# Calibration and Correlation:
## Learning Uncertainties the Frequentist Way

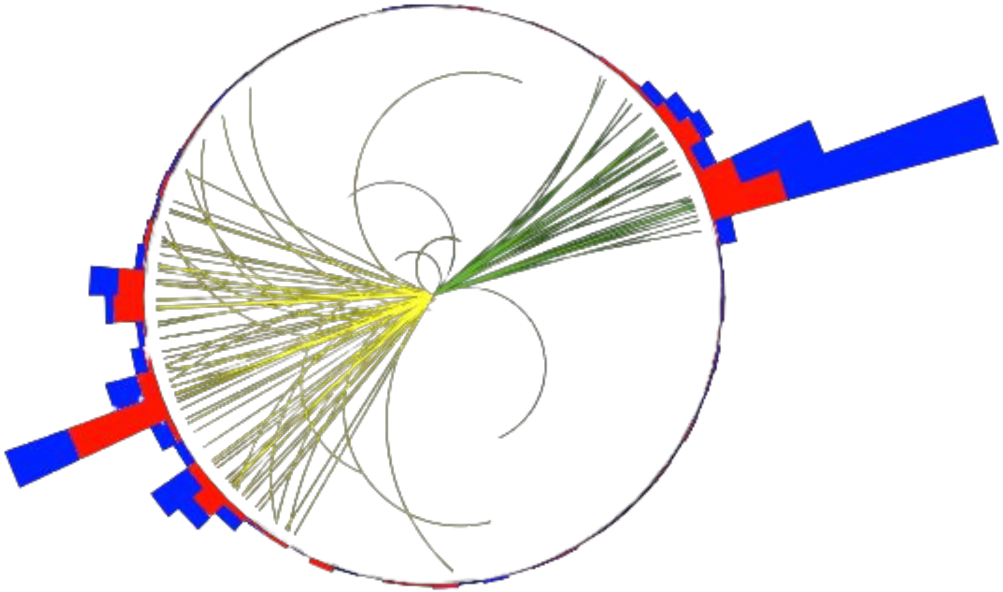
Rikab Gambhir

In collaboration with Jesse Thaler & Ben Nachman

ML4Jets, July 7 2021

# Outline

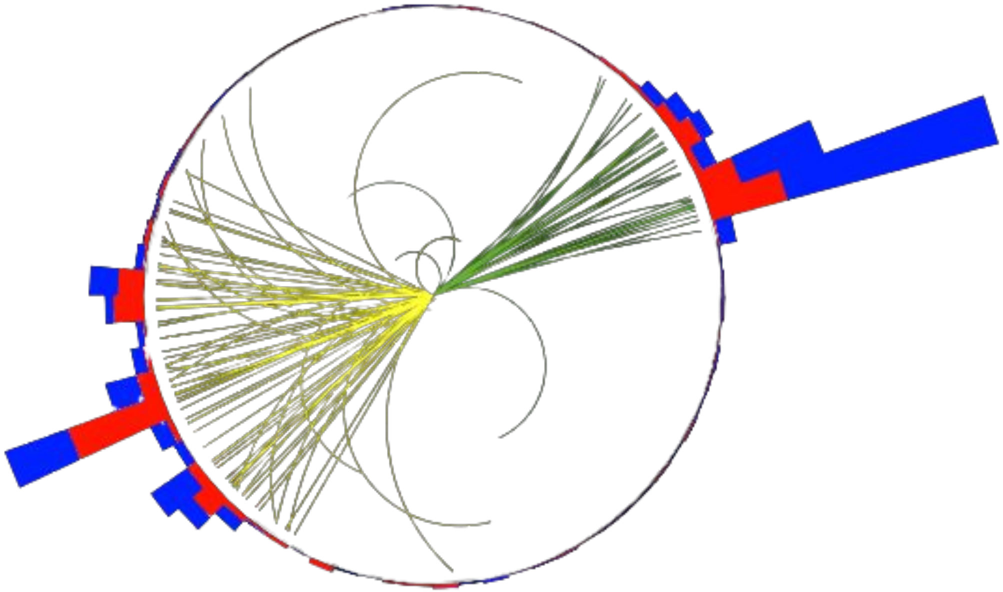
1. Motivation and Theory
   a. Mutual Information
   b. Frequentist Inference
2. Machine Learning Framework
   a. Machine Learning Algorithm
   b. Gaussian Ansatz
   c. Inference and uncertainties
3. Jet Energy Calibration



[CMS, 2004.08262]

# Outline

1. Motivation and Theory
    a. Mutual Information
    b. Frequentist Inference
2. Machine Learning Framework
    a. Machine Learning Algorithm
    b. Gaussian Ansatz
    c. Inference and **uncertainties**
3. Jet Energy Calibration



Learn frequentist uncertainties directly and in one training, and quantify correlations!

[CMS, 2004.08262]

# Motivation and Theory

# Problem Statement

Given data samples of two random variables, ***X*** and ***Y***, we can ask the following questions about them:

