Novel approach to jet substructure measurement in pp collisions at $\sqrt{s}$ = 200 GeV in STAR

Youqi Song (youqi.song@yale.edu) for the STAR Collaboration

APS DNP 2022, New Orleans, LA

10/29/2022

Supported in part by U.S. DEPARTMENT OF ENERGY Office of Science

DNP2022

Motivation

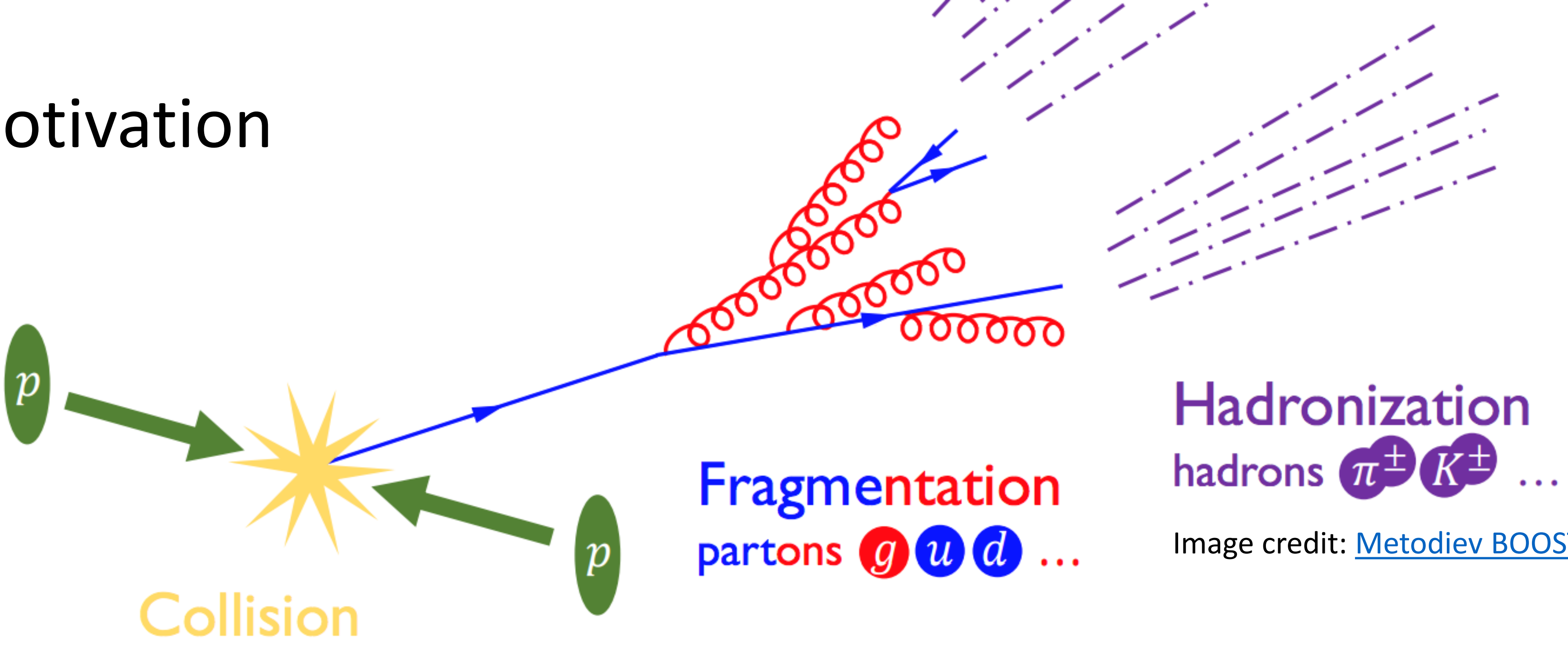


Image credit: [Metodiev BOOST 2019]

Motivation

Jets are proxies for partons

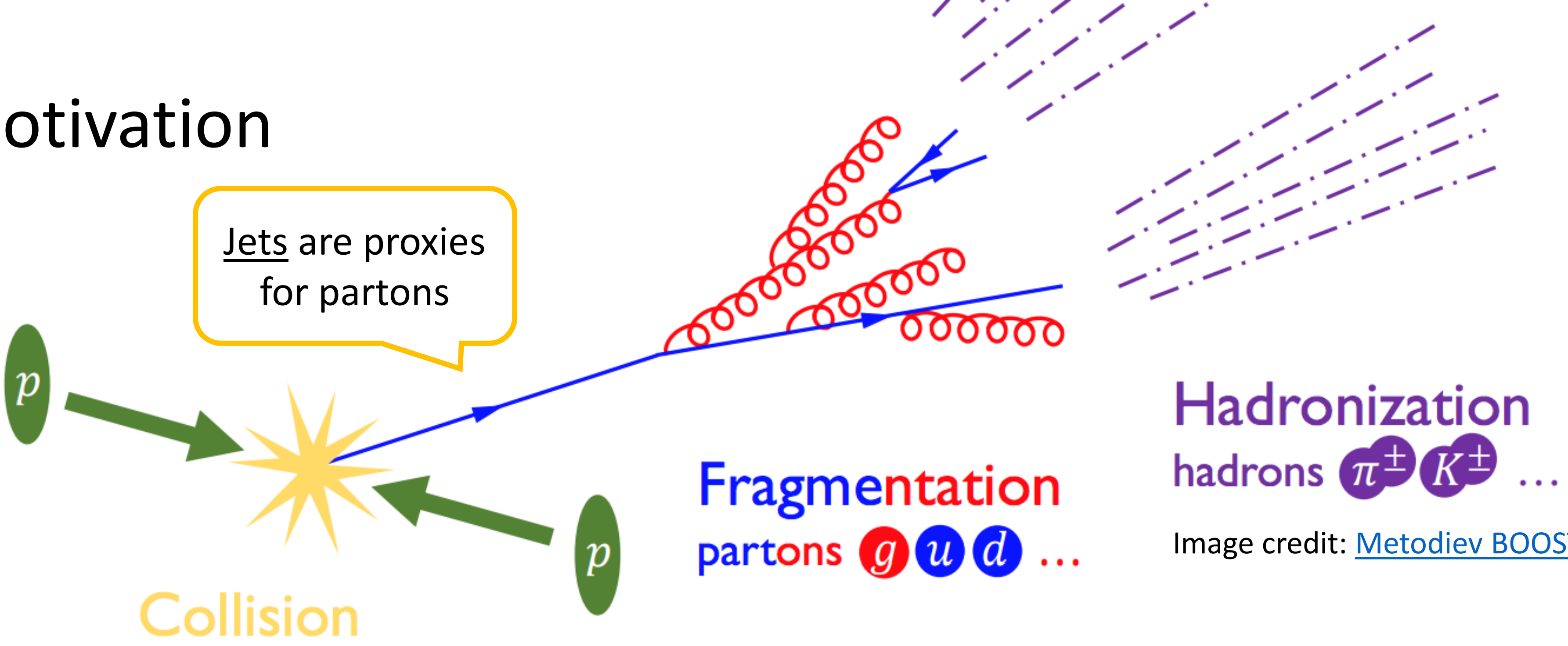


Image credit: [Metodiev BOOST 2019]

Motivation

Jets are proxies for partons

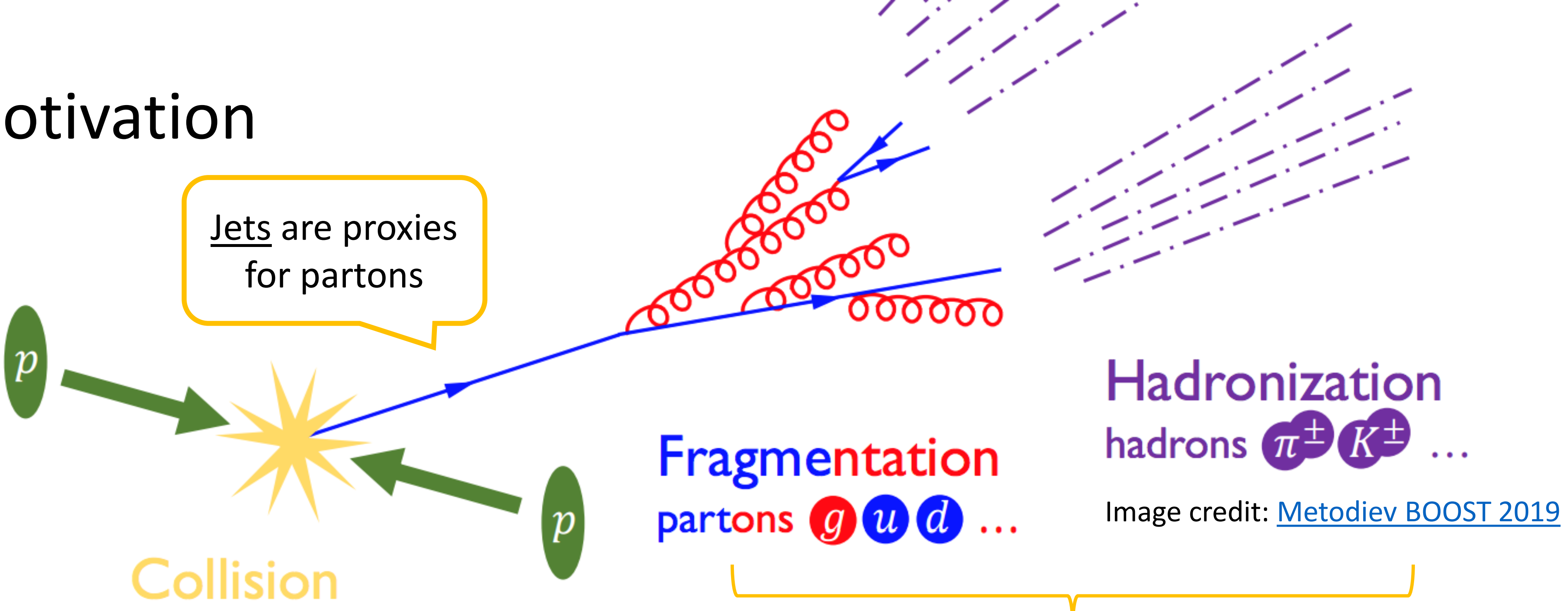


Image credit: [Metodiev BOOST 2019]

Jet substructure measurements can shed light on how fragmentation and hadronization happen

Motivation

STAR

Jets are proxies for partons

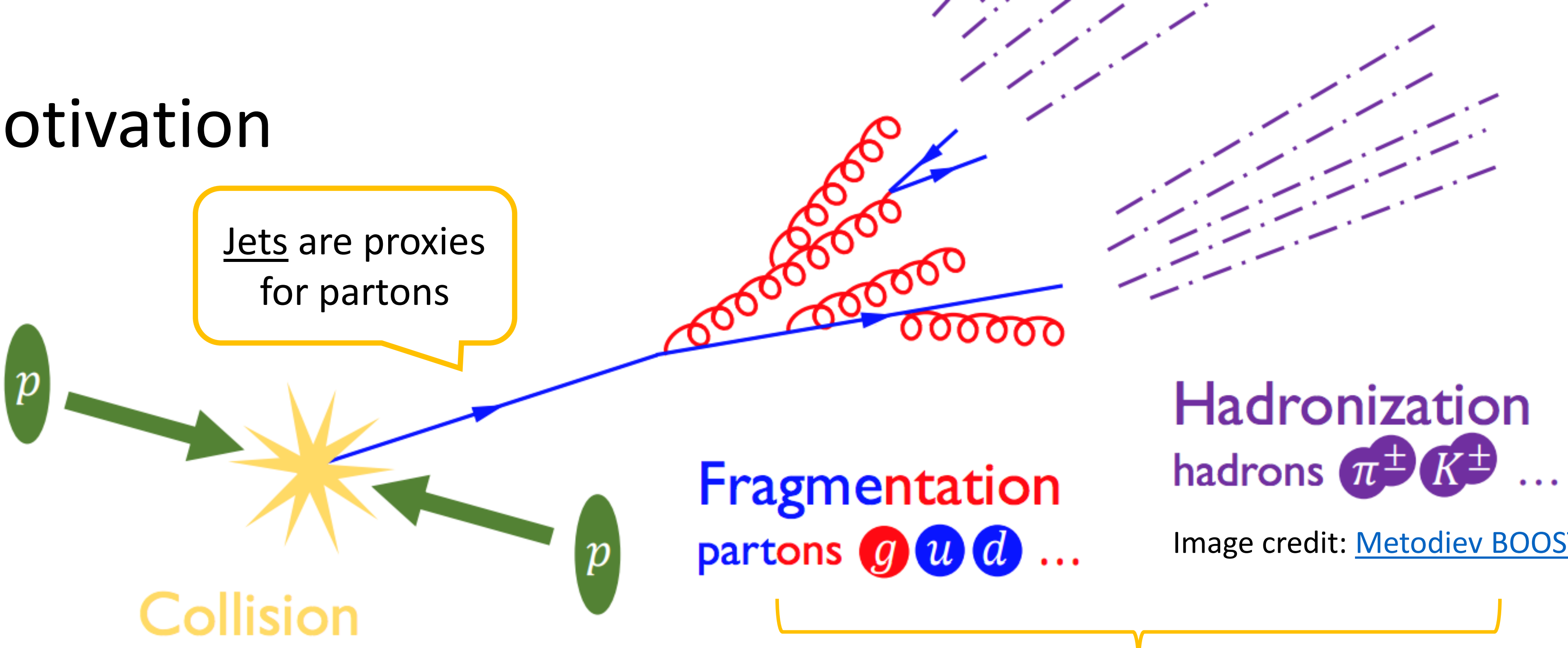
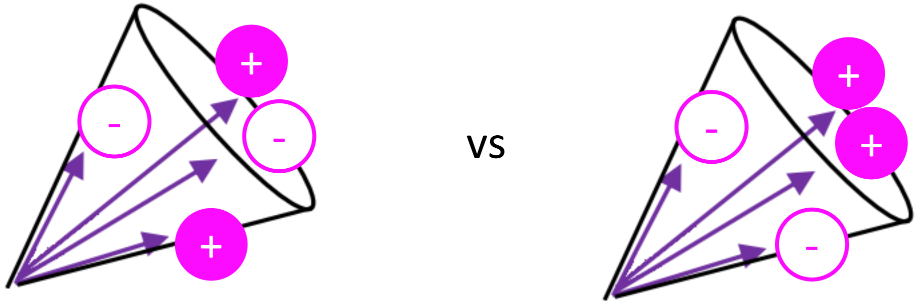


Image credit: [Metodiev BOOST 2019]

