Tuning the Parton Shower Parameters with the Marginal Likelihood

Kyle Cranmer, **Matthew Drnevich**, Sebastian Macaluso, Lauren Greenspan

July 6th, 2021

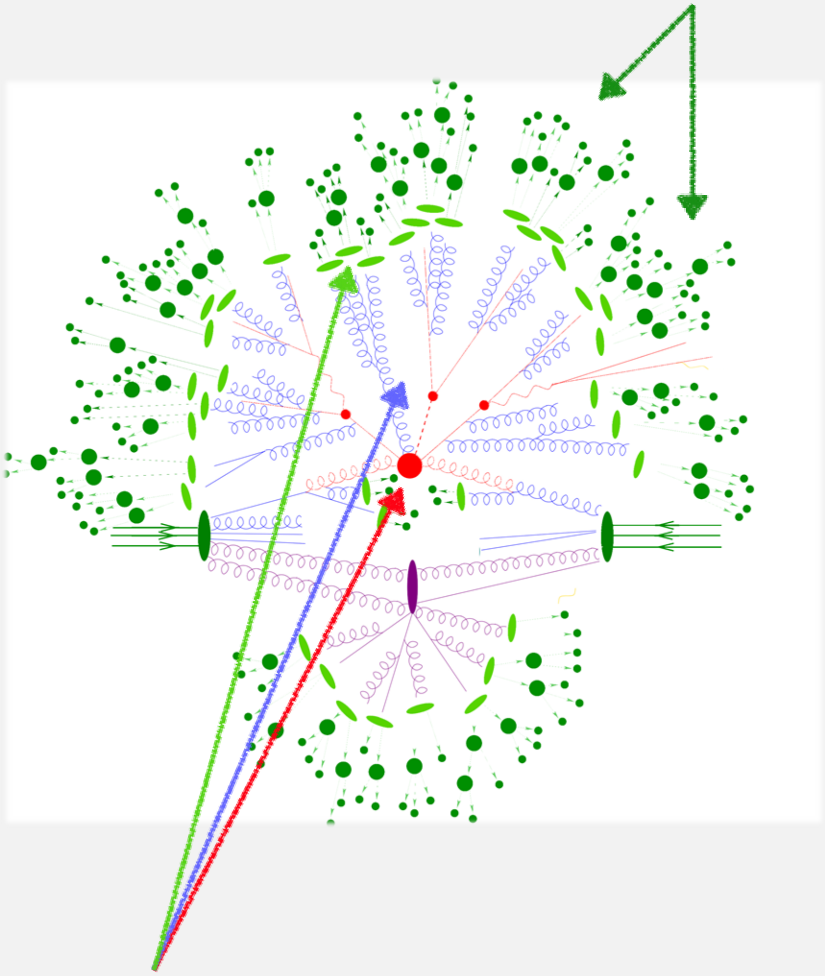
NYU

Parton Shower Models

simulating jet physics

- We have generators which can simulate parton showers
  - e.g. Pythia, Sherpa, Herwig, etc.
- They depend on a set of physically motivated parameters
- These determine the distribution of final state particles
- The detector only observes the final state particles
- The evolution of the shower is of interest for reconstruction
- It also depends on the shower parameters

We only observe the particles



Evolution of the shower is latent

Monte Carlo Tuning

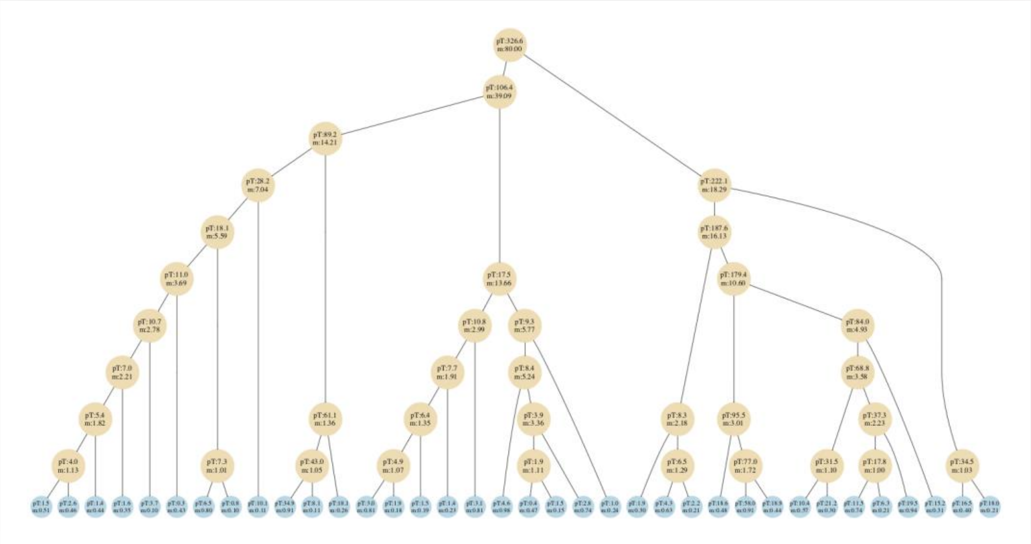
- Traditional approach:
  - Generate data using parton shower models
  - Make 1D projections of observables
  - Tune parameters by matching observables with data
  - Professor w/ Rivet
    - Buckley, et. al. https://arxiv.org/abs/0907.2973
- Inefficient and wastes information
- Traditional approaches don't have access to a likelihood $p(data \mid parameters)$
- Shower Deconstruction
    - Tractable likelihood, but brute force marginalization (scalability issues); focused on tagging not tuning
    - Phys. Rev. D 2011 https://inspirehep.net/literature/889900
    - Soper, Spannowsky

- Other ML related work based on surrogates for the intractable likelihood
  - Adversarial Variational Optimization
    - Non-differentiable simulator; GAN-like NN classifier
    - AISTATS 2019 https://arxiv.org/abs/1707.07113
    - Louppe, Hermans, Cranmer
  - DCTR
    - NN classifier as likelihood ratio surrogate; differentiable optimization
    - Phys. Rev. D 2020 https://inspirehep.net/literature/1744598
    - Andreassen, Nachman

Statistical Framing of Tuning

maximum likelihood estimation

- Ideally, we would use the likelihood to fit the model to observed data
- Model parameters can be inferred through stastical estimation, e.g. MLE or Bayesian methods
- These use the full information of the model and are statistically efficient
- Ginkgo combined with the Trellis algorithm allows us to directly compute the likelihood for inference



$p($  $|\theta)$

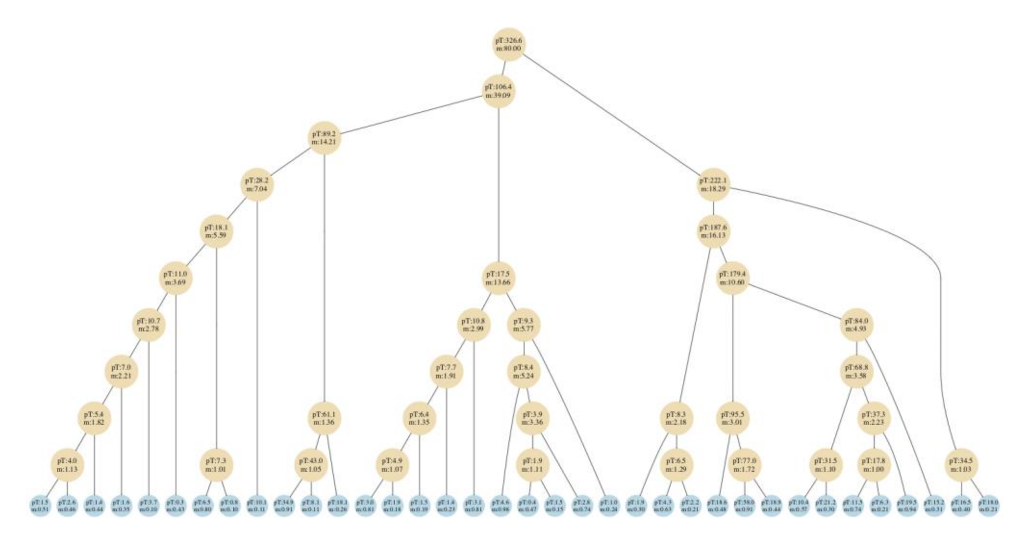
$$\theta_{MLE} = \mathrm{ArgMax}_\theta \; p(x_{obs}|\theta)$$