1. Given a sample $x$, can we predict $y$, *with uncertainties*?
2. Precisely how correlated are $X$ and $Y$?



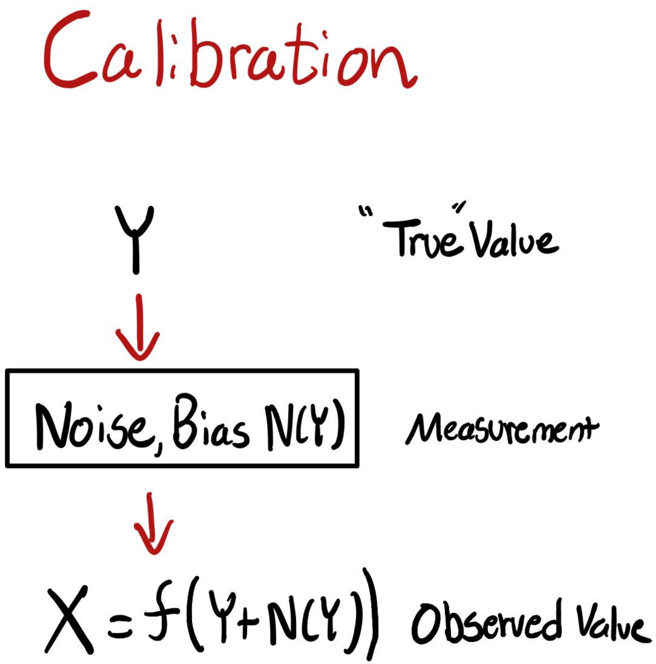
Rikab Gambhir

# Problem Statement

Given data samples of two random variables, ***X*** and ***Y***, we can ask the following questions about them:

1. Given a sample *x*, can we predict *y*, *with uncertainties*?
2. Precisely how correlated are *X* and *Y*?

In the context of calibration, we would like to do this in a **frequentist** way!



# Problem Statement

Given data samples of two random variables, ***X*** and ***Y***, we can ask the following questions about them:

1. Given a sample $x$, can we predict $y$, ***with uncertainties***? → **Frequentist inference**: Find and characterize $p(x|y)$
2. Precisely how correlated are $X$ and $Y$? → **Mutual Information**: Calculate $I(X;Y)$

# Problem Statement

Given data samples of two random variables, ***X*** and ***Y***, we can ask the following questions about them:

1. Given a sample $x$, can we predict $y$, ***with uncertainties***? → **Frequentist inference**: Find and characterize $p(x|y)$
2. Precisely how correlated are $X$ and $Y$? → **Mutual Information**: Calculate $I(X;Y)$

We can answer both questions at the same time, only looking at the data once!

# Problem Statement

Given data samples of two random variables, ***X*** and ***Y***, we can ask the following questions about them:

1. Given a sample $x$, can we predict $y$, ***with uncertainties***?
2. Precisely how correlated are $X$ and $Y$?

Rich existing literature!

Simulation based inference & Uncertainty Estimation:
[Cranmer, Brehmer, Louppe 1911.01429;
Alaa, van der Schaar 2006.13707;
Abdar et. al, 2011.06225;
Tagasovska, Lopez-Paz, 1811.00908;
And many more!]

Bayesian techniques:
[Jospit et. al, 2007.06823;
Wang, Yeung 1604.01662;
Izmailov et. al, 1907.07504;
Mitos, Mac Namee, 1912.1530;
And many more!]

We can answer both questions at the same time, only looking at the data once!

# Mutual information

A measure for non-linear interdependence is the **Mutual Information**:

$$I(X;Y) = \sum_{x\in\Omega_X, y\in\Omega_Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Answers the question: How much information, in terms of bits, do you learn about *Y* when you measure *X* (or vice versa)?

Can be written as the well-known **KL-Divergence**:

$$I(X;Y) = D_{KL}(P(x,y)\,||\,P(x)P(y))$$

where $D_{KL}(P||Q) = \int du\, P(u)\log\left(\frac{P(u)}{Q(u)}\right)$

# The Donsker-Varadhan Representation

We can write the KL-Divergence in the **Donsker-Varadhan Representation**:

$$I(X,Y) \geq -\inf_{T_\theta \in \mathcal{T}} \mathcal{L}[T_\theta]$$

$$\text{where } \mathcal{L}[T] = -\left\{\mathbb{E}_{p(x,y)}[T(x,y)] - \log\left(\mathbb{E}_{p(x)p(y)}\left[e^{T(x,y)}\right]\right)\right\}$$

If the class of parameterized functions $\mathcal{T}$ is expressive enough, the bound will be saturated.

**Goal**: Find the $T$ minimizing this loss functional!

*Other representations, such as those based on *f-divergences*, have also been tried, but suffer convergence issues

[Belghazi, Baratin, Rajeswar, Ozair, Bengio, Courville, Hjelm, 1801.04062; Le, Nguyen, Phung, 1711.01744 Nowozin; Cseke, 1606.00709]

# Estimating Likelihoods

The bound is saturated for the learned function (Well-known!):

$$T_\theta(x,y) = \log\left(\frac{p(x|y)}{p(x)}\right) + 1$$

This contains the **likelihood**! So we can perform maximum likelihood inference (Assuming the network is well trained!):

$$\hat{y}(x) = \underset{y \in \Omega_Y}{\operatorname{argmax}}\, T_\theta(x,y)$$

# Estimating Uncertainties

Standard Uncertainty contours given by:

$$\Gamma_{1\sigma}(x) = \{ y \mid T(x,y) = T(x, \hat{y}(x)) - 1/2 \}$$

# Estimating Uncertainties

Uncertainty contours given by:

$$\Gamma_{1\sigma}(x) = \left\{ y \mid T(x,y) = T(x, \hat{y}(x)) - \tfrac{1}{2} \right\}$$

Too hard! Settle for Gaussian error bars:

$$\mathrm{cov}_y(x) = -1 \left( \frac{d^2 T(x,y)}{dy_i \, dy_j} \right)^{-1} \Bigg|_{y = \hat{y}(x)}$$

# Estimating Uncertainties

Uncertainty contours given by:

$$\Gamma_{1\sigma}(x) = \{ y \mid T(x,y) = T(x, \hat{y}(x)) - 1/2 \}$$

Too hard! Settle for Gaussian error bars:

$$\mathrm{cov}_y(x) = -1\left( \frac{d^2 T(x,y)}{dy_i \, dy_j} \right)^{-1} \Bigg|_{y = \hat{y}(x)}$$

**Goal**: Extract this value (without any extra work)!

# Framework

# Maximum Likelihood

For a measurement *X*, what was the *Y* most likely to have produced it?

