Learning Uncertainties the Frequentist Way

Calibration and Correlation in High Energy Physics

Rikab Gambhir

With Jesse Thaler and Benjamin Nachman

Learning Uncertainties the Frequentist Way

Calibration and Correlation in High Energy Physics

Rikab Gambhir

With Jesse Thaler and Benjamin Nachman

Based on work in:

[**RG**, Nachman, Thaler, PRL 129 (2022) 082001]
[**RG**, Nachman, Thaler, PRD 106 (2022) 036011]

Download our repo!

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]

Problem

I saw this …

Detector hits *x*

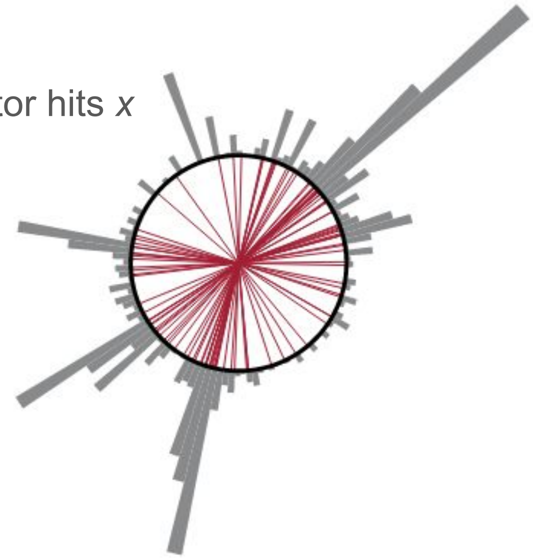


Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]

Problem

I saw this … … but I *want* this …

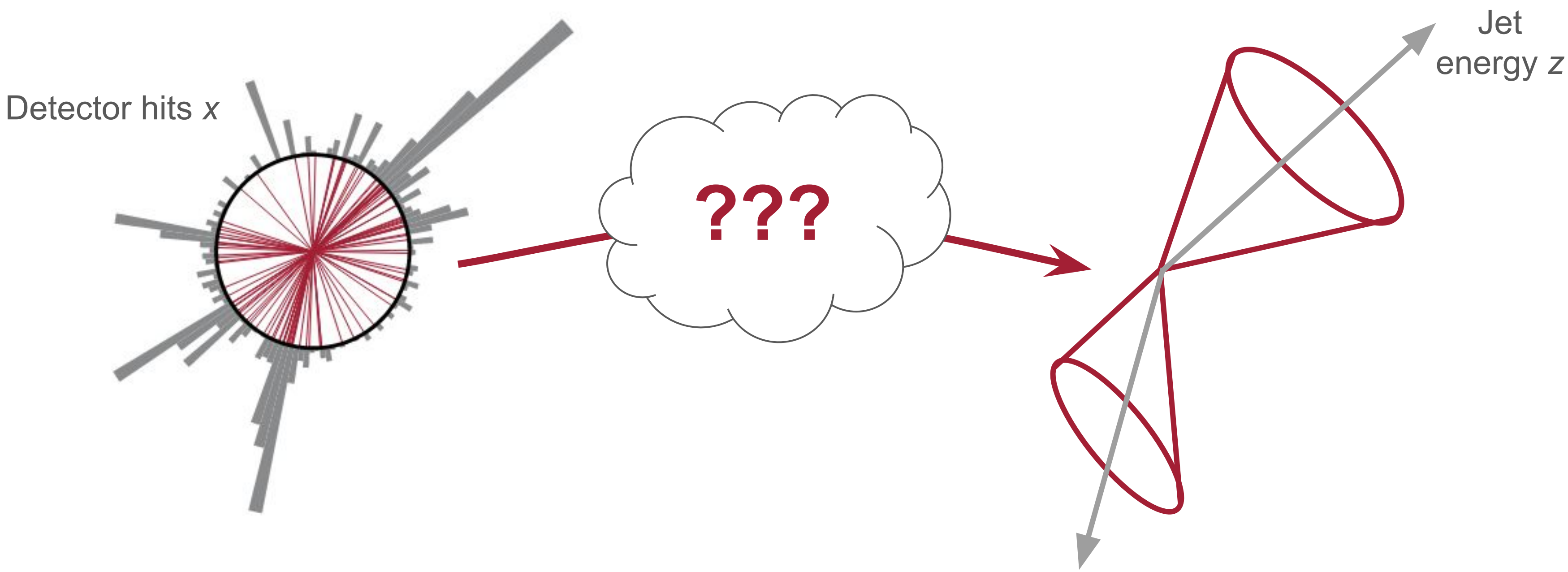
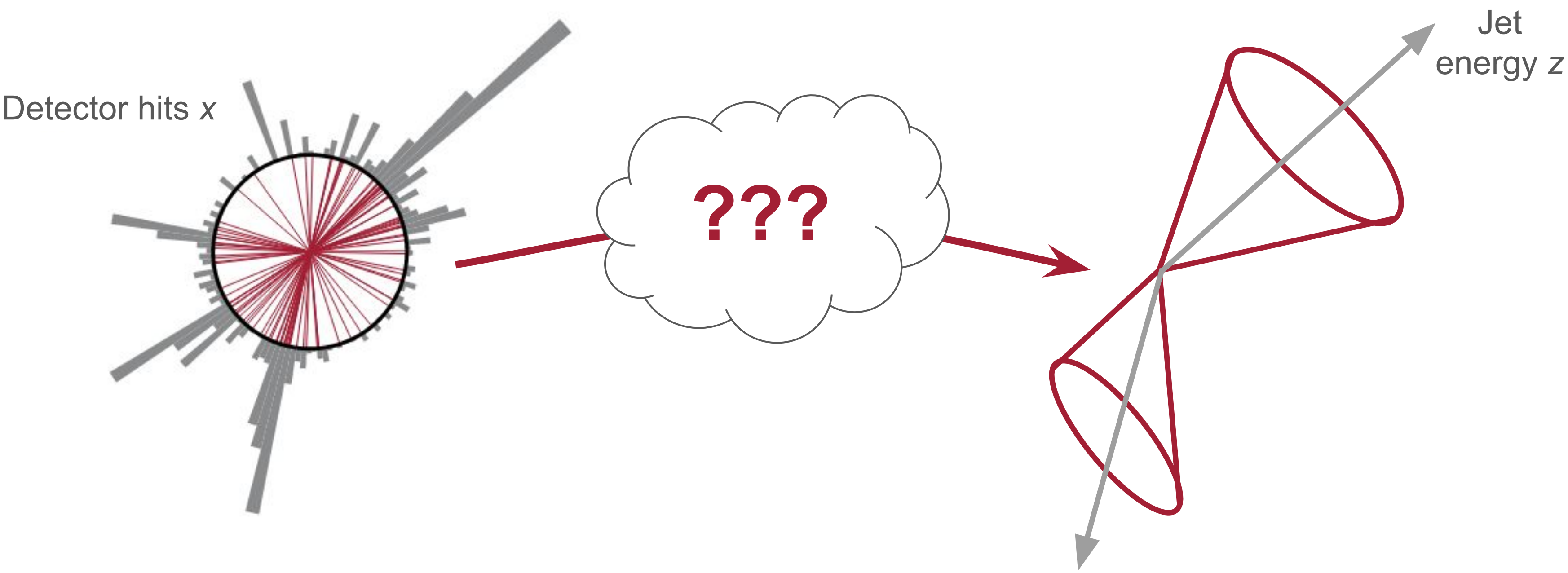


Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]

Problem

I saw this … … but I *want* this …

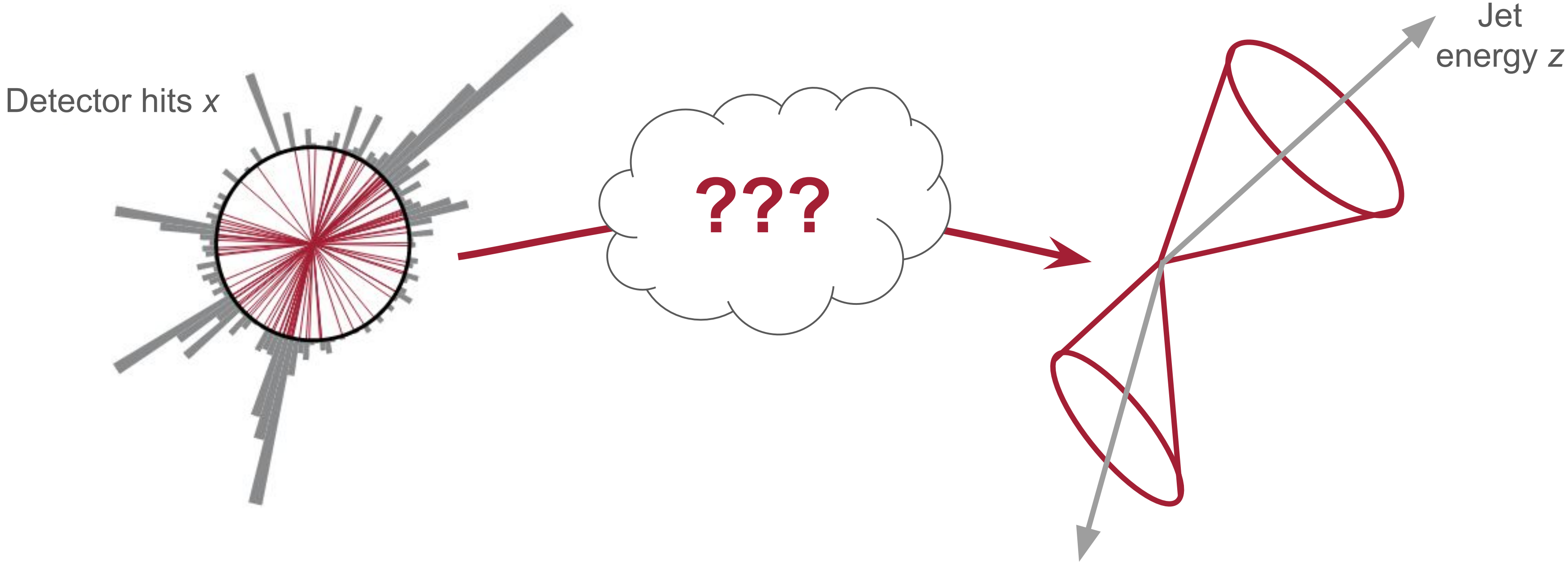


… with **uncertainties** …

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]

