Machine Learning & Artificial Intelligence for Physics

Part 2: Applications & Advanced Models

Julia Gonski

17 July 2024
African School of Physics
Marrakech Morocco

Outline

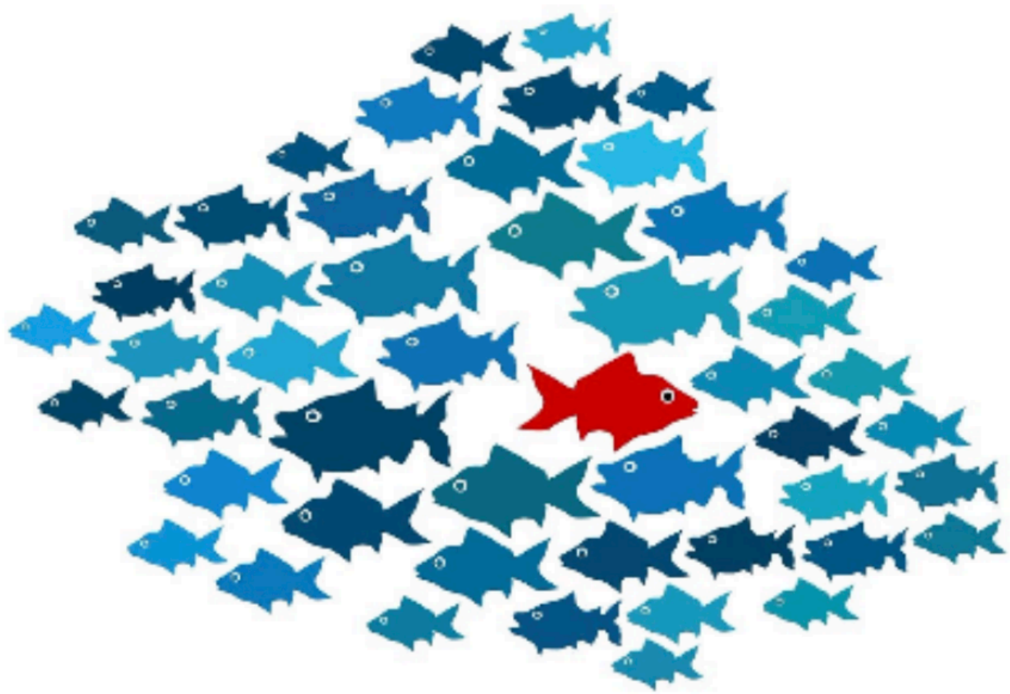
- Lecture 1: principles & key tools
  - What is AI/ML? How is it useful in physics research?
  - Basics of neural nets: architecture & development

- Lecture 2: applications & advanced models
  - Anomaly detection
  - Geometrical ML
  - Hardware acceleration

Anomaly Detection

Anomaly Detection

- Identify elements of the data that are inconsistent with a background-only model
- Don't need to know the features of your signal! Can be data-driven

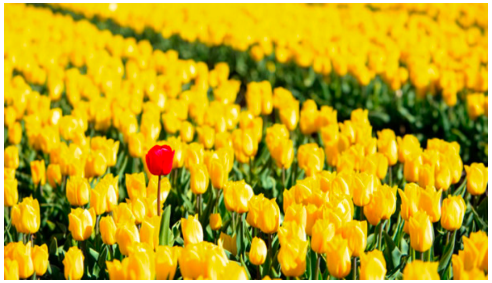


Less-Than Supervised Learning

- Unsupervised: train over unlabeled events
- Weakly supervised: noisy labels ("signal-enriched" instead of pure)
- Semi-supervised: partially labeled (some knowledge of signal model)

Autoencoders

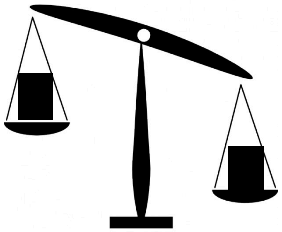
I know how to predict all collisions



Are there any collisions that I cannot predict?

Weakly-Supervised

I know regions where new physics does not exist

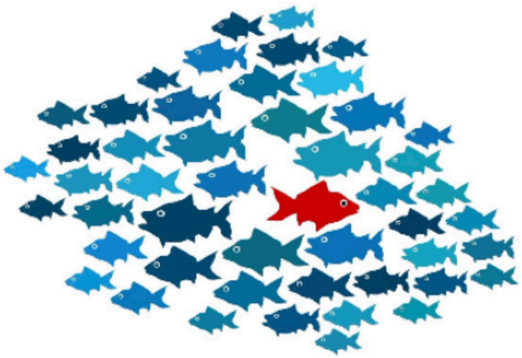


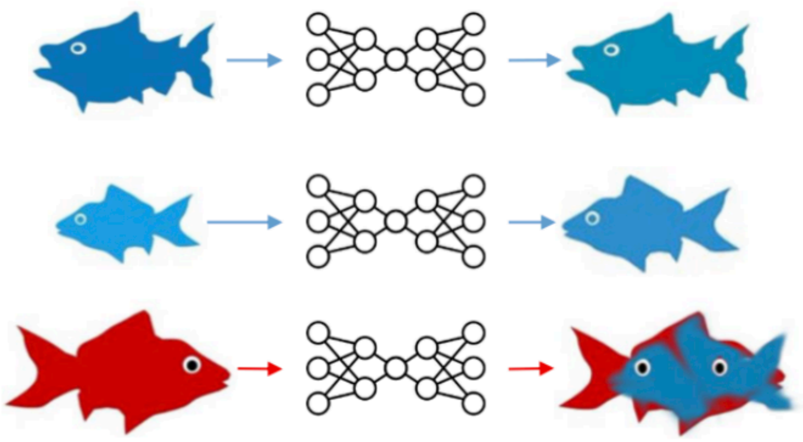
I want to leverage those regions against other parts of the data to find differences

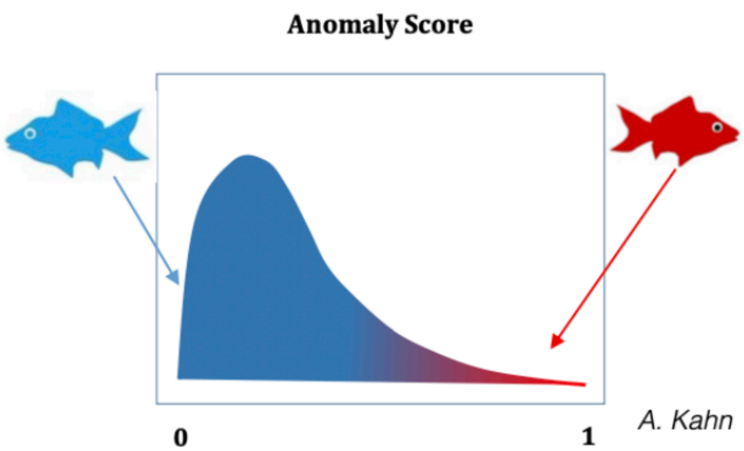
P. Harris

Unsupervised ML

- Train an ML architecture to reconstruct its input
  - Train over data = fully **unsupervised**
- Rarer events with unusual features will be poorly reconstructed → reconstruction accuracy is a good discriminant

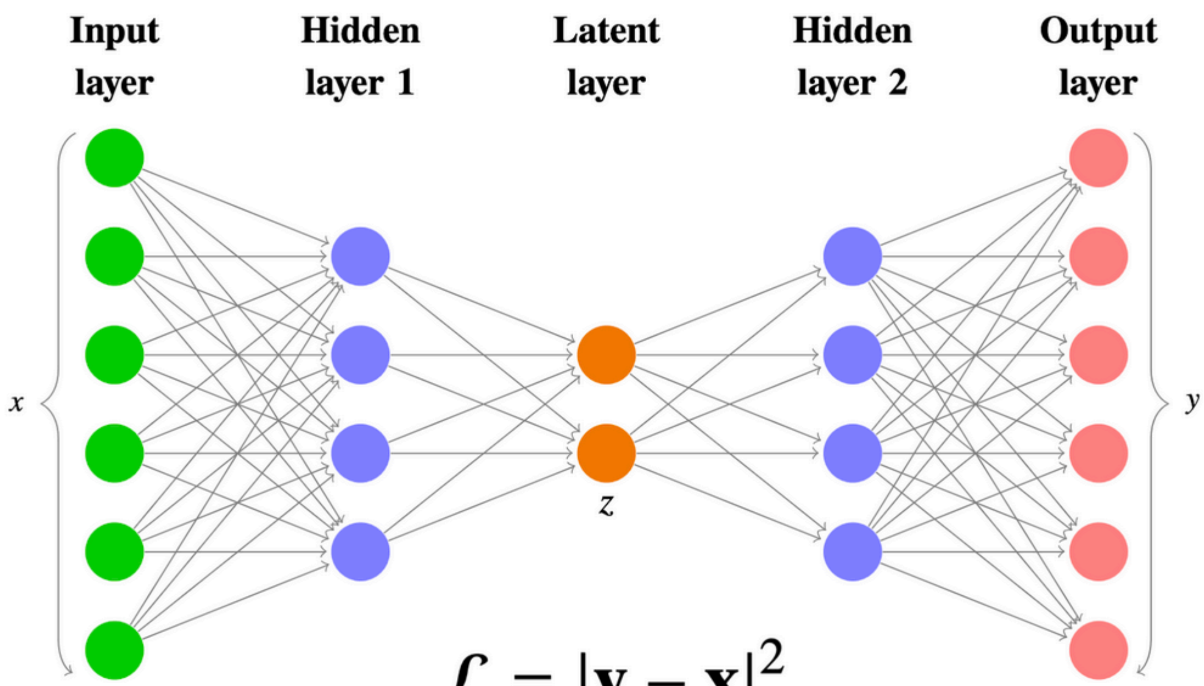






Autoencoders

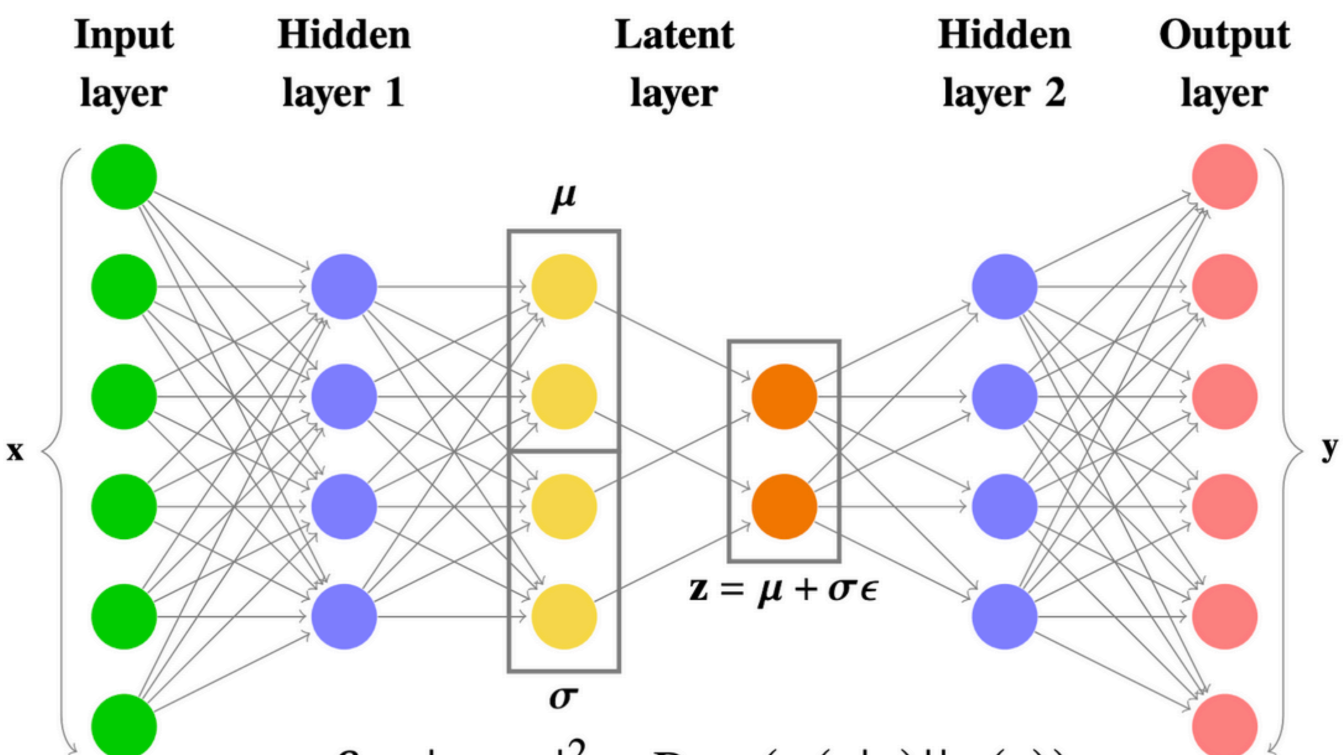
- Autoencoder: deep neural net with **encoder** and **decoder** stages
  - Lossy compression into *latent space* forces NN to encode most salient features
- Loss = mean squared reconstruction error between generated output and truth input