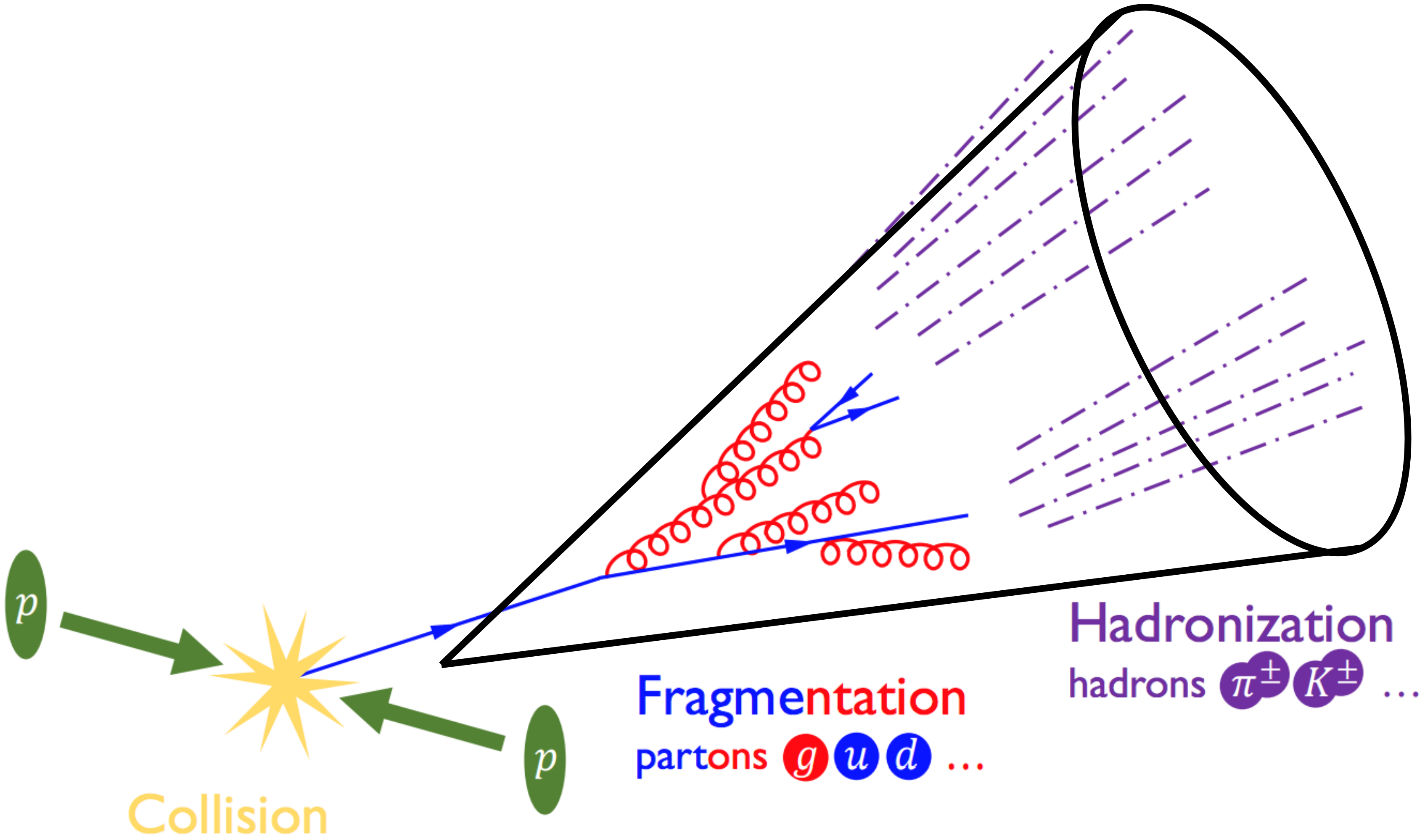
Jet substructure measurements can shed light on how fragmentation and hadronization happen

Multi-dimensional jet substructure measurements help us distinguish different fragmentation patterns

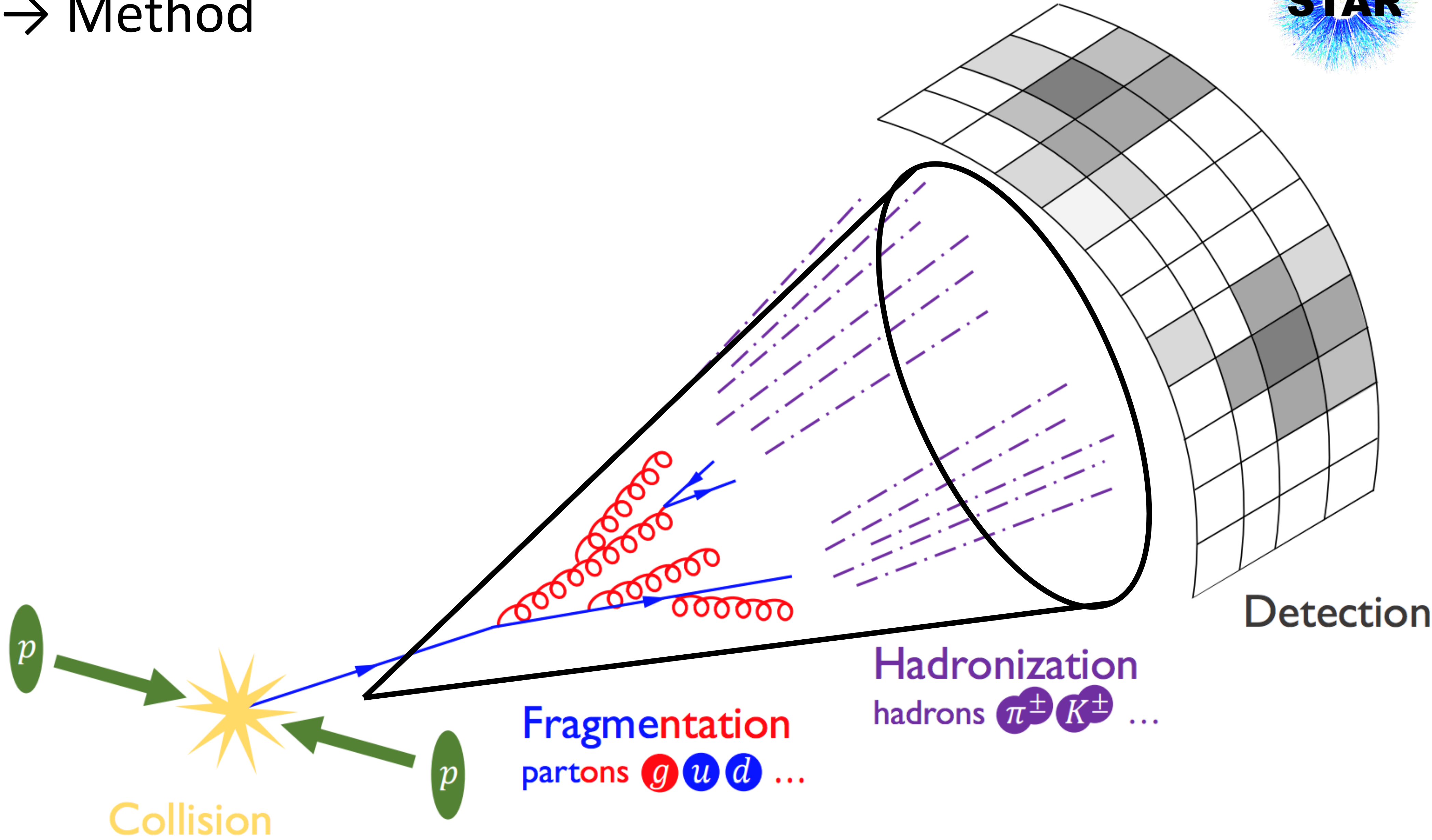


Same jet mass,
Different p_T–weighted jet charge

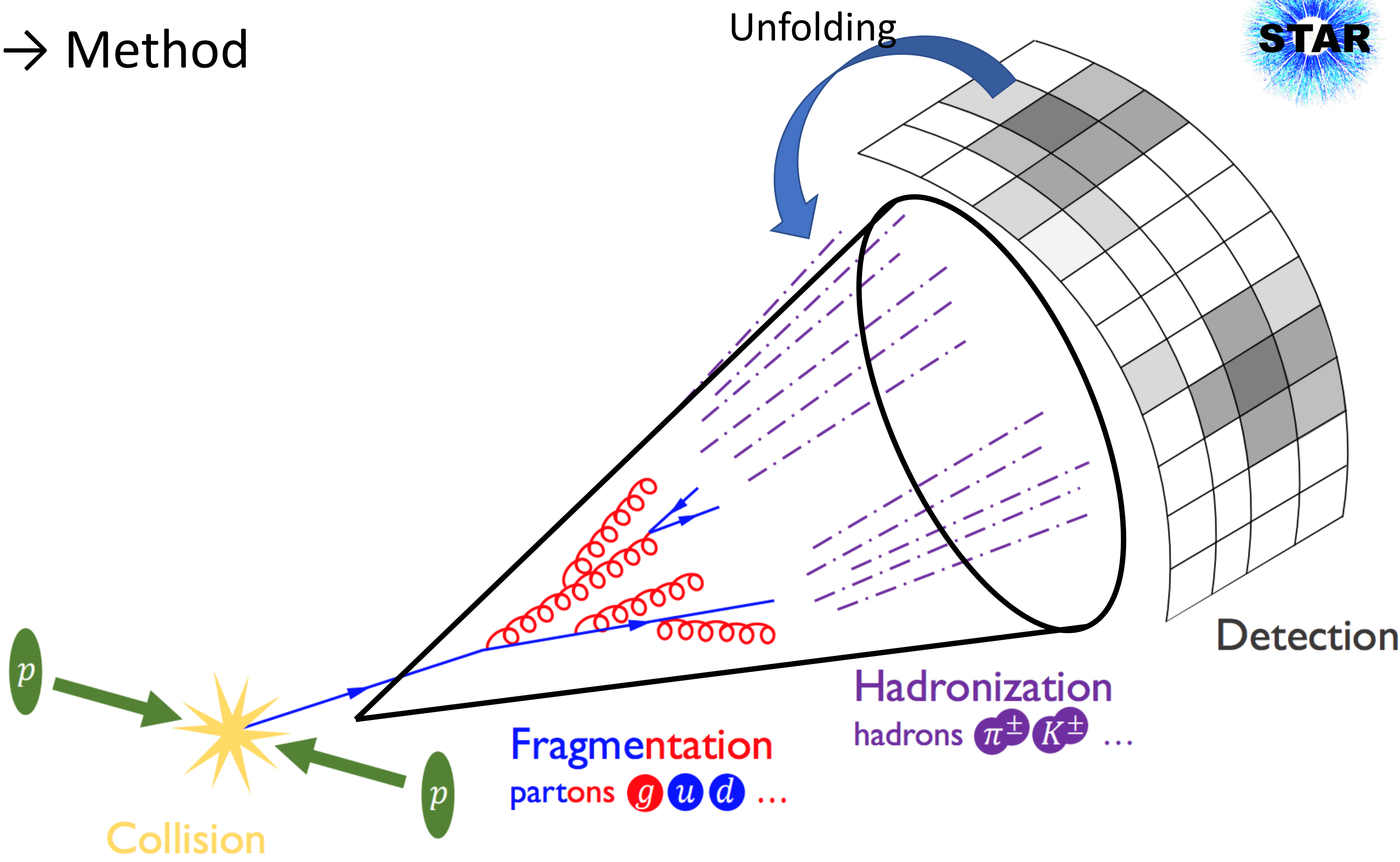
Motivation → Method



Motivation → Method

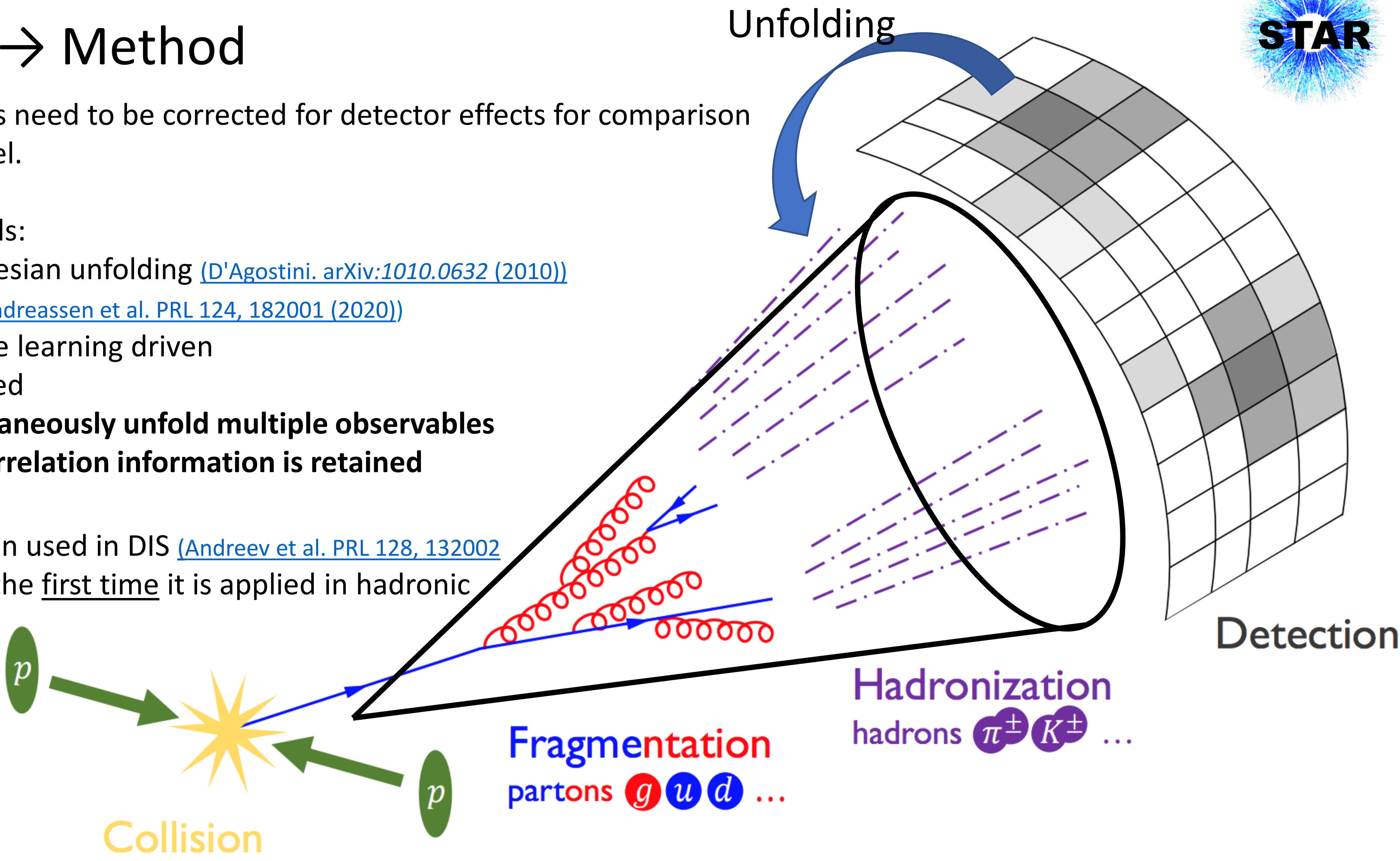


Motivation → Method



Motivation → Method

- Jet measurements need to be corrected for detector effects for comparison with theory/model.
- Unfolding methods:
  - Iterative Bayesian unfolding (D'Agostini. arXiv:*1010.0632* (2010))
  - **MultiFold** (Andreassen et al. PRL 124, 182001 (2020))
    - Machine learning driven
    - Unbinned
    - **Simulataneously unfold multiple observables**
      - ➢ **Correlation information is retained**
- MultiFold has been used in DIS (Andreev et al. PRL 128, 132002 (2022)), but this is the <u>first time</u> it is applied in hadronic collision data.



