Neural Empirical Bayes

Maxime Vandegar, Michael Kagan
SLAC

Collaborators: Antoine Wehenkel & Gilles Louppe
Uliège

2021 Meeting of the Division of Particles and Fields of the American Physical Society (DPF21)

Motivation

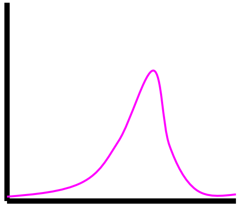
- Aim: find the distribution of uncorrupted data from corrupted observations by the detector (unfolding)

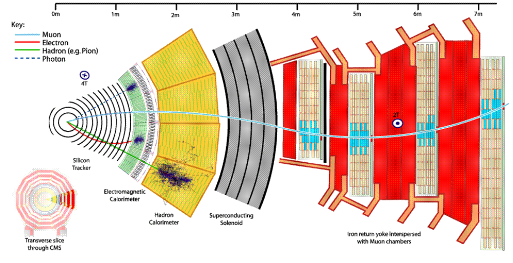
$$p(y) = \int p(y|x)p(x)dx$$

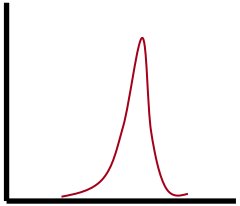
$y \sim p(y)$ = observed distribution

$p(y|x)$ = detector smearing (likelihood function)

$x \sim p(x)$ = true distribution





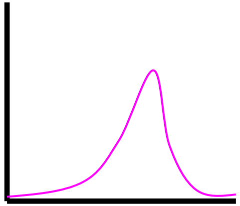


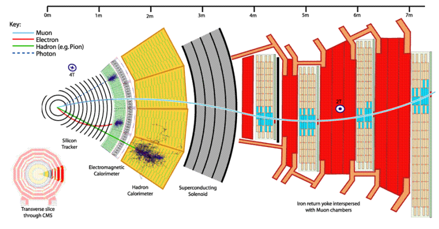
Histogram Approach

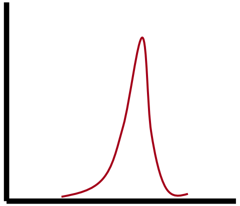
$y \sim p(y)$ = observed distribution

$p(y|x)$ = detector smearing

$x \sim p(x)$ = true distribution

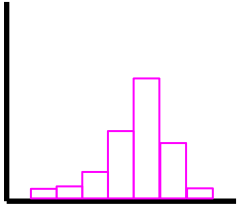




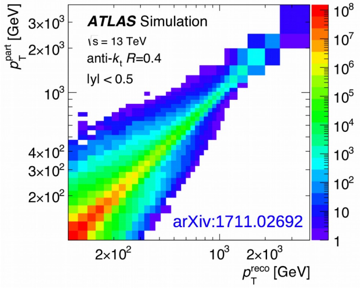


$$p(y) = \int p(y|x)p(x)dx$$

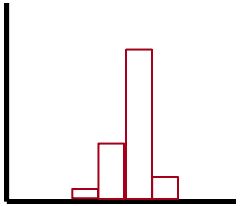
- Usually in HEP: solve discrete linear inverse problem: $b = Ra$



