A simultaneous measurement of 24 observables in Z+jets events using the ATLAS detector

Laura Miller

Carleton University

Canadian Association of Physicists Congress 2023

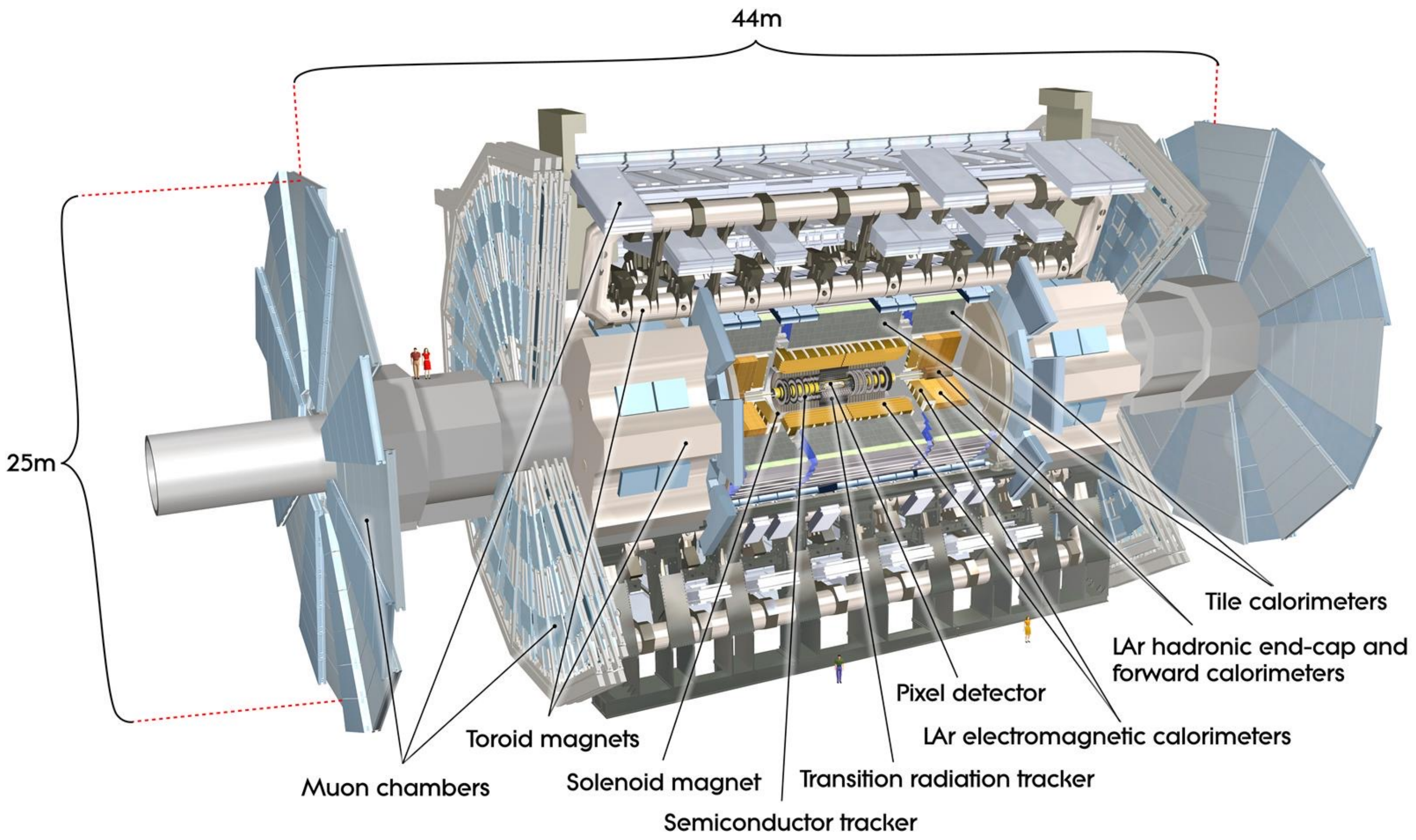
Fredericton, NB

June 19th, 2023

Overview

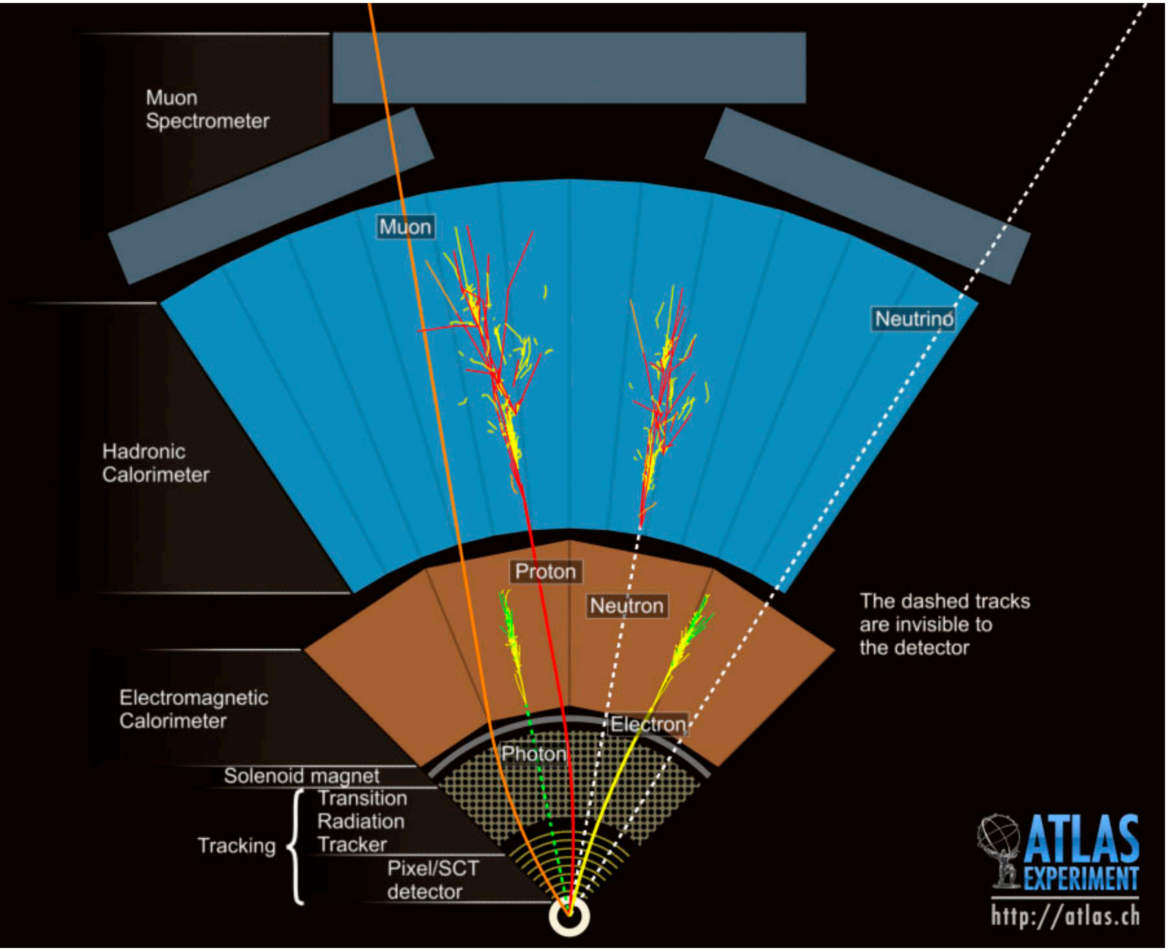
- **Ultimate goal: perform a high-dimensional and unbinned measurement using the ATLAS detector**

- **Part 1: The measurement**
  - The ATLAS detector
  - Z+jets events
  - Analysis details
- **Part 2: The MultiFold method**
  - Unfolding
  - Reweighting with neural networks
  - MultiFold method
- **Part 3: Results**