Problem

I saw this … … but I *want* this …



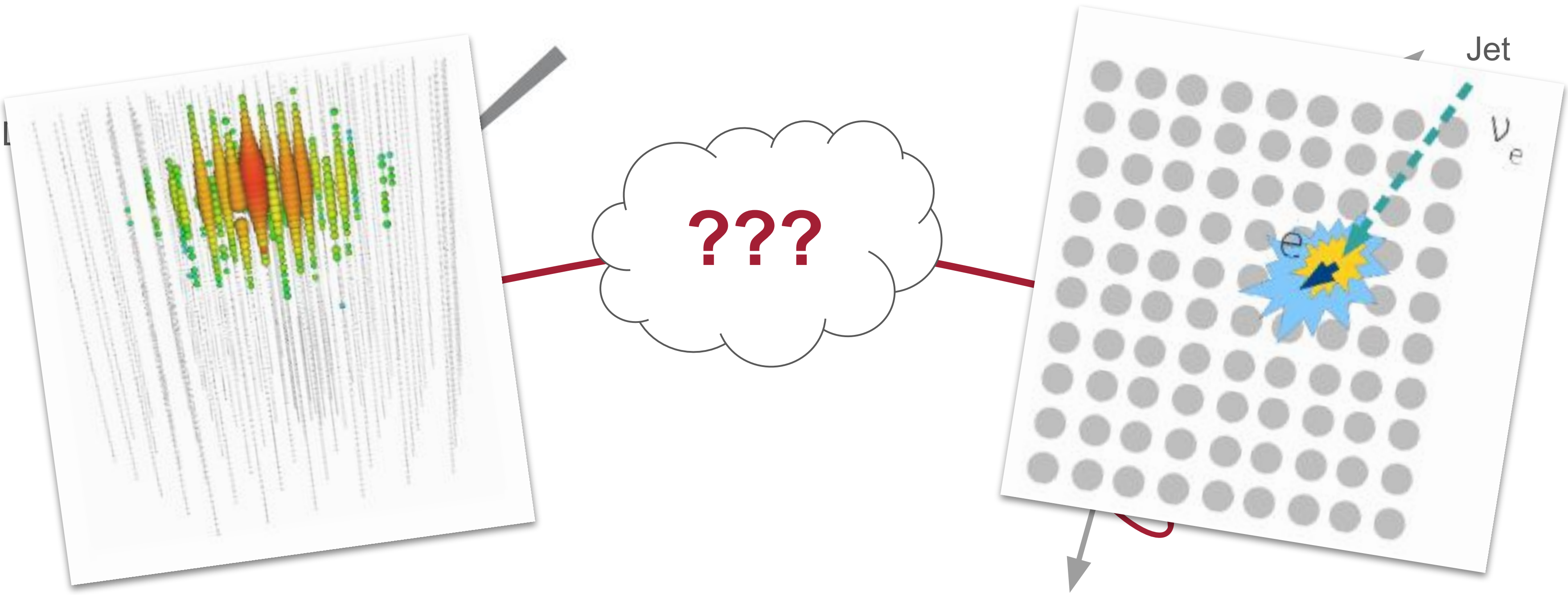
… with **uncertainties** …

… regardless of which event sample I use!

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]
[Rongen, 1911.02016]

Problem - Ubiquitous!

I saw this … … but I *want* this …



… with **uncertainties** …

… regardless of which event sample I use!

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]
[Rongen, 1911.02016]
[Arjona Martínez, Cerri, Spiropulu, Vlimant, Pierini, Eur. Phys. J. Plus 134, 333 (2019)]

Problem - Ubiquitous!

I saw this …

… but I *want* this …

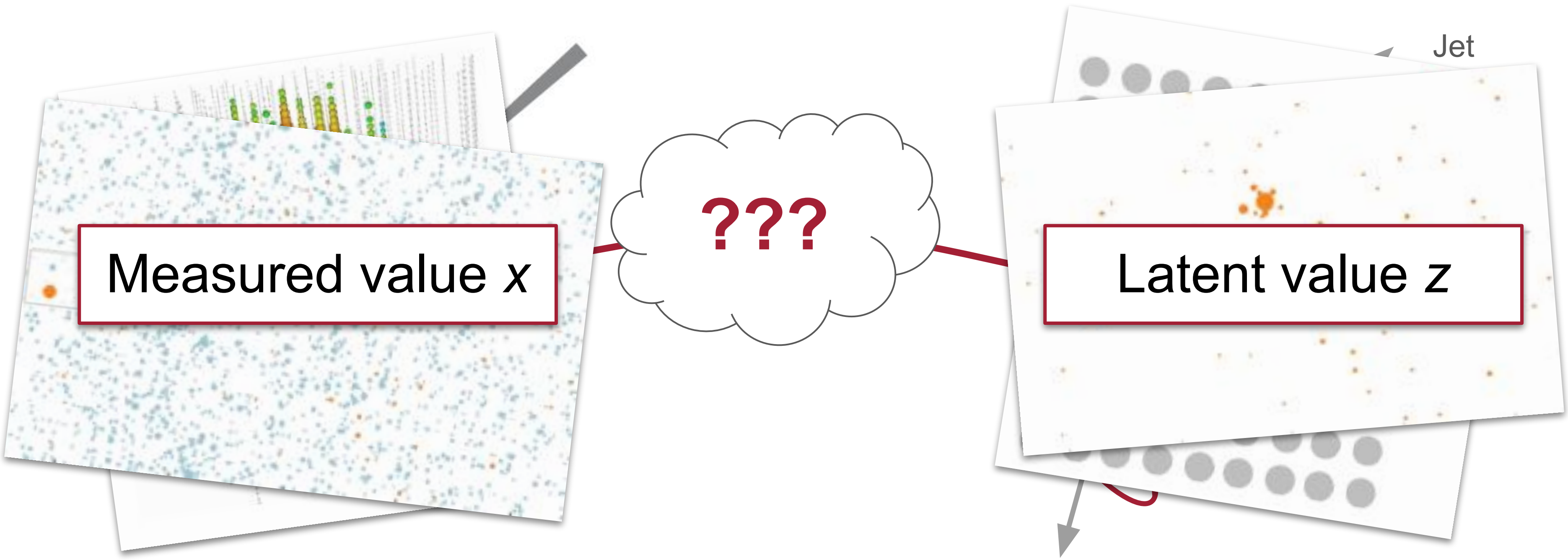
… with **uncertainties** …

… regardless of which event sample I use!

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]
[Rongen, 1911.02016]
[Arjona Martínez, Cerri, Spiropulu, Vlimant, Pierini, Eur. Phys. J. Plus 134, 333 (2019)]

Problem - Ubiquitous!

I saw this … … but I *want* this …



… with **uncertainties** …

… regardless of which event sample I use!

Image Credit: [Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 101 (2020) 034009]
[Rongen, 1911.02016]
[Arjona Martínez, Cerri, Spiropulu, Vlimant, Pierini, Eur. Phys. J. Plus 134, 333 (2019)]

Problem - Ubiquitous!

I saw this … … but I *want* this …

Jet
nergy *z*

Mea z

es …

Solution

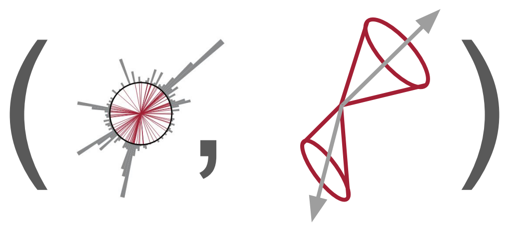
Choose a **Gaussian Ansatz** …

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

.. and a **special loss (DVR)** …

$$\mathcal{L}_{\mathrm{DVR}}[T] = -(\mathbb{E}_{P_{xz}}[T] - \log(\mathbb{E}_{P_x \otimes P_z}[e^T]))$$

Train on a sample of (x,z) pairs …



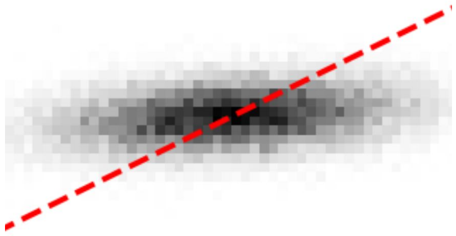
Then the **MLE inference** of *z* given x, with **uncertainties**, is …

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -[C(x, B(x))]^{-1}$$

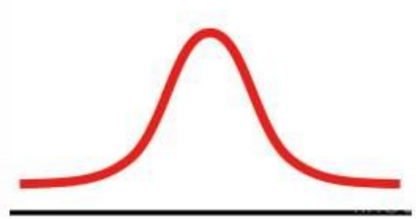
[**RG**, Nachman, Thaler, PRL 129 (2022) 082001]

… regardless of which event sample I use!