$$\mathcal{L} = |\mathbf{y} - \mathbf{x}|^2$$

Variational Autoencoders

- **Variational autoencoder**: autoencoder that can perform Bayesian inference
  - Latent space is continuous distributions, not single points
- Loss includes **Kullback-Leibler divergence** term: keeps the approximating distribution *q* close to that of the truth *p*



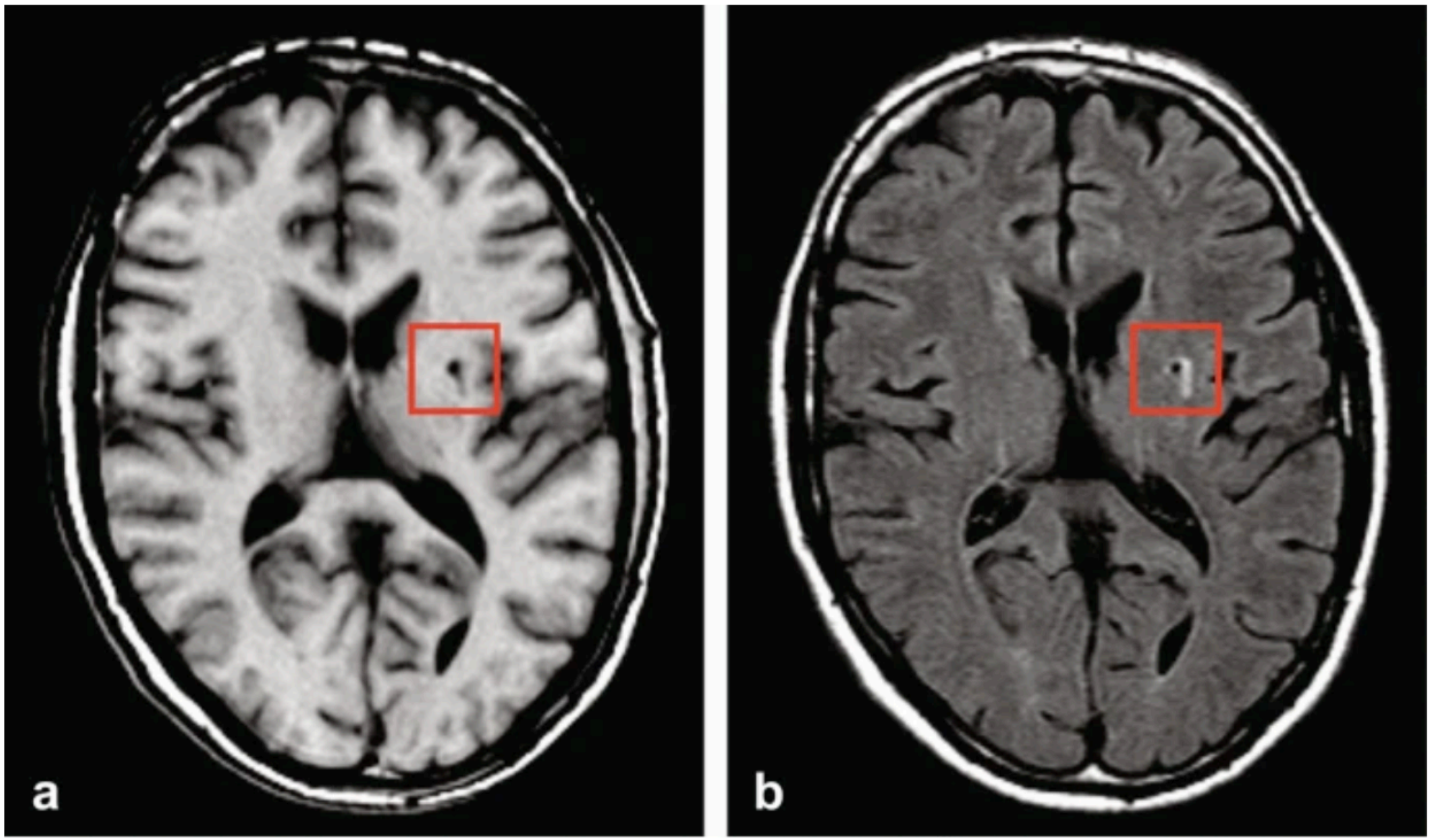
$$\mathcal{L} = |\mathbf{y} - \mathbf{x}|^2 + D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

Anomaly Detection in Real Life

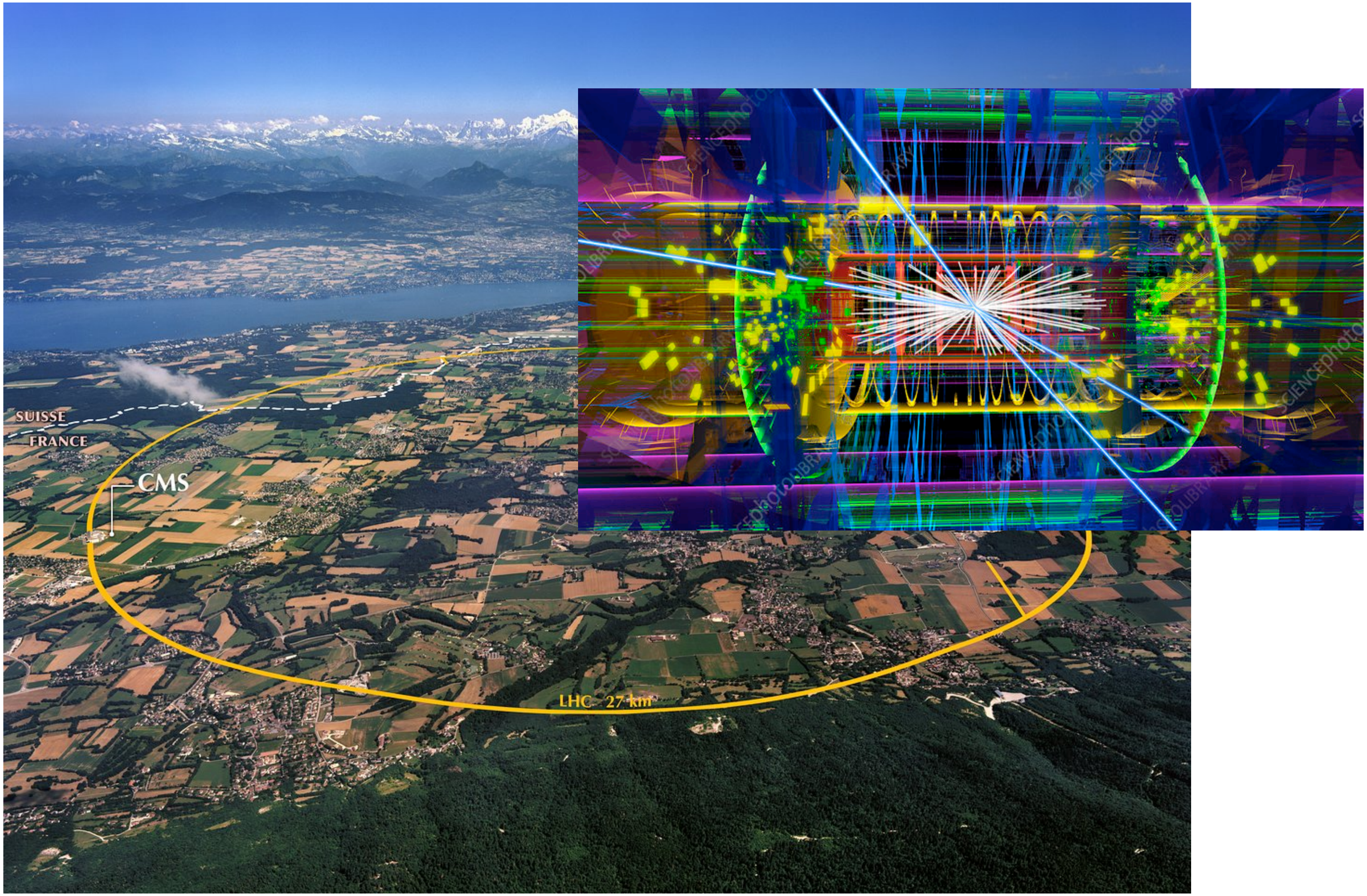
- **Finance:** can detect "pump and dump" trading patterns in stock prices

Anomaly Detection in Real Life

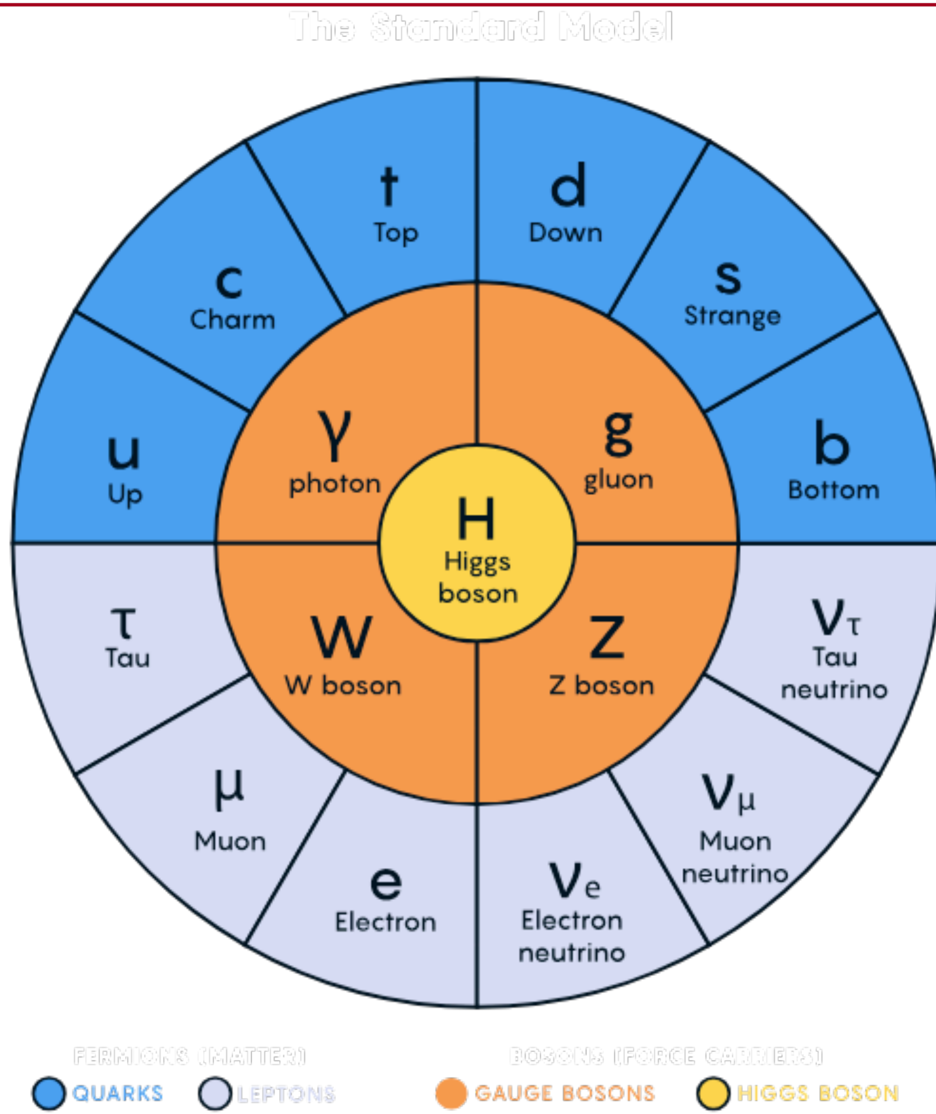
- **Medicine:** can automatically identify subtle issues in scans