Inherently independent of the prior for *Y* - the **calibration** task

$$\mathrm{argmax}_{y \in \Omega_Y} \{ p(x|y) \} = \mathrm{argmax}_{y \in \Omega_Y} \{ T(x,y) \}$$

We can also extract Gaussian uncertainties given by

$$\mathrm{cov}_Y(x) = -1 \left( \frac{d^2 T(x,y)}{dy_i \, dy_j} \right)^{-1} \Bigg|_{y = \mathrm{argmax}_{y \in \Omega_Y} \{ T(x,y) \}}$$

Technically, we can calculate these from our trained network *T*. But finding maxima and derivatives* is extremely hard!

*If you use the ReLU activation function, all second derivatives are zero.

# The Gaussian Ansatz

Parameterize *T(x,y)* in the following way (the **Gaussian Ansatz**):

$$T(x,y) = A(x) + (y - B(x)) \cdot D(x) + \frac{1}{2}(y - B(x))^T \cdot C(x,y) \cdot (y - B(x))$$

where

- $A : \Omega_X \to \mathbb{R}$
- $B : \Omega_X \to \mathbb{R}^{\dim(\Omega_Y)}$
- $C : \Omega_X \times \Omega_Y \to \mathrm{Sym}(\mathbb{R}, \dim(\Omega_Y))$
- $D : \Omega_X \to \mathbb{R}^{\dim(\Omega_Y)}$

Are parameterized functions

# The Gaussian Ansatz

$$T(x,y) = A(x) + (y - B(x)) \cdot D(x) + \frac{1}{2}(y - B(x))^T \cdot C(x,y) \cdot (y - B(x))$$

This ansatz is fully expressive: any smooth function of *X* and *Y* can be written in this form! The networks *A, B, C,* and *D* are all learned functions.

If we take the limit *D→0* (forced during training), then we can see:

$$\hat{y}(x) = \underset{y \in \Omega_Y}{\operatorname{argmax}} \, T(x,y) = \underline{B(x)}$$

$$\sigma_{\hat{y}}(x) = -1 \left( \frac{d^2 T(x,y)}{dy_i \, dy_j} \right)^{-1} \Bigg|_{y=\hat{y}(x)} = \underline{-C^{-1}(x, B(x))}$$

The maximum likelihood solution for *Y* given *X*, plus its uncertainty, are manifest in the Gaussian Ansatz! No need for difficult maximization problems

# Example Calibration Problem

**Premise:** A noisy voltmeter

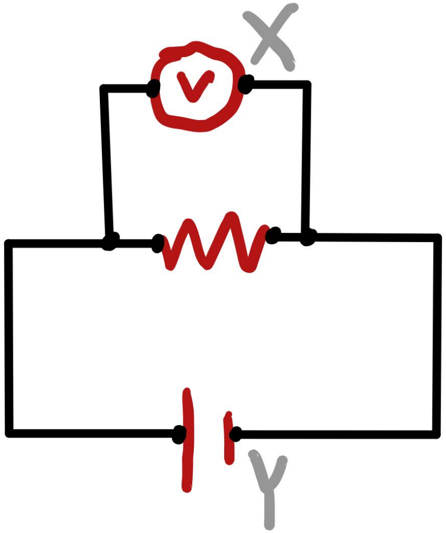
The "true" voltage $Y$ is a random number given by $P(y) = U(-5,5)$ *

The voltmeter adds Gaussian noise $N$ with a standard deviation of 1 Volt: Observe $X = Y + N$

Given the observation $X$, what was $Y$ and its uncertainty?

Expect to learn the likelihood $P(x|y) = Norm(y, 1)$

**Inherently frequentist!**




*Technically, I don't need to tell you P(x) or P(y) because of prior independence!