Outline



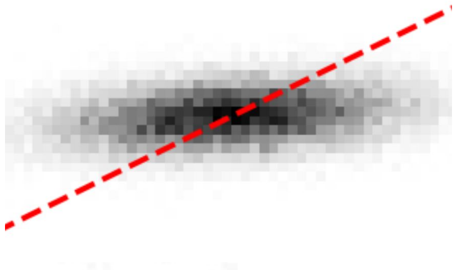
Calibration and Correlation



The Gaussian Ansatz

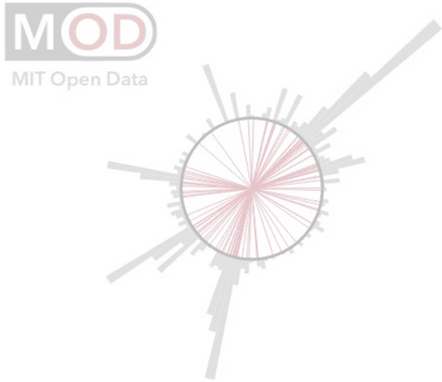
MOD
MIT Open Data

Empirical Studies



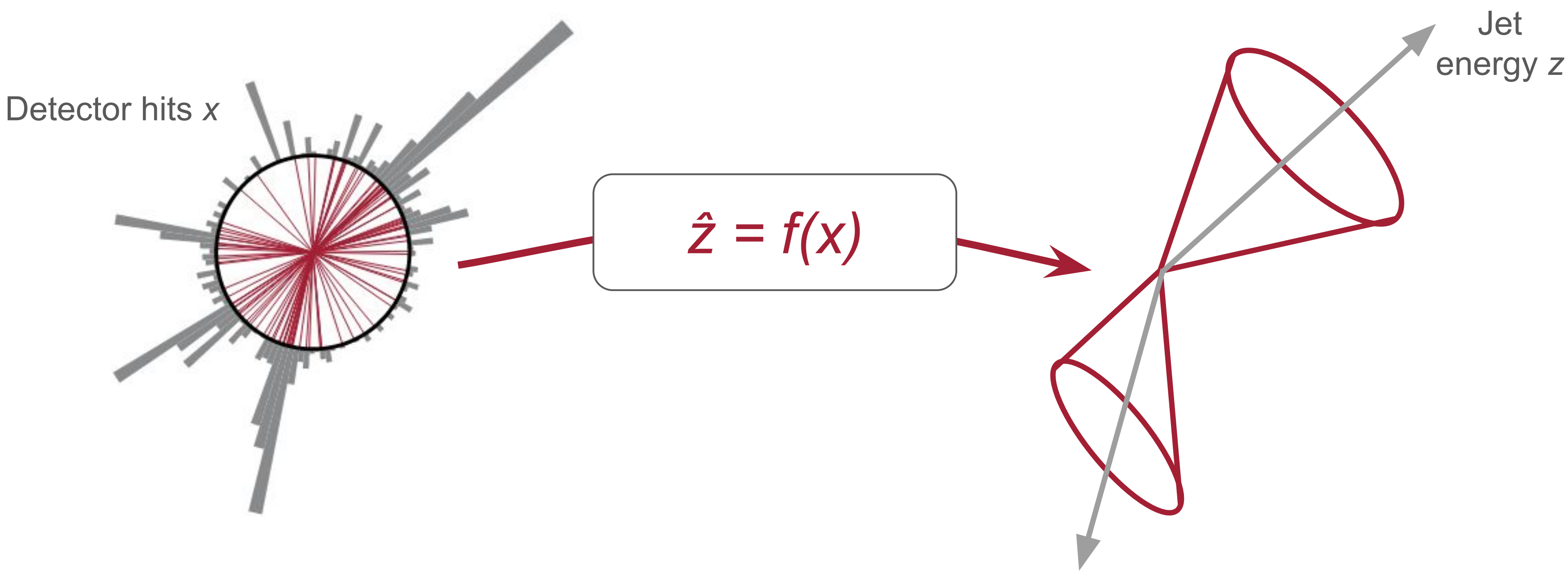
Calibration and Correlation

The Gaussian Ansatz



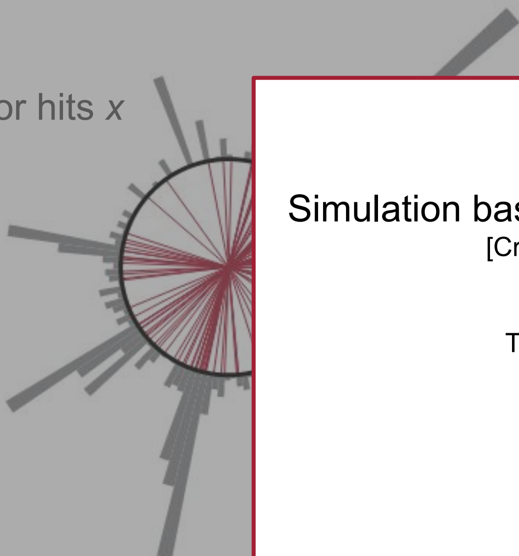
Empirical Studies

Calibration



Given a training set of (x,z) pairs, can we find an f such that $f(x)$ estimates z?

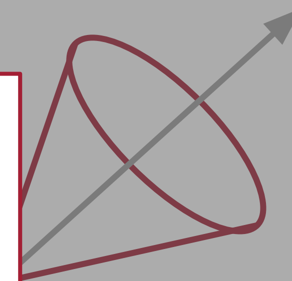
Calibration



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!]



Given a training set of (x,z) pairs, can we find an f such that $f(x)$ estimates z?

Calibration

Our function *f* should satisfy some key properties to be a calibration

1. **Closure**: On average, *f(x)* should be correct for each *x*! That is, *f* is **unbiased**.

2. **Universality**: *f(x)* should not depend on the choice of sampling for *z*. That is, *f* is **prior-independent**.

Calibration

Our function *f* should satisfy some key properties befitting a calibration

1. **Closure**: On average, *f(x)* should be correct for each *x*! That is, *f* is **unbiased**.

$$b(z) = \mathbb{E}_{\text{test}}[f(X) - z | Z = z] \\ = 0$$

2. **Universality**: *f(x)* should not depend on the choice of sampling for *z*. That is, *f* is **prior-independent**.

$$f \text{ depends only on } p(x|z)\text{, and not } p(z)$$

Likelihood: Detector simulation, noise model, etc

What if the detector simulation is imperfect? Ask me later!

Finding *f*: MSE?

Naive guess: *f* should minimize the mean squared error: $\underset{g}{\operatorname{argmin}}\, \mathbb{E}_{\text{train}}[(g(X) - Z)^2]$

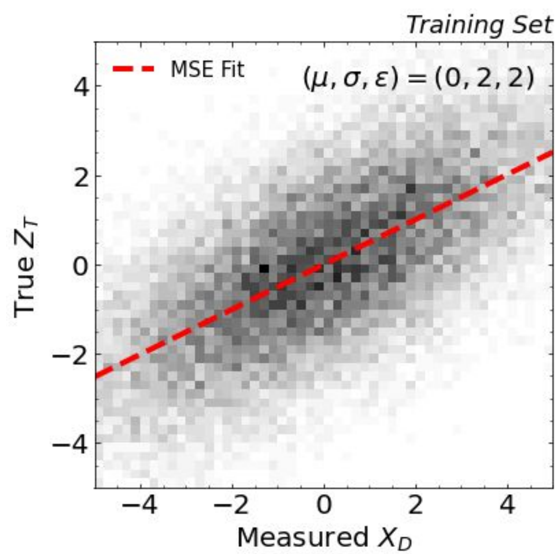
Intuitively, our guess $\hat{z}$ given x is the average of all *z* in the training set in the *x* bin.

Finding *f*: MSE?

Naive guess: *f* should minimize the mean squared error: $\underset{g}{\operatorname{argmin}}\, \mathbb{E}_{\text{train}}[(g(X) - Z)^2]$

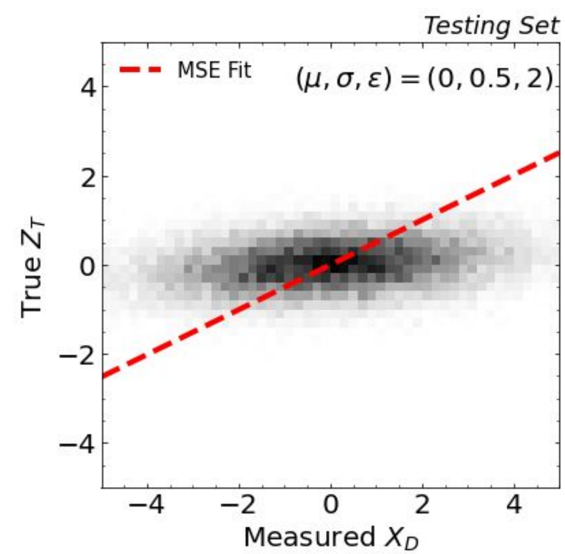
NOPE!

Intuitively, our guess $\hat{z}$ given x is the average of all *z* in the training set in the *x* bin.



Same "detector" sim *p(x|z)*, only different priors *p(z)*!

We can't apply our calibration universally.



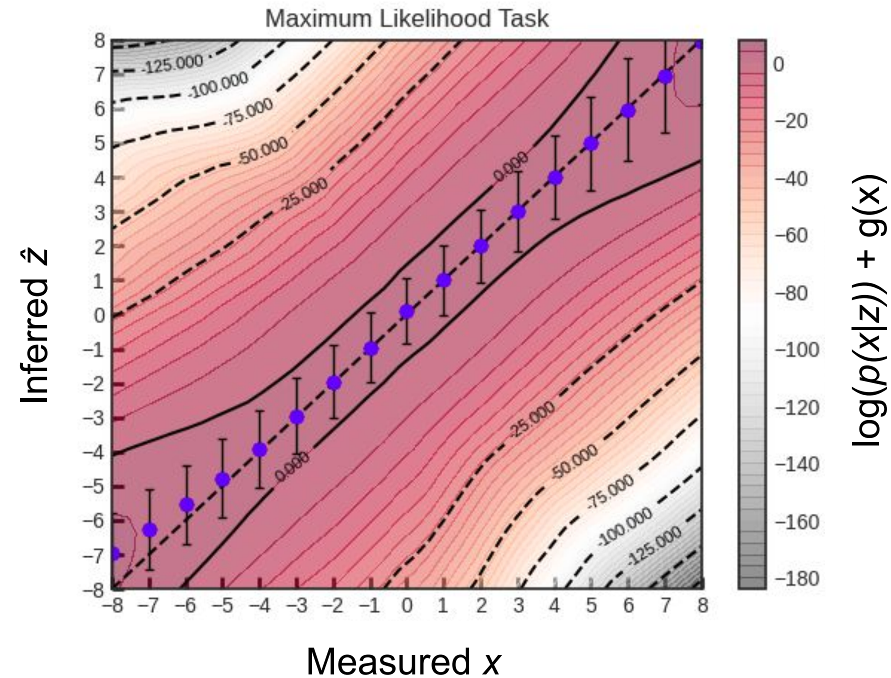
Can show analytically that f_{MSE} is both biased and non-universal, and biased even if the test prior is the same as training

Maximum Likelihood Calibration (MLC)

Instead:

$$f_{\mathrm{MLC}}(x) = \underset{z}{\operatorname{argmax}}\, p_{\mathrm{train}}(x|z)$$

"What z was most likely to have produced my x?
Prior independent by construction!