Example: High Energy Physics

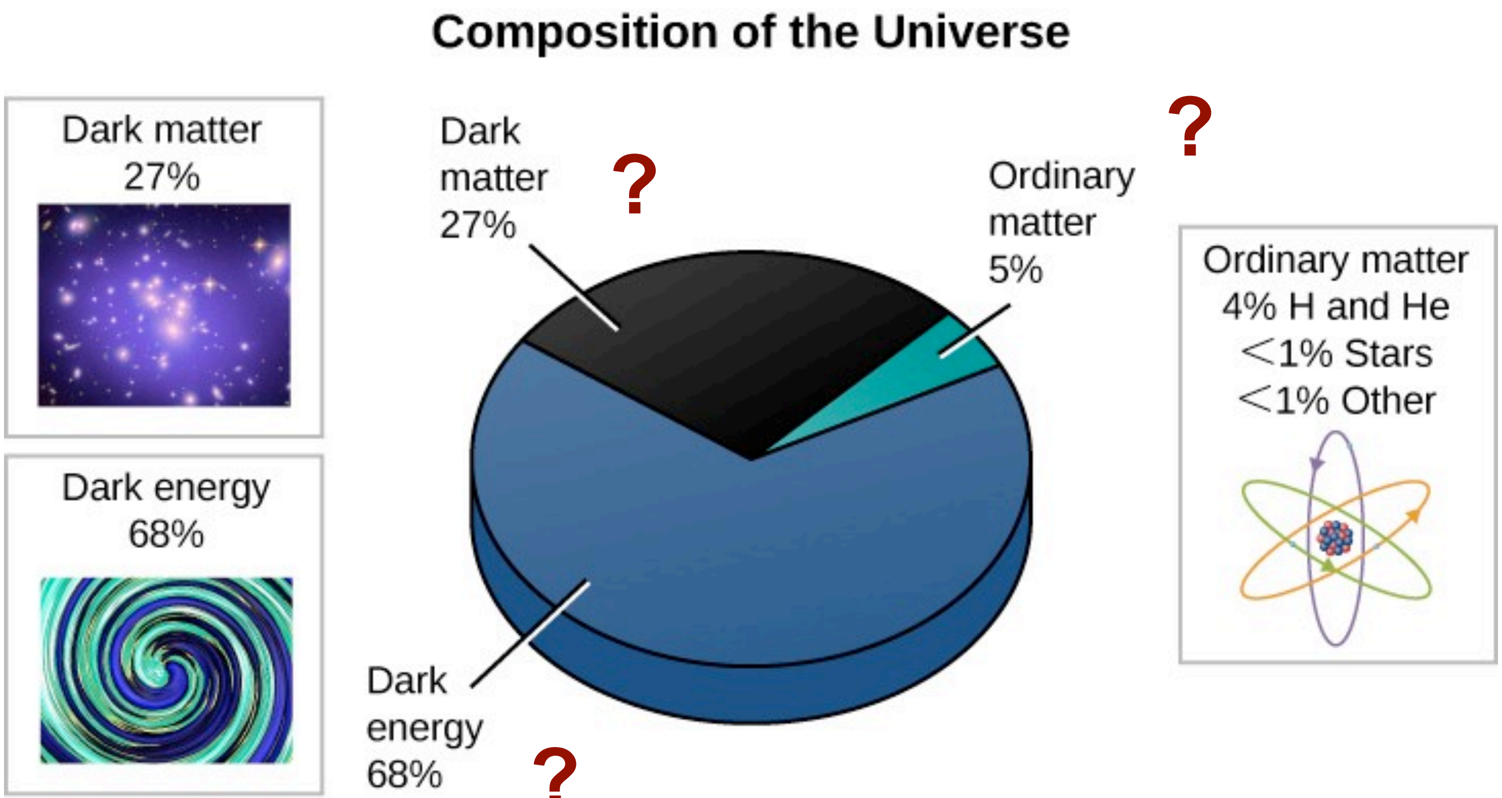


What We Understand So Far



Searching for "New" Physics

- Hoping to produce particles that can explain the unexplained in our universe….

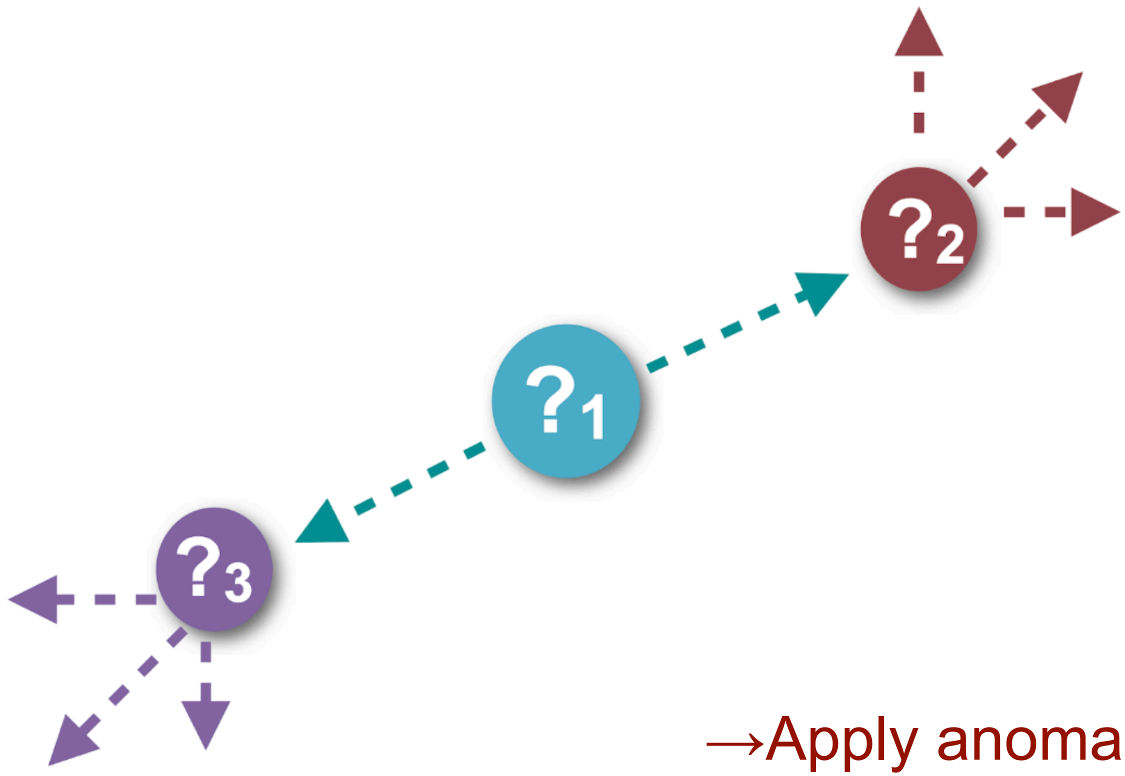


How to Search

- From a data science perspective, LHC analysis is a **signal-to-noise problem**
- Higgs bosons are produced in **1 out of 10 billion** proton collisions

But What Are We Looking For?

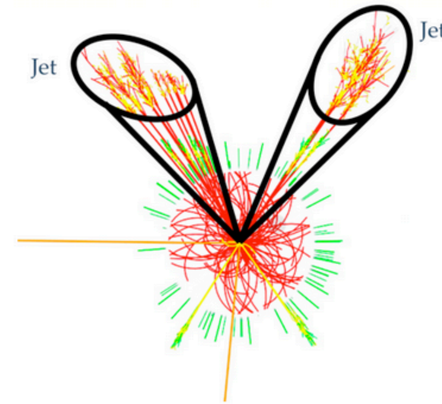
- New physics could be an **"unknown unknown"**: how do we design a way to enrich signal-to-noise if we don't know what our signal looks like?

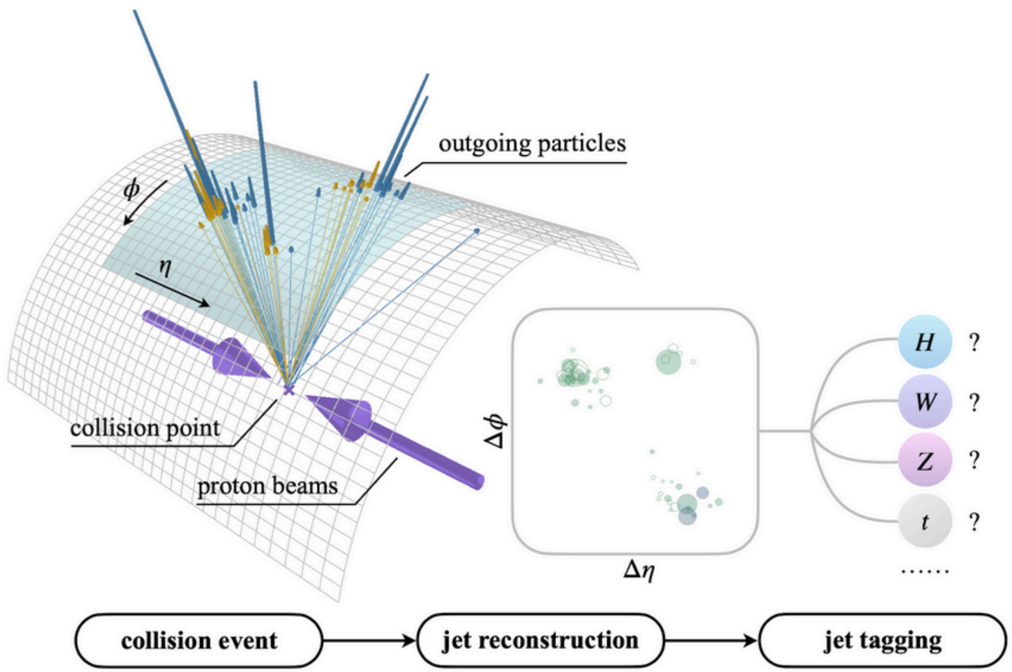


→Apply anomaly detection to LHC datasets

Anomaly Detection for Jets

- Jet: spray of hadronic particles emerging from proton collision
  - Multiple kinds of correlated inputs (tracks + calorimeter energy clusters)
  - High-dimensional: O(100s) particles in each jet, clustered