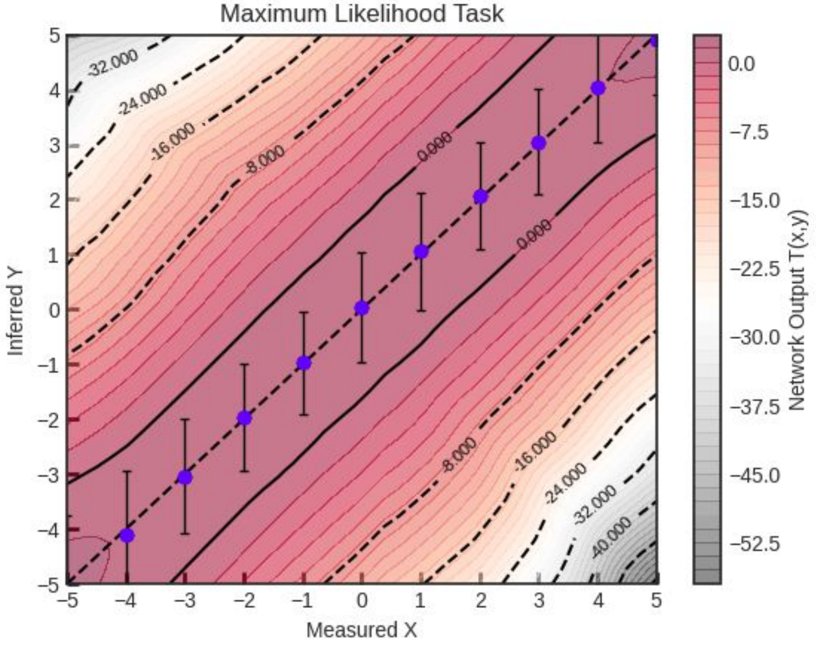
# Example Calibration Problem

**Model**:

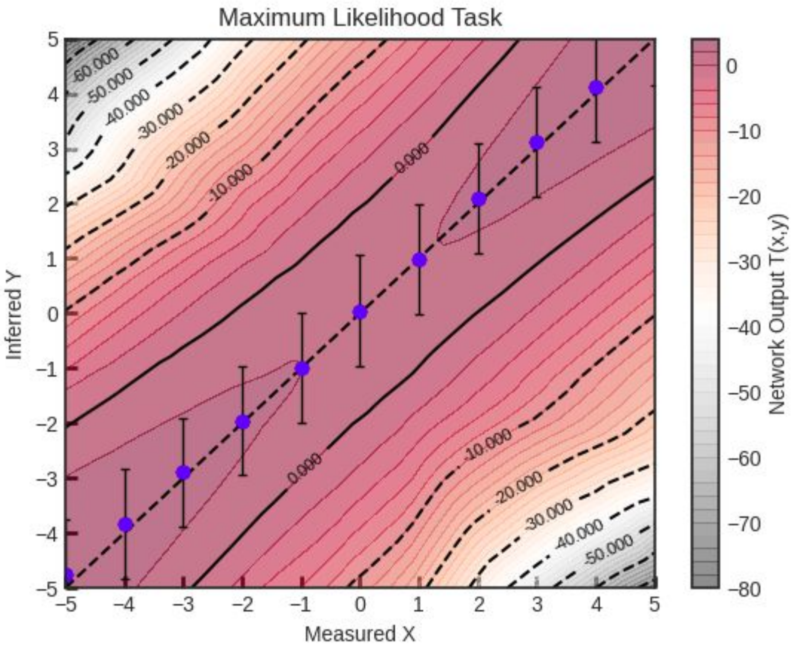
- The A, B, C, and D networks are each Dense networks with 4 layers of size 32
- ReLU activations
- All parameters have an L2 regularization ($\lambda$ = 1e-6)
- The D network output has an L1 regularization ($\lambda$ = 1e-4)

Learned mutual information of 1.05 natural bits

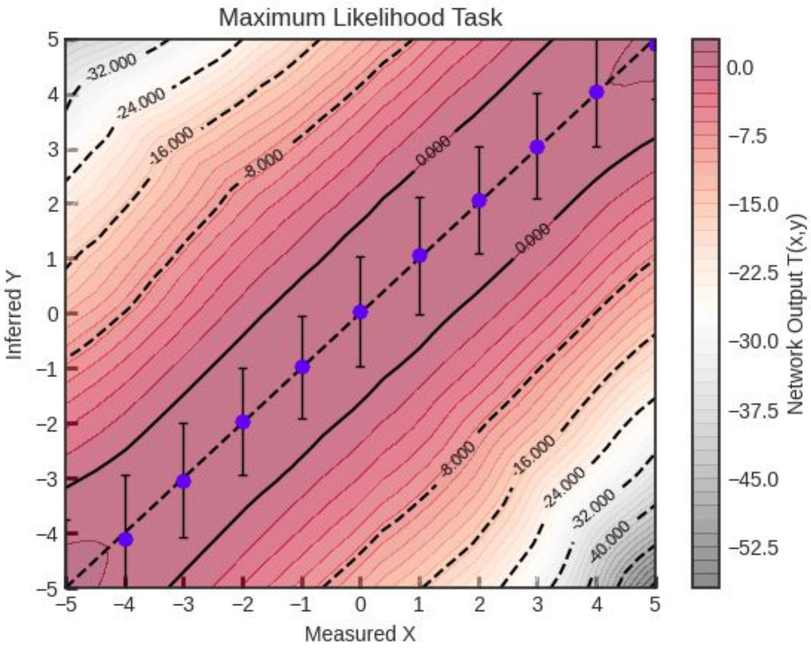
Reproduces the expected maximum likelihood outcome and the correct resolution!



# Example Calibration Problem - Prior Independence




**P(Y) = N(0, 2.5)**



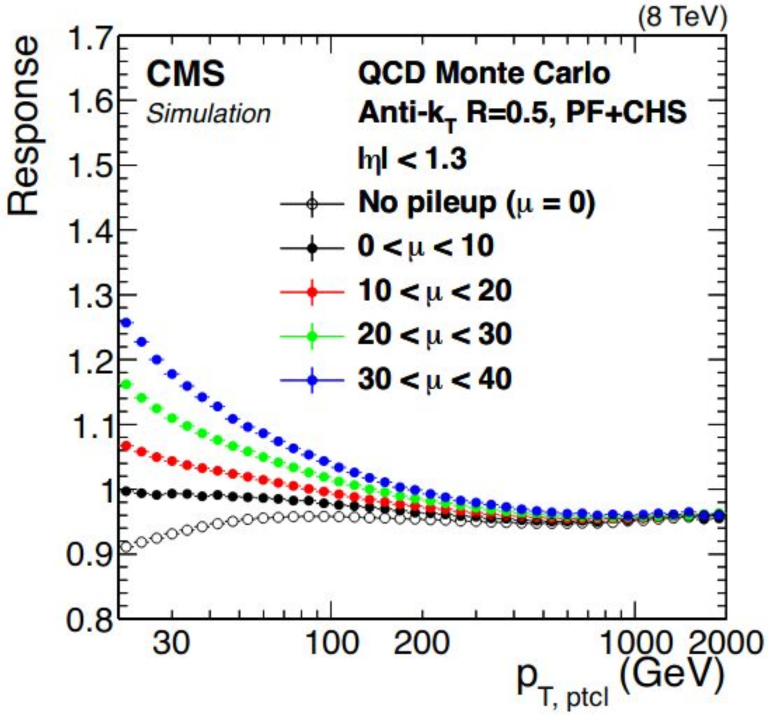

**P(Y) = U(-5, 5)**

# Jet Energy Calibrations

# Problem Statement

Measure a set particle flow candidates $X$ in the detector. What is the underlying jet $p_T$, $Y$, and its uncertainty?