Observables

- p_{T}: transverse momentum
- $Q^{\kappa} = \frac{1}{(p_{\mathrm{Tjet}})^{\kappa}} \sum_{i \in \mathrm{jet}} q_i \cdot (p_{\mathrm{T}i})^{\kappa}$ → Choose κ=2
- $M = |\Sigma_{i \in \mathrm{jet}}\ p_i| = \sqrt{E^2 - |\vec{p}|^2}$

4-momentum of the constituent i

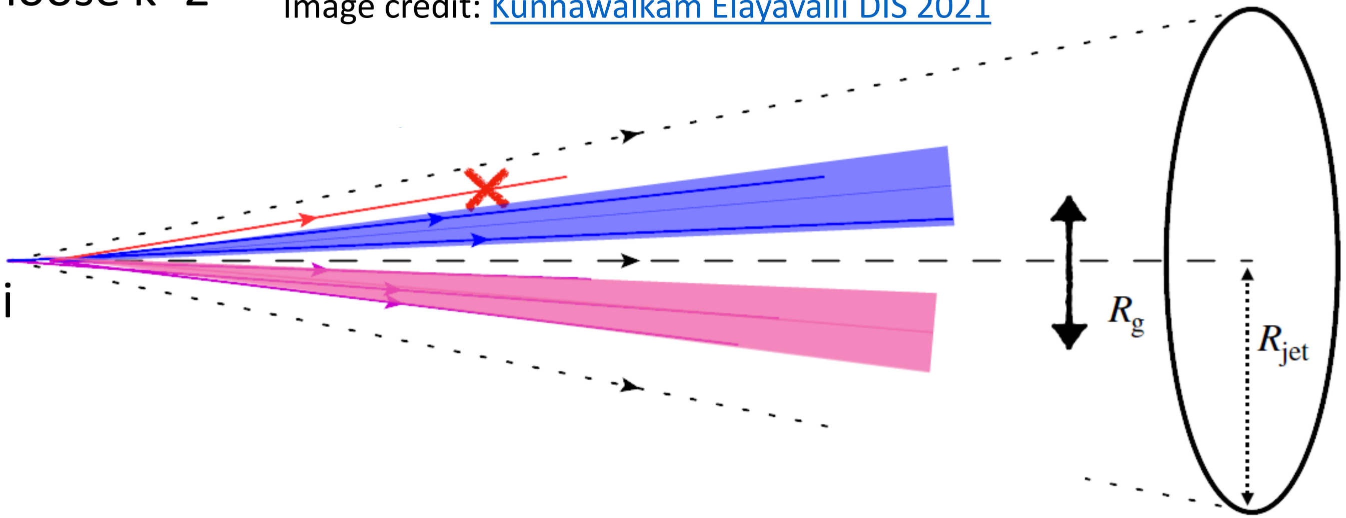
Observables

- p_{T}: transverse momentum
- $Q^{\kappa} = \frac{1}{(p_{\mathrm{T}\text{jet}})^{\kappa}} \sum_{i\in\text{jet}} q_i \cdot (p_{\mathrm{T}i})^{\kappa}$ → Choose κ=2
- $M = |\Sigma_{i\in\text{jet}}\ p_i| = \sqrt{E^2 - |\vec{p}|^2}$

4-momentum of the constituent i

SoftDrop grooming

Larkoski, et al. JHEP 2014, 146 (2014).
Dasgupta et al. JHEP 2013, 29 (2013).
Image credit: Kunnawalkam Elayavalli DIS 2021



Require subjet momentum fraction to pass

$$z_g = \frac{\min(p_{T,1}, p_{T,2})}{p_{T,1} + p_{T,2}} > z_{\text{cut}}(R_g/R_{\text{jet}})^{\beta}$$

$z_{\text{cut}} = 0.1$
$\beta = 0$

Observables

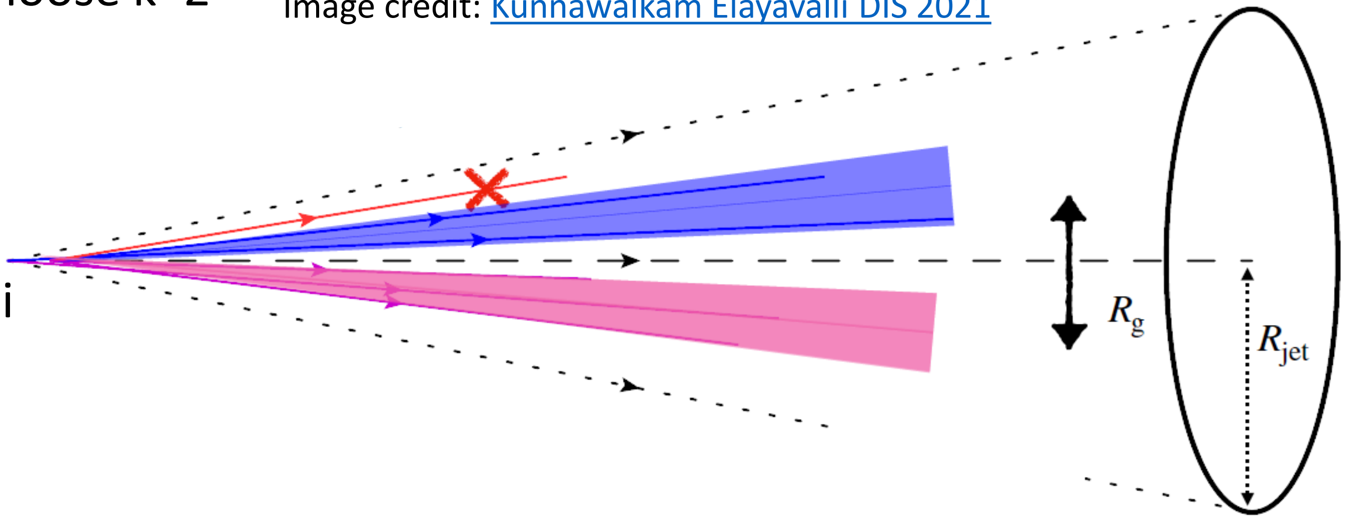
- p_{T}: transverse momentum
- $Q^{\kappa} = \frac{1}{(p_{\mathrm{T jet}})^{\kappa}} \sum_{i \in \mathrm{jet}} q_i \cdot (p_{\mathrm{T}i})^{\kappa}$ → Choose κ=2
- $M = |\Sigma_{i \in \mathrm{jet}}\ p_i| = \sqrt{E^2 - |\vec{p}|^2}$

4-momentum of the constituent i

- R_g: groomed jet radius
- z_g: shared momentum fraction
- M_g: groomed jet mass

SoftDrop grooming

Larkoski, et al. JHEP 2014, 146 (2014).
Dasgupta et al. JHEP 2013, 29 (2013).
Image credit: Kunnawalkam Elayavalli DIS 2021



Require subjet momentum fraction to pass

$$z_g = \frac{\min(p_{T,1}, p_{T,2})}{p_{T,1} + p_{T,2}} > z_{\mathrm{cut}}(R_g/R_{\mathrm{jet}})^{\beta}$$

$z_{\mathrm{cut}} = 0.1$
$\beta = 0$

Observables

- p_{T}: transverse momentum
- $Q^{\kappa} = \frac{1}{(p_{\mathrm{Tjet}})^{\kappa}} \sum_{i \in \mathrm{jet}} q_i \cdot (p_{\mathrm{T}i})^{\kappa}$ → Choose κ=2
- $M = |\Sigma_{i \in \mathrm{jet}}\ p_i| = \sqrt{E^2 - |\vec{p}|^2}$

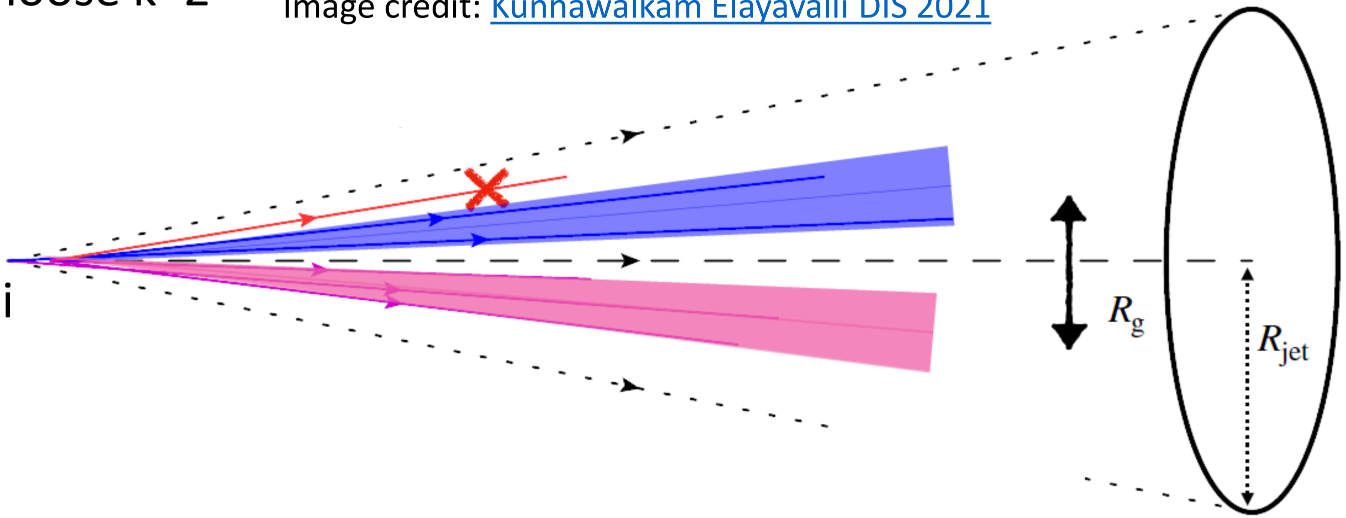
4-momentum of the constituent i

- R_g: groomed jet radius
- z_g: shared momentum fraction
- M_g: groomed jet mass

All 6 observables are simultaneously unfolded in an unbinned way!

SoftDrop grooming

Larkoski, et al. JHEP 2014, 146 (2014).
Dasgupta et al. JHEP 2013, 29 (2013).
Image credit: Kunnawalkam Elayavalli DIS 2021



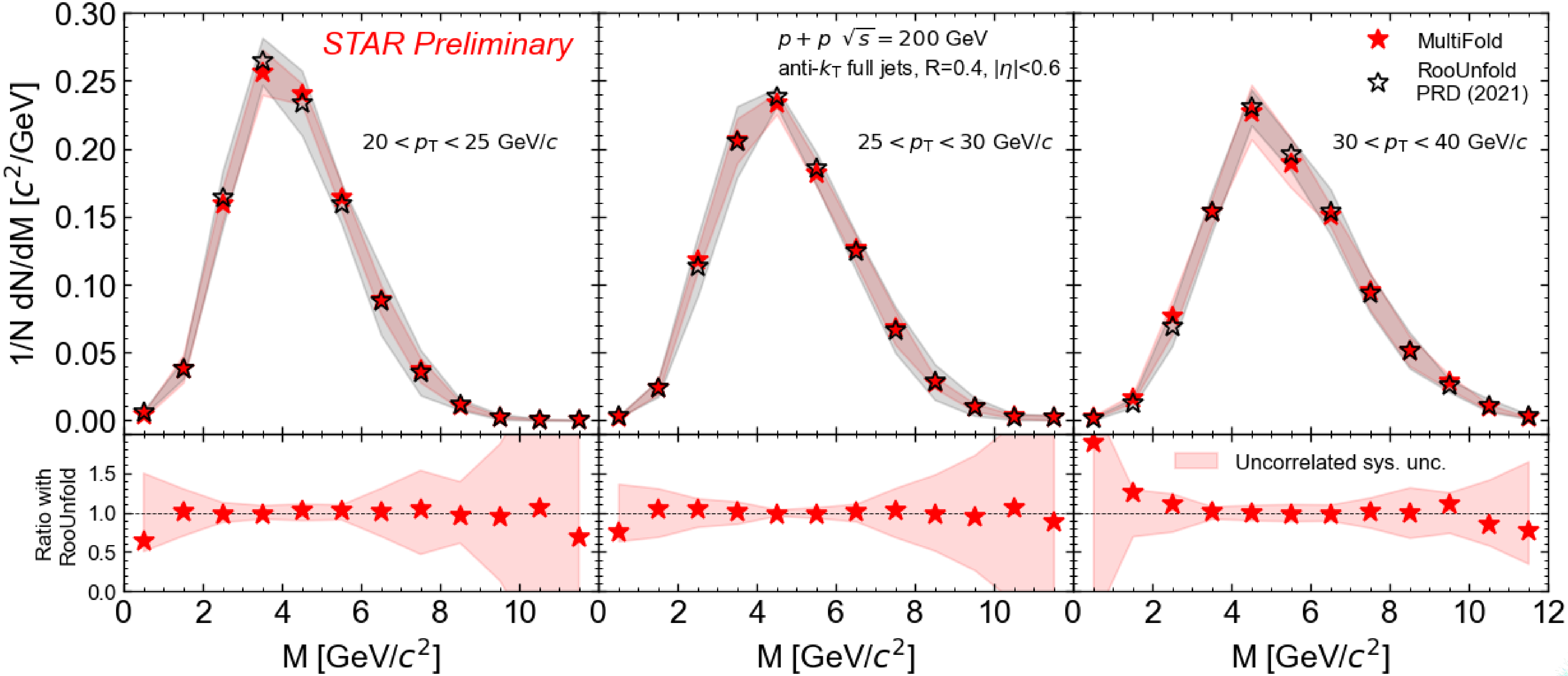
Require subjet momentum fraction to pass

$$z_g = \frac{\min(p_{T,1}, p_{T,2})}{p_{T,1} + p_{T,2}} > z_{\mathrm{cut}}(R_g/R_{\mathrm{jet}})^{\beta} \qquad z_{\mathrm{cut}} = 0.1 \quad \beta = 0$$

Fully corrected jet M

$$M = \left|\sum_{i\in \text{jet}} p_i\right| = \sqrt{E^2 - \boldsymbol{p}^2},$$

MultiFold result agrees with **RooUnfold** result (STAR Collaboration. PRD 104, 052007(2021)) HEPData



… but **MultiFold** also gives us correlation between observables!

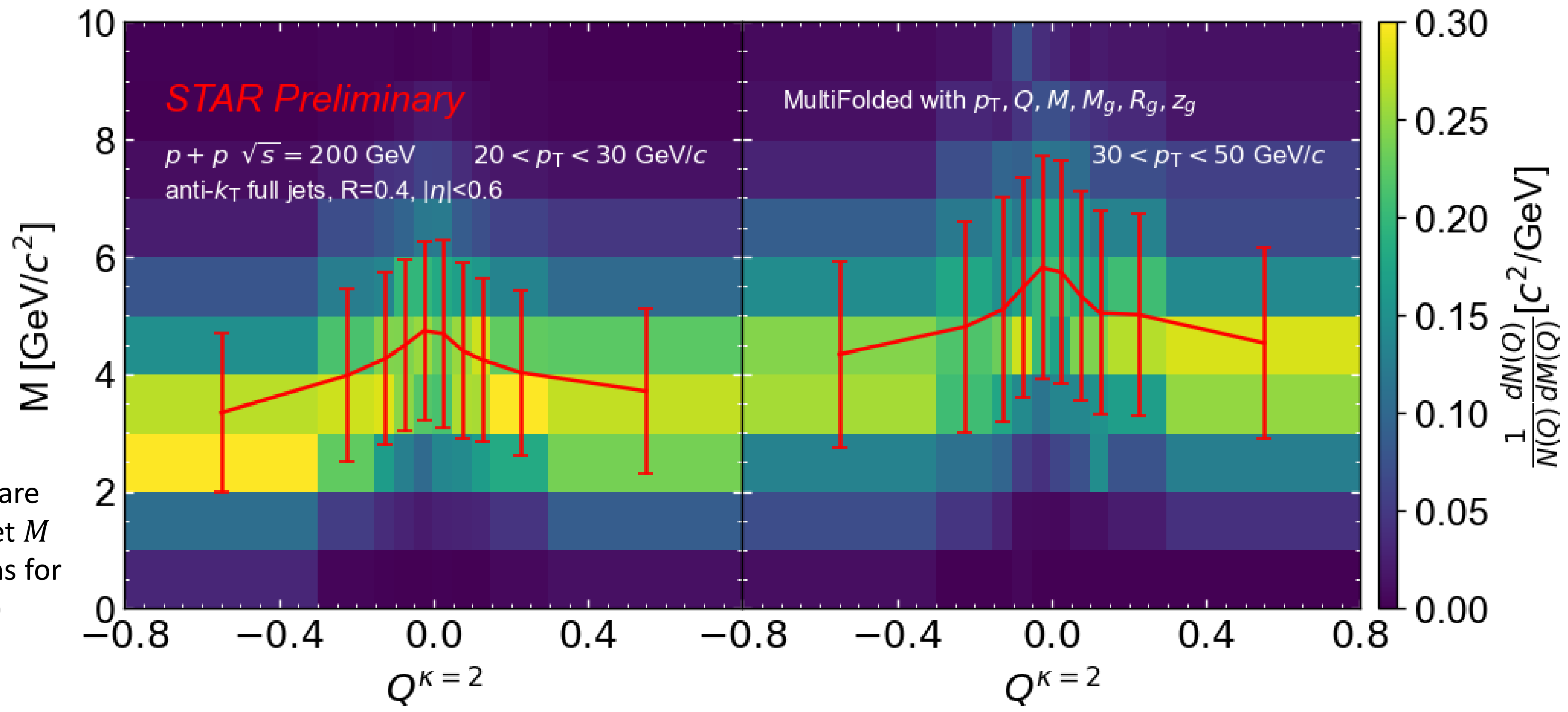
See backup slides for a breakdown of systematics.