Can even quantify the uncertainty on $\hat{z}$: Contours of z that were also likely to produce x

Learning MLC

How do we calculate *f*?

$$f_{\text{MLC}}(x) = \underset{z}{\operatorname{argmax}}\, p_{\text{train}}(x|z)$$
$$= \underset{z}{\operatorname{argmax}} \underbrace{\log \frac{p_{\text{train}}(x,z)}{p_{\text{train}}(x)p_{\text{train}}(z)}}_{T(x,z)}$$

The function *T* is the likelihood ratio between *p(x,z)* and *p(x)p(z)*.

↓ Neyman–Pearson

T is the optimal classifier between (*x*,*z*) pairs and shuffled (*x*,*z*) pairs!

Learning MLC

How do we calculate *f*?

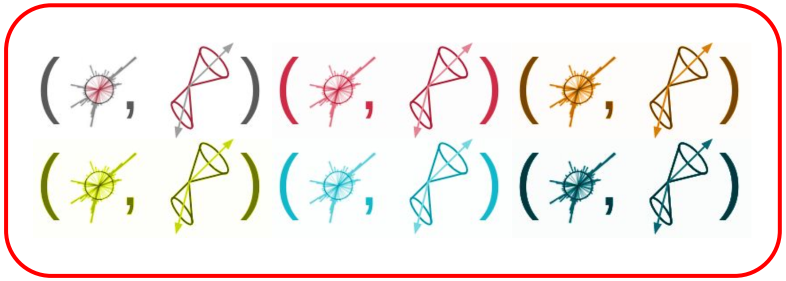
$$f_{\mathrm{MLC}}(x) = \operatorname*{argmax}_z p_{\mathrm{train}}(x|z)$$
$$= \operatorname*{argmax}_z \underbrace{\log \frac{p_{\mathrm{train}}(x,z)}{p_{\mathrm{train}}(x)p_{\mathrm{train}}(z)}}_{T(x,z)}$$

The function *T* is the likelihood ratio between *p(x,z)* and *p(x)p(z)*.

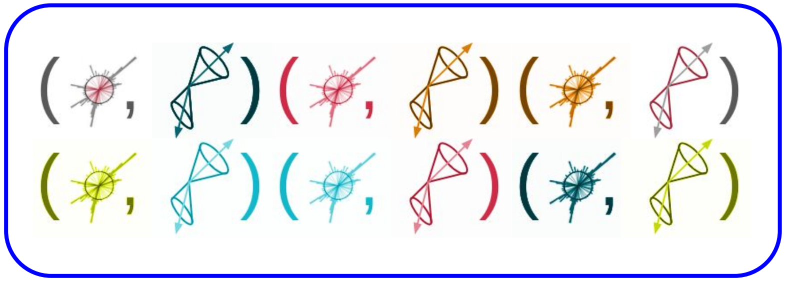
↓ Neyman–Pearson

T is the **optimal classifier** between (*x*,*z*) pairs and shuffled (*x*,*z*) pairs!

Class P



Class Q



Classify between ***P*** and ***Q***!

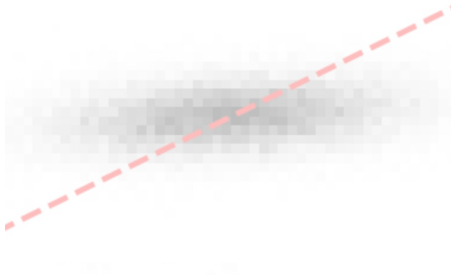
Aside: Mutual information

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

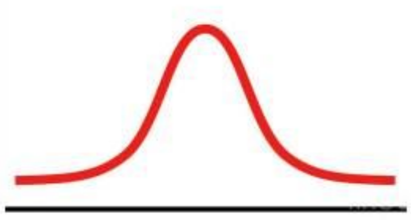
$$I(X;Z) = \int \mathrm{d}x\,\mathrm{d}z\,p(x,z)\log\frac{p(x,z)}{p(x)\,p(z)}$$
$$= \mathbb{E}_{\mathrm{train}} T(X,Z)$$

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

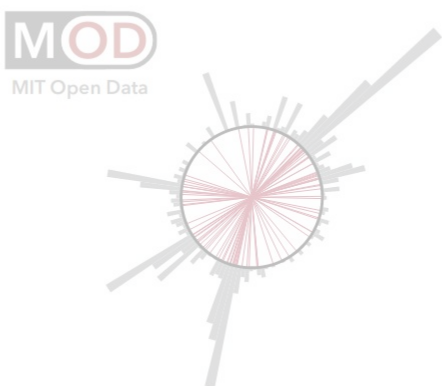
When doing calibration this way, we get a measure of the **correlation** between *X* and *Z*, *for free*.



Calibration and Correlation



The Gaussian Ansatz



Empirical Studies

Learning *T*

The **Donsker-Varadhan Representation (DVR)** of the KL divergence has been used in the statistics literature for mutual information estimation

$$\mathcal{L}_{\mathrm{DVR}}[T] = -\left(\mathbb{E}_{P_{XZ}}[T] - \log\left(\mathbb{E}_{P_X \otimes P_Z}\left[e^T\right]\right)\right)$$

Strict bound on *I(X;Z)*

Minimized when $$T(x,z) = \log\frac{p(x|z)}{p(x)} + c$$

Lots of other losses also work, but DVR has very nice convergence properties - ask me later!

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

Learning *T*

The **Donsker-Varadhan Representation (DVR)** of the KL divergence has been used in the statistics literature for mutual information estimation

$$\mathcal{L}_{\mathrm{DVR}}[T] = -\left(\mathbb{E}_{P_{XZ}}[T] - \log\left(\mathbb{E}_{P_X \otimes P_Z}\left[e^T\right]\right)\right)$$

Interestingly, a nonlocal loss!

Strict bound on *I(X;Z)*

What we want!

Minimized when $$T(x,z) = \log \frac{p(x|z)}{p(x)} + c$$

Unimportant

Lots of other losses also work, but DVR has very nice convergence properties - ask me later!

Inference using *T*

We can use a neural net to parameterize *T(x,z)*, and use standard gradient descent techniques to minimize the DVR loss. Then we can do …

$$\hat{z}(x) = \underset{z}{\operatorname{argmax}}\, T(x,z)$$

Inference

$$\left[\hat{\sigma}_z^2(x)\right]_{ij} = -\left[\frac{\partial^2 T(x,z)}{\partial z_i\, \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}$$

Gaussian Uncertainty Estimation

Inference using *T*

We can use a neural net to parameterize *T(x,z)*, and use standard gradient descent techniques to minimize the DVR loss. Then we can do …

$$\hat{z}(x) = \underset{z}{\operatorname{argmax}}\, T(x, z)$$

Inference

$$\left[\hat{\sigma}_z^2(x)\right]_{ij} = -\left[\frac{\partial^2 T(x, z)}{\partial z_i\, \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}$$

Gaussian Uncertainty Estimation

BUT!

- Maxima are hard to estimate – even *more* gradient descent?
- Second derivatives are sensitive to the choice of activations in *T* – ReLU spoils everything!

Inference using *T*

We can use a neural net to parameterize *T(x,z)*, and use standard gradient descent techniques to minimize the DVR loss. Then we can do …

$$\hat{z}(x) = \underset{z}{\operatorname{argmax}}\, T(x, z)$$

Inference

$$\left[\hat{\sigma}_z^2(x)\right]_{ij} = -\left[\frac{\partial^2 T(x,z)}{\partial z_i\, \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}$$

Gaussian Uncertainty Estimation

BUT!

- Maxima are hard to estimate – even *more* gradient descent!
- Second derivatives are sensitive to the choice of activations in *T* – ReLU spoils everything!

We solve both problems with the **Gaussian Ansatz**

The Gaussian Ansatz

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

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

Where *A(x), B(x), C(x,z), and D(x)* are learned functions. Then, if *D→0*, our inference and uncertainties are given by …

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -\big[C(x, B(x))\big]^{-1}$$

The Gaussian Ansatz

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

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

Where *A(x), B(x), C(x,z), and D(x)* are learned functions. Then, if *D→0*, our inference and uncertainties are given by …

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -\big[C(x, B(x))\big]^{-1}$$

No additional postprocessing or numerical estimates required!

The Gaussian Ansatz

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

Universal function approximator - any function that admit a taylor expansion in *z* around some *B(x)* can be written this way!

If there exists maxima $z = B^*$ anywhere, we can freely choose $D = 0$ by expanding around these maxima

Every smooth probability distribution looks like a Gaussian near the maximum!

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -\left[C(x, B(x))\right]^{-1}$$

There are a lot of additional subtleties to this training procedure - ask me about them!
[**RG**, Nachman, Thaler, PRL 129 (2022) 082001]

Algorithm

TensorFlow Implementation→

1. Initialize the *A(x), B(x), C(x,y),* and *D(x)*. Initialize the parameter $\lambda_D = 0$
2. On a batch of *(x,z)* pairs, compute the loss:

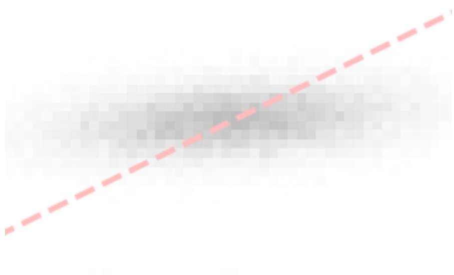
$$\mathcal{L}_{\rm DVR}[T] = -\Big(\mathbb{E}_{P_{XZ}}[T] - \log\left(\mathbb{E}_{P_X \otimes P_Z}\left[e^T\right]\right)\Big) + \lambda_D \mathbb{E}_{P_{XZ}}|D(X)|$$

The marginal distribution can be estimated by shuffling *z*'s between *(x,z)* pairs

3. Perform a gradient update on *A(x), B(x), C(x,y),* and *D(x)*. Increase λ_D.
4. Repeat 2-3 until *D* is everywhere 0 and the loss has converged.