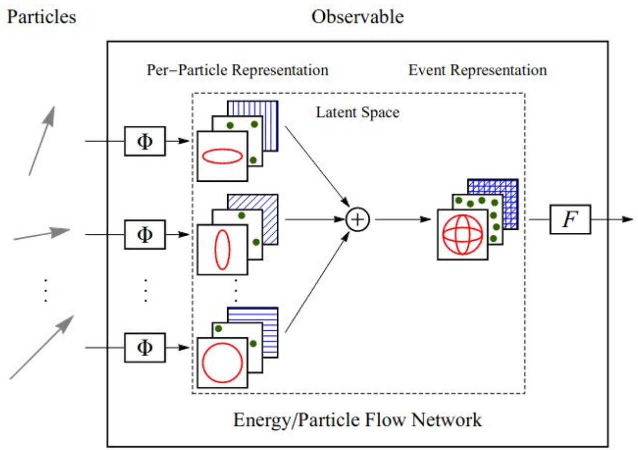
Define the **jet energy scale (JES)** and **jet energy resolution (JER)** as the ratio of the underlying jet $p_T$ (resolution) to the measured total jet $p_T$




[CMS, 1607.03663]

# Models

- **DNN**: $X$ = (Jet $p_T$, Jet $\eta$, Jet φ), Dense Neural Network
- **EFN**: $X$ = {(PFC $p_T$, PFC $\eta$, PFC φ)}, Energy Flow Network
- **PFN**: $X$ = {(PFC $p_T$, PFC $\eta$, PFC φ)}, Particle Flow Network
- **PFN-PID**: $X$ = {(PFC $p_T$, PFC $\eta$, PFC φ, PFC PID)}, Particle Flow Network




Permutation-invariant function of point clouds
For EFN's, manifest IRC Safety

All models use ReLU activations with the Adam optimizer (α = 1e-3). Model parameters have an L2 regularization (λ = 1e-6), and the *D* network output has an L1 regularization (λ = 1e-4)

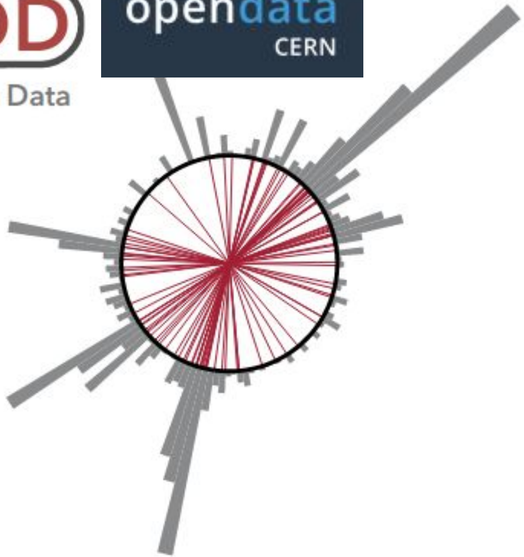
[Komise, Metodiev, Thaler, 1810.05165]

[Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 2020;
Larkoski, Marzani, Thaler, Tripathee, Xue, 1704.05066;
Cacciari, Salam, Soyez, 0802.1189;
http://opendata.cern.ch/]

# Jet Dataset

Using CMS Open Data:

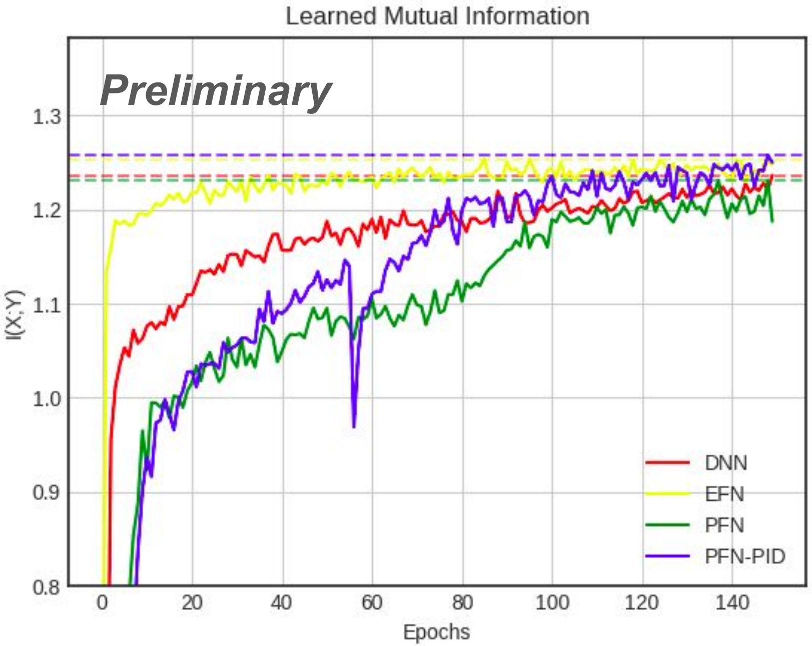
- *CMS2011AJets* Collection, SIM/GEN QCD Jets (AK 0.5)
- Select for jets with 500 GeV < Gen $p_T$ < 1000 GeV, $|\eta|$ < 2.4, quality ≥ 2
- Select for jets with ≤ 150 particles
- Jets are rotated such that jet axis is centered at (0,0)
- Train on 100k jets