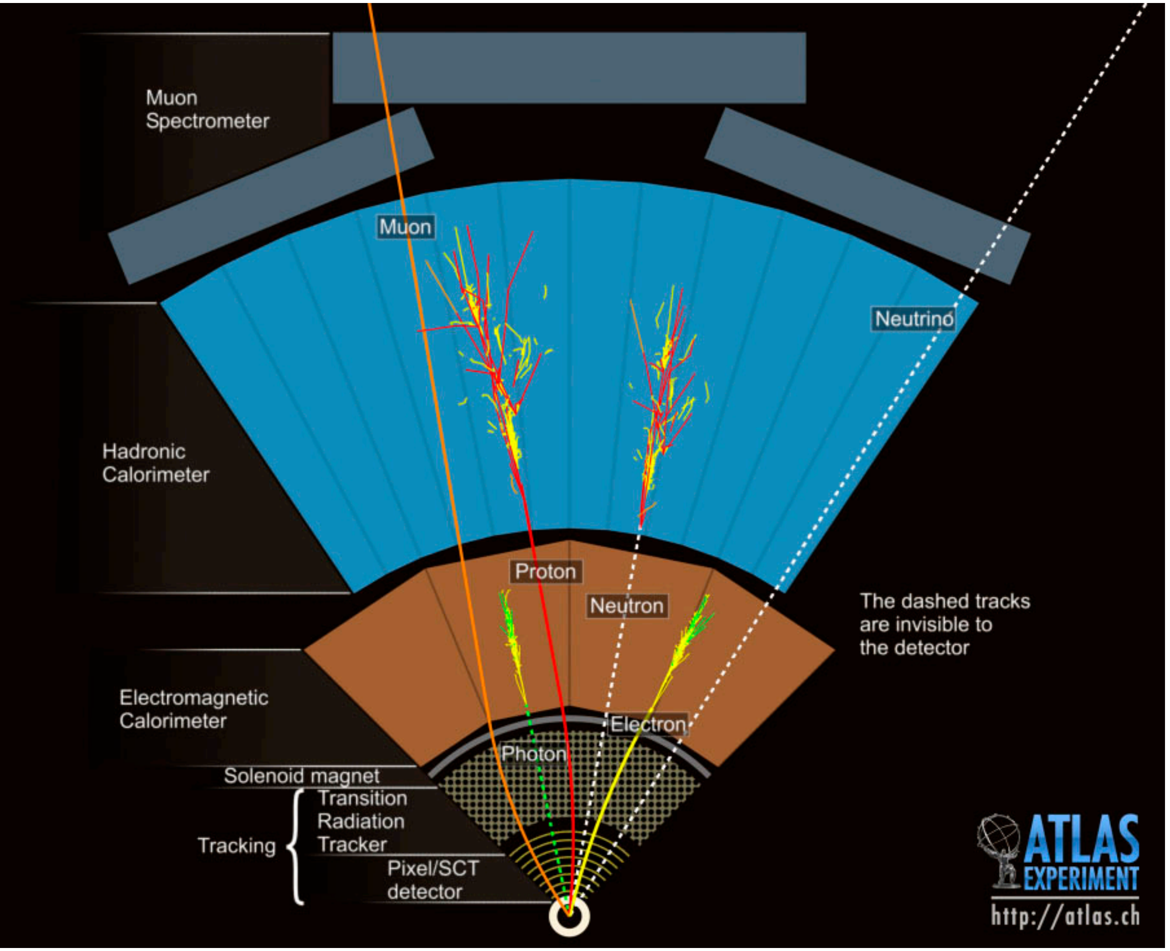
Part 1: The Measurement

The ATLAS detector



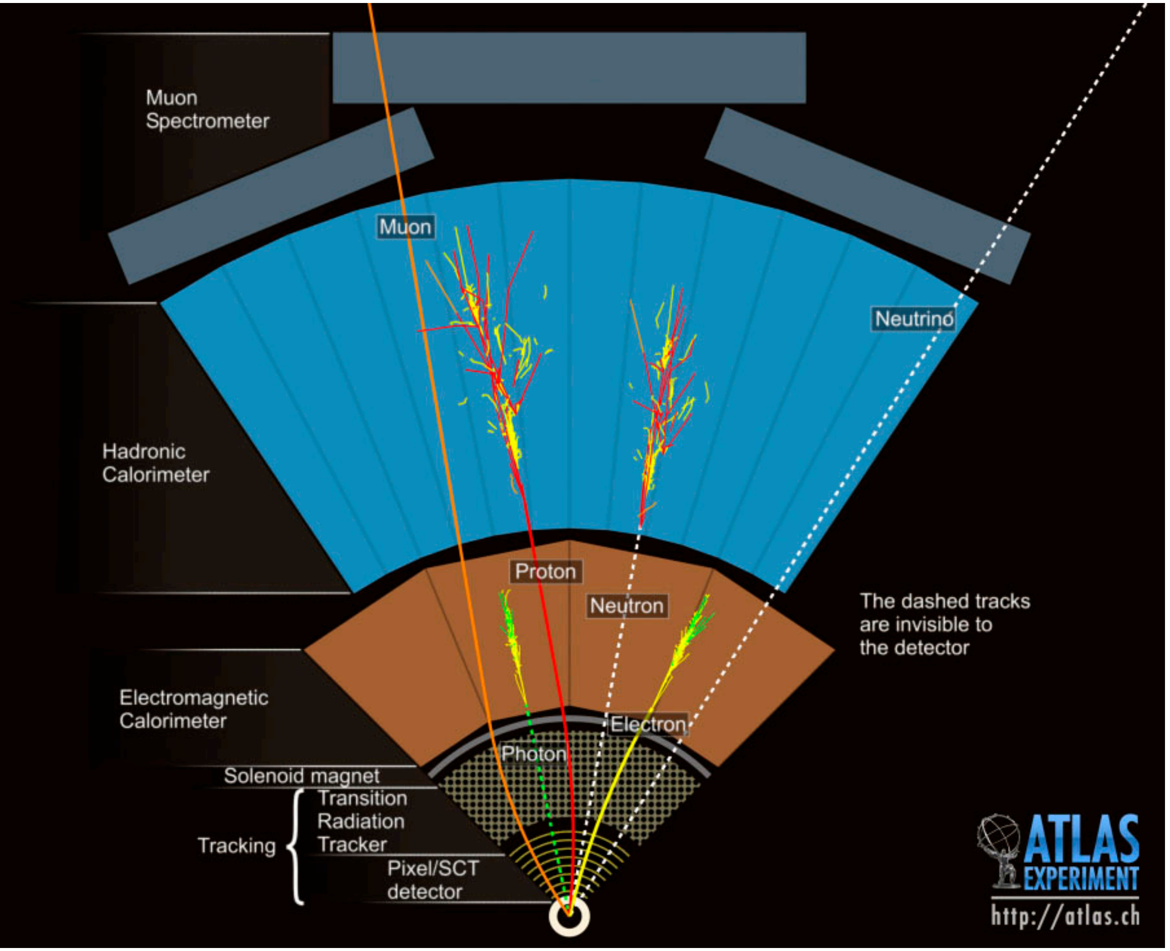
- The ATLAS detector is a general purpose detector located at the Large Hadron Collider at CERN
- Main components:
    - Inner detector: measures charged particles by constructing tracks using hits from the various silicon and gas detectors

The ATLAS detector



- The ATLAS detector is a general purpose detector located at the Large Hadron Collider at CERN
- Main components:
  - Electromagnetic and hadronic calorimeters: series of sampling calorimeters designed to contain all electromagnetic and hadronic activity
  - Inner detector: measures charged particles by constructing tracks using hits from the various silicon and gas detectors

