STATISTICAL INFERENCE IN QUANTUM MACHINE LEARNING FOR JET IMAGE CLASSIFICATION

Antonio Greco

Physics and Advanced Technologies

University of Siena

24th October 2025

OVERVIEW

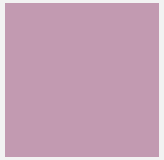
"What if the same tools we use to study the universe could help us understand the digital world?"

Physics

- Scientific Motivation
- Jet Classification

Neural Network

- QNN vs CNN
- Statistics
- Inference



IT-Sphere

- Cross-Domain Analogy
- IT-Sphere Events
- Accuracy &IT

Physics

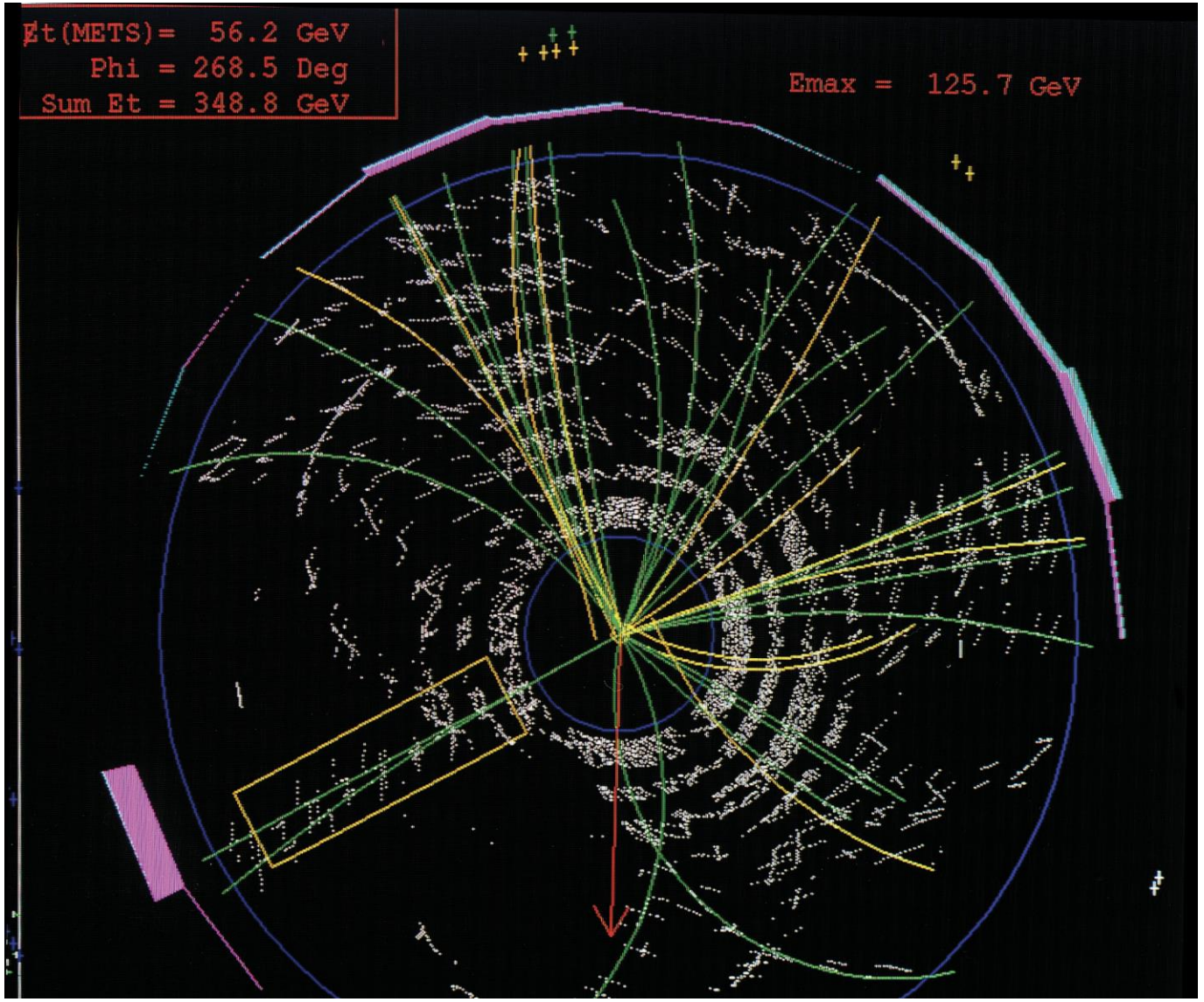
Scientific Motivation

Jet Classification

Top Quark vs QCD

JET CLASSIFICATION

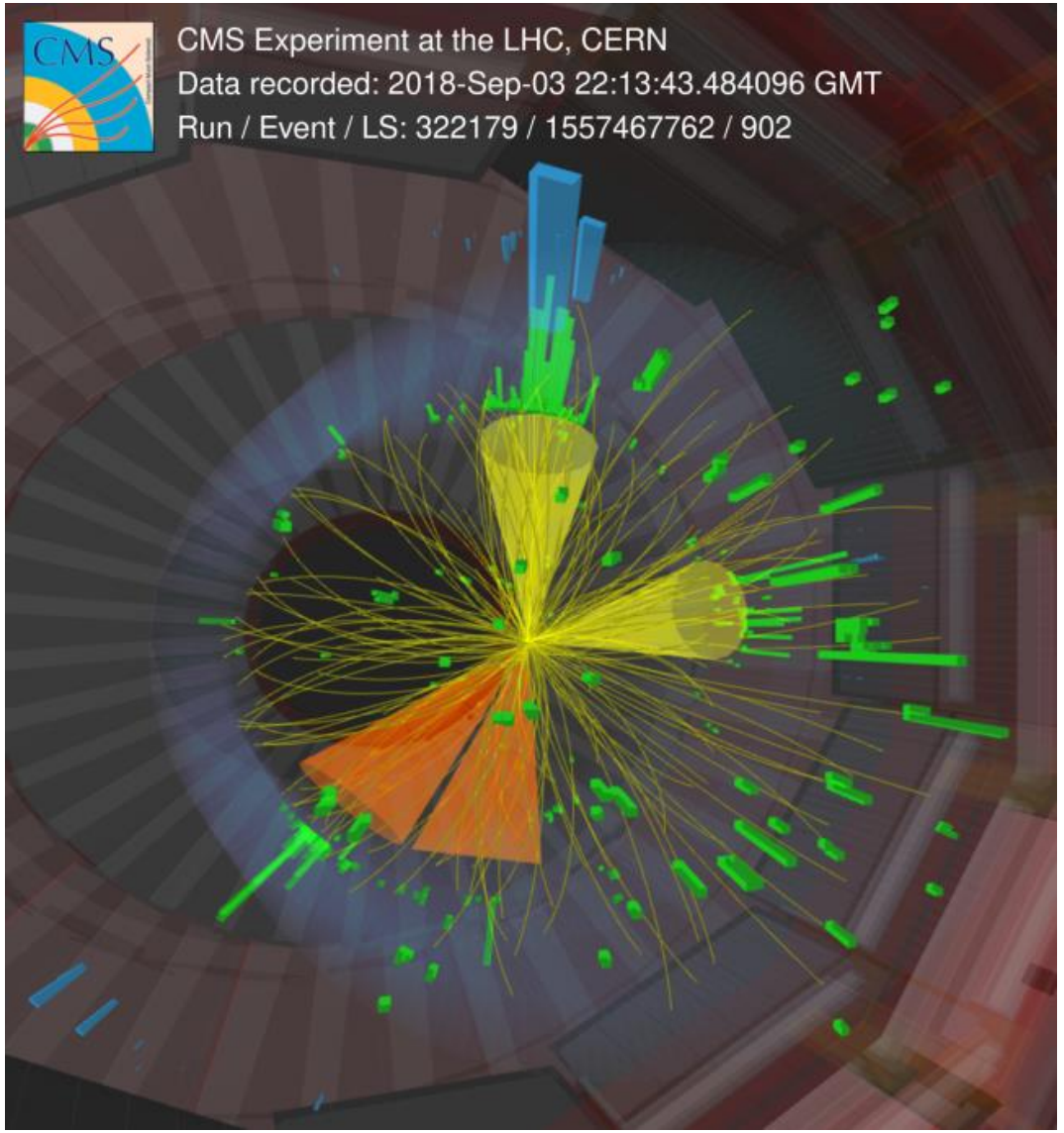
Jet classification—distinguishing top quark jets from QCD background—**is fundamentally a problem of statistical inference**, where each jet image represents a stochastic observation and the model estimates class probabilities via likelihood-based reasoning.



TOP QUARK

- The **top quark** is the heaviest known elementary particle in the Standard Model, with a mass of approximately **173 GeV/c2**.
- It carries an electric charge of +2/3e, has spin 1/2, and interacts via the **strong**, **electromagnetic**, and **weak** forces.
- It has an **extremely short lifetime** (~5×10^{-25} s)
- Main decay channel:
  t (top quark)→ **W boson + b-quark** → W boson decays into leptons or quarks → forms a jet.
- In high-energy collisions (e.g. HL-LHC), top quarks are often **highly boosted** meaning their decay products are **collimated** into a single jet.

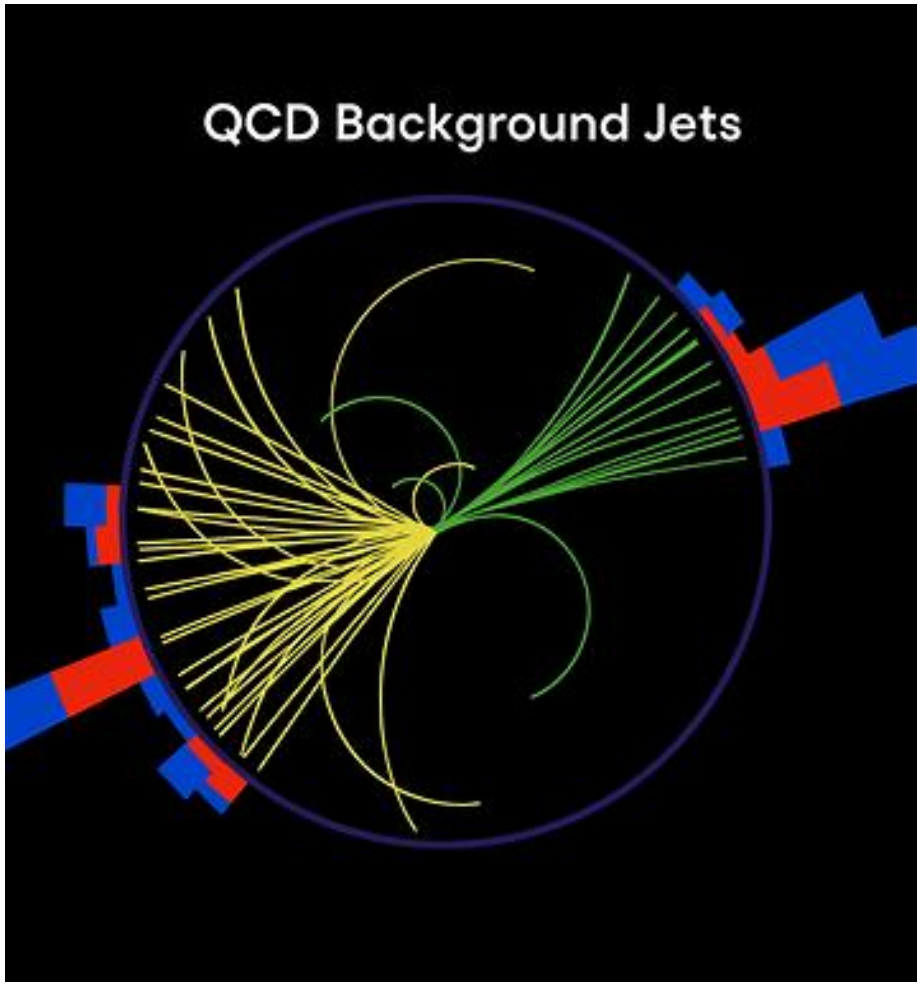
See Appendix - Boosted Top Quark



QCD BACKGROUND JETS

- **QCD** stands for **Quantum Chromodynamics**, the theory describing the strong force that binds quarks and gluons.
  See Appendix - Glucons
- When high-energy collide (e.g. at the LHC), their constituent quarks and gluons interact and fragment.
- These fragments **hadronize**—they form color-neutral particles (hadrons) that travel in roughly the same direction, creating a **jet**.
  See Appendix - Hadronize See Appendix - Color-neutral Particles
- Jets are **ubiquitous** in collider experiments and form a large part of the background noise when searching for more exotic signals (like top quark decays or new particles).
  Accurate top-tagging reduces false positives, improving the **purity and reliability** of experimental results.

- These jets can resemble QCD background jets → **challenging classification task**.
- This leads to **better statistical significance** in measurements and discoveries.



SCIENTIFIC MOTIVATION

In high-energy collisions (e.g., at the LHC), **jets** are produced as collimated sprays of particles originating from quarks and gluons.

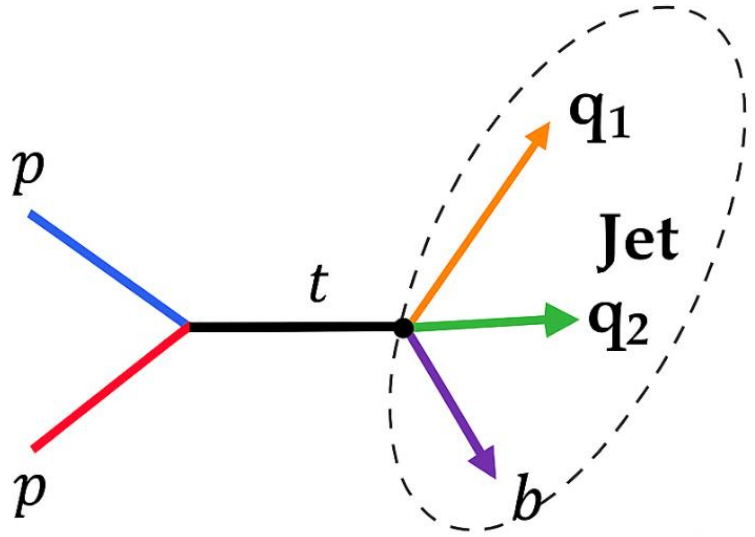
Identifying the **origin of a jet**—whether from a top quark decay or from QCD background—is crucial for:

- Studying top quark production
- Searching for **new physics** (e.g. heavy resonances decaying into top quarks)
- Improving **signal-to-background ratio** in collider analyses

TOP-TAGGING

Top-tagging is a technique used in particle physics to **identify jets originating from the decay of top quarks**, especially when the top quark is **highly boosted**—meaning it has high momentum and its decay products are tightly collimated into a single jet: jet classification is a **statistical inference problem**

- Top-tagging is a **benchmark task** for testing new .machine learning architectures (CNNs, QCNNs,GNNs) (*)
- It provides a real-world application of **statistical inference**, where models estimate using likelihood-based methods.



- A **proton-proton collision** at the LHC
- The production of a top quark
- Its decay into **three quarks** (b, q_1, q_2) forming a jet
- A contour highlighting the jet and suggesting the application of top-tagging techniques

(*) CNNs: **Convolutional Neural Networks**

QCNN**: Quantum: Convolutional Neural Networks**

GNNs**: Graph Neural Networks**

Neural Network

QNN vs CNN

Statistics

Inference

Fundamental ingredient - The model

Given some data, we need to

1. Identify all relevant observation x (data)
2. identify all relevant unknown parameters m
3. Construct a model for both

The model:

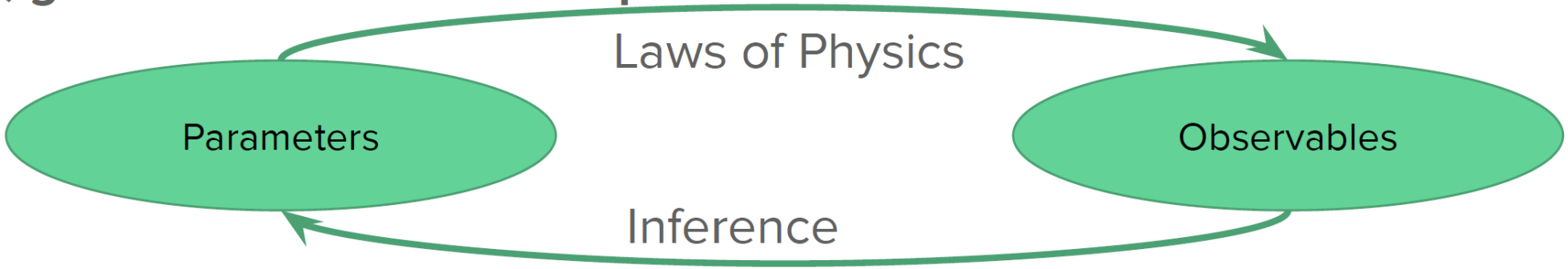
$$P(\text{data} \mid \text{theory}) = P(x ; \mu)$$

See Appendix -See Appendix - Model

Inference

In Physics there are quantities that can be measured (**observables x, y, etc**), and these observables can depend somehow by **parameters (μ, ν, etc).**

The laws of physics usually determine the values of the observables (and their evolution), given some values of the parameters.



The main goal for the inference is to get information on the parameters given some measured observables

From: Statistical Treatment and Analysis of the Data by Annovi

Likelihood function

Model evaluated at fixed data. Essential in most Bayesian and Frequentist inference

See Appendix - MLE vs Bayesian

- probability density function $p(x|m)$ of observing generic data x, given the unobservable value of the parameter m.
- Then take actual sample of observed data x_0 and evaluate $p(x_0|m)$
- The likelihood $L(m) = p(x_0|m)$ is a function of parameter m given your data

Connected to *probability for observing data x* for different choices of the value of the parameter m, **not** the probability that m has some value given the data.

Likelihood is a complete summary of the data information relevant to the estimate at hand. Ideally should be published as is.