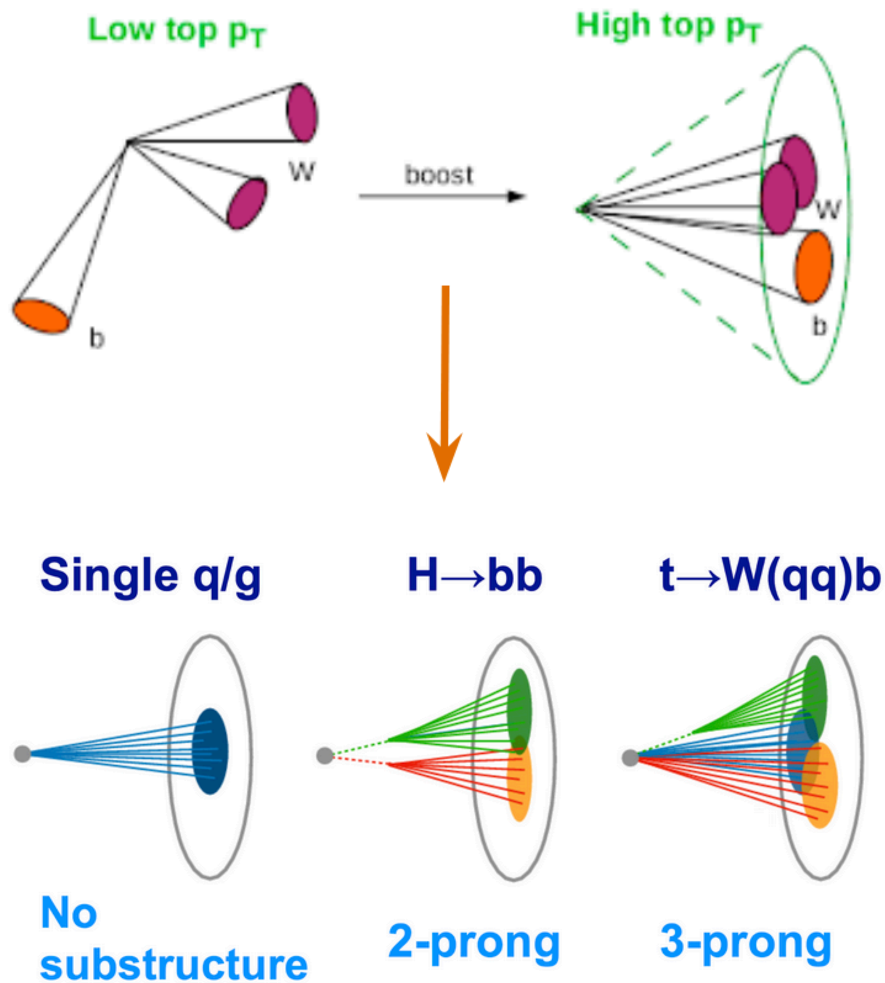




15

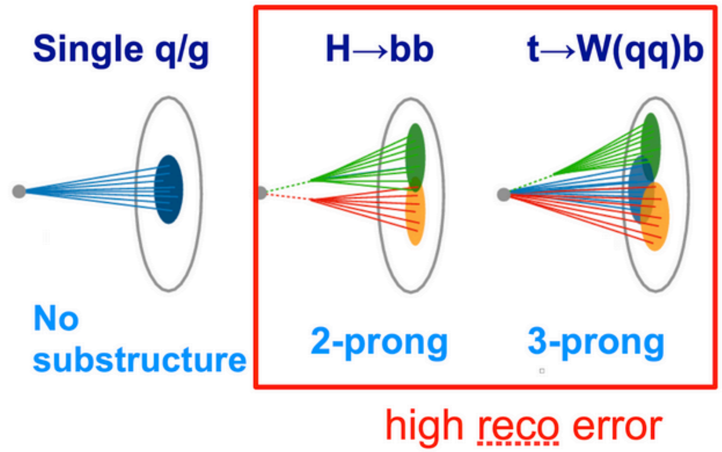
Input Modeling for Jets

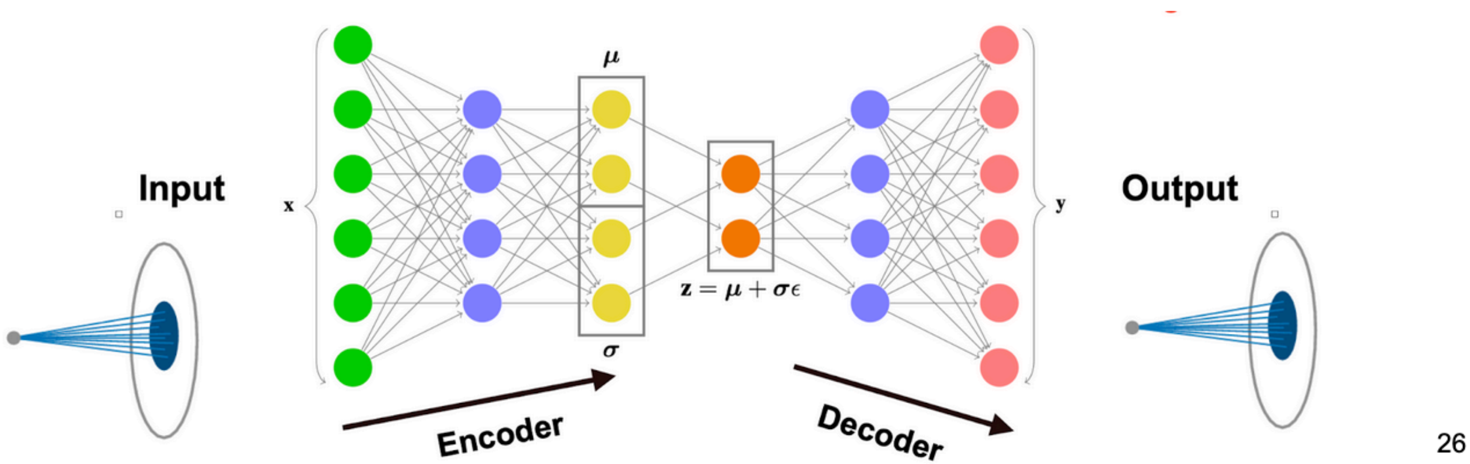
- Made of constituent particles (each of which can be described by their four-vectors)
- If a jet is produced with considerable energy, its decay is collimated, meaning the constituent particles overlap
  - Within the jet cone you can detect **substructure**



VAEs for Jet Identification

- Train over jets from LHC data modeled by constituent 4-vectors
- VAE learns background distribution, and so can **identify any event that doesn't look like the Standard Model**

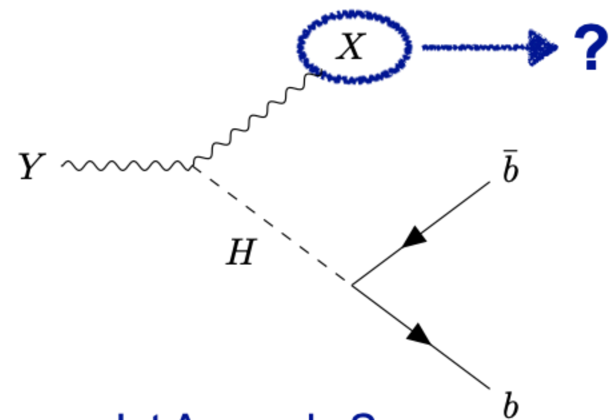




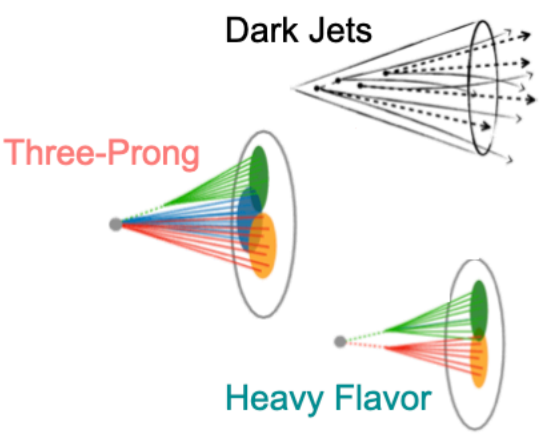
Looking for New Physics with VAEs

- Anomaly detection tool trained without labels over full Run 2 dataset of high momentum large-R jets
  - Per-jet anomaly score provides signal-model-independent selection of anomalous X candidate jets

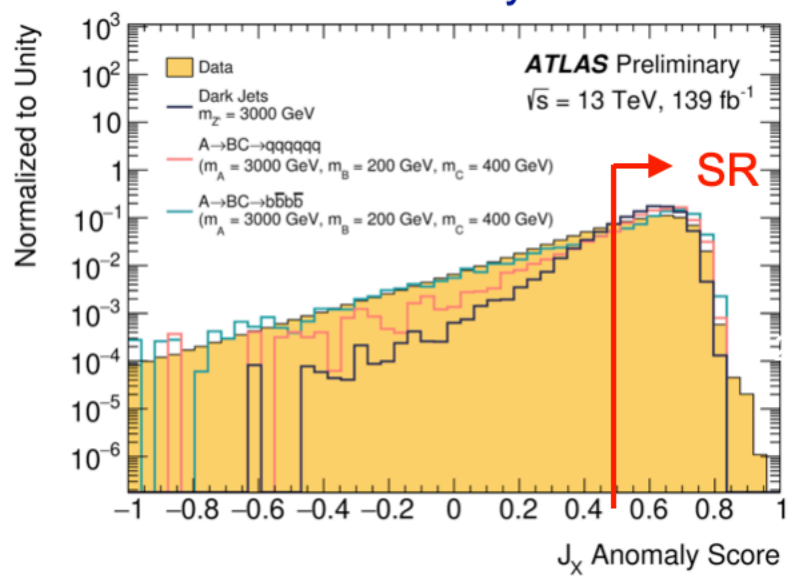
➡ **First application of fully unsupervised machine learning to an ATLAS analysis**