Ginkgo Refresher

a toy parton shower simulator

- Ginkgo uses a recursive algorithm to generate a binary tree where the leaves are the jet constituents
- Implements some key features needed for physics simulator:
  - Momentum conservation
  - A running splitting scale
- Model parameters that can be tuned
  - Threshold energy for stopping (like confinement)
  - Decay rate (jet-type dependent)

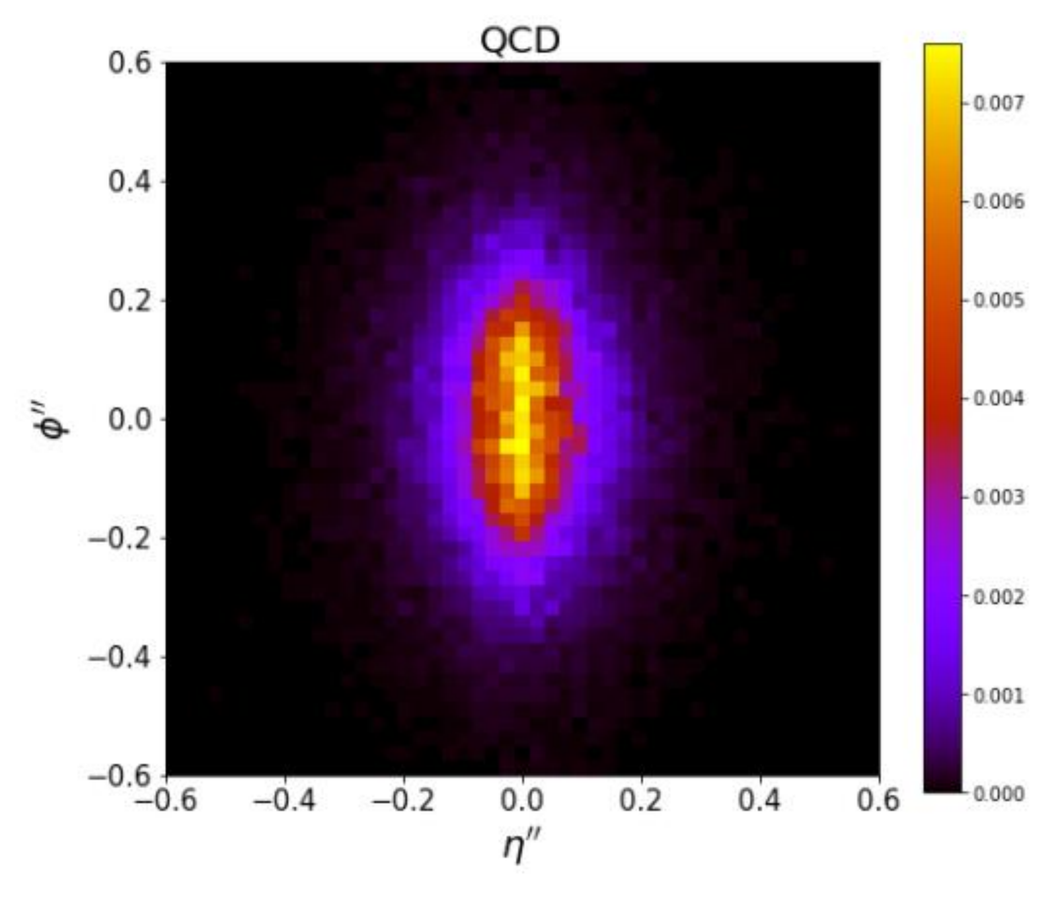
https://github.com/SebastianMacaluso/ginkgo



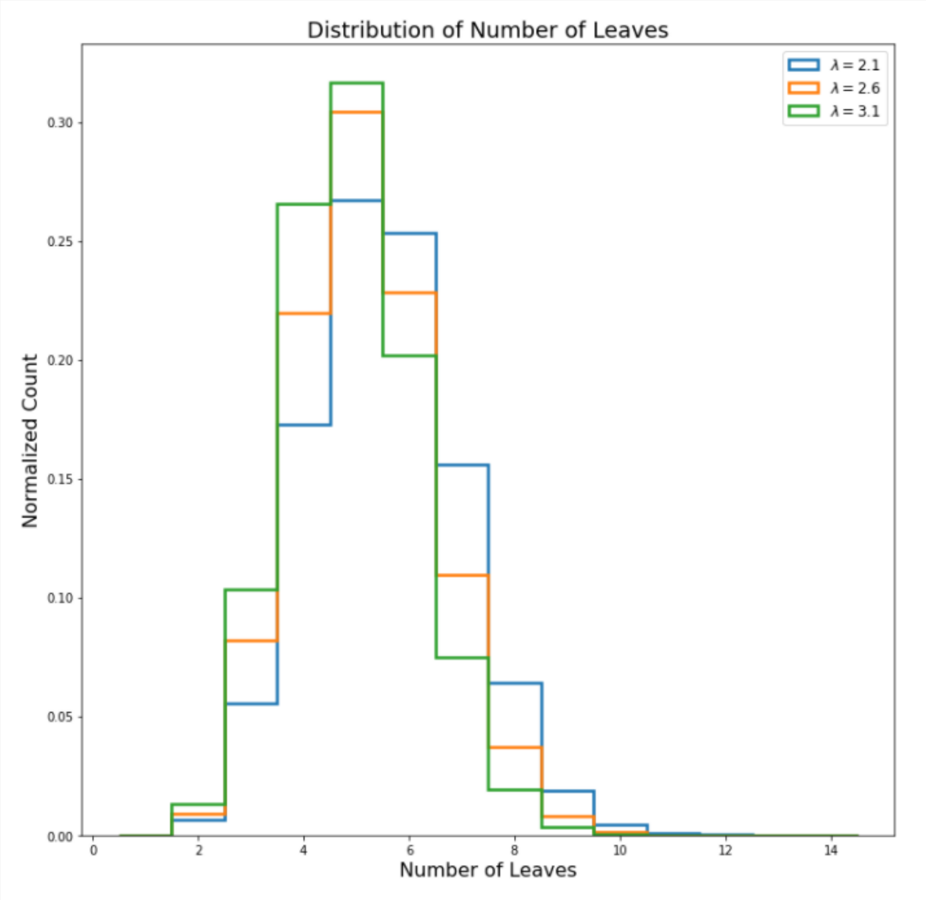
Kyle Cranmer, Sebastian Macaluso, Duccio Pappadopulo