See Appendix - Likelihood

From: Statistical Treatment and Analysis of the Data by Annovi

Wilks theorem

Asymptotically (large N), the distribution of the likelihood ratio

$$-2\ln \mathrm{LR}(m_0) = -2\ln \frac{p(x|m_0)}{p(x|\hat{m})}$$

approaches a χ^2 distribution with # of degrees of freedom equal to # of additional free parameters in the denominator wrt the numerator

Samuel S. Wilks (1906-1964)

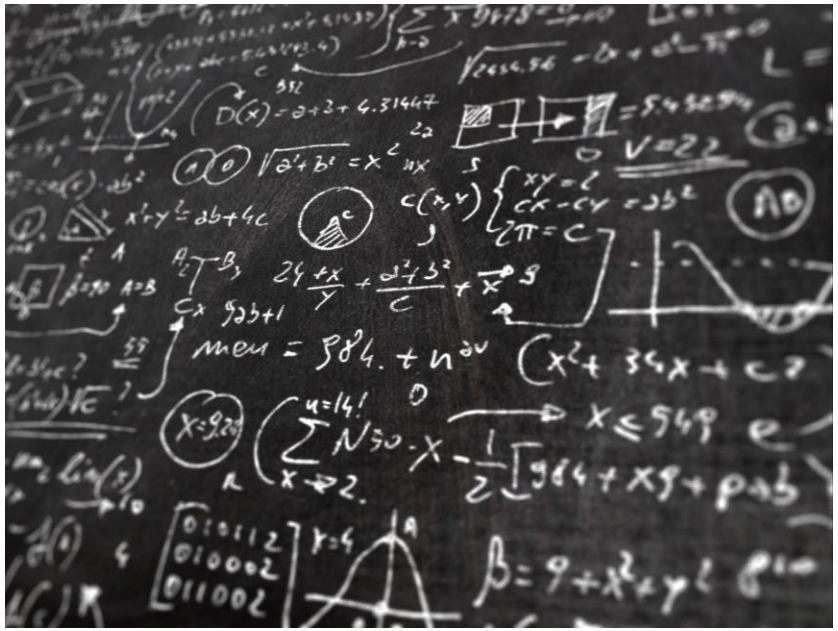
See Appendix - Wilks' theorem

This holds independently of the shape of p(x|m) and on the value of m.

Great helps in usage of likelihood- and profile-likelihood-ratio as ordering quantities in the construction of intervals. If the likelihood is regular enough to be in asymptotic regime, one can avoid massive production of simulated experiments.

From: Statistical Treatment and Analysis of the Data by Annovi

TOP-QUARK TAGGING: STATISTICS INFERENCE



In top-quark tagging, we don't observe the top quark directly, but only the **products of its decay** (b-quark, W boson → leptons, neutrinos, jets).

Therefore:

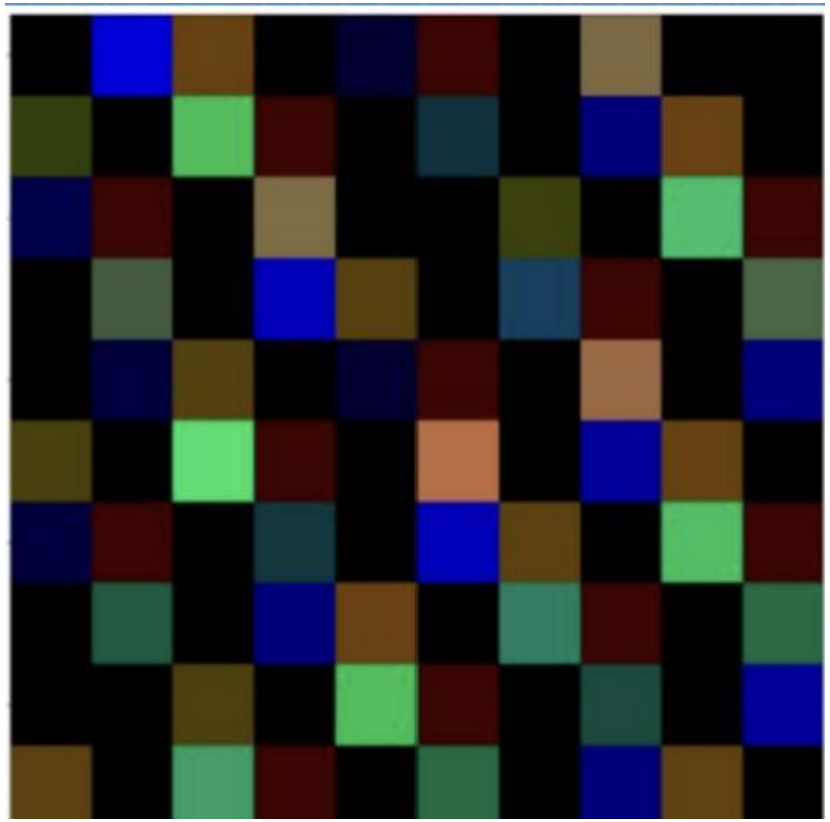
- The **observables** (energy, momentum, angular distributions) are **random variables:** jet features are not fixed—they vary across events:
  - Each collision is governed by quantum and detector-level uncertainties.
  - Jet formation involves stochastic processes: parton showering, hadronization, detector response.
  - Therefore, observables are **samples from a stochastic process**.

  See Appendix - Definition of Stochastic Process

- The jets we observe are **realizations of a stochastic process**.
- The dataset (e.g., JetNet) is a **sample** from a **theoretical population** of events. See Appendix - JetNet Datasets
- The model (CNN, QCNN, GNN) performs **Statistical Frequentist Inference**: estimating the probability that a given jet is of top-quark or QCD origin.

DEFINITION OF CNN

A **Convolutional Neural Network (CNN)** is a type of machine learning architecture widely used in image classification tasks. Its core mechanism involves applying filters—also known as **convolutional kernels**—that scan across input data (typically images) to detect patterns or features. These filters perform localized operations by multiplying pixel values with corresponding weights, enabling the network to identify relevant structures in the image.



MINIMAL CLASSICAL CNN ARCHITECTURE

- Develop a classifier for jet images (top vs QCD) using a classical convolutional neural network (CNN).
- This serves as a baseline (classical benchmark) to
- Evaluate whether QCNNs offer advantages in terms of accuracy, robustness, and parameter efficiency.
- This pipeline is intentionally kept simple to match the QCNN pipile
- parameter count of the QCNN setups, ensuring a fair comparison.

See Appendix - Classic CNN - Purpose in the Study

Layer	Function
Conv2D	Extracts spatial features from jet images.
MaxPooling	Reduces dimensionality while preserving key features.
Flatten	Converts 2D feature maps into a 1D vector.
Dense ×2	Performs classification into top-quark vs QCD jet categories.
Activation Functions	ReLU, Sigmoid, or Tanh depending on the loss function used.

QCNN PIPELINE

See Appendix - Quantum Pipeline details

Stage	Description
Jet Image	Classical input image representing energy distribution of jet constituents.
Encoding Layer	Converts classical image pixels into quantum states.
Convolution Layer	Applies quantum gates to pairs of qubits to extract spatial features. Tested circuits: SO(4) and SU(4).
Pooling Layer	Reduces the number of qubits by half using CNOT and rotation gates.
Measurement Layer	Measures the final qubit using Pauli-Z to produce a prediction.
Output	Prediction: top-quark jet or QCD jet.

STATISTICAL VALIDATION OF QCNN PERFORMANCE

- **The QCNN** with **SO(4)** and **HEE1** encoding achieved: See Appendix - QCNN vs CNN - See Appendix - SO(4) & HEE1

  Accuracy=99.22%±0.11%

  The classical CNN, with matched parameter count, achieved:

  Accuracy=94.76%±2.17%

  Evaluate whether **QCNNs offer advantages in terms of accuracy**, robustness, and **parameter efficiency**

These values represent the **empirical frequency of correct classifications** → exactly what your binomial model describes - **Accuracy can be treated as MLE** (p^) in a binomial model - See Appendix - Accuracy

if you treat the QCNN as a classifier and the test set as a sample, then:

- Each classification is a trial.
- The number of correct predictions follows a **binomial distribution**.
- The confidence interval computed via Wilks provides a **rigorous measure of uncertainty** on the model's efficiency.

DEFINITIONS IN STATISTICAL INFERENCE

Observables

- The number of trials N=1000 and the number of observed successes K=960 are the **observables**. They represent the empirical evidence used to infer the underlying probability p.

Parameter

- The parameter is the **success probability** $p \in (0,1)$.
- It governs the binomial distribution and is the quantity we aim to estimate using the data.

STATISTICAL INFERENCE PROBLEM: PIPELINE

Estimate the binomial success probability p using MLE, analyze the likelihood function, and construct a confidence interval using **Wilks' theorem**. **This study adopts a frequentist approach.**

See Appendix - Wilks'theorem

Step	Purpose
1	Binomial Model: Define the parameters
2	Compute the Maximum Likelihood Estimate (MLE)
3	Likelihood: A Function of the Parameter L(p)
4	Likelihood Curve and MLE Visualization
5	Compute and visualize the LLR (log-likelihood ratio)
6	Solve for uncertainty intervals bounds
7	Physical Interpretation

BINOMIAL MODEL: DEFINE THE PARAMETERS

$$P(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}$$

In your case:

- $n = 1000$
- $k = 960$
- The model is $P(960; 1000, p)$

We assume each trial has a **probability p of success**. The binomial model gives the probability of observing K successes out of N trials:

These are the inputs to your binomial model. You observed 960 successes out of 1000 trials.

N = 1000 # total number of trials

K = 960 # number of observed successes

COMPUTE THE MAXIMUM LIKELIHOOD ESTIMATE (MLE)

- The **maximum likelihood estimate (MLE)** for the **binomial success probability** is simply the observed proportion of successes **(Analytical Calculation)**:

$$\hat{p} = 960/1000 = 0.96$$

Define the range of p values to explore:

- `0.9`: lower bound of the interval
- `1.0`: upper bound
- `500`: number of points in the interval. This creates a fine grid of 500 values between 0.9 and 1.0 to evaluate the likelihood function.

See Appendix - Analytical Calculation
See Appendix - Numerical Calculation
See Appendix - Code Compute (MLE)

LIKELIHOOD: A FUNCTION OF THE PARAMETER L(P)

The **likelihood** is the binomial formula viewed as a function of p, with N**=1000 and K=960 fixed:**

$$L(p) = \binom{1000}{960} p^{960}(1-p)^{40}$$

This function L(p) **tells us how plausible different values of p are, given the observed data.** For each value of p, compute the probability of observing exactly K=960 successes out of N=1000 trials. This is the likelihood function L(p).

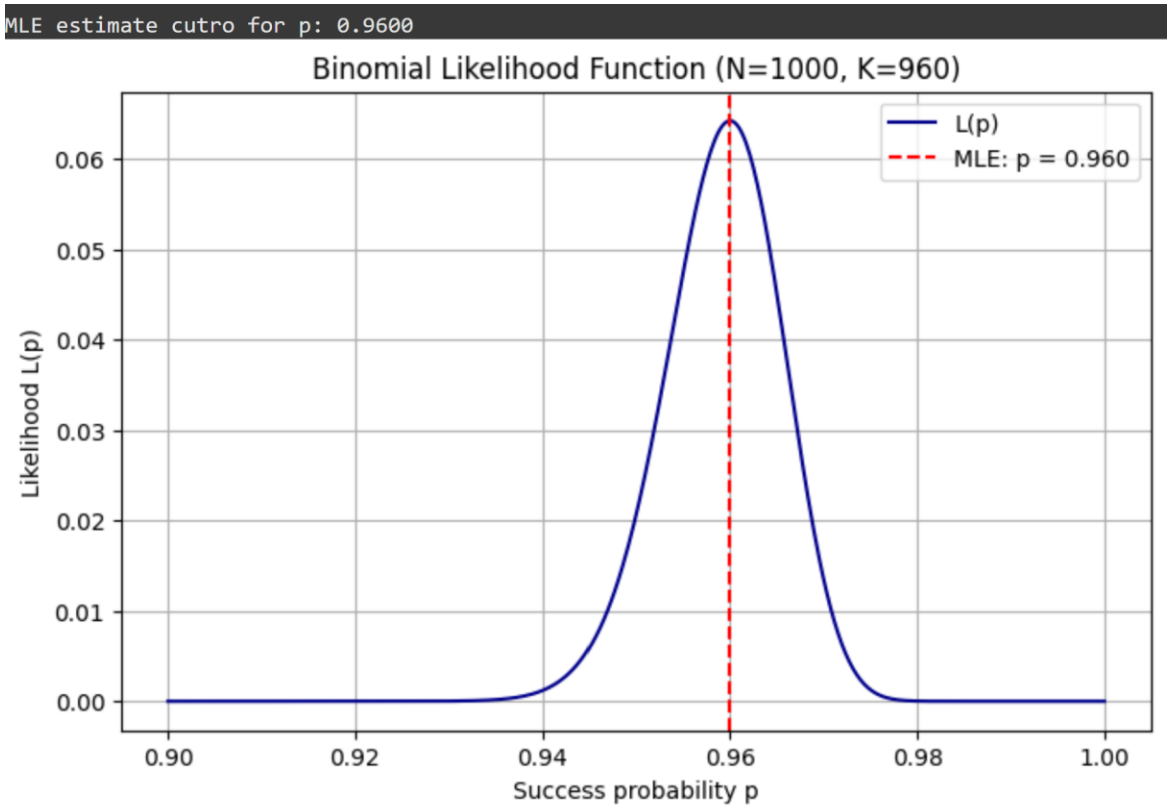
The **maximum likelihood estimate (MLE)** is the value of p that maximizes this function:

p^=960/1000=0.96

So:

- **Likelihood** is a mathematical tool.
- It's a function of the parameter p.
- It tells you which values of p best explain the observed data. See Appendix - Code Likelihood

LIKELIHOOD CURVE AND MLE VISUALIZATION



Plot the likelihood curve and highlight the **MLE** Mark the MLE p^=0.96 with a vertical red dashed line.

- The peak of the curve corresponds to **p^**
- Visual intuition: **values of p near the peak** are more compatible with the data.

See Appendix - Code - Plot the likelihood function

COMPUTE THE LOG-LIKELIHOOD RATIO (I)

In likelihood-based inference, we often want to compare how well different values of a parameter explain the observed data. The **Wilks statistic** provides a way to quantify this comparison using the **log-likelihood ratio**.

Let's say:

- L(p) is the likelihood of a parameter value p
- L(p^) is the maximum likelihood, i.e. the likelihood at the best-fit value p^

The **Wilks statistic** is defined as **LLR(p)**:

$$\mathrm{LLR}(p) = -2\log\left(\frac{L(p)}{L(\hat{p})}\right)$$

- **Likelihood comparison**: It compares the likelihood of any value p to the maximum likelihood at p^.
- **If p=p^, then L(p)=L(p^) and LLR = 0 → perfect fit.**
- **Penalty for deviation**: As p moves away from p^, the **likelihood decreases, and the LLR increases**. *This reflects how incompatible that value of p is with the observed data*.

See Appendix - Definition LLR(p)

COMPUTE THE LOG-LIKELIHOOD RATIO (II)

Under regular conditions and large sample sizes, Wilks' theorem tells us that this statistic follows a **chi-square distribution**.

- This allows us to define confidence intervals without needing to simulate pseudo-experiments.
- For one parameter:
- A value of **1** corresponds to a 1σ
- A value of **4** corresponds to a 2σ

See Appendix - Code - Compute the log-likelihood ratio

VISUALIZE LOG-LIKELIHOOD RATIO

- It quantifies the **uncertainty** around the estimate via Wilks' Theorem
- According to **Wilks' theorem**, the LLR statistic follows a chi-square distribution with 1 degree of freedom (since p is a single parameter).
- This allows us to define confidence intervals without simulations:

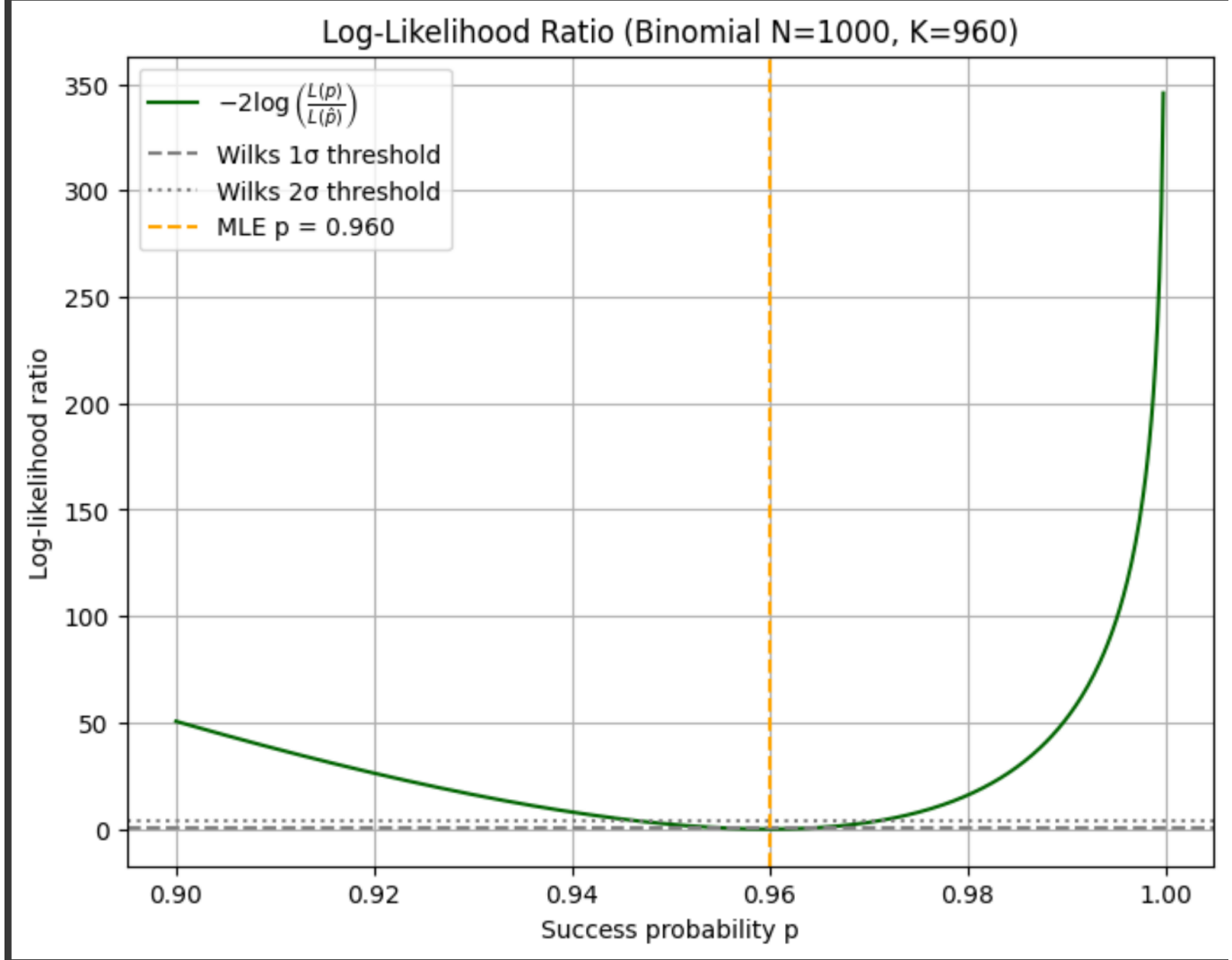
  LLR = 1 → (1σ)

  LLR = 4 → (2σ)

- The **uncertainty** includes all values of p for which the LLR is below the threshold.

