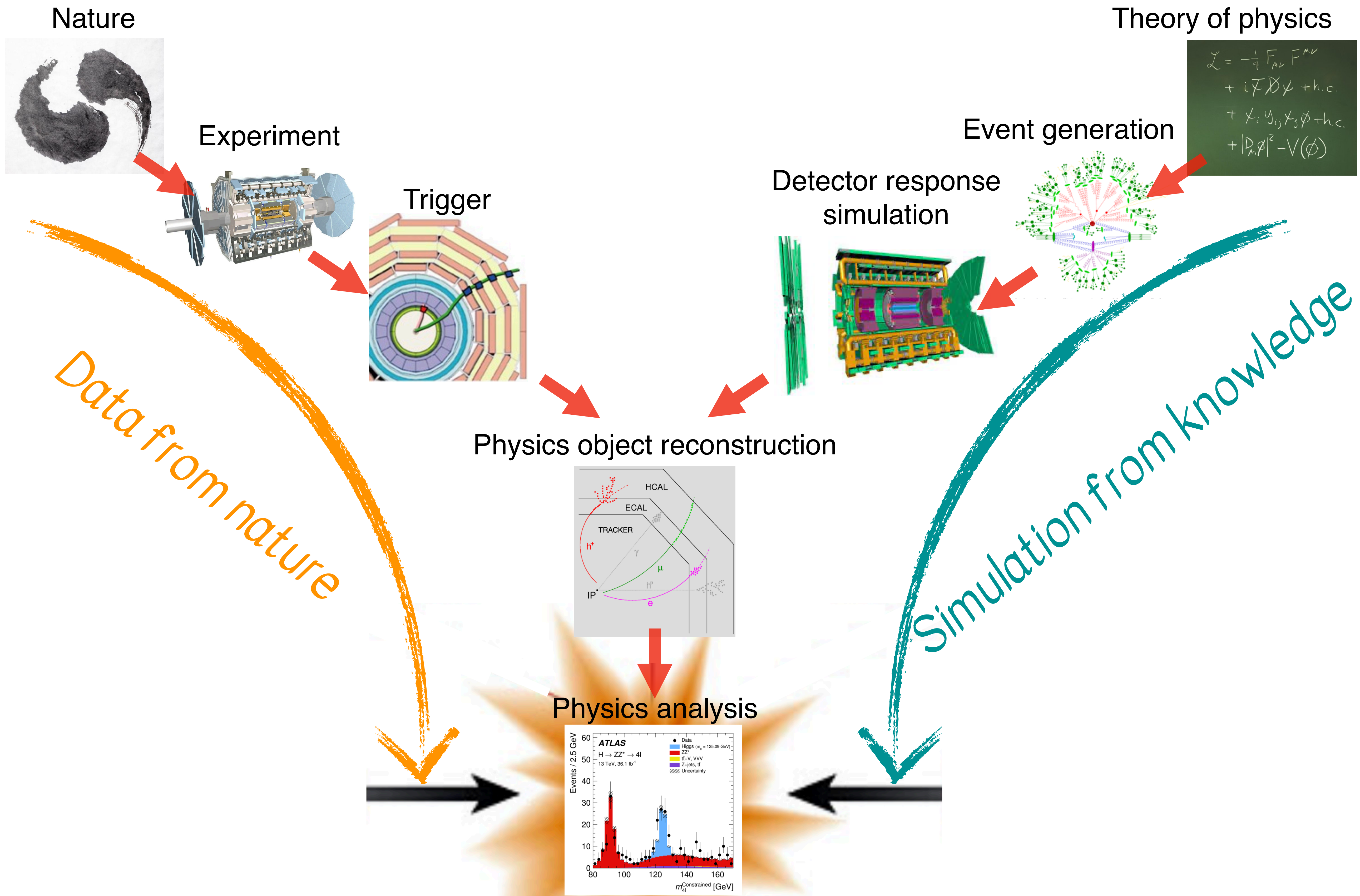
Overview of ML studies at CMS and ATLAS

Rui Zhang (张 瑞)

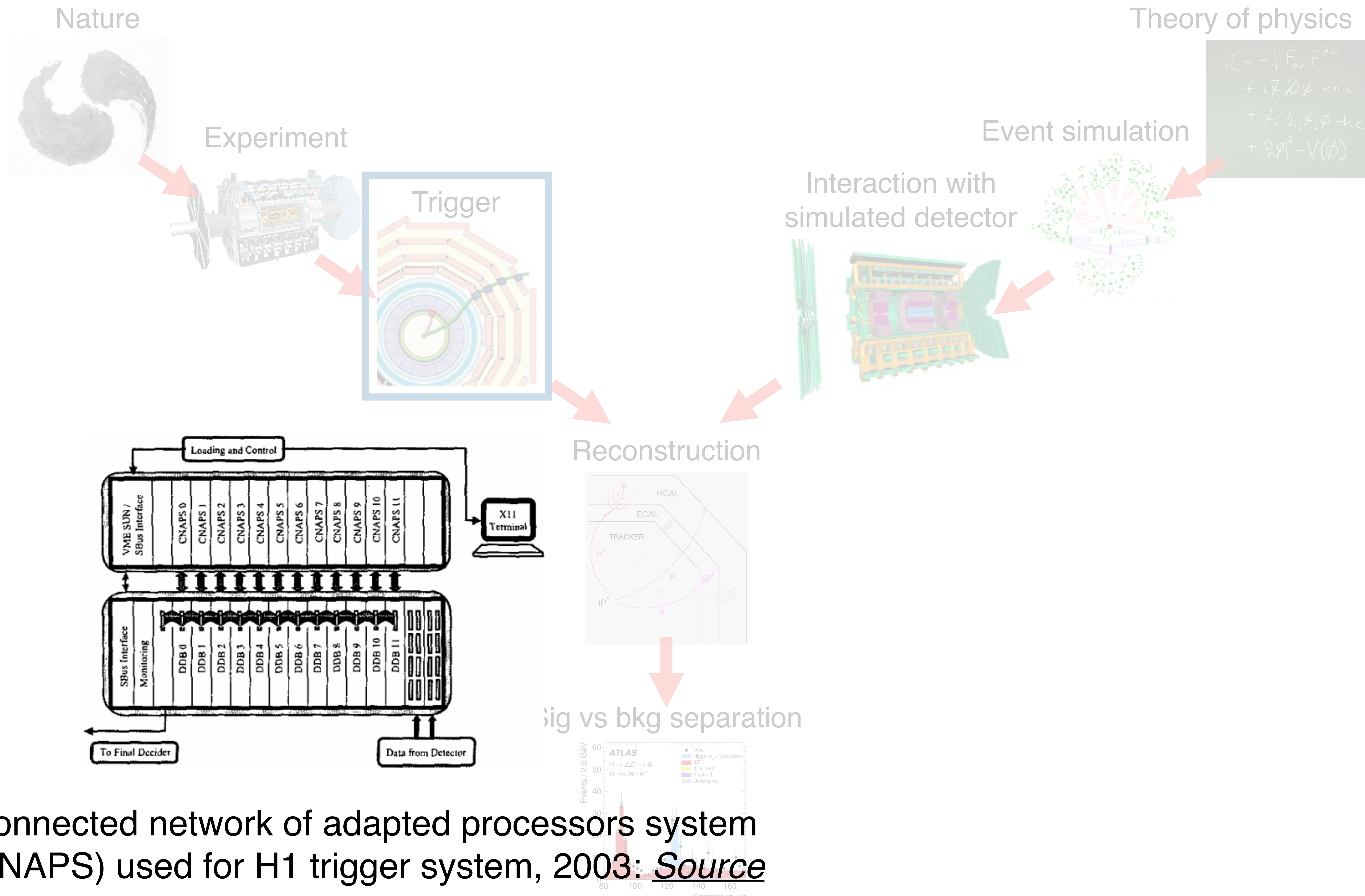
University of Wisconsin-Madison, Wisconsin

06 Aug 2024, Changchun, China

A typical LHC Physics Analysis Workflow



ML is an old friend of HEP



Connected network of adapted processors system (CNAPS) used for H1 trigger system, 2003: *Source*

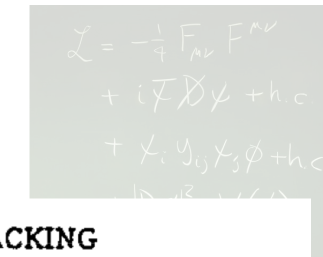
ML is an old friend of HEP

Nature

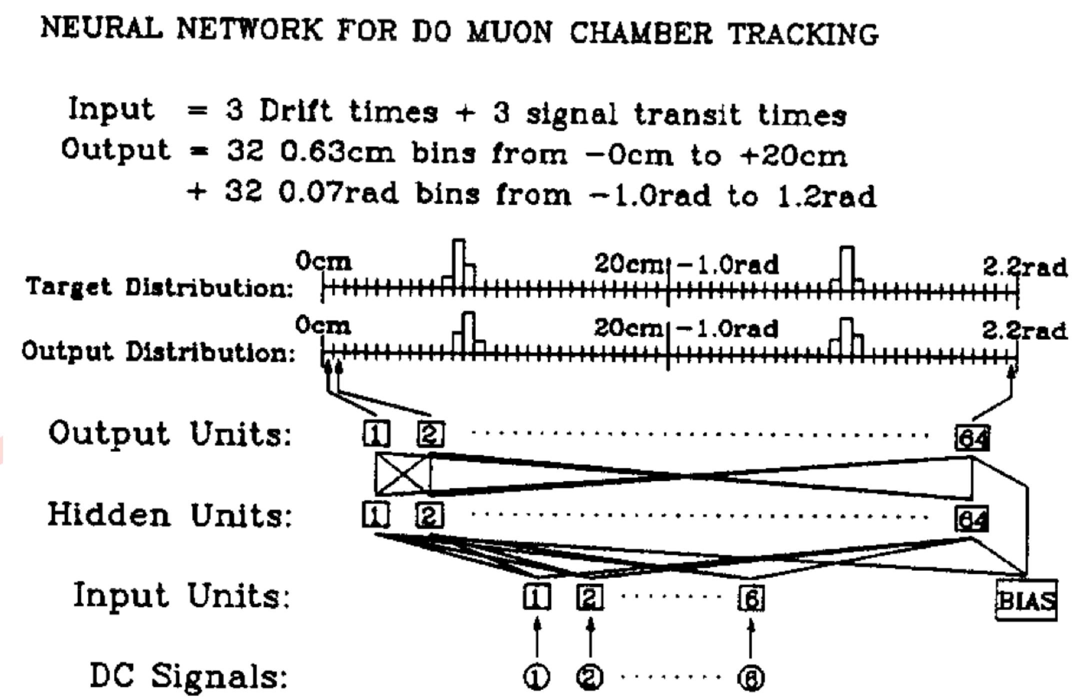
Experiment

Trigger

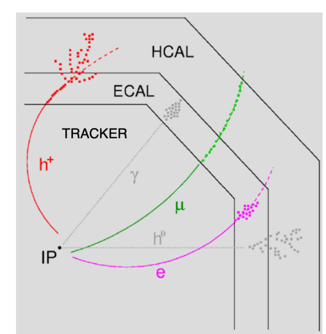
Theory of physics



Event simulation



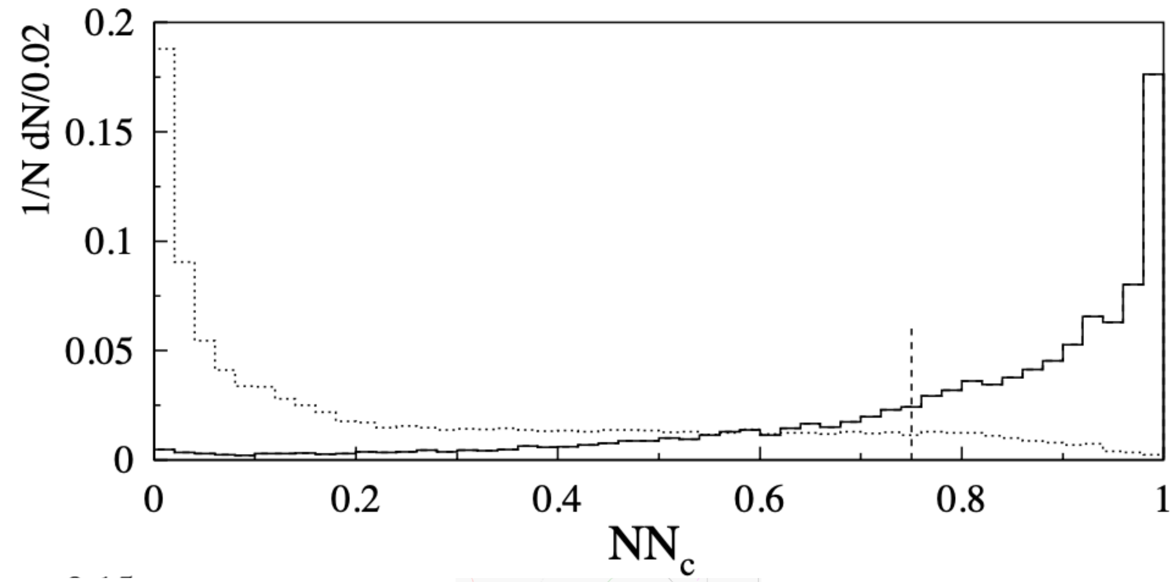
Reconstruction



Primary vertexing based on the fired wires at E735, Fermilab, 1991: *Source*

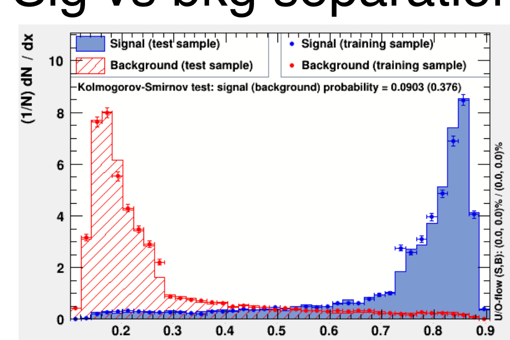
Sig vs bkg separation

ML is an old friend of HEP



Selection of b hadrons at ALEPH, 1999:
Source

Sig vs bkg separation



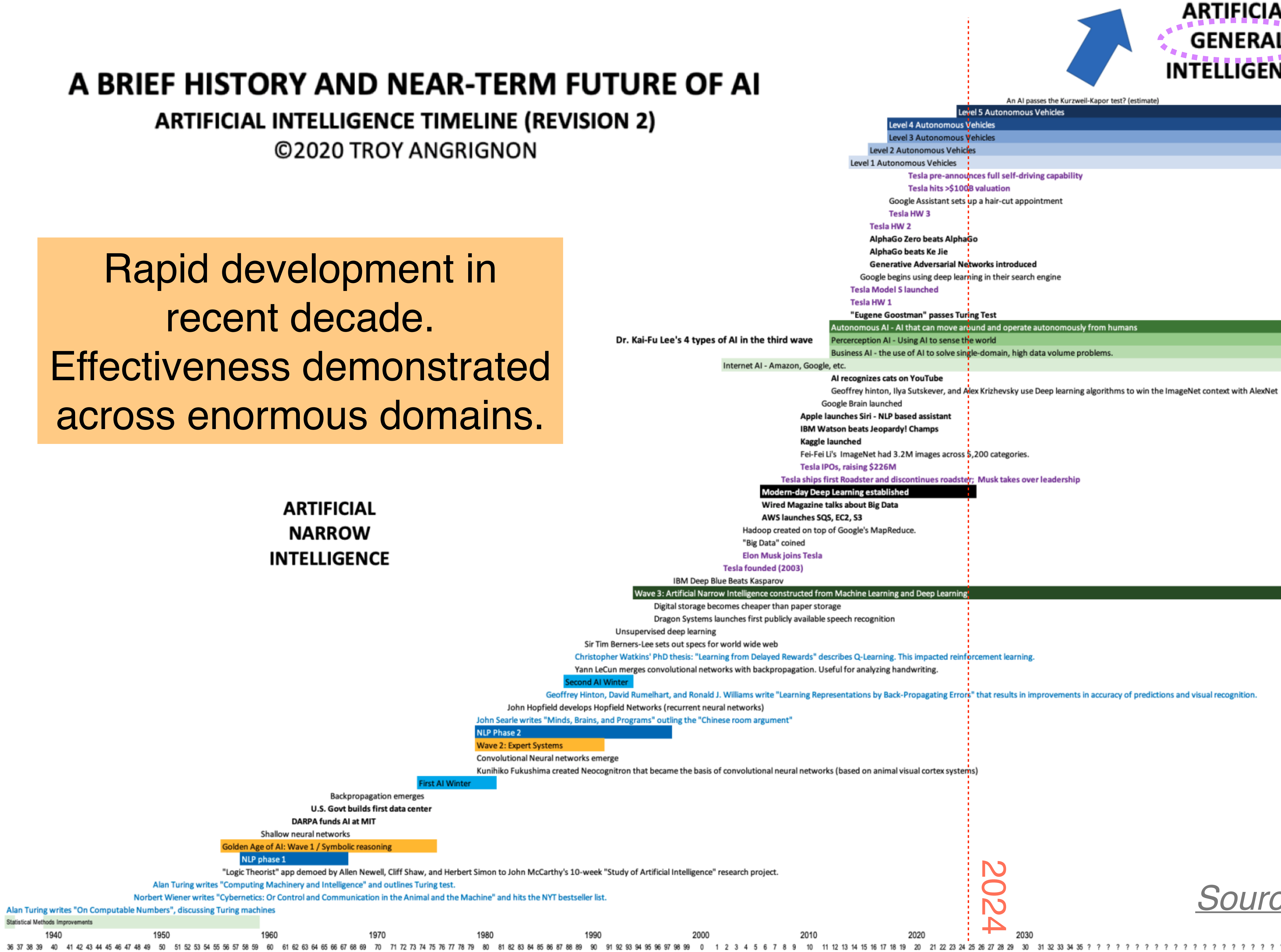
A lot more, see reviews in 1993 and in 1999

A BRIEF HISTORY AND NEAR-TERM FUTURE OF AI

ARTIFICIAL INTELLIGENCE TIMELINE (REVISION 2)

©2020 TROY ANGRIGNON

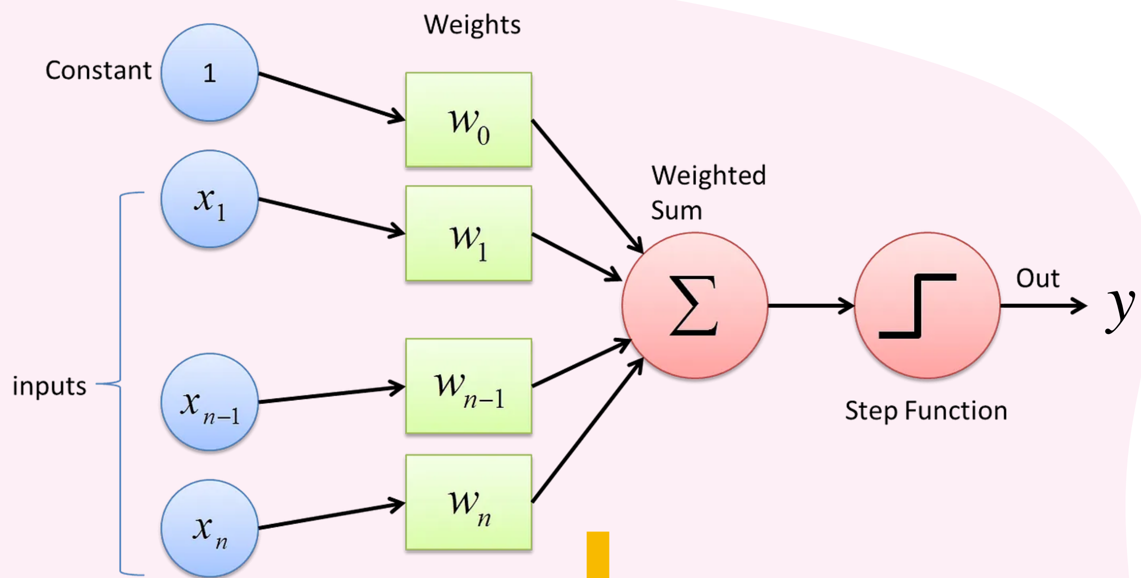
Rapid development in recent decade. Effectiveness demonstrated across enormous domains.



[Source](#)

ML is not a magic

It's built upon linear algebra and information theory



$$\sigma\left(\Sigma \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = y$$

$$y = f_W(x)$$

Neural network is a function that maps input to output; "universal approximation theorem"

Learning procedure is to compress the input to output.

$$\begin{aligned} y_1 &= f_1(x) \\ y_2 &= f_2(x) \\ &\vdots \\ y_n &= f_n(x) \end{aligned}$$

Which function is close to truth?

Need to quantify "similarity" between y_i and y_{truth}.

- Both are distributions (PDF)
- Also known as "loss" => min(Loss(y_i, y_{truth}))

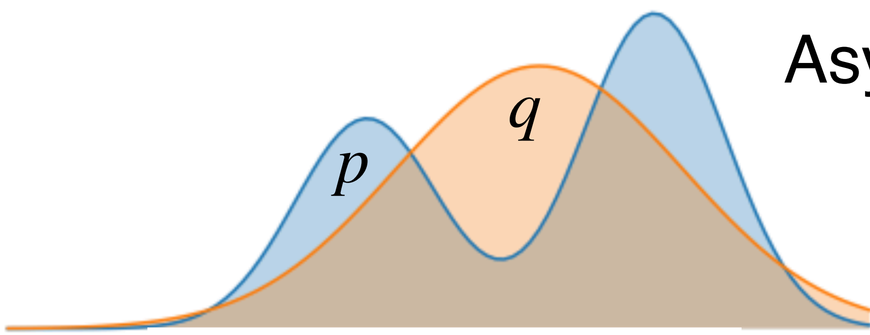
Information theory offers measures for quantifying similarity

- Entropy: disorder of 1 PDF
- Divergence: disorder between 2 PDFs

ML is not a magic: divergences

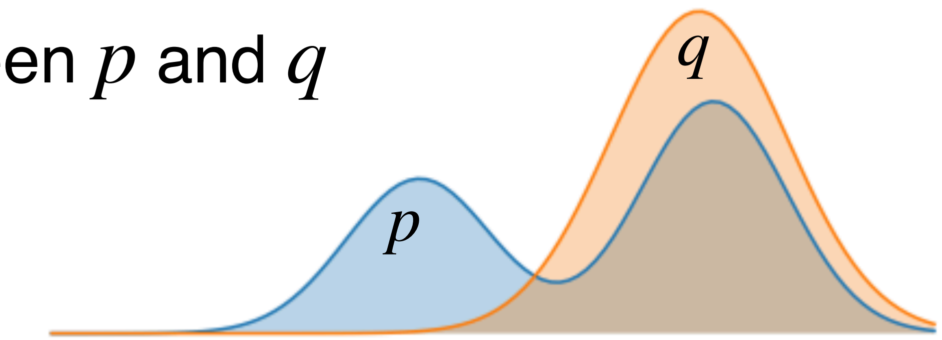
Divergence is a measure of statistical distance between two distributions.