# Mutual Information

| Model | I(X;Y) [Natural Bits] |
|---|---|
| DNN | 1.23 |
| EFN | 1.25 |
| PFN | 1.25 |
| PFN-PID | 1.27 |

*Preliminary*





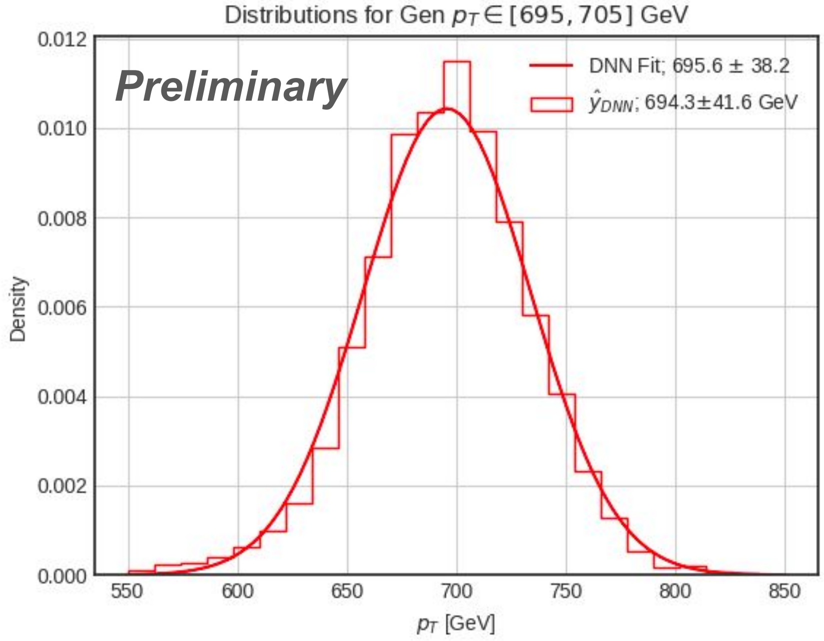

Rikab Gambhir

# Jet Energy Scales

For jets with a true $p_T$ of 700 GeV, we should expect well-trained models to predict 700 GeV on average!

| Model | Gaussian Fit [GeV] |
|---|---|
| DNN | 695 ± 38.2 |
| EFN | 692 ± 37.7 |
| PFN | 702 ± 37.4 |
| PFN-PID | 693 ± 35.9 |
| CMS Open Data | 695 ± 37.4 |

***Preliminary***



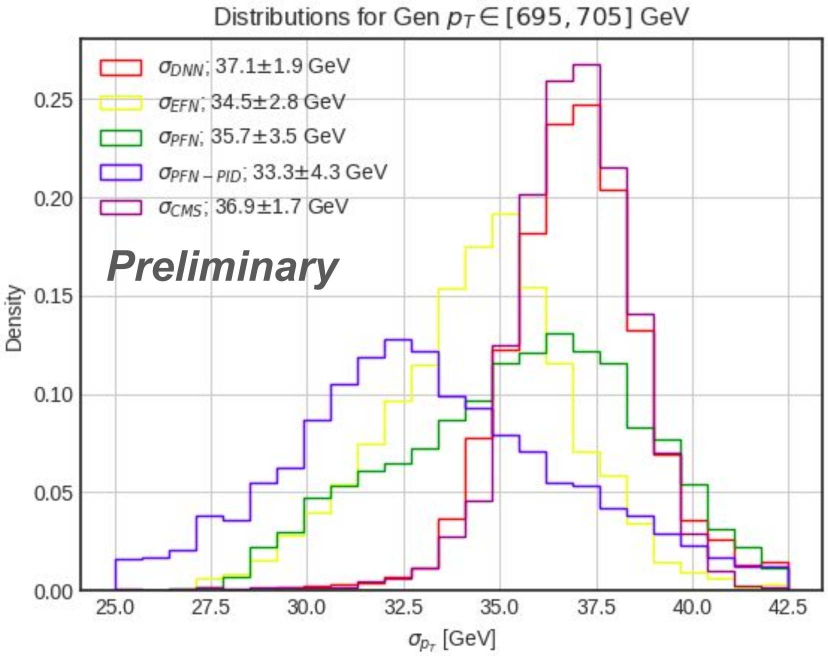

DNN $\hat{y}$ distribution for $y \in [695, 705]$ GeV

# Jet Energy Resolution

Predicted uncertainty distributions for the different models - The higher the learned mutual information, the better the resolution!

| Model | Avg Resolution [GeV] |
|---|---|
| DNN | 37.1 ± 1.9 |
| EFN | 34.5 ± 2.8 |
| PFN | 35.7 ± 3.5 |
| PFN-PID | 33.3 ± 4.3 |
| CMS Open Data | 36.9 ± 1.7 |