Ginkgo Jets

Phi vs Eta



Jet Multiplicity



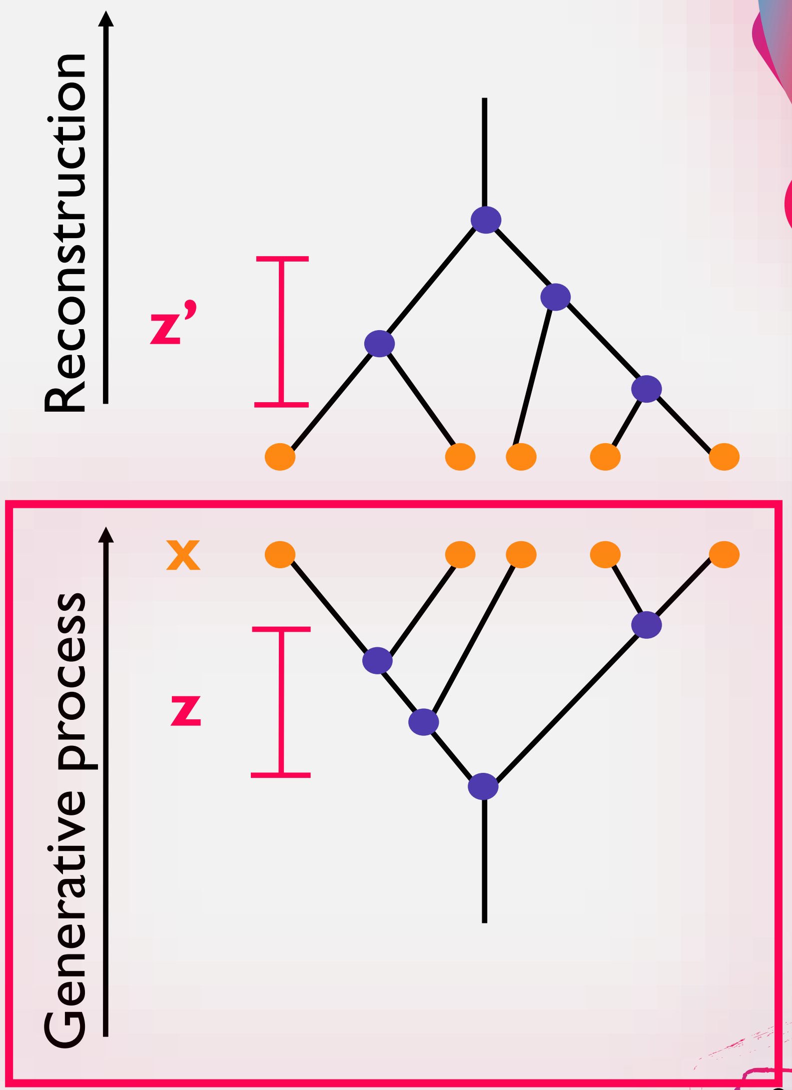
Ginkgo Generative Process

constructing a tractable likelihood

- The evolution at every step depends only on the parent 4-momentum

$$t_L \sim f(t|\lambda, t_P) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{t_P} e^{-\frac{\lambda}{t_P} t} \quad t_R \sim f(t|\lambda, t_P, t_L) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{(\sqrt{t_P} - \sqrt{t_L})^2} e^{-\frac{\lambda}{(\sqrt{t_P}-\sqrt{t_L})^2} t}$$

- If $t < t_{cut}$ then stop, where t is the invariant mass squared
- Asymmetric under $L \leftrightarrow R$
- Particles are randomly permuted after splitting



from Sebastian Macaluso

Ginkgo Likelihood

constructing a tractable likelihood

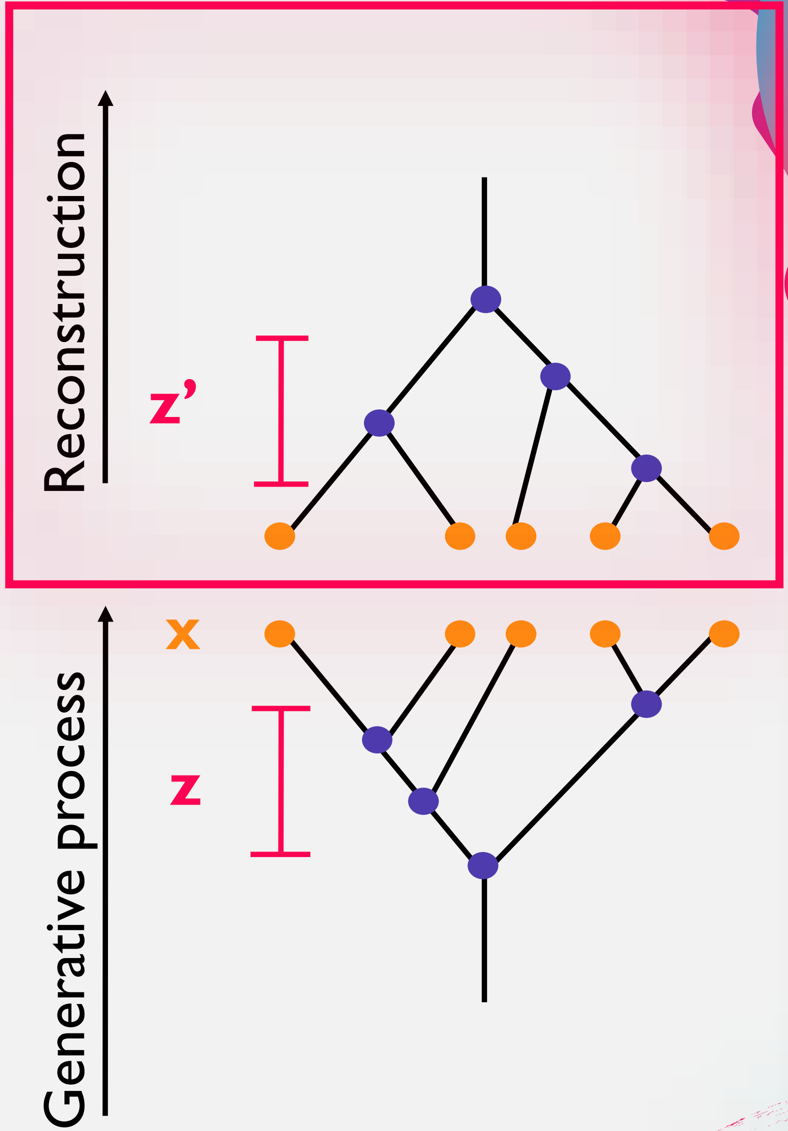
- The joint likelihood is factorized in an autoregressive form, depending on the particular tree, z

$$p(x, z|\theta) = \prod_j p(x_j|z_{\text{parent}(x_j)}, \theta) \prod_i p(z_i|z_{\text{parent}(z_i)}, \theta)$$

- In order to evaluate the probability of the splitting, we need to reconstruct the parent from the children

$$t_L \sim f(t|\lambda, t_{\text{P}}) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{t_{\text{P}}} e^{-\frac{\lambda}{t_{\text{P}}}t} \quad t_R \sim f(t|\lambda, t_{\text{P}}, t_L) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{(\sqrt{t_{\text{P}}} - \sqrt{t_L})^2} e^{-\frac{\lambda}{(\sqrt{t_{\text{P}}}-\sqrt{t_L})^2}t}$$

- Reconstruction probability is symmetrized
  - $\frac{1}{2}(f(t_L, t_R|\, \lambda, t_P) + f(t_R, t_L|\, \lambda, t_P))$



from Sebastian Macaluso

The Marginal Likelihood

how to integrate out the showering history

- Need to be able to evaluate this on observed data
- This also allows us to compute the exact likelihood ratio
  - See Lauren Greenspan's talk next!
- We must integrate out the showering history
- Typically, this is an intractable problem and where other approaches fail
  - Grows as (2N-3)!!
- Some machine learning approaches that approximate the marginal likelihood

$$p(x, z|\theta) = \prod_j p(x_j|z_{\text{parent}(x_j)}, \theta) \prod_i p(z_i|z_{\text{parent}(z_i)}, \theta)$$

$$\underbrace{p(x|\theta)}_{?} = \int dz \; \underbrace{p(x, z|\theta)}_{\text{known}}$$