- See Appendix - Code Log-likelihood ratio
  See Appendix - Define the log-likelihood ratio function

```
log_likelihood_ratio = -2 * np.log(likelihood / L_max)
```




```
Intervallo di confidenza 1σ: [0.95349, 0.96589]
```

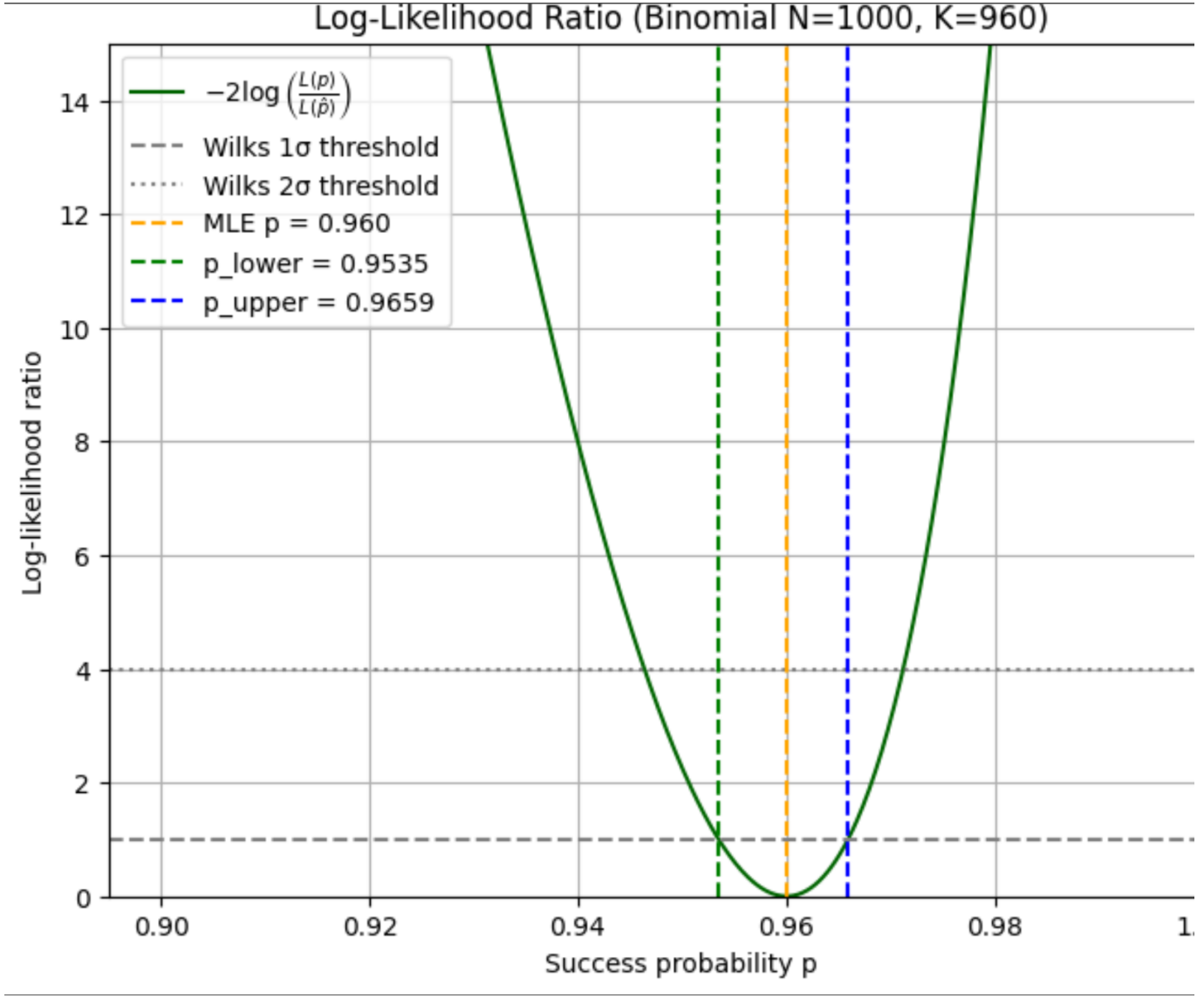
VISUALIZE THE UNCERTAINTY

- Display the **Uncertainty interval** around the estimate p^=0.96:

[0.9535,0.9659]

- **Visual meaning**: *The* Uncertainty *interval includes all values of p for which the LLR is below the threshold. It's the region where the likelihood is "not significantly worse" than the maximum.*
- **All values of p for which the LLR is below 1 are considered statistically compatible with the data at the 1σ level**

- See Appendix - Code - Find the 1σ confidence interval



CONCLUSIONS & PHYSICAL IMPACT

The **log-likelihood ratio (LLR)** compares how well different values of the parameter p explain the observed data.

- It is **zero** at the maximum likelihood estimate $\hat{p}=0.96$, meaning that this value fits the data best.
- As p moves away from $\hat{p}$, the likelihood decreases, and the LLR increases — this reflects **decreasing compatibility** with the data.

Confidence Interval via Wilks' Theorem

- According to **Wilks' theorem**, the LLR statistic follows a chi-square distribution with 1 degree of freedom (since p is a single parameter).
- chi-square distribution is valid **asymptotically**—that is, under conditions of large sample size and regularity
- This allows us to define confidence intervals without simulations:
- LLR = 1 → (1σ) See Appendixc 1σ interval – Statistical Meaning
- LLR = 4 → (2σ)

PHYSICAL IMPACT

What it means physically

- *You are not claiming that "there is a about 68% probability that p lies in this interval." That would be a Bayesian interpretation.*

Instead, you're saying:

- "If we repeated this experiment many times, and each time computed a 1σ confidence interval using the same method, about **68% of those intervals would contain the true value of** p."
- The **true efficiency** of your QCNN classifier (its ability to correctly tag jets) is **likely between 95.35% and 96.59%**, based on the observed data.
- This quantifies the **uncertainty** in your estimate due to statistical fluctuations in the test set.

MEAN AND VARIANCE (N=1000; K=960)

We can calculate the sample mean and variance

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad s^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \bar{x})^2$$

```
mean = df.mean()[0]
print(f'The sample mean value is {mean}')
```

```
The sample mean value is 0.95
/tmp/ipython-input-2548494797.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be
  mean = df.mean()[0]
```

```
var = df.var()[0]
print(f'The sample variance value is {var}')
```

```
The sample variance value is 0.0008383500467869598
/tmp/ipython-input-1193463208.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be
  var = df.var()[0]
```

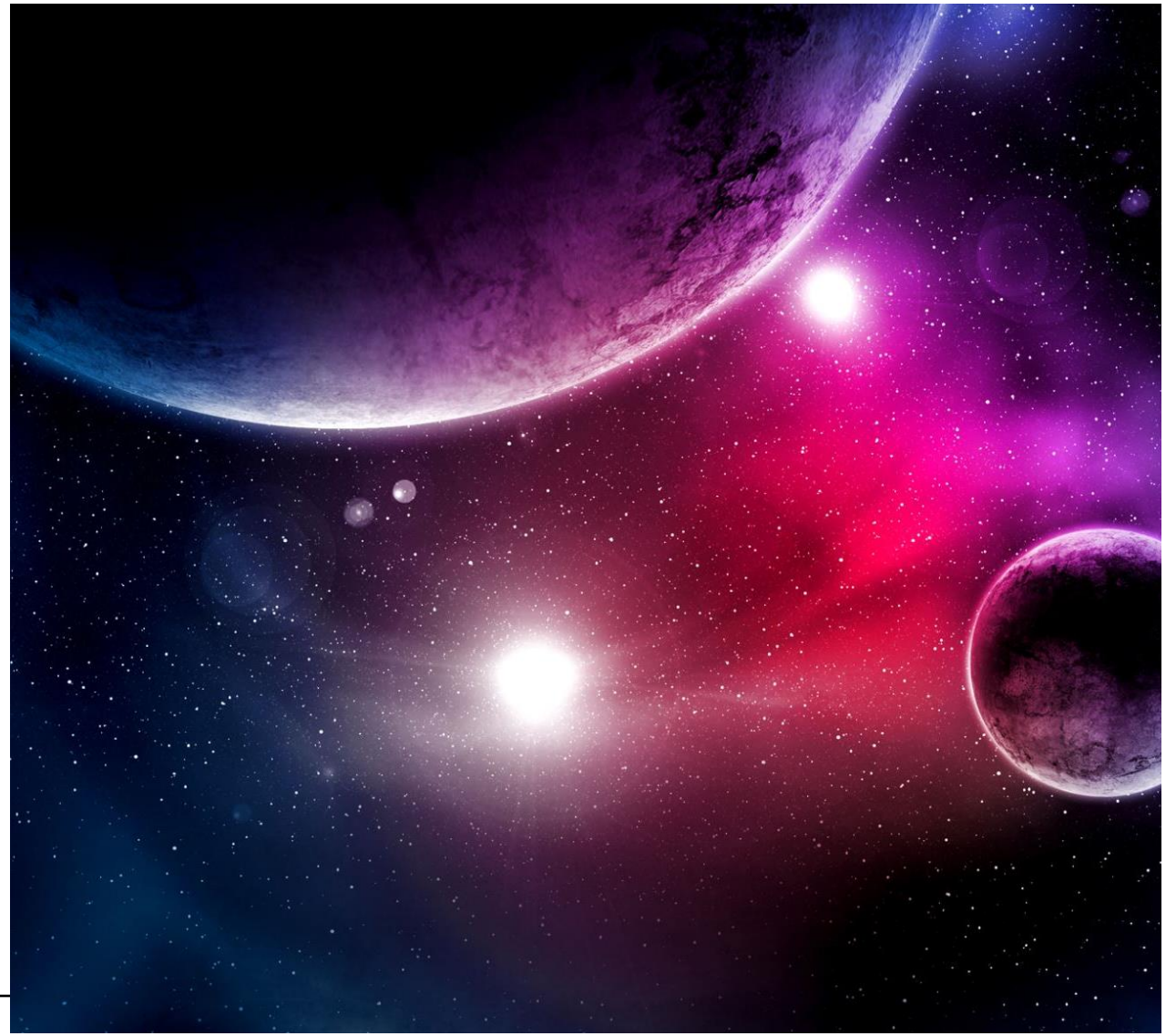
IT-Sphere

Cross-Domain Analogy

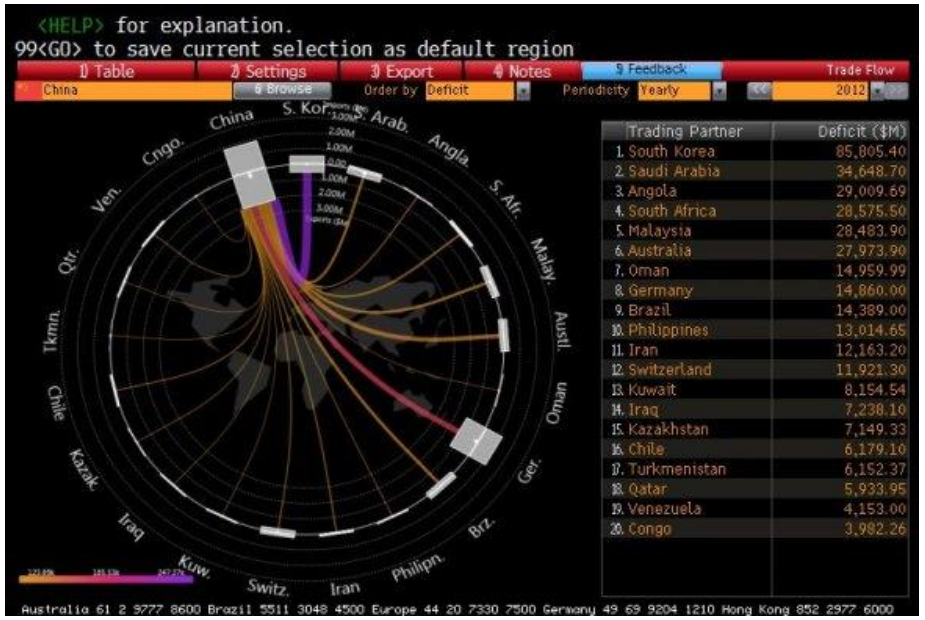
IT-Sphere Events

Accuracy &IT

FROM THE UNIVERSE TO IT SYSTEM: WHY CORRELATING EVENTS IS A SHARED CHALLENGE



COMPLEX EVENTS, INTELLIGENT CORRELATIONS



Top Quark vs QCD :

- Jet classification—distinguishing Top Quark jets from QCD background **using CNN/QNN through statistical inference**.

IT-Sphere (e.g., banking sector):

- Detects sparse anomalies (logs, alerts, transactions) in distributed systems.
- Reconstructs “IT events” (incidents, fraud, cyberattacks) from fragmented signals.
- Uses AI to correlate logs and alerts and anticipate systemic impact.

Why this analogy is useful:

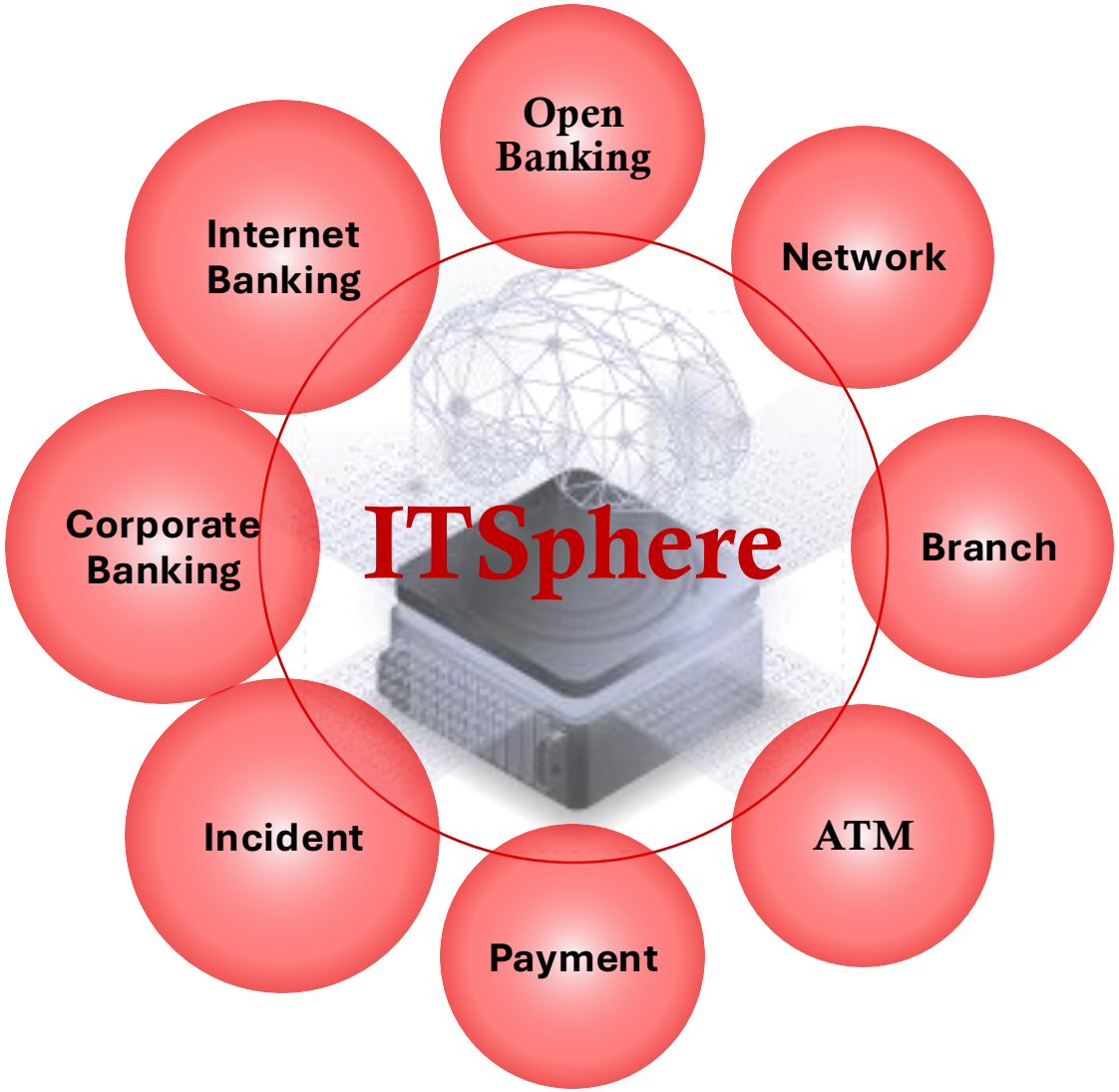
- Both domains deal with **rare, complex, and distributed events**.
- Require **intelligent models** to correlate signals and reconstruct reality.
- **Statistical Inference** techniques developed in physics can **inspire advanced IT solutions**.

INTELLIGENT EVENT CORRELATION: TOP QUARK AND IT BANK EVENTS VIA STATISTICAL INFERENCE

- Jet classification—distinguishing Top Quark jets from QCD background **through Statistical Inference**.
- IT Bank Events (**IT-Sphere**) detectes IT events across distributed banking systems.

IT-Sphere Overview

IT-Sphere ecosystem: centralized architecture for managing diverse components of a banking IT infrastructure.



DEEPCORE (IT-SPHERE) OVERVIEW

Datacenter within the IT-Sphere framework is a highly integrated physical and virtual infrastructure that hosts thousands of interconnected components—including **servers**, **databases**, **network devices**, **security systems**, and **enterprise applications**.

IT-Sphere ecosystem Datacenter

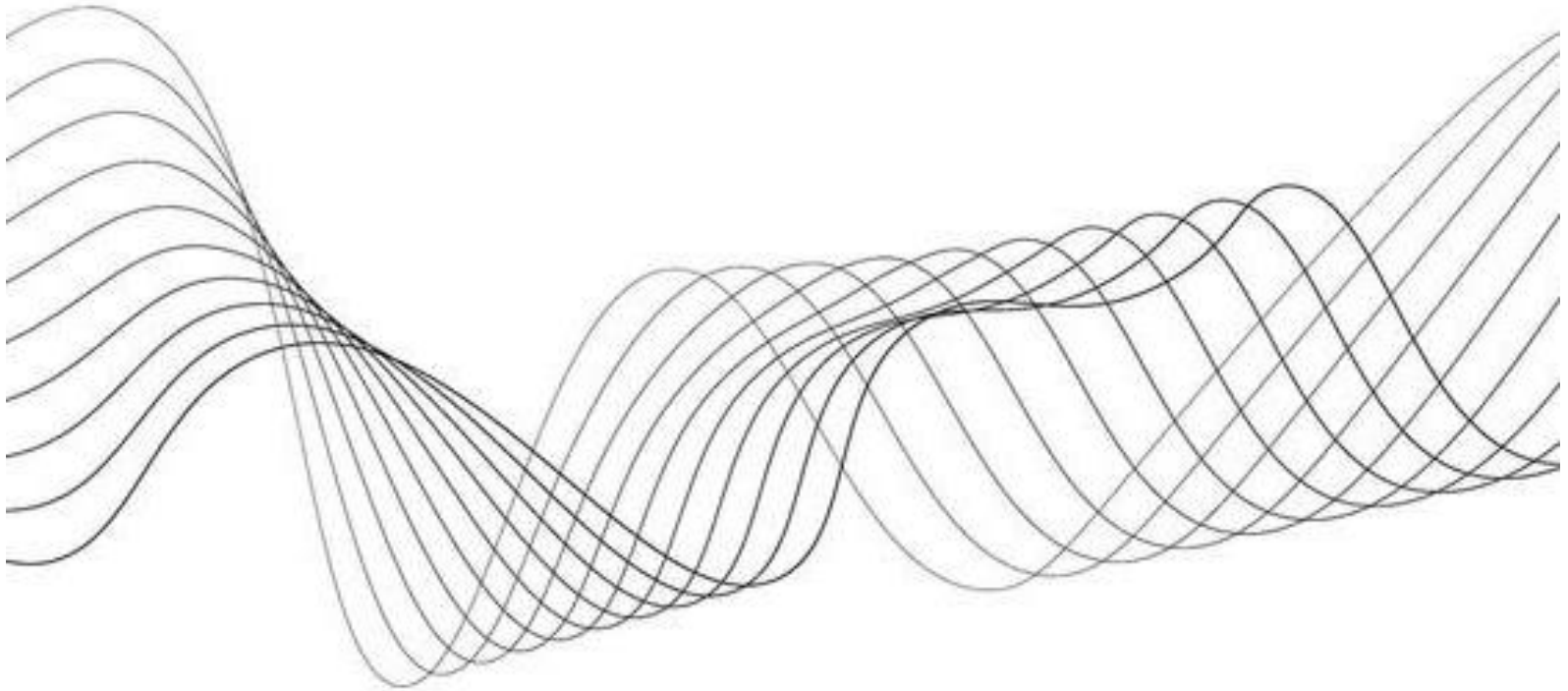
DETECTION MECHANISM (IT-SPHERE)

- A **datacenter monitoring system** within the IT-Sphere ecosystem is a complex ensemble of infrastructures and technologies designed to continuously observe, analyze, and interpret the operational state of thousands of interconnected components—including servers, databases, network devices, security modules, and banking applications.
- It enables **proactive incident response**, ensures service continuity, real-time analytics and event correlation engines.