Most popular one: Kullback-Leibler (KL) divergence: $D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_{\mathrm{x}\sim P}\left[\log \frac{P(x)}{Q(x)}\right]$



$q^* = \mathrm{argmin}_q D_{\mathrm{KL}}(p\|q)$

Asymmetric between p and q



$q^* = \mathrm{argmin}_q D_{\mathrm{KL}}(q\|p)$

Wish all events in p will be found in q

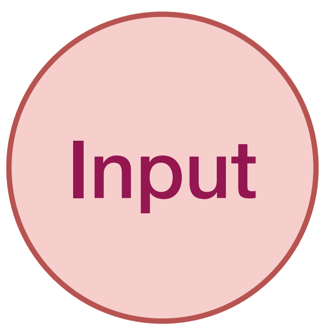
Wish all events in q will be found in p

- Jensen-Shannon Divergence (JSD)
- Wasserstein Distance (Earth Mover's Distance)
- Total Variation Distance (TV Distance)
- Bhattacharyya Distance
- Hellinger Distance
- f-Devergence
- Rényi divergence
- …

Choice of divergence can impact the accuracy and efficiency of ML.

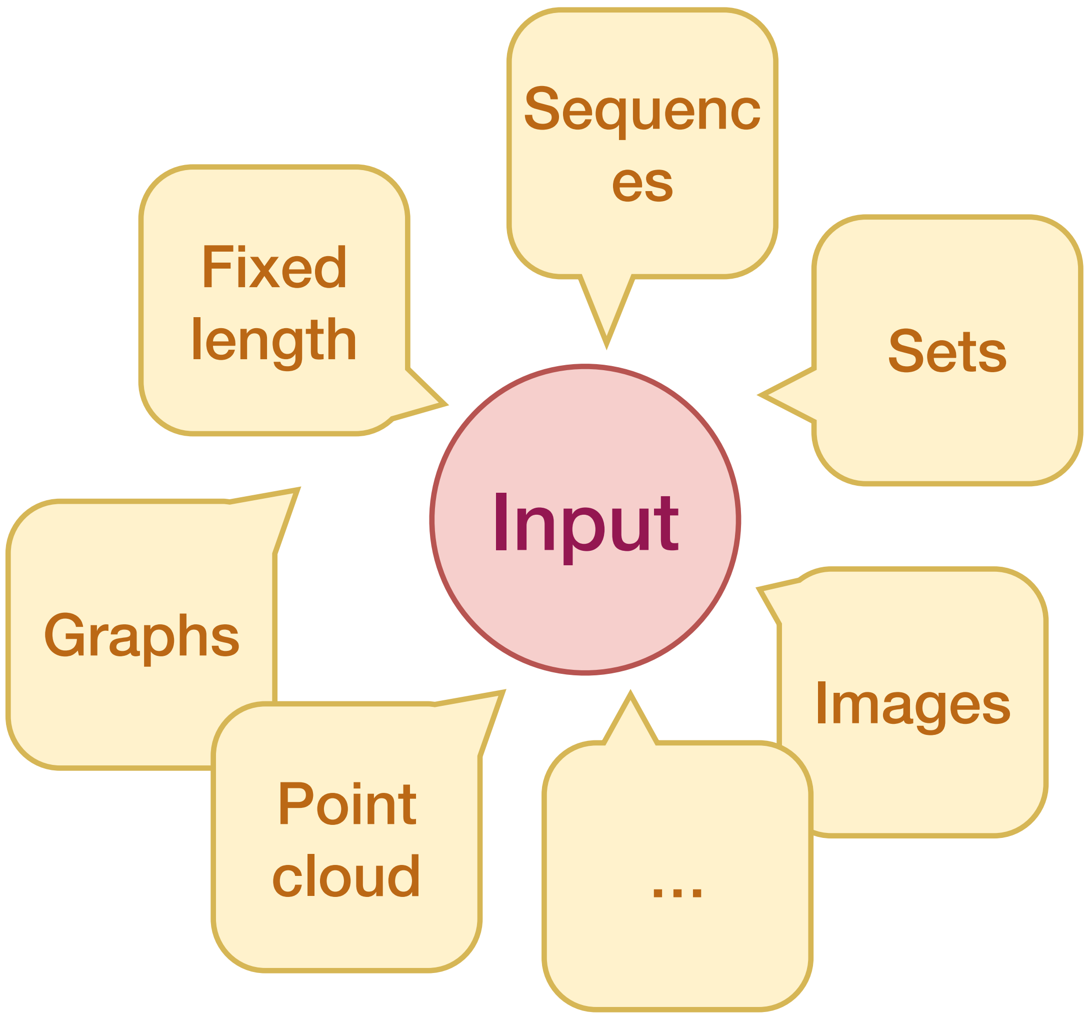
ML is a Tool for HEP

- Step 1: how to represent data
- Step 2: set up the learning task

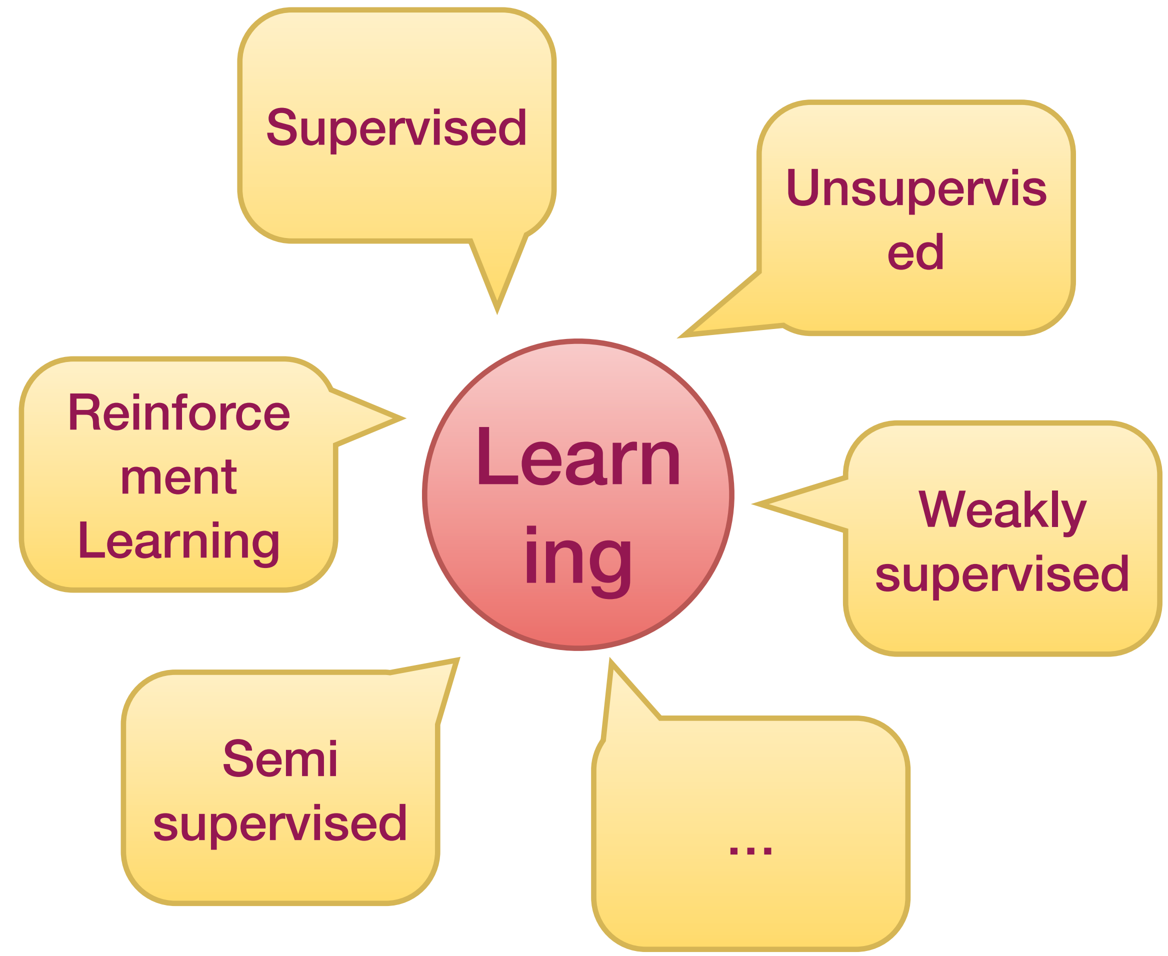


ML is a Tool for HEP

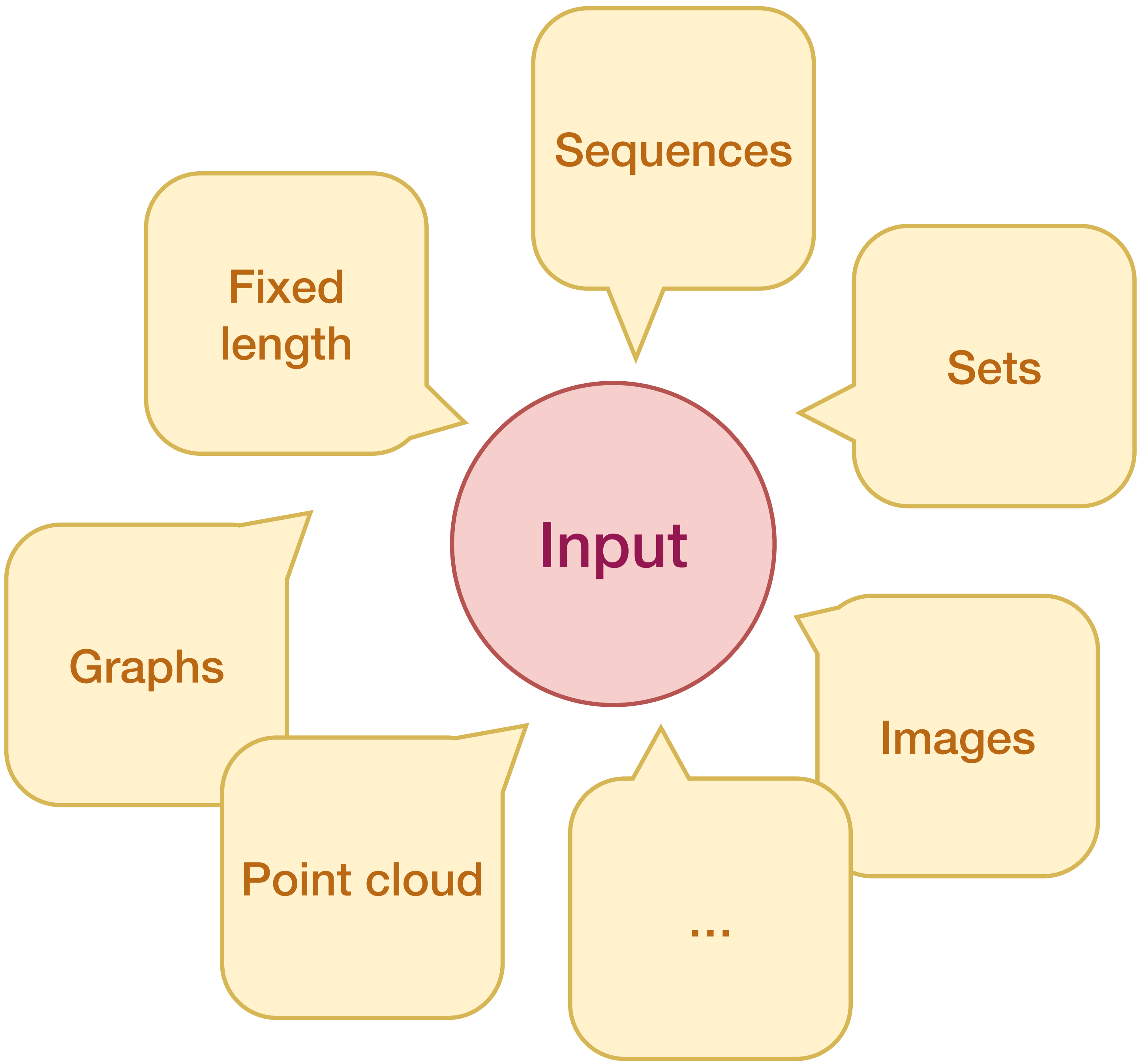
- Step 1: how to represent data



- Step 2: set up the learning task



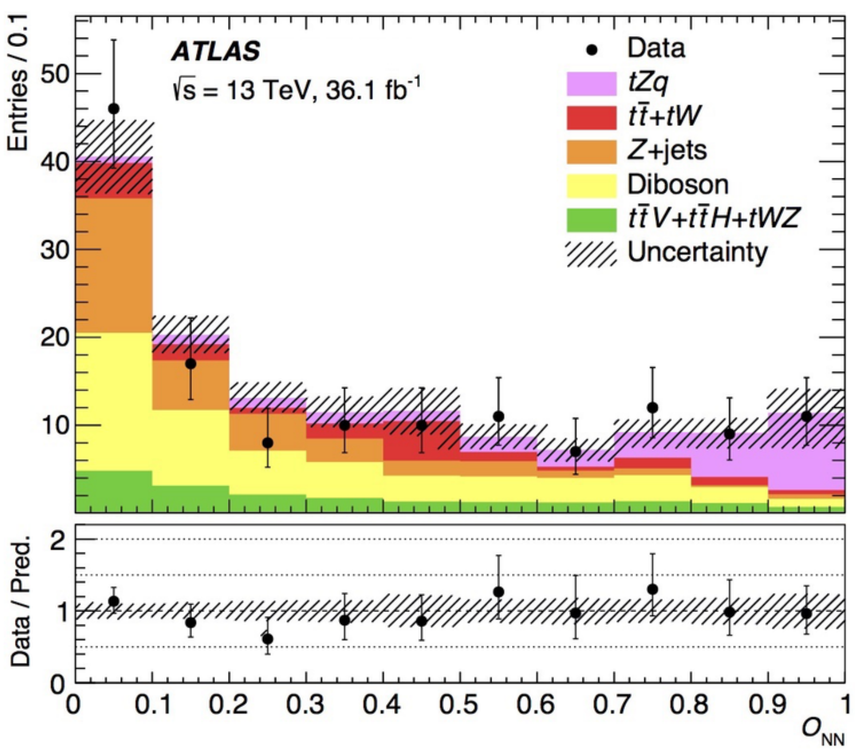
Step 1: how to represent data?



1.1 Input has fixed length

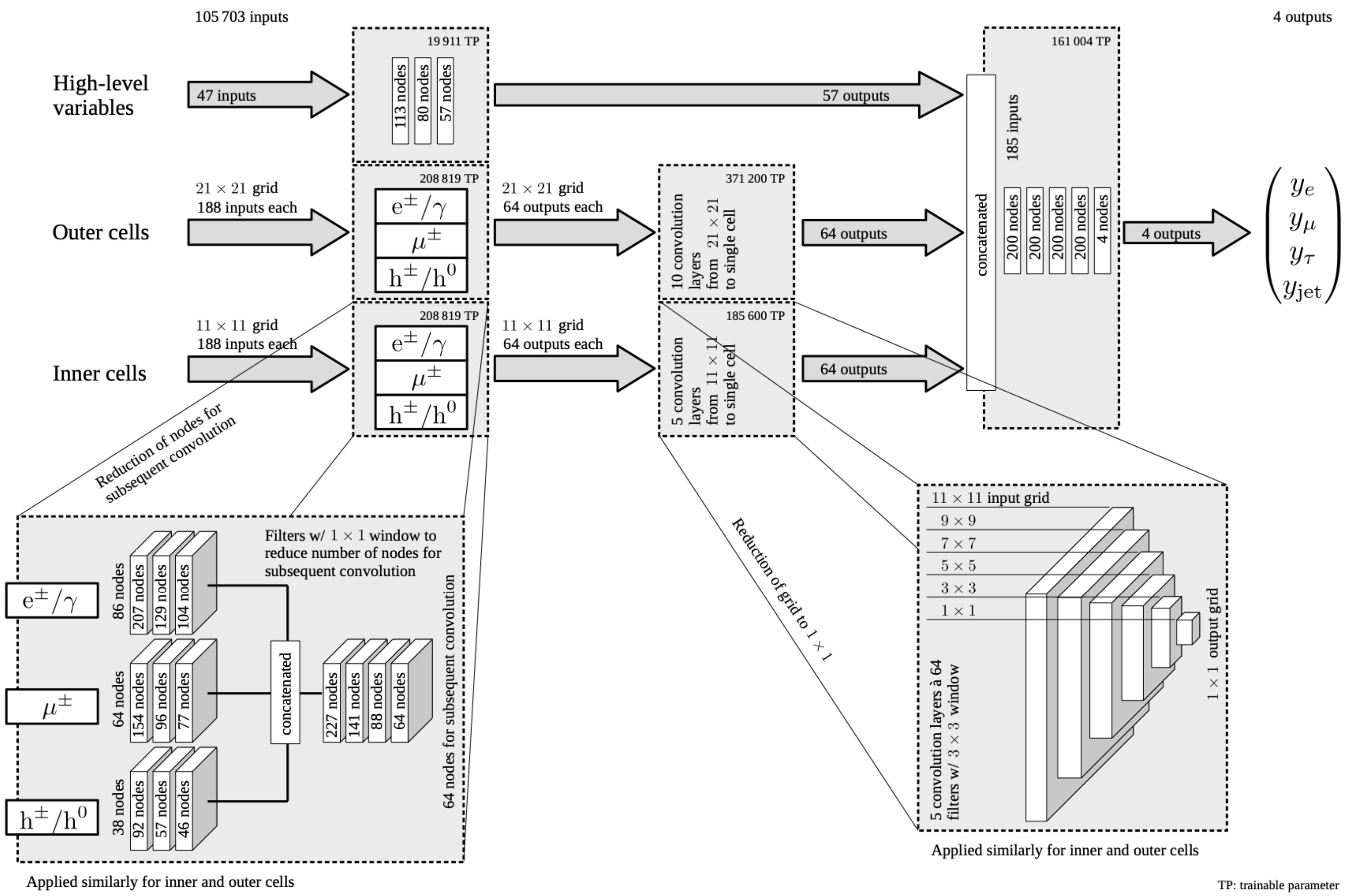
- Decide in advance variable list for training, then train a deep neural network / BDT

A typical signal extraction using NN



[Phys. Lett. B 780 (2018) 557]

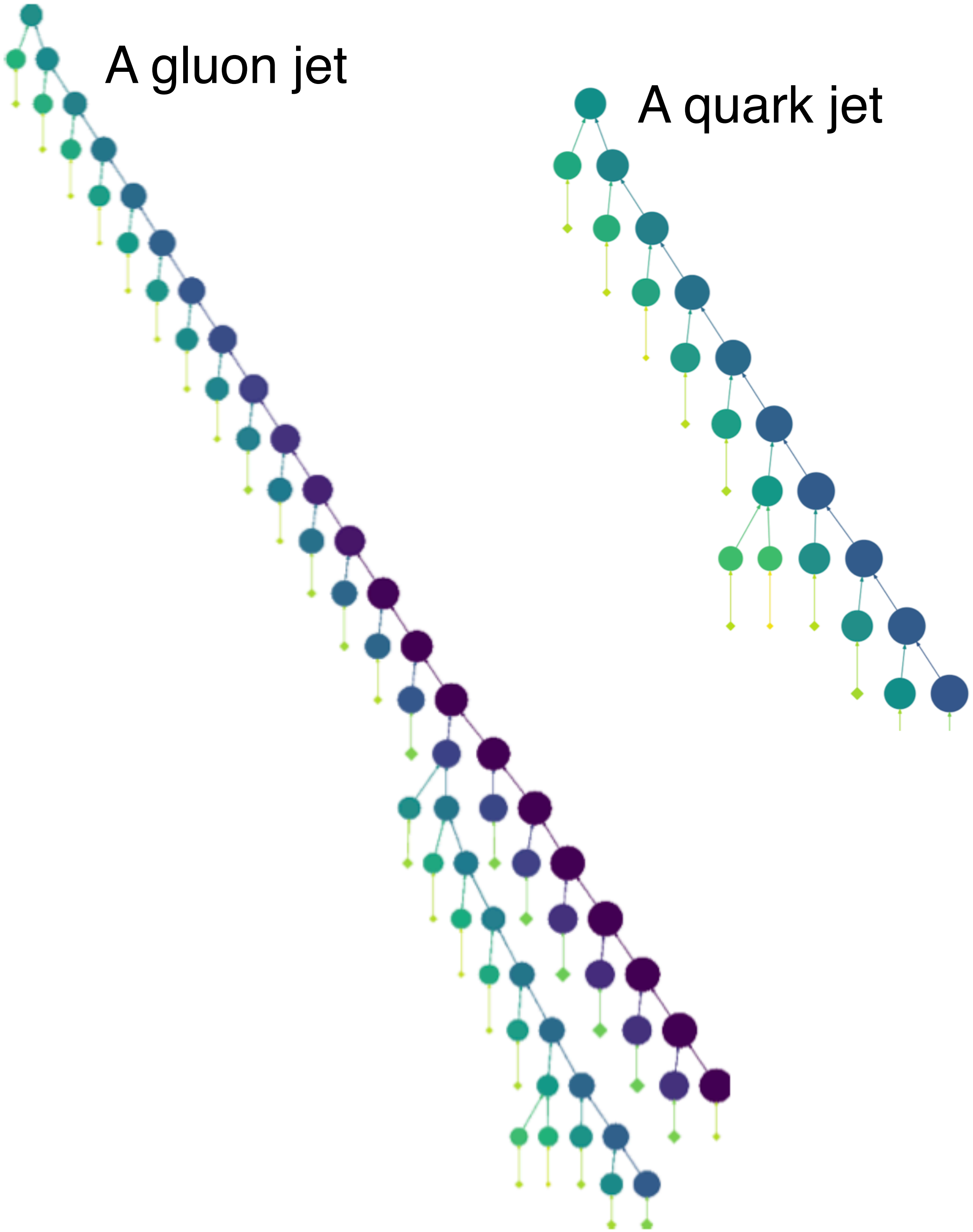
CMS tau ID deep network



[JINST 17 (2022) P07023]