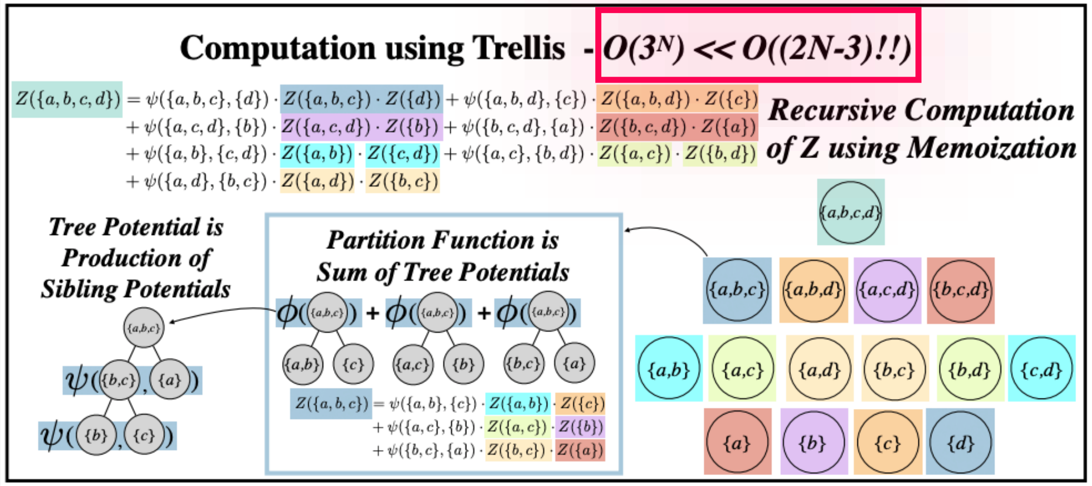
Hierarchical Cluster Trellis

how to integrate out the showering history

- The Hierarchical Cluster Trellis algorithm empowers this marginalization
- This is a model-independent algorithm for marginalizing over all binary tree histories
- Ginkgo is a perfect test case
- There are $(2N-3)!! \times 2^{N-1}$ possible binary trees including permuations
- Trellis reduces the computational complexity from brute force

https://github.com/SebastianMacaluso/ClusterTrellis

# Leaves	4	7	9	12
# Trees	120	~665k	~520M	~28T



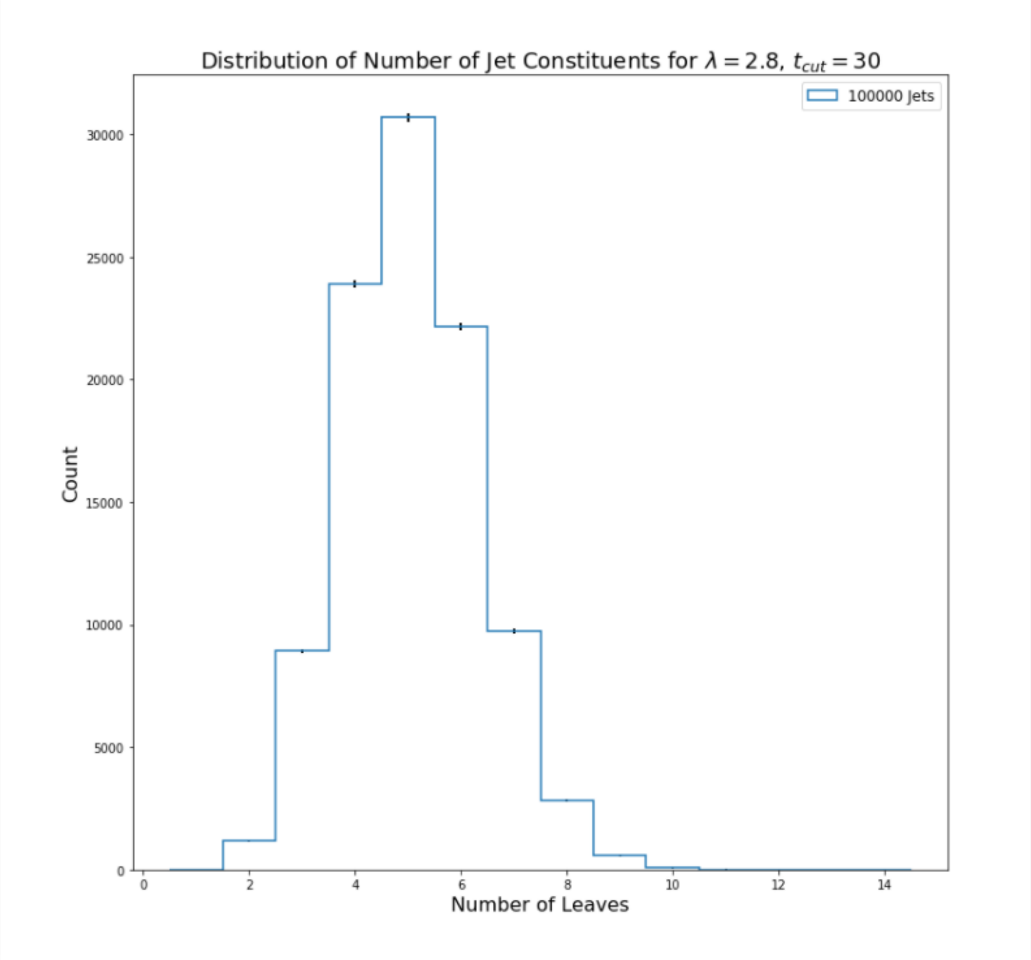
from Sebastian Macaluso

Results

Tuning using Maximum Likelihood Estimation

Observed Dataset

- Generated a dataset of 100k jets using Ginkgo
- Parameters fixed to $\lambda = 2.8, t_{cut} = 30\ GeV^2$, with $|\vec{p}_{jet}| = 400\ GeV$ and $m_{jet} = 30\ GeV$
- Small number of jet constituents for proof of concept