Then, the loss is an estimate of the mutual information *I(X;Z)*, and *B* and *C* can be used to compute

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -\left[C(x, B(x))\right]^{-1}$$



Calibration and Correlation

The Gaussian Ansatz

MOD
MIT Open Data

Empirical Studies

Example 1: Gaussian Calibration Problem

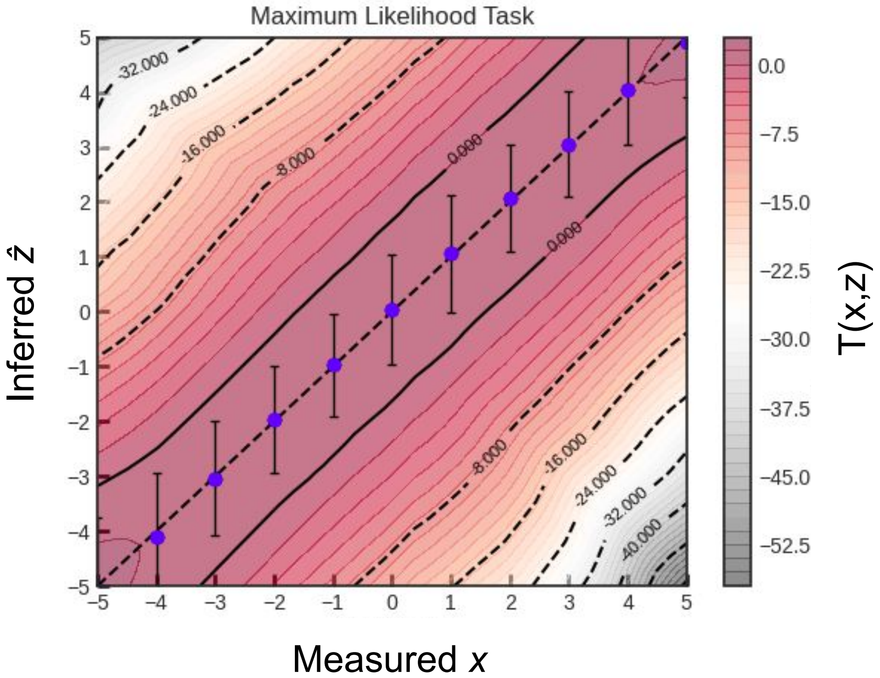
Gaussian noise model: p(x|z) ~ N(z, 1)

Model:

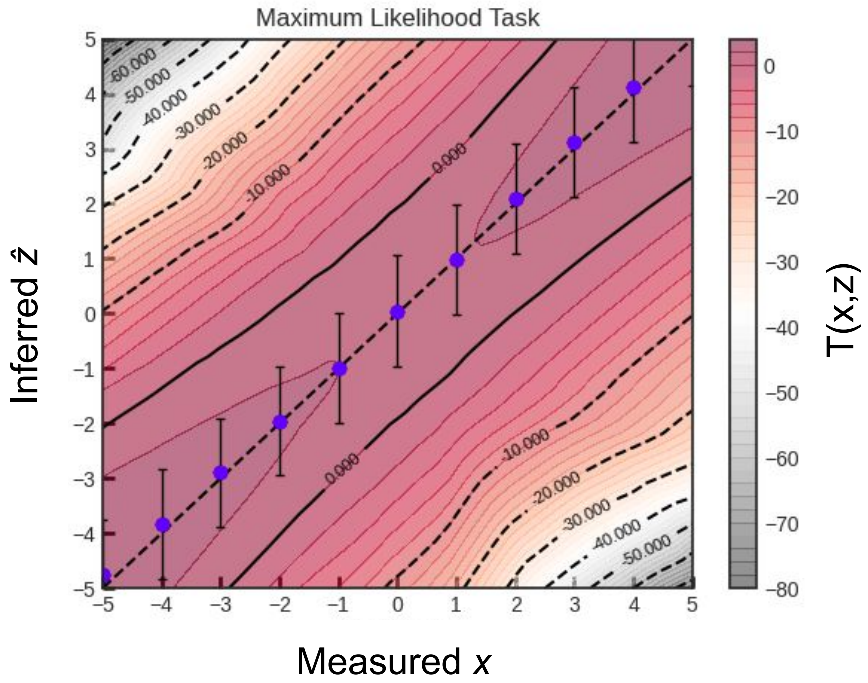
- The *A*, *B*, *C*, and *D* networks are each Dense networks with 4 layers of size 32
- ReLU activations
- All parameters have an L2 regularization (λ = 1e-6)
- The D network regularization slowly increased to (λ_D = 1e-4)

Learned mutual information of 1.05 natural bits

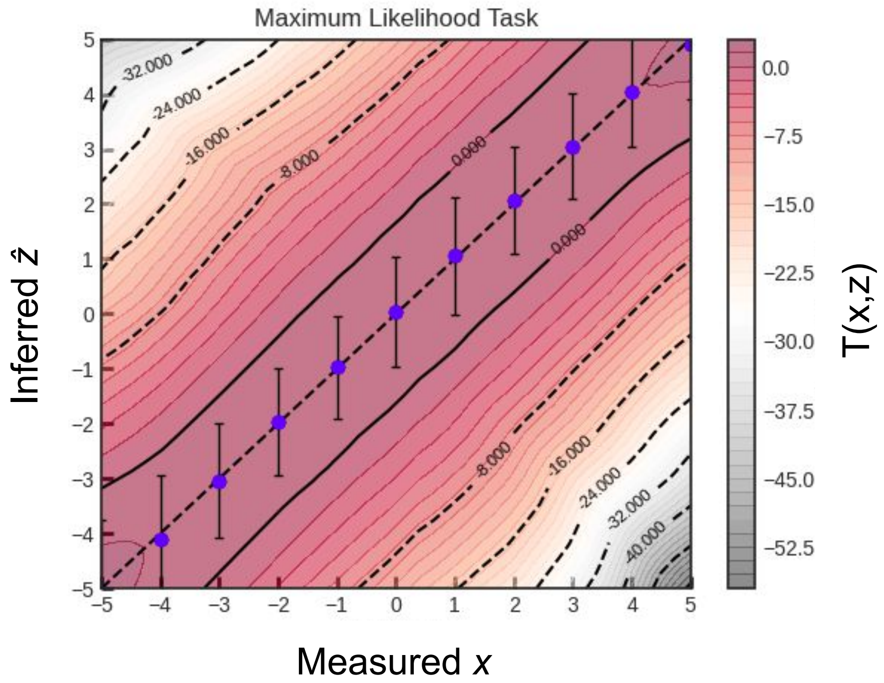
Reproduces the expected maximum likelihood outcome and the correct resolution!



Example 1 - Prior Independence



P(z) ~ N(0, 2.5)



P(z) ~ U(-5, 5)

[**RG**, Nachman, Thaler, PRD 106 (2022) 036011]

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

Example 2: QCD and BSM Dijets

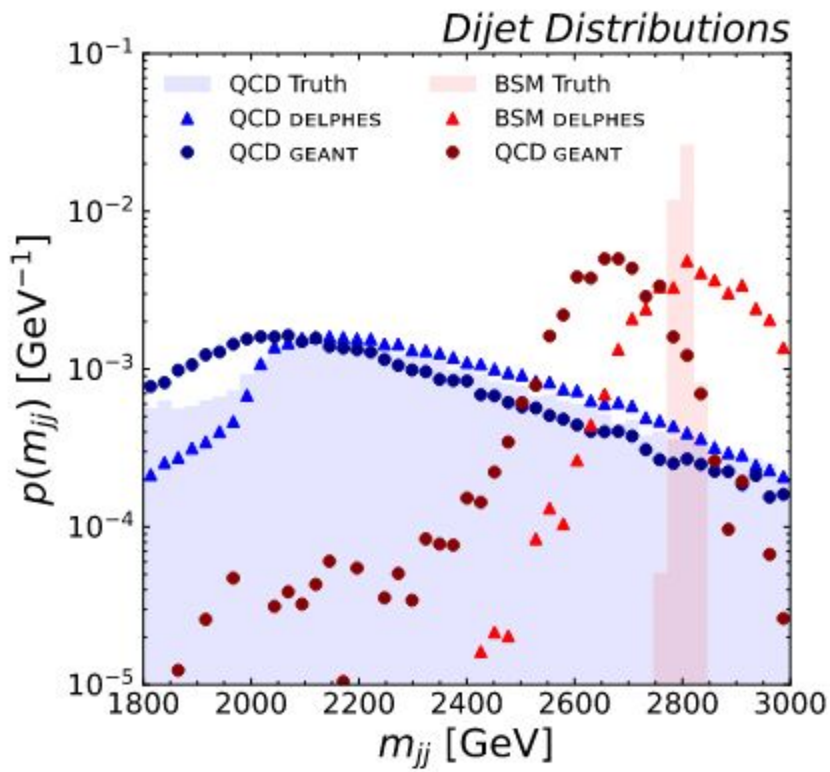
From CMS Open Data, a PYTHIA 6 sample of QCD dijet events:

- AK5 jets, hard p_T > 1 TeV, Z2 tune
- GEANT4 detector simulation

Want to infer the "true" $z = m_{jj}$ from the "reco" $x = m_{jj}$.