NOISE & TRIGGERING (IT-SPHERE)

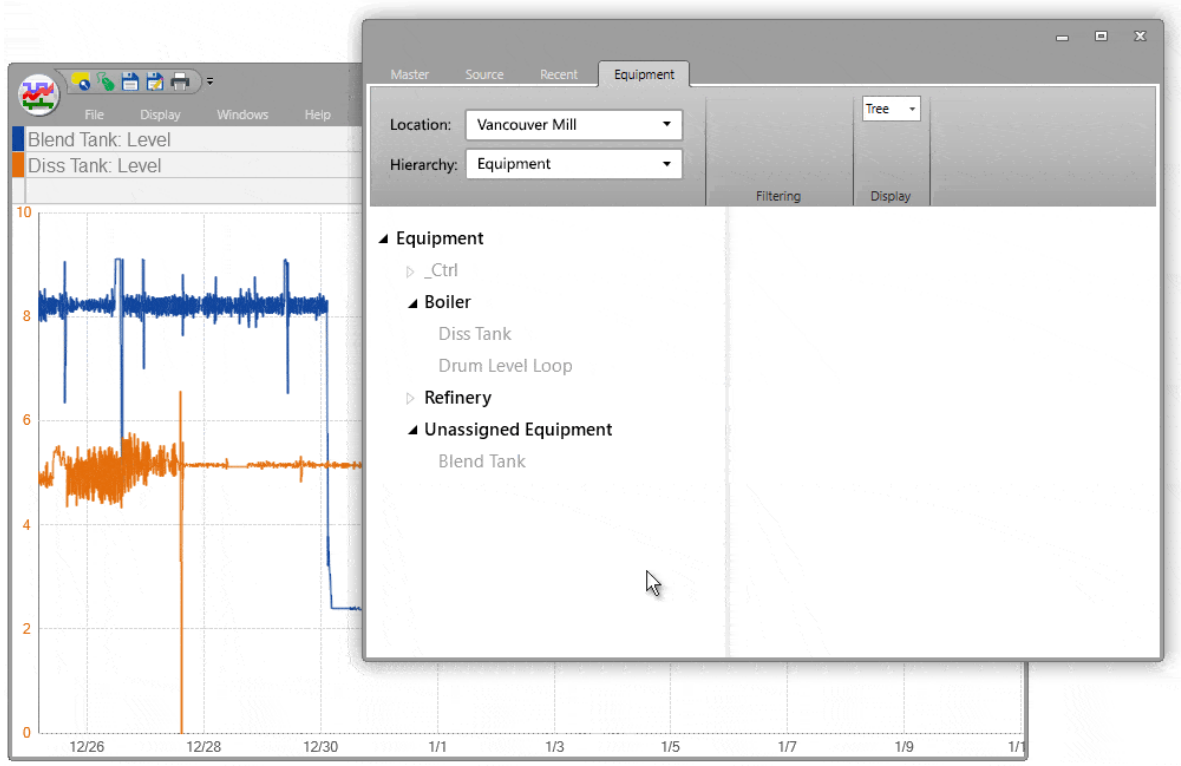
Monitoring system detects **rare and critical events**—such as system failures, security breaches, or transactional anomalies—hidden within the vast stream of **transactional data**, often referred to as **Log Waves**.

Log waves represent the background noise of routine operations, making the identification of meaningful alerts a non-trivial challenge.



EVENTS AND IT-SPHERE

- An **IT-Sphere event** is a discrete occurrence within the IT infrastructure that signifies a deviation from expected operational behavior—typically associated with a system anomaly, performance degradation, or potential security threat
- Tn the context of datacenter monitoring, such an event is synthetically referred to as an **alarm**, representing a rare and significant signal emerging from the continuous flow of transactional data, often described as *log waves.*



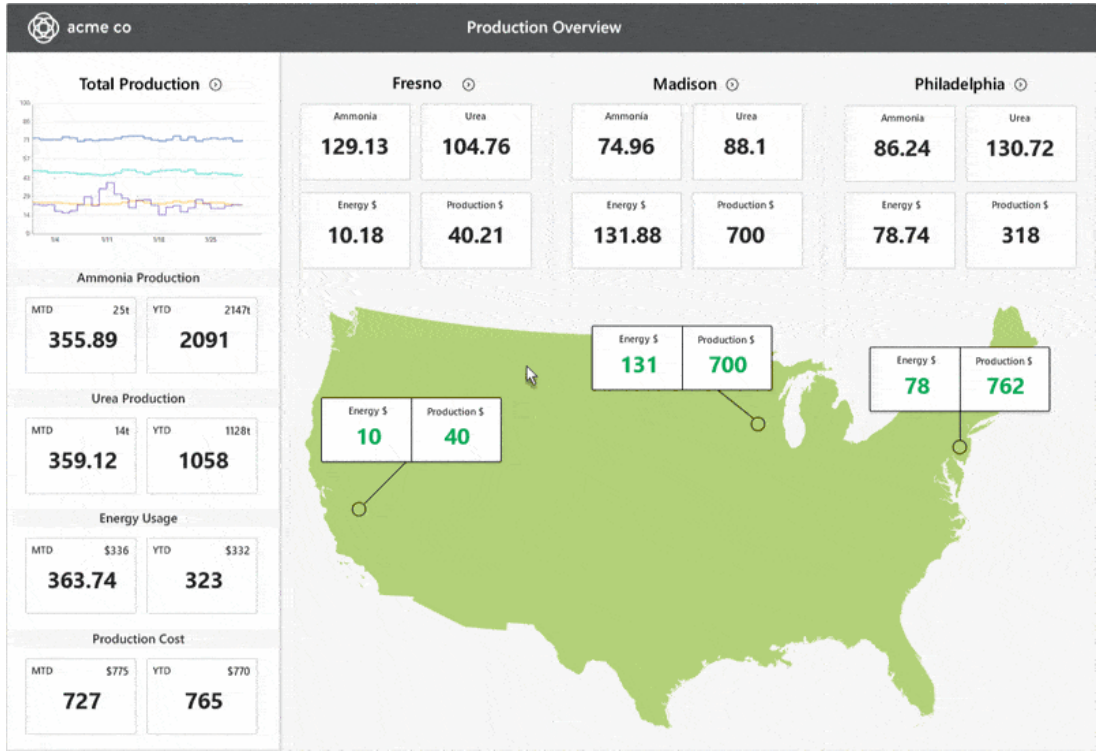
Real-Time Manufacturing Analytics Software | PARCview

ALARM-LIKE TOP QUARK JET CLASSIFICATION

Alarms like *Top Quark Jet* - Rare and significant

- Triggered by anomalous or critical system behavior
- Require precise correlation and contextual interpretation
- Often associated with service impact, security breach, or infrastructure failure
- High informational value per event
- May persist over time (active/cleared states)

See Appendix - Alarms features
See Appendix -Two Levels of Parameters in IT-Sphere

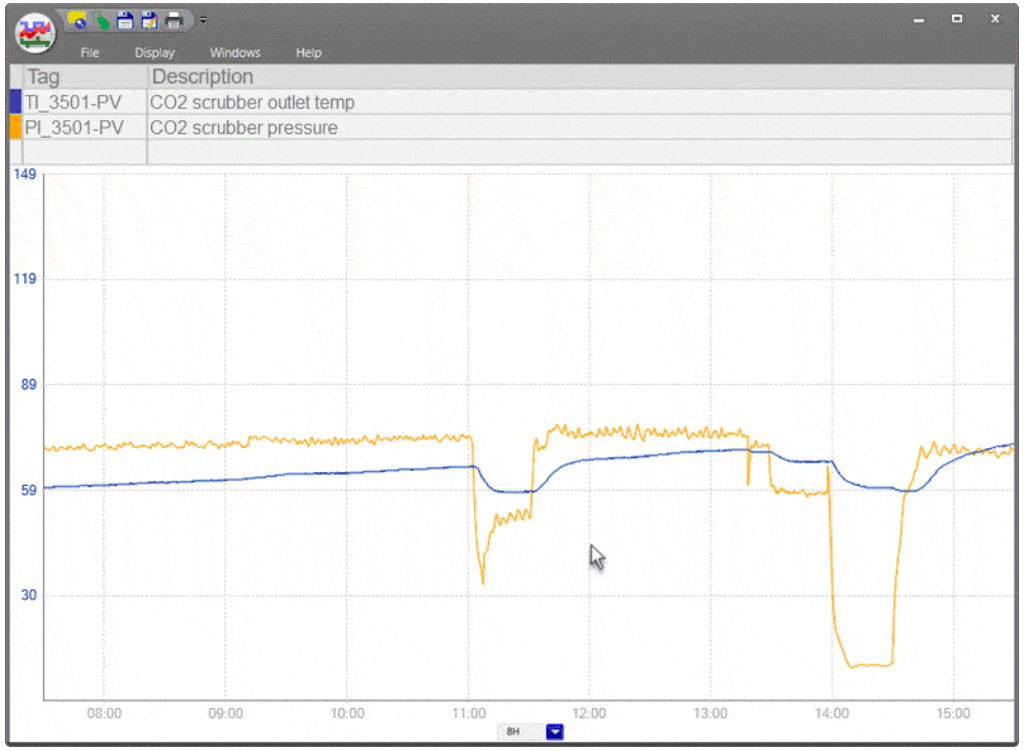


Real-Time Manufacturing Analytics Software | PARCview

LOG WAVES-LIKE *QCD NOISE:* CLASSIFICATION

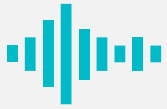
Log Waves like *QCD Background noise*

- Frequent, low-significance system logs
- Represent routine operations (e.g., access logs, heartbeat signals, status updates)
- Typically uncorrelated and non-critical
- Can obscure meaningful alarms if not filtered
- Low informational value per event
- Often transient and stateless



Real-Time Manufacturing Analytics Software | PARCview

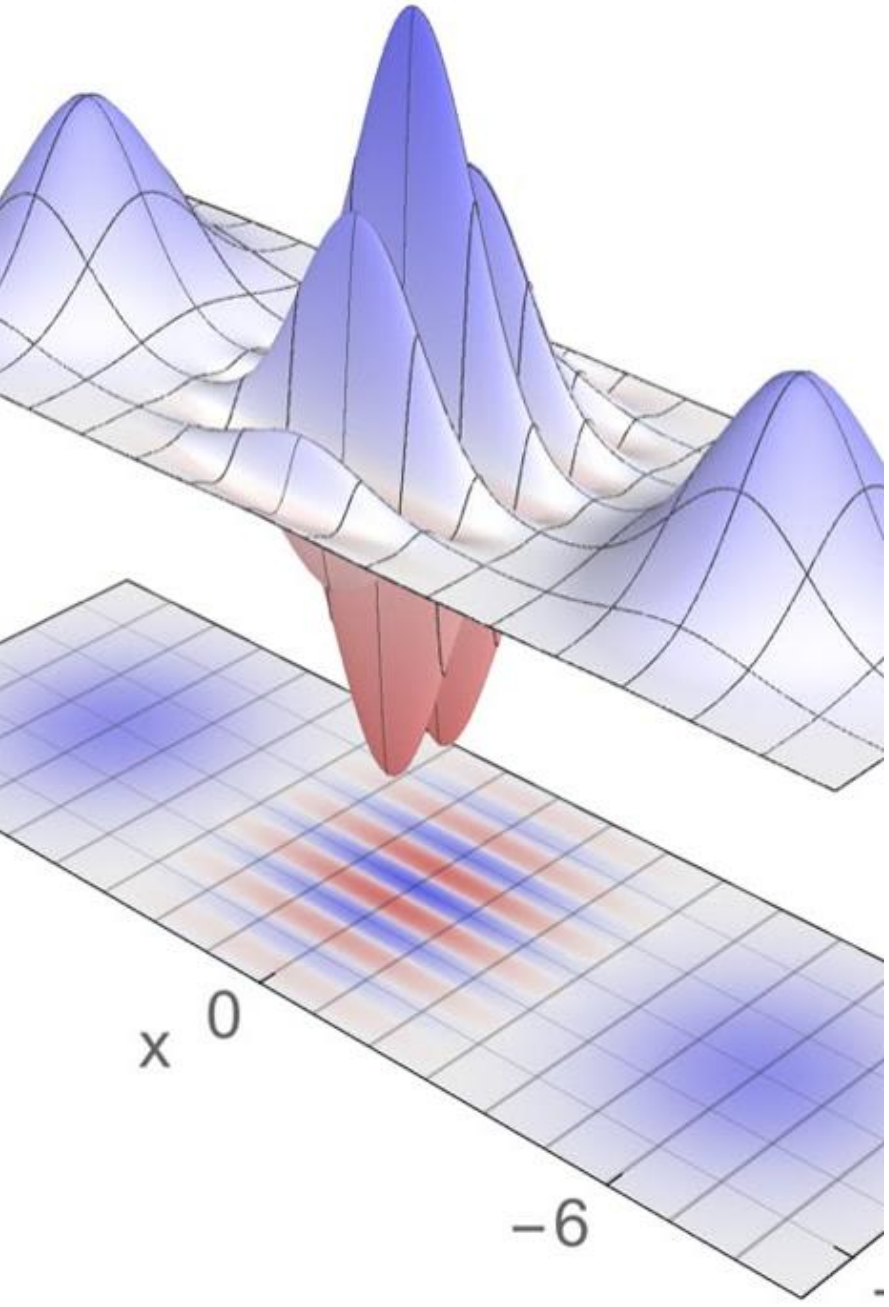
CORRELATION TOP QUARK JET AND IT-SPHERE CLASSIFICATION



Suppress log wave noise - analogous to QCD Background rejection in (e.g. at the LHC)

Enhance signal extraction for alarms (Top Quark Jet events)

Model temporal and causal relationships to distinguish meaningful patterns



JET CLASSIFICATION IN PARTICLE PHYSICS AND EVENT DETECTION IN IT SYSTEMS (I)

Stochastic Observations

- In physics: each jet image is a realization of a **stochastic process** governed by quantum and detector-level uncertainties.
- In IT: each system event (e.g., API call, transaction, log entry) is a **random outcome** influenced by network, software, and user behavior. Each minute (or second) is a sampling window.

Binary Classification Tasks

- In jet tagging: classify each jet as either **top-quark** or **QCD**.
- In IT monitoring: classify each event as either **normal** or **anomalous** (e.g., failure, breach).

Both can be modeled as **Bernoulli trials**, and accuracy is computed as:

Accuracy=*Number of successes / Total number of trials (Volume, Performance, Error)*

JET CLASSIFICATION IN PARTICLE PHYSICS AND EVENT DETECTION IN IT SYSTEMS (II)

Frequentist Inference

- In both cases, the success rate p is unknown and estimated via **Maximum Likelihood Estimation (MLE)**.
- Confidence intervals around p^ are constructed using **Wilks' theorem**, treating each classification or system event as a trial.

Signal vs Background

- In physics: distinguish rare top-quark jets from abundant QCD background.
- In IT: detect rare alarms within noisy log waves.

  This leads to similar statistical challenges: **low signal-to-noise ratio**, **false positives**, and the need for **robust inference**.

APPENDIX

QCNN VS CNN

Accuracy: better classification with fewer errors

- The **QCNN with SO(4)** and HEE1 encoding achieved **99.84% accuracy** using MSE, outperforming the equivalent CNN (90.81%).
- This is especially evident in the low-parameter regime, where CNNs tend to suffer from higher variance and poorer generalization.
- **Why?** The quantum structure can capture non-linear and global correlations between pixels (qubits) that classical filters struggle to model.

Robustness: resistance to barren plateaus and overfitting

- QCNNs use **shallow-depth circuits**, which are less prone to barren plateaus—regions in the optimization landscape where gradients vanish.
- Encodings like HEE1 and TPE show **better trainability** compared to HEE2 and CHE, which induce plateaus and hinder convergence.
- Moreover, the **classical simulability** of QCNNs makes them robust even on noisy intermediate-scale quantum (NISQ) devices

UNCERTAINTY INTERVAL IN IT MONITORING

In statistical inference, we can construct a **uncertainty interval** around the estimate p^ via Wilks' Theorem.

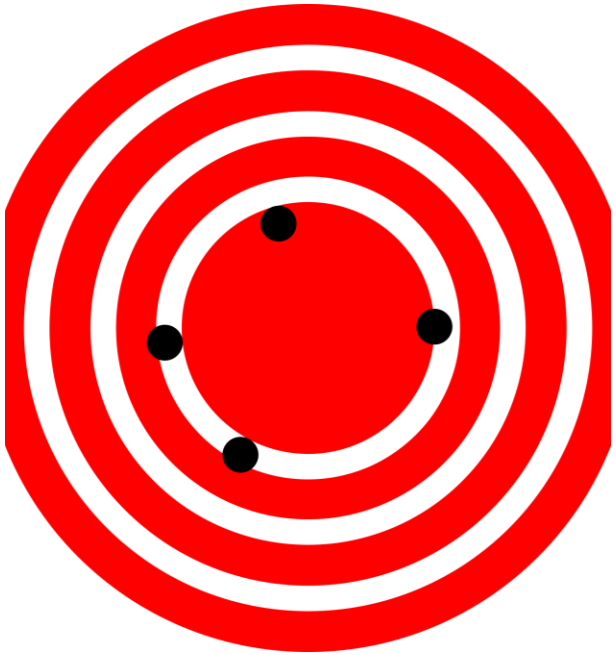
You can apply the same logic to IT systems:

- **Instead of assuming the observed success rate is exact**, you acknowledge **measurement uncertainty: i**nterval for **system reliability**, based on observed uptime or error rates.