Fully corrected jet M vs Q vs p_T

$$Q_J = \frac{1}{(p_{T,J})^\kappa} \sum_{i \in Tracks} q_i \times (p_{T,i})^\kappa \qquad M = \left|\sum_{i\in \text{jet}} p_i\right| = \sqrt{E^2 - \mathbf{p}^2},$$

(normalization is done per Q bin)

(Error bars are widths of jet M distributions for each Q bin)



Jet M increases with increasing jet p_T → Higher p_T means larger phase space for radiation

Jet M increases with decreasing jet $|Q|$ → High p_T track contributes more to jet $|Q|$

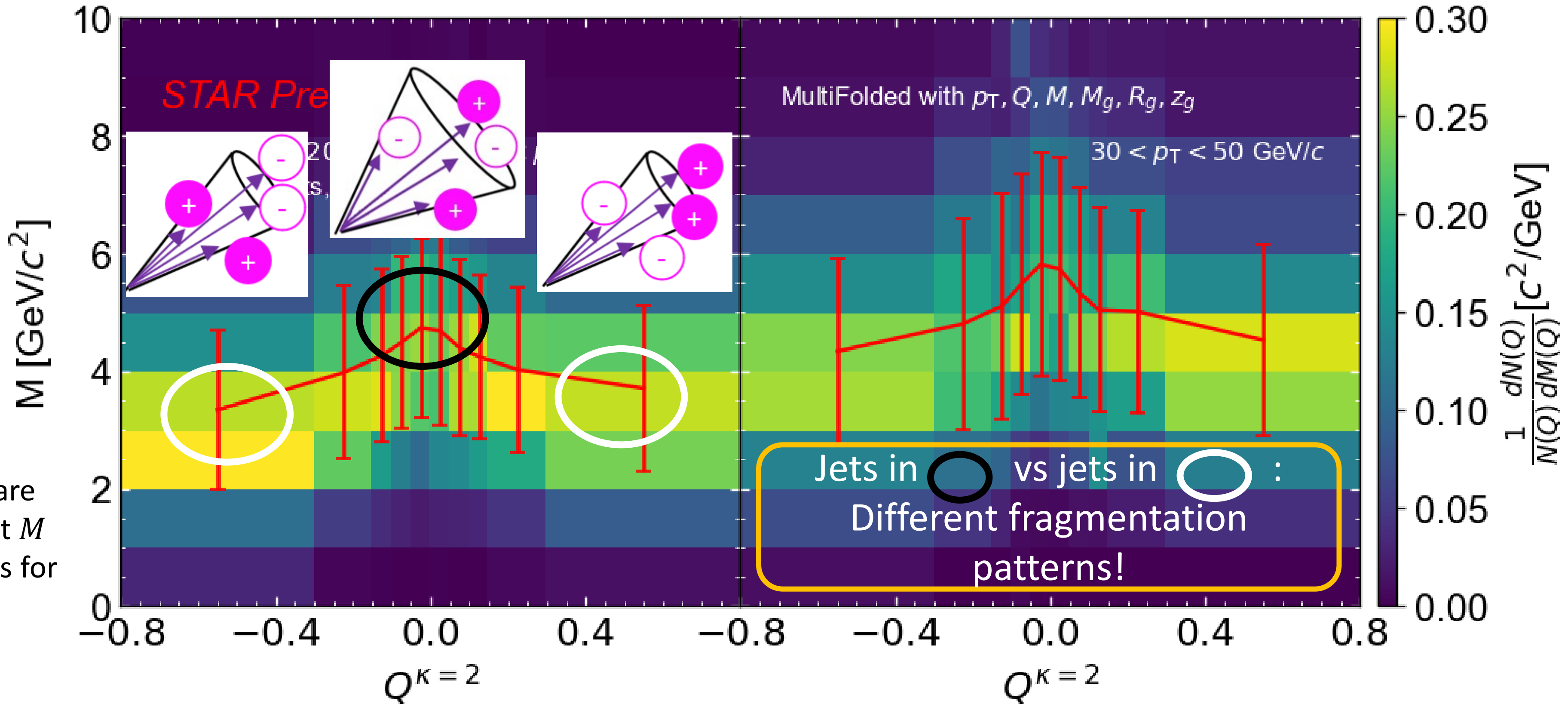
→ Wider jets tend to have lower $|Q|$

Fully corrected jet M vs Q vs p_T

$$Q_J = \frac{1}{(p_{T\,J})^\kappa} \sum_{i \in Tracks} q_i \times (p_{T,i})^\kappa \qquad M = \left|\sum_{i\in \mathrm{jet}} p_i\right| = \sqrt{E^2 - \boldsymbol{p}^2},$$

(normalization is done per Q bin)

(Error bars are widths of jet M distributions for each Q bin)



Jet M increases with increasing jet p_T → Higher p_T means larger phase space for radiation

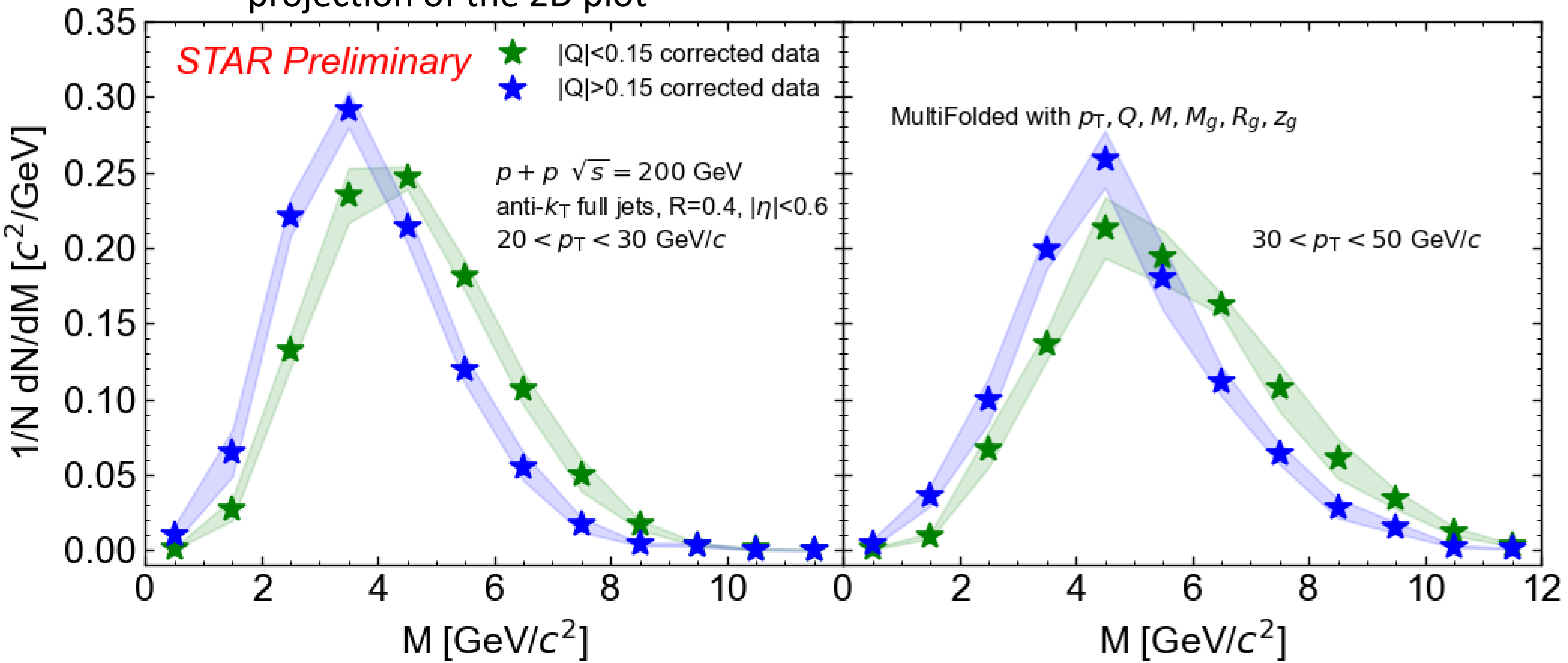
Jet M increases with decreasing jet $|Q|$ → High p_T track contributes more to jet $|Q|$

→ Wider jets tend to have lower $|Q|$

Fully corrected jet M vs Q vs p_T

$$Q_J = \frac{1}{(p_{\mathrm{T}J})^{\kappa}} \sum_{i \in Tracks} q_i \times (p_{\mathrm{T,i}})^{\kappa} \quad M = \left|\sum_{i\in \mathrm{jet}} p_i\right| = \sqrt{E^2 - \boldsymbol{p}^2},$$

projection of the 2D plot

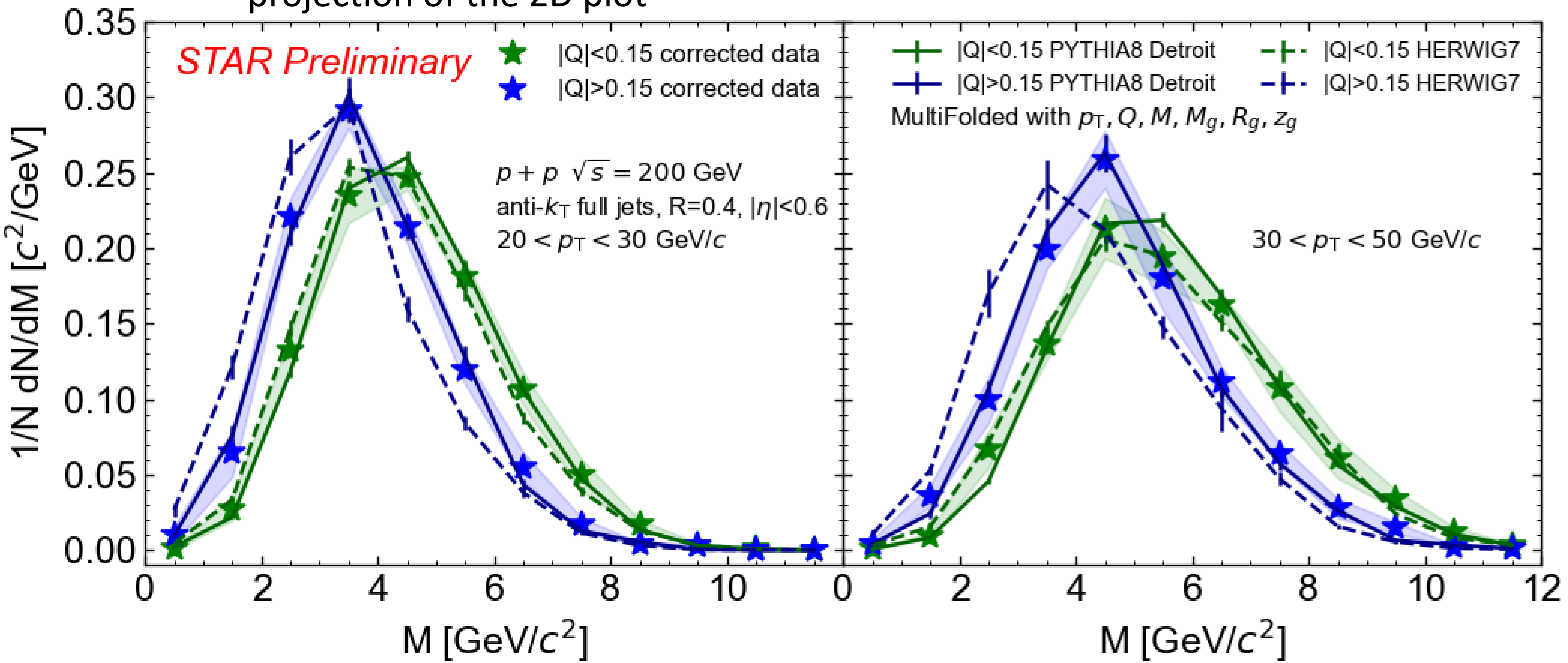


Mean and spread of jet M distribution both increase with decreasing $|Q|$

Fully corrected jet M vs Q vs p_T

$$Q_J = \frac{1}{(p_{\mathrm{T}\,J})^{\kappa}} \sum_{i \in Tracks} q_i \times (p_{\mathrm{T,i}})^{\kappa} \qquad M = \left|\sum_{i\in \mathrm{jet}} p_i\right| = \sqrt{E^2 - \mathbf{p}^2},$$

projection of the 2D plot



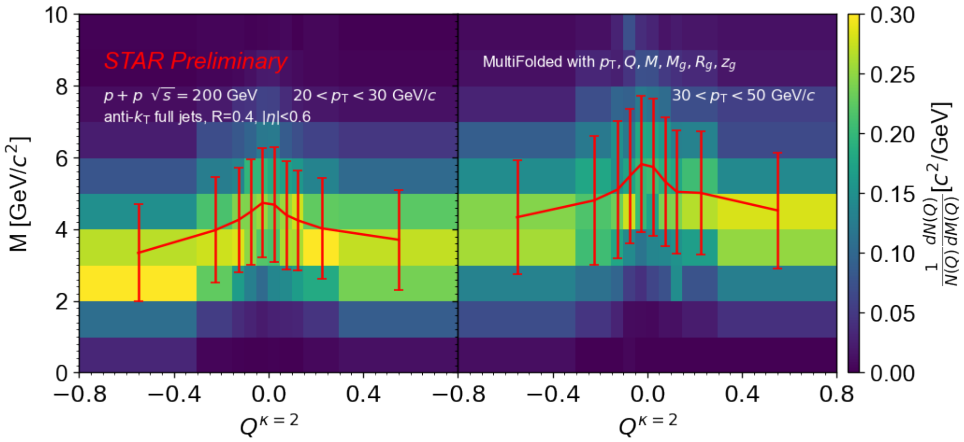
PYTHIA6 Perugia 2012 STAR tune: Skands, PRD 82, 074018 (2010)
PYTHIA8 Detroit tune: Aguilar et al. PRD 105, 016011(2022)
HERWIG7: Bellm, et al. PRC 76, 1-8 (2016)

Mean and spread of jet M distribution both increase with decreasing $|Q|$

PYTHIA8 Detroit tune: Describes jet M vs $|Q|$ well

HERWIG7: Underpredicts jet M for large $|Q|$ significantly

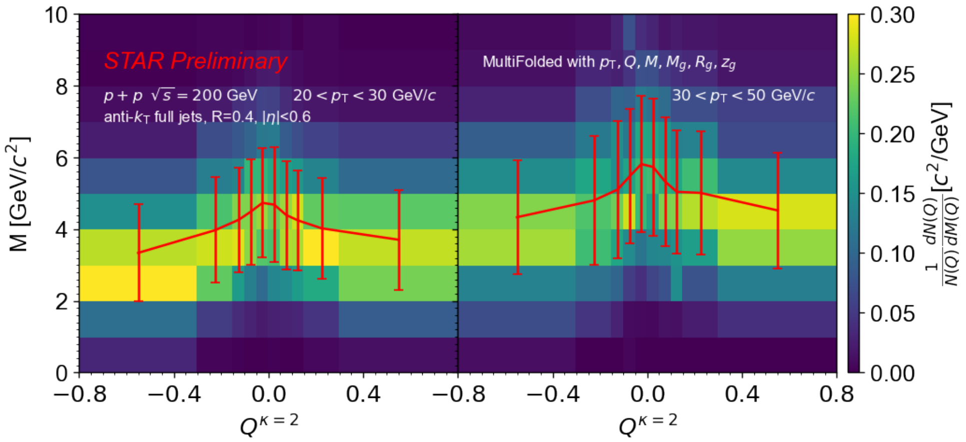
Summary and outlook



- First measurement in pp that uses machine learning based method for unfolding
  - **Multi-dimensional** and unbinned
  - Nice **agreement with RooUnfold**

- Fully-corrected measurement of jet M vs Q vs p_T in $\sqrt{s}$ = 200 GeV pp collisions.
  - Jet M increases with increasing p_T ; jet M increases with decreasing $|Q|$.
  - **PYTHIA8 Detroit tune** describes the data well; **HERWIG7** underpredicts jet M for large $|Q|$.

