ATL-SOFT-PUB-2018-001, ATLAS-SIM-2019-004
- Slides from G Khattak, F. Carminati, S. Vallecorsa
- Slides from A. Maevskiy, et. al. on behalf of LHCb
- Slides from T. Ferber for Belle II
- Slides from V. Belavin