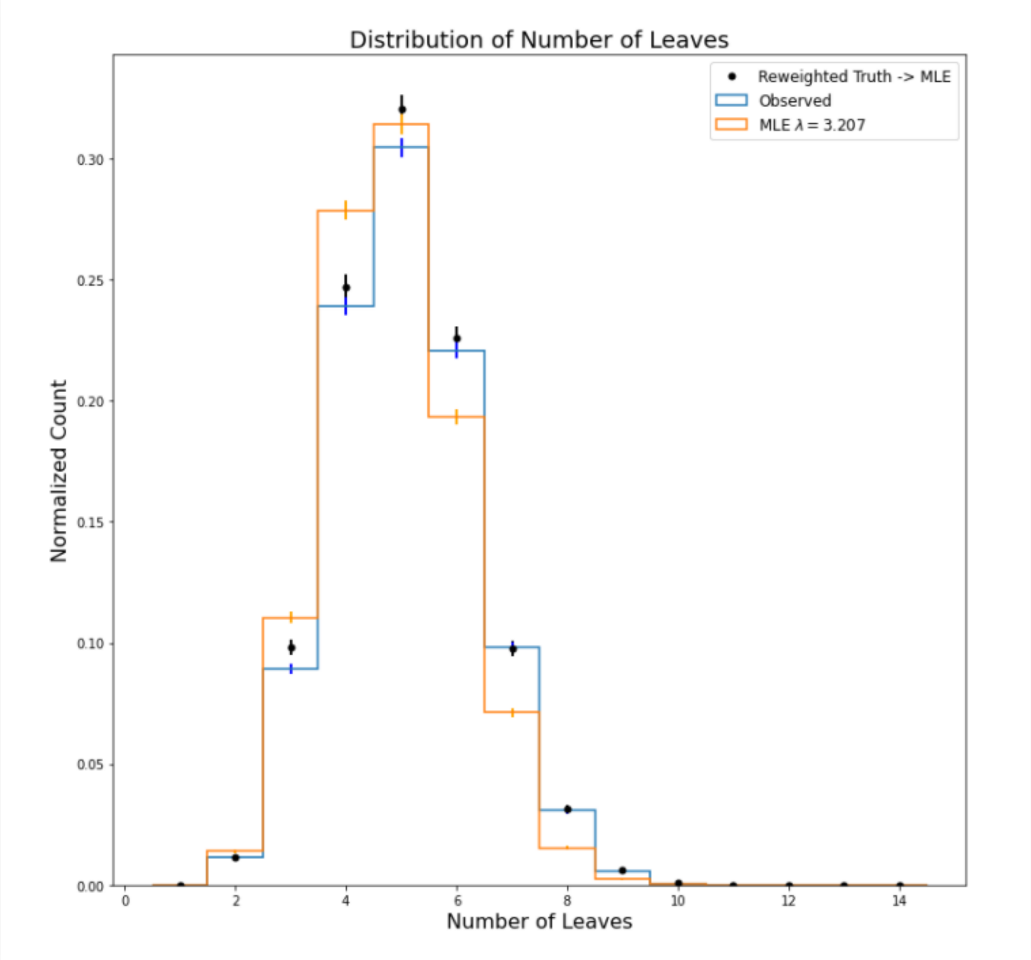
Subtle Normalization Issue

reweighting-based closure

- With the likelihood we can reweight the distribution from one value of λ to another
- This reveals a normalization problem with the likelihood, which we have not been able to track down *yet*
- This leads to a bias in the MLE
- But we can measure

$$f(\lambda, \lambda_t) = \mathbb{E}_{p(x|\lambda_t)}\left[p(x|\lambda)/p(x|\lambda_t)\right]$$

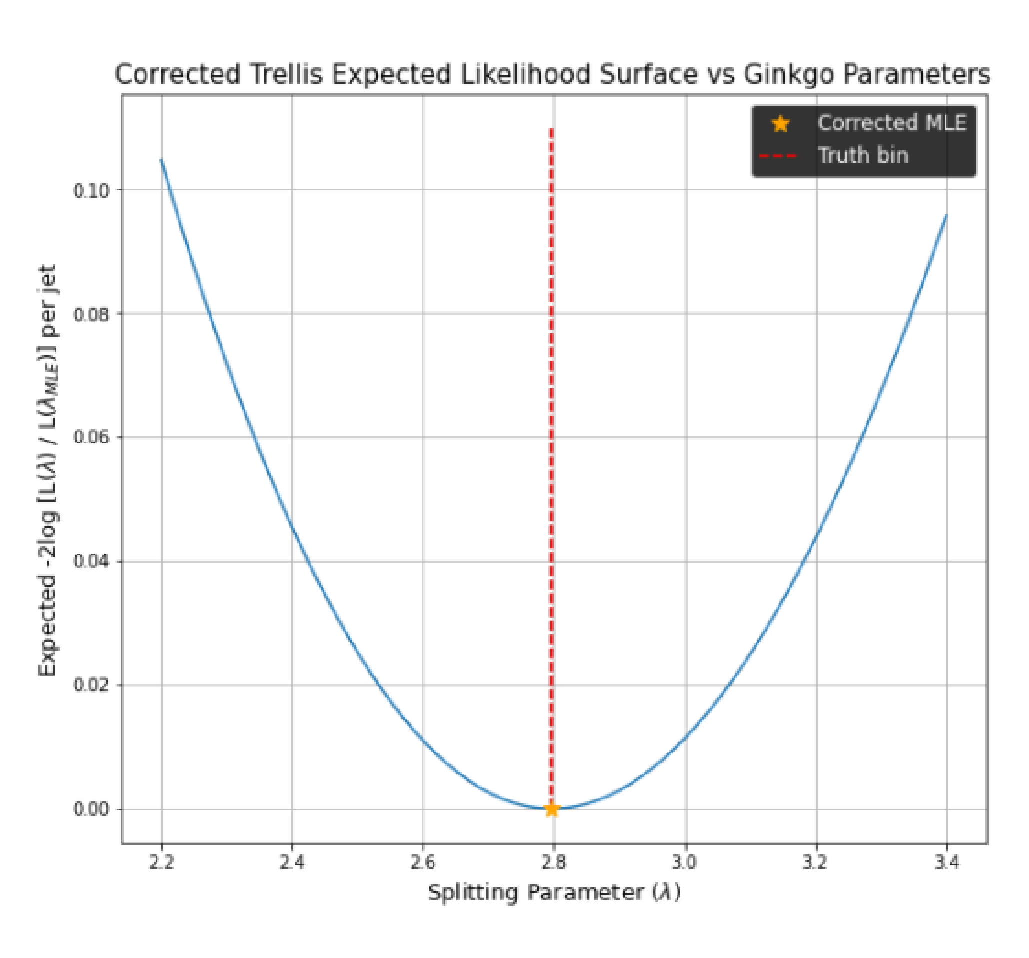
  which should be unity
- Correct the likelihood by $1/f$



Maximum Likelihood Estimate

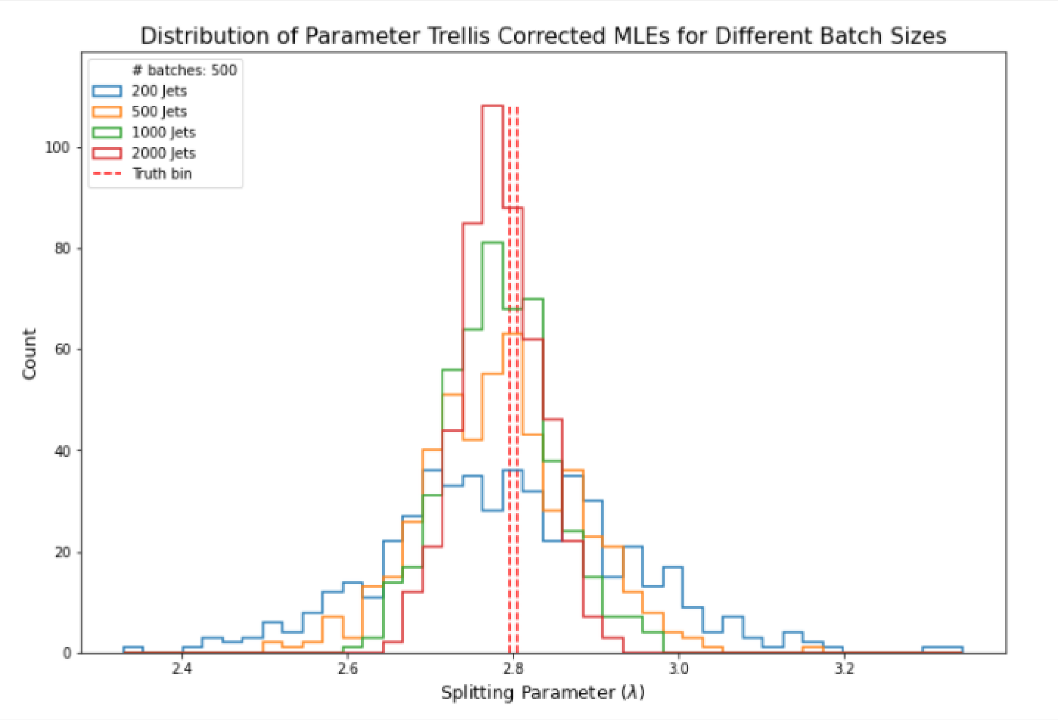
tuning procedure

- A 1D grid search is performed for optimizing λ, the splitting rate
- Includes the normalization correction