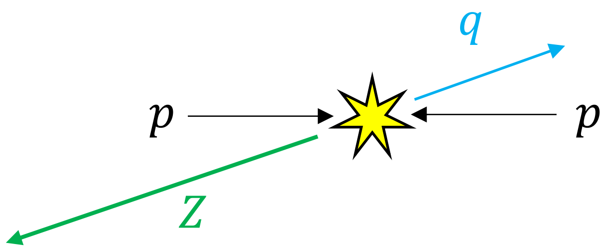
The ATLAS detector



- The ATLAS detector is a general purpose detector located at the Large Hadron Collider at CERN
- Main components:
  - Muon spectrometer: measures the muons, as they are minimum ionizing particles they are not contained in the calorimeters
  - Electromagnetic and hadronic calorimeters: series of sampling calorimeters designed to contain all electromagnetic and hadronic activity
  - Inner detector: measures charged particles by constructing tracks using hits from the various silicon and gas detectors

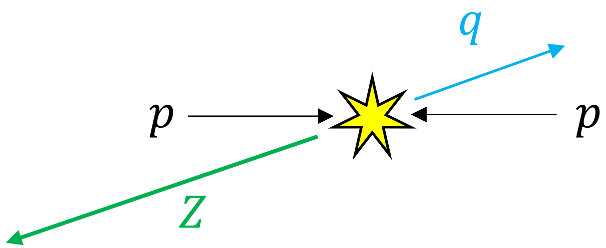
Z+jets events in the ATLAS detector

- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model



Z+jets events in the ATLAS detector

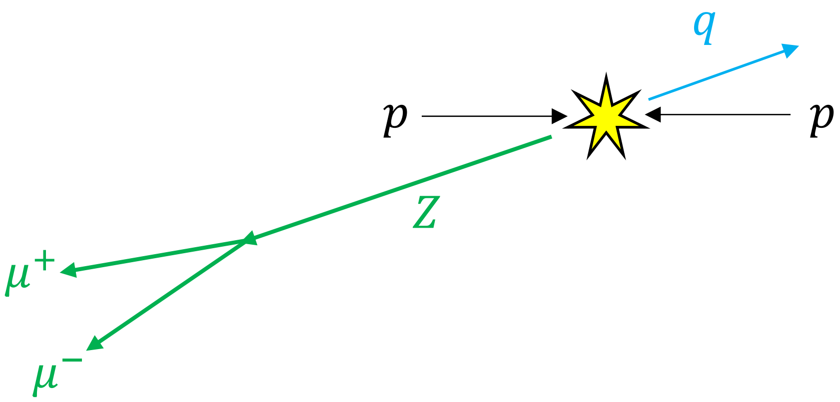
- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model



- The Z boson is a massive gauge boson
- Mediator of electroweak interaction

Z+jets events in the ATLAS detector

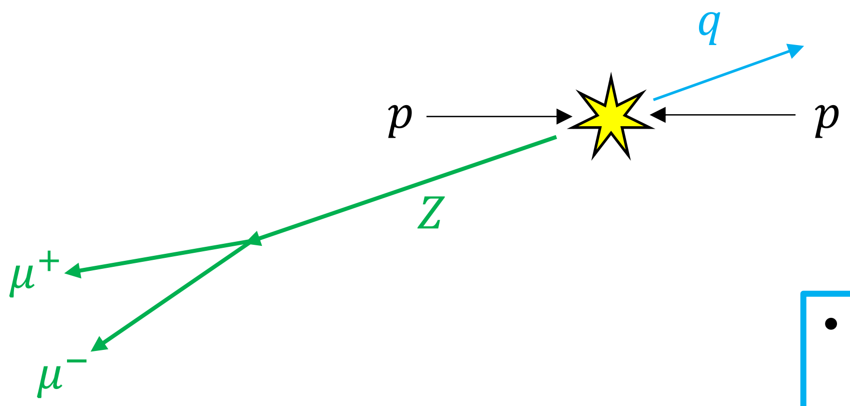
- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model



- The Z boson is a massive gauge boson
- Mediator of electroweak interaction
- In this case, interested in the decay to a muon and an anti-muon
  - ~3.4% of the time
  - "Easy" to reconstruct

Z+jets events in the ATLAS detector

- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model

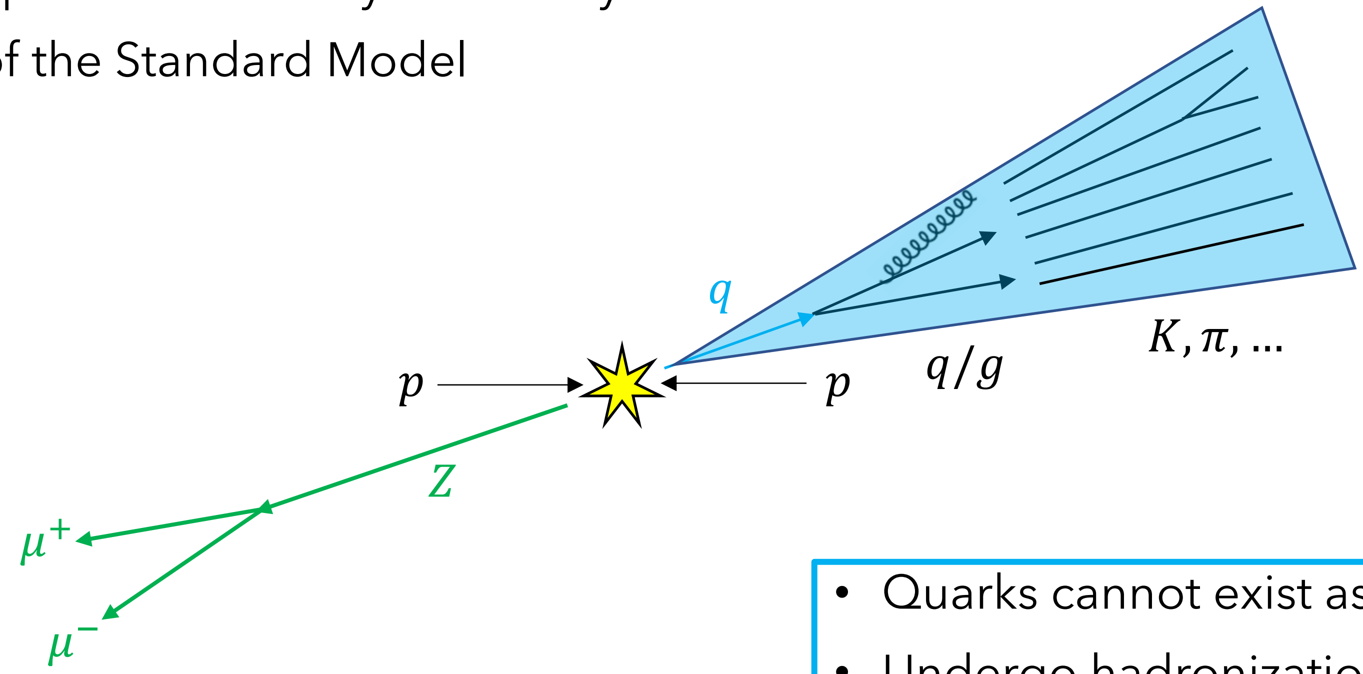


- The Z boson is a massive gauge boson
- Mediator of electroweak interaction
- In this case, interested in the decay to a muon and an anti-muon
  - ~3.4% of the time
  - "Easy" to reconstruct