Summary and outlook



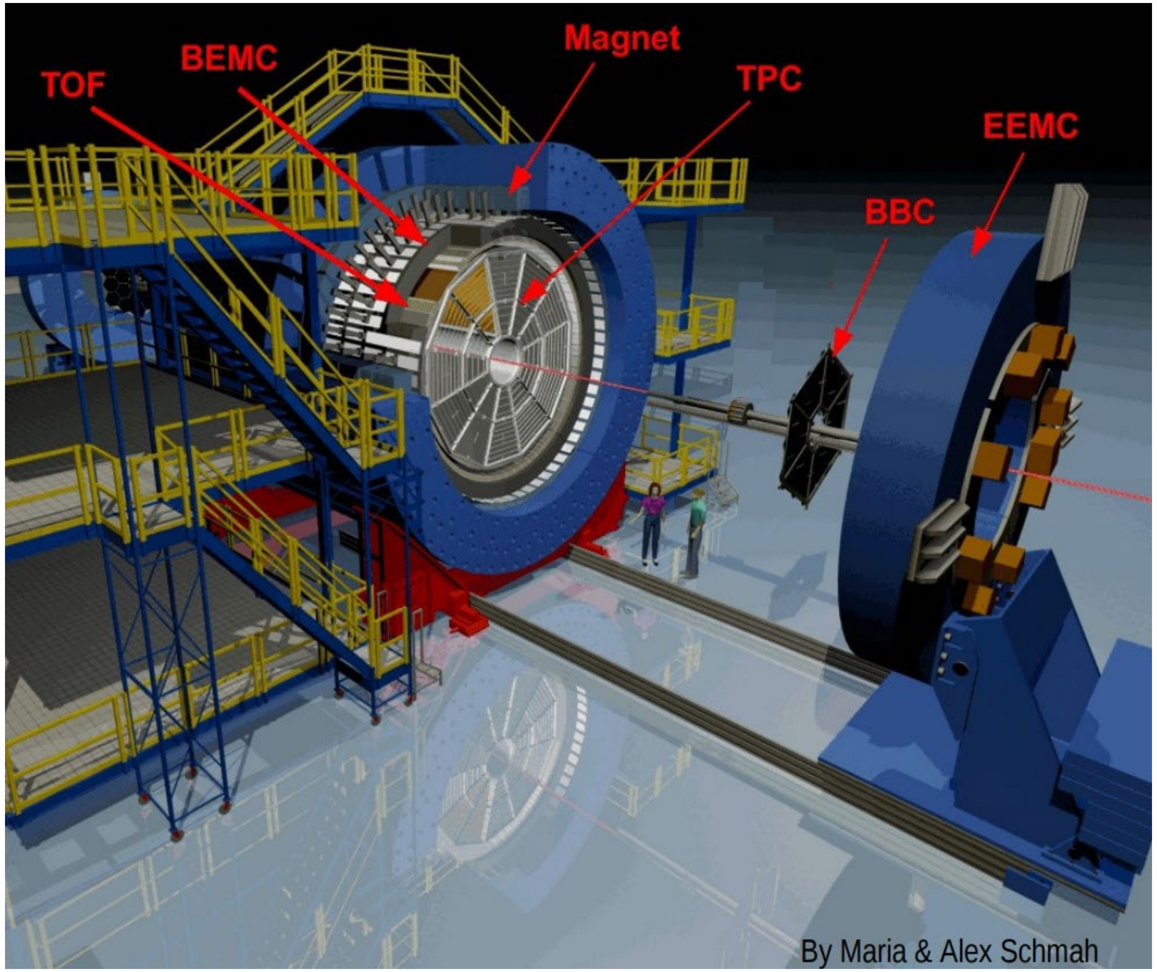
- <u>First measurement</u> in pp that uses machine learning based method for unfolding
  - **Multi-dimensional** and unbinned
  - Nice **agreement with RooUnfold**

- Fully-corrected measurement of jet M vs Q vs p_T in $\sqrt{s}$ = 200 GeV pp collisions.
  - Jet M increases with increasing p_T ; jet M increases with decreasing $|Q|$.
  - **PYTHIA8 Detroit tune** describes the data well; **HERWIG7** underpredicts jet M for large $|Q|$.

- Future directions
  - Selecting jets with different **fragmentation** patterns → study **hadronization**
  - 6-dimensional jet information → separate **quark vs gluon** jets

Backup

Jet reconstruction at STAR



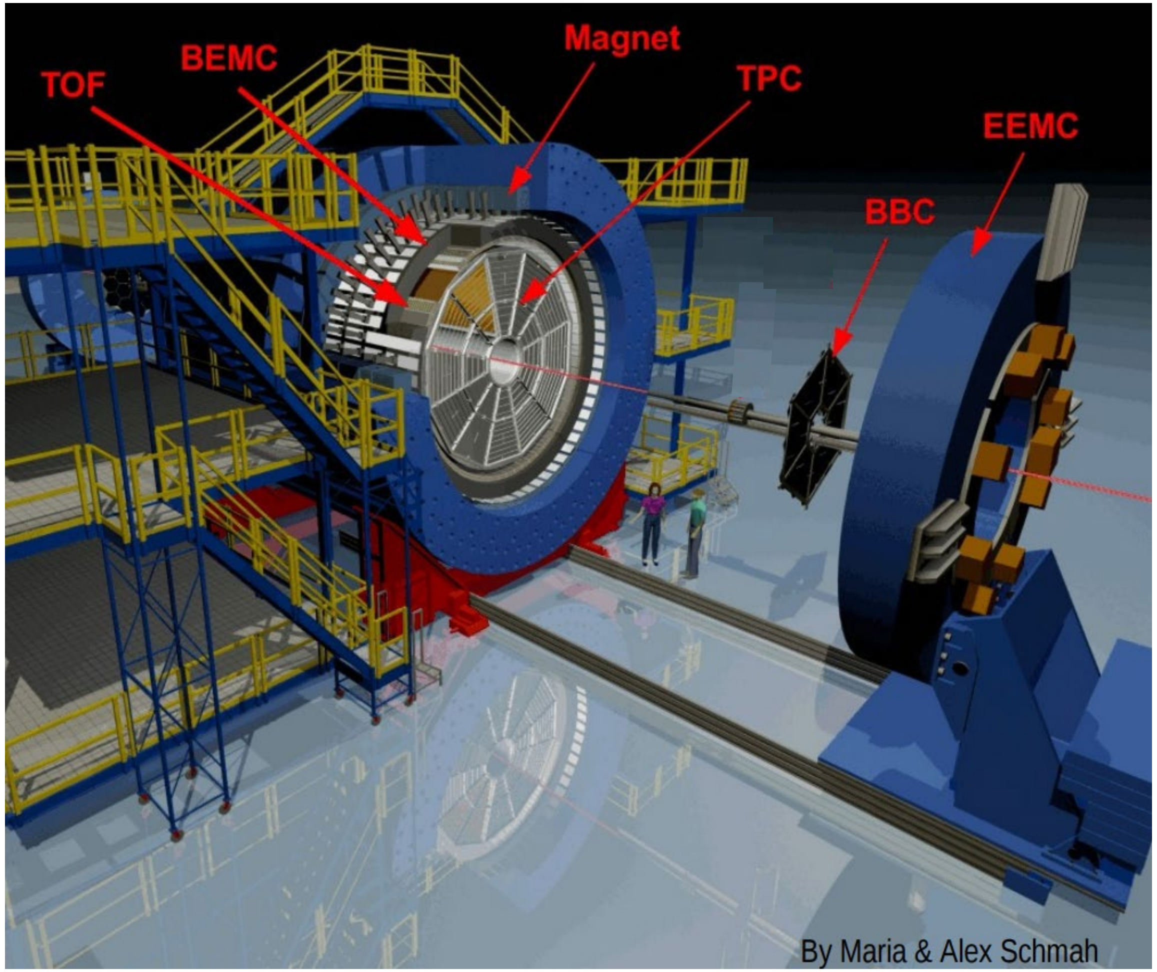
Jet reconstruction at STAR

Important subdetectors for **pp $\sqrt{s}$ = 200 GeV** collisions data-taking during 2012 RHIC run

- **TPC** (Time Projection Chamber)
  - For **charged** particle track reconstruction
  - $|\eta| < 1$, full azimuthal coverage

- **BEMC** (Barrel ElectroMagnetic Calorimeter)
  - For **neutral** energy measurement and triggering
  - $|\eta| < 1$, full azimuthal coverage

➢ Reconstruct anti-k_T **full jets**
  - Jet resolution parameter **R=0.4**
  - $|\eta_{jet}| < 0.6$



Jet reconstruction at STAR

Important subdetectors for **pp** $\sqrt{s}$ **= 200 GeV** collisions data-taking during 2012 RHIC run

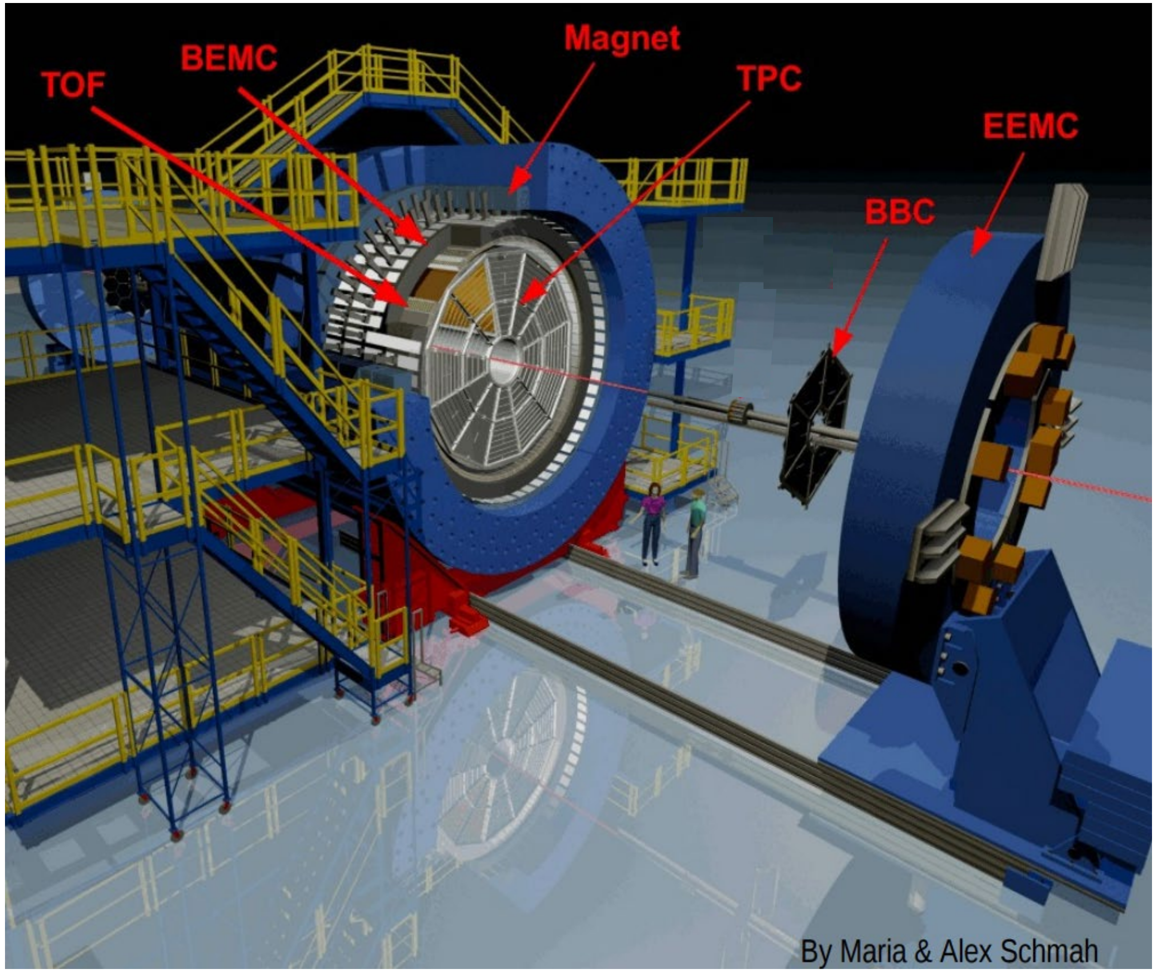
- **TPC** (Time Projection Chamber)
  - For **charged** particle track reconstruction
  - $|\eta| < 1$, full azimuthal coverage

- **BEMC** (Barrel ElectroMagnetic Calorimeter)
  - For **neutral** energy measurement and triggering
  - $|\eta| < 1$, full azimuthal coverage

➢ Reconstruct anti-k_T **full jets**
  - Jet resolution parameter **R=0.4**
  - $|\eta_{jet}| < 0.6$

Additional selections

- Tracks (Towers): $0.2 < p_{\mathrm{T}}(E_{\mathrm{T}}) < 30$ GeV
- Jets
  - $p_{\mathrm{T}} > 15$ GeV/c, $M > 1$ GeV/c^2, neutral p_{T} fraction < 0.9
  - Passes SoftDrop with z_{cut}= 0.1 and β = 0



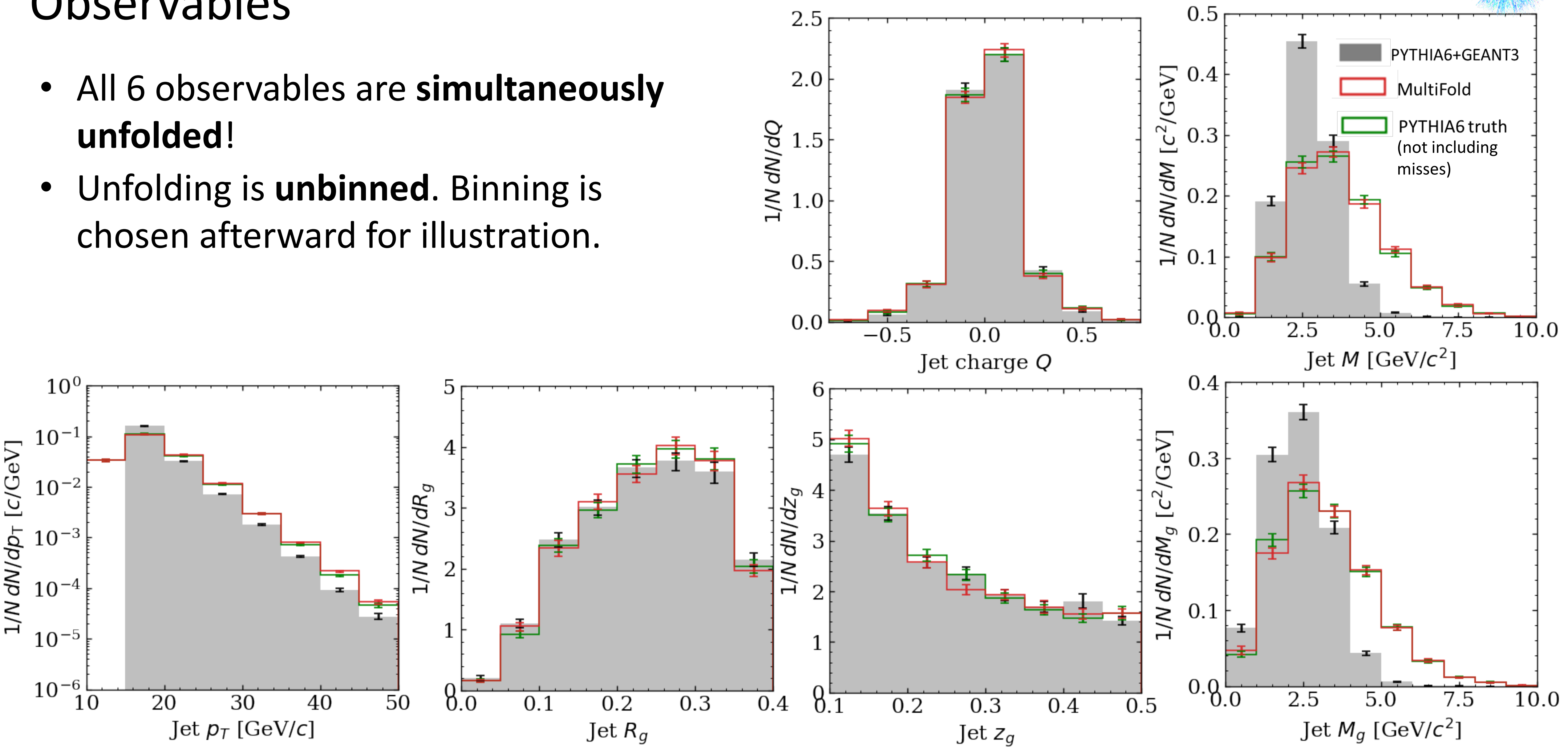
Observables

- All 6 observables are **simultaneously unfolded**!
- Unfolding is **unbinned**. Binning is chosen afterward for illustration.