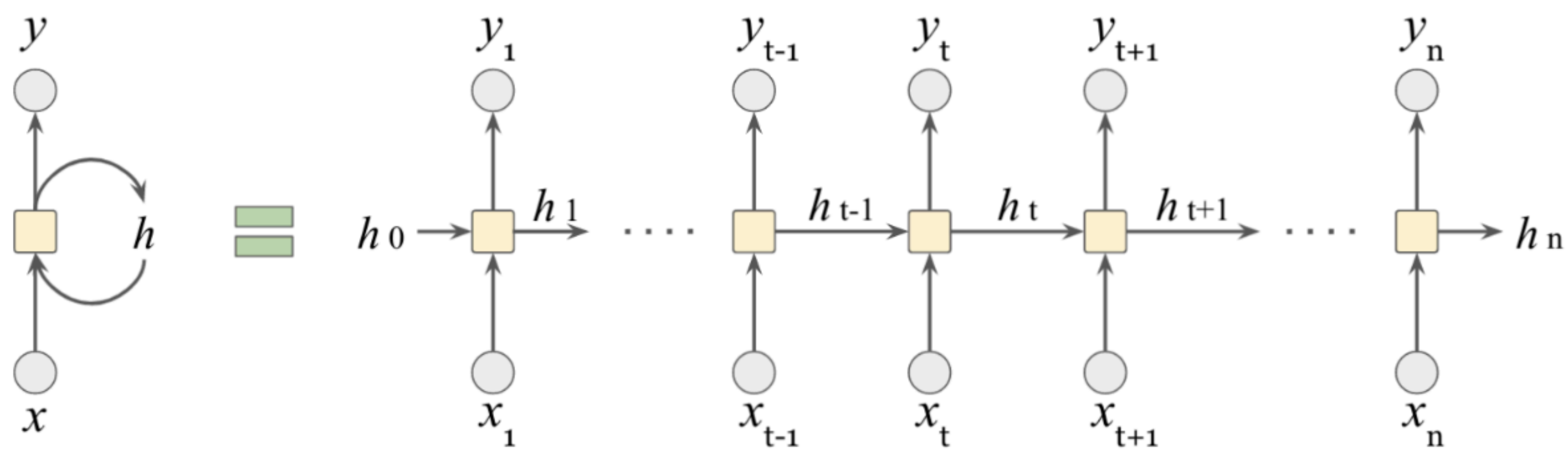
1.2 Input as sequences

- In some situations, fixed length is not suitable
  - e.g. Jets contain a variable number of particles
  - Recurrent Neural Networks shows great performance for Natural Language Processing tasks
  - Information across the entire sequence can be accumulated and used

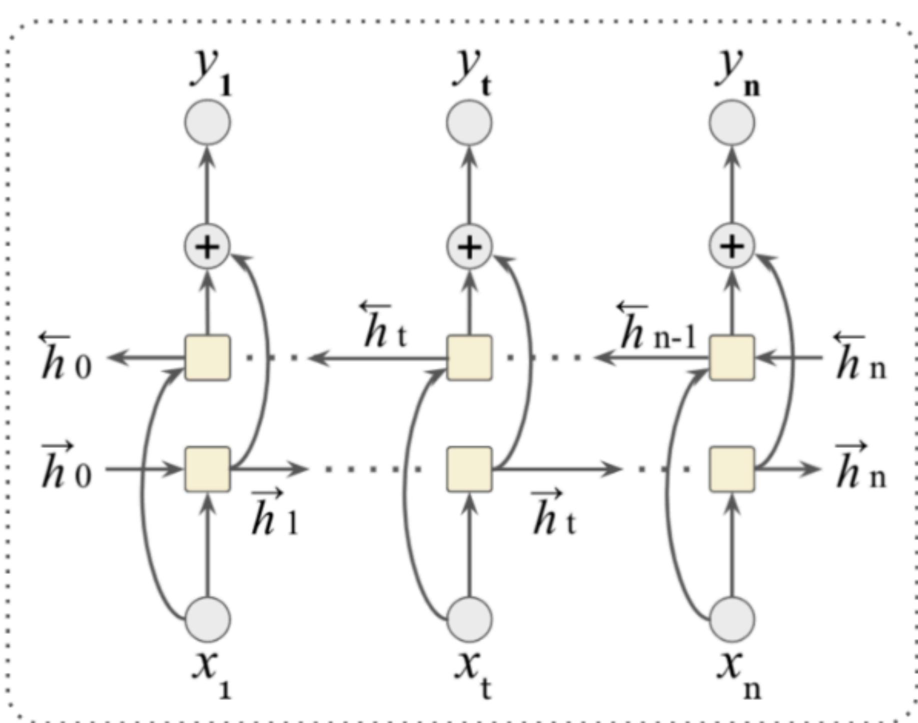


[Recursive Neural Networks in Quark/Gluon Tagging]

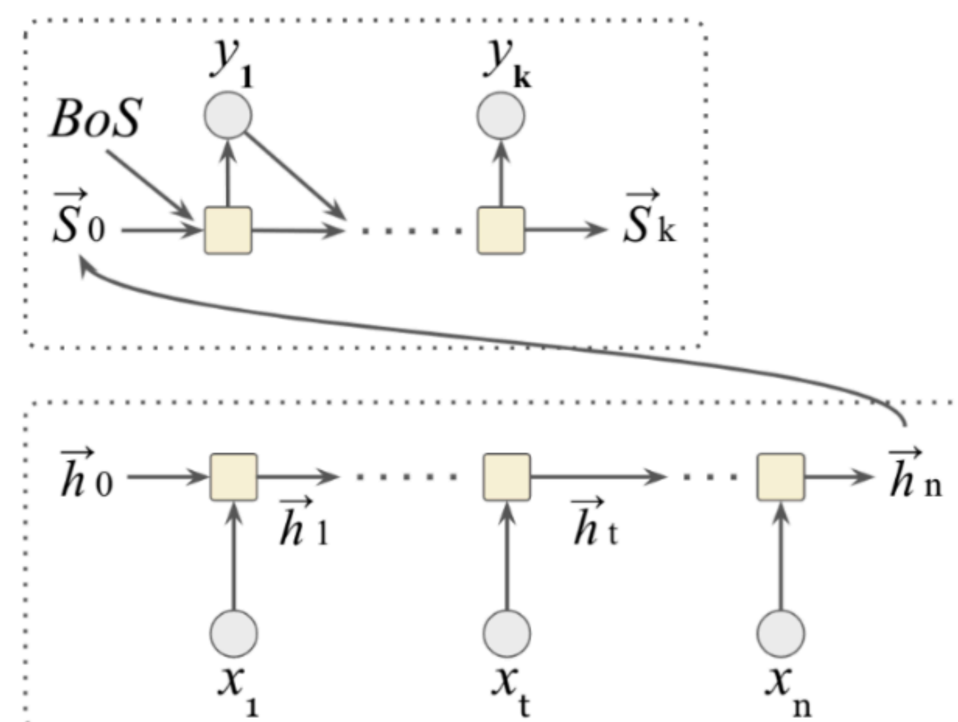
Recurrent Neural Networks



$$h_t = g_h(h_{t-1}, x_t, \theta)$$

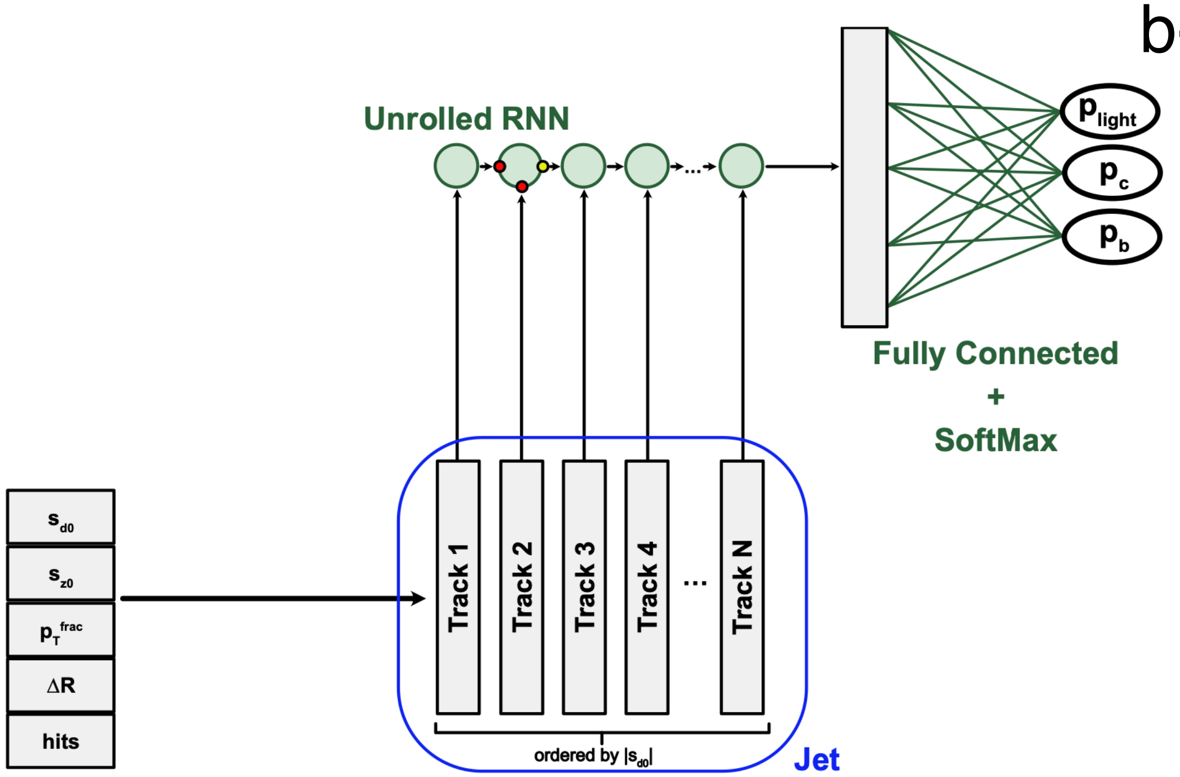


Bi-directional RNN

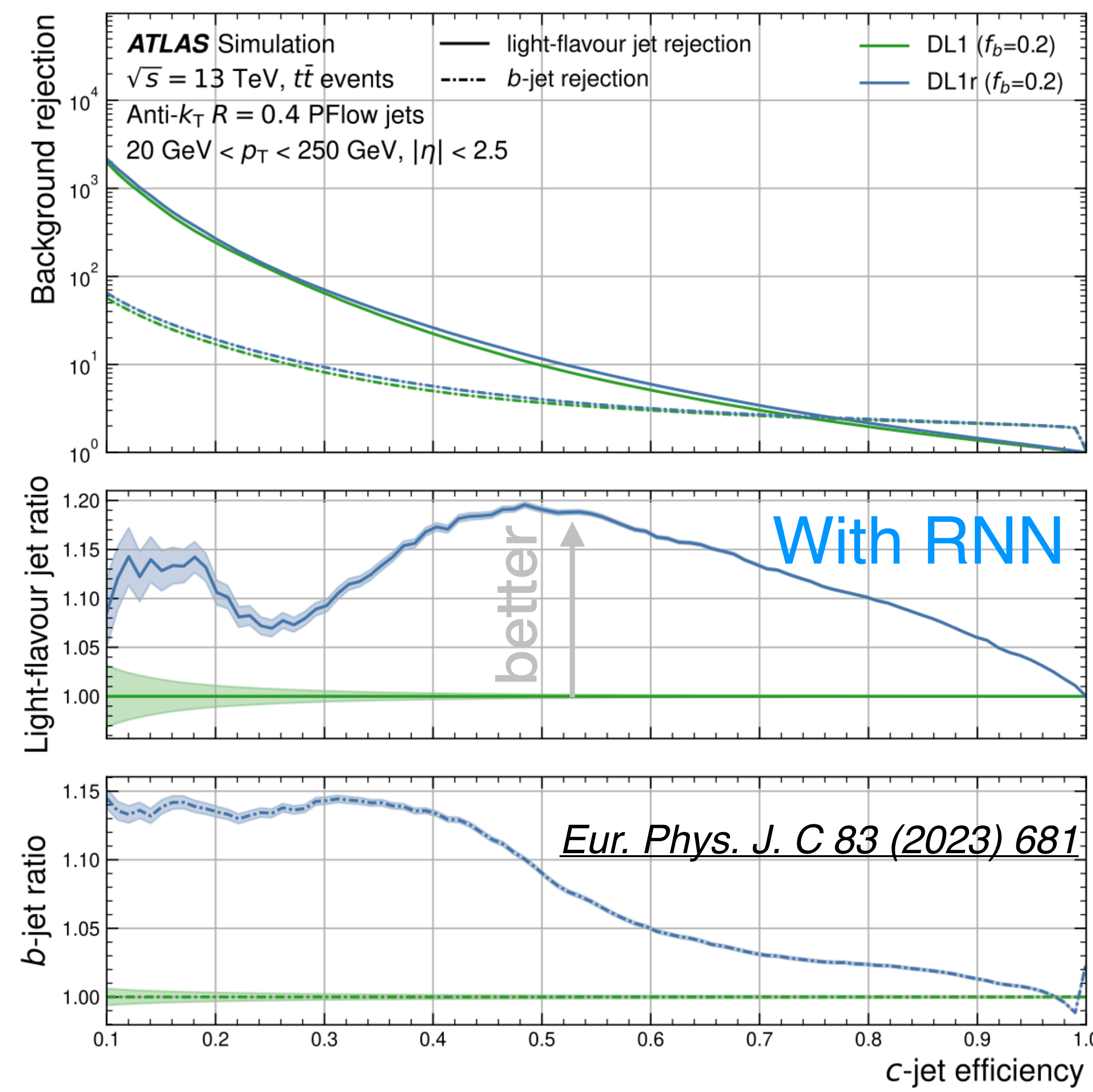


RNN Encoder-Decoder

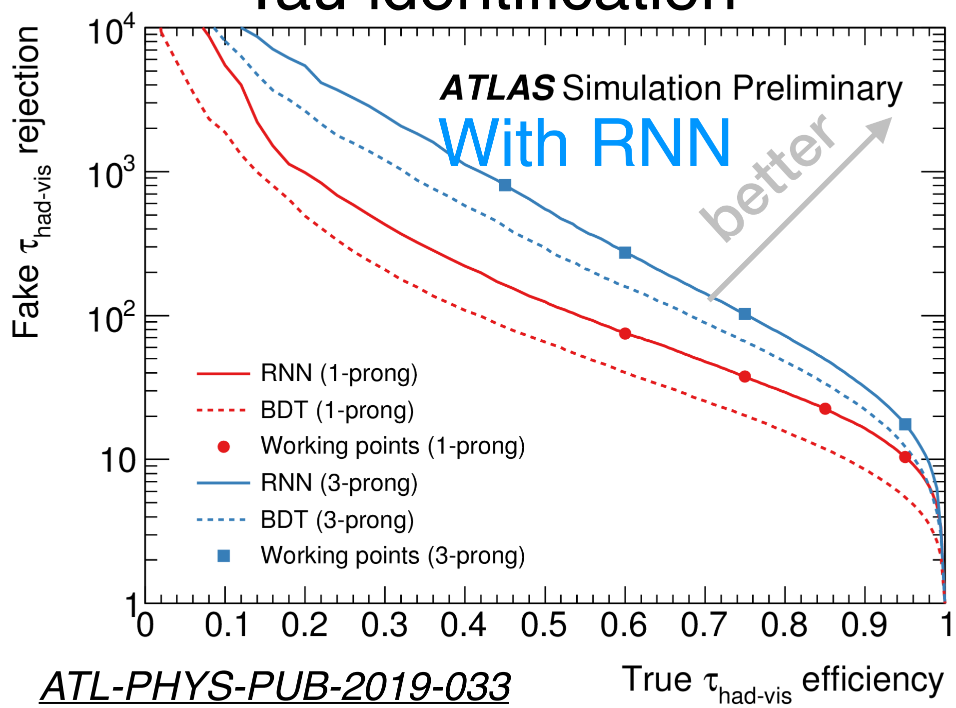
RNN application



b-tagging



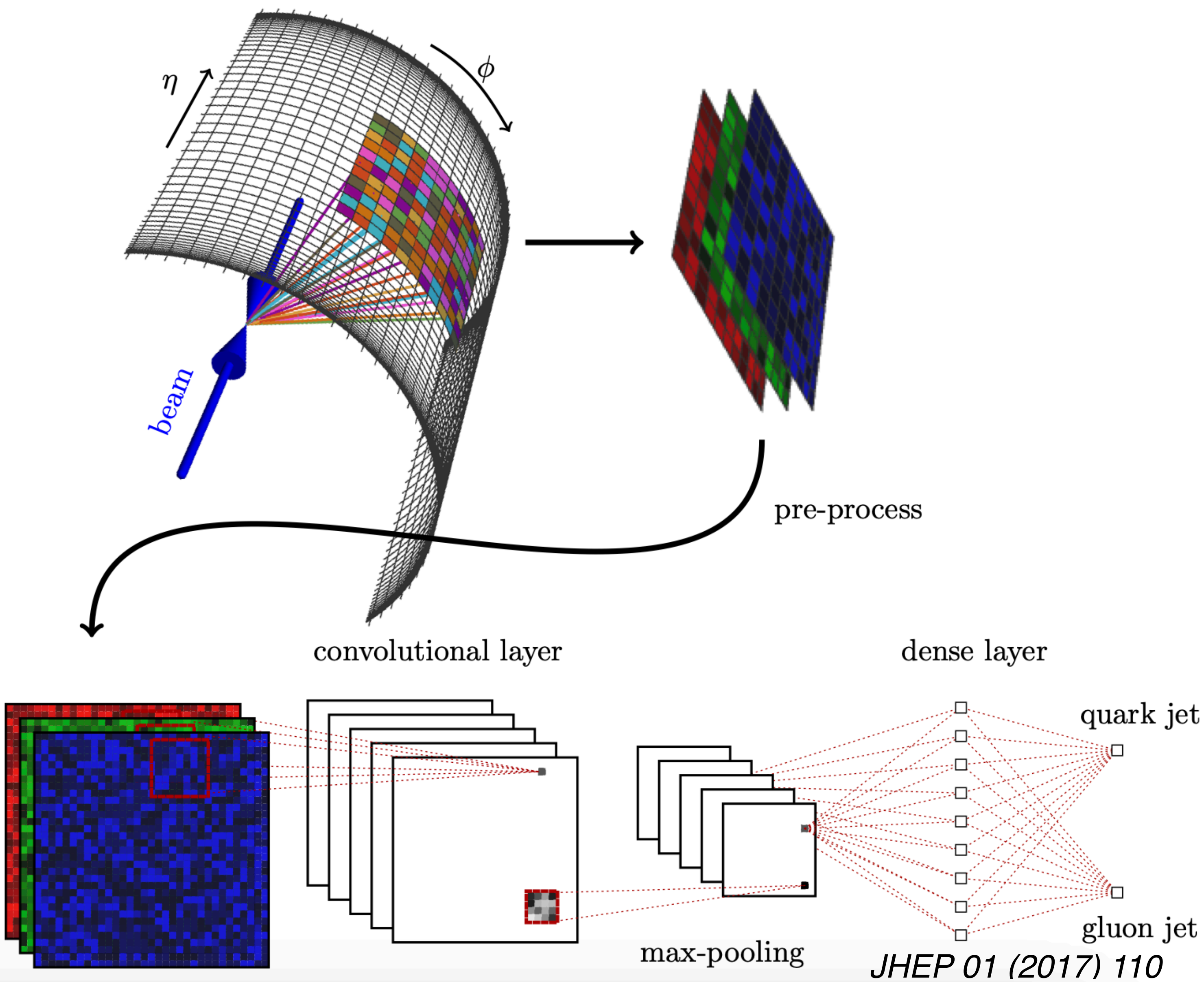
Tau identification



ATL-PHYS-PUB-2019-033

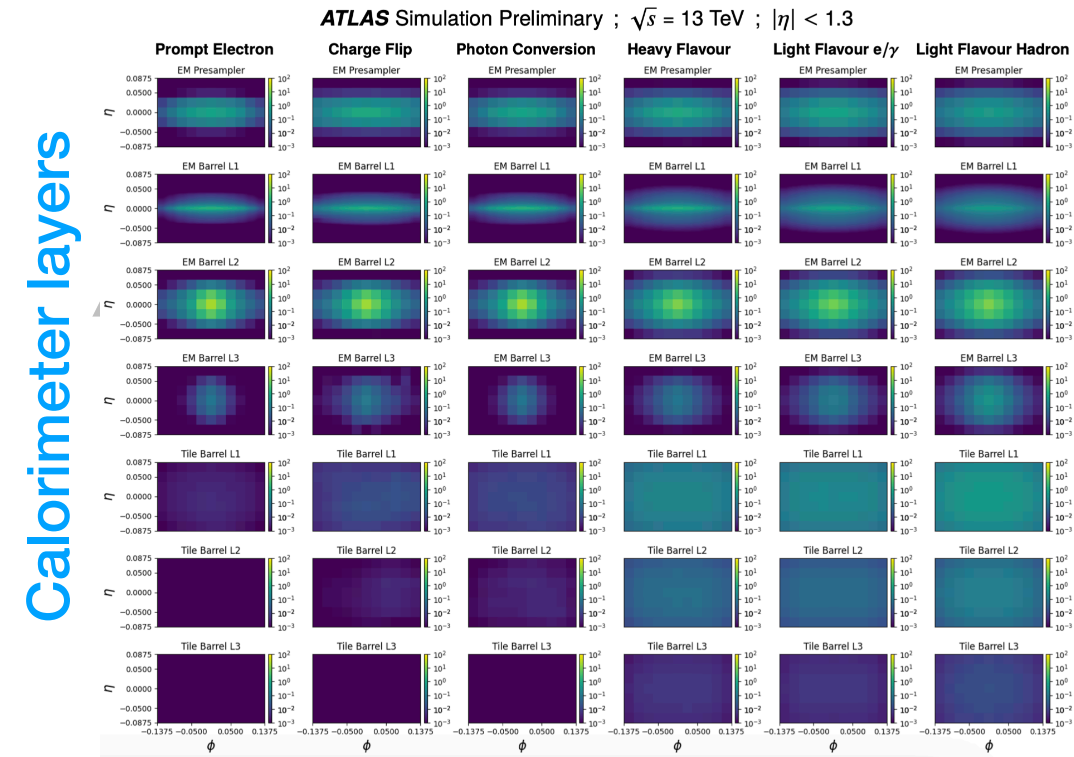
1.3 Input as images

- Jets can be viewed as images
  - So as electrons



JHEP 01 (2017) 110

Electron classes

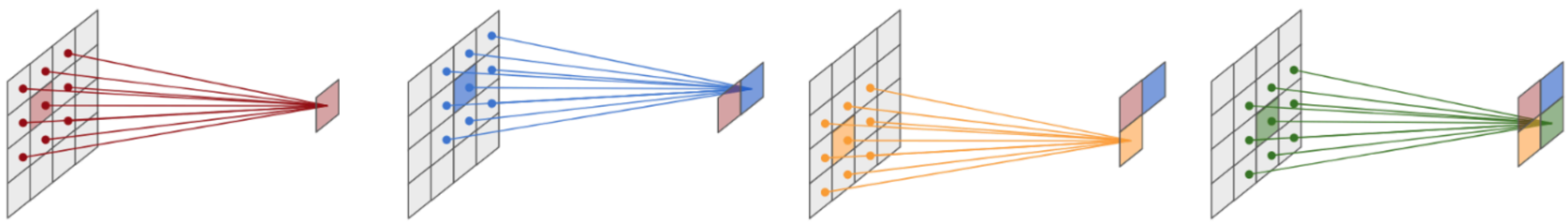


ATL-PHYS-PUB-2023-001

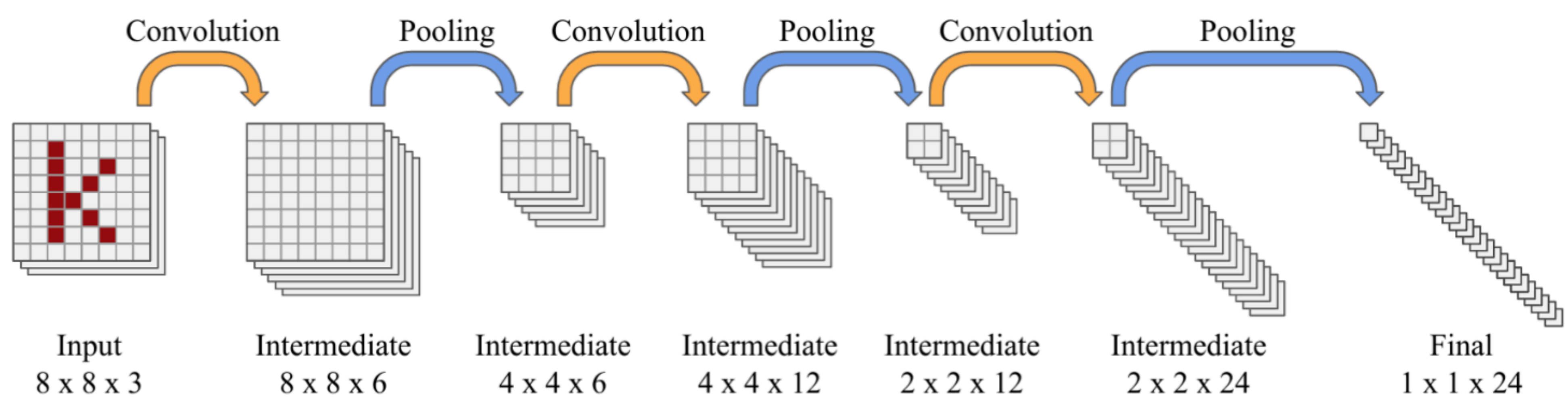
Convolutional neural network

- Convolutional neural network shows great performance for computer vision tasks
  - Nice features: sparse interactions, parameter sharing and equivariance