- Quarks cannot exist as free particles

Z+jets events in the ATLAS detector

- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model



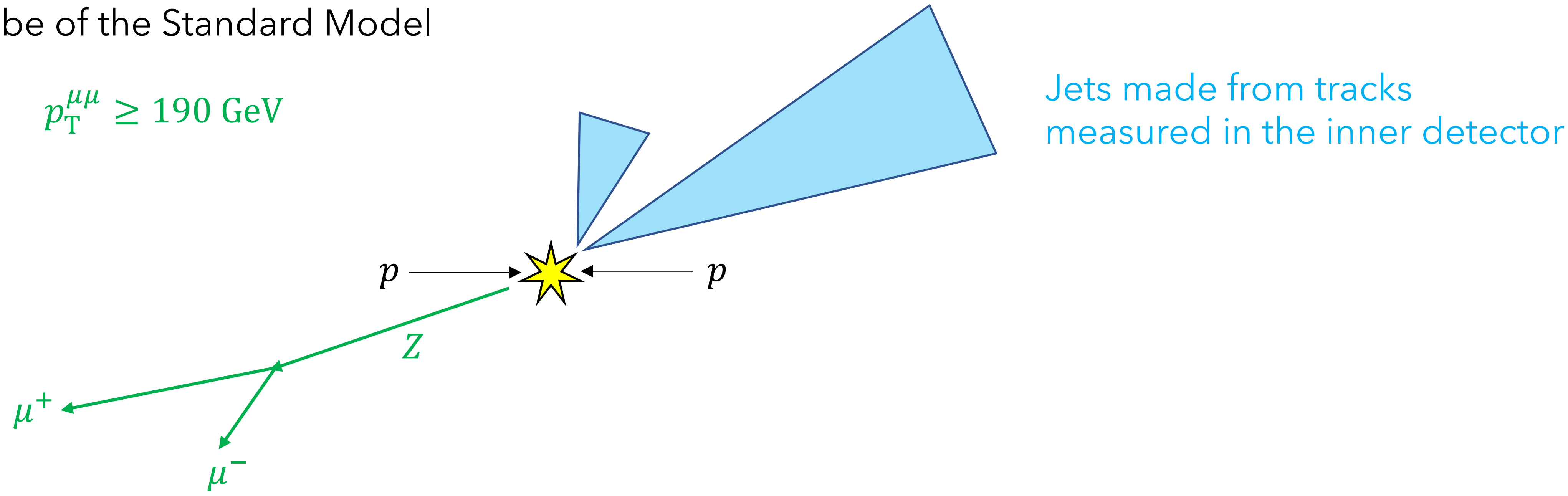
Note that it is possible to have more than one jet

- The Z boson is a massive gauge boson
- Mediator of electroweak interaction
- In this case, interested in the decay to a muon and an anti-muon
    - ~3.4% of the time
    - "Easy" to reconstruct

- Quarks cannot exist as free particles
- Undergo hadronization, producing a collimated shower of particles known as a jet
- Reconstructed by using a clustering algorithm to group together tracks in the inner detector, calorimeter energy deposits, or a combination of both

Z+jets events in the ATLAS detector

- Common process in the ATLAS detector and can be measured very precisely
- Low background process with easy-to-identify Z boson
- Precision probe of the Standard Model

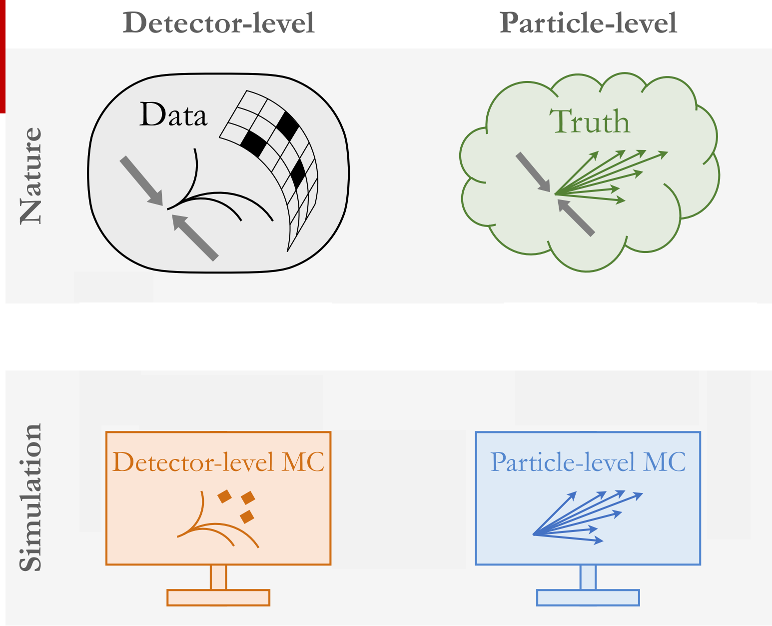


- **24 observables:** related to the dimuon and muon kinematics, track jet kinematics and track jet substructure
- Using the full ATLAS Run 2 dataset with $\sqrt{s} = 13$ TeV

Part 2: The MultiFold Method

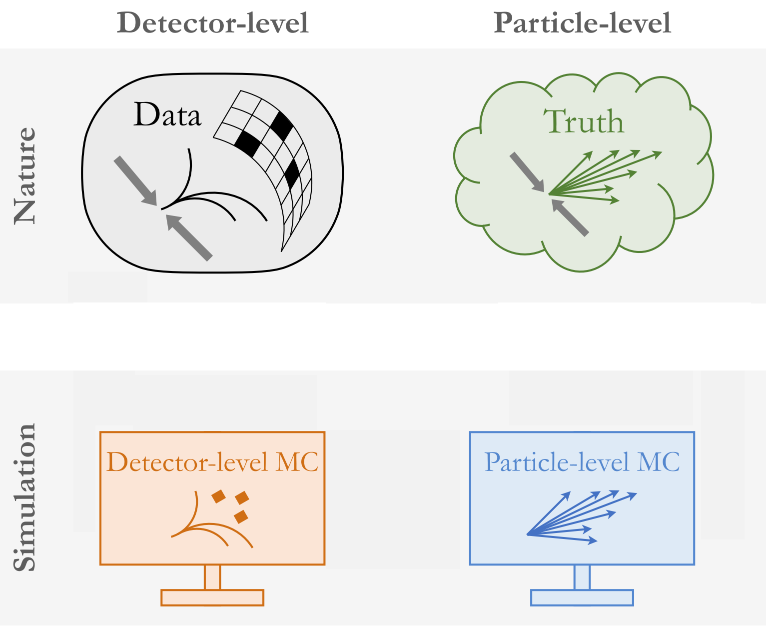
Unfolding preliminaries

- At ATLAS, have a few types of samples directly available to us:
  - **Data: what we measure**
  - **Truth-level Monte Carlo Simulation: what a Monte Carlo generator produces**
  - **Reconstructed (Reco)-level Monte Carlo Simulation: the truth-level MC after it passes through the simulated ATLAS detector**
- Want measurements of **truth data**, i.e. what actually happens in nature
  - Needed to compare with results from other experiments and theory predictions



- The process of correcting the data for the effects of the ATLAS detector is known as **unfolding**

Unfolding preliminaries



- At ATLAS, have a few types of samples directly available to us:
  - **Data: what we measure**
  - **Truth-level Monte Carlo Simulation: what a Monte Carlo generator produces**
  - **Reconstructed (Reco)-level Monte Carlo Simulation: the truth-level MC after it passes through the simulated ATLAS detector**
- Want measurements of **truth data**, i.e. what actually happens in nature
  - Needed to compare with results from other experiments and theory predictions
- The process of correcting the data for the effects of the ATLAS detector is known as **unfolding**
- Traditional unfolding methods: work with 1D binned data