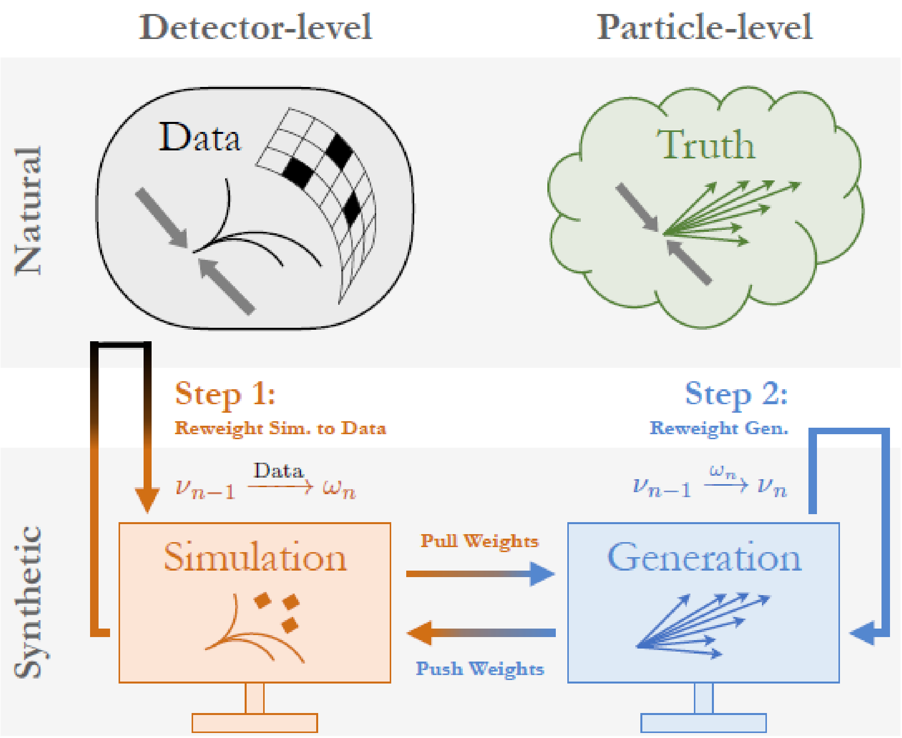
Observables

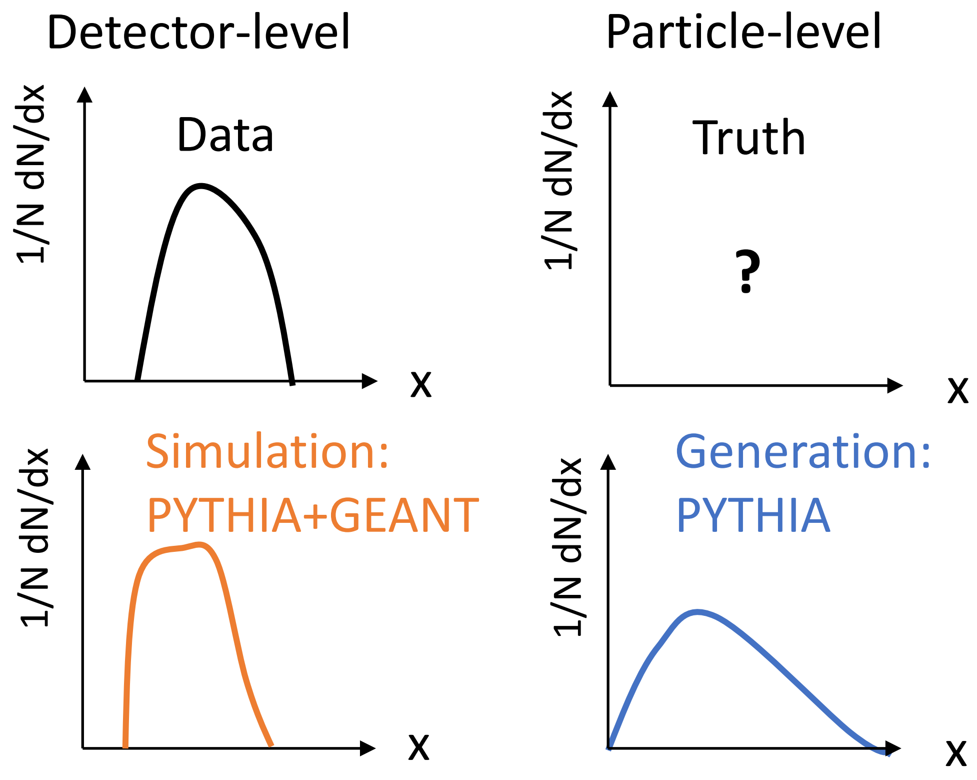
- All 6 observables are **simultaneously unfolded**!
- Unfolding is **unbinned**. Binning is chosen afterward for illustration.

pp $\sqrt{s}$ = 200 GeV

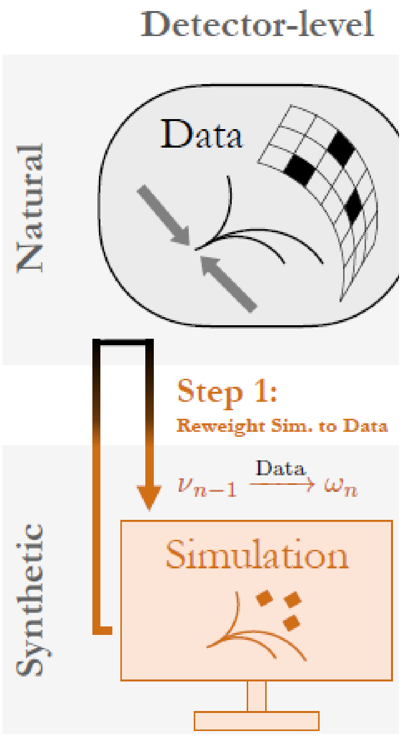


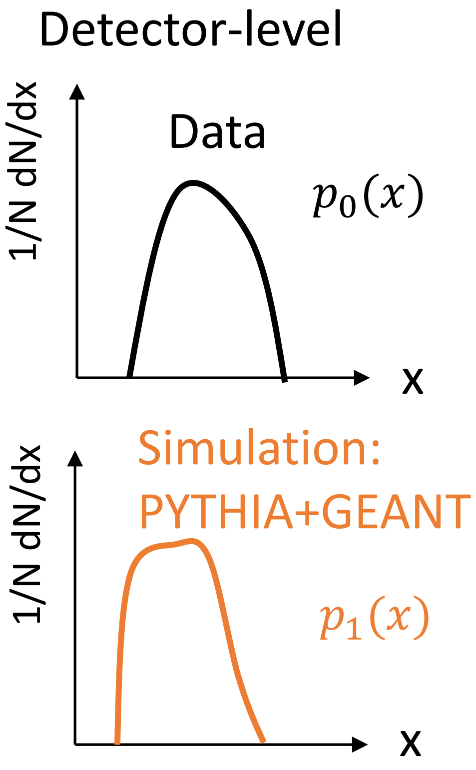
Method


Detector-level
Particle-level
Natural
Data
Truth
Step 1:
Reweight Sim. to Data
Step 2:
Reweight Gen.
Synthetic
$\nu_{n-1} \xrightarrow{\text{Data}} \omega_n$
$\nu_{n-1} \xrightarrow{\omega_n} \nu_n$
Simulation
Pull Weights
Push Weights
Generation


Detector-level
Particle-level
1/N dN/dx
Data
x
1/N dN/dx
Truth
?
x
1/N dN/dx
Simulation:
PYTHIA+GEANT
x
1/N dN/dx
Generation:
PYTHIA
x

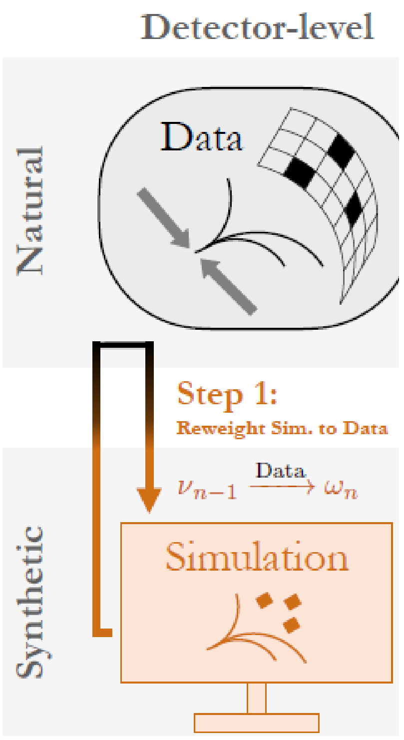
Method: machine learning



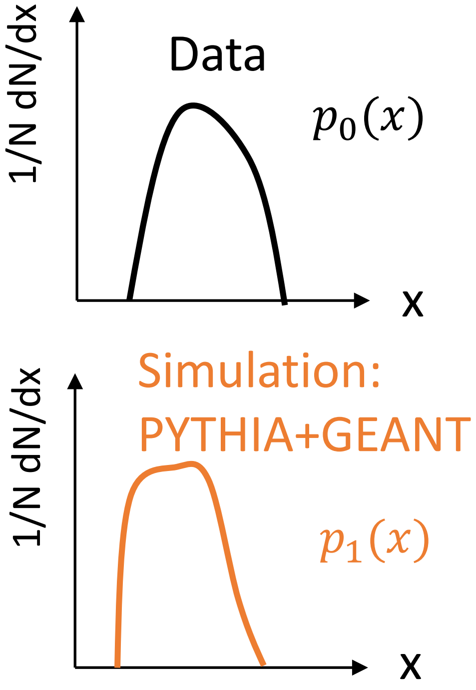


Where does the machine learning part come in?

Method: machine learning



Detector-level



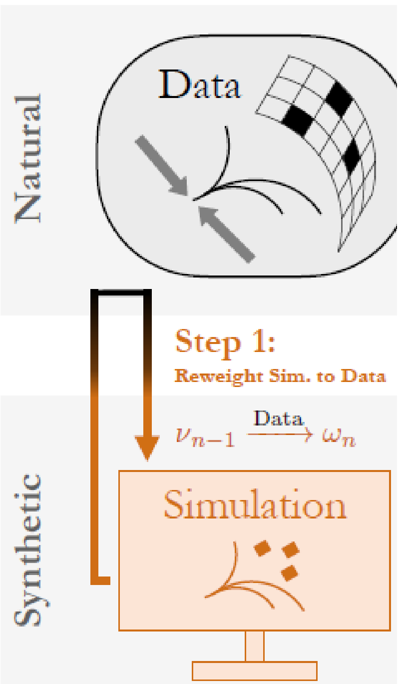
E.g., Iteration 1, step 1:

Weights: $w(x) = p_0(x)/p_1(x)$

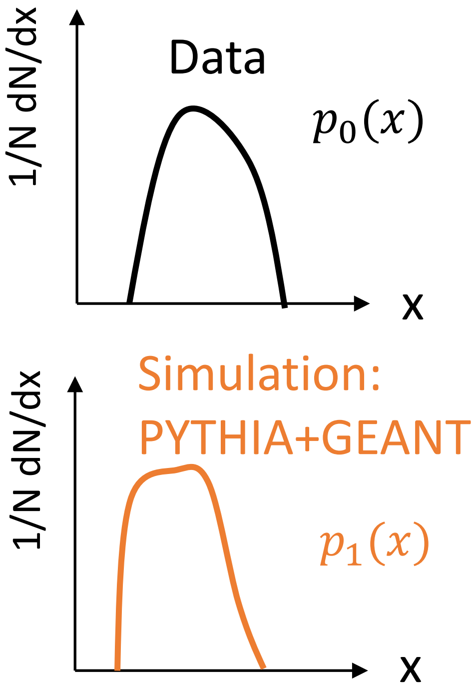
Where does the machine learning part come in?

Method: machine learning

Detector-level



Detector-level

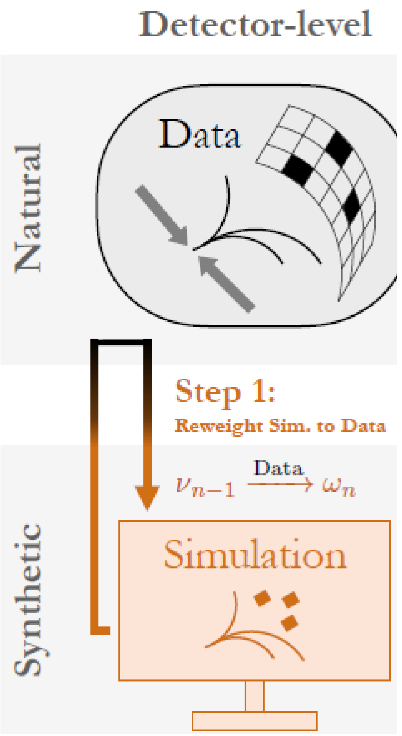


E.g., Iteration 1, step 1:

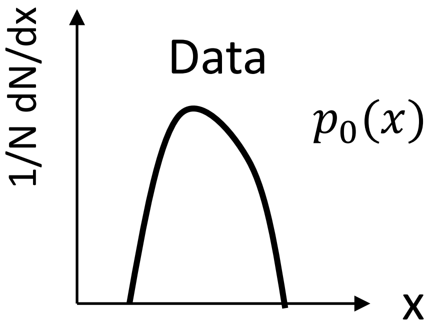
Weights: $w(x) = p_0(x)/p_1(x)$ Ok for 1D

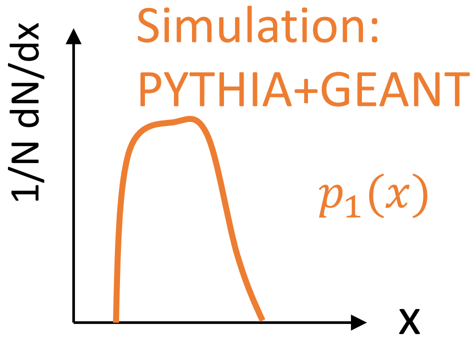
Where does the machine learning part come in?

Method: machine learning



Detector-level





Where does the machine learning part come in?