Convolution operation:



Convolution operation:



PDG Machine Learning

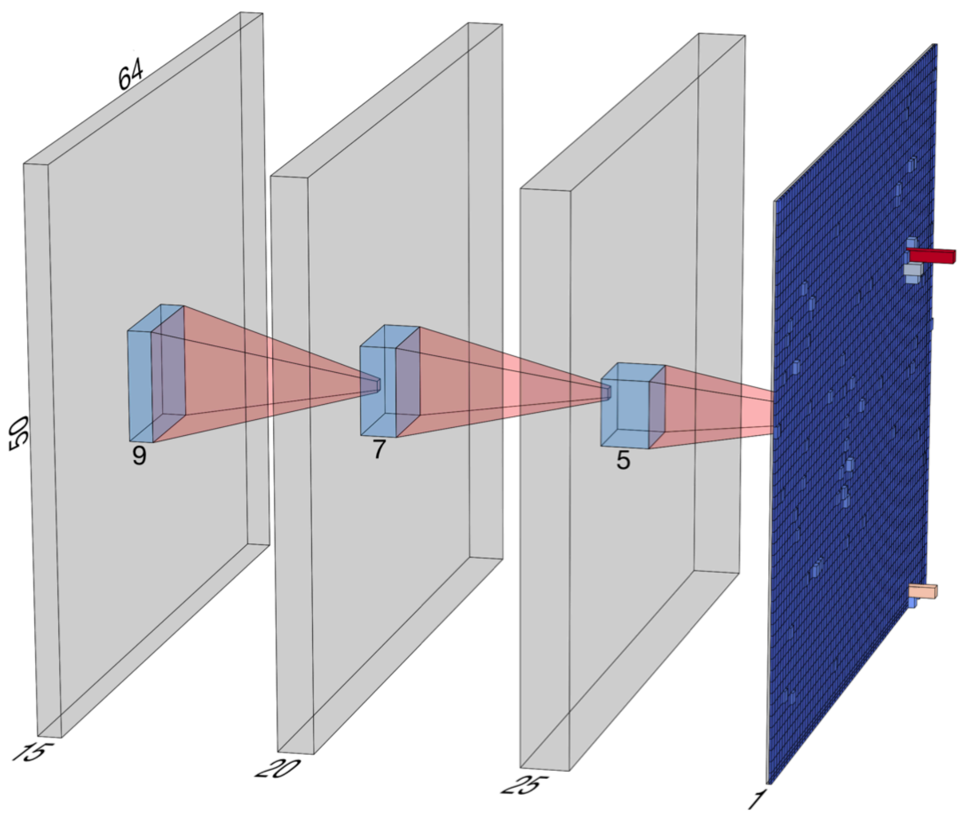
Goodfellow et al. Deep learning. MIT press, 2016.

CNN applications

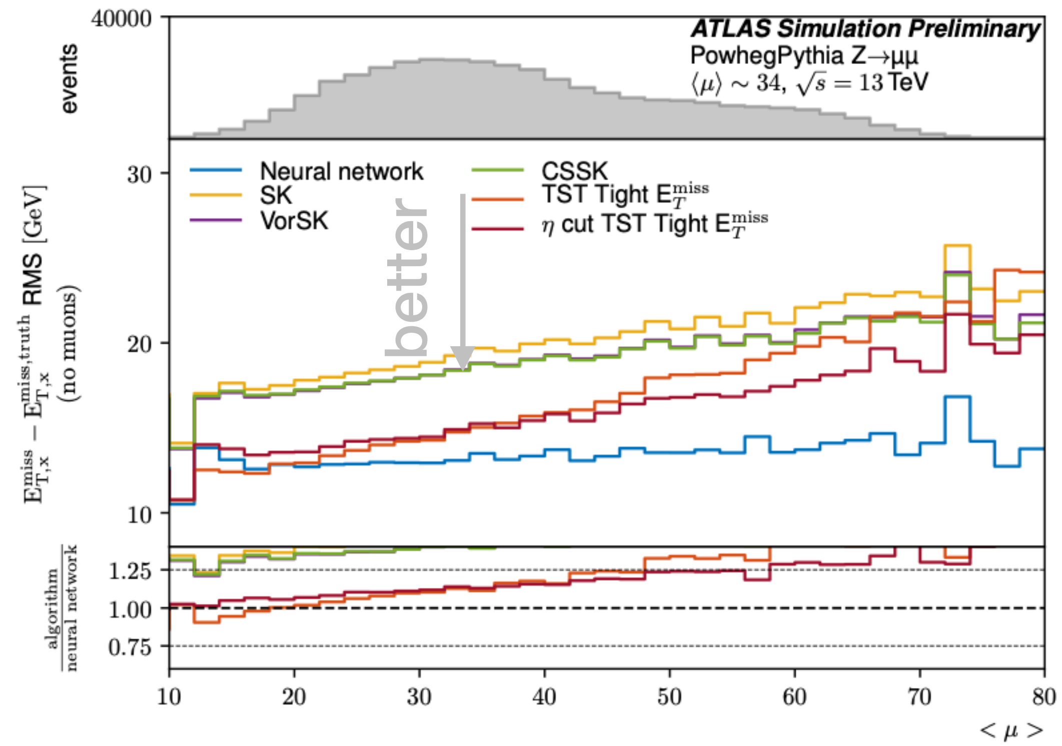
ATL-PHYS-PUB-2019-028

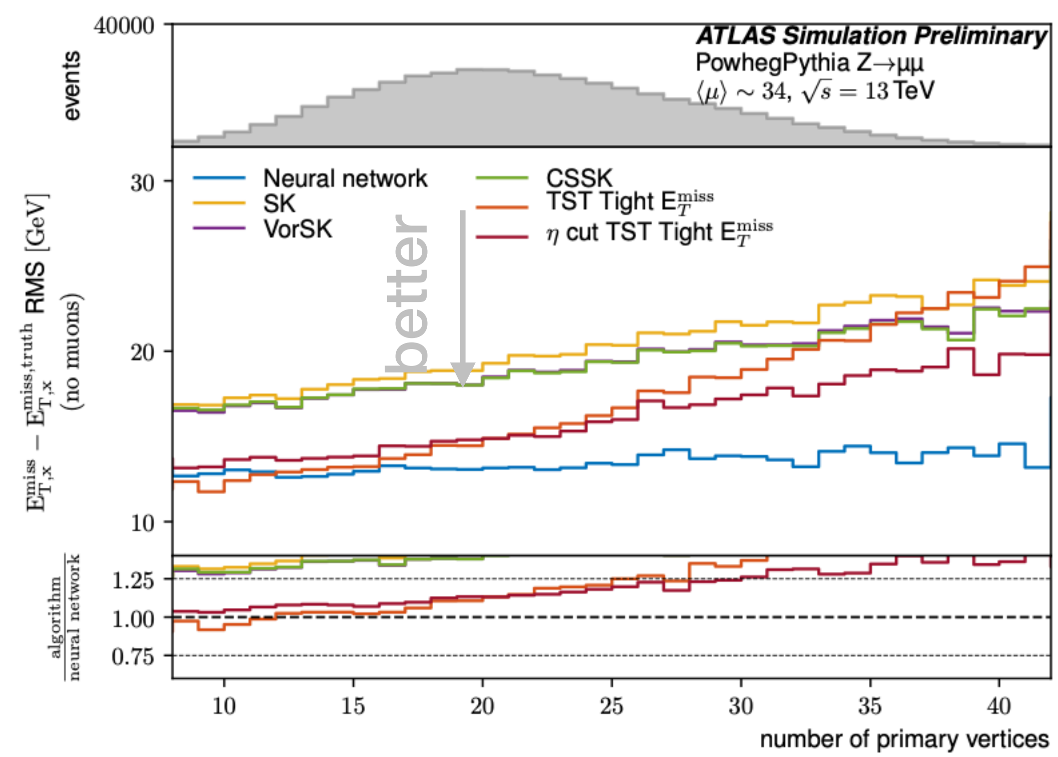
E_T^{miss} reconstruction

An event



Convolution

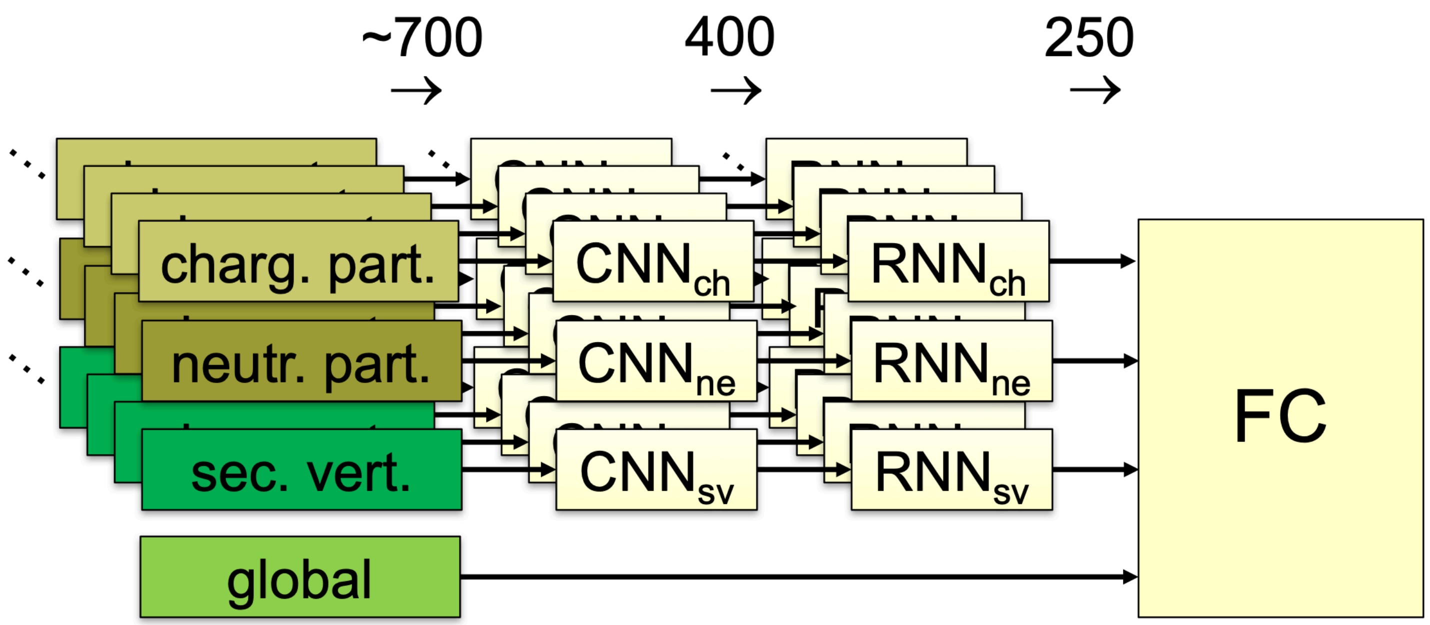




Robust against μ and primary vertices

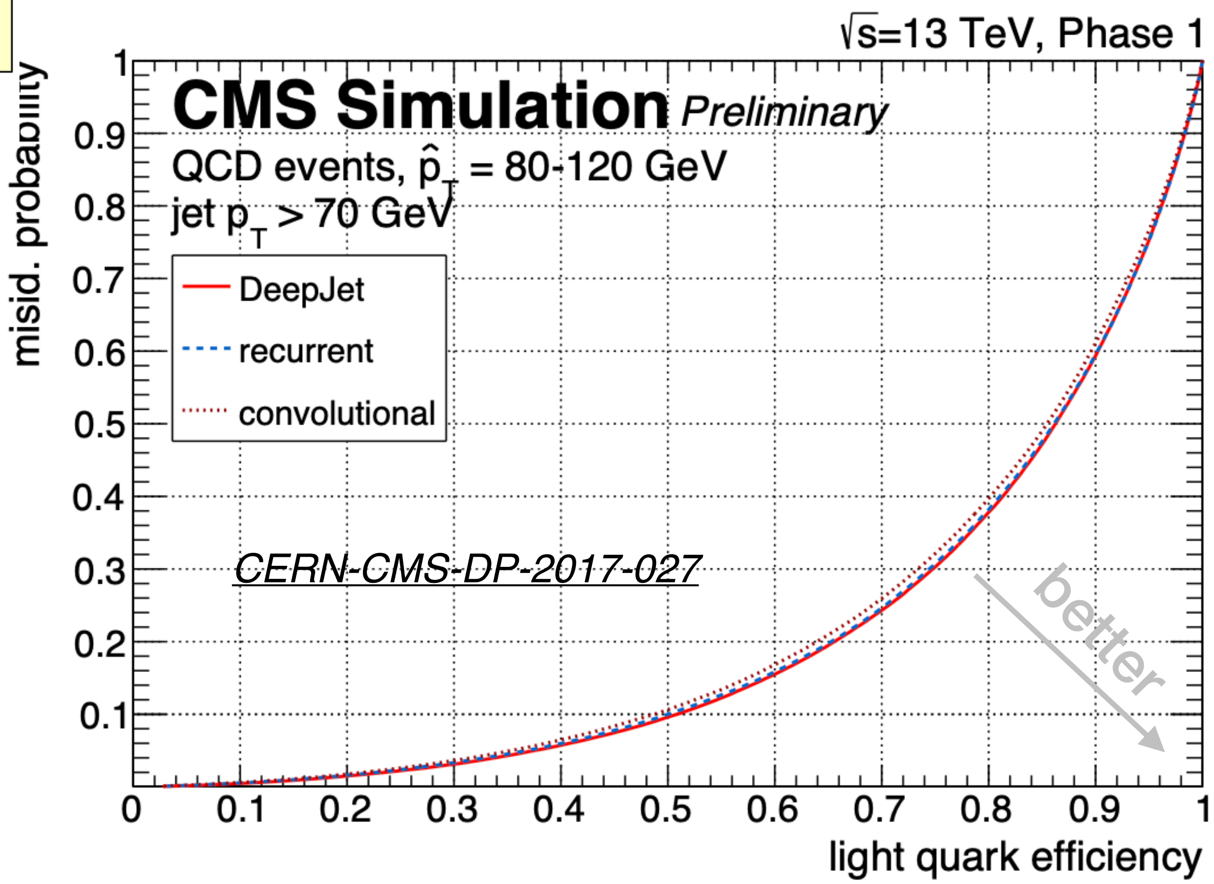
Hybrid: DNN + RNN + CNN application

Particle and vertex based DNN: DeepJet



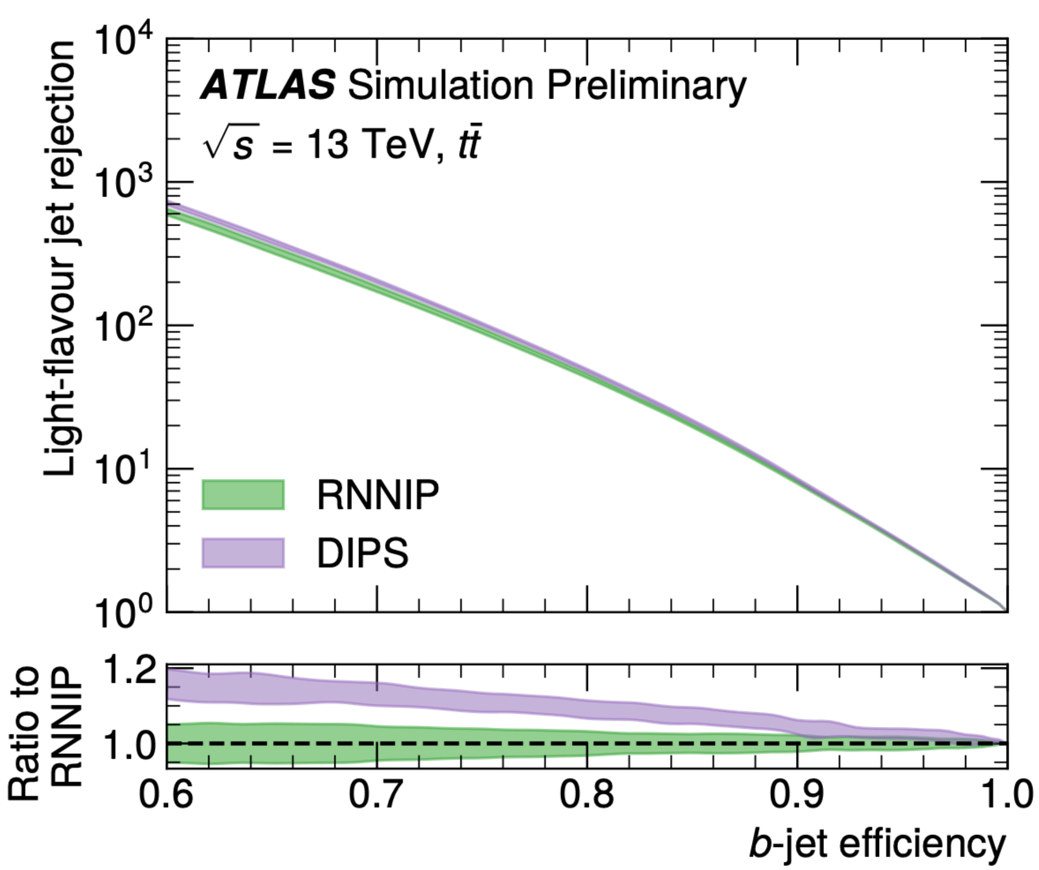
~ 700 inputs and 250.000 model parameters

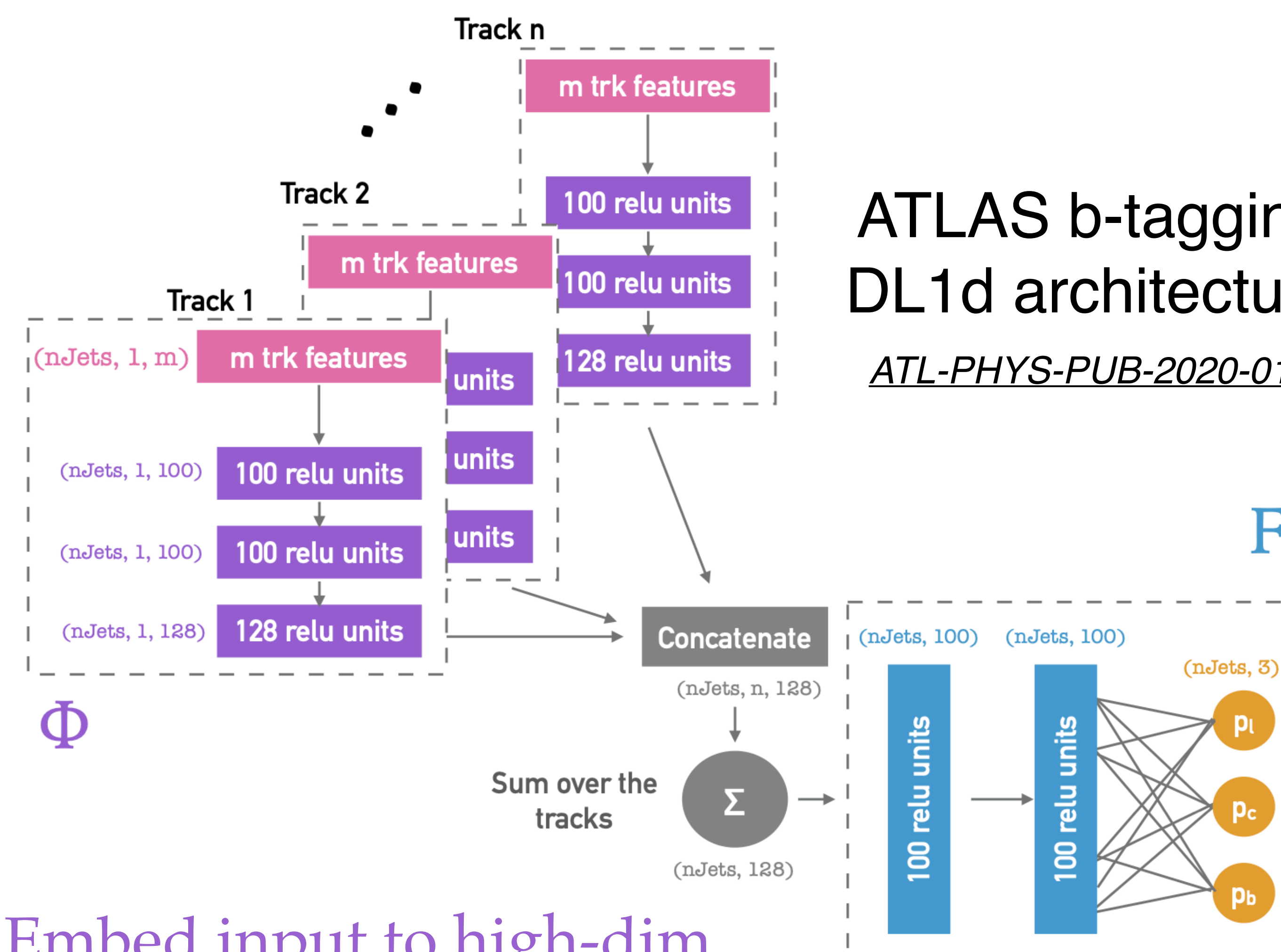
- CMS **DeepJet** algorithm used CNN, RNN and fully connected DNN at the same time