Two priors:

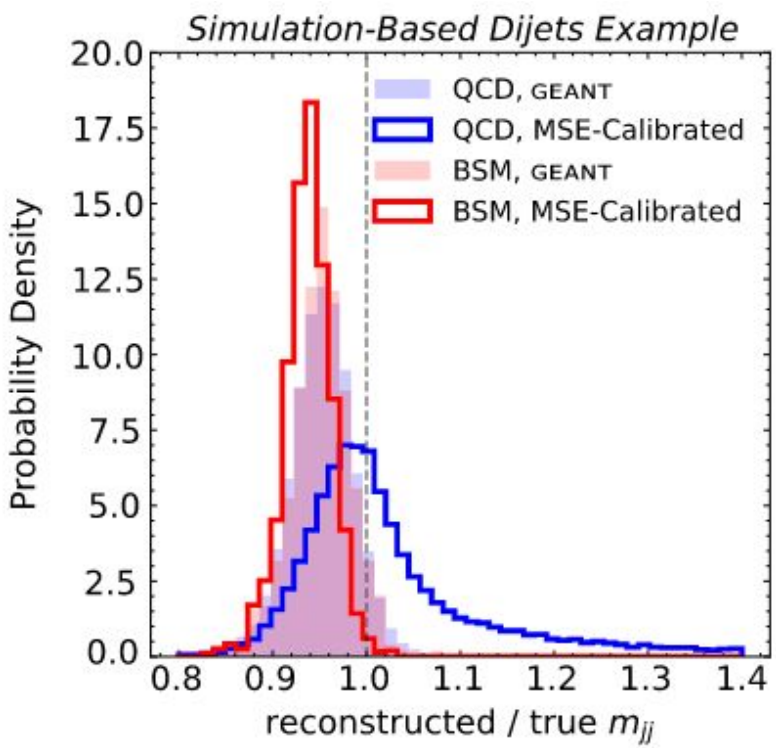
- **QCD**: Unaltered PYTHIA events
- **BSM**: Same events, reweighted such that *p(z)* is a sharp resonance



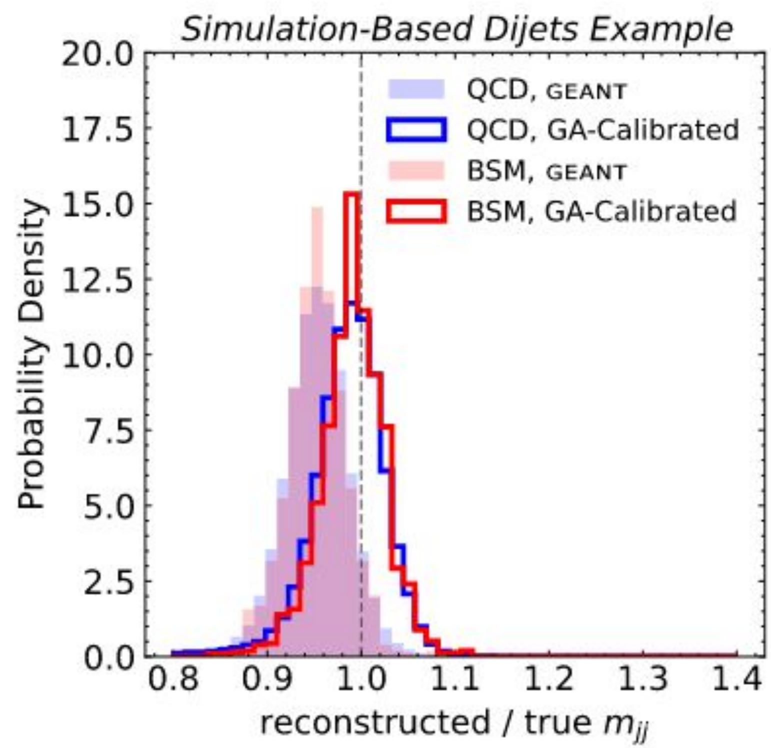
The DELPHES curves are related to a separate study about Data-Based Calibration. Ask me about it!

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

Example 2: QCD and BSM Dijets



(Left) MSE-fitted network.



(Right) Gaussian Ansatz-fitted network

Jet Energy Calibrations

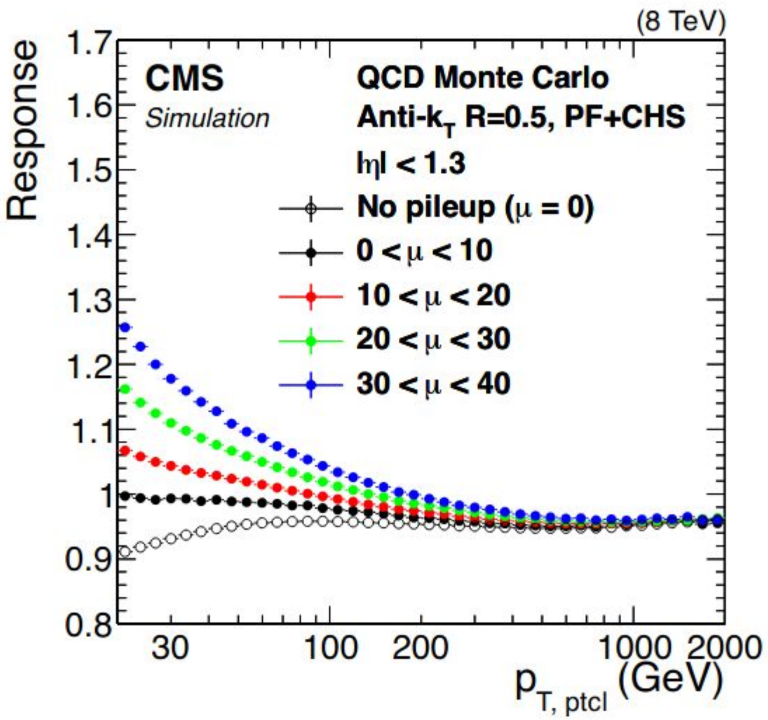
Example 3: Jet Energy Calibrations

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

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

$$\hat{p}_T \equiv \text{JEC} \times p_{T,\text{SIM}} \approx p_{T,\text{GEN}}$$

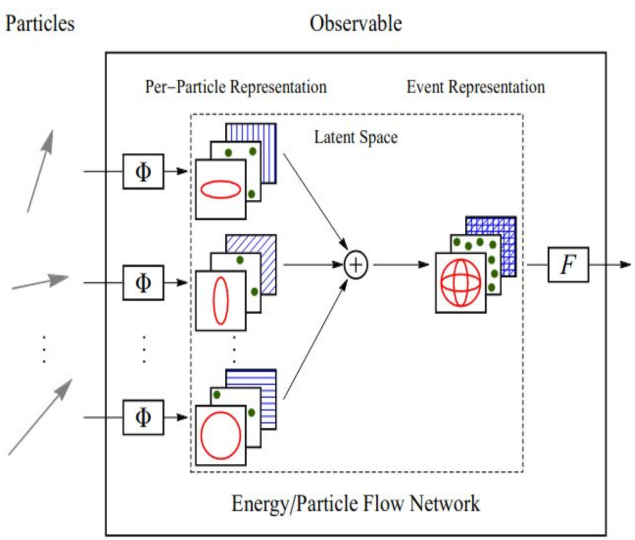
$$\hat{\sigma}_{p_T} = \text{JER} \times p_{T,\text{SIM}}$$



Example 3: Models

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

For each model, *A(x)*, *B(x)*, *C(x,z)*, and *D(x)* are all of the same type.



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

Details on hyperparameters can be found in [**RG**, Nachman, Thaler, PRL 129 (2022) 082001]

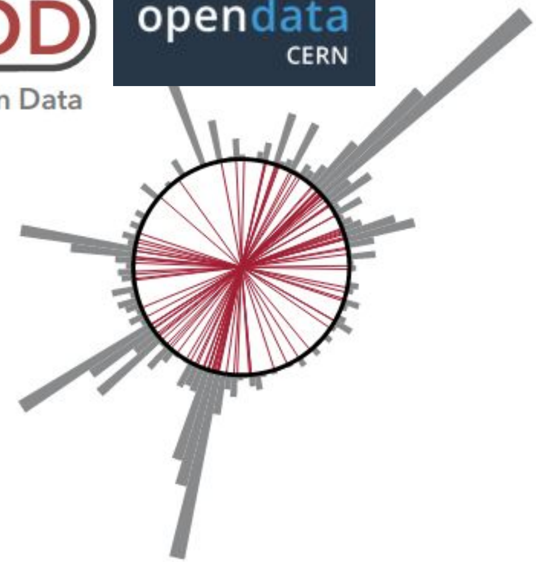
[Komiske, Mastandrea, Metodiev, Naik, Thaler, PRD 2020;
Larkoski, Marzani, Thaler, Tripathee, Xue, 1704.05066;
Cacciari, Salam, Soyez, 0802.1189;
http://opendata.cern.ch/]

Example 3: Jet Dataset

Using CMS Open Data:

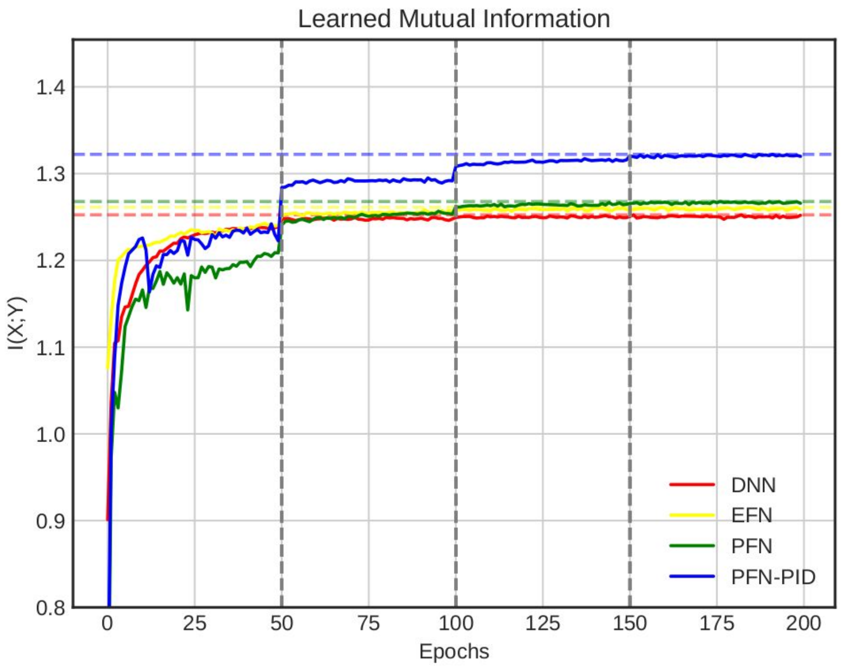
- *CMS2011AJets* Collection, SIM/GEN QCD Jets (AK 0.5)
- Select for jets with 500 GeV < Gen p_T < 1000 GeV, $|\eta|$ < 2.4, quality ≥ 2
- Select for jets with ≤ 150 particles
- Jets are rotated such that jet axis is centered at (0,0)
- Train on 100k jets