E.g., Iteration 1, step 1:

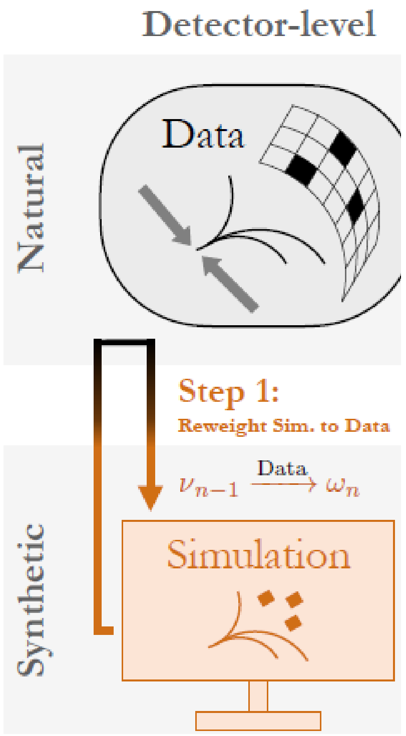
Weights: $w(x) = p_0(x)/p_1(x)$ Ok for 1D

$$\approx f(x)/(1 - f(x))$$

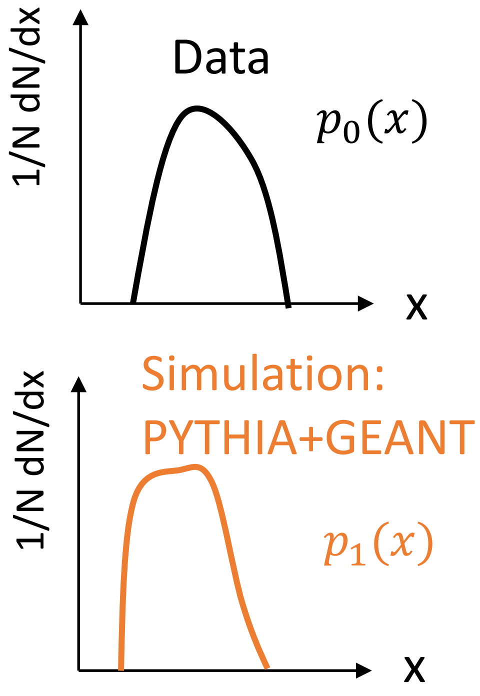
(Andreassen and Nachman PRD 101, 091901 (2020))

where f(x) is a neural network and trained with the binary cross-entropy loss function

Method: machine learning



Detector-level



E.g., Iteration 1, step 1:

Weights: $w(x) = p_0(x)/p_1(x)$ Ok for 1D

$$\approx f(x)/(1 - f(x))$$

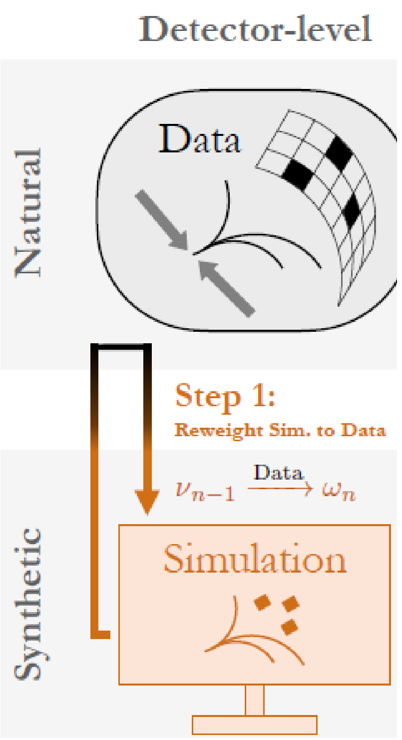
(Andreassen and Nachman PRD 101, 091901 (2020))

where f(x) is a neural network and trained with the binary cross-entropy loss function

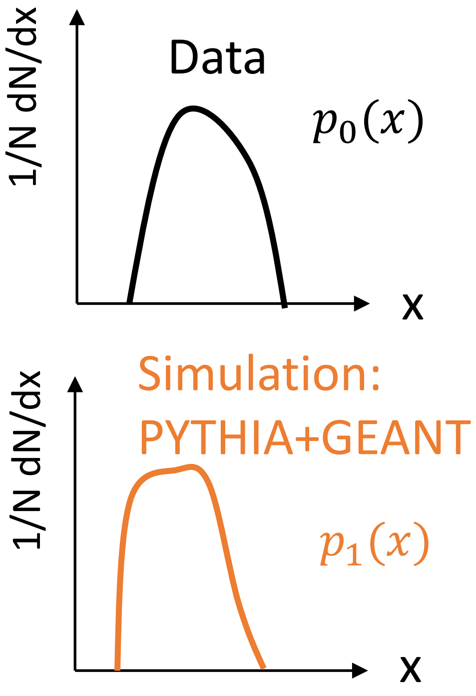
to distinguish jets coming from data vs from simulation

Where does the machine learning part come in?

Method: machine learning



Detector-level



E.g., Iteration 1, step 1:

Weights: $w(x) = p_0(x)/p_1(x)$ Ok for 1D

$$\approx f(x)/(1 - f(x))$$

(Andreassen and Nachman PRD 101, 091901 (2020))

where f(x) is a neural network and trained with the binary cross-entropy loss function

to distinguish jets coming from data vs from simulation

Unfolding → Reweighting histograms → Classification → Neural network

Where does the machine learning part come in?

Method

- Architechture: Dense neural network Activation function for dense layers: Rectified linear unit
- Activation function for output layer: Sigmoid
- Loss function: Binary cross entropy

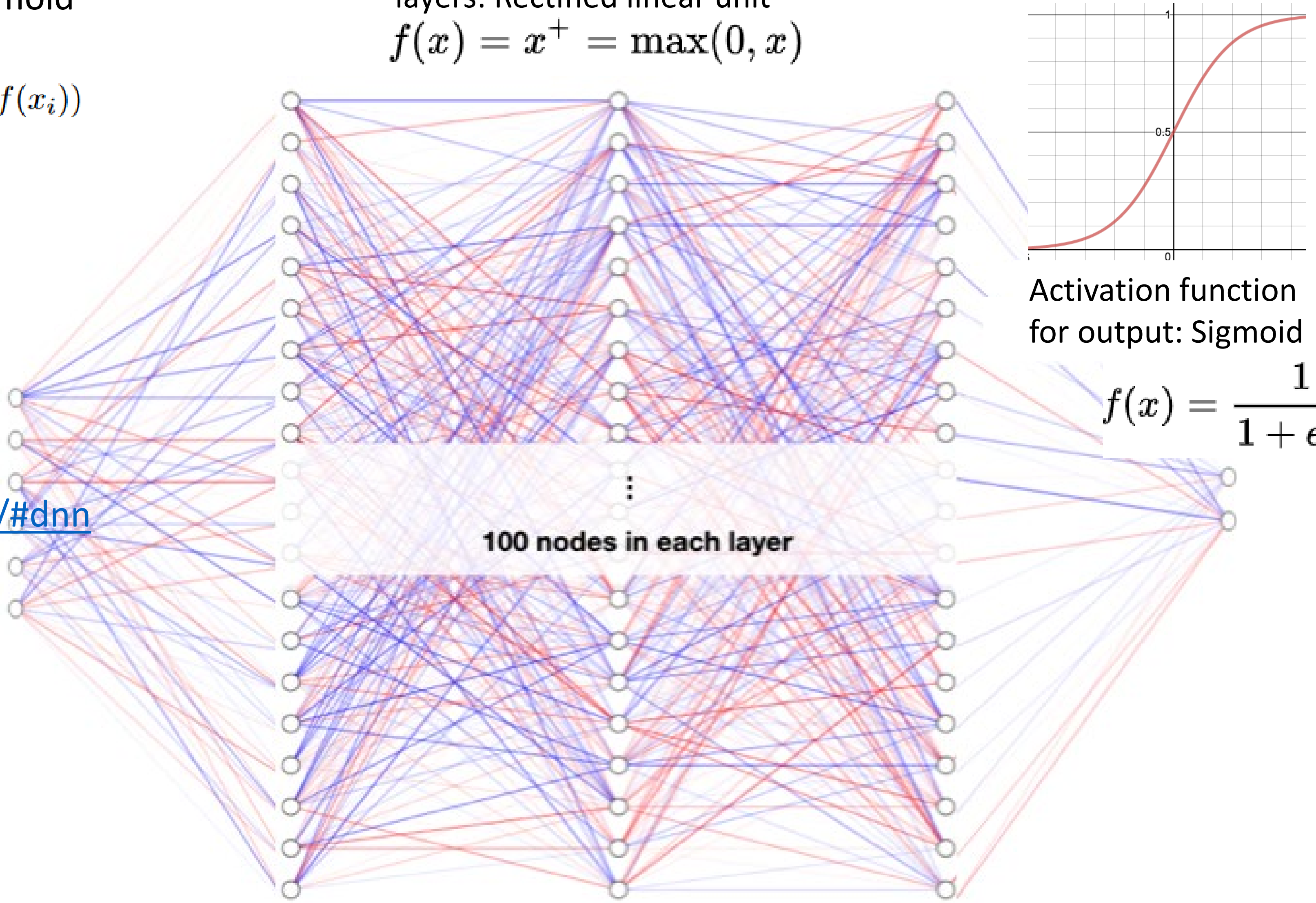
$$\mathrm{loss}(f(x)) = -\sum_{i\in 0} \log f(x_i) - \sum_{i\in 1} \log(1 - f(x_i))$$

- Optimization algorithm: Adam
  https://arxiv.org/pdf/1412.6980.pdf
- Nodes per dense layer: [100,100,100]
- Output dimension: 2
- Input dimension: 6

- All hyperparameters are default:
  https://energyflow.network/docs/archs/#dnn

Activation function for dense layers: Rectified linear unit

$$f(x) = x^+ = \max(0, x)$$



Activation function for output: Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

Method

- https://energyflow.network/docs/archs/

Compilation Options

- **loss**= `'categorical_crossentropy'` : *str*
  - The loss function to use for the model. See the Keras loss function docs for available loss functions.
- **optimizer**= `'adam'` : Keras optimizer or *str*
  - A Keras optimizer instance or a string referring to one (in which case the default arguments are used).
- **metrics**= `['accuracy']` : *list* of *str*
  - The Keras metrics to apply to the model.
- **compile_opts**= `{}` : *dict*
  - Dictionary of keyword arguments to be passed on to the `compile` method of the model. `loss`, `optimizer`, and `metrics` (see above) are included in this dictionary. All other values are the Keras defaults.

Output Options

- **output_dim**= `2` : *int*
  - The output dimension of the model.
- **output_act**= `'softmax'` : *str* or Keras activation
  - Activation function to apply to the output.

Callback Options

- **filepath**= `None` : *str*
  - The file path for where to save the model. If `None` then the model will not be saved.
- **save_while_training**= `True` : *bool*
  - Whether the model is saved during training (using the `ModelCheckpoint` callback) or only once training terminates. Only relevant if `filepath` is set.
- **save_weights_only**= `False` : *bool*
  - Whether only the weights of the model or the full model are saved. Only relevant if `filepath` is set.
- **modelcheck_opts**= `{'save_best_only':True, 'verbose':1}` : *dict*
  - Dictionary of keyword arguments to be passed on to the `ModelCheckpoint` callback, if it is present. `save_weights_only` (see above) is included in this dictionary. All other arguments are the Keras defaults.
- **patience**= `None` : *int*
  - The number of epochs with no improvement after which the training is stopped (using the `EarlyStopping` callback). If `None` then no early stopping is used.
- **earlystop_opts**= `{'restore_best_weights':True, 'verbose':1}` : *dict*
  - Dictionary of keyword arguments to be passed on to the `EarlyStopping` callback, if it is present. `patience` (see above) is included in this dictionary. All other arguments are the Keras defaults.

Method

- https://energyflow.network/docs/archs/#dnn

Required DNN Hyperparameters

- **input_dim** : *int* =6
  - The number of inputs to the model.
- **dense_sizes** : {*tuple*, *list*} of *int*=[100,100,100]
  - The number of nodes in the dense layers of the model.

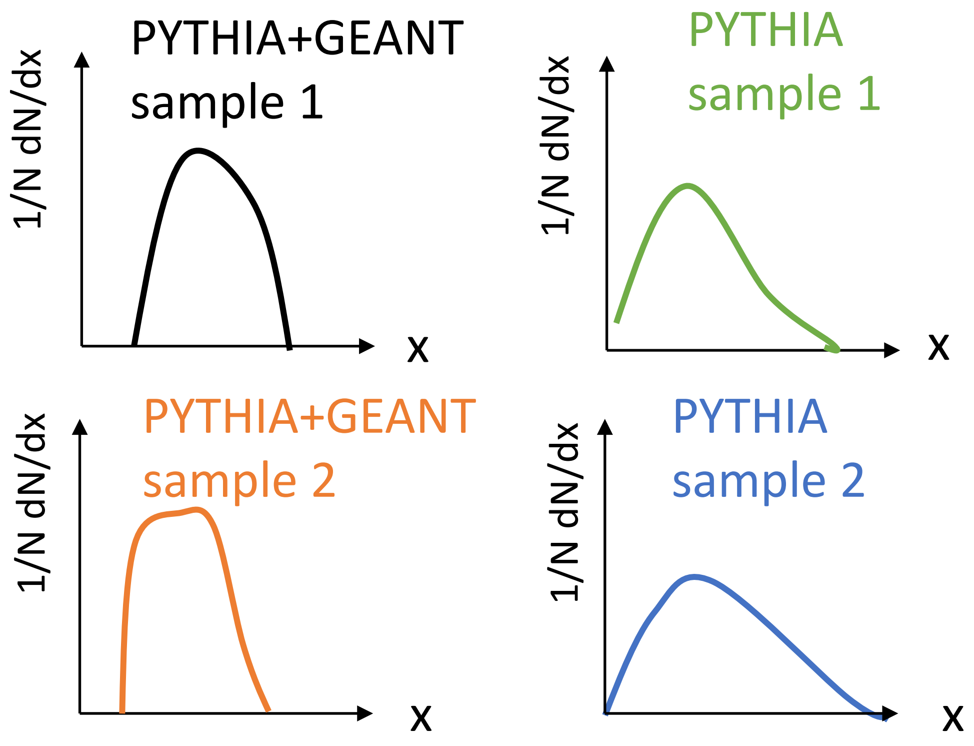
Default DNN Hyperparameters

- **acts=** `'relu'` : {*tuple*, *list*} of *str* or Keras activation
  - Activation functions(s) for the dense layers. A single string or activation layer will apply the same activation to all dense layers. Keras advanced activation layers are also accepted, either as strings (which use the default arguments) or as Keras `Layer` instances. If passing a single `Layer` instance, be aware that this layer will be used for all activations and may introduce weight sharing (such as with `PReLU`); it is recommended in this case to pass as many activations as there are layers in the model.See the Keras activations docs for more detail.
- **k_inits=** `'he_uniform'` : {*tuple*, *list*} of *str* or Keras initializer
  - Kernel initializers for the dense layers. A single string will apply the same initializer to all layers. See the Keras initializer docs for more detail.
- **dropouts=** `0` : {*tuple*, *list*} of *float*
  - Dropout rates for the dense layers. A single float will apply the same dropout rate to all layers. See the Keras Dropout layer for more detail.
- **l2_regs=** `0` : {*tuple*, *list*} of *float*

Closure test for unfolding

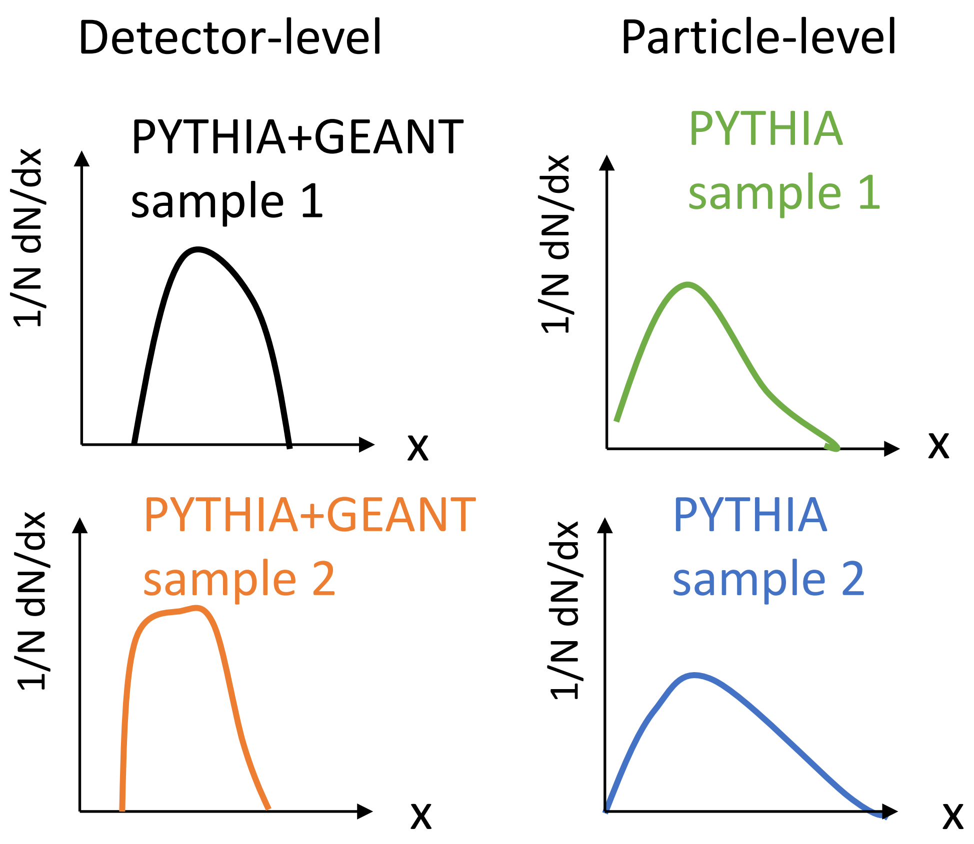
- Step 1: Separate matched jets from PYTHIA and PYTHIA+GEANT into 2 samples