1.4 Input as sets

- Sequence (and also image) implies certain ordering
  - Lack of permutation invariance $f(x_1, x_2) \neq f(x_2, x_1)$
- Deepset [Manzil et al]
  - *for any permutation* $\pi : f(\{x_1, \ldots, x_M\}) = f(\{x_{\pi(1)}, \ldots, x_{\pi(M)}\})$
  - e.g. f = max, mean, etc





ATLAS b-tagging DL1d architecture

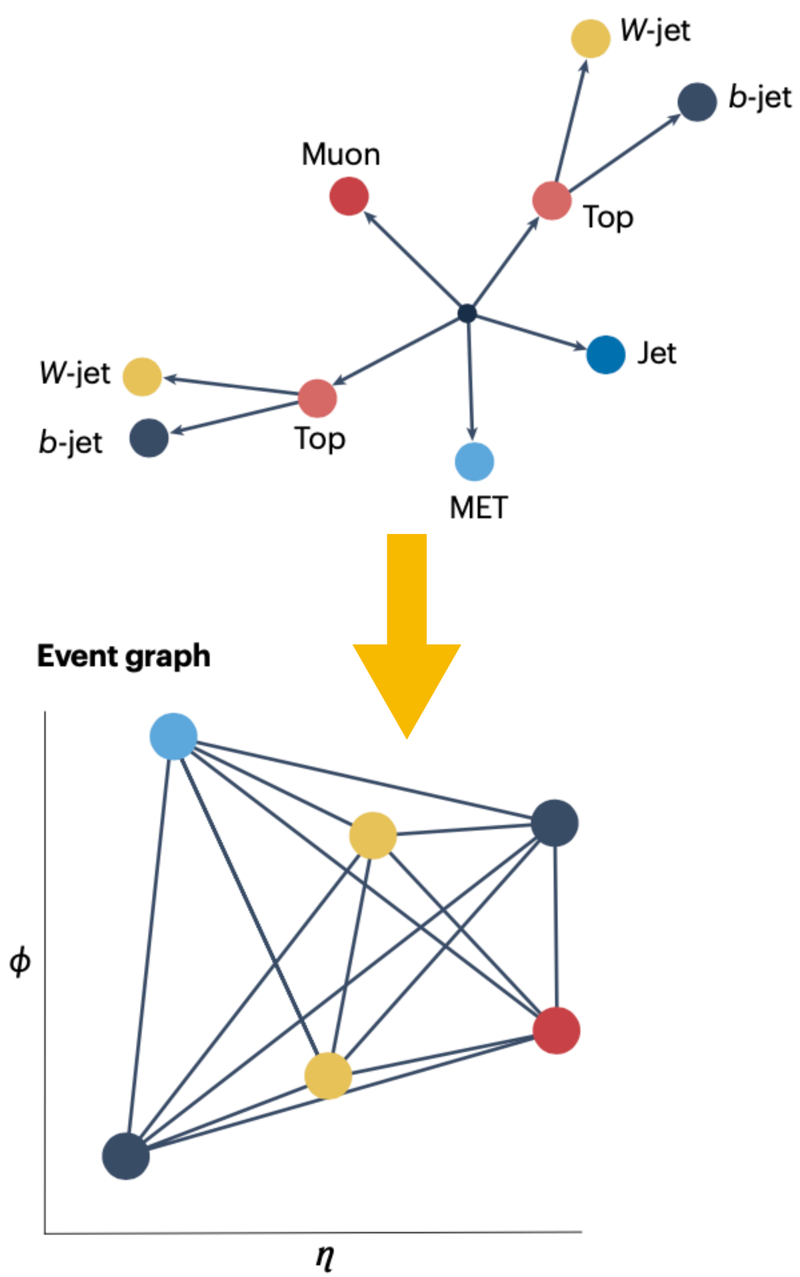
ATL-PHYS-PUB-2020-014

Φ: Embed input to high-dim space to preserve properties

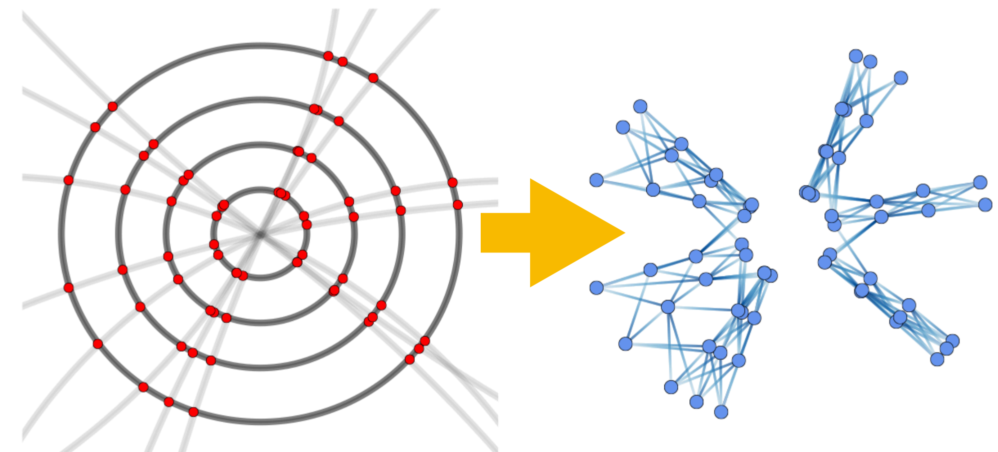
1.5 Input as graphs (including point cloud)

- Graph is also a natural way to represent LHC data

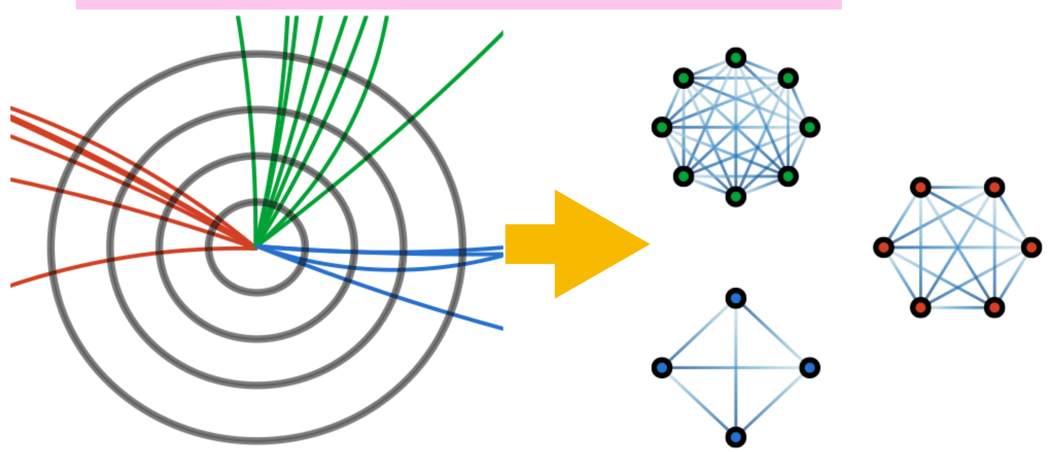
A $t\bar{t}$ event



Hits in tracker



Jet is a graph of particles



[Graph neural networks in particle physics](#)

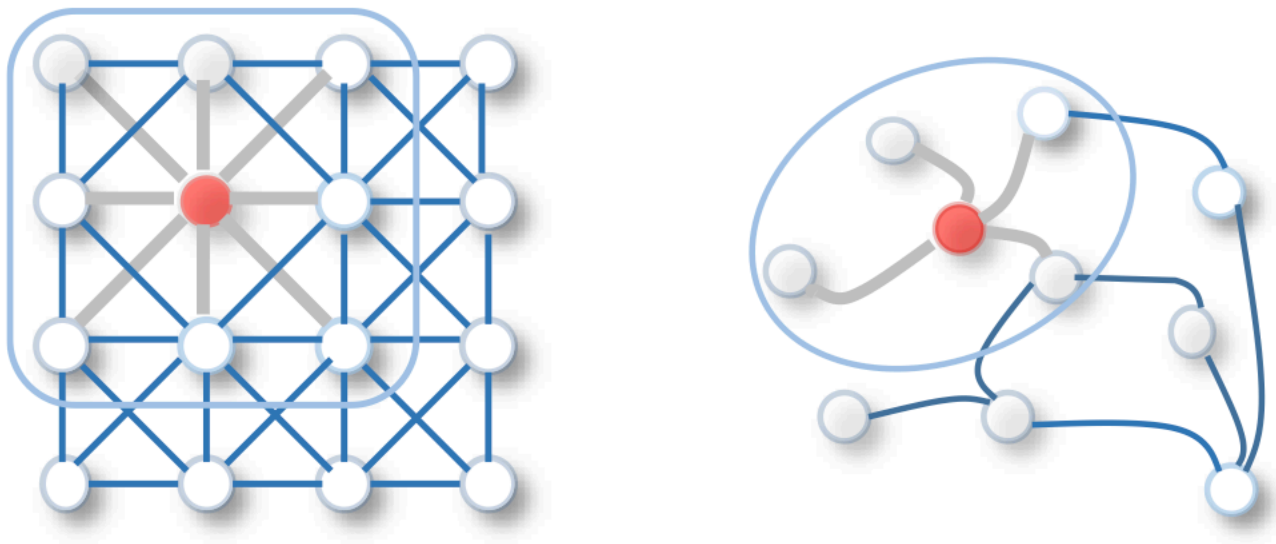
[Graph Neural Networks for Particle Tracking and Reconstruction](#)

[Graph neural networks at the Large Hadron Collider](#)

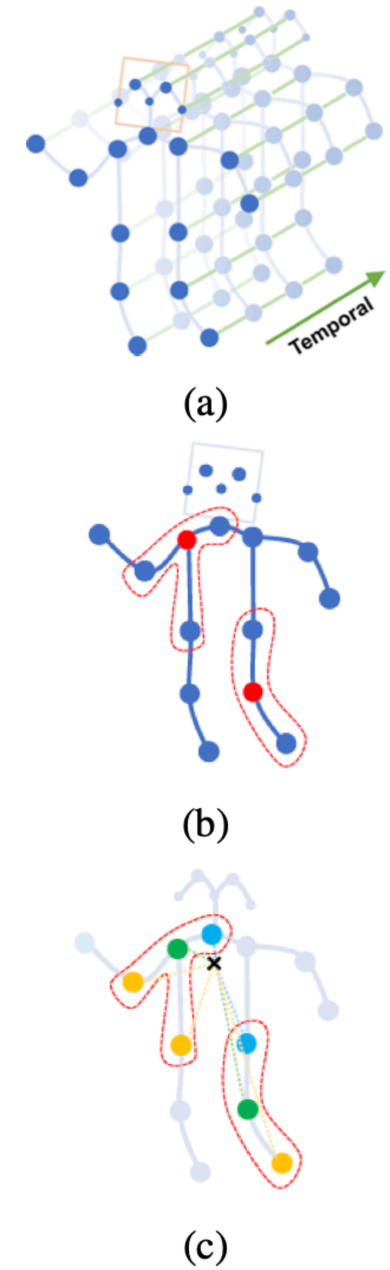
[Graph Neural Networks in Particle Physics: Implementations, Innovations, and Challenges](#)

Graphs neural networks

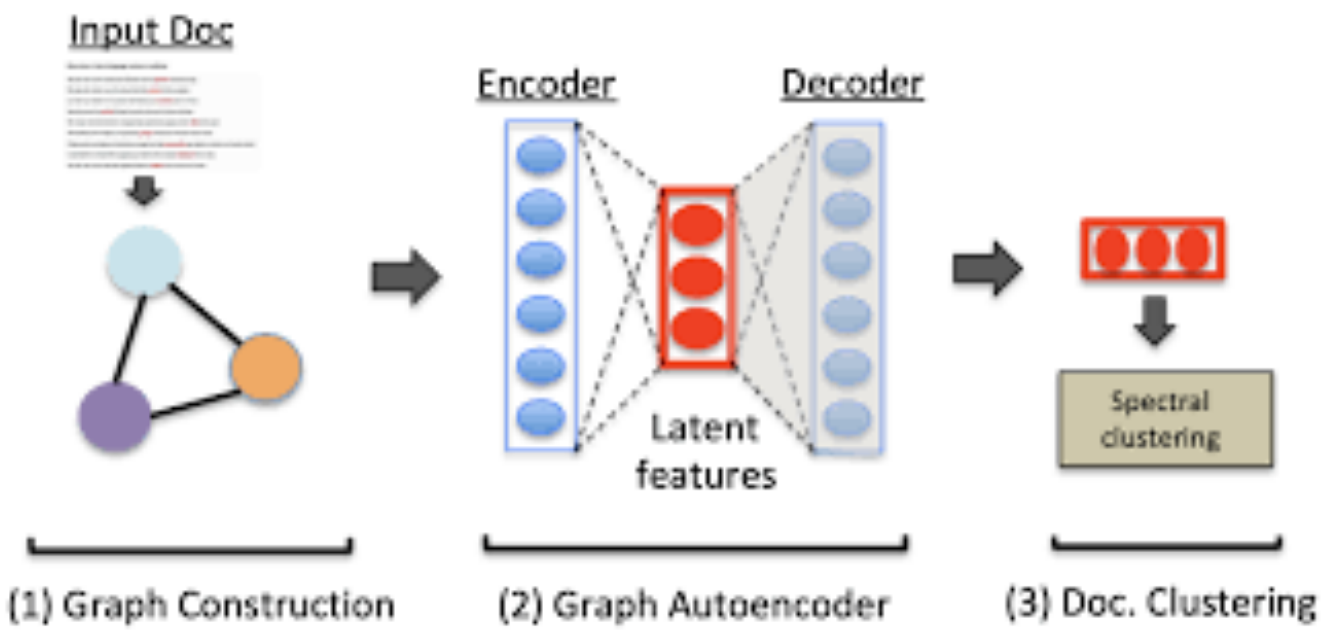
Convolutional graph neural networks (ConvGNNs)



Spatial-temporal graph neural networks (STGNNs)



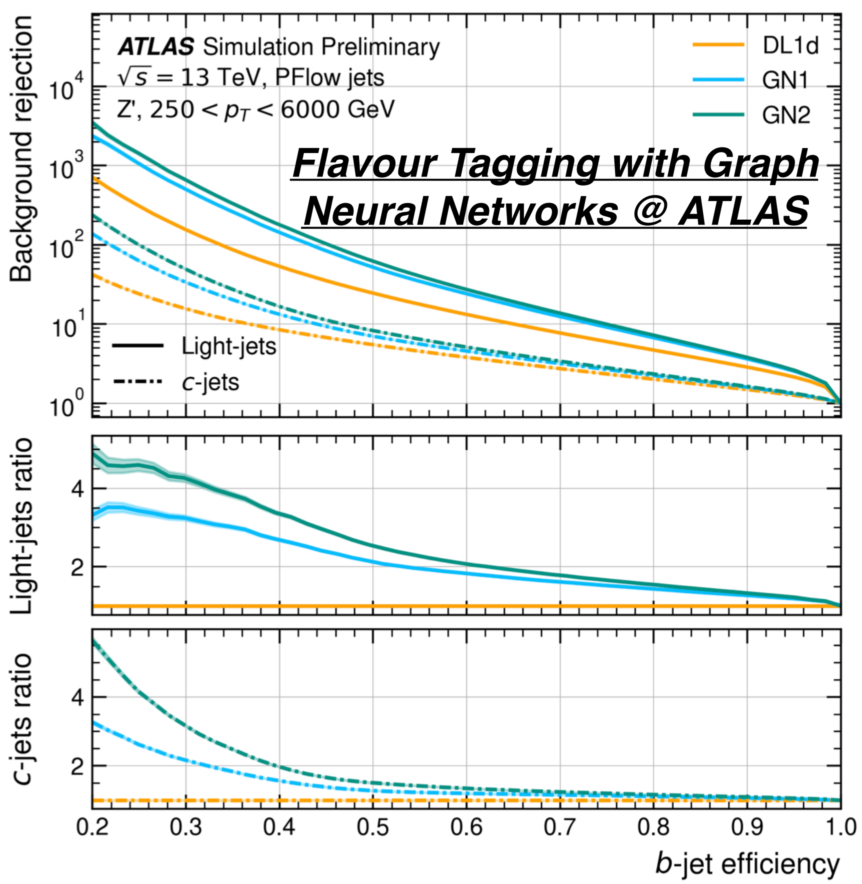
Graph autoencoders (GAEs)



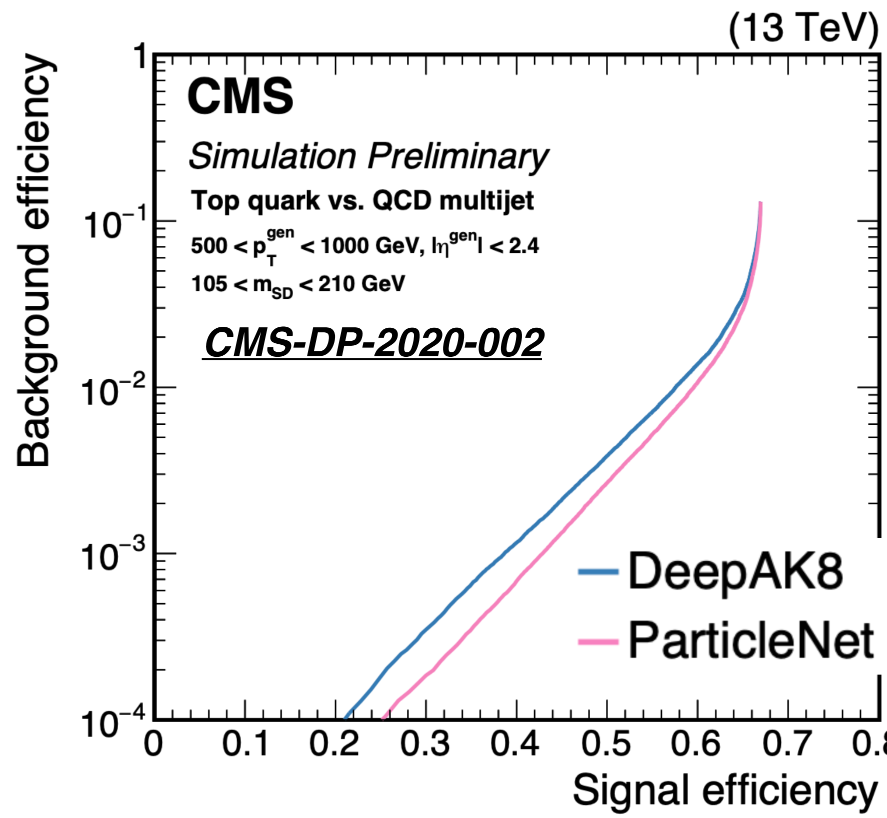
A Comprehensive Survey on Graph Neural Networks

Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition

GNN applications



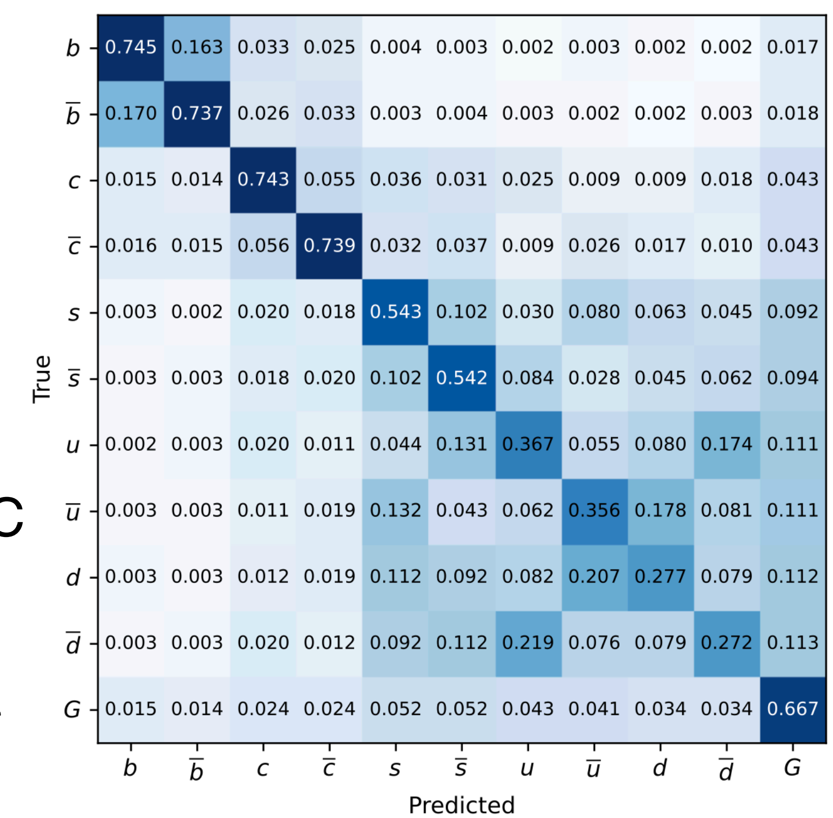
DL1d is DeepSet based b-tagger
GN1 is GNN based b-tagger
GN2 is optimised GN1 + attention mechanics [1706.03762]