Possible Signal Jet Models



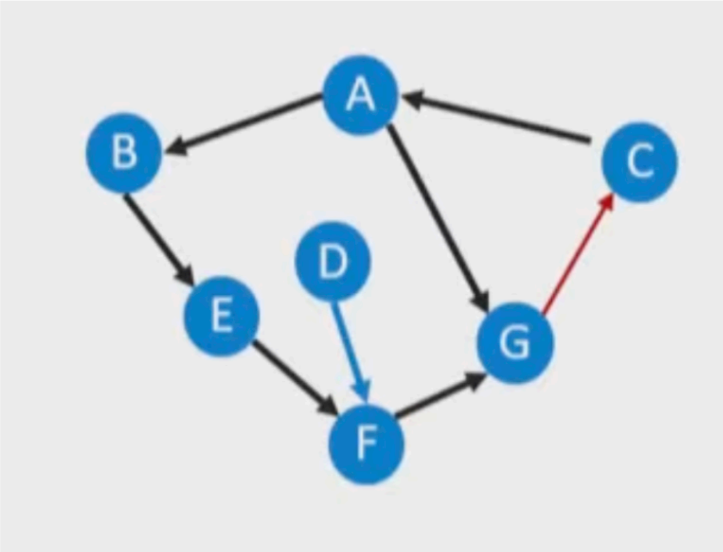
Jet Anomaly Score



Graph ML

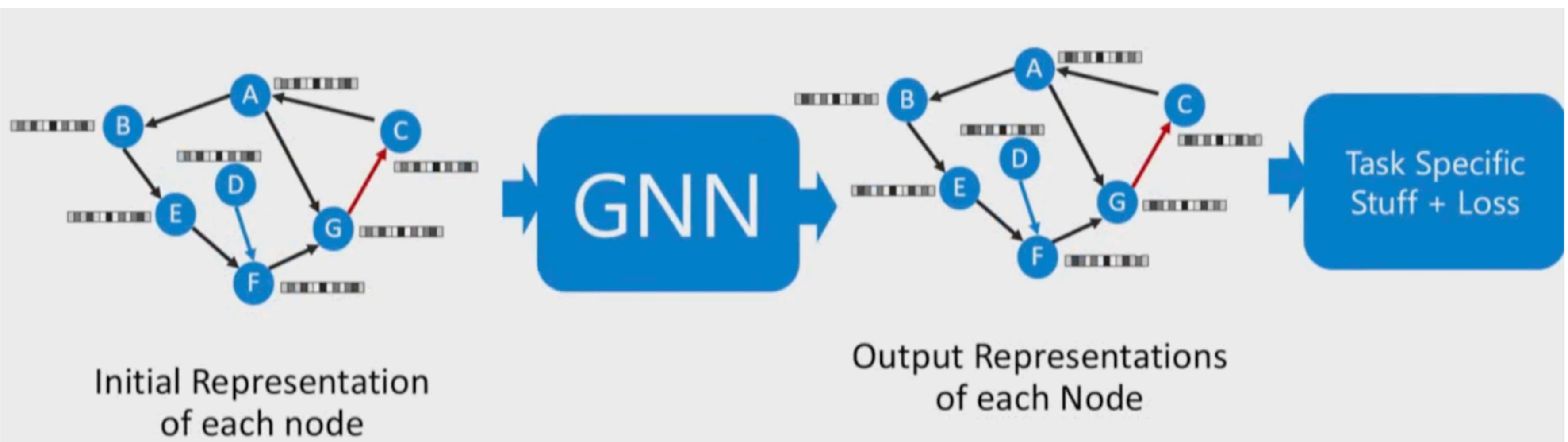
What are Graphs?

- Structure of data with multidimensional relationality & permutation invariance
- Nodes/vertices connected by edges
- Graphs can be used to model data that is:
  - Distributed unevenly in space
  - Sparse
  - Variable size
  - No defined order of inputs
  - Interconnected



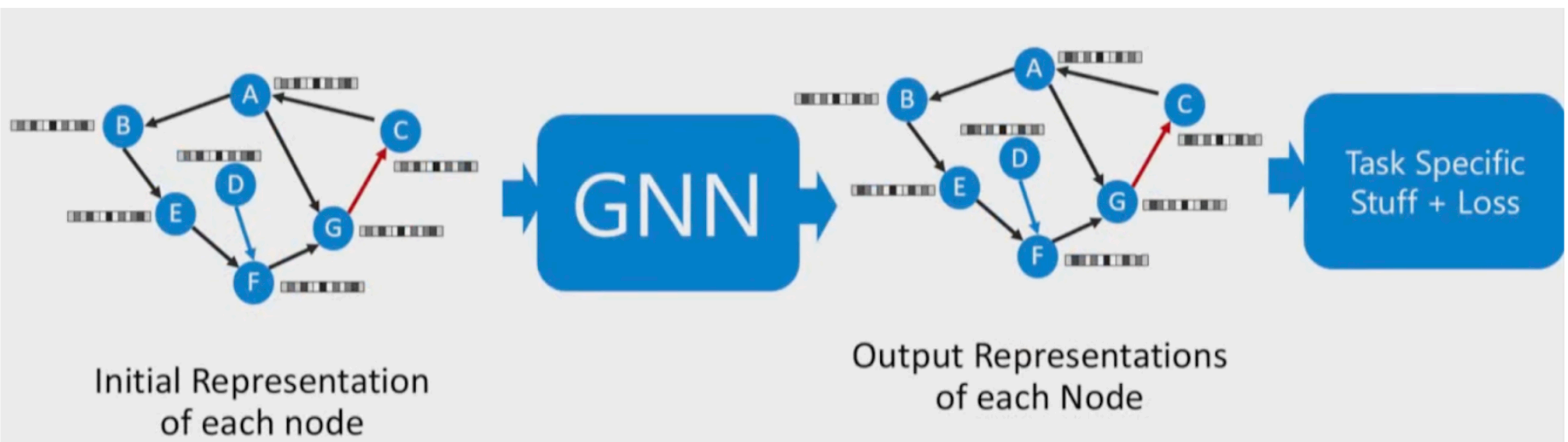
Graph Neural Networks

1. Each node/edge have features and are given an *embedding* in the graph



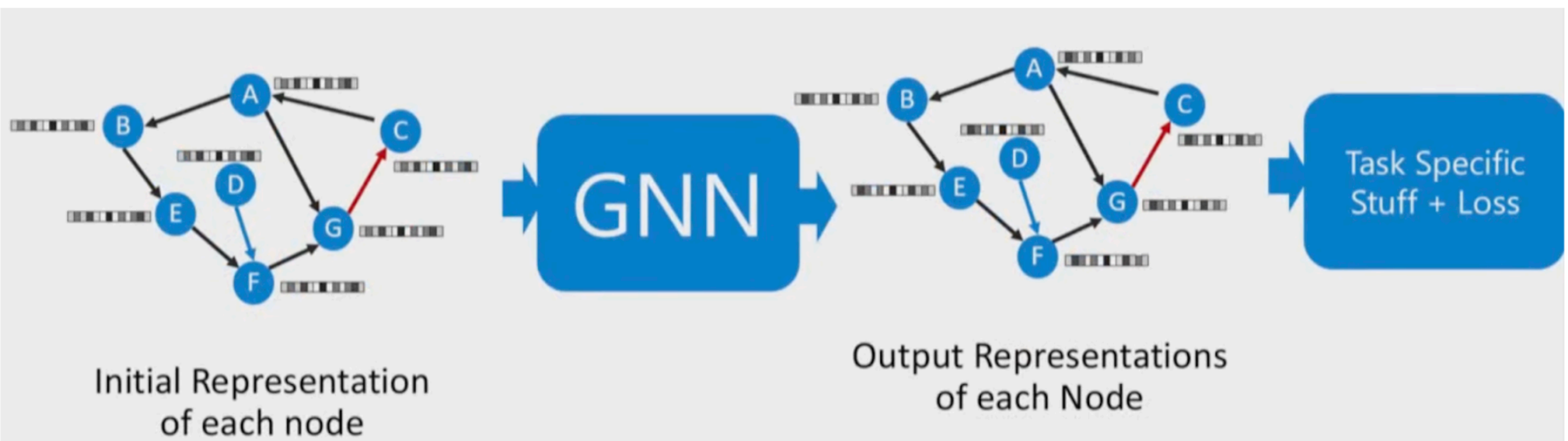
Graph Neural Networks

1. Each node/edge have features and are given an *embedding* in the graph
2. **Training:** “message-passing” to update node/edge embeddings using neighboring information
    a. Must preserve graph symmetry (eg. permutation invariance)



Graph Neural Networks

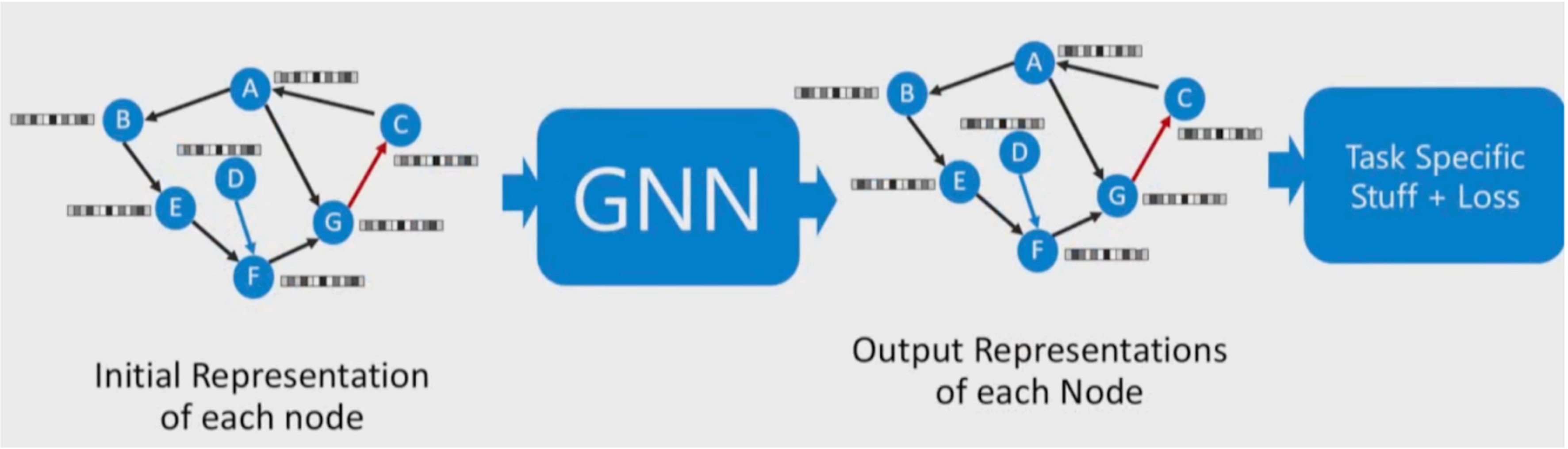
1. Each node/edge have features and are given an *embedding* in the graph
2. **Training:** “message-passing” to update node/edge embeddings using neighboring information
   a. Must preserve graph symmetry (eg. permutation invariance)
3. **Evaluation**: can ask questions about a node (what role does this node have?), an edge (what is the nature of the relationship?), or the global graph (what is the nature of this event?)



Graph Neural Networks

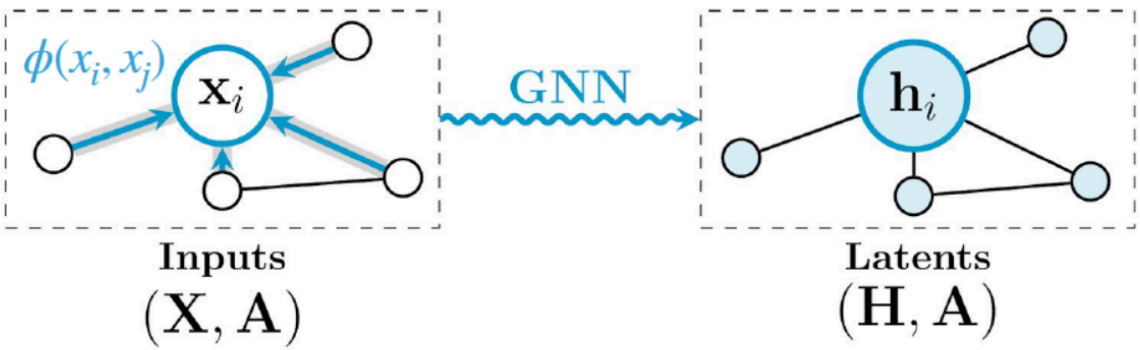
1. Each node/edge have features and are given an *embedding* in the graph
2. **Training:** “message-passing” to update node/edge embeddings using neighboring information
   a. Must preserve graph symmetry (eg. permutation invariance)
3. **Evaluation**: can ask questions about a node (what role does this node have?), an edge (what is the nature of the relationship?), or the global graph (what is the nature of this event?)