This allows IT teams to say:

"Given our data, the true success rate of the system is likely between 94.2% and 97.4%."

Which is statistically rigorous and mirrors your physics-based inference.



RESOLUTION VS ACCURACY: KEY CONCEPTS

Accuracy

- **What it is:** Measures how close the average estimate is to the true value.
- **Underlying question:** "How correct is my estimate?"
- **Example:** If the true energy is 50 GeV and the average estimate is 49 GeV, accuracy is high (low bias).
- **Typical metrics:** Mean Absolute Error (MAE), bias, Root Mean Square Error (RMSE).

Resolution

- **What it is:** Measures how tightly clustered the estimates are around the average—i.e., the **spread** of errors.
- **Underlying question:** "How consistent are my estimates?"
- **Example:** If estimates vary between 48 and 50 GeV, resolution is high; if they vary between 40 and 60 GeV, resolution is low.
- **Typical metrics:** Inter-percentile spread (e.g., $(P_{84}-P_{16})/2$, standard deviation.

MLE VS BAYESIAN: KEY DISTINCTION

- **What does it do?** It finds the parameter value that **maximizes the likelihood**—the value that makes the observed data most probable.
- **It does not assume a prior distribution** over the parameter.
- **The parameter is fixed but unknown**, and the data are considered random.
- **Example in your case**: You observe 960 successes out of 1000 → the MLE of p is $\hat{p}=0.96$.

Bayesian – **Probabilistic Approach to the Parameter**

- **What does it do?** It assumes the parameter is a **random variable** with a prior distribution $\pi(p)$.
- It uses **Bayes' theorem** to update beliefs about the parameter after observing data:

  $\text{Posterior}(p) \propto L(p) \cdot \pi(p)$

  The parameter is random, and the data are fixed.
- **Example**: If you assume $\pi(p)=\text{Uniform}(0,1)$, the posterior distribution is proportional to the likelihood—similar numerically to the frequentist result, but with a different interpretation.

1Σ INTERVAL – STATISTICAL MEANING

What does "The 1σ interval might be [0.93, 0.99]" mean?

This sentence indicates that, based on the observed data and the likelihood function, the estimated value of the parameter (e.g., p) is **most compatible with the data** if it lies **within the interval [0.93, 0.99]**.

Why is it called a "1σ interval"?

- "1σ" refers to a **68% confidence level**, derived from the **normal (Gaussian) distribution**.
- It means that if we repeated the experiment many times, **about 68% of the time the true value of** p would fall within that interval.
- **In practice, it's a way of saying: "We are reasonably confident that the true value of p lies between 0.93 and 0.99.**

1Σ INTERVAL – PHYSICAL MEANING

in the context of particle physics (e.g., top-quark jet tagging):

- If the parameter p represents the probability that a jet is a top quark, then saying $p \in [0.93, 0.99]$ at the 1σ level means:

  "With 68% confidence, the data are compatible with a value of p in this interval."

- If a hypothesis proposes $p=0.85$, but the LLR compared to $\hat{p}=0.96$ is > 1, then **that hypothesis is statistically less compatible** with the data.

SO(4) & HEE1 = HARDWARE-EFFICIENT ENCODING (1 LAYER)

Parameter Efficiency: fewer parameters, same expressivity

- The QCNN with SO(4) uses **30 parameters**, compared to **33 in the CNN**, yet achieves **higher accuracy**.
- Dimensional Expressivity Analysis (DEA) allows the quantum circuit to **eliminate redundant parameters** while preserving its discriminative power.
- This is crucial in high-energy physics, where the number of features (qubits/pixels) is limited and each additional parameter increases the risk of overfitting.

SO(4) = Tipo di circuito di convoluzione

- **SO(4)** è un'unità quantistica a due qubit che realizza **trasformazioni reali ortogonali**.
- Ha **meno parametri** rispetto a SU(4), quindi è **più efficiente** e meno soggetta a overfitting.
- In questo contesto, SO(4) è usato per costruire i blocchi di convoluzione del QCNN.

HEE1 = Hardware-Efficient Encoding (1 layer)

- È uno dei metodi per **codificare dati classici (pixel)** in stati quantistici.
- HEE1 usa **una sola layer** di porte quantistiche, rendendolo **più semplice e più trainabile** rispetto a HEE2 o CHE.
- Questo encoding è stato **il più performante** nello studio, con accuratezza fino al **99.84%**.

BINOMIAL INFERENCE: ESTIMATING ACCURACY AND UNCERTAINTY

Objective: Estimate the unknown success probability p of a **binomial process using Maximum Likelihood Estimation** (MLE), and construct a **uncertainty interval** around the estimate based on the log-likelihood ratio and Wilks' theorem.

Context: An experiment consists of N=1000 independent Jet trials, where K=960 successes are observed. **The underlying success probability p is unknown and must be inferred from the data**.

Goal: Understand how **likelihood-based inference** works in the binomial model, and how uncertainty intervals can be constructed using theoretical thresholds from the chi-squared distribution.

See Appendix - Why a Binomial Distribution Was Chosen
See Appendix - Definition of Confidence Interval

MACHINE LEARNING AND LIKELIHOOD-BASED INFERENCE

Inference Type	Description
Frequentist	Estimates probabilities from empirical frequencies; no priors involved.
Discriminative ML	Learns $P(y \mid x)$: conditional probability of class given features.

The machine learning models used in jet classification—such as CNNs, QCNNs, or GNNs, are grounded in **statistical theory**, specifically in **likelihood-based inference**

- The model learns to estimate the probability **P(class|features)**, such as the probability that a given jet image belongs to a specific class, given its observable features**.**

Where:
class: This represents the jet category being predicted. In this study, it refers to either a **top-quark jet** or a **QCD jet**.

So, class ∈ {top, QCD}.

features: These are the measurable properties extracted from the jet image. They include pixel intensities (representing energy fractions), spatial distributions of particles. These features serve as input to the CNN or QCNN models.

So, the model estimates the likelihood that a jet with certain features belongs to a particular class—essentially performing **statistical inference for jet classification**

Some considerations on the LH function

The probability density function p(x|m) is a parametric function of the observable data x.

The likelihood function L(m) is a function of the unobservable parameter m.

The pdf, a probability density of the data (random variable), should be normalized to unity over the domain of the random variable.

$$\int_X p(x|m)dx = 1$$

The likelihood, a function of the parameter m, obeys no specific normalization.

$$\int_M p(x_0|m)dm = ?$$

In addition, the function values L(m) are invariant under reparametrization of m into f(m): L(m) = L[f(m)]. No Jacobians here, reinforcing the notion that L(m) is not a pdf for m.

From: Statistical Treatment and Analysis of the Data by Annovi

Likelihood based confidence intervals

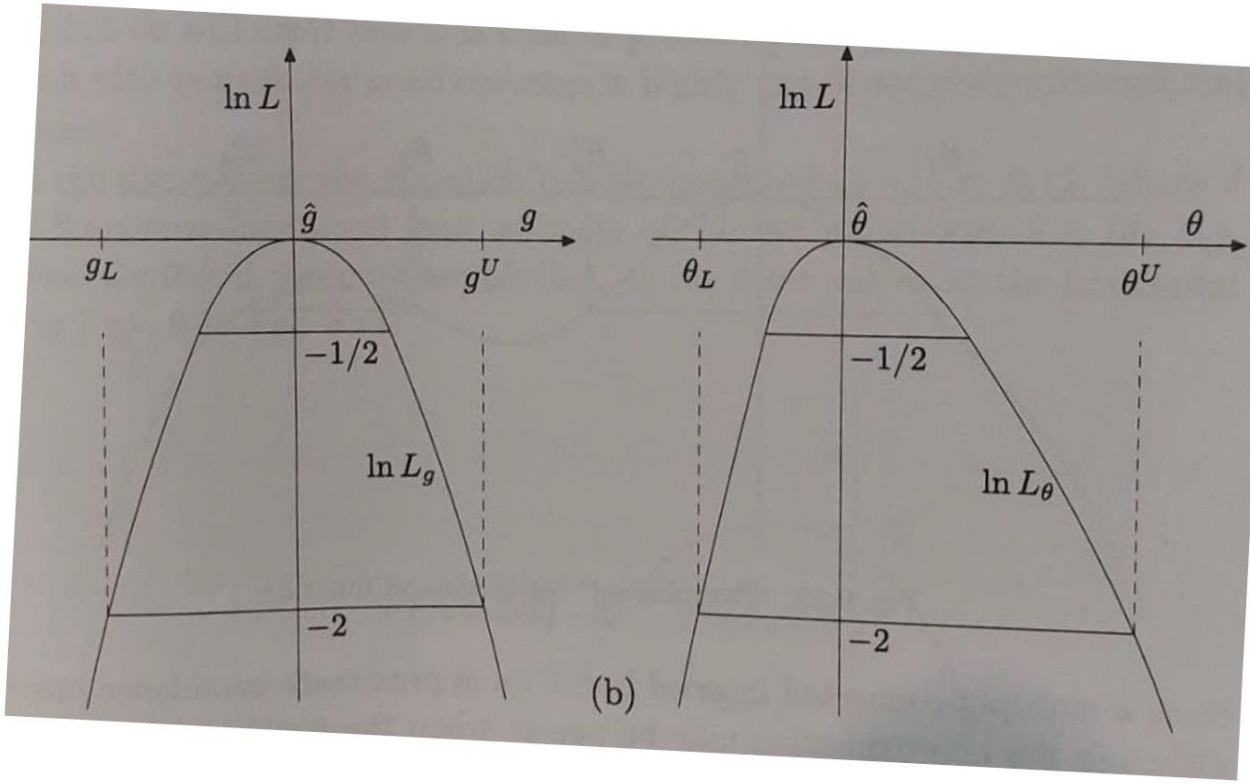
The likelihood is invariant for parameter transformation

Let's suppose we have the ln L on the right, with a change of variable we can make it parabolic (left plot). The original likelihood needs to be regular enough to have a 1-1 transformation

The parabolic limits at -½ or -2 are the 1σ, and 2σ limits.

Since the likelihood and the likelihood ratio are invariant, the 1σ, and 2σ limits can be derived based on the ln L values at -½ or -2 directly on the right plot.

This method is approximate. It should be correct to order 1/N, but not to higher (smaller) orders.



From: Statistical Treatment and Analysis of the Data by Annovi

DEFINITION OF STOCHASTIC PROCESS

A **stochastic process** is a collection of **random variables** indexed by time or space, used to describe the **evolution of a system under uncertainty**. Each variable represents the state of the system at a given point, and the process captures how these states change in a probabilistic manner.

In mathematical terms:

A stochastic process is a family of random variables $\{Xt\}_{t \in T}$ defined on a probability space, where T is the index set (often representing time), and each X_t takes values in a state space S.

Approximate models

The model p(x ; μ) is assumed as your best approximation of the relationship between μ and x relevant for the problem at hand.

Systematic uncertainty is, in any statistical inference procedure, the uncertainty due to the incomplete knowledge of the probability distribution of the observables.
G. Punzi, What is systematics ?

=> Parametrize differences with actual physics trough additional dependence of nuisance parameters p(x|μν).

The unknown ν values are uninteresting for the measurements but do influence the outcome. Lack of knowledge of ν introduces an uncertainty in the shape of p(x|μν).

Uncertainties in the shape of p(x|μ) reflects into the systematic uncertainty of the inference

From: Statistical Treatment and Analysis of the Data by Annovi

See Appendix - Statistical Uncertainty
See Appendix - Systematic Uncertainty
See Appendix - Model Uncertainty

Sufficiency principle

A statistics s is sufficient for μ if it keeps the full information on μ of a original sample.

Principle of Sufficiency : If $s(x_1...,x_n)$ is sufficient for μ, then any inference on μ depends on the sample $x_1...,x_n$ only through the statistics $s(x_1...,x_n)$.

Definition: S is sufficient for μ if the conditional probability of the sample $x_1...,x_n$ given a certain value $s(x_1...,x_n)$ does not depend on μ.

$p(x_1...,x_n | s(x_1...,x_n), \mu) = p(x_1...,x_n | s(x_1...,x_n))$

DEFINITIONS (PROBABILITY)

Random event: an event that has >1 possible outcome. The outcome isn't predicted deterministically, but a probability for each outcome is known.

"(Random) variables" or **"observables"**: Random events are associated to variables x, which take different values, corresponding to different possible outcomes.
Each x value has its probability **p(x)**. The outcomes generate a probability distribution of x.

Population (observable space): A collection of random events that forms the hypothetical infinite set of repeated independent and (nearly) identical experiments.

Observed distributions: finite-size random samplings from the corresponding population's parent distributions.

See Appendix -Probability (I)

From: Statistical Treatment and Analysis of the Data by Annovi

DEFINITIONS (INFERENCE)

Some definitions

Observables: x,y These are the quantities we measure in the experiment

Parameters: μ,ν, These are the parameters we want to estimate.

The estimate of the parameters are based on statistics:

Statistics: s(x,y) are functions of the observables ONLY. They are used to estimate the values of the parameters.

NOTE: Since the observables are following a probability distribution, s is following a probability distribution

Statistical inference is the science that studies how to use statistics (s) to do inference on the parameters

Definition: Statistical inference is the process of drawing conclusions about a population based on a sample of observed data.

In Particle Physics Context:

- We observe jet features (energy, momentum, angular distributions).
- These are treated as **random variables**.
- The goal is to infer the **underlying class** (top vs QCD) from these observables.

See Appendix - Inference

From: Statistical Treatment and Analysis of the Data by Annovi

GLUCONS

Definition of Gluons in QCD (Quantum Chromodynamics)

In **Quantum Chromodynamics (QCD)**, **gluons** are the **elementary gauge bosons** that mediate the **strong interaction** between quarks. They are the force carriers of QCD, analogous to photons in electromagnetism, but with a crucial difference: **gluons themselves carry color charge**, allowing them to interact with each other.

- Gluons are **massless**, **spin-1** particles.
- There are **eight types** of gluons, corresponding to the eight generators of the SU(3) color gauge group.
- Their self-interaction leads to key QCD phenomena such as **color confinement** (quarks and gluons are never observed in isolation) and **asymptotic freedom** (quarks behave as free particles at high energies).

Input Parameters of Interest

- THESE FEATURES ALLOW TO UNDERSTAND THE THREE-DIMENSIONAL GEOMETRY OF THE DETECTOR AND ANALYZE HOW ALARMS PROPAGATES DURING THE EVENT.
- TEMPORAL AND KEY PARAMETERS HELP DISTINGUISH ALARM EVENTS FROM BACKGROUND NOISE, SUCH AS WAVE LOG.

Feature	Description
Entity	The object or system component that triggered the alarm (e.g., server, database, application). Acts as the spatial anchor of the event.
time	Timestamp of the first event detection
AlertGroup	The family or category of the alarm (e.g., network, security, application). Provides semantic context similar to identifying the interaction type or topology when high-energy collide (e.g. at the LHC)
AlertKey	A unique identifier or subcategory within the AlertGroup (e.g., “CPU overload” under “Server Health”). Refines the classification—like distinguishing between Top Quark Jet and QCD Jet.
Message	The descriptive summary of the alarm, often including timestamp, severity
Severity	A predefined classification of alarm gravity, indicating the urgency and potential impact of an event. Common levels include *critical*, *major*, *minor*, and *informational*, guiding the response priority.

TWO LEVELS OF PARAMETERS IN IT-SPHERE GNN RECONSTRUCTION

- The **node features** (Entity, Timestamp, Severity, Alertgroup, AlertKey, Message, describe the detector response.
- The **target parameters** (Impact, Root cause, Alarm vs Log waves, etc.) are the quantities we want to infer from that response.
- The **sensor readings** (position, time, intensity) are your **inputs**.
- The **event that caused those signals** is the **target** you want to reconstruct.

Type	Examples	Role in high-energy collide
Input Features (Node-level)	**Entity** (server, database, application), **Timestamp** (t), **Severity** (critical, major, minor), **AlertGroup, AlertKey, Message**	These are the observable quantities for each alarm.
Target Parameters (Event-level)	**Event impact**, **Root cause location, Alarm vs log wave** classification, **Structured vs diffuse alarm topology**	These are the quantities that QNN aims to reconstruct or classify for each event (alarms vs log waves)

QUANTUM ENCODING LAYER

- **Purpose**: Converts classical image data into quantum states.
- **Encodings tested**:
- **TPE (Tensor Product / Angle Encoding)**: Simple rotation-based encoding.
- **HEE1 / HEE2 (Hardware-Efficient Encoding)**: Uses native quantum gates, one or two layers.
- **CHE (Classically Hard Encoding)**: More complex, less trainable in this context.
- **Insight**: HEE1 showed best performance and trainability for this task.

QUANTUM CONVOLUTION AND POOLING

Convolution Layer

- Applies two-qubit gates to extract spatial correlations.
- Two circuit types:
    - **SO(4)**: Real-valued, fewer parameters (6 trainable).
    - **SU(4)**: Universal, more expressive (15 trainable).
- Connects neighboring qubits to simulate classical filters.

Pooling Layer

- Reduces system size by half.
- Uses CNOT + rotation gates.
- Repeated until only one qubit remains for final measurement.

MEASUREMENT AND OUTPUT

- Final qubit is measured using **Pauli-Z**.
- The expectation value is interpreted as the **classification score**.
- Output: **top-quark jet** or **QCD jet**.

Sufficiency of a statistics

Example:

- $x_1,\ldots,x_n$ ~**Bernoulli** (p)
  => the number of successes is sufficient
- $x_1,\ldots,x_n$ ~**Gaussian** with known variance and unknown mean value μ.
  => The sample mean is a sufficient statistics for μ.
- Same as before for **poissonian** with rate μ
- Same as before for **exponential** with characteristic time τ
- x_{max} is a sufficient statistics for m for the **uniform** U(0,m).
- What about $x_1,\ldots,x_n$ ~gaussian with unknown variance and known mean value μ?

ACCURACYIT: MEASURING QUALITY IN IT SYSTEMS

In IT environments, systems are monitored continuously for:

- **Volumes**: Number of transactions, requests, or data processed per minute.
- **Performance**: Response times, latency, throughput.
- **Errors**: Failed requests, exceptions, timeouts.

Each minute, the system generates a **batch of measurements**, which can be treated as a sample of events.