Create 1D histogram of the observable with the data events

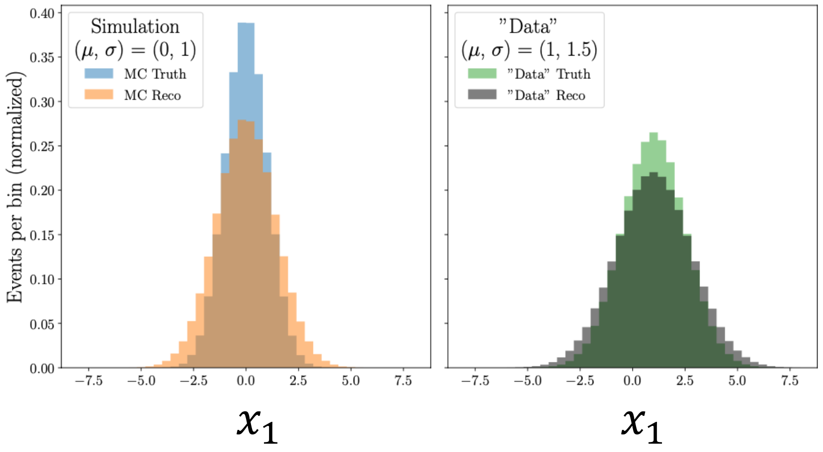
Use information about detector response from the MC samples to correct the histogram for detector effects in each bin

1D observable histogram with number of unfolded events per bin as output

- Can do this in a **multi-dimensional** and **unbinned** way with the help of neural networks

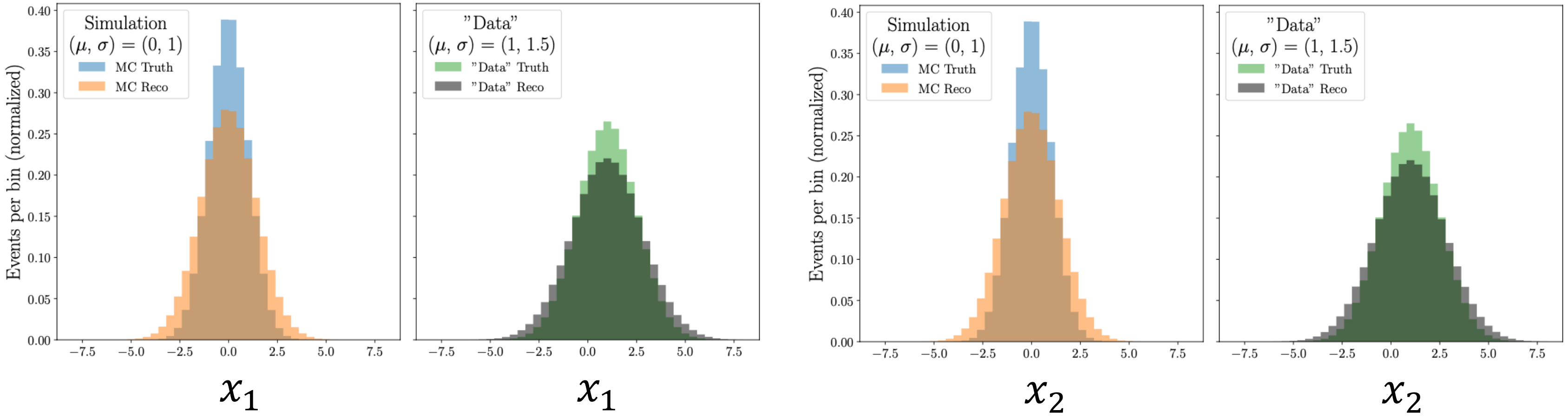
Sample reweighting

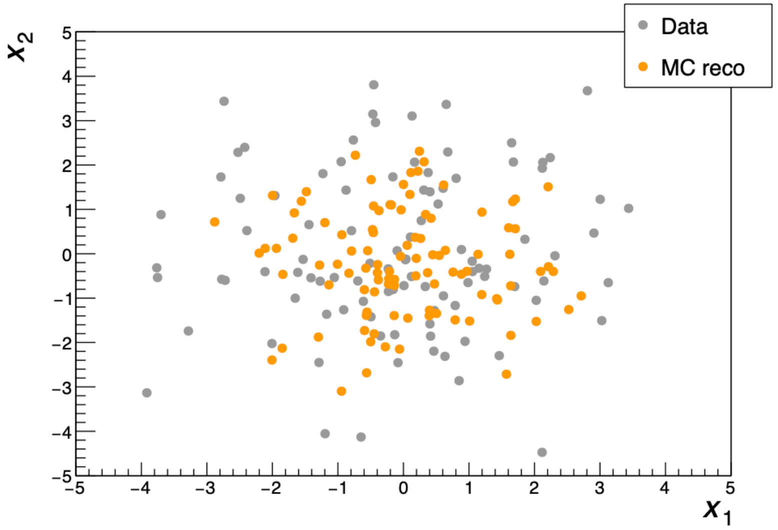
- Let's reweight the **reco MC** to match the **data** using a simple Gaussian example
- Each sample contains a set of events, each with a set of features $\vec{x} = (x_1, \ldots, x_{24})$



Sample reweighting

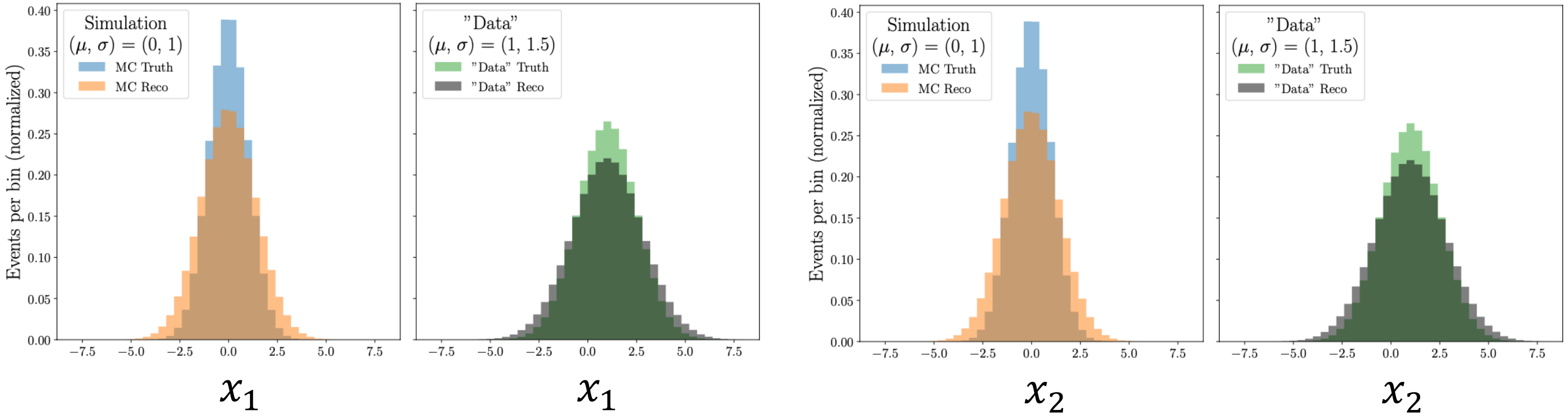
- Let's reweight the **reco MC** to match the **data** using a simple Gaussian example
- Each sample contains a set of events, each with a set of features $\vec{x} = (x_1, \ldots, x_{24})$