MOD
MIT Open Data
opendata CERN



Example 3: Mutual Information

Model	I(X;Z) [Natural Bits]
DNN	1.23
EFN	1.26
PFN	1.27
PFN-PID	**1.32**

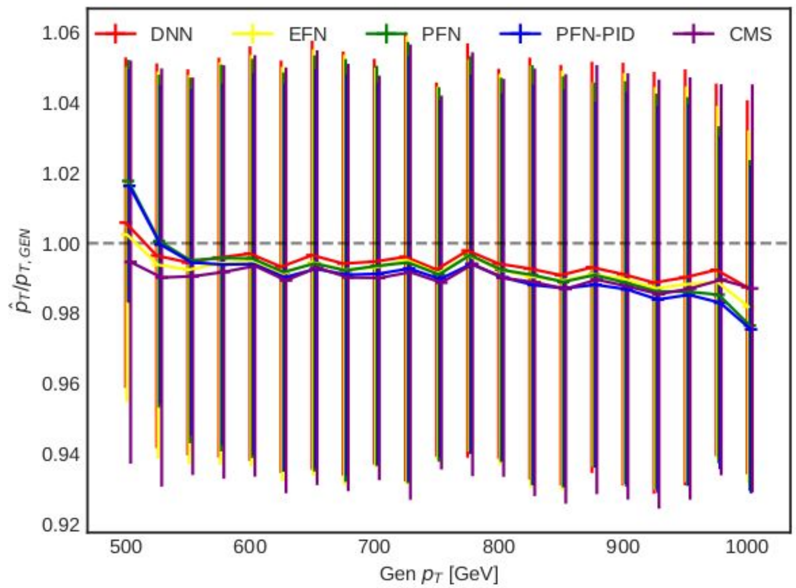


Reflects addition of more information in *X* for each model!

Jet Energy Scales

For jets with a true p_T between 695-705 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

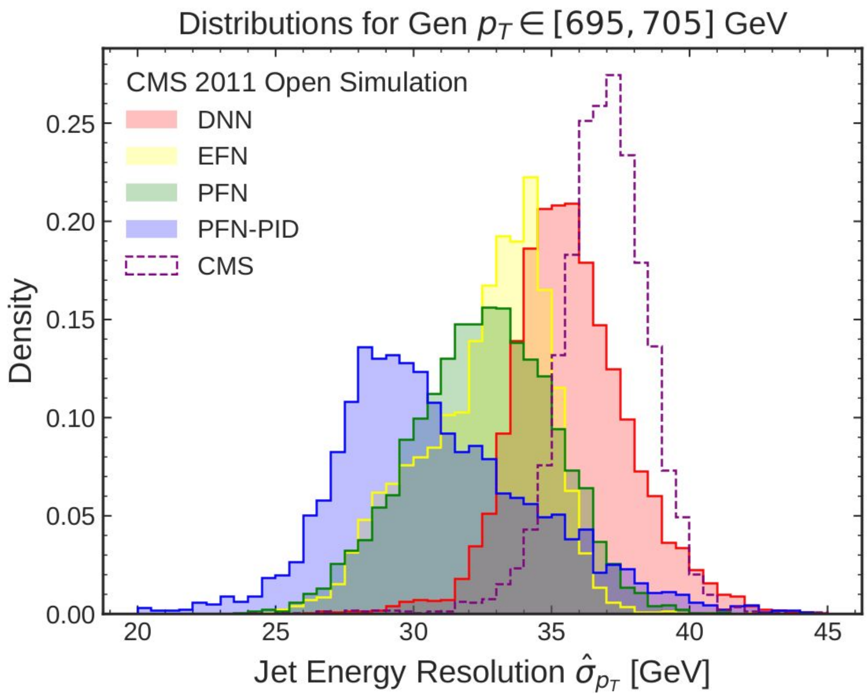


Close to 1.00 – unbiased estimates!

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	35.7 ± 2.1
EFN	32.6 ± 2.3
PFN	32.5 ± 2.5
PFN-PID	**30.8 ± 3.6**
CMS Open Data	36.9 ± 1.7

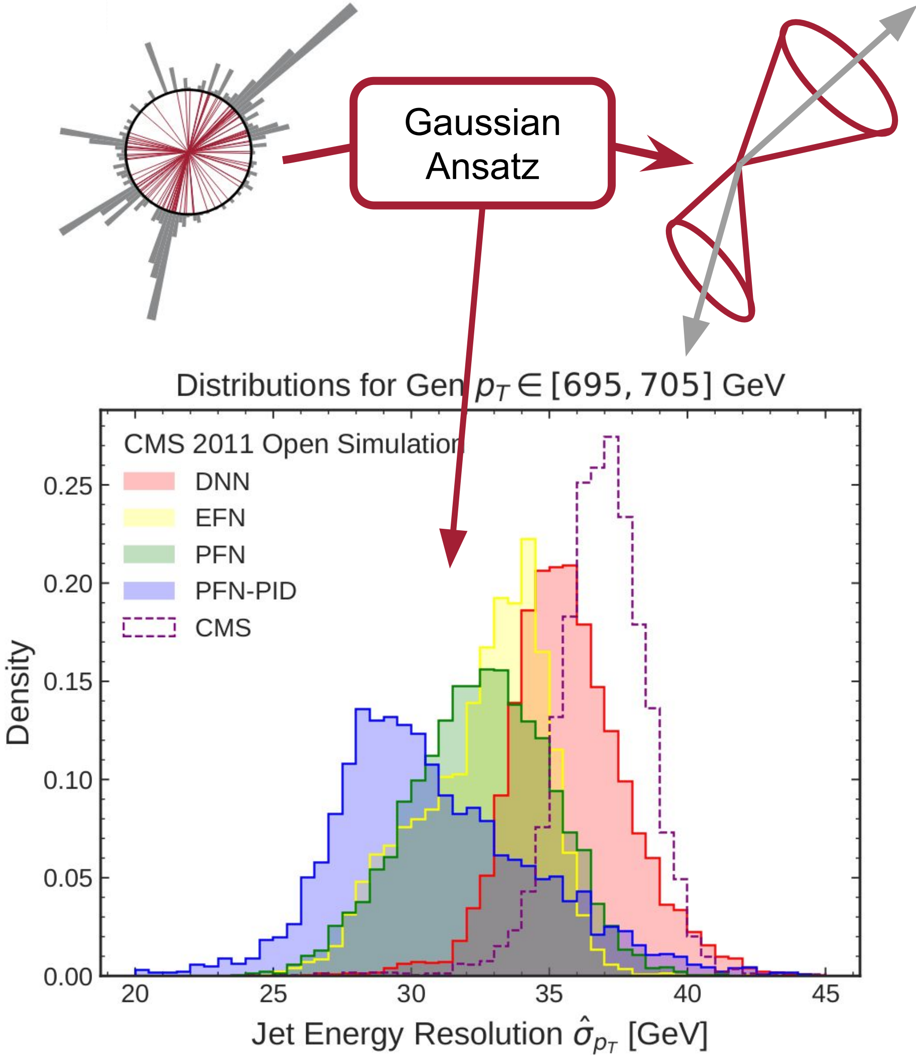


Conclusion

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 Z given X
- Estimating the **uncertainty** on Y for said inference

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



Download our repo!

Appendices

Data Based Calibration

"What if my detector simulation *p(x|z)* is imperfect"?

Given a *bad* simulator $p_{SIM}(x|z)$, we can correct it by matching it to data:

$$\hat{p}(x_D|z_T) = p_{\text{sim}}(h(x_D)|z_T)|h'(x_D)|$$

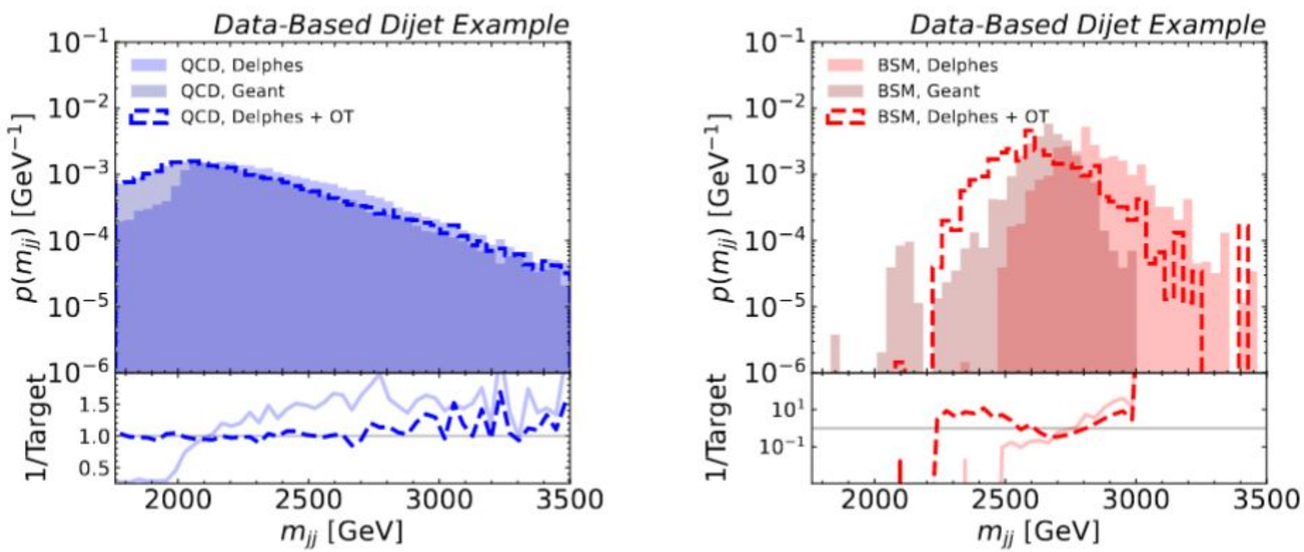
Where

$$h(x_D) = P_{\text{data}}^{-1}(P_{\text{sim}}(x_D))$$

The function *h* "optimally transports" points to where they belong and reweights them.