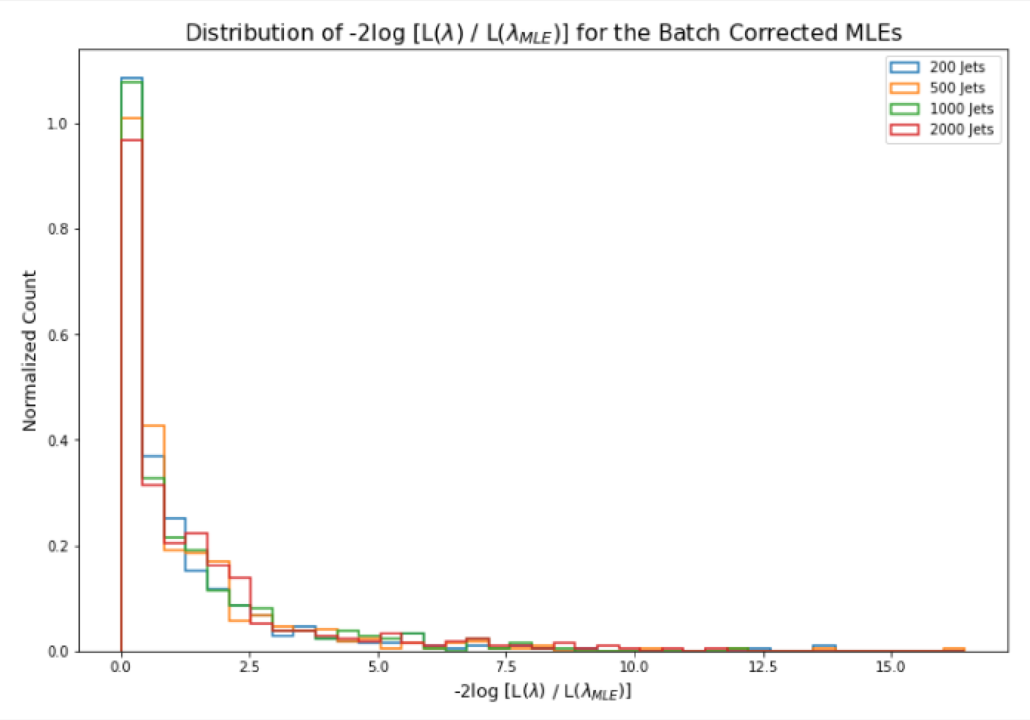


Maximum Likelihood Estimate

Distribution of the MLE



Distribution of the χ^2 test statistic



Summary

- Explored showering parameter tuning on Ginkgo, which has a tractable likelihood
- Implemented parameter tuning using the full information of the simulator model via maximum likelihood estimation
- The Trellis algorithm enabled marginalizing over possible jet clusterings
- Generalizes well to other probabilistic showering models
- To the best of our knowledge, this is the first demonstration of tuning the shower model using the full likelihood marginalized over all showering histories

Probabilistic Parton Shower

+

Cluster Trellis

=

Tuning with Marginal Likelihood

Thank You!

Some Github links

- Ginkgo: https://github.com/SebastianMacaluso/ginkgo
- Cluster Trellis: https://github.com/SebastianMacaluso/ClusterTrellis
- Ginkgo Inference: https://github.com/mdkdrnevich/ginkgo-inference

Backup

Ginkgo Likelihood Math Written Out

$$p(x, z|\theta) = \prod_j p(x_j|z_{\mathrm{parent}(x_j)}, \theta) \prod_i p(z_i|z_{\mathrm{parent}(z_i)}, \theta)$$

$$t_L \sim f(t|\lambda, t_{\mathrm{P}}) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{t_{\mathrm{P}}} e^{-\frac{\lambda}{t_{\mathrm{P}}} t} \quad t_R \sim f(t|\lambda, t_{\mathrm{P}}, t_L) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{(\sqrt{t_{\mathrm{P}}} - \sqrt{t_L})^2} e^{-\frac{\lambda}{(\sqrt{t_{\mathrm{P}}}-\sqrt{t_L})^2} t}$$

$$p(z_i|z_{parent(z_i)}, \theta) = p(t_L, t_R|\lambda, t_P) = f(t_R|\lambda, t_P, t_L) f(t_L|\lambda, t_P)$$