b



R



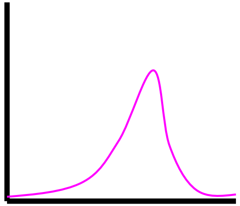
a

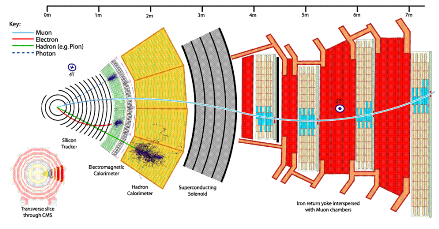
Histogram Approach

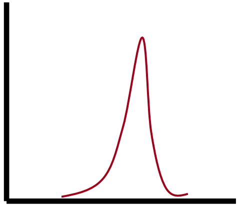
$y \sim p(y)$ = observed distribution

$p(y|x)$ = detector smearing

$x \sim p(x)$ = true distribution

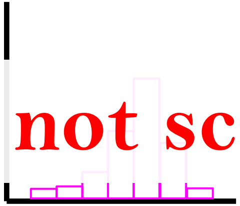


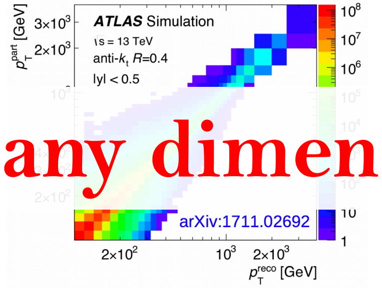


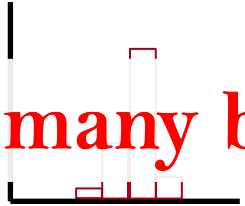


$$p(y) = \int p(y|x)p(x)dx$$

- Usually in HEP: solve discrete linear inverse problem: $b = Ra$







Does not scale to many dimensions or many bins

b R a

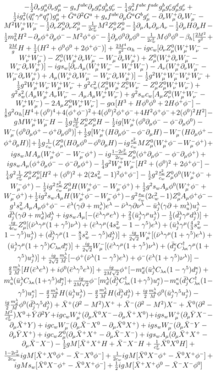
Empirical Bayes

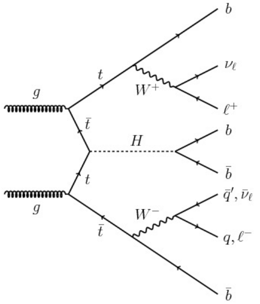
$$p(y) = \int p(y|x)p(x)dx$$

$$\approx \sum_i p(y|x_i) \quad where \quad x_i \sim p(x)$$

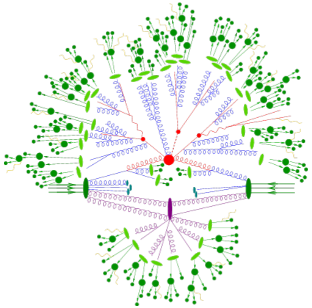
- **Goal:** Find $p(x)$ to maximize marginal likelihood of observations $\{y_i\}_{i=1}^{M}$
  - Continuous approach & scales better to many dimensions
- **Approach**
  - Learn a likelihood from simulated data → Use Neural Network
  - Parameterize family of possible source distributions → Use NN
  - Approximate integral with Monte Carlo integration
  - Learn parameters of source NN to make data more likely

Learn a likelihood from simulated data

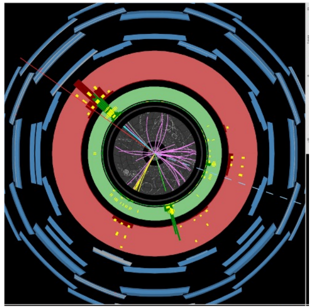




O(10) particles



O(100) particles

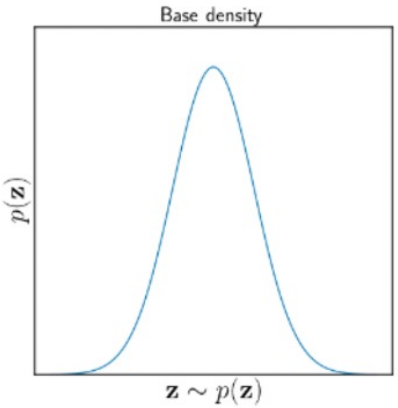


O(10^8) detector elements

- Don't know likelihood $p(y|x)$
- But we can simulate this process
  - Mechanistic understanding of interactions, put into code
  - Generate plausible samples of observations & fit a **density estimator** to the generated data

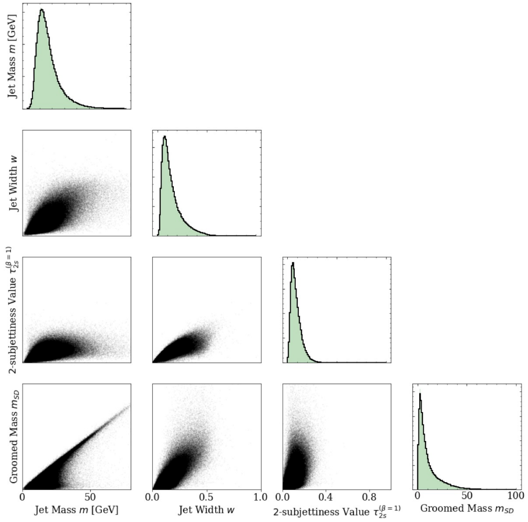
Parameterize family of possible source distributions