→ Scales better for larger data than other algorithms



Graph Neural Networks

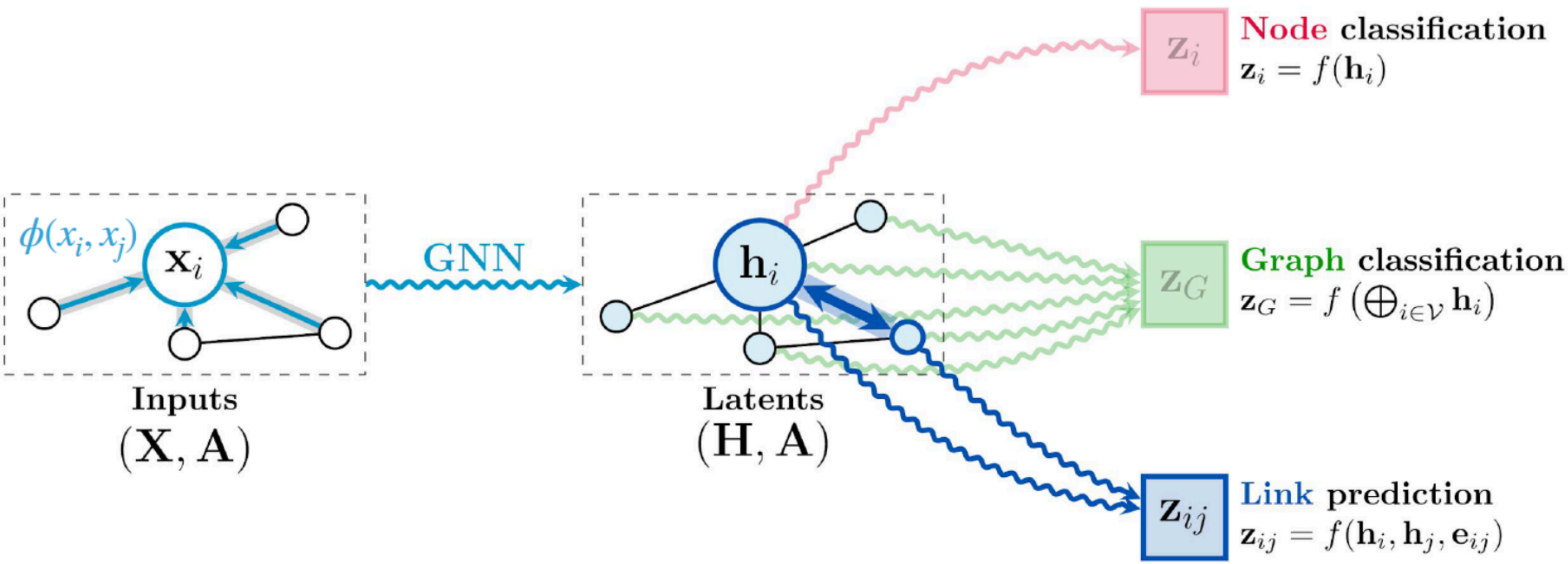
- For all neighbors j of node i compute a "message" via a NN: $\phi(x_i, x_j)$
- Update the node features by summing all messages:
$$h_i = \sum_j \phi(x_i, x_j)$$



"message passing"

J. Duarte

Graph Neural Networks

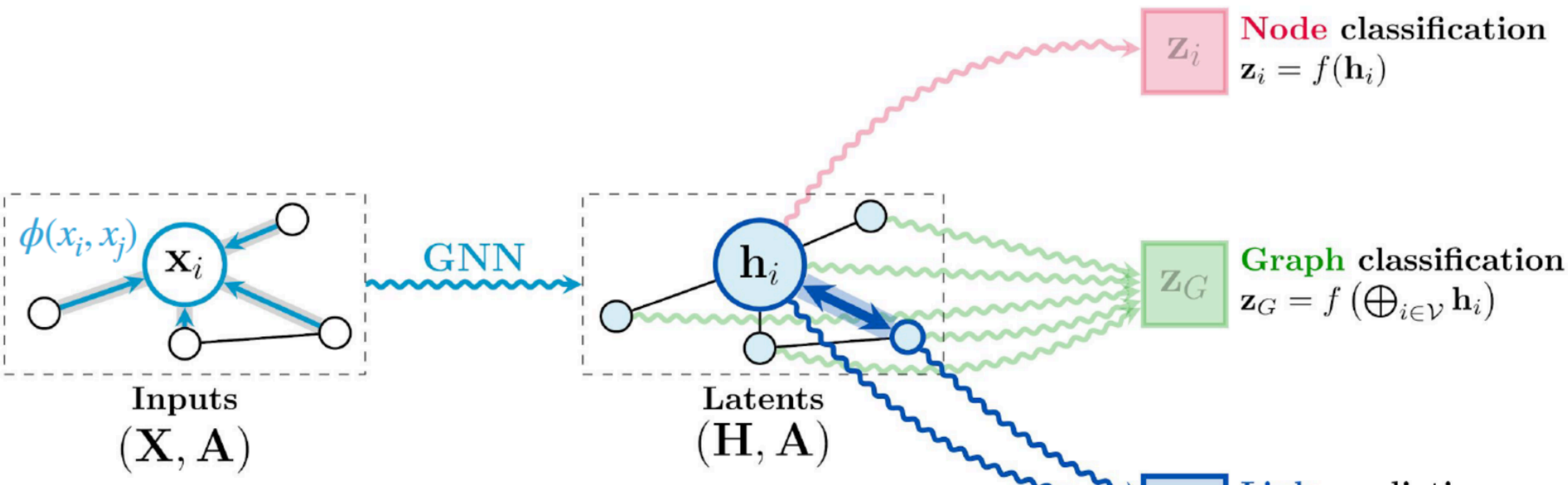


J. Duarte

Graph Neural Networks

Lots of uses for jets!

- Node-level tasks
  - Identify "pileup" particles
- Graph-level tasks
  - **Jet tagging**



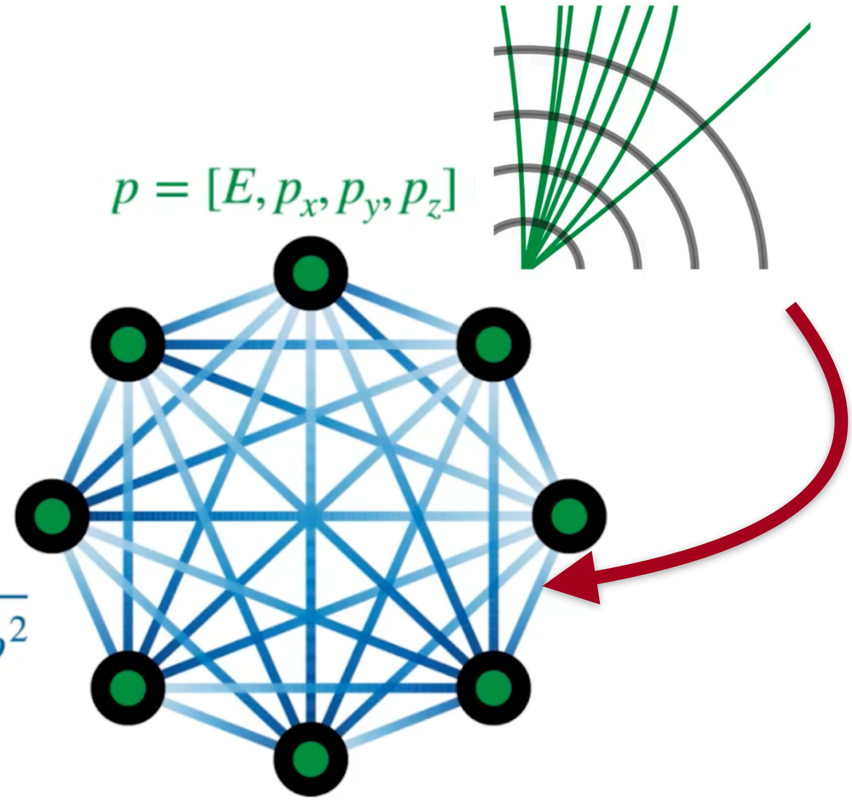
- Edge-level tasks
  - Identify good track segments

J. Duarte

GNNs for Jets

- Node features $\mathbf{v}_i$: particle 4-momentum
- Edge features $\mathbf{e}_k$: pseudoangular distance between particles
- Graph (global) features $\mathbf{u}$: jet mass

$$p = [E, p_x, p_y, p_z]$$

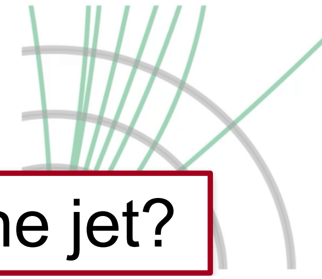


$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$$

$$m = \sqrt{\sum_{i \in \text{jet}} E_i^2 - p_{x,i}^2 - p_{y,i}^2 - p_{z,i}^2}$$

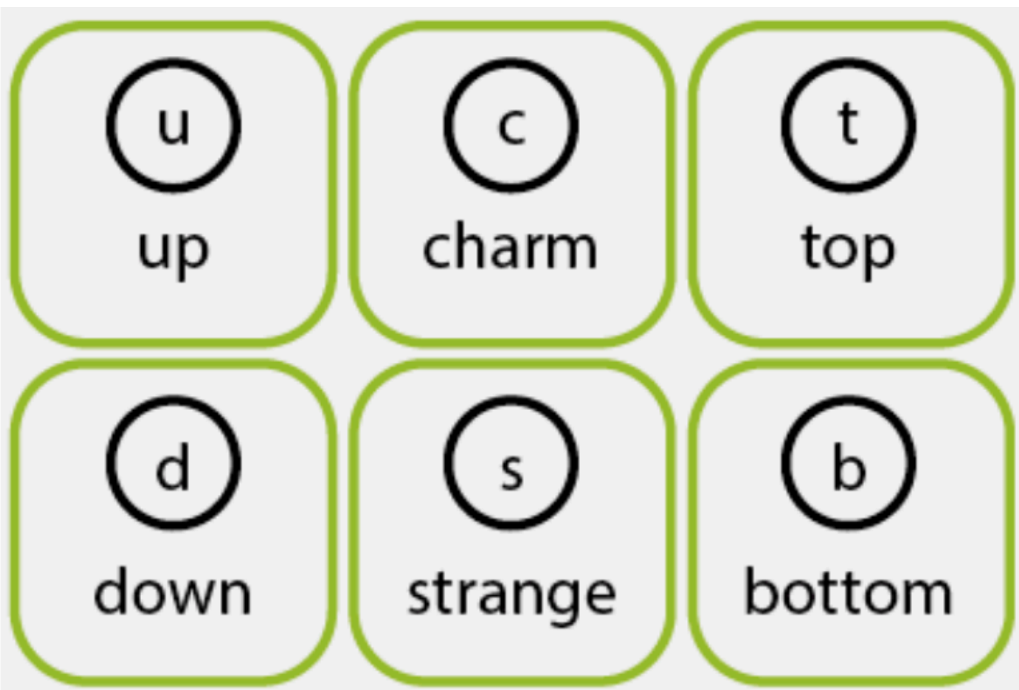
J. Duarte

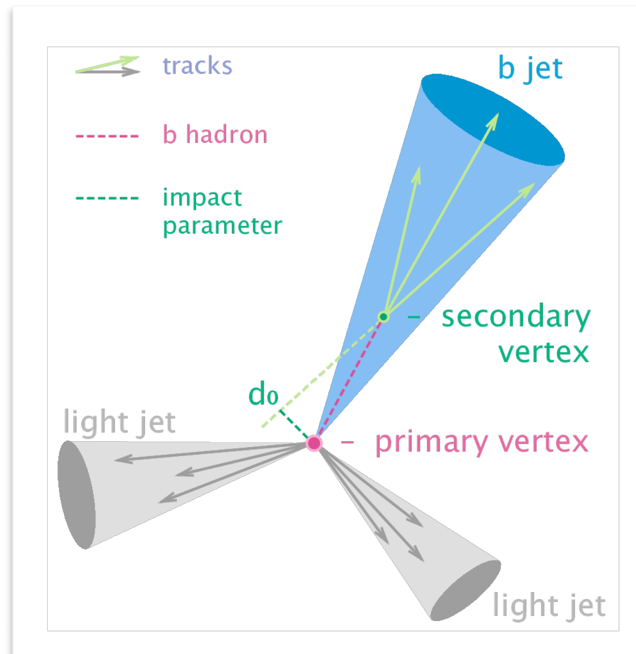
GNNs for Jets



- Nod

Jet-level question: which quark originated the jet?