***Preliminary***



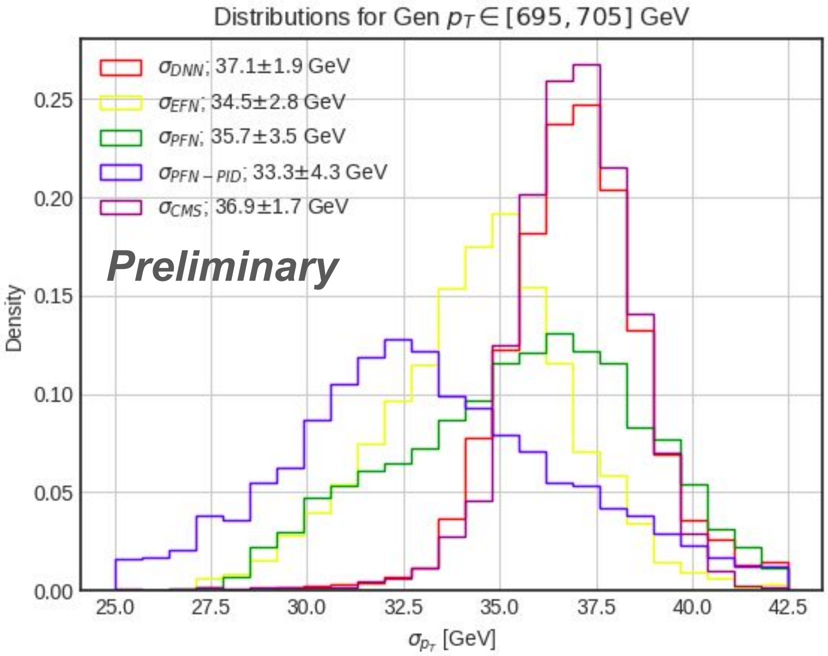

Uncertainty distribution for *y* ∈ *[695, 705]* GeV

# Conclusion

We have presented a framework useful for (all at the same time!):

- Estimating **mutual information**, a measure of the nonlinear interdependence between random variables
- Performing **frequentist** maximum likelihood inference for *Y* given *X*
- Estimating the **uncertainty** on *Y* for said inference
- Moreover, the Gaussian Ansatz makes the above manifest

Given nothing but example *(x,y)* pairs, in a single training. All of these tasks are useful in high energy physics, such as for jet energy calibration!

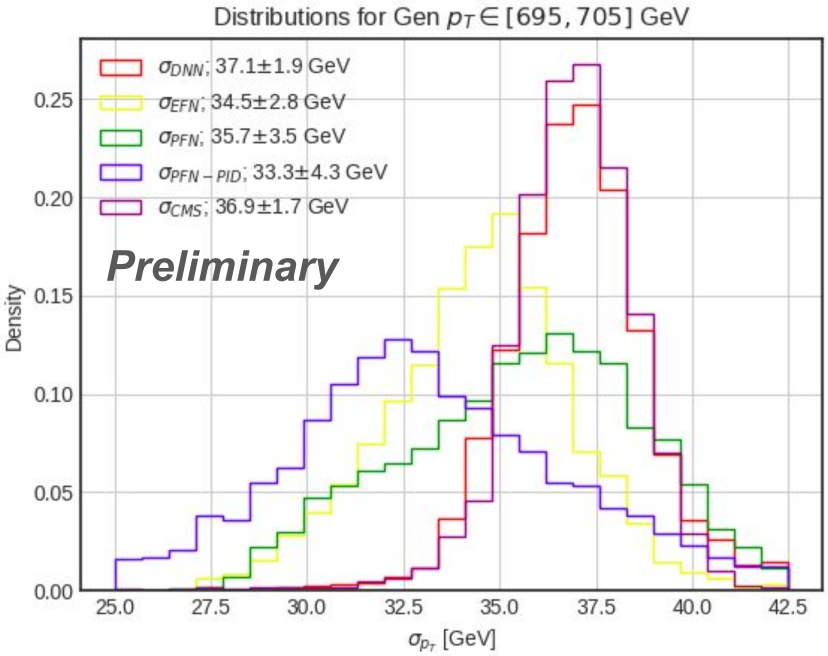


# Conclusion

We have presented a framework useful for (all at the same time!):

- Estimating **mutual information**, a measure of the nonlinear interdependence between random variables
- Performing **frequentist** maximum likelihood inference for $Y$ given $X$
- Estimating the **uncertainty** on $Y$ for said inference
- Moreover, the Gaussian Ansatz makes the above manifest

Given nothing but example *(x,y)* pairs, in a single training. All of these tasks are useful in high energy physics, such as for jet energy calibration!




**Thank you!**

# Appendices

# Algorithm

Initialize a parameterized function $T_\theta(x,y)$

1. Draw $b$ batch samples from $P(X,Y)$: $\{(x_1,y_1) \ldots (x_b,y_b)\}$
2. Draw $b$ batch samples from $P(Y)$: $\{y_1', \ldots y_b'\}$
3. Compute the loss $L(\{\theta\}) = -1/b \sum[T_\theta(x,y)] + \log(\sum[e^{T\theta(x,y')}])$
4. Update weights $\theta' = \theta - \nabla_\theta L(\{\theta\})$ (or use your favorite optimizer!)

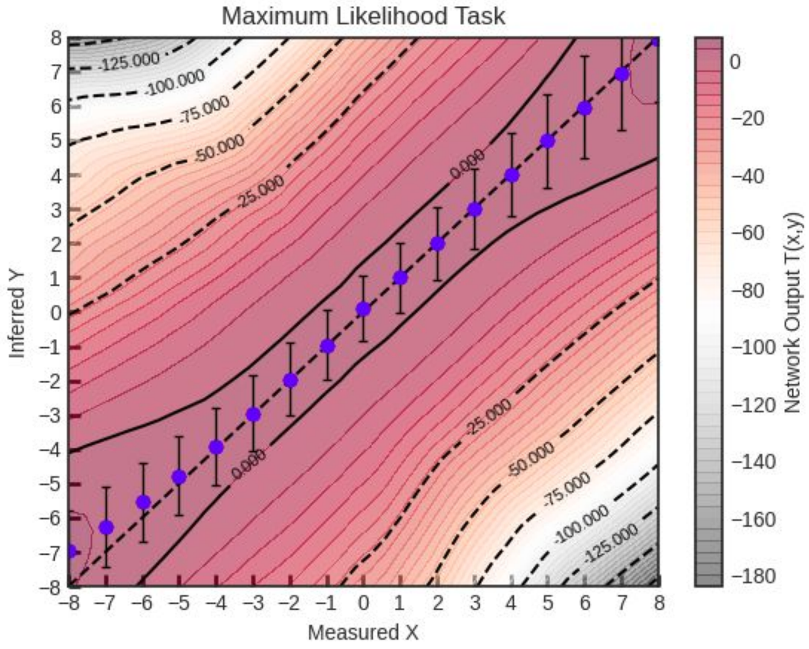
When converged, $-L$ will be a lower bound for $I(X;Y)$, and $T$ will contain the likelihood