Detector-level   Particle-level

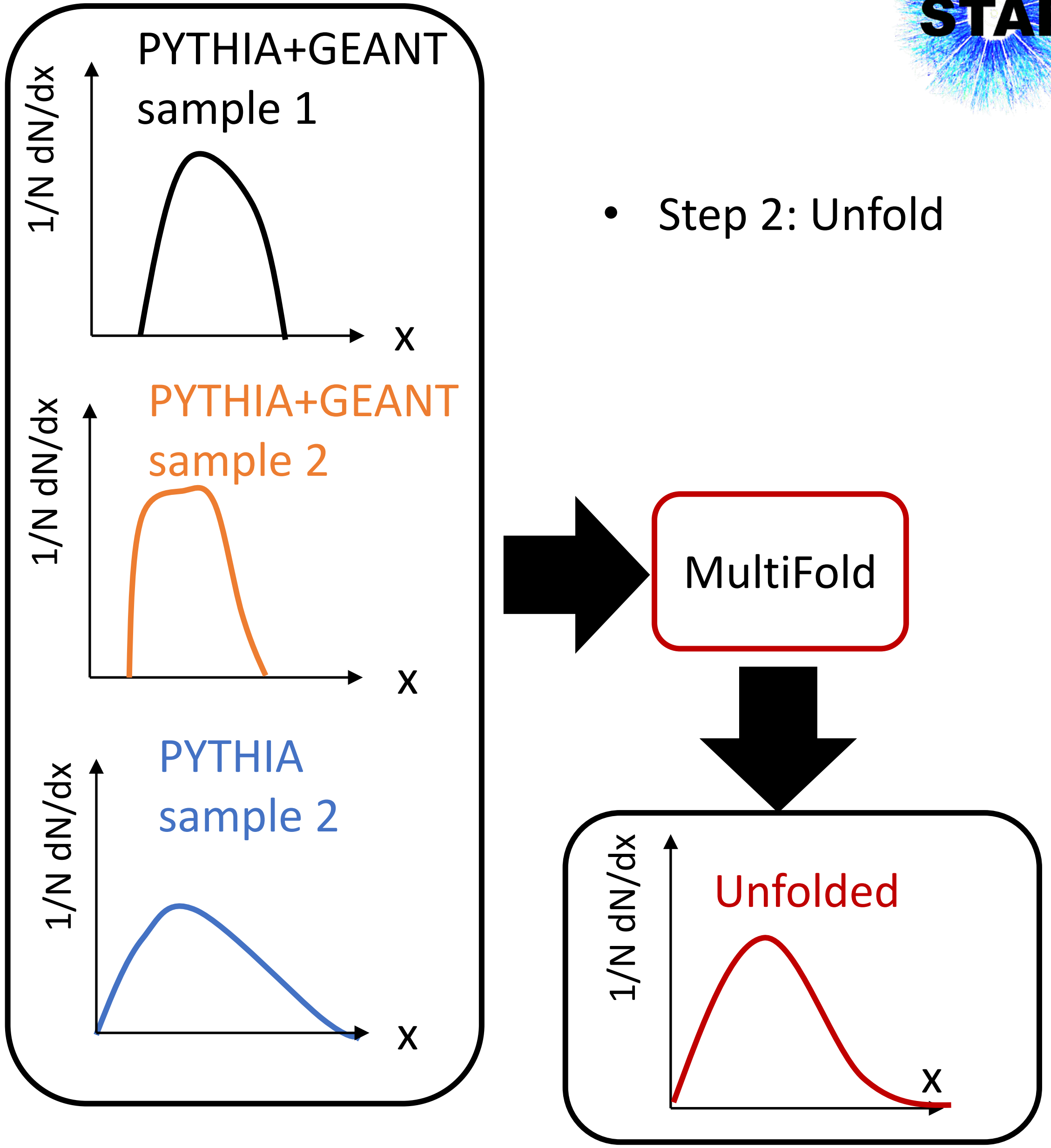


Closure test for unfolding

- Step 1: Separate matched jets from PYTHIA and PYTHIA+GEANT into 2 samples



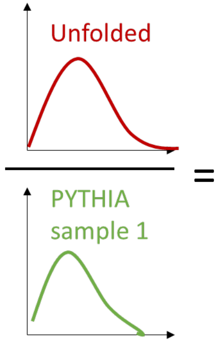
- Step 2: Unfold

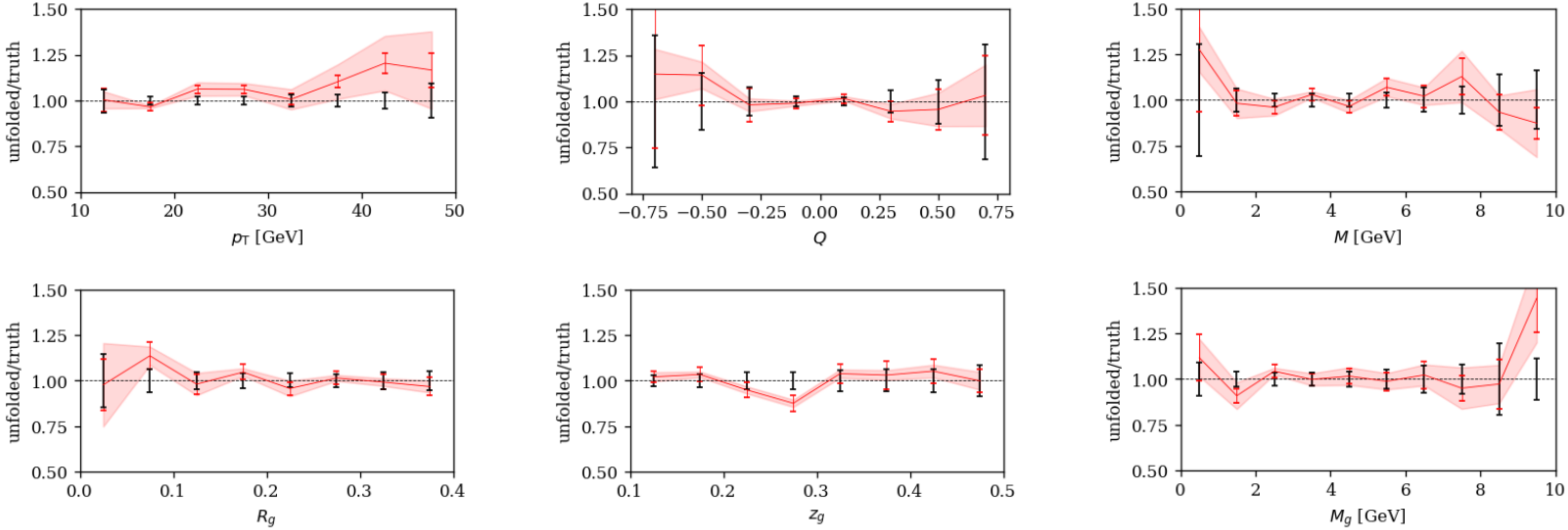


Closure test for unfolding: results

- Decent **closure** for all substructure observables

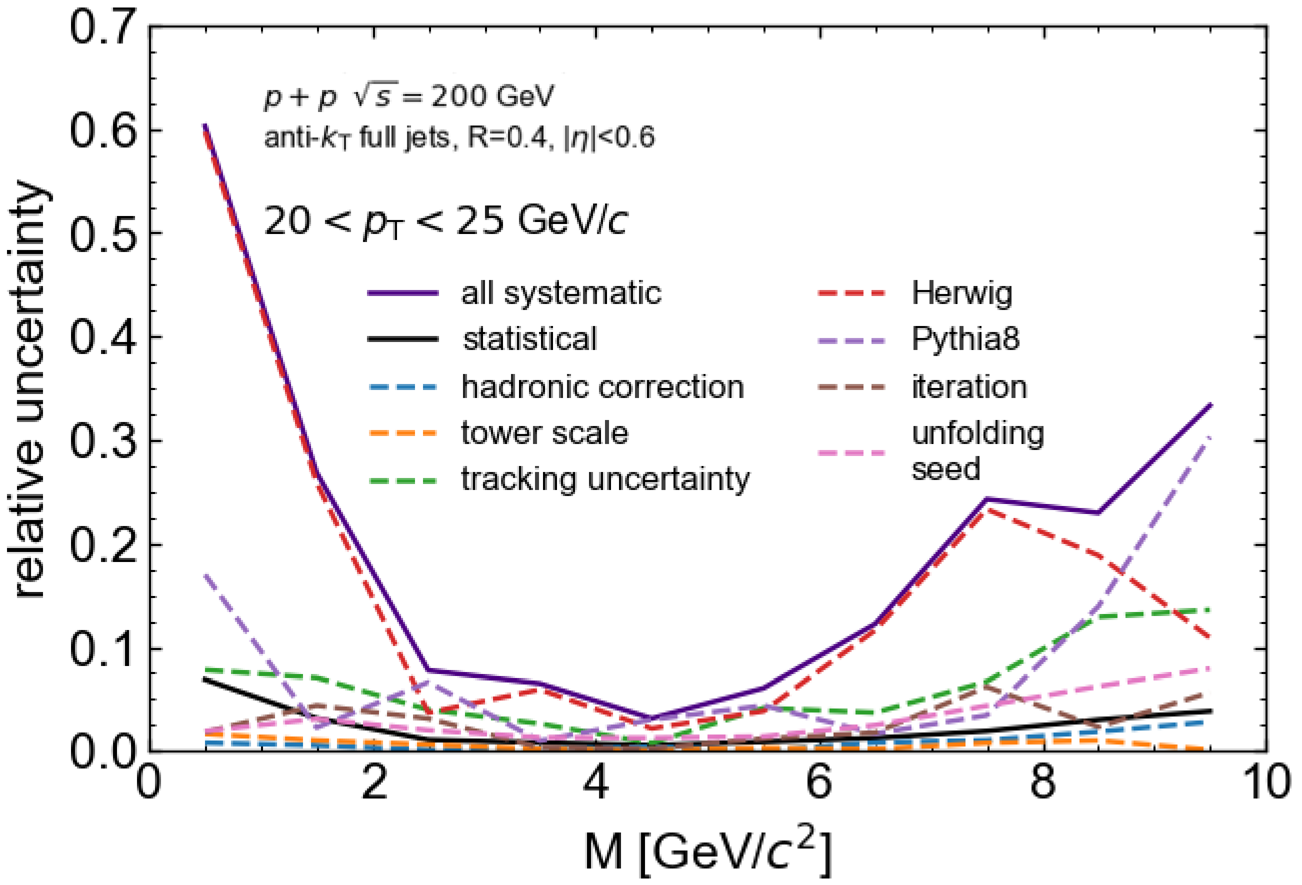
Unfolding unc. on data (not including misses)
Stat. unc. on sample 1
Stat. unc. on sample 2



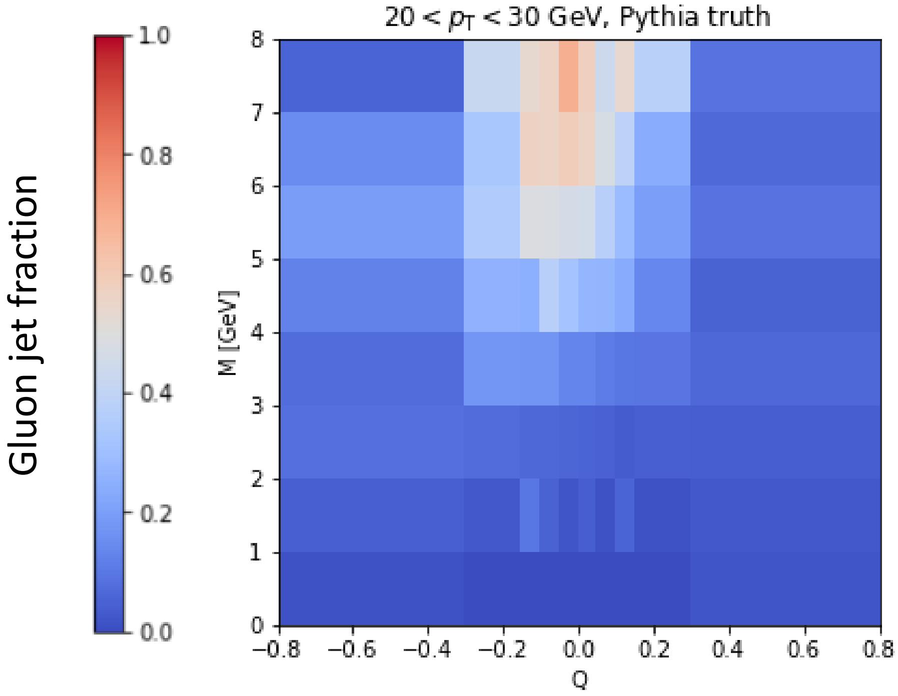


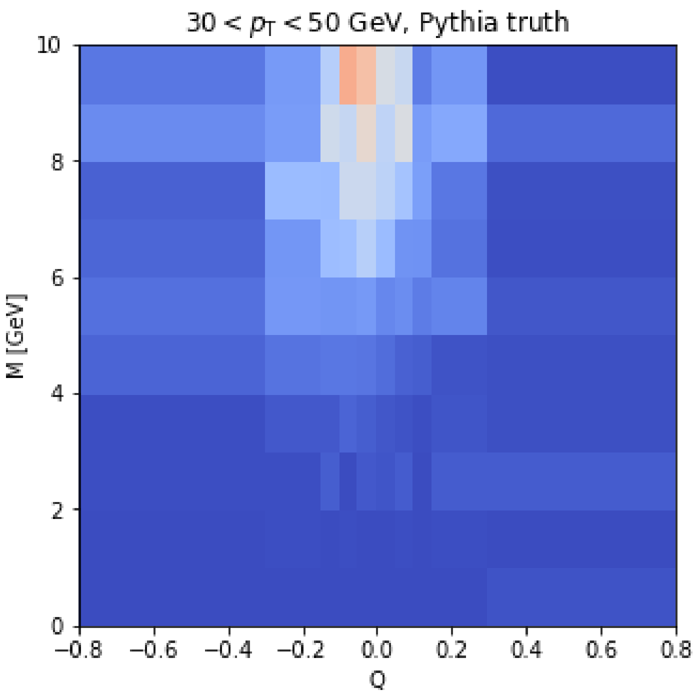
Systematic uncertainties

- Detector uncertainties (correlated with RooUnfold)
  - Hadronic correction 100% -> 50%
  - Tower scale +3.8%
  - Tracking efficiency -4%
- Unfolding uncertainties
  - Prior shape variation: Reweight jet mass distributions by HERWIG7 (LHC-UE-EE-4-CTEQ6L1 tune) and PYTHIA8 (Detroit tune)
  - Unfolding seed variation: Due to randomization of the initial weights
  - Iteration number variation

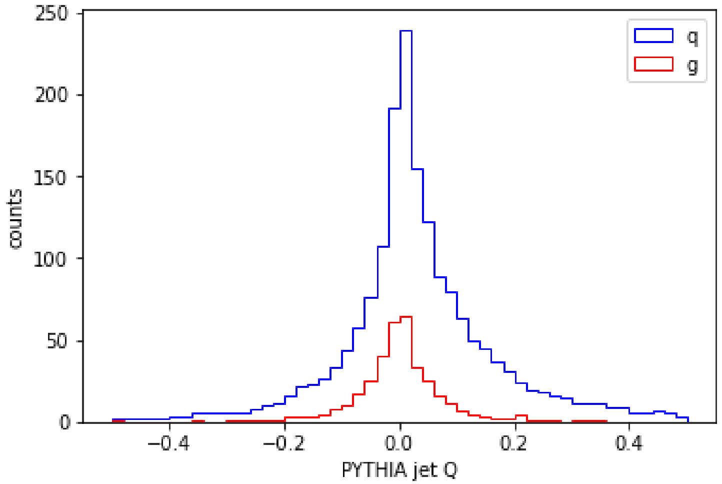


What's the best purity we can achieve for q vs g separation?





Full jets anti-kT R=0.4.
Pythia8 Detroit tune



- In 20 < pT < 30 GeV, **gluon fraction ~ 35%**
- To select a jet population with **gluon fraction = 67%**, cut on -0.025 < Q < 0 AND M > 7 GeV. (1.1% of all jets).
  - If we only cut on M > 7 GeV, **gluon fraction = 58%**. (Although we will have higher statistics).
  - If we want to reach **gluon fraction = 67%** with just a M cut, need M > 8.6 GeV. (0.8% of all jets).

- In 30 < pT < 50 GeV, **gluon fraction ~ 20%**
- To select a jet population with **gluon fraction = 65%**, cut on -0.08 < Q < -0.01 AND M > 9 GeV. (1.1%).
  - No cut on jet M/Q alone can achieve such a purity.