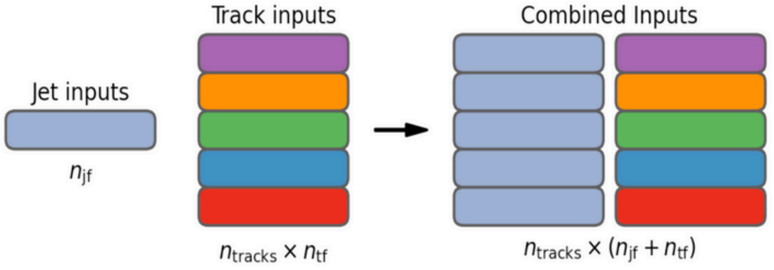


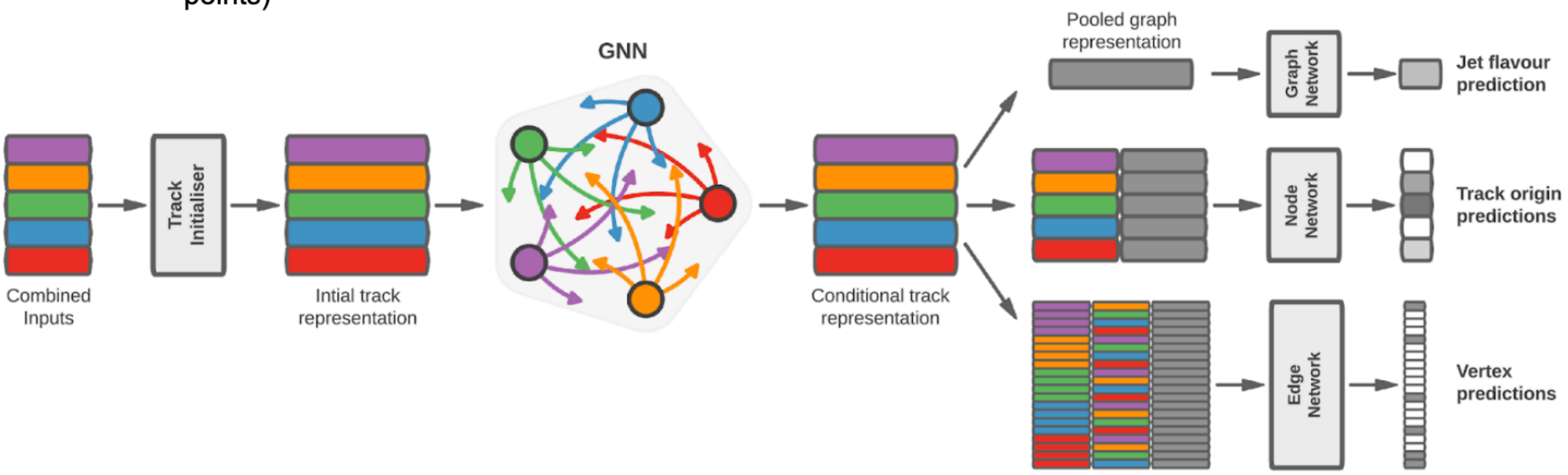


J. Duarte

Example: GNN for Jet Flavor Tagging

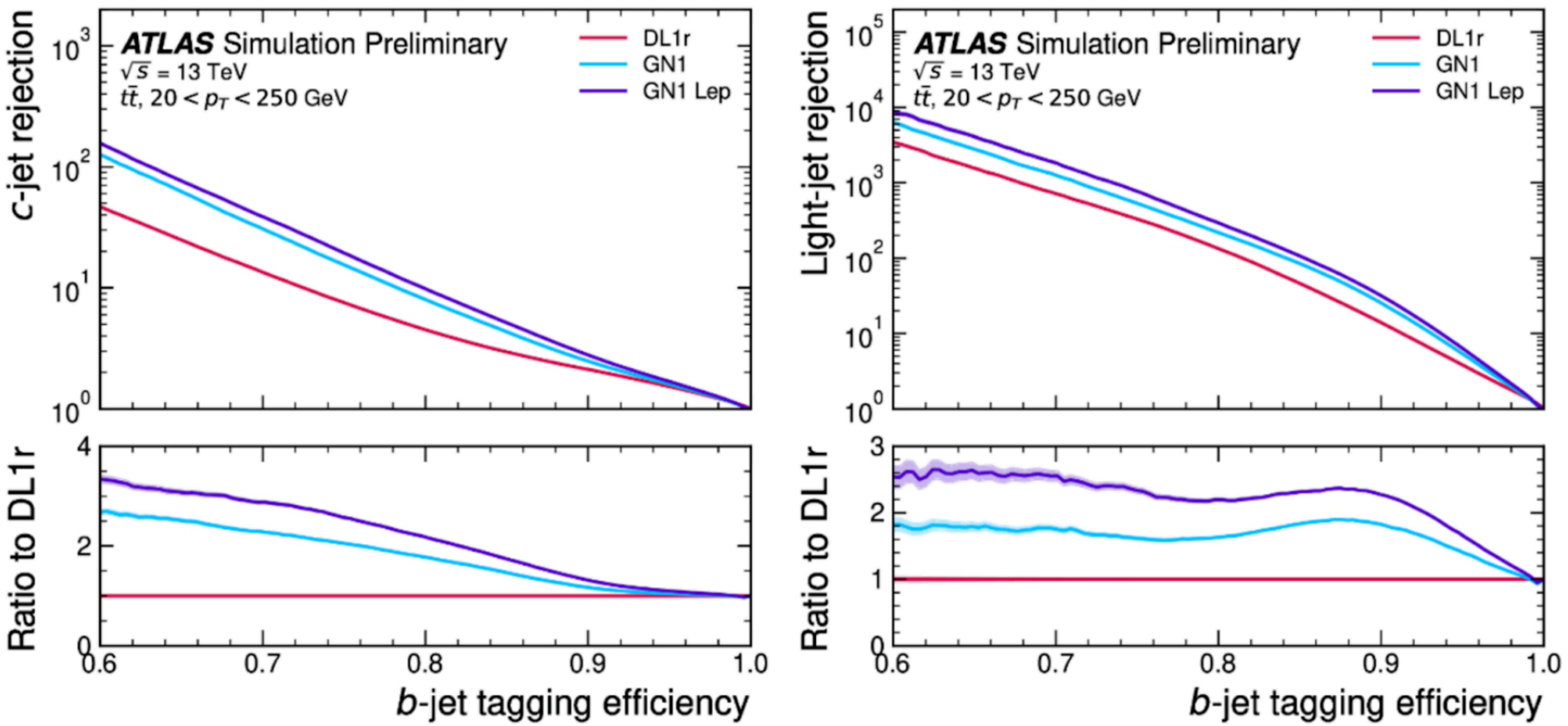
- How to build an algorithm that can identify jets from b-quarks from other (boring) "light" quarks?
- Use graph trained over constituent *tracks* in the jet
  - Very low-level inputs, high-dimensional (21 features per track, 40 tracks)
- Graph usage:
  - Classification: predict whether jet came from b-quark or not
  - Reconstruction: groups tracks into *vertices* (shared origin points)





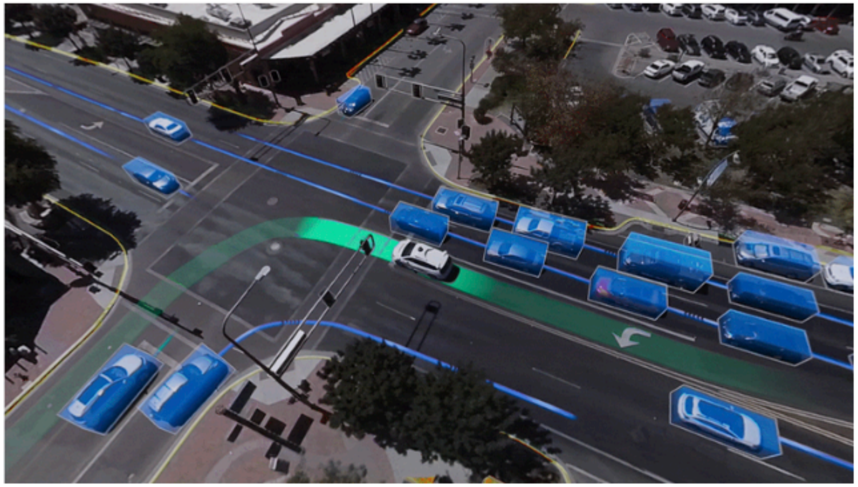
GNN Flavor Tagging Performance

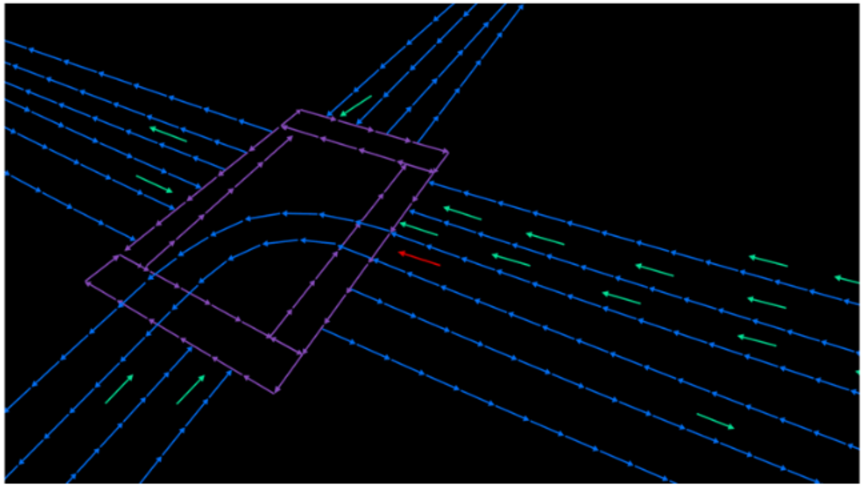
- Factor of 2-6 improvement in signal efficiency over simple high-level DNN
  - Just from a good choice of input modeling!



Graphs in Real Life

- Self-driving car company Waymo uses a **hierarchical graph ML model** to model object trajectories as a function of time and predict interactions between them





Ref.