: CNN based large R jet tagger
: GNN based large R jet tagger

Jet origin identification using ParticleNet at CEPC
Excellent performance in confusion matrix

arXiv:2310.03440
see Manqi's talk on 13th Saturday for more details

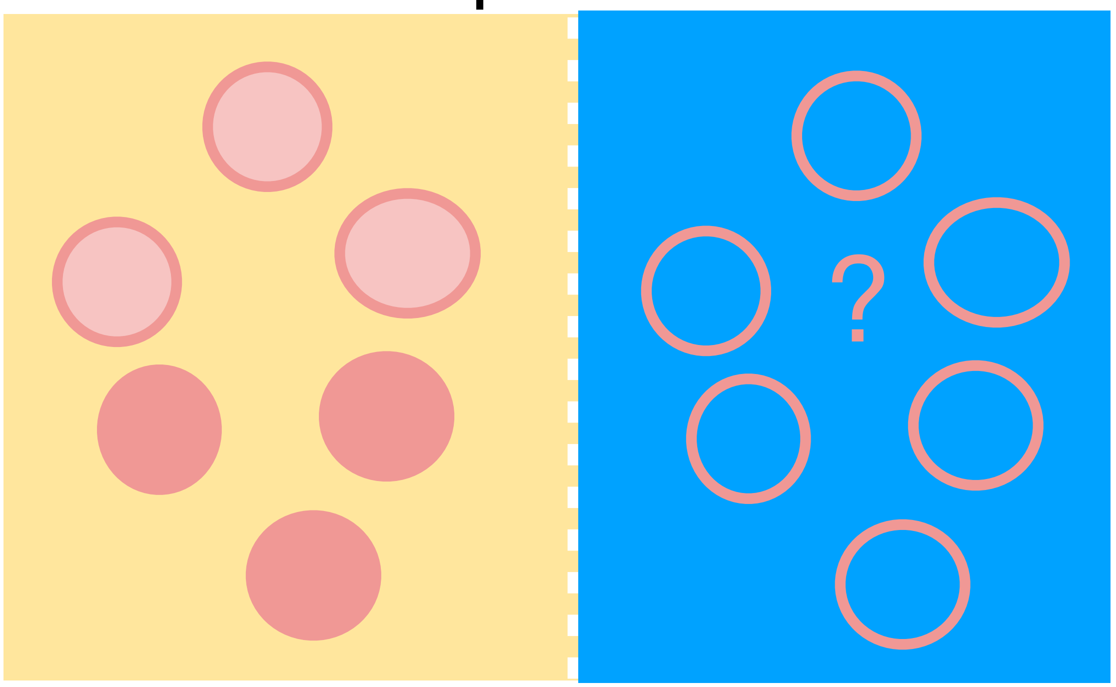


Step 2: set up the learning task

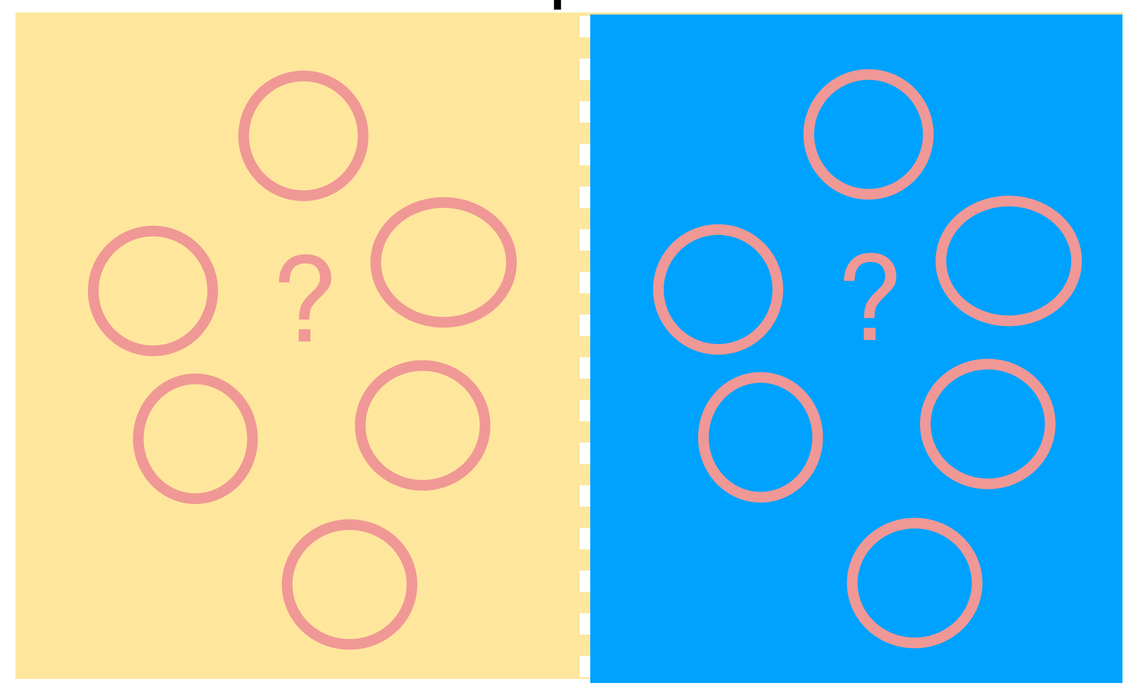
- A common feature of previous examples are **supervised** machine learning
- Trained on data with known signal, known background → known labels

Step 2: set up the learning task

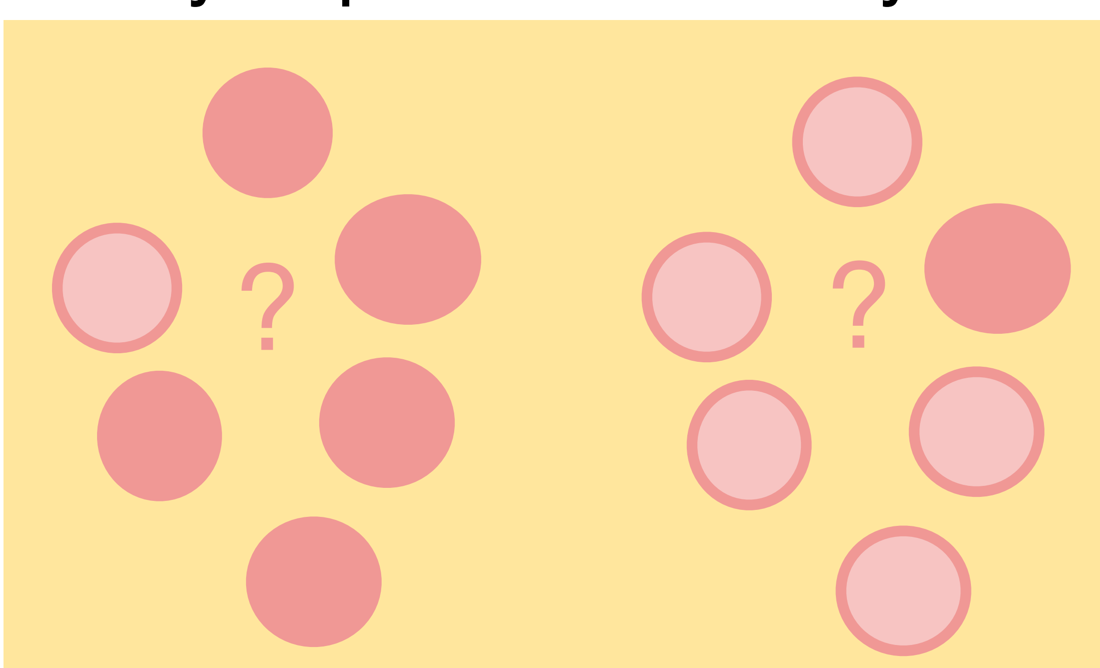
Supervised



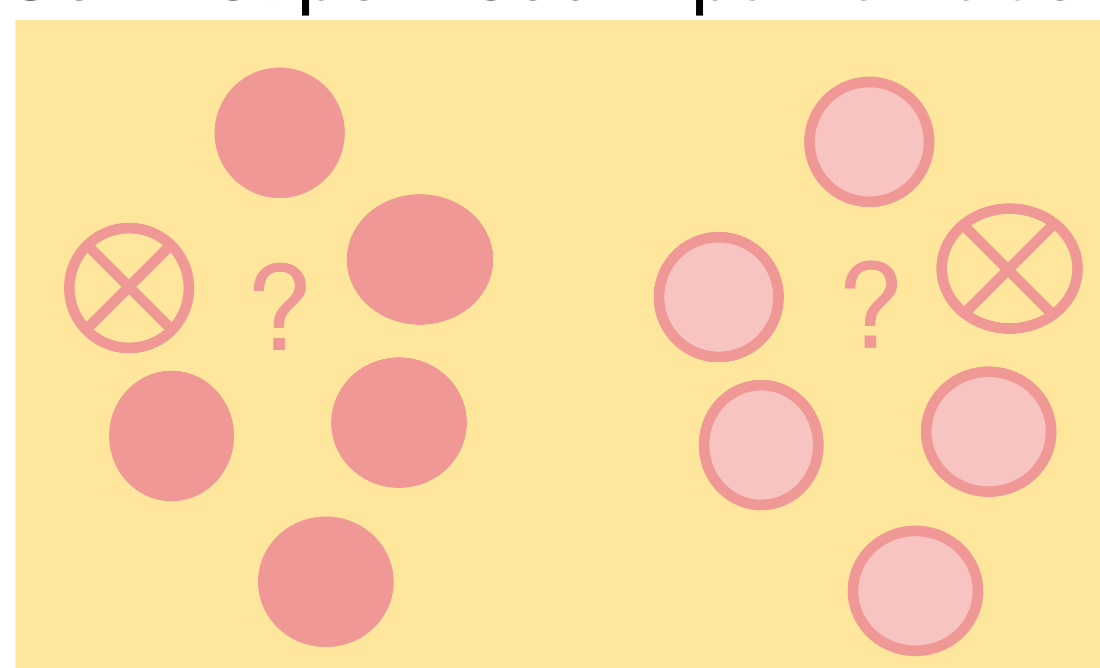
Unsupervised



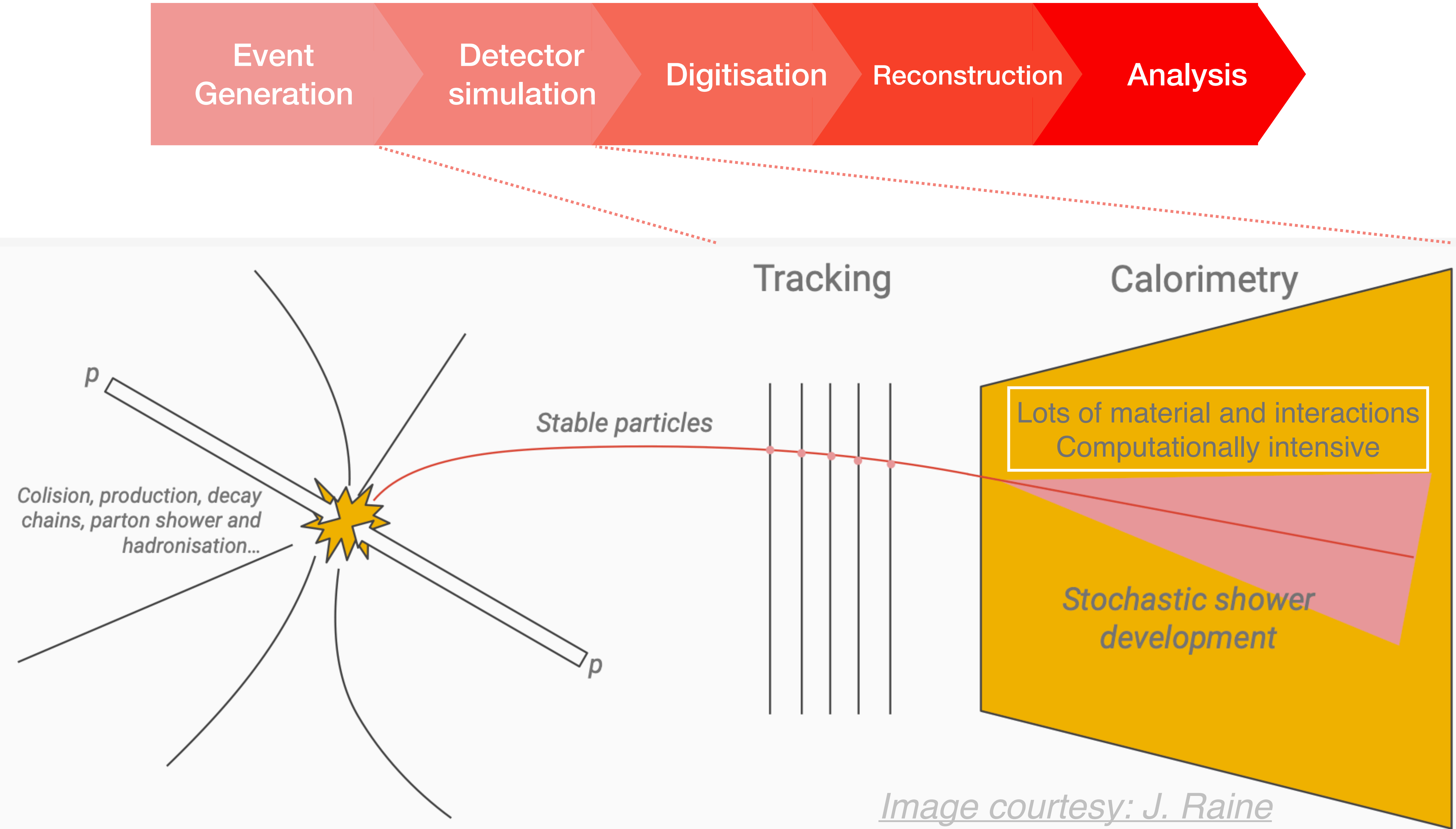
Weakly-supervised = noisy labels



Semi-supervised = partial labels

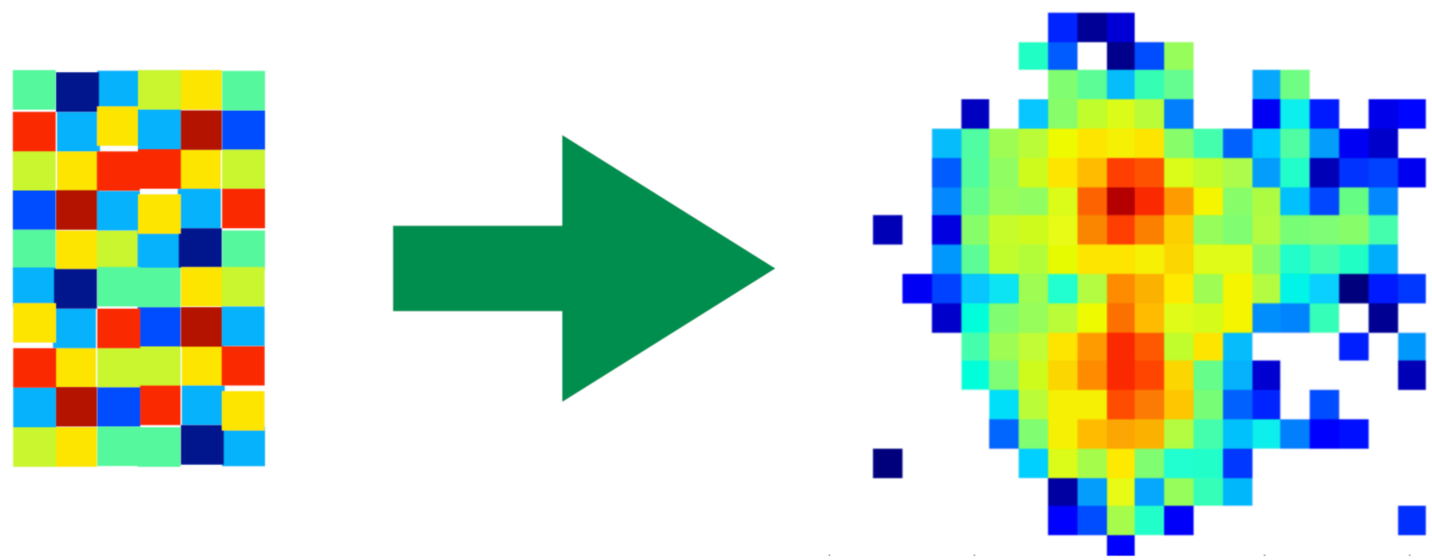


Unsupervised—fast simulation (than Geant4)



Fast simulation: Generative models

A generator is a function that maps random numbers to structure.



Generative models are typically unsupervised.

GAN
Generative Adversarial Networks

PRD 97, 014021 (2018)
2309.06515
2207.04340
…

VAE
Variational Autoencoders

2211.15380
2203.00520
2210.07430
…

NF
Normalizing Flows

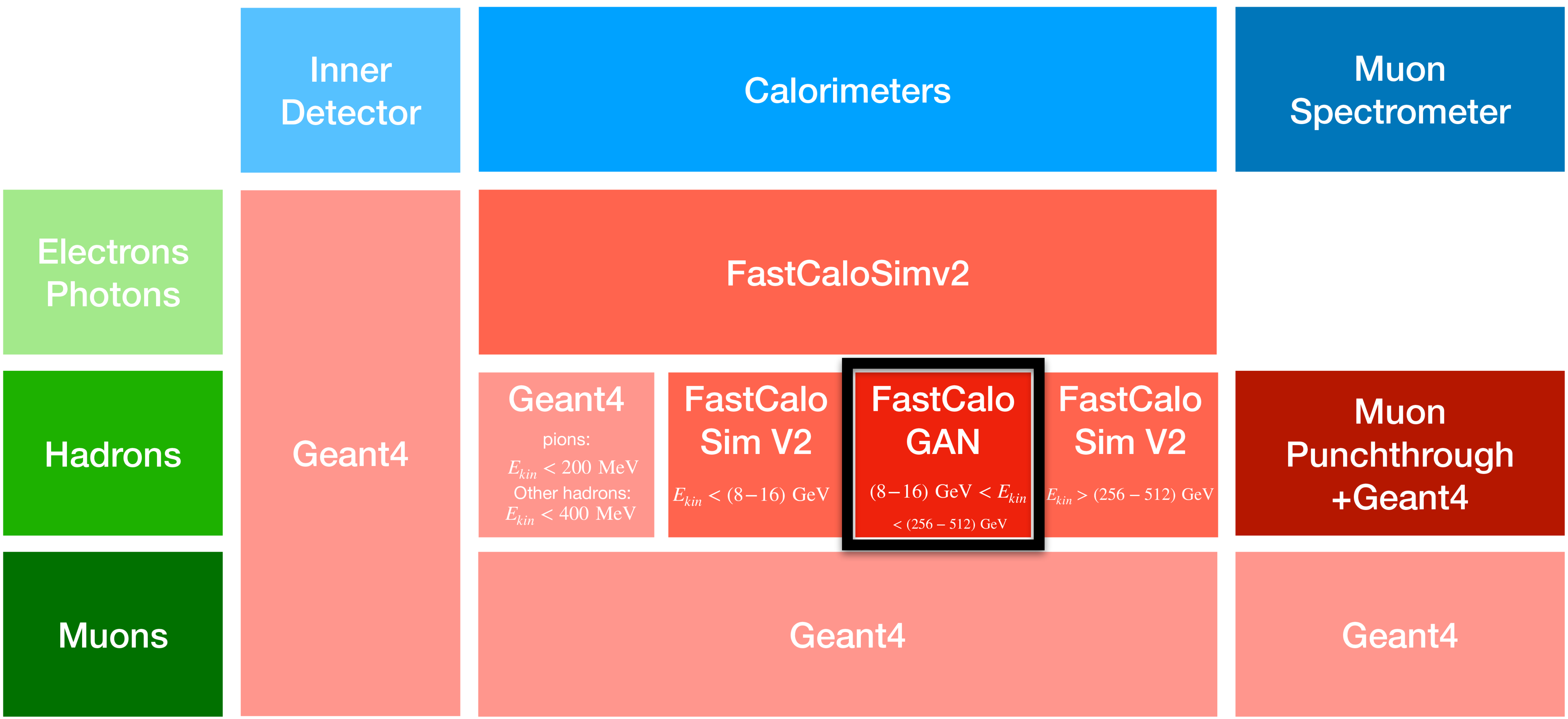
JINST 2023 18 P10017
2308.11700
PRD 107.113003
2305.11934
…

Diffusion
Diffusion model

PRD 108 (2023) 072014
2309.05704
2308.03847
…

Example—Integrated into real experiment

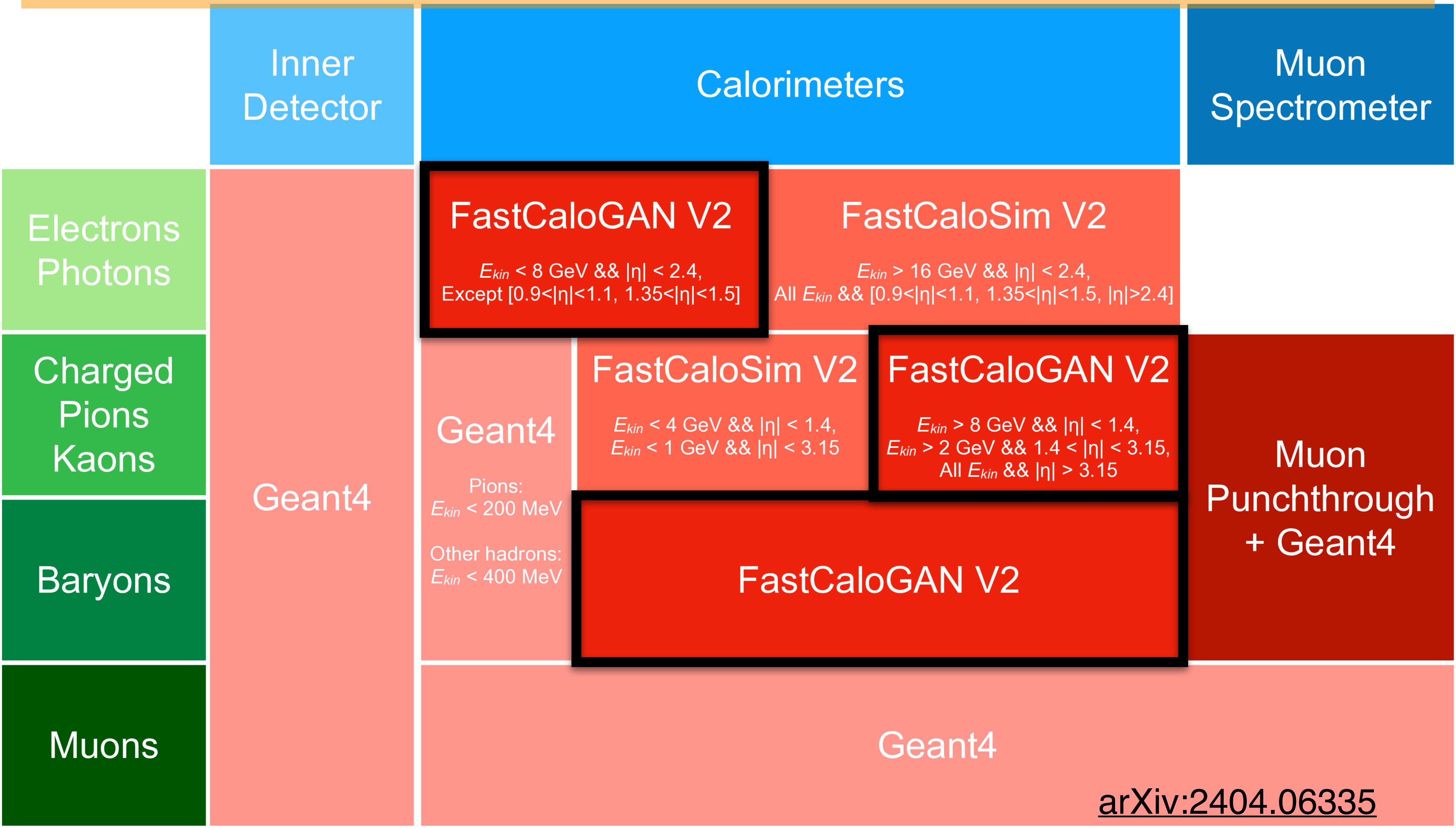
ATLAS fast simulation includes a GAN at intermediate energies for hadrons