$$\frac{1}{2}(f(t_L, t_R|\,\lambda, t_P) + f(t_R, t_L|\,\lambda, t_P))$$

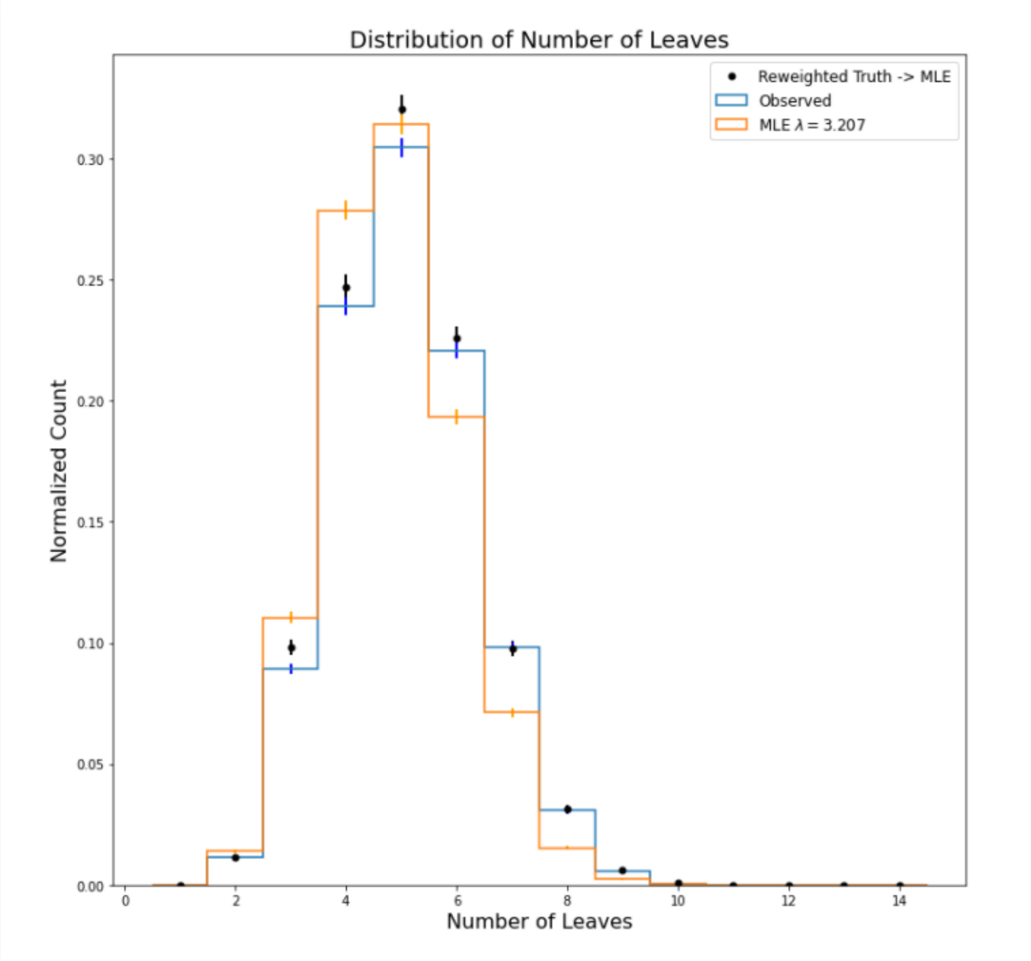
Iterative Reweighting Procedure

correcting the MLE

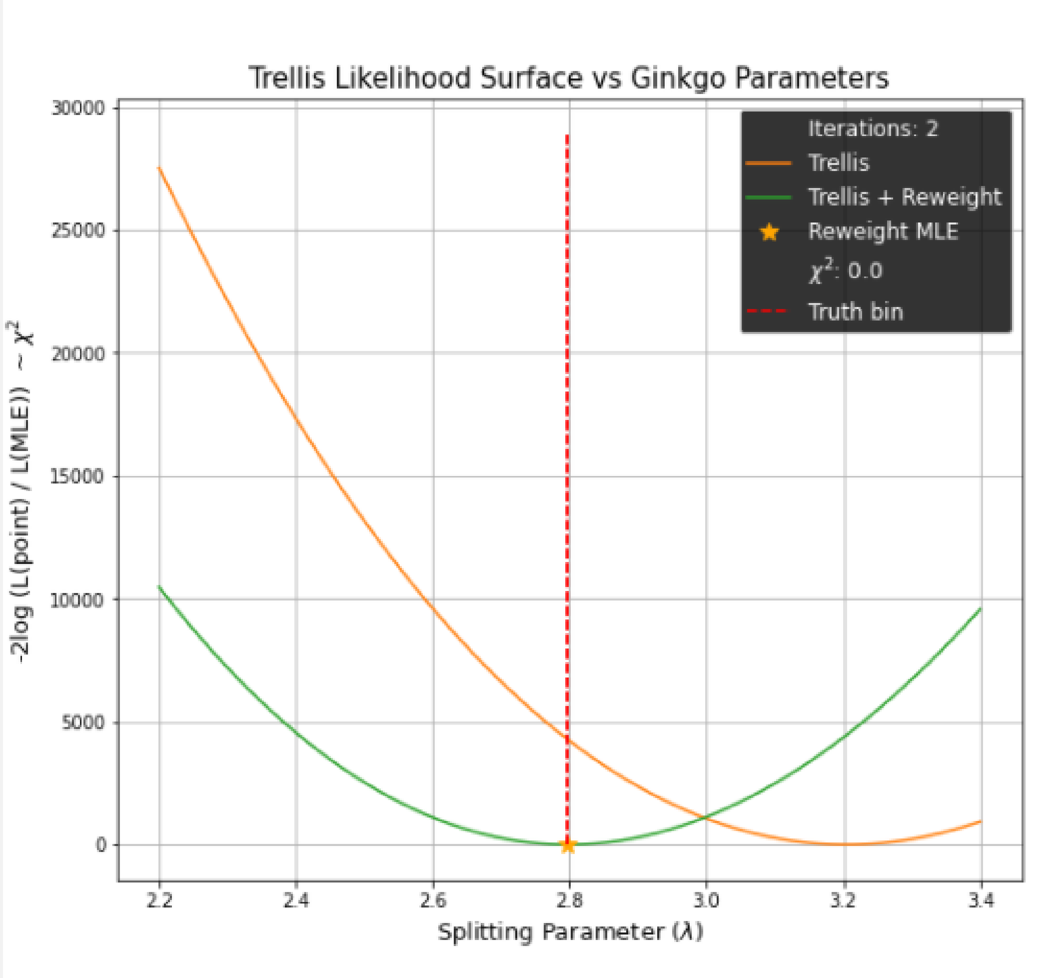
- The MLE is initially biased due to something missing in the Ginkgo likelihood
- We correct for this bias by introducing a reweighting procedure
- Build a simple approximation for

  $$f(\lambda, \lambda_t) = \mathbb{E}_{p(x|\lambda_t)}\left[p(x|\lambda)/p(x|\lambda_t)\right]$$
- Define an iterative correction

  $$\lambda^*_{n+1} = \mathrm{ArgMax}_\lambda p(x_{obs}|\lambda)/f(\lambda, \lambda^*_n)$$
- Fast and effective