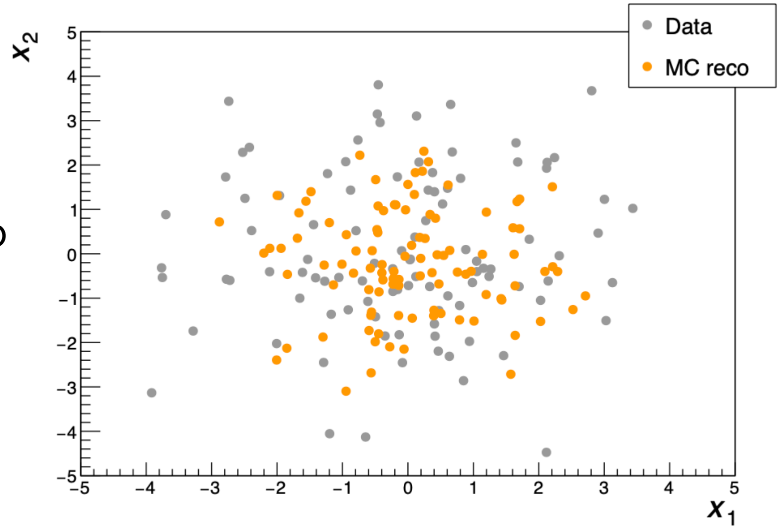


Sample reweighting

- Let's reweight the **reco MC** to match the **data** using a simple Gaussian example
- Each sample contains a set of events, each with a set of features $\vec{x} = (x_1, \ldots, x_{24})$

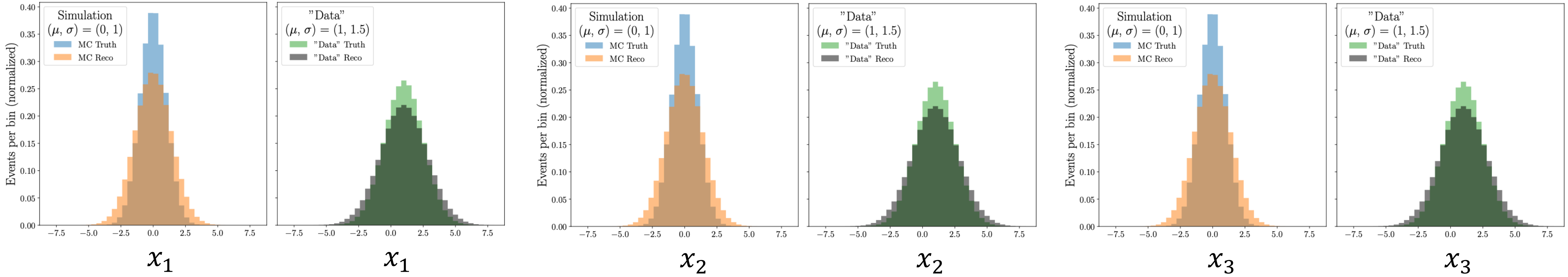


To reweight, can estimate the density ratio between the two samples

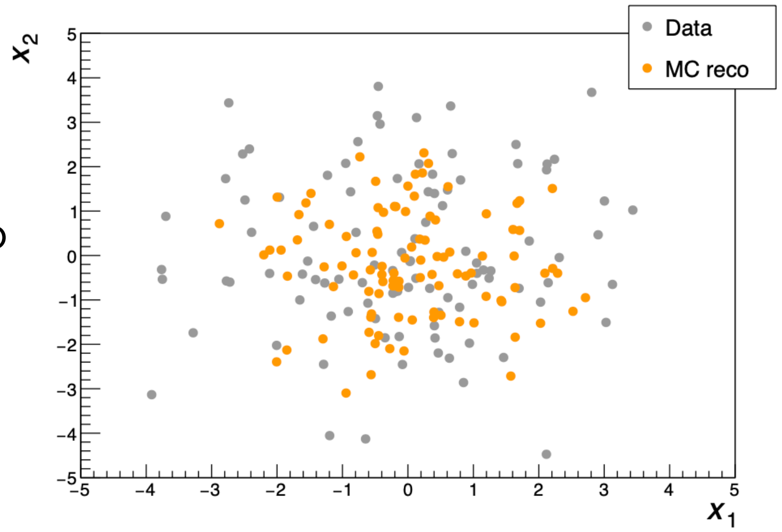


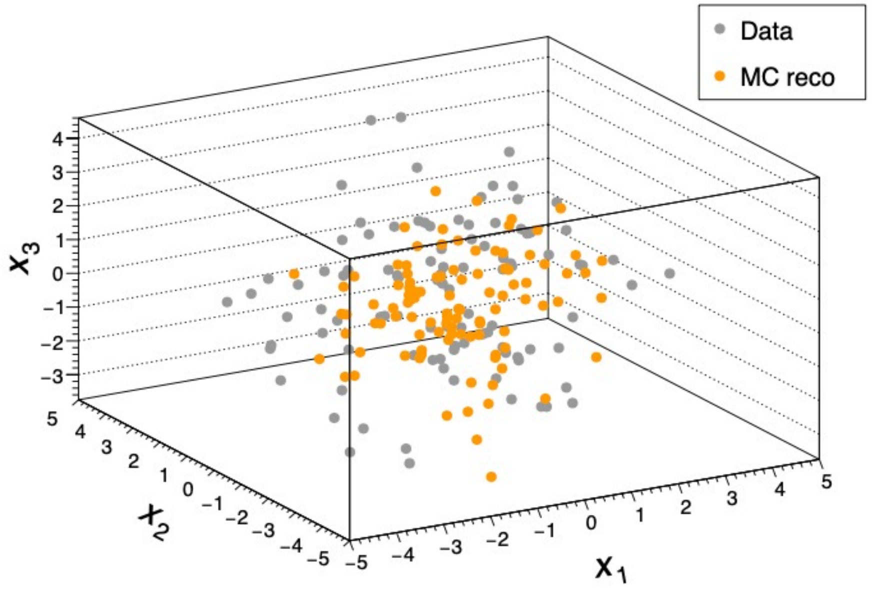
Sample reweighting

- Let's reweight the **reco MC** to match the **data** using a simple Gaussian example
- Each sample contains a set of events, each with a set of features $\vec{x} = (x_1, \ldots, x_{24})$



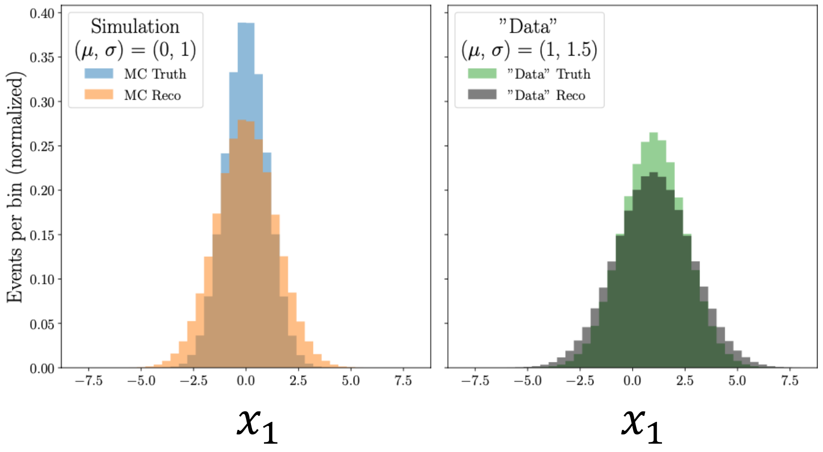
To reweight, can estimate the density ratio between the two samples