COMPUT SOFTW BIG SCI 6, 7 (2022)

Example—Integrated into real experiment

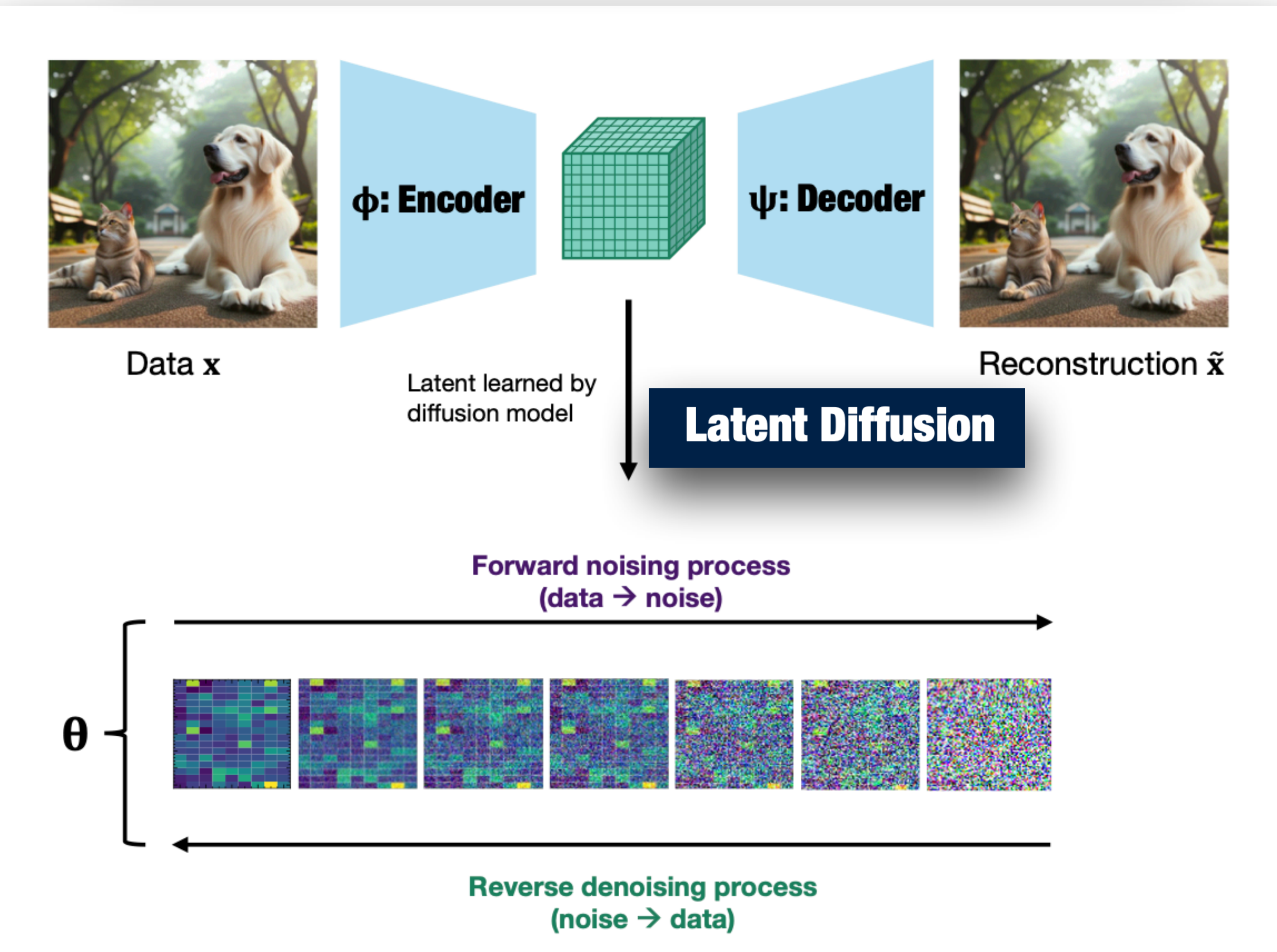
FastCaloGAN has been expanded from Run 2 to Run 3



Marrying generative techniques (in R&D)

CaloLatent: Score-based Generative Modelling in the Latent Space for Calorimeter Shower Generation

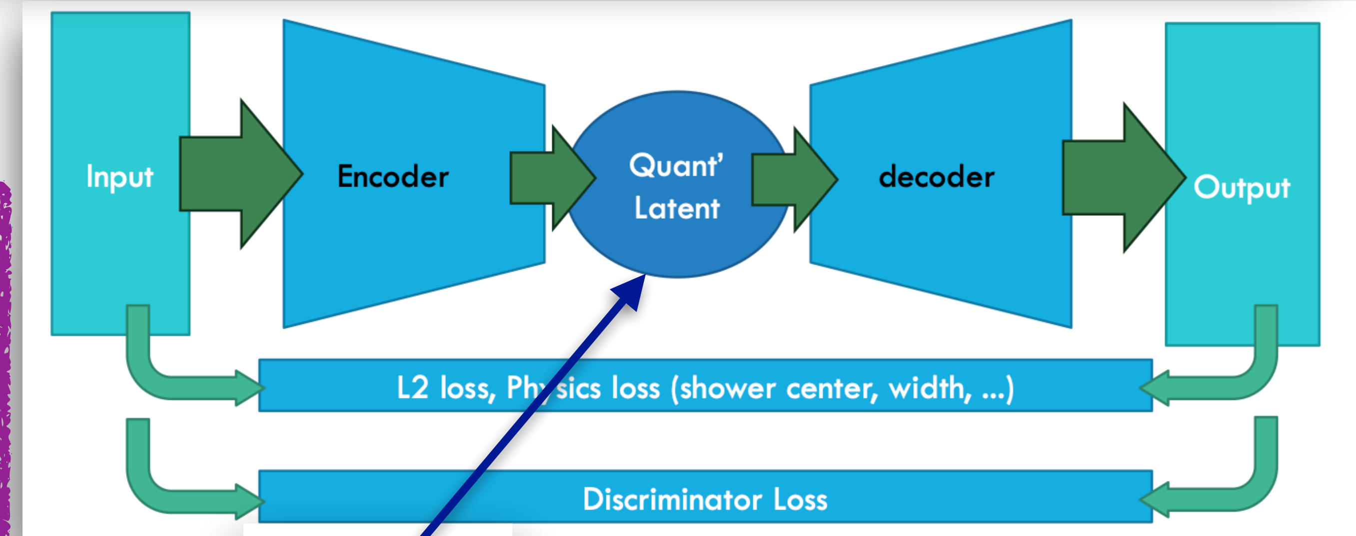
Thandikire Madula: UCL
Vinicius M. Mikuni: NERSC

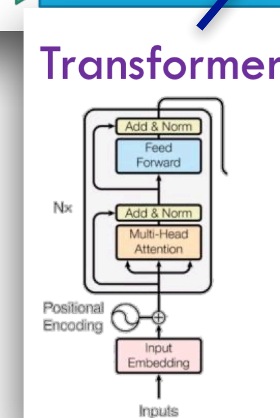


Tsung-Dao Lee Institute
Yale

Latent Generative Model for Calorimeter Fast Simulation

Qibin LIU, Chase Shimmin, Xiulong LIU, Eli Shlizerman, Shih-Chieh HSU, Shu LI





Transformer for sampling in latent space

“modern” concepts as a plugin put into traditional architectures

Un-/weakly/semi supervised—anomaly detection

Why anomaly detection?

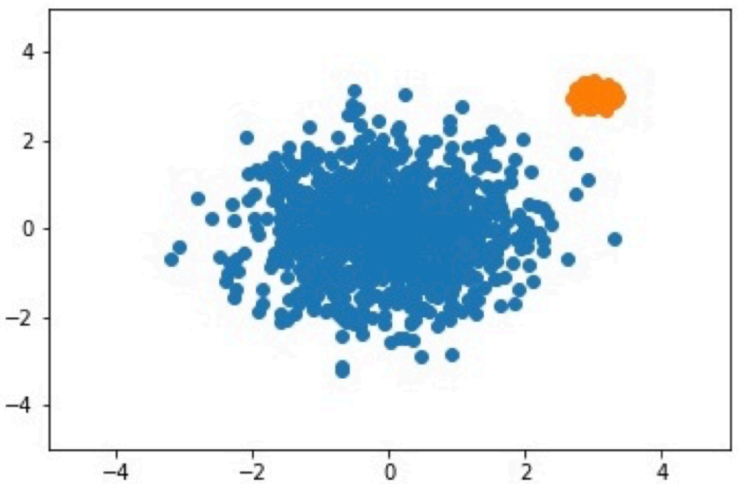
Typical Searches

- Looking for a specific, physics motivated signal
- Maximum sensitivity (using supervised learning e.g. BDT) for a specific model
- Not very useful for other signal models

Anomaly Detection

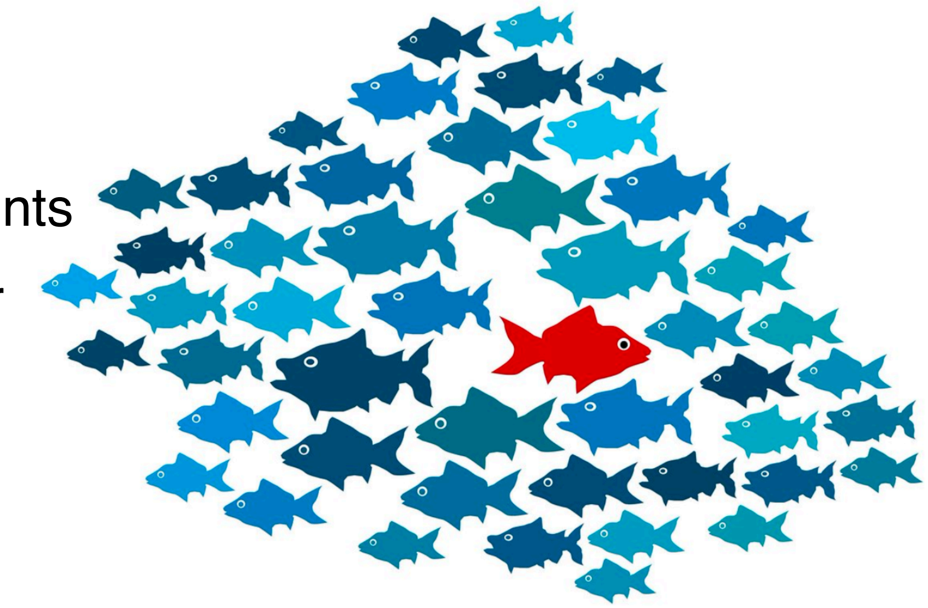
- Model agnostic/ independent search
- Looking for deviations from background only
- Less sensitive to any specific model, but can look for multiple different models

Two types of anomaly detection

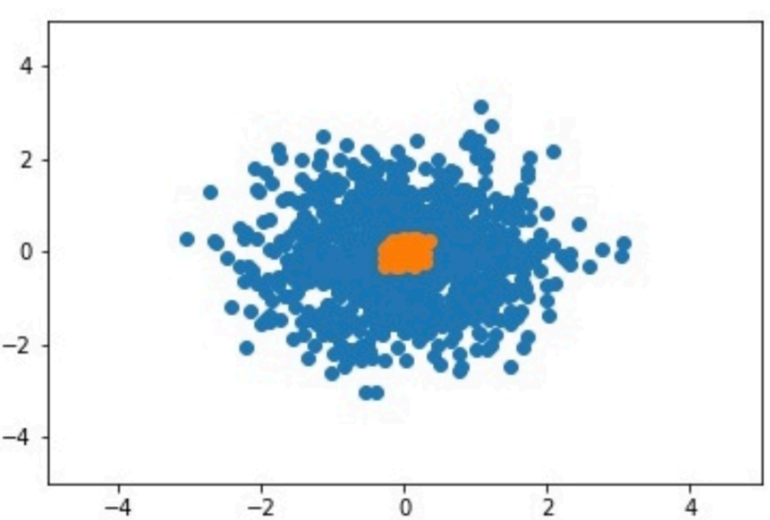


Outlier Detection

- Searching for unique or unexpected events
- In HEP, this is the tails of distributions or uncovered phase space

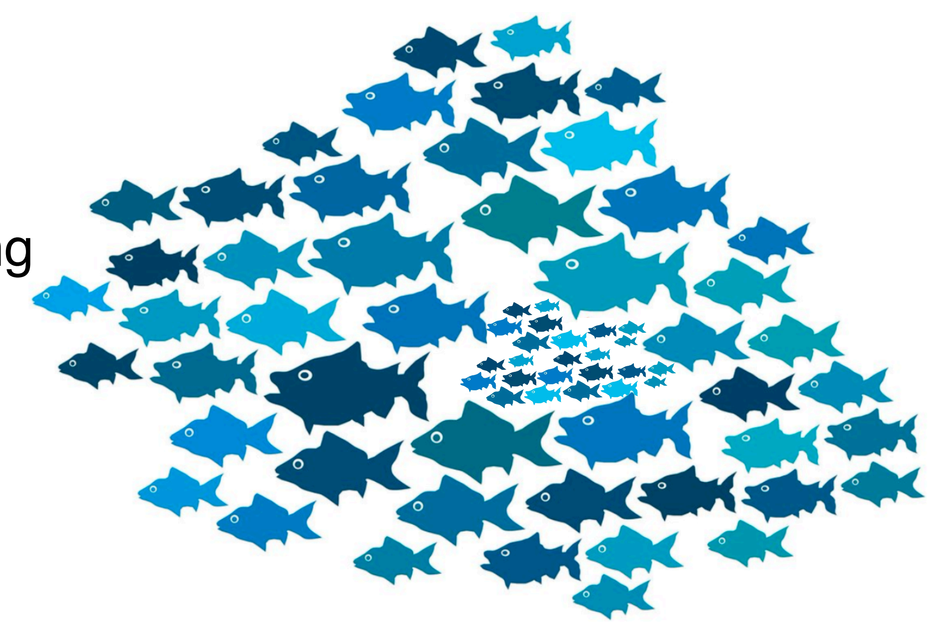


[1807.10261, 1808.08979, 1808.08992, 1811.10276, 1903.02032, 1912.10625, 2004.09360, 2006.05432, 2007.01850, 2007.15830, 2010.07940, 2102.08390, 2104.09051, 2105.07988, 2105.10427, 2105.09274, 2106.10164, 2108,03986, 2109.10919, 2110.06948, 2112.04958, 2203.01343,2206.14225, 2303.14134, 2304.03836, 2306.03637, 2308.02671, 2309.10157, 2309.13111, ...]



Overdensity detection

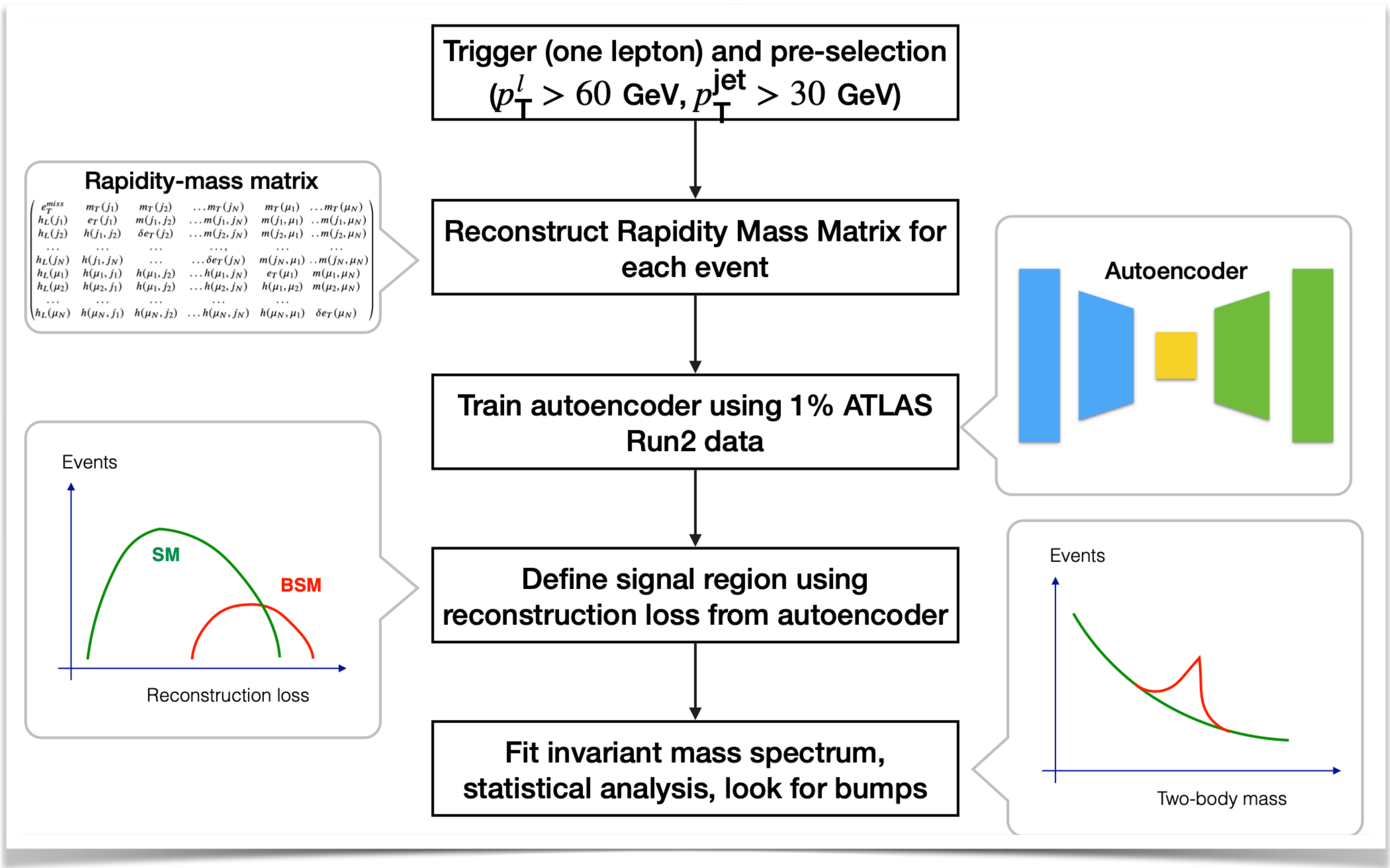
- Analogous to the traditional bump hunting



[1805.02664, 1806.02350, 1902.02634, 1912.12155, 2001.05001, 2001.04990, 2012.11638, 2106.10164, 2109.00546, 2202.00686, 2203.09470, 2208.05484, 2210.14924, 2212.11285, 2305.04646, 2305.15179, 2306.03933, 2307.11157, 2309.12918, 2310.06897, 2310.13057, ...]

Inspired by this presentation

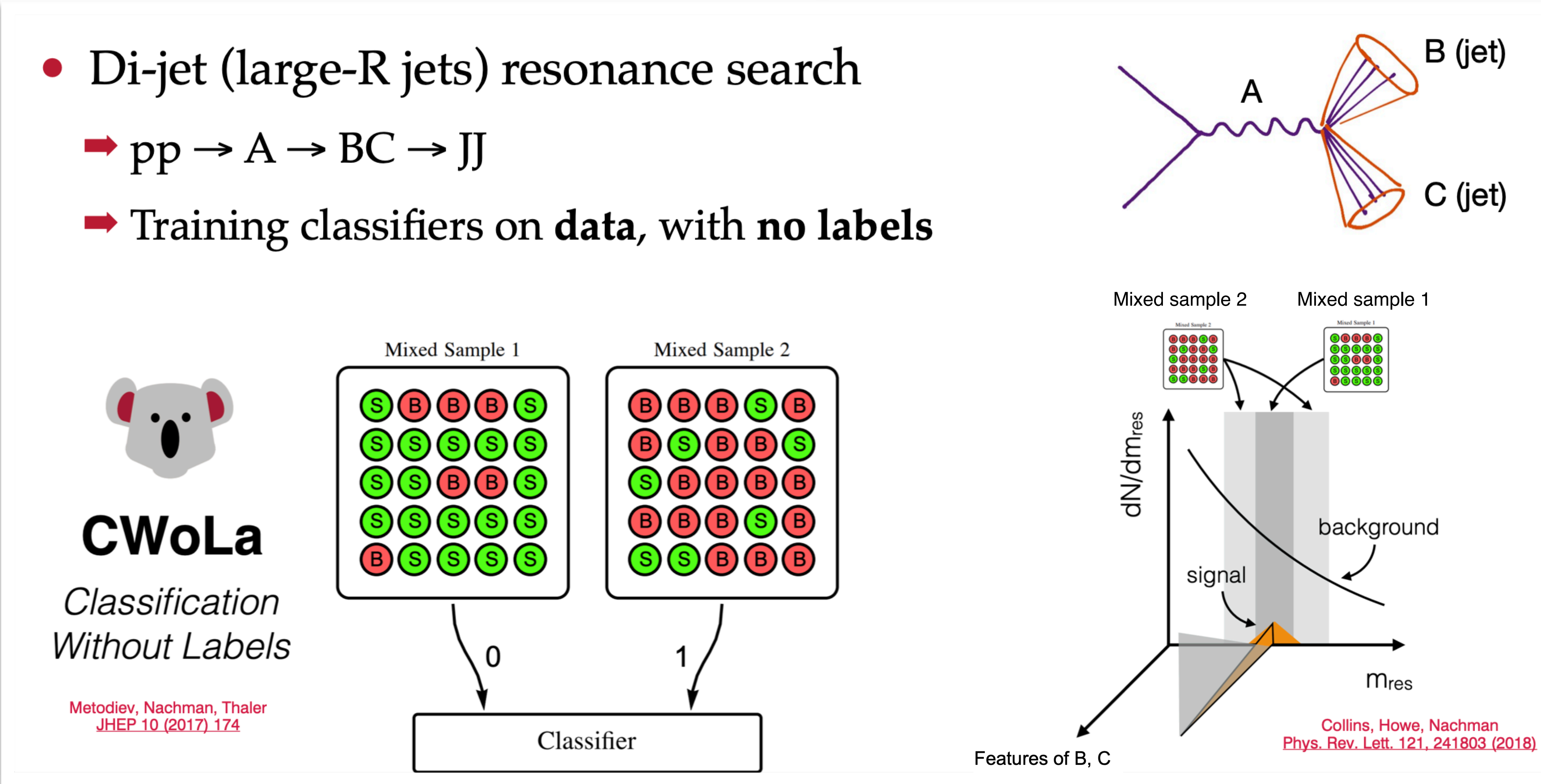
Outlier Detection in experiments (ATLAS)



- Full event level anomaly detection
- Searched in 9 invariant masses including di-jet, di-b-jet, with three anomaly regions => demonstrating high efficiency in the search

arXiv: 2307.01612

Overdensity Detection in experiments (ATLAS)



- Performed on di-fatjet resonant search
- Network is learning difference between Prob(b) and Prob(s+b)

Phys. Rev. Lett. 125 (2020) 131801

Summary

- ML and HEP: an enduring partnership
  - ML has been a longstanding companion in HEP in various stages of the data analysis pipeline
- ML as a Toolset for HEP
  - ML serves as a valuable assistant, maximising the exploration of costly collision data
  - Choosing ML architectures based on the data structures to optimise efficiency
  - Evolution towards unsupervised and semi-supervised learning on more generative tasks
- Future directions
  - Expanding training data to refine ML models
  - Delving into lower level features to uncover hidden patterns
  - Incorporating physics knowledge for a deeper contextual understanding

ML continues to unlock breakthroughs within the realm of HEP.