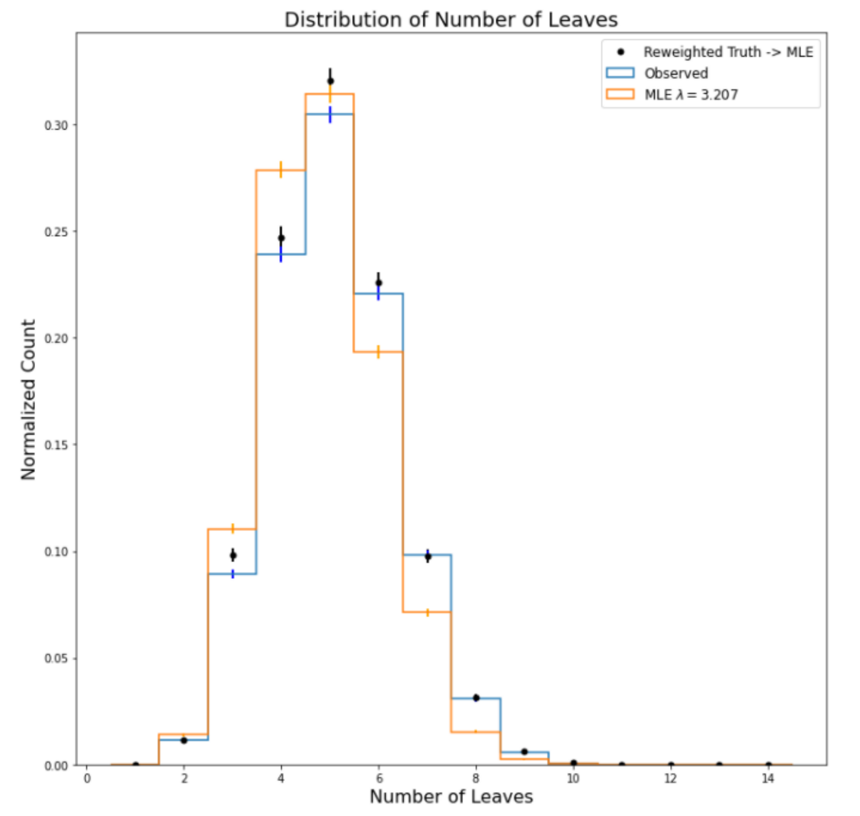
Iterative Reweighting Procedure



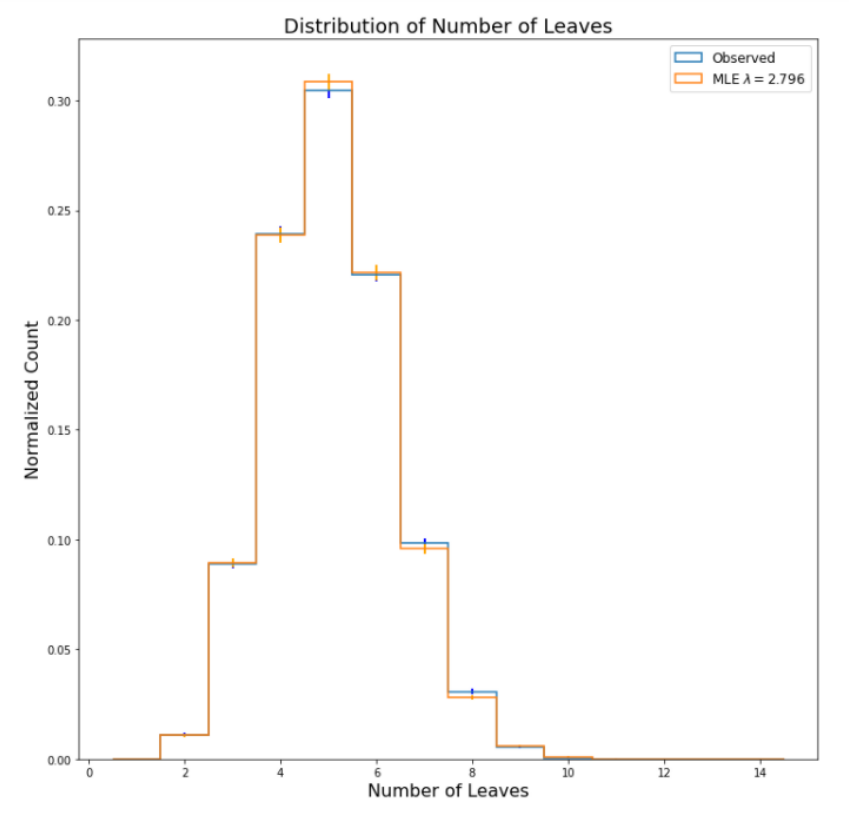
Likelihood Correction on Observed Data

Iterative Reweighting Procedure

Before

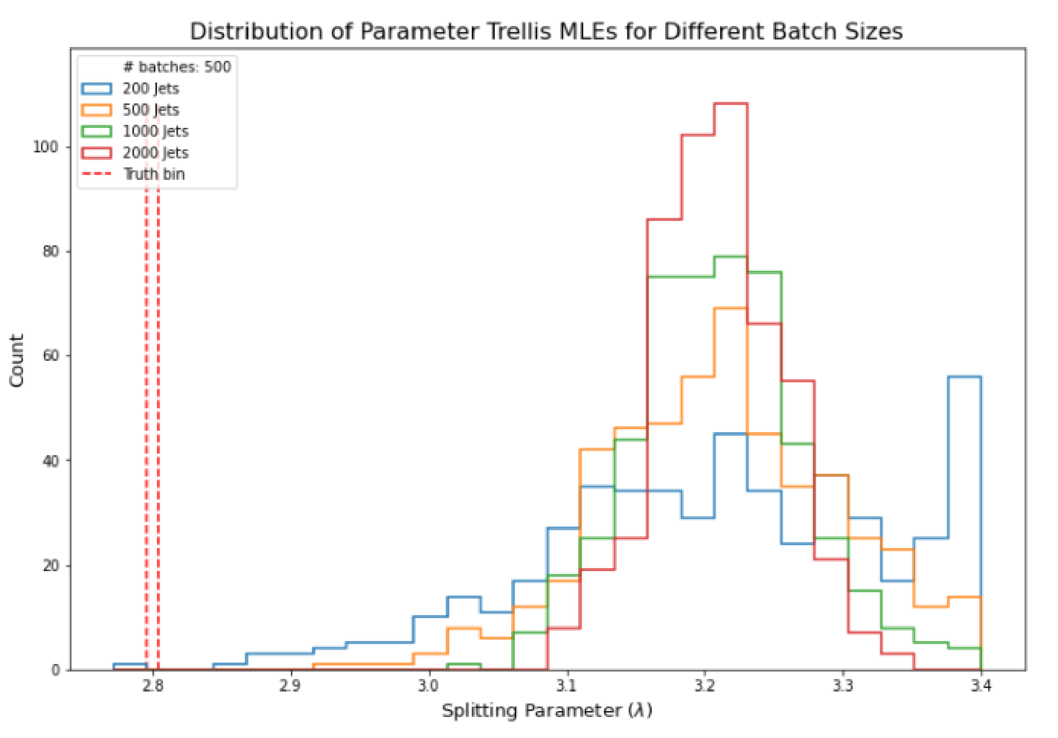


After 2 Iterations

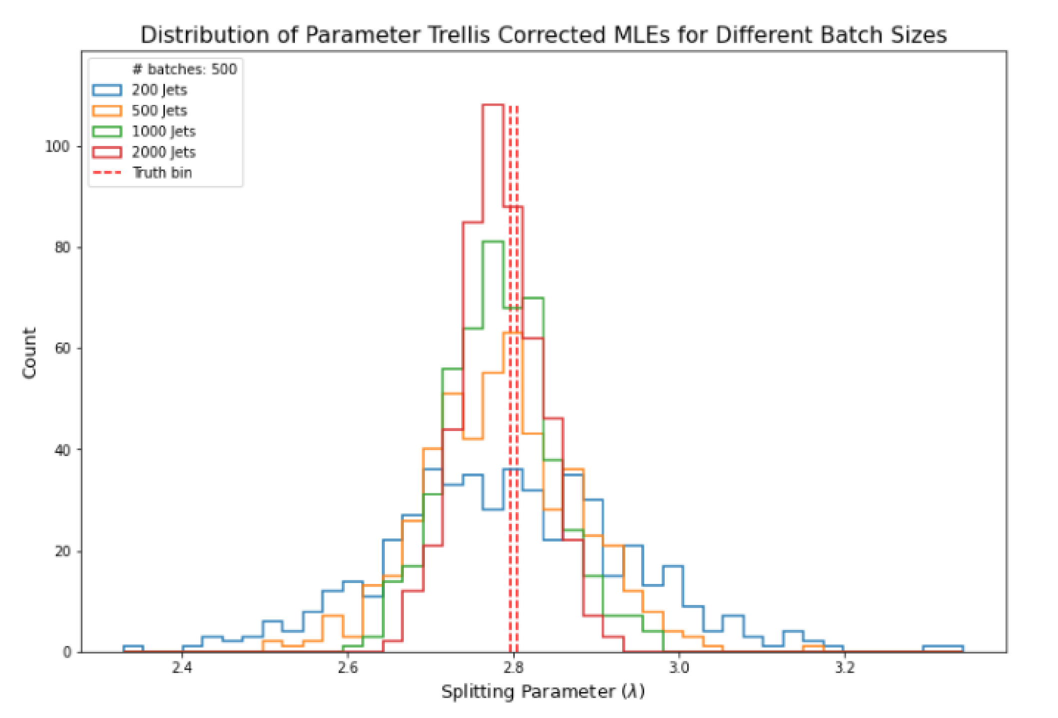


MLE Distributions

Marginal MLE

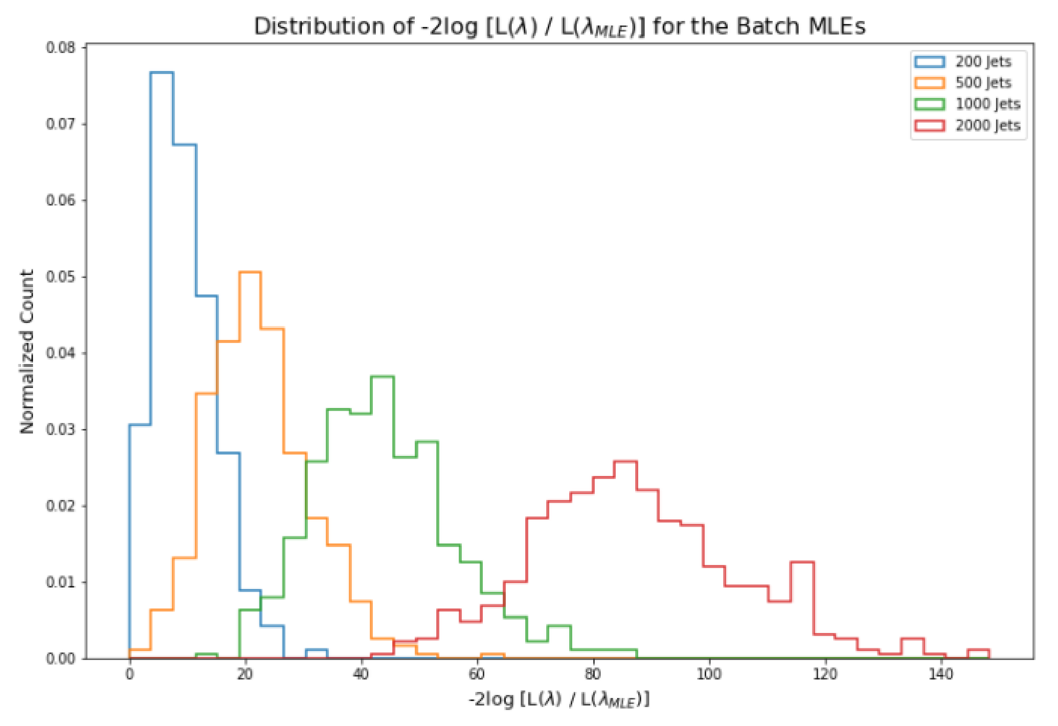


Reweighted MLE



χ^2 Test Statistic Distributions

Marginal χ^2 Test Statistic



Reweighted χ^2 Test Statistic