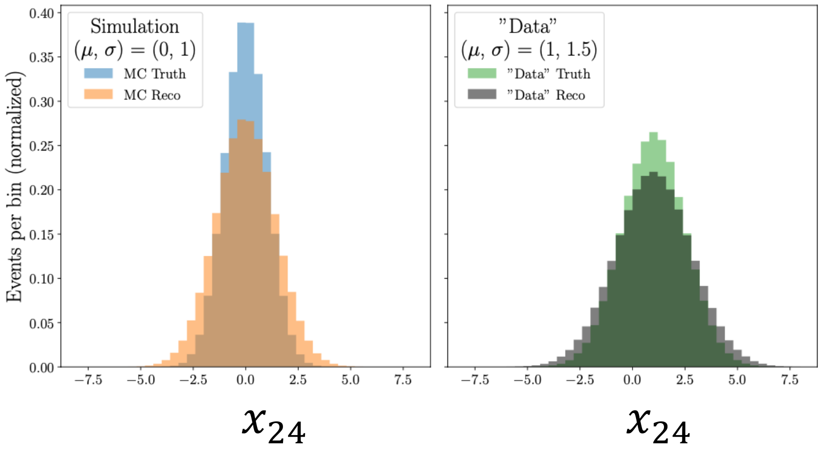


Sample reweighting

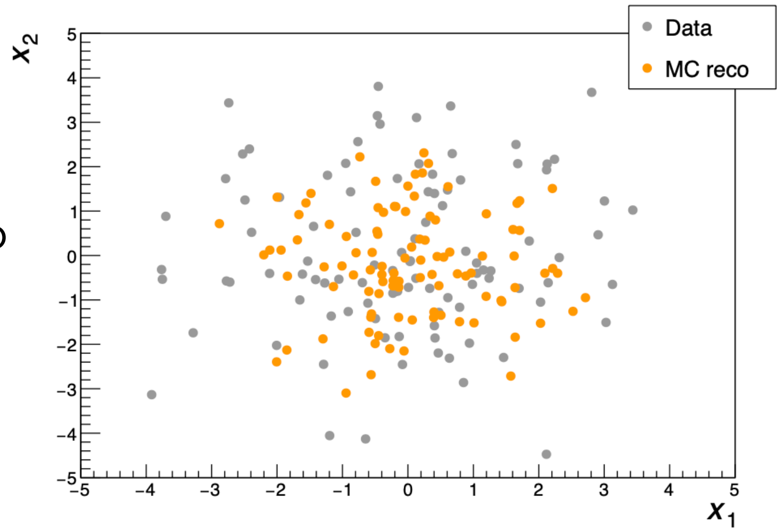
- Let's reweight the **reco MC** to match the **data** using a simple Gaussian example
- Each sample contains a set of events, each with a set of features $\vec{x} = (x_1, \dots, x_{24})$

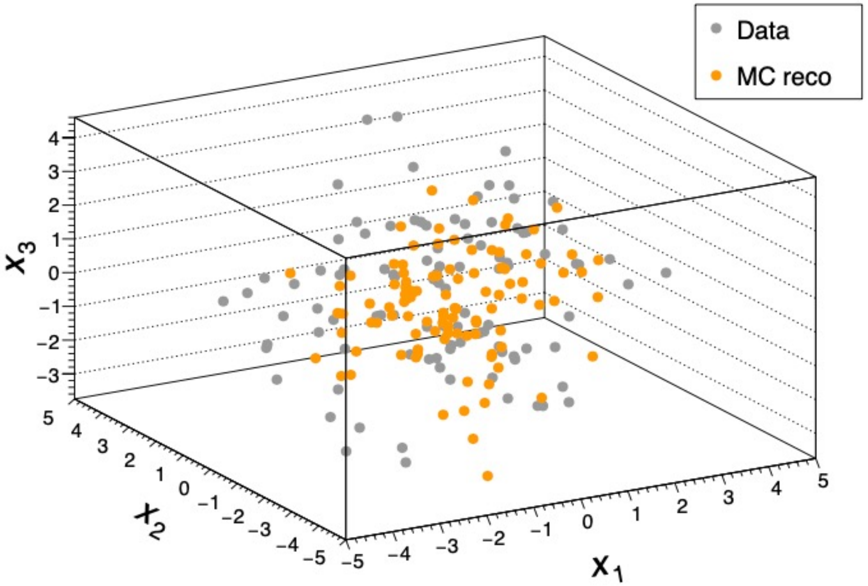


…



To reweight, can estimate the density ratio between the two samples



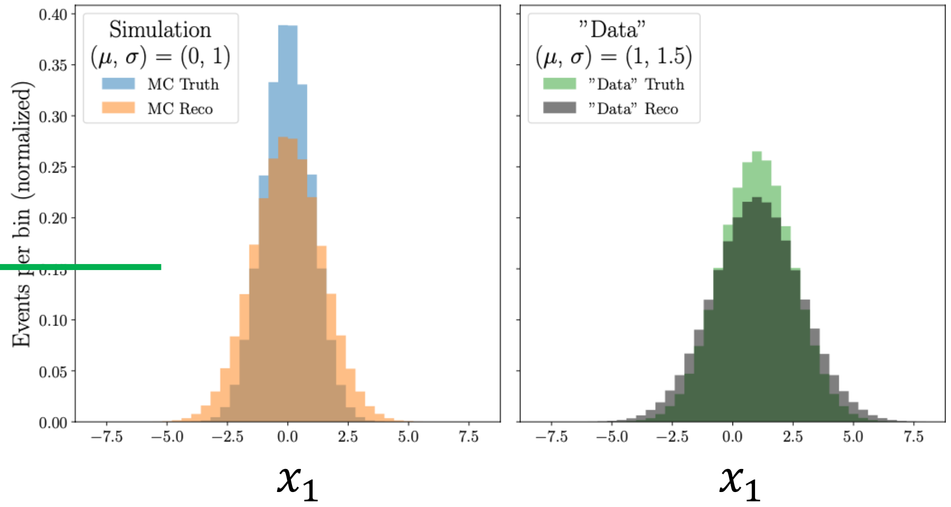


…

Neural networks for reweighting

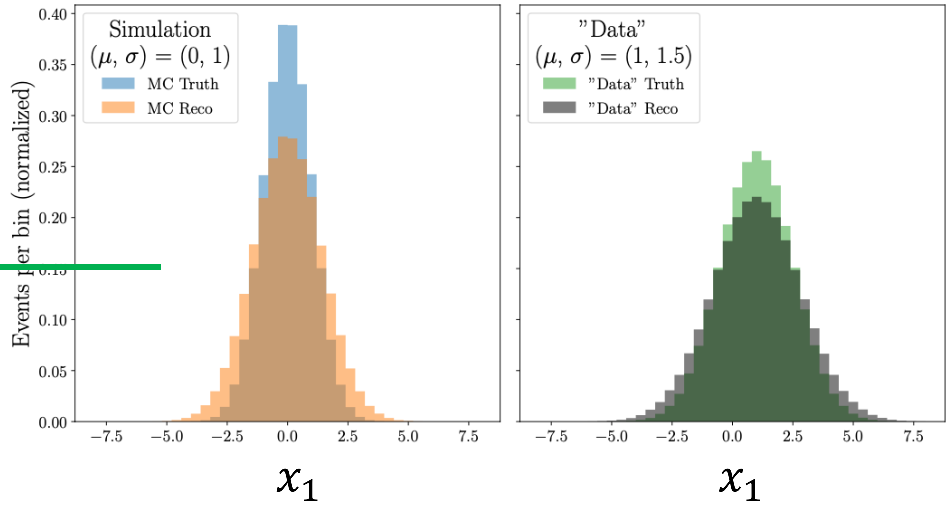
- Neural networks are well suited for reweighting tasks
- Main principle: train a classifier and reinterpret the outputs

Train a classifier using $\vec{x} = (x_1, \ldots, x_{24})$ as input to differentiate between **reco MC** and **data**



Neural networks for reweighting

- Neural networks are well suited for reweighting tasks
- Main principle: train a classifier and reinterpret the outputs



Train a classifier using $\vec{x} = (x_1, \dots, x_{24})$ as input to differentiate between **reco MC** and **data**

Use the classifier output, $f(\vec{x}) \in [0,1]$, to construct reweighting function

$$\Downarrow$$

$$\omega = \frac{f(\vec{x})}{1 - f(\vec{x})} \propto \frac{p(\vec{x}|\text{Data})}{p(\vec{x}|\text{MC})}$$

Neural networks for reweighting

- Neural networks are well suited for reweighting tasks
- Main principle: train a classifier and reinterpret the outputs

Train a classifier using $\vec{x} = (x_1, \ldots, x_{24})$ as input to differentiate between **reco MC** and **data**