- Out of 1000 API calls in one minute, 960 return successful responses.
- This yields: **AccuracyIT=960/1000=0.96**

Examples

Binomial

$$p(\mathbf{k};p) = \prod_{i=1}^{N} \binom{n}{k_i} p^{k_i}(1-p)^{n-k_i} = \\ = \left(\prod_{i=1}^{N} \binom{n}{k_i}\right) \cdot p^{N\bar{k}}(1-p)^{Nn-N\bar{k}}.$$

Poisson

$$p(\mathbf{k};\mu) = \prod_{i=1}^{N} \frac{\mu^{k_i}}{k_i!} e^{-\mu} = \\ = \left(\prod_{i=1}^{N} \frac{1}{k_i!}\right) \cdot \mu^{N\bar{k}} e^{-N\mu}.$$

Exponential

$$p(\mathbf{t};\lambda) = \prod_{i=1}^{N} \lambda e^{-\lambda t_i} = \\ = \lambda^N e^{-N\lambda\bar{t}}.$$

TEMPORAL SAMPLING = STATISTICAL TRIALS

In IT:

- Each **minute** (or second) is a sampling window.
- Each **event** (e.g., transaction, ping, query) is a trial.
- Over time, you build a dataset of binary outcomes → just like in jet classification.

This means:

- You can apply **frequentist inference** to system logs.
- You can use **log-likelihood ratios** to compare system configurations.
- You can apply **Wilks' thresholds** to detect statistically significant performance drops.

EFFICIENCY

Definition: **Efficiency** is the probability that a physical event (e.g., a top jet) is **correctly identified** by the selection system or classifier.

$$\text{Efficiency} = \frac{\text{Number of correctly selected signal events}}{\text{Total number of signal events present}}$$

Context:

- Used in **experimental physics** to evaluate detector or algorithm performance.
- Refers **only to the events of interest** (e.g., top jets), not the entire dataset.

Example: If the dataset contains 500 top jets and the model correctly identifies 480 of them:

Efficiency=480/500=96%

Statistics and inference

We want to make inference on the parameter m of our model by using our dataset $x_1,\ldots,x_n$

The statistics $s(x_1,\ldots,x_n)$ is a reduction of the size of data.

What if we do inference on our parameter m by using $s(x_1,\ldots,x_n)$?

Note: if $s(x_1,\ldots,x_n)= s(y_1,\ldots,y_n)$, the inference on m using s will be identical.

Does it mean that the inference using $x_1,\ldots,x_n$ and $y_1,\ldots,y_n$) would give the same identical inference on μ?

In general no! Is there anything special when instead this is happening?

Goal: We want to use statistics that are not removing information we need, just the inessential ones....

Two principles: Sufficiency and Likelihood

Some first properties of the Likelihood

- If we have more independent variables, the likelihood is the product of the L(m | x) for the single variable.
- Since sums are easier to be handled then multiplication, one uses logL(m|x) instead of L(m|x)
- **NOTE: L is a function of m, as such, it has a domain. It is very important to verify if the domain depends on m.**
- If you have several replicas of x, you will have several different functions L. Each functions will show a slightly different behaviors WRT m.
- If you fix a certain value of m=m0 in the domain of L, L(m=m0|x) becomes a statistics. One can calculate its PDF
- Similarly, one can define the point at which L is maximum for given x data. That again is a statistics, and so one can calculate the PDF.
    - This is the PDF of the which is the parameter value that maximizes the likelihood as function of the x data

From: Statistical Treatment and Analysis of the Data by Annovi

ACCURACY

These two terms may sound similar, but they have distinct meanings and roles depending on the context — especially in jet classification and statistical inference.

Accuracy

Definition: **Accuracy** is the proportion of correct classifications out of the total number of classifications made.

Accuracy=Number of correct classifications/Total number of total samples

Context:

- Used to evaluate **classification models** (CNN, QCNN, etc.)
- It's a **global metric**: includes both true positives and true negatives.

  Example: In the QCNN paper, if the model correctly classifies 960 out of 1000 images:

Accuracy=960/1000 -> 96%

PHYSICAL INTERPRETATION

- In the study, the authors train quantum and classical neural networks to classify **jet images** as either **top-quark jets** or **QCD jets**, using the JetNet TopTagging dataset. Each image is labeled, and the model outputs a prediction: correct or incorrect.

This setup naturally leads to a **binomial model**:

- Each classification is a Bernoulli trial (success/failure).
- The total number of trials is N=1000 (or a subset, e.g. test set).
- The number of correct classifications is K, which varies depending on the model and setup.

Physical Meaning in Jet Classification

- **Efficiency**: The MLE p^=0.96 tells us the best estimate of the model's ability to correctly classify jets. See Appendix - Efficiency
- **Uncertainty**: The confidence interval (e.g. [0.942, 0.974]) tells us how precise that estimate is.
- **Model comparison**: If a CNN has p^=0.93 with a wider interval, the QCNN may be statistically superior.
- **Experimental relevance**: This efficiency affects downstream physics analyses — e.g. cross-section measurements, background rejection, signal purity.

ACCURACY VS EFFICIENCY

In your binomial pipeline:

- If you consider **all classified events**, you're measuring **accuracy**.
- If you focus only on **top jets** and ask how many were correctly identified, you're measuring **efficiency**.

Concept	Accuracy	Efficiency
What it measures	Correctness over **all examples**	Correctness over **target events only**
Denominator	All examples (top + QCD)	Only signal examples (e.g., top jets)
Typical context	Machine learning, classification	Experimental physics, event selection
Sensitivity	Includes false positives and false negatives	Focuses on true positives

EXAMINER-LEVEL INSIGHT

By applying binomial inference:

- You've translated raw classification counts into a **statistically rigorous estimate**.
- You've used **likelihood theory** and **Wilks' theorem** to extract uncertainty.
- You've interpreted the result in terms of **detector performance**, **model reliability**, and **physics impact**.

  This shows mastery of both statistical inference and its physical application — exactly what your examiners are looking for.

🧠 Two Ways to Find the Most Likely Value $\hat{p}$

- **1. Analytical Calculation:** $\hat{p} = \frac{K}{N}$
- This is the **Maximum Likelihood Estimation (MLE)** method for the binomial distribution.
- It comes from differentiating the log-likelihood:

$$\log L(p) = K \log p + (N - K) \log(1 - p)$$

and solving:

$$\frac{d}{dp} \log L(p) = 0 \quad \Rightarrow \quad \hat{p} = \frac{K}{N}$$

TWO WAYS TO FIND THE MOST LIKELY VALUE P^

Analytical Calculation

TWO WAYS TO FIND THE MOST LIKELY VALUE P^

Numerical or Graphical Search for the Maximum of $L(p)$

🔹 **2. Numerical or Graphical Search for the Maximum of** $L(p)$

- You evaluate $L(p)$ over an interval (e.g. $p \in [0.9, 1.0]$) and look for the peak.
- Useful when:
  - Analytical derivation is hard or impossible
  - You want to **visualize** the likelihood shape
  - You're working with more complex models (e.g. GNNs, QCNNs, nonlinear systems)

✅ WHY USE BOTH?

🔍 In Practice:

- You compute p^=0.96 = 0.96 analytically
- Then you **verify** that the peak of L(p) in the plot is at p=0.96
- This **confirms that theory and practice align** — a great way to show **consistency** between approaches

✅ Why Use Both?

Method	Advantages	Purpose
Analytical $\hat{p} = K/N$	Fast, exact, theoretical	Direct estimation
Graphical $\max L(p)$	Visual, intuitive, generalizable	Verification, teaching, extension

COMPUTE THE MAXIMUM LIKELIHOOD ESTIMATE (MLE)

Compute the Maximum Likelihood Estimate (MLE)

$$p\_hat = K/N$$

$$print(f"MLE\ estimate\ cutro\ for\ p:\ \{p\_hat:.4f\}")$$

Define the range of p values to explore:

$$p\_values = np.linspace(0.9,\ 1.0,\ 500)$$

- `0.9`: lower bound of the interval
- `1.0`: upper bound
- `500`: number of points in the interval This creates a fine grid of 500 values between 0.9 and 1.0 to evaluate the likelihood function.

COMPUTE THE BINOMIAL LIKELIHOOD FOR EACH P

Compute the binomial likelihood for each p

$$likelihood = binom.pmf(K, N, p\_values)$$

For each value of p, compute the probability of observing exactly K=960 successes out of N=1000 trials. This is the likelihood function L(p)

Store results in a DataFrame:

$$df = pd.DataFrame(\{"p": p\_values, "likelihood": likelihood\})$$

Create a table with two columns: one for p, one for L(p). Useful for plotting and further analysis.

LIKELIHOOD ANALYTIC EQUATION

Since the binomial coefficient (N K) is constant with respect to p, we can ignore it when comparing likelihood profiles. So we work with the **log-likelihood**:

$$\log L(p) = K \log p + (N - K) \log(1 - p)$$

PLOT THE LIKELIHOOD FUNCTION

- Plot the likelihood curve
- Mark the MLE p^=0.96 with a vertical red dashed line

```
plt.figure(figsize=(8, 5))
plt.plot(df["p"], df["likelihood"], color="darkblue", label="L(p)")
plt.axvline(K/N, color="red", linestyle="--", label=f'MLE: p = {K/N:.3f}')
```

COMPUTE THE LOG-LIKELIHOOD RATIO

- *L_max = binom.pmf(K, N, p_hat)*
- *log_likelihood_ratio = -2 * np.log(likelihood / L_max)*

Where*:*

- `L_max`: maximum likelihood value at p^
- `log_likelihood_ratio`: measures how much worse each p is compared to the best one.

This is the Wilks statistic:

$$\mathrm{LLR}(p) = -2\log\left(\frac{L(p)}{L(\hat{p})}\right)$$

PLOT THE LOG-LIKELIHOOD RATIO

```
plt.figure(figsize=(8, 6))

plt.plot(df["p"], df["log_L_ratio"], color="darkgreen", label=r"$-2 \log \left( \frac{L(p)}{L(\hat{p})}
    \right)$")

plt.axhline(1, color="gray", linestyle="--", label="Wilks 1σ threshold")

plt.axhline(4, color="gray", linestyle=":", label="Wilks 2σ threshold")

plt.axvline(p_hat, color="orange", linestyle="--", label=f"MLE p = {p_hat:.3f}")
```

- Plot the log-likelihood ratio curve
- Add horizontal lines at 1 and 4 (Wilks thresholds for 1σ and 2σ)
- Mark the MLE again

DEFINE THE LOG-LIKELIHOOD RATIO FUNCTION

DEFINE THE LOG-LIKELIHOOD RATIO FUNCTION

- *def llr(p):*
- *return -2 * np.log(binom.pmf(K, N, p) / L_max)*

This function computes the log-likelihood ratio for any value of p. **Needed for root-finding**.

Find the 1σ confidence interval:

from scipy.optimize import brentq

p_lower = brentq(lambda p: llr(p) - 1, 0.85, p_hat)

p_upper = brentq(lambda p: llr(p) - 1, p_hat, 0.999)

- Use **Brent's method to find the values of p where the log-likelihood ratio equals 1**
- These are the lower and upper bounds of the 68% confidence interval

FIND THE 1Σ CONFIDENCE INTERVAL

- Use Brent's method to find the values of pp where the log-likelihood ratio equals 1
- These are the lower and upper bounds of the 68% confidence interval

from scipy.optimize import brentq

p_lower = brentq(lambda p: llr(p) - 1, 0.85, p_hat)

p_upper = brentq(lambda p: llr(p) - 1, p_hat, 0.999)

Print and visualize the confidence interval:

print(f"Intervallo di confidenza 1σ: [{p_lower:.5f}, {p_upper:.5f}]")

plt.axvline(p_lower, color='green', linestyle='--', label=f'p_lower = {p_lower:.4f}')

- *plt.axvline(p_upper, color='blue', linestyle='--', label=f'p_upper = {p_upper:.4f}')*
- Display the confidence interval
- Add vertical lines at the bounds to the plot

DEFINITION LLR(P)

The formula for the **log-likelihood ratio** (**LLR**) in the binomial model is:

This version is often used in numerical implementations to avoid computing the binomial coefficient repeatedly.

$$\mathrm{LLR}(p) = -2\log\left(\frac{L(p)}{L(\hat{p})}\right)$$

Where:

- $L(p) = \binom{N}{K} p^K (1-p)^{N-K}$ is the binomial likelihood for a given value of p
- $\hat{p} = \frac{K}{N}$ is the maximum likelihood estimate (MLE)
- $L(\hat{p})$ is the likelihood evaluated at the MLE

Since the binomial coefficient $\binom{N}{K}$ is constant for fixed N and K, it cancels out in the ratio. So the simplified expression becomes:

$$\mathrm{LLR}(p) = -2\log\left(\frac{p^K (1-p)^{N-K}}{\hat{p}^K (1-\hat{p})^{N-K}}\right)$$

This version is often used in numerical implementations to avoid computing the binomial coefficient repeatedly.

LLR(P) - INTERPRETATION (I)

- **Likelihood comparison**: It compares the likelihood of any value p to the maximum likelihood at p^.
- **If p=p^, then L(p)=L(p^) and LLR = 0 → perfect fit.**
- **Penalty for deviation**: As p moves away from p^, the **likelihood decreases, and the LLR increases**. *This reflects how incompatible that value of p is with the observed data*.
- **Logarithmic scale**: The log transformation makes the comparison more sensitive to small differences and ensures symmetry around the peak.
- **Multiplying by −2**: This scaling aligns the statistic with a chi-square distribution under regular conditions (Wilks' theorem), allowing us to use standard thresholds for confidence intervals.

WHAT IS THE LIKELIHOOD PRINCIPLE?

You have a **statistical model** p(x|m), where:

- x is the observed data
- m is the unknown parameter

L(m|x) is the **likelihood function**: it tells us how compatible the parameter m is with the data x

LLR(P) - INTERPRETATION (III)

Physical Analogy

Imagine you're reconstructing a neutrino energy from IceCube data:

- p^ is your best estimate.
- The LLR tells you how much worse other energy values are, based on how well they explain the observed hits.
- ***The confidence interval*** *is the range of energies that are statistically compatible with the data.*
- The LLR curve is U-shaped, centered at p^.
- **The horizontal lines at LLR = 1 and LLR = 4 mark the 1σ and 2σ thresholds.**
- **The vertical lines at the intersection points define the confidence interval.**

LLR(P) INTERPRETATION (II)

- Confidence Interval via Wilks
- **Wilks' theorem**: For large samples and regular models, the LLR follows a chi-square distribution with degrees of freedom equal to the number of parameters tested.
- **Thresholds**:
  - LLR=1 → 68% confidence level (1σ)
  - LLR=4 → 95% confidence level (2σ)
- **Visual meaning**: *The confidence interval includes all values of p for which the LLR is below the threshold. It's the region where the likelihood is "not significantly worse" than the maximum.*

STATISTICS

What it is: a **statistic** is a function of the measured observables, e.g. s(x), used to estimate unknown parameters of the model.

Examples:

- Sample mean $\bar{x}$
- Standard deviation
- Test statistic t(x) for hypothesis testing

Role: The statistic is the operational tool used for inference. It is not the model, not the probability—it's a number computed from data.

A **statistic is a number computed from the data**. It's not theoretical—**it's concrete: the mean, the standard deviation, the value of a test statistic.**

Example:

You observe 3, 4, 5, 6 photons in 4 events. The sample mean is:

$\bar{x}=3+4+5+6/4=4.5$

This is a **statistic**: it helps you estimate μ, but it's not a probability.

STATISTICS VS PROBABILITY

- These two concepts are often confused, but they play very different roles in statistical inference. Let's break it down clearly:
  Key Difference
- **Probability** belongs to the model: it tells you how likely a data point is.
- **Statistics** belong to the data: they summarize or analyze what you've observed.

Definition	A function of observed data	A theoretical measure of how likely an event is
Depends on	Measured data x	The model $p(x;\mu)$
Purpose	To estimate parameters or test hypotheses	To describe uncertainty in the model
Example	Sample mean $\bar{x}$, variance, test $t(x)$	$p(x=3 \mid \mu=5)=0.14$ (Poisson)

MODEL (I)

- **What it is**: The **model** is your theoretical assumption about how the data are generated. Formally, it's the probability density function $p(x; \mu)$ that describes the likelihood of observing x given a parameter value μ

- **Examples**:
- Gaussian distribution for measurements with uncertainty
- Poisson distribution for event counts
- Monte Carlo simulation model for IceCube events
- **Role**: The model is the bridge between theory and observables. It's used to compute probabilities, likelihoods, and guide inference.

MODEL (II)

The **model** is your theoretical description of how the data are generated. It's a mathematical function that links the parameters of interest (like energy, direction, or signal strength) to the observables (like hits in DOMs, reconstructed tracks, etc.).

Formally:

- Model=p(x; μ)
- x: observed data (e.g., number of hits, timing, energy)
- μ: parameters (e.g., true energy, signal strength, neutrino type)

  Example:

In IceCube, you might model the number of hits in a DOM as a Poisson distribution:

$$p(n;\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$$

Here, the model is the Poisson function, and λ is the expected number of hits (depends on energy, geometry, etc.).

PROBABILITY (I)

What it is: **Probability** is a measure of the expected frequency of an event. In frequentist statistics, it's defined as the limit of relative frequency in repeated experiments.

Examples:

- p(x): probability of observing x
- p(x|μ): conditional probability **given the parameter** (μ fixed)
- p(A|B): conditional probability (Bayes)

Role: **Probability describes uncertainty in the observables, given a model**. It's used to:

- Define distributions
- Compute p-values
- Build confidence intervals

PROBABILITY (II)

Probability is a numerical value that tells you how likely a specific outcome is, given a model.

Formally:

$$\textbf{Probability} = p(x = x0 \mid \mu)$$

It's the value you get when you plug your data x_0 into the model **for a given parameter μ.**

Example:

Using the Poisson model above, **if you expect λ=5** hits, the probability of observing exactly 3 hits is:

$$p(n = 3; \lambda = 5) = \frac{5^3 e^{-5}}{3!} \approx 0.14$$

PROBABILITY VS MODEL

Concept	What it describes	Depends on	Role in inference
Model	The full function $p(x;\mu)$	Parameters μ	Describes how data are generated
Probability	A number $p(x_0;\mu)$	Data x_0, parameters μ	Quantifies likelihood of an outcome

LIKELIHOOD

What it is: The **likelihood** is the function $L(\mu)=p(x_{obs};\mu)$, i.e. the probability of observing the actual data **xobs** given a parameter value μ. **Unlike probability, here the data are fixed and the parameter varies.**

Examples:

- $L(\mu)=\prod_i p(x_i;\mu)$ for independent data
- Likelihood ratio: $\Lambda=L(\mu 0)/L(\mu\hat{})$
- Profile likelihood: maximized over nuisance parameters

Role: **Likelihood is the core of frequentist inference. It's used to:**

- Estimate parameters (maximum likelihood)
- Build tests (Neyman-Pearson, Feldman-Cousins)
- **Derive confidence intervals (Wilks)**

🧭 Summary Table

Concept	Depends on	Fixed Variables	Varying Variables	Main Purpose
Statistic $s(x)$	Data	Parameters	Observables	Estimate parameters
Model $p(x;\mu)$	Theory	Parameters	Observables	Describe the process
Likelihood $L(\mu)$	Observed data	Observables	Parameters	Infer parameters
Probability $p(x)$	Model	Parameters	Observables	Quantify uncertainty

INFERENCE (I)

Inference is the broader process of drawing conclusions about p using the likelihood and other statistical tools.