Backup

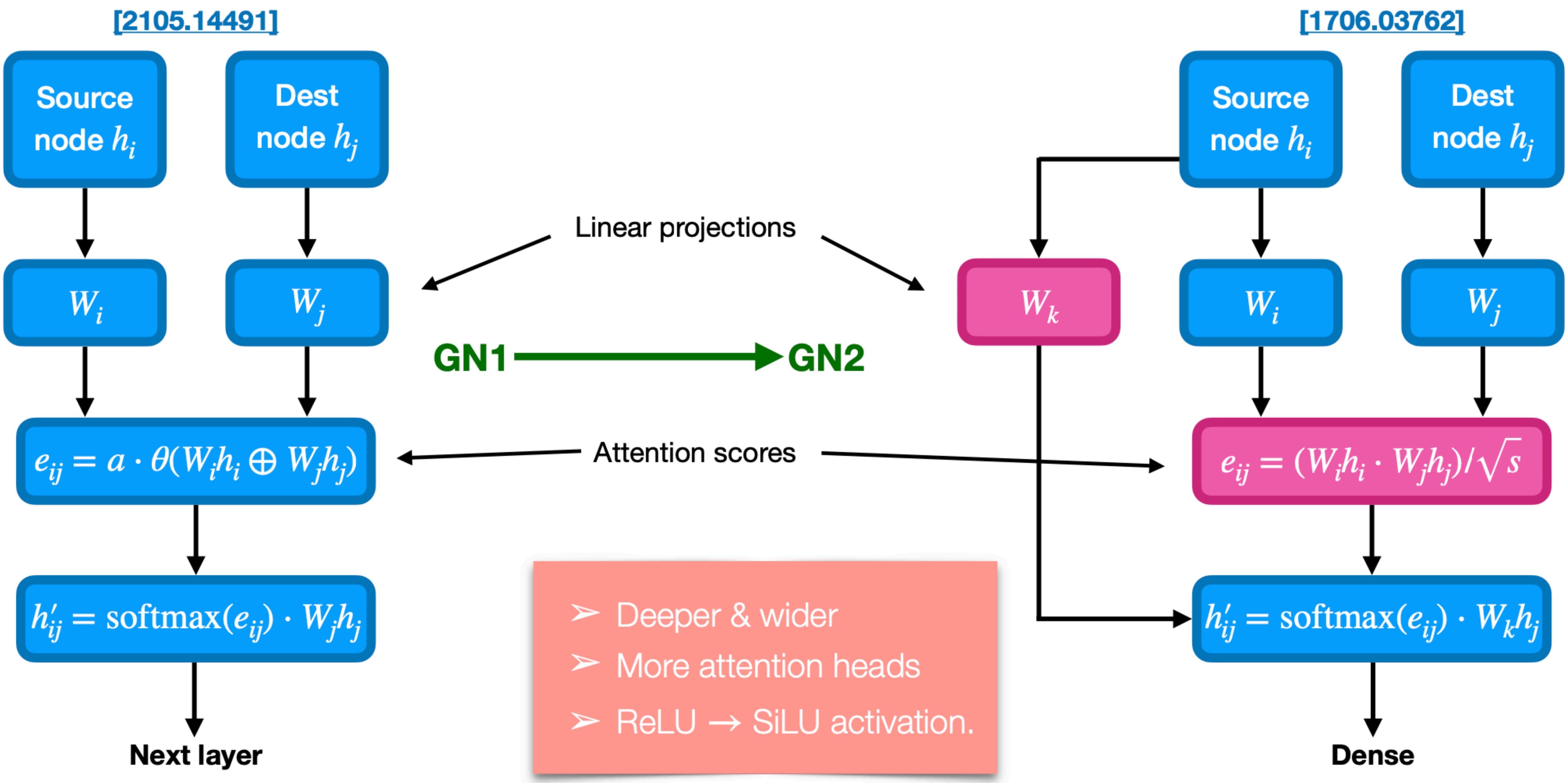
References

- A Living Review of Machine Learning for Particle Physics
  - https://iml-wg.github.io/HEPML-LivingReview/
  - Updated summary on arXiv available submission in machine learning in HEP
- Neural Networks, Types, and Functional Programming
  - http://colah.github.io/posts/2015-09-NN-Types-FP/
  - Deep learning introduction in 10 min
- (New) Machine learning chapter in the particle data group book:
  - https://pdg.lbl.gov/2023/reviews/rpp2022-rev-machine-learning.pdf

GN1 vs GN2

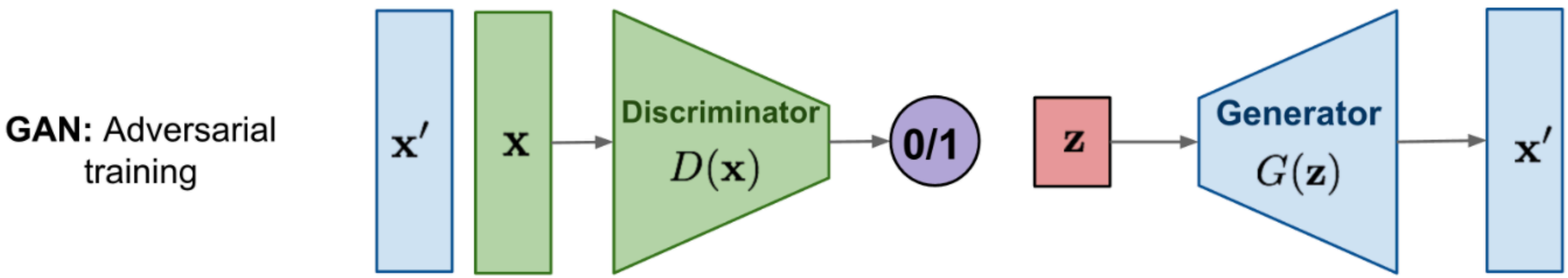
Updated Attention Mechanism

GN2 follows more closely the *transformer* architecture [1706.03762]



Generative model: GAN

[A summary blog]



Generative Adversarial Networks (GANs): A pair of networks where one produce realistic data and the other classifies it as fake or real.

🙂 High-quality output

☹️ Training instability

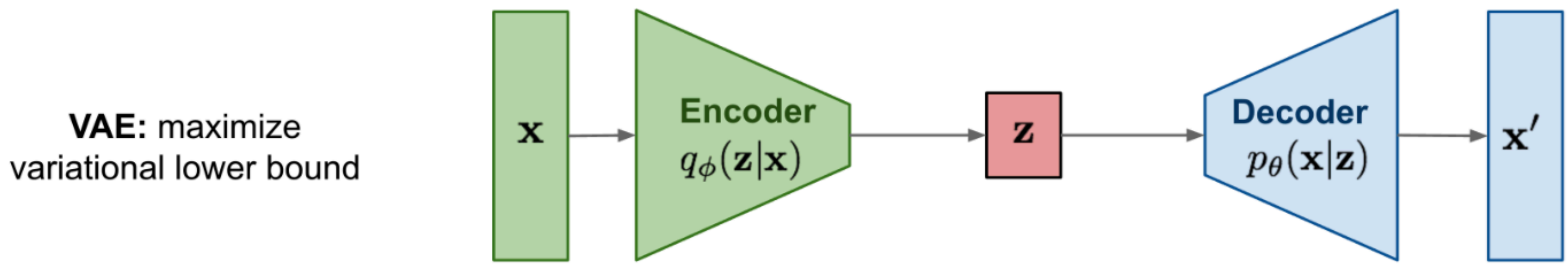
☹️ Mode collapse

Paganini, et al, Phys. Rev. D 97, 014021 (2018)
Fauuci et al, arXiv:2309.06515
Ratnikov et al, arXiv:2207.04340

Generative model: VAE

A summary blog

Variational Autoencoders: A pair of networks where one embed the data into a latent space with a given prior and the other decode back to the data space.



🙂 Structured latent representations 🙁 Less realistic outputs

Cresswell, et al, arXiv:2211.15380
Touranakou et al, arXiv:2203.00520
Abhishek et al, arXiv:2210.07430

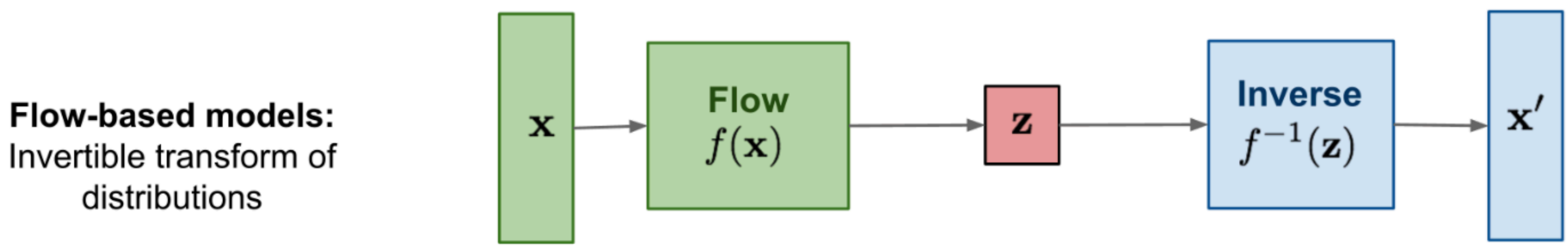
Generative model: NF

A summary blog

Normalising flow: invertible transformations to map a simple distribution to a complex one.

🙂 Exact likelihood computation
🙂 High generative capacity

☹️ Slow sampling



Diefenbacher et al, 2023 JINST 18 P10017
Pang et al, arXiv:2308.11700
Krause et al, PhysRevD.107.113003
Buckley et al, arXiv:2305.11934

Generative model: Diffusion model

A summary blog

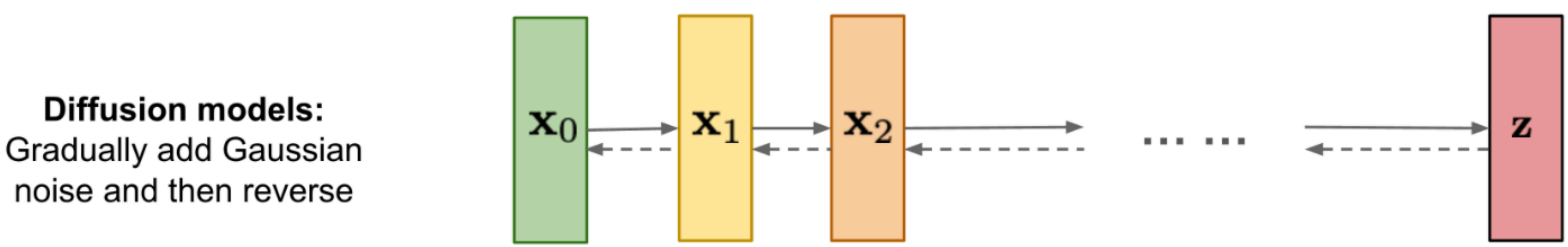
Diffusion model: gradually add Gaussian noise to input and learn the added noise using NN.

🙂 Strong generative performance

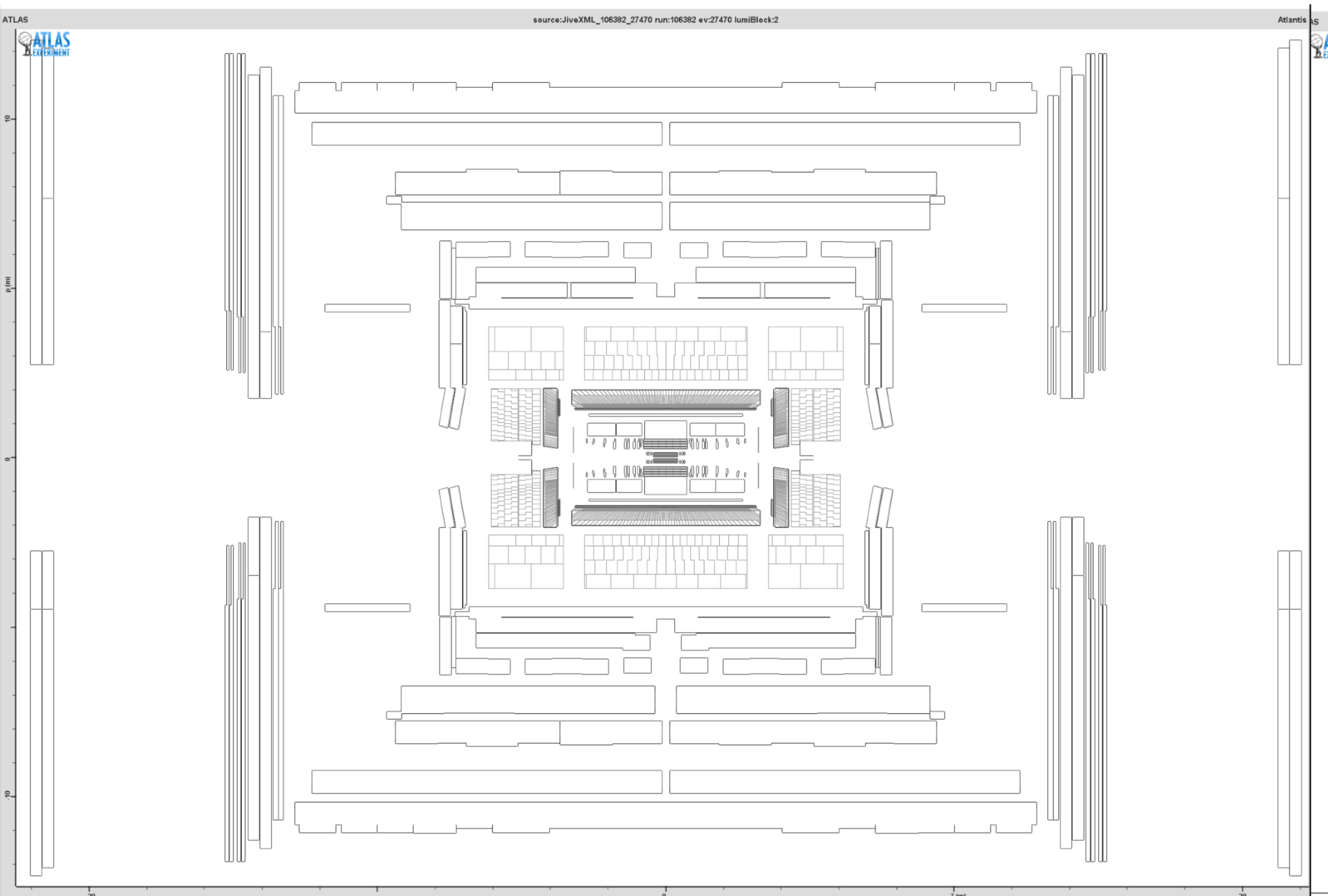
☹️ Slow sampling

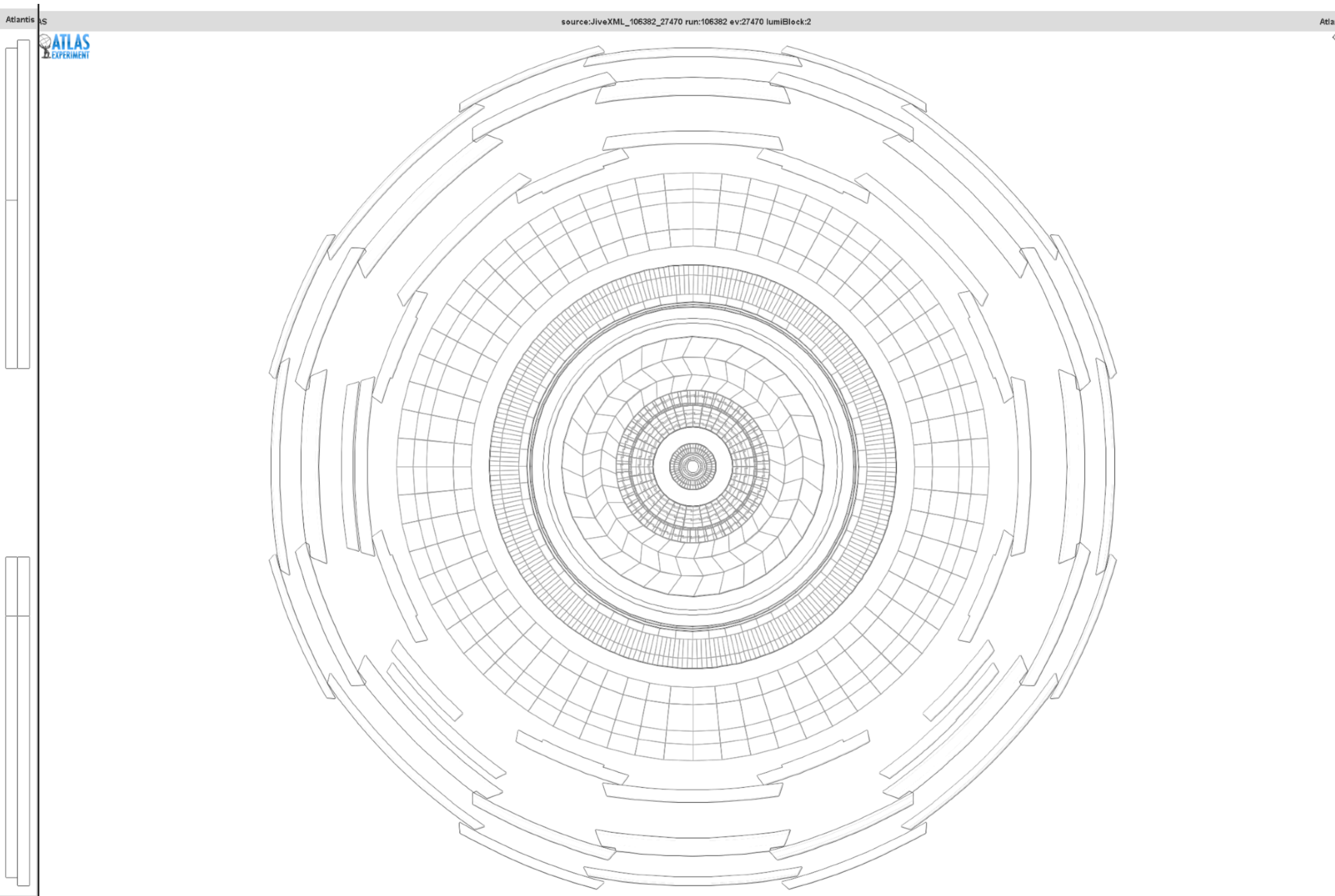
☹️ Difficult to train

Amram et al, Phys. Rev. D 108 (2023) 072014
Buhmann et al, arXiv:2309.05704
Mikuni et al, arXiv:2308.03847



Non-uniform ATLAS geometry



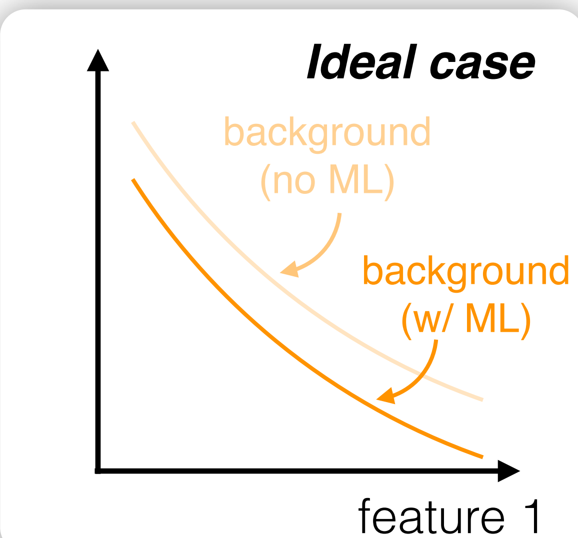


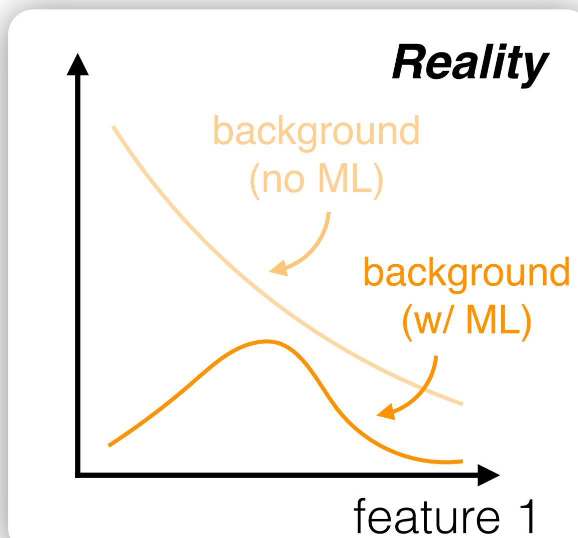
Decorrelation

Caution Part I

35

How can we learn a classifier that does not sculpt a bump in the background?





Enforcing Independence

39

Train e.g. a neural network with a **custom loss functional**

$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i),1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i),0) + \lambda \sum_{i\in b} L_{\text{decor}}(f(x_i),m_i)$$

Recent proposals:

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Nachman, Overview of Machine Jet Image Learning for Particle Physics