# Outside Uniform Prior

Prior is still U(-5, 5), extrapolate anyways

Same maximum likelihood result

Larger errors due to limited statistics



# Ensembles

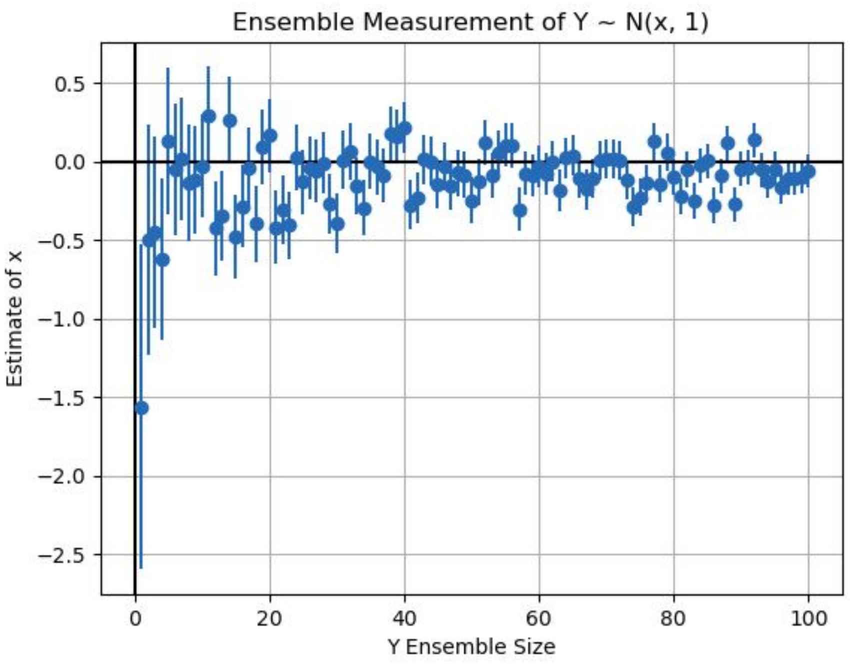
Once we have a procedure for estimating the maximum likelihood *Y* for a measured *X*, can extend to estimating a model parameter $\theta$ given *N* data *I.I.D.* points $X_i$ - the **unfolding** problem.

If $\frac{d}{dy} C(x,y)$ is small:

$$\hat{\Theta}(\{x\}) = \left[ \sum_i C(x_i, B(x_i)) \right]^{-1} \sum_i C(x_i, B(x_i))$$

$$\mathrm{cov}_{\hat{\Theta}}(\{x\}) = \left[ \sum_i C(x_i, B(x_i)) \right]^{-1}$$

Could potentially use this to *directly* estimate Lagrangian parameters from data!



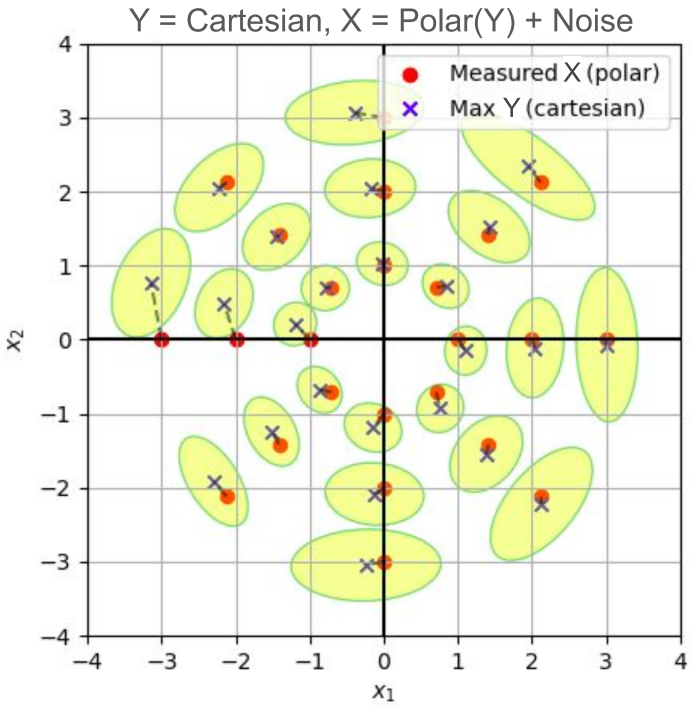
# Multi Dimensional Test

Polar Coordinates Conversion

- $Y = \text{Uniform}( (-4,-4) , (-4, 4)$
- $X = (r, \varphi) + (N(0,0.25), N(0,\pi/12))$

$\varphi$ is in the coordinate patch $(-\pi, \pi)$

Explains the weirdness near $\pi$



# Convergence Test

Simple X = Y + Gaussian Noise example

10 trials

- Red: DV Loss
- Green: F-Divergence Loss
- Yellow: F-Divergence + regularization

Whenever the green or yellow blow up (more accurately, blow down), set the MI to 0.0 because that is the best bound.