- Source (unfolded) model that can draw samples from a distribution with parameters θ
  - Defines a mapping from random noise to a learned source distribution $q_\theta(x)$



Random noise

Generative model



Density $q_\theta(x)$ defined by the model

Learn parameters of NN to make data more likely

- How to learn θ so that $q_\theta(x)$ approximates well source $p(x)$?
  - Fit θ in order to maximize the likelihood of having observed $\{y_i\}_{i=1}^{M}$
  - Learning by optimization

$$\begin{aligned}\theta^* &= \arg\max_\theta \mathcal{L} \\ &= \arg\max_\theta \mathbb{E}_{p_{data(y)}}\left[\log p(y)\right]\end{aligned}$$

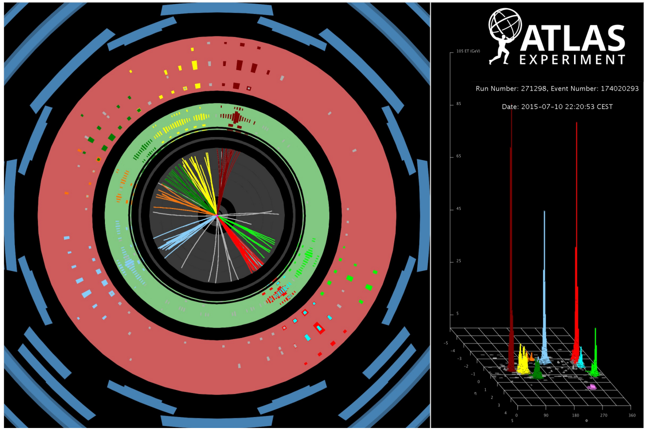
  - $\log p(y)$ depends on $q_\theta(x)$ but cannot be computed (requires to solve an intractable integral) → approximations

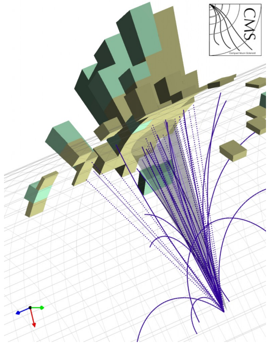
Approximate integral with Monte Carlo integration

$$\mathcal{L} = \mathbb{E}_{p_{data(y)}} [\log p(y)]$$
$$= \mathbb{E}_{p_{data(y)}} \left[\log \int p(y|x)p(x)dx\right]$$
$$\approx \sum_k \log \left[\sum_i p(y_k|x_i)\right], \quad x_i \sim q_\theta(x)$$