Fast ML & Hardware Accelerators

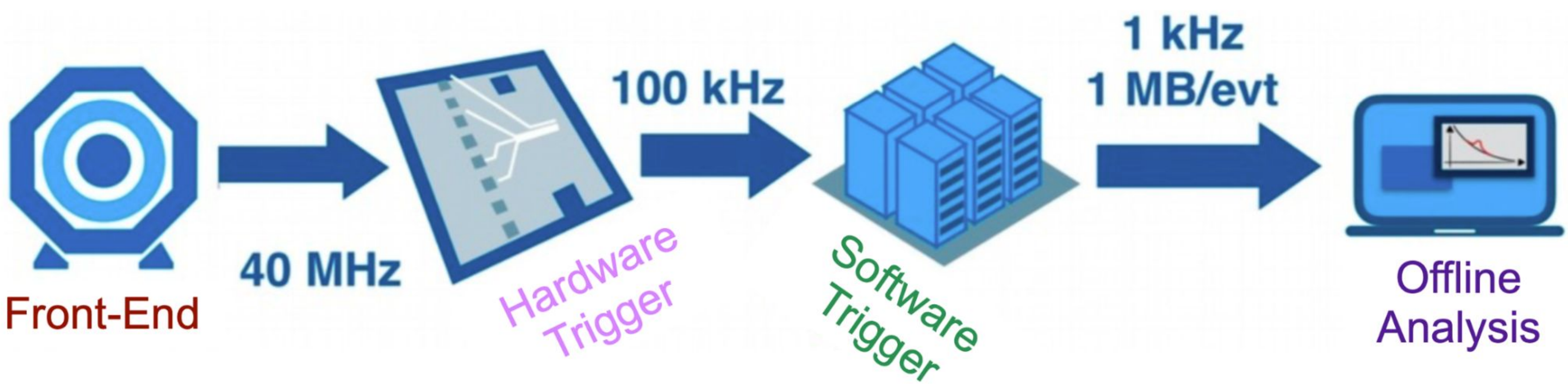
Extreme Environments in Physics

- Environments of high energy physics experiments are **"extreme"**
  - Very high radiation doses
  - Extreme temperatures (cryogenic)
  - Very high data rates/density
  - Spatial constraints (no room for cooling)
  - Very low latencies (eg. collisions every 25 ns…)

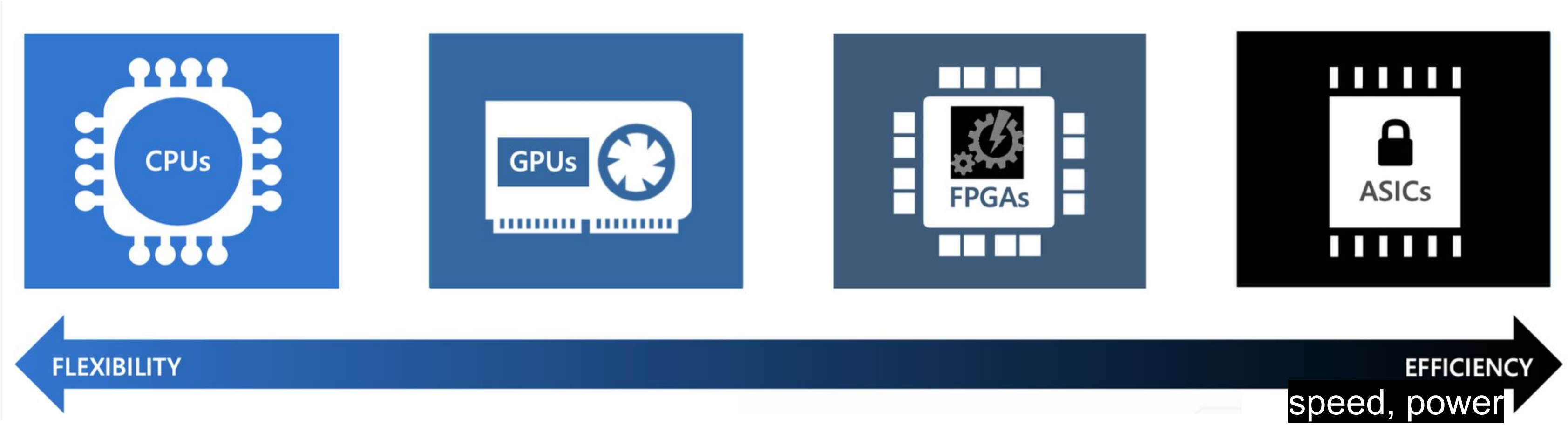
- Acquiring data from experiments requires performant inference (classification, regression):
  - Can benefit from **machine learning** throughout data acquisition systems

Fast Machine Learning

- **“Fast ML”** = hardware acceleration of ML algorithms running in software
  - Lower power, smaller footprint, faster inference time
  - Allows for advanced ML algorithms to run within collider data acquisition/triggering scheme
- **Latency** = time between starting processing and receiving the result
  - GPUs can only get you down to O(ms)
  - But we need much faster!
- Ex. LHC: front-end readout has O(ns) latency and hardware trigger O(μs)



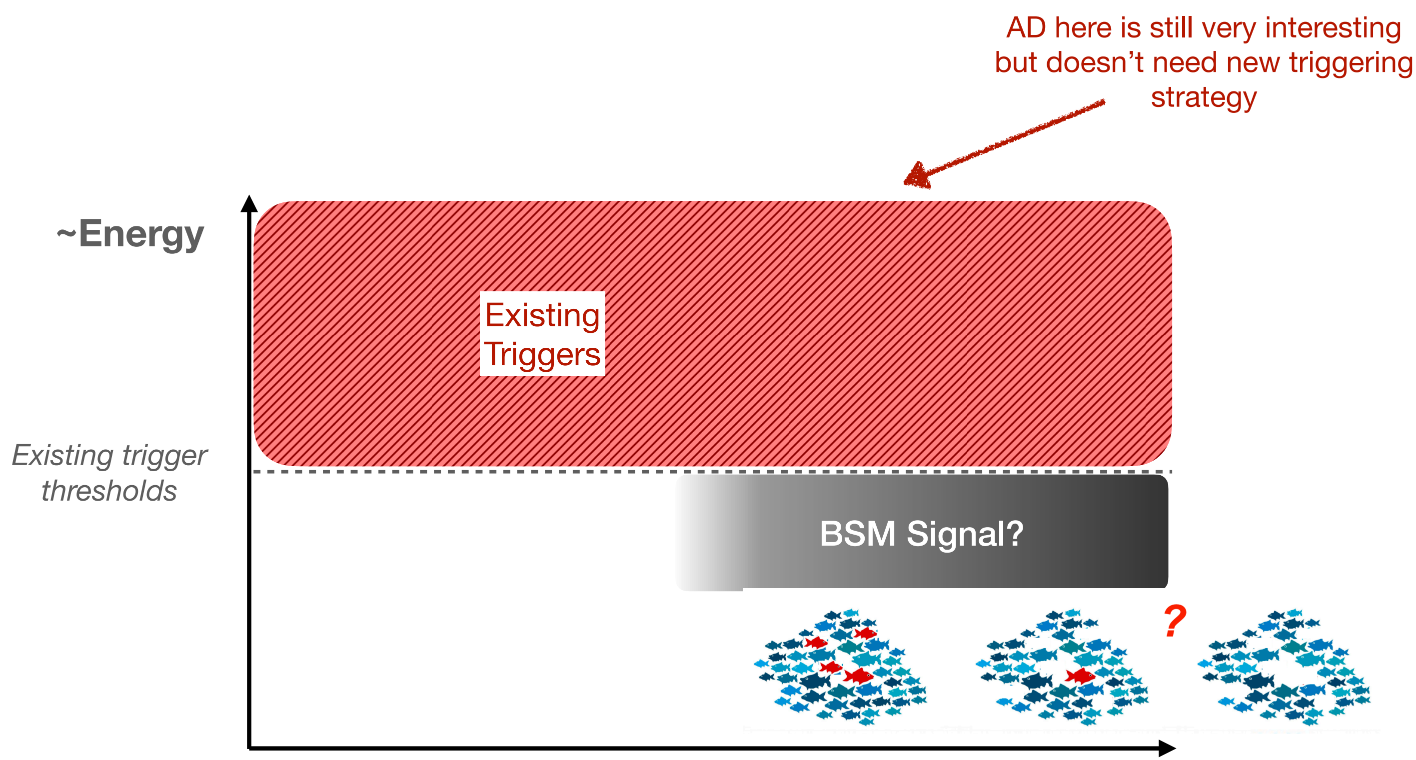
Computing Structures



- **Software (CPUs)**: total flexibility, can be reprogrammed as much as you want (eg. you can switch from a word processor to a photo editor)
- **Firmware (FPGA)**: instruction sets to interface hardware with operating system
- **Hardware (ASIC)**: physical components (features in silicon) that perform logical operations

Fast ML for Collider Triggers

- *Can I run anomaly detection in real-time to trigger on unusual events?*
  - Evaluate a VAE in < 25 ns?!

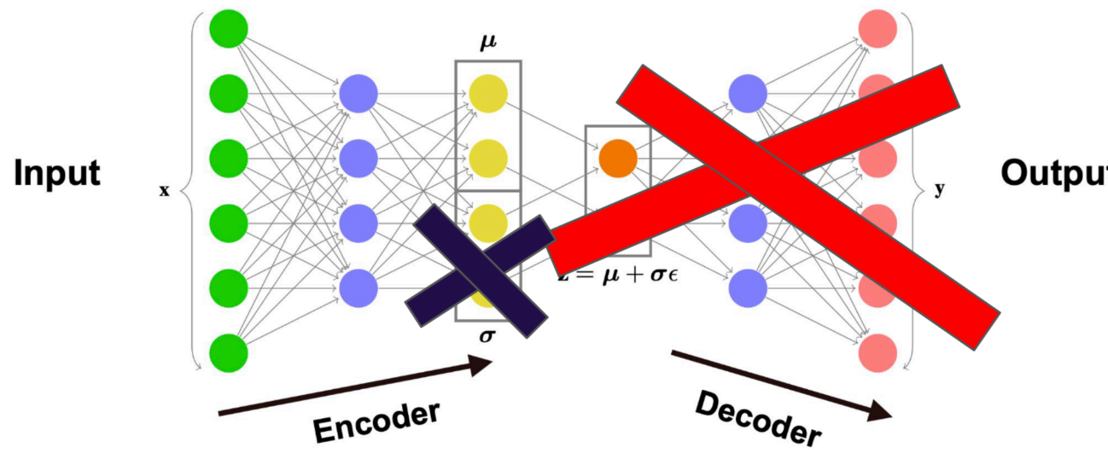


Fast ML for Collider Triggers

- *Can I run anomaly detection in real-time to trigger on unusual events?*
  - Evaluate a VAE in < 25 ns?!

- Tactics:
  - Pruning: remove unneeded nodes
  - Quantization: calculate with fewer bits per numbers
  - Minimize calculations: truncate loss function

"Truncated" VAE Cell



"Clipped" KL divergence

$$2 \cdot \mathrm{KL} = \mu^2 + \cancel{\sigma^2 - 1 - \log\sigma^2}$$

2108.03986

Recap

Recap

- **Anomaly detection** can leverage NNs to identify unusual elements in a dataset without a "signal model"
- **Geometrical ML** with graph neural networks can provide an apt input modeling for certain natural datasets and scale better to higher complexity
- **"Fast ML"** allows you to run ML evaluation faster by implementing ML algorithm in computing hardware
- These examples are all widely used across sciences and industry!

Conclusions

- AI/ML is rapidly advancing towards new and more complex models
  - Driven by increases in dataset sizes and computational power to accommodate large models
- When designing an AI/ML tool, think carefully about:
  - The best modeling of the input data
  - What kind of tasks (or tasks) you need to do
  - How complex your model needs to be (don’t bring a complex architecture to a simple problem!)
- Fundamental sciences can offer unique datasets and data processing challenges
  - The original “big data”: get valuable experience with the cutting edge of AI/ML, microelectronics/high performance computing, etc.
- AI/ML in science is fun! Think creatively, learn science, gain new transferable skills

Backup