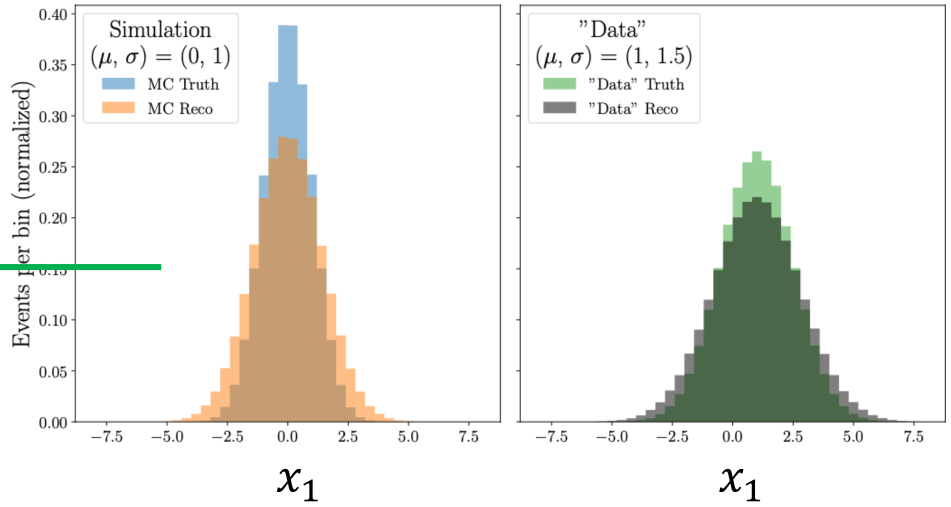
Use the classifier output, $f(\vec{x}) \in [0,1]$, to construct reweighting function

$\Downarrow$

$$\omega = \frac{f(\vec{x})}{1 - f(\vec{x})} \propto \frac{p(\vec{x}|\text{Data})}{p(\vec{x}|\text{MC})}$$

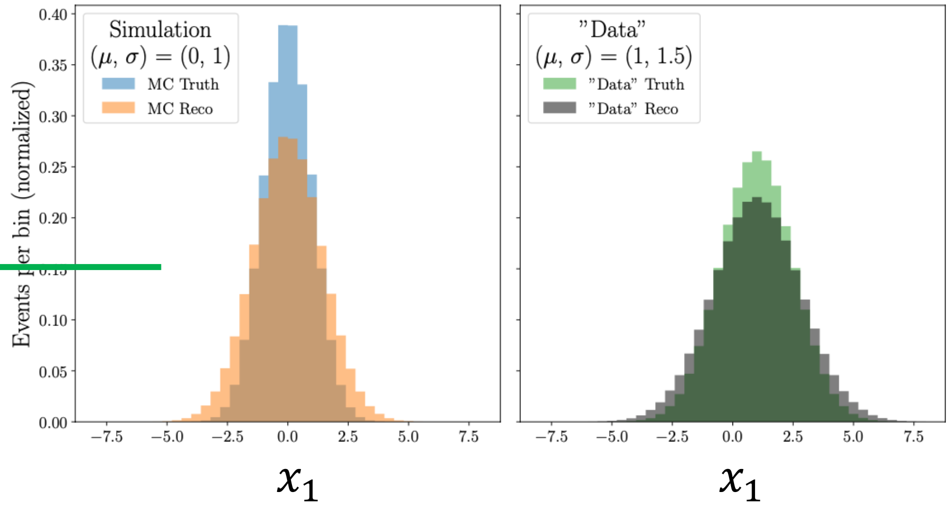
Higher weight if event is more "data-like"

Simultaneously reweights all 24 dimensions



Neural networks for reweighting

- Neural networks are well suited for reweighting tasks
- Main principle: train a classifier and reinterpret the outputs



Train a classifier using $\vec{x} = (x_1, \dots, x_{24})$ as input to differentiate between **reco MC** and **data**

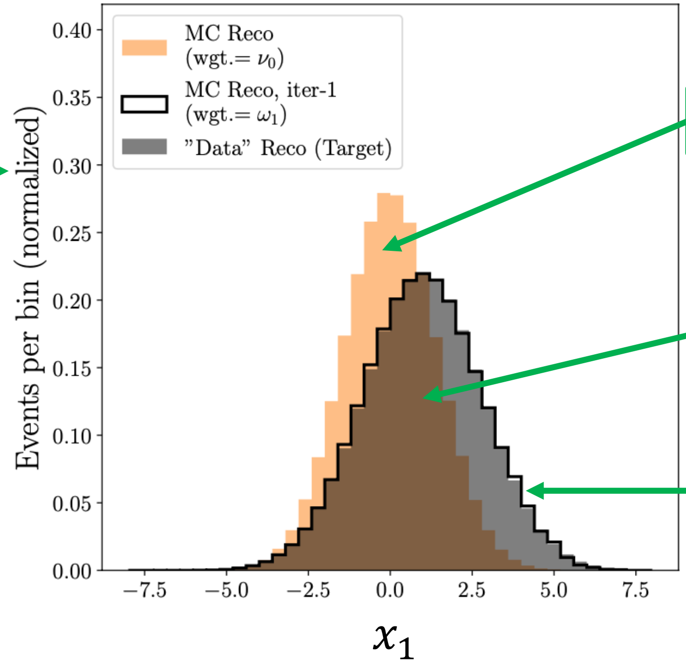
Use the classifier output, $f(\vec{x}) \in [0,1]$, to construct reweighting function

$$\Downarrow$$

$$\omega = \frac{f(\vec{x})}{1 - f(\vec{x})} \propto \frac{p(\vec{x}|\text{Data})}{p(\vec{x}|\text{MC})}$$

Higher weight if event is more "data-like"

Simultaneously reweights all 24 dimensions



Original **reco MC** distribution

Data is the target for our reweighting

Solid black line: **reco MC reweighted with ω**

Neural networks for reweighting

- Neural networks are well suited for reweighting tasks
- Main principle: train a classifier and reinterpret the outputs

Train a classifier using $\vec{x} = (x_1, \ldots, x_{24})$ as input to differentiate between **reco MC** and **data**

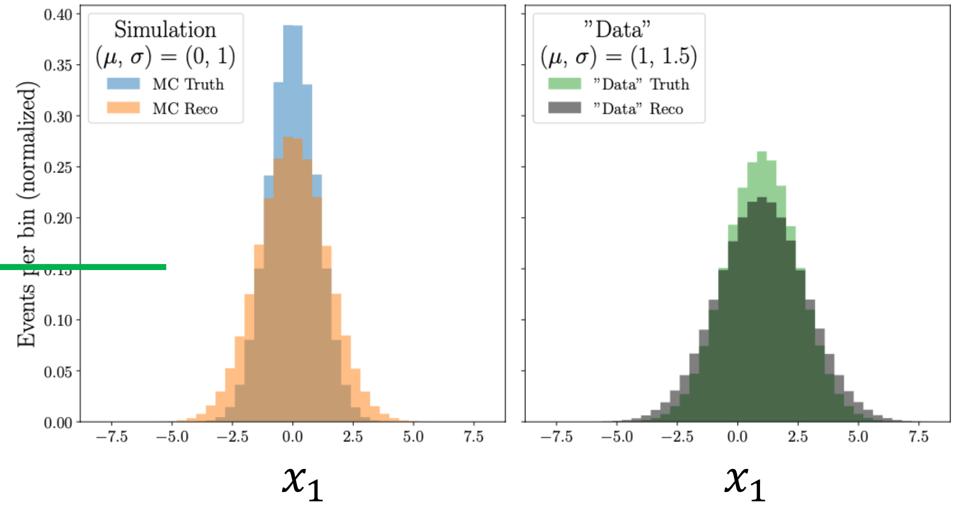
Use