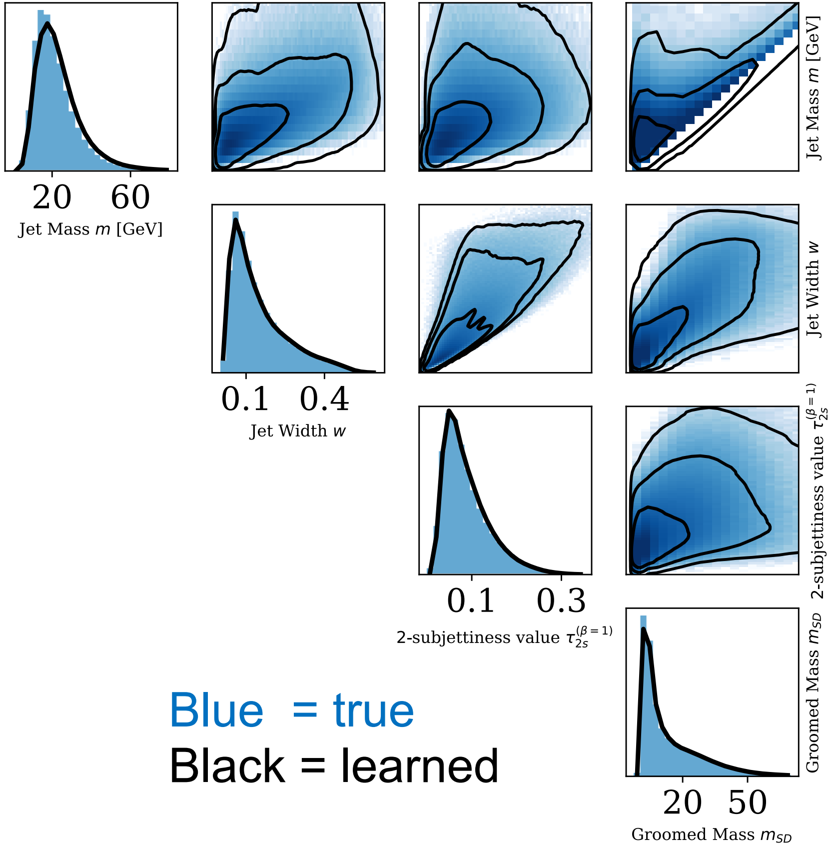
- Monte Carlo approximation of integrals
- **Efficient**
  - Sampling from learned source $q_\theta(x)$ is cheap
  - Evaluating learned $\mathrm{p(y|x)}$ is cheap
  - $q_\theta(x)$ and $\mathrm{p(y|x)}$ are NNs → computations can be parallelized on GPUs

Unfolding Jet Variables in Z+jet Events at the LHC





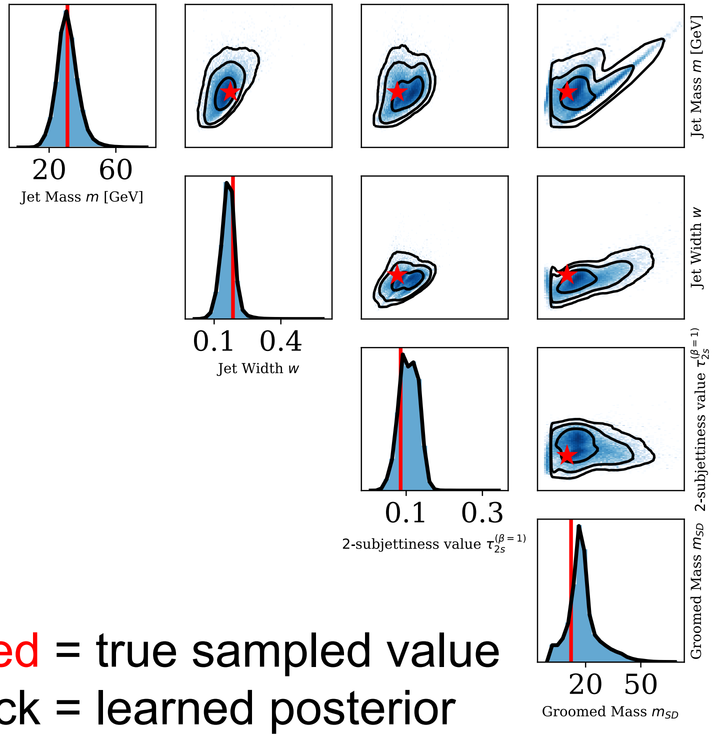
Jet: stream of particles produced by high energy quarks and gluons



Blue = true
Black = learned

Posterior Estimation in Z+Jets

- Posterior: $p(x|y) \propto p(y|x)p(x)$
  - Distribution of true values given observation
- Use case: reconstruction, with estimate of uncertainty
- **Method**
  1. Observation y
  2. Sample $x \sim p(x)$
  3. Rejection sampling: keep x w/ prob. $p(y|x)$



Red = true sampled value
Blue / Black = learned posterior

Inductive bias

- Ill-posed inverse problem:
  - Relevant domain knowledge (inductive bias) can be embedded in the structure of the generative model for constraining the solution space
  - Leads to considerable improvements
  - E.g. introducing symmetries, bounds & smoothness

Conclusion

SLAC

- Formulate unfolding as an Empirical Bayes problem
- Can use learned model for reconstruction w/ posterior
- Inductive bias helps mitigate the ill-posed nature of problems, and is easily introduced in the models

arXiv:2011.05836