Data Based Calibration

BUT! There is a cost. We have to give up prior independence.



"Fixing" the Delphes simulation to match Geant4 works when trained on **Prior 1 (QCD)**, but fails miserably when applied to **Prior 2 (BSM)**, despite being the same detector simulation!

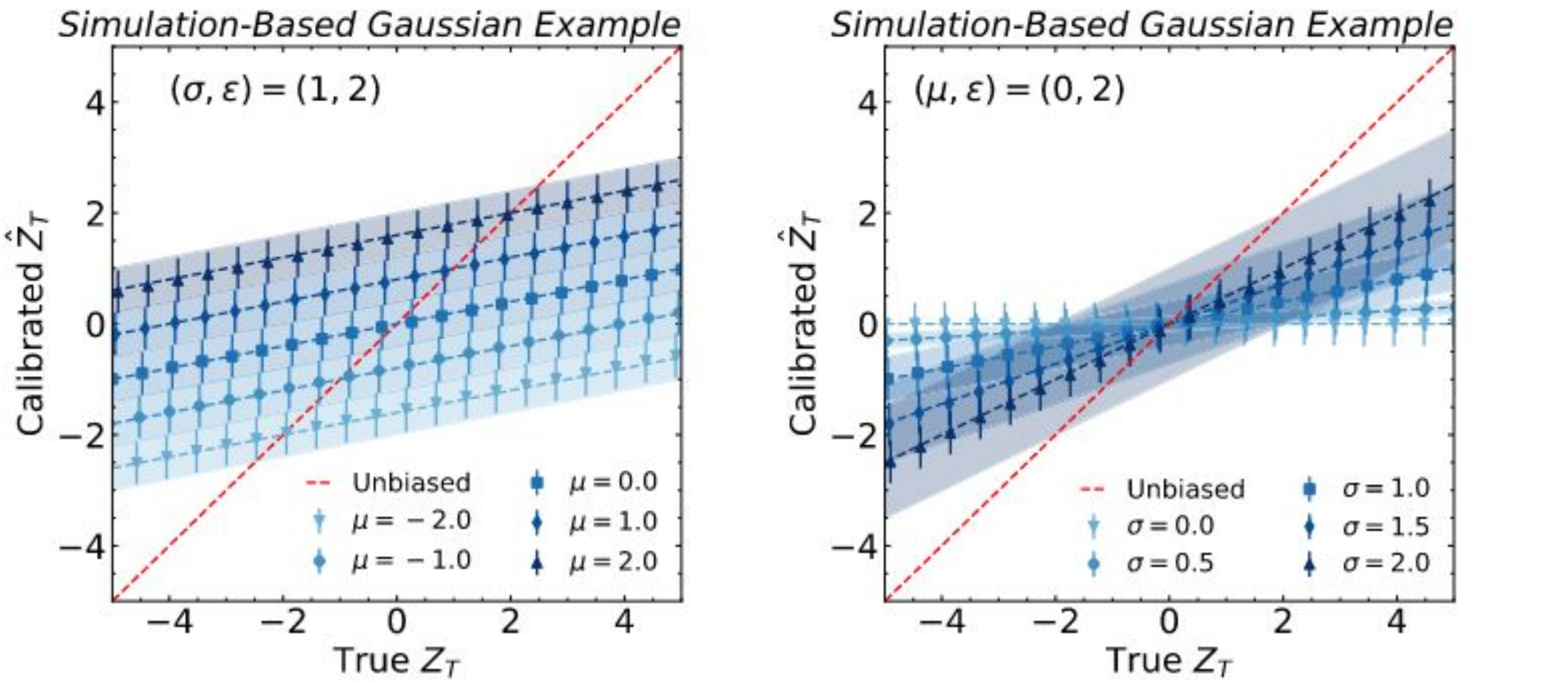
No (known) method of prior independent DBC, but no proof it is impossible!

Prior dependence of MSE

MSE fits for a gaussian noise model, for different choices of *z* prior.

Left: Different choices of mean

Right: Different choices of width

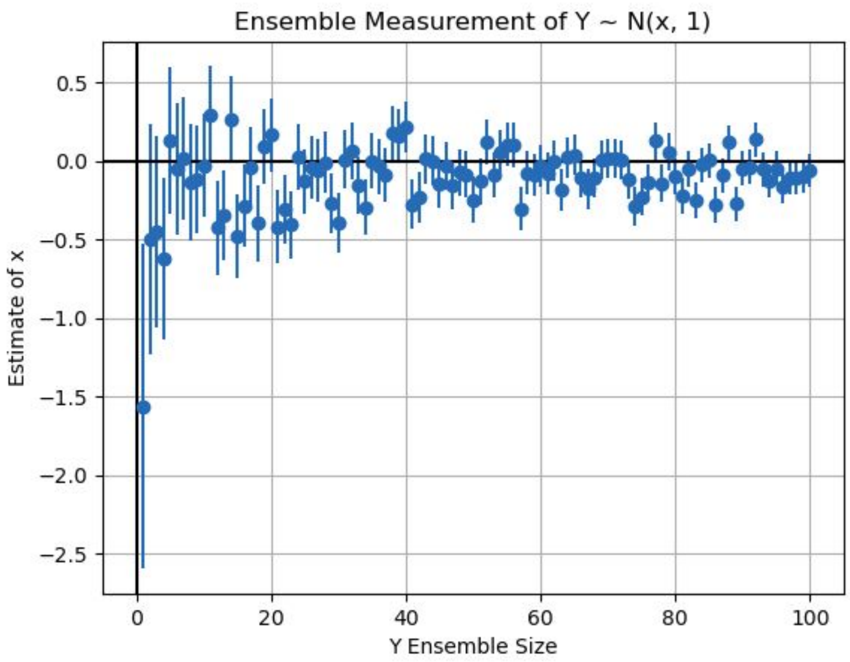


Ensembles and Unfolding

Once we have a procedure for estimating the maximum likelihood *Y* for a measured *X*, can extend to estimating a model parameter θ given an ensemble *N* data *I.I.D.* points X_i easily.

Or, we can **unfold** rather than have *x* and *z* be events, have *x* and *z* be the entire histogram. Training sets can be built by bootstrapping!

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

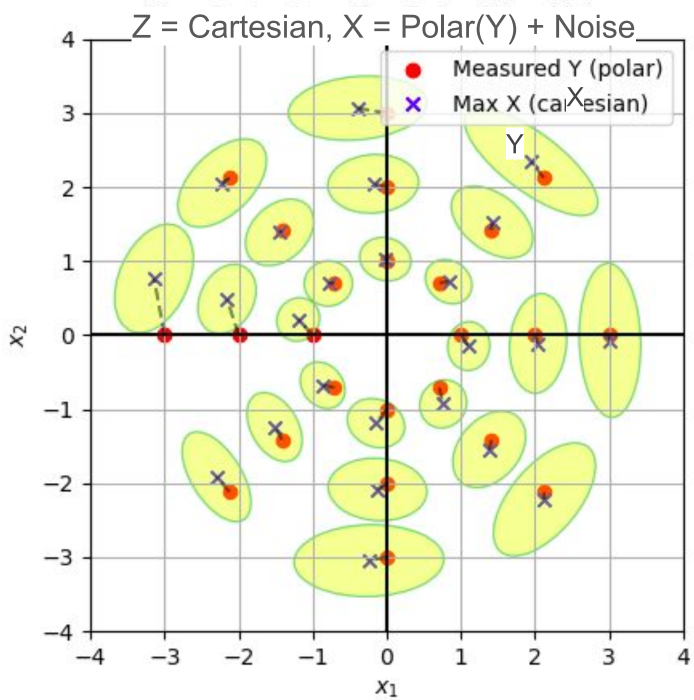


Multi Dimensional Test

Polar Coordinates Conversion

- Z = Uniform((-4,-4) , (-4, 4)
- X = (r, φ) + (N(0,0.25), N(0, π/12))

φ is in the coordinate patch (-π, π)



Other losses - Convergence

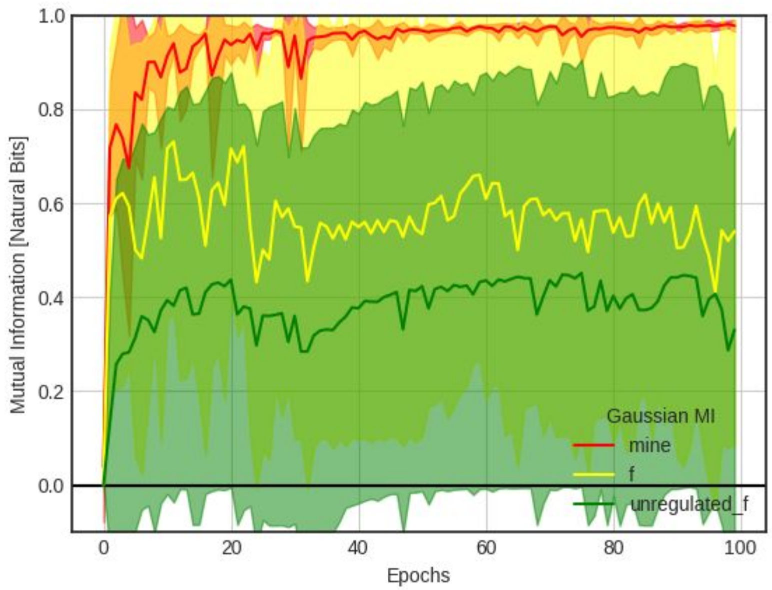
Simple X = Y + Gaussian Noise example

10 trials

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

$$\mathcal{L}_{\mathrm{DVR}}[T] = -\left(\mathbb{E}_{P_{XZ}}[T] - \log\left(\mathbb{E}_{P_X \otimes P_Z}\left[e^T\right]\right)\right)$$

$$\mathcal{L}_{\mathrm{MLC}}[T] = -\left(\mathbb{E}_{P_{XZ}}[T] - \mathbb{E}_{P_X \otimes P_Z}\left[e^T - 1\right]\right)$$



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

Note for any given *T*, DVR is a better bound on MI than MLC