In this binomial example, inference might include:

- **Point estimation**: Using the MLE $\hat{p}=0.96$
- **Interval estimation**: Constructing a confidence interval for p e.g. using Clopper-Pearson or l**ikelihood ratio methods**
- **Hypothesis testing**: Testing $H0:p=0.95$ vs $H1:p\neq0.95$
- **Goodness of fit**: Evaluating whether the binomial model with $p=0.95$ fits the data well.

INFERENCE (II)

- **Inference** is the goal.
- It uses the likelihood (and other tools) to make decisions or estimates.
- It answers questions like "What is the best estimate of p?", "Can we reject H_0?", or "How uncertain is our estimate?"

Concept	Likelihood	Inference
What it is	A function $L(p)$ based on fixed data	A process of reasoning using data and models
Depends on	Fixed data $k = 960$, variable parameter p	Likelihood, data, statistical framework
Purpose	Quantify how well each p explains the data	Estimate, test, or bound the parameter p
Example	$L(p) = \binom{1000}{960} p^{960}(1-p)^{40}$	"We estimate $p = 0.96$, with 95% CI = [0.94, 0.97]"

APPENDIX - SUMMARY CHI QUADRO

✅ **Summary**

Symbol	Meaning
$\Lambda(p)$	Relative log-likelihood statistic
χ^2_1	Limiting distribution under Wilks' theorem
$\Lambda(p) < 1$	Values of p compatible at 68% level
Convergence	Justifies using thresholds to build intervals

🔹 **3. How is it used to build confidence intervals?**

Thanks to this convergence, we can say:

- Values of p such that $\Lambda(p) < 1$ form a **68% confidence interval** (i.e., 1σ)
- Values such that $\Lambda(p) < 4$ form a **95% confidence interval** (i.e., 2σ)

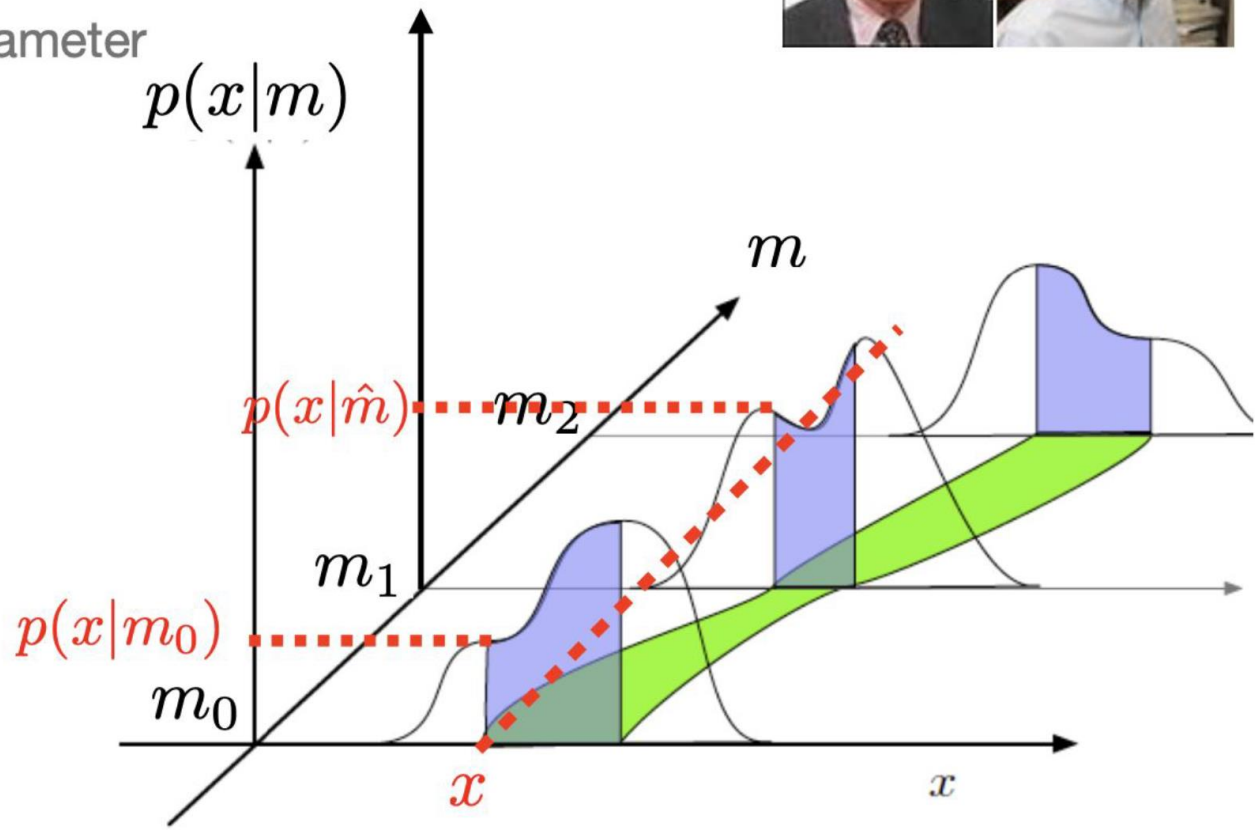
In practice:

$$\text{Confidence interval} = \left\{ p \mid \Lambda(p) < \chi^2_{\text{threshold}} \right\}$$

LR Ordering (Feldam-Cousin)

Those issues were solved by adapting a ordering, based on the likelihood ratio
Choose a value m_0 of the parameter and for each x calculate

$$\mathrm{LR} = \frac{p(x|m_0)}{p(x|\hat{m})}$$



The “accumulation score” of each element in x, no longer depends only on p(x|m_0) but also on p(x|m) at other m values

[Cranmer]

Likelihood ratio ordering

1. Choose one value for m, m_0 and generate simulated pseudodata accordingly.
2. For each observation x calculate (i) the value of the likelihood at m_0, $p(x|m_0)=L(m_0)$ and (ii) the maximum likelihood $L(\hat{m})$ over the space of m values.
3. Rank all x in decreasing order of likelihood ratio $LR=L_x(m_0)/L_x(\hat{m})$.
4. Accumulate starting from the x with higher LR until the desired CL is reached.
5. Repeat for all m

As the likelihood is metric-invariant so is the ratio of likelihoods. Therefore LR-ordering preserves the metric, mostly avoids empty confidence regions and has several other attractive features. By far the most popular ordering in HEP.

Take LR-ordering as default option unless there are strong motivations against it.

m0 is “true” value

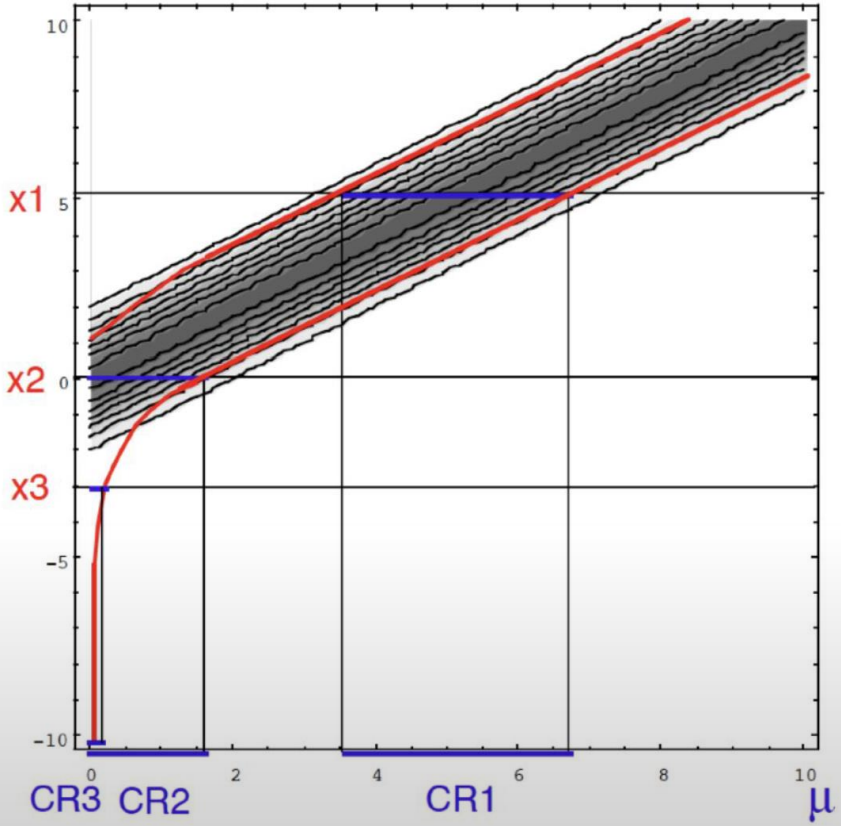
the pseudo data is data generated based on m0

$LR=L_x(m_0)/L_x(\hat{m})$, its ordering, and accumulation is done for each given m0.

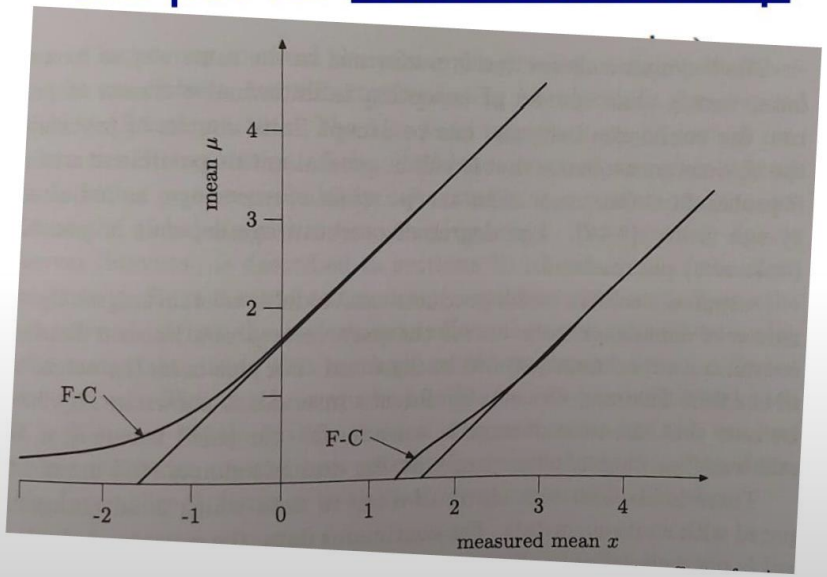
LR ordering in Practice

https://journals.aps.org/prd/abstract/10.1103/PhysRevD.57.3873

[Phys Rev D 57,3873 (1998)]



- Use a different ordering algorithm: *Likelihood-Ratio ordering*. NB: depends on the pdf for other values of μ



- Removes unpleasant empty intervals and avoids flip-flopping
- Invariant for change of *observable* (not mentioned in paper!)

Common type of inferences

- **Point estimation:** s(x) estimates the "best" value of μ in its space A

$$s(x) : x \mapsto \mu_x \in A$$

- **Interval estimation:** s(x) estimates the an interval for μ in its space A

$$s(x) : x \mapsto I_x(\mu) \subset A$$

- **Hypothesis testing**: s(x) decides this hypothesis H_i is favorite by X (i=0, 1, 2, etc.)

$$s(x) : x \mapsto i;\ \ H_i$$

- **Goodness of Fit (testing):** s(x) decides how well the data agree with the model.

$$s(x) : x \mapsto p_{value};\ \ p_{value} \in [0, 1]$$

UNCERTAINTY VS CONFIDENCE INTERVAL

Incertezza (Uncertainty)

Refers to the **degree of doubt** or **variability** in a measurement or estimate. It's a **general concept** that can be expressed in many ways: standard deviation, standard error, confidence intervals, credible intervals, etc.

- It quantifies how much the estimate might fluctuate due to sampling variability, noise, or model assumptions.
- In physics, uncertainty often refers to **instrumental or statistical error**.

Confidence Interval

A **specific statistical tool** used to express uncertainty about a parameter estimate.

- It defines a **range of values** that, with a given confidence level (e.g. 68%, 95%), is likely to contain the true value of the parameter.
- Based on sampling theory and often derived from likelihood ratios, standard errors, or bootstrap methods.

Example from Your Exercise

- You estimate $\hat{p} = 0.96$
- You compute a **confidence interval:**

$$[p_{\text{lower}}, p_{\text{upper}}] \quad \text{such that} \quad \Lambda(p) < 1$$

- This interval expresses the **uncertainty** in your estimate of p, but it is **not the only way** to quantify uncertainty.

LIKELIHOOD-BASED INFERENCE

In frequentist statistics, likelihood-based inference estimates unknown parameters by evaluating the likelihood function:

$$L(\theta;x)=\mathbf{P}(x \mid \theta)$$

This function expresses how probable **the observed data x** is under different values of the parameter θ

- The goal is to find the value of θ that maximizes L(θ;x), i.e., the parameter that best explains the data.
  Observables x are random variables sampled from a distribution
- Parameters θ are unknown quantities to be inferred
- Statistics s(x) are functions of observables used to estimate θ
- Since x is stochastic, s(x) also follows a distribution

See Appendix - Frequentist Statistics

UNCERTAINTY VS CONFIDENCE INTERVAL

✅ **Summary Table**

Concept	Definition	Expressed As
Uncertainty	General measure of doubt or variability	Std. dev, error bars, CI, etc.
Confidence Interval	Specific range with statistical meaning	Based on likelihood or sampling

🧮 **Example from Your Exercise**

- You estimate $\hat{p} = 0.96$
- You compute a **confidence interval:**

$$[p_{\text{lower}}, p_{\text{upper}}] \quad \text{such that} \quad \Lambda(p) < 1$$

- This interval expresses the **uncertainty** in your estimate of p, but it is **not the only way** to quantify uncertainty.

WILKS' THEOREM

Statement of the Theorem

Wilks' Theorem states that, under regularity conditions and for large sample sizes, the **log-likelihood ratio statistic**:

converges in distribution to a **chi-squared distribution** with degrees of freedom equal to the number of parameters being tested.

Interpretation

- L(p): likelihood for a candidate parameter value
- L(p^): maximum likelihood
- Λ(p): measures how much worse p fits the data compared to p^

$$\Lambda(p) = -2\log\left(\frac{L(p)}{L(\hat{p})}\right)$$

Λ(p) measures **how much worse** a candidate value p explains the data compared to the optimal value p^

- It is **zero** at the MLE: Λ(p^)=0
- It **grows** as p moves away from p^

In Wilks' Theorem:

- Λ(p) asymptotically follows a **chi-squared distribution** with degrees of freedom equal to the number of parameters
- This allows us to define **uncertainty intervals**

📐 What Does $\Lambda(p) < \chi^2_{\text{threshold}}$ Mean?

- **Symbol meanings**

- $\Lambda(p)$: the **log-likelihood ratio statistic,** defined as:

$$\Lambda(p) = -2\log\left(\frac{L(p)}{L(\hat{p})}\right)$$

It measures how much worse a candidate value p fits the data compared to the best-fit value $\hat{p}$.

- $\chi^2_{\text{threshold}}$: a **critical value** from the chi-squared distribution, corresponding to a desired confidence level.

WILKS' THEOREM

In Wilks' Theorem:

- Λ(p) asymptotically follows a **chi-squared distribution** with degrees of freedom equal to the number of parameters
- This allows us to define **confidence intervals** by finding values of p such that:

WILKS' THEOREM

🎯 **Interpretation of the inequality**

> The inequality $\Lambda(p) < \chi^2_{\text{threshold}}$ defines the set of parameter values that are **statistically compatible** with the observed data at a given confidence level.

In other words:

- If $\Lambda(p) < 1$, then p lies within the **68% confidence interval** (1σ)
- If $\Lambda(p) < 4$, then p lies within the **95% confidence interval** (2σ)

Symbol	Meaning
$\Lambda(p)$	Log-likelihood ratio statistic
$\chi^2_{\text{threshold}}$	Critical value from chi-squared distribution
Inequality	Defines confidence region for parameter p

🧮 **In your exercise**

You use this inequality to find the values of p such that:

$$-2\log\left(\frac{L(p)}{L(\hat{p})}\right) < 1$$

This gives the **1σ confidence interval** around the MLE $\hat{p} = 0.96$.

WILKS' THEOREM: REGULARITY

For the log-likelihood ratio (LLR) to follow a chi-square distribution, several **regularity conditions** must be met:

- The **likelihood function must be sufficiently smooth** (i.e., differentiable with respect to the parameter).
- The **estimated parameter must lie in the interior** of the parameter space (not on the boundary).
- The **model must be identifiable**: each parameter value must correspond to a unique probability distribution.
- The **sample size must be large**: the chi-square approximation is **asymptotic**, meaning it becomes accurate as $N \to \infty$.

WHY A BINOMIAL DISTRIBUTION WAS CHOSEN

The **binomial distribution** is appropriate because the experiment involves:

- A fixed number of independent trials N=1000
- Each trial results in a binary outcome: **success** or **failure**
- The **probability of success p** is assumed to be constant across trials

  This matches the definition of a binomial process. Therefore, the number of observed successes K=960 follows a binomial distribution:

$$K \sim \text{Binomial}(N,p)$$

DEFINITION OF CONFIDENCE INTERVAL

A **confidence interval** is a range of values for an unknown parameter that is believed to contain the true value with a specified level of confidence.

It reflects the **uncertainty** in the estimate due to sampling variability.

Example from This Exercise

- You observe K=960 successes out of N=1000 trials
- You estimate p_hat = 0.96 using Maximum Likelihood
- Using the **log-likelihood ratio** and **Wilks' theorem**, you find the values of p for which:

$$-2\log\left(\frac{L(p)}{L(\hat{p})}\right) < 1$$

- This defines the **68% confidence interval** (1σ) around $\hat{p}$

Result:

$$\text{Confidence interval } (1\sigma) = [p_{\text{lower}}, p_{\text{upper}}]$$

This interval contains the values of p that are statistically compatible with the observed data at the 68% confidence level.

🔓 STRONG LIKELIHOOD PRINCIPLE

Even if x and y come from **different models**,

if $L(m|x)=\text{const}\cdot L(m|y)$ then they should give the same inference about m.

⚠️ But be careful:

- The strong principle **can be violated**, for example:
    - When the data collection method (experimental design) affects inference
    - In frequentist hypothesis testing, where the **p-value** depends on unobserved data (the "sampling space")

CROSS-ENTROPY LOSS

- **Purpose:** Measures the performance of a classifier that outputs probabilities between 0 and 1.
- **Formula:**
  $$L_{\text{Cross-Entropy}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(f_i) + (1-y_i)\log(1-f_i)\right]$$
  where:
  - y_i is the true label (0 or 1),
  - f_i is the predicted probability,
  - N is the number of training samples.
- **Use case:** Binary classification with probabilistic outputs.

MEAN SQUARED ERROR (MSE)

2. Mean Squared Error (MSE)

- **Purpose**: Measures the average squared difference between predicted and true values.
- **Formula**:
  $$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$
  where:
  - y_i is the true label (±1),
  - $\hat{y}_i$ is the predicted value.
- **Use case**: Regression or binary classification with ±1 labels.

HINGE LOSS

3. Hinge Loss

- **Purpose**: Commonly used in support vector machines; penalizes predictions that are not confidently correct.
- **Formula**:
  $$L_{\text{Hinge}} = \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i \cdot \hat{y}_i)$$
  where:
  - y_i is the true label (±1),
  - $\hat{y}_i$ is the predicted value.
- **Use case**: Binary classification with margin-based decision boundaries.

CONFIDENCE INTERVAL

A confidence interval provides a range of values that, with a certain probability (e.g., 95%), contains the true value of the parameter.

Definition: For a parameter estimate $\hat{\theta}$, a 95% confidence interval is:

$$\hat{\theta} \pm z_{0.975} \cdot SE(\hat{\theta})$$

where $z_{0.975}$ is the critical value from the standard normal distribution (≈ 1.96), and $SE(\hat{\theta})$ is the standard error of the estimator.

Interpretation: If we repeated the experiment many times, 95% of the intervals constructed this way would contain the true value of θ

STANDARD DEVIATION

Standard deviation **measures the spread of values around the mean**. In inference, it reflects how much an estimate can fluctuate across different samples.

In your case: If you train the QCNN 50 times, the standard deviation of the **accuracy** across runs tells you how stable the model is.

Formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

where x_i are the observed values and $\bar{x}$ is their mean.

VARIANCE

The variance of an estimator quantifies how much the estimate θ^ varies around its expected value.
This formula expresses the **variance of the estimator** θ, which is a measure of **how much the estimate of a parameter can fluctuate** around its average value if the experiment were repeated many times.

Formula:

$$\mathrm{Var}(\hat{\theta}) = E[(\hat{\theta} - E[\hat{\theta}])^2]$$

Low variance → high precision
High variance → the estimator is sensitive to data fluctuations

Interpretation

The variance of the estimator tells you **how precise** the estimator is. If the variance is **small**, the estimator is **stable** and gives similar values each time. If it's **large**, the estimator is **unstable** and can produce very different results depending on the data.

Step-by-Step Explanation

- $\hat{\theta}$: the estimator, i.e., the value you obtain from the data (e.g., the average accuracy of your QCNN).
- $E[\hat{\theta}]$: the expected value of the estimator, meaning the theoretical average you'd get if you could repeat the experiment infinitely.
- $(\hat{\theta} - E[\hat{\theta}])^2$: the squared distance between the estimator and its mean → this measures how much it "oscillates."
- $E[\,\cdot\,]$: the expectation operator → it takes the average of all those oscillations.

VARIANCE OF THE ESTIMATOR – TOP JET TAGGING

Application to Your Research

In the QCNN paper:

- Multiple runs (50 per setup) allow computing **the mean and standard deviation of accuracy**
- **Confidence intervals can be constructed to compare performance**
- Dimensional Expressivity Analysis (DEA) helps **reduce model variance by eliminating redundant parameters**

🧪 **Example in Your Context**

In the QCNN paper, suppose you estimate the average accuracy over 50 runs:

- Each run gives a value $\hat{\theta}_i$
- You compute the mean $E[\hat{\theta}]$
- Then you calculate the variance as:

$$\mathrm{Var}(\hat{\theta}) = \frac{1}{50}\sum_{i=1}^{50}(\hat{\theta}_i - \bar{\theta})^2$$

If the variance is low, it means your QCNN model is **robust** and **consistent**.

CONFIDENCE INTERVAL – TOP JET TAGGING

In the paper, each model configuration (e.g., SU(4) circuit with HEE encoding) is trained **50 times** to assess performance. Let's say you're analyzing the classification accuracy across those 50 runs.

Interpretation

We are 95% confident that the true accuracy of the QCNN model lies between **98.09% and 98.31%**.

This interval reflects the **statistical uncertainty** due to sample variability. It's a rigorous way to report model performance—not just a single number, but a range that accounts for randomness in training and data.

Suppose the QCNN model yields the following results:

- **Mean accuracy** across 50 runs:

$$\hat{\theta} = 0.982$$

- **Standard deviation** of accuracy:

$$\sigma = 0.004$$

- **Standard error of the mean**:

$$\mathrm{SE}(\hat{\theta}) = \frac{\sigma}{\sqrt{n}} = \frac{0.004}{\sqrt{50}} \approx 0.00057$$

To compute a **95% confidence interval**, use the formula:

$$\hat{\theta} \pm z_{0.975} \cdot \mathrm{SE}(\hat{\theta})$$

Where $z_{0.975} \approx 1.96$. So:

UNCERTAINTY IN STATISTICAL INFERENCE: SYSTEMATIC UNCERTAINTY

- Refers to uncertainty in the shape of the probability distribution $P(x|\mu)$, where μ are model parameters.
- If there are unknown nuisance parameters ν (e.g., preprocessing choices, detector effects), they affect the shape of $P(x|\mu,\nu)$.
- Lack of knowledge about ν introduces **systematic uncertainty** in the inference process.

UNCERTAINTY IN STATISTICAL INFERENCE: MODEL UNCERTAINTY

- This concerns whether the chosen model (CNN, QCNN, encoding type, loss function) is truly optimal for the task.
- The paper addresses this by comparing different architectures and circuits (SO(4), SU(4), HEE, CHE), and applying dimensional expressivity analysis (DEA) to reduce parameter redundancy while preserving performance.

BOOSTED TOP QUARKS COMPLICATE THINGS

- At high energies (e.g. HL-LHC), top quarks are **highly boosted** → their decay products are tightly packed into a single jet.
- These **boosted top jets** can look very similar to QCD jets in terms of shape, energy distribution, and substructure.
- Sophisticated classification methods (like CNNs or QCNNs) are needed to **tease apart subtle differences**.

HIGH-ENERGY COLLISIONS (TOP QUARK E JET)

- **“In high-energy collisions (e.g. HL-LHC)”** → In extremely energetic particle collisions, such as those at the Large Hadron Collider (LHC) or its high-luminosity upgrade (HL-LHC), particles are produced with very high energies.
- **“top quarks are often highly boosted”** → The top quarks generated in these collisions carry a large amount of momentum (relativistic boost), meaning they move very fast relative to the lab frame.
- **“decay products become collimated into a single jet”** → When the top quark decays (into a b-quark and a W boson), its decay products are so energetic that they don’t spread out in space. Instead, they travel in nearly the same direction, forming a tightly packed stream of particles—a **jet**.

HADRONIZE

In high-energy collisions (like those at the LHC), quarks and gluons are produced with a lot of energy. But here's the catch: **quarks and gluons can't exist freely**—they're always confined inside larger particles called **hadrons** (like protons, neutrons, pions, etc.).

So when these energetic quarks and gluons fly out of a collision, they undergo a process called:

Hadronization — the transformation of free quarks and gluons into bound, color-neutral particles (hadrons).

This happens because of the strong force described by **Quantum Chromodynamics (QCD)**. It's like nature saying: "You can't leave the party alone—you must form a group!"

HADRONIZE

- A high-energy **parton** (quark or gluon) initiates the process.
  See Appendix - Partons
- Through **hadronization**, it fragments into multiple **color-neutral hadrons**.
  See Appendix - Color-Neutral Particles
  See Appendix - What Does "Color-Neutral" Mean?
- These hadrons travel in roughly the same direction, forming a **jet**.

COLOR-NEUTRAL PARTICLES

What Are "Color-Neutral Particles"?

- In QCD, "color" is a property of quarks and gluons (not related to visual color). To be stable, particles must be **color-neutral**—meaning their color charges cancel out. Hadrons are such combinations.
- "Color" is a quantum property of **quarks** and **gluons**, not related to visual color.
- There are three types of color charge: **red**, **green**, and **blue** (and their corresponding anticolors).
- These are used to describe how particles interact via the **strong force**, which is mediated by **gluons**.

Why Do They "Travel in the Same Direction"?

When a high-energy quark or gluon is produced, it moves fast in a certain direction. As it hadronizes, it creates a **spray of particles** (hadrons) that follow roughly the same path. This spray is what we call a:

Jet — a collimated stream of hadrons resulting from the fragmentation of a high-energy parton.

Why Is This Important?

In experiments, we don't see quarks directly—we detect **jets**. So understanding hadronization is crucial for interpreting what happened in the collision. And in your paper, distinguishing **top-quark jets** from **QCD background jets** depends on analyzing these hadronized patterns

WHAT DOES "COLOR-NEUTRAL" MEAN?

- A particle is **color-neutral** (or "white") when its constituent color charges **combine to cancel out**.
- For example:
- A **baryon** (like a proton or neutron) contains three quarks: one red, one green, one blue → together they form a neutral combination.
- A **meson** contains a quark and an antiquark with opposite colors (e.g., red and anti-red) → also neutral.
- Only **color-neutral combinations** can exist as **free, stable particles** in nature.
- If a particle had a net color charge, it would be subject to confinement—it couldn't exist independently and would quickly form a neutral bound state.
- This is a consequence of **color confinement**: quarks and gluons are never observed in isolation, only in bound states that are color-neutral.

So in essence, **color-neutrality is a prerequisite for physical particles to be stable and observable**. It's a bit like electric neutrality in atoms—charged particles tend to combine into neutral systems to minimize energy and become stable.

PARTONS

The term *parton* was introduced by Richard Feynman in 1969 to describe the internal constituents of hadrons (like protons and neutrons) during high-energy collisions. Today, we understand that partons are:

- **Quarks**: fundamental particles with fractional electric charge and spin ½.
- **Gluons**: bosons that mediate the strong interaction, binding quarks together.

  In practice, when a proton is accelerated and collided, it doesn't behave like a rigid particle, but rather like a "bag" full of partons that interact with each other and with partons from other incoming particles.

GLUONS

Gluons are the carriers of the **strong interaction**, one of the four fundamental forces of nature. Here are their key properties:

Unlike photons (which mediate the electromagnetic force), gluons **interact with each other** because they carry color charge. This makes quantum chromodynamics (QCD) much more complex and fascinating.

Property	Value/Description
Type	Gauge boson
Spin	1
Mass	Theoretically zero (experimental limit < 20 MeV/c^2)
Electric charge	0
Color charge	Yes (eight distinct types)
Role	Bind quarks inside hadrons (protons, neutrons)

GLUONS: CONNECTION TO YOUR WORK

In the context of **jet tagging** and **hadronic events**, partons are the "seeds" from which **particle showers** (parton showers) develop, eventually forming the **jets** we observe. Gluons play a crucial role in these showers, emitting radiation and contributing to jet structure. This is exactly the kind of phenomenon your GNN and QCNN models aim to classify or reconstruct.

What Kind of "Radiation" Do Gluons Emit?

Unlike photons, **gluons do not emit electromagnetic radiation**. However, in the context of **QCD**, we talk about **gluon radiation** in an analogous way to photon emission in QED:

- When an **accelerated quark** (or another gluon) interacts via the strong force, it can **emit a gluon**.
- This process is known as **gluon emission** or **parton showering**.
- It's a form of **color radiation**, not light: gluons carry **color charge**, so their emission alters the chromodynamic configuration of the system.

  In practice, it's as if gluons "radiate" other gluons or quarks, generating **cascades of partons** that eventually hadronize into jets.

DO GLUONS CONTRIBUTE TO **JET TAGGING**?

1. **Jets Initiated by Gluons**

- Gluons, like quarks, can **initiate a parton shower** that leads to the formation of a **jet**.
- **Gluon jets** tend to have:
    - **Wider angular spread**
    - **Higher particle multiplicity**
    - **More diffuse energy distribution**
- These features are used to distinguish **gluon jets** from **quark jets**—a key aspect of **jet tagging**.

2. **Role in Hadronic Events**

- In high-energy collisions (like at the LHC), gluons are **often the initial partons** in interactions.
- Their abundance in protons (especially at low momentum fraction, i.e., low x) makes them central in **hadronic events**.
- Additionally, during **hadronization**, gluons combine with quarks/antiquarks to form **color-neutral hadrons**.

GLUONS SHAPE JET MORPHOLOGY

In your context—event reconstruction in IceCube or jet classification using GNNs/QCNNs—understanding **how gluons shape jet morphology** is crucial. For example:

- A **gluon jet** might mimic a **top jet** or a **QCD background jet**, complicating tagging.
- The **structural differences** between quark and gluon jets can be learned by neural networks (classical or quantum) to improve discrimination.

IS THE **GLUON** A SIGNAL OR NOISE?

Connection to Your Work

In your case—jet classification using GNNs or QCNNs—the gluon can be:

- A **signal**, if you're trying to distinguish **gluon jets** from other types.
- **Noise**, if you're trying to isolate **top-quark jets** and want to **suppress gluon jets** that resemble the signal.

  In short, the gluon is **ambivalent**: it can be either a protagonist or a distractor, depending on your experimental goal

OBSERVABLES ARE RANDOM VARIABLES"—WHAT IT MEANS

In high-energy physics experiments, **observables** like energy, momentum, and angular distributions are not fixed values. Instead, they behave as **random variables** because:

- Each collision or event is governed by **quantum mechanics** and **probabilistic interactions**.
- Even under identical initial conditions, the **final state particles** (and their measured properties) can vary.
- What we measure—like the energy of a jet constituent or the angle between decay products—is a **sample from a probability distribution**.
  we mean that:
- Their values **change from event to event**.
- They follow **statistical distributions** (e.g., Gaussian, Poisson, or more complex ones).
- We use **statistical inference** to extract meaningful patterns or classify events (e.g., top vs QCD jets).

WHAT IS A "STOCHASTIC PROCESS" IN THIS CONTEXT?

A **stochastic process** is the evolution of random variables over time or space. In particle physics:

- Each proton-proton collision (e.g., at the LHC) produces **different outcomes**, even under similar initial conditions.
- Jet formation depends on **quantum interactions**, **gluon emission**, **hadronization**, and **detector effects**—all of which have probabilistic components.
- Therefore, each jet is a **unique realization** of a process that can generate many different configurations.
- **Statistical inference**: you're trying to deduce the nature of a jet from an observed sample.
- **Probabilistic models**: like neural networks that output probabilities.
- **Metrics like AUC and cross-entropy**: which evaluate inference over random variables.

PERFORMS STATISTICAL INFERENCE

Saying that a model "performs statistical inference" means that it is **drawing probabilistic conclusions** from observed data, rather than making deterministic predictions. Here's what that implies in your context:
What Is Statistical Inference?

Statistical inference is the process of:

- **Estimating** unknown parameters of a population (e.g., the probability that a jet is from a top quark)
- **Testing** hypotheses (e.g., is this jet more likely to be QCD or top?)
- **Quantifying uncertainty** in decisions (e.g., error margins, confidence intervals)
  When a Model "Performs Statistical Inference"
- In your case, a model like a CNN, QCNN, or GNN:
- Receives **random observables** (energy, momentum, angular distributions)
- Learns to **map those observables** to a **probability of class membership** (top vs QCD)
- The output is not a binary answer, but a **probability distribution**—e.g., "this jet has a 92% chance of being from a top quark"

WHAT DOES "THIS REFLECTS A MAXIMUM LIKELIHOOD ESTIMATION APPROACH" MEAN?

This phrase means that the model is trying to **find the parameters** (or probabilities) that **maximize the likelihood of observing the given data**—in other words, the parameters that make the observed data **most probable** under the model.

What Is Maximum Likelihood Estimation (MLE)?

Definition: MLE is a statistical method that seeks the parameter values that **maximize the likelihood function**:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, L(\theta \mid x)$$

Where:

- xx is the observed data (e.g., jet features)
- θ are the model parameters (e.g., neural network weights)
- L(θ|x) is the **likelihood**: how probable the data is given the parameters

IN YOUR MODEL

When your QCNN or GNN minimizes **cross-entropy loss**, it's actually **maximizing the log-likelihood** of the correct class labels. In other words:

- The model tries to **assign high probabilities** to the correct classes (top, QCD) for each event.
- This is exactly what MLE does: it finds the parameters that make the observed data **most likely**.

PRECISION MEASUREMENTS

- Accurate jet classification improves measurements of **cross sections**, **branching ratios**, and **top quark properties**.
- It also helps reduce **systematic uncertainties** in experimental analyses.

Searches for New Physics:

- Many BSM models predict **top-rich final states**.
- Efficient top tagging allows physicists to **isolate rare events** that could hint at new particles or interactions.
  The paper explores whether **Quantum CNNs** can outperform classical CNNs in this classification task—especially in regimes where data is limited or the jet structure is complex.

WHY IS "WHY STATISTICS – INFERENCE" RELEVANT?

n top-quark tagging, we don't observe the top quark directly, but only the **products of its decay** (b-quark, W boson → leptons, neutrinos, jets). Therefore:

- The **observables** (energy, momentum, angular distributions) are **random variables**.
- The jets we observe are **realizations of a stochastic process**.
- The dataset (e.g., JetNet) is a **sample** from a **theoretical population** of events.
- The model (CNN, QCNN, GNN…) performs **statistical inference**: estimating the probability that a given jet is of top-quark or QCD origin.

JETNET DATASETS

- **Open-access datasets**: Includes benchmark datasets like *TopTagging*, with labeled jets (e.g. top vs. QCD) and particle-level features.
- **Preprocessing utilities**: Tools to convert raw jet data into formats suitable for ML, including jet images and particle clouds.
- **Model evaluation**: Built-in support for comparing architectures like CNNs, GNNs, and QCNNs on standardized tasks.
- **Integration with ML frameworks**: Compatible with PyTorch and TensorFlow, making it easy to plug into existing workflows.

Relevance to Your Work
In the paper you uploaded, JetNet's *TopTagging* dataset is used to train both classical CNNs and quantum CNNs (QCNNs) to distinguish top-quark jets from QCD jets. The dataset includes jets in the **hadronic channel**, and preprocessing steps (like PCA and Gram-Schmidt transformations) help reduce dimensionality and normalize jet images for training.

If you're working on top-tagging with GNNs or quantum models, JetNet is a great starting point for benchmarking and reproducibility.

SIGMA IN STATISTICS

- **Sigma (σ)** is the symbol for **standard deviation**, which measures **how spread out the data are** around the mean.
- Saying 1σ means you're considering a range that spans **one standard deviation above and below the mean**.

Physical interpretation of the interval: Clarify that the **1σ interval does not mean** that pp has a 68% probability of being in that range, but rather that the **data are statistically compatible** with those values

📊 **Practical interpretation of** 1σ

If your data follow a **normal (Gaussian) distribution**, then:

Interval	Percentage of data covered
$\mu \pm 1\sigma$	≈ **68.27%** of the data
$\mu \pm 2\sigma$	≈ **95.45%** of the data
$\mu \pm 3\sigma$	≈ **99.73%** of the data

So when your slide refers to a **68% confidence interval**, it's equivalent to saying **a** 1σ **interval** — assuming the distribution of your estimator is approximately Gaussian.

ACCURACYIT VS STATISTICAL ACCURACY

Both domains rely on **binary evaluations**:

- In **statistical inference**, each trial (e.g., jet classification) results in either a correct or incorrect prediction → modeled as a **Bernoulli trial** (success/failure).
- In **IT monitoring**, each system event (e.g., API call, database query) results in either a success or failure → also a Bernoulli trial.

Accuracy as a **Universal Metric**:

AccuracyIT=*Number of successes/Total number of trials*

This is the same formula whether you're classifying jets or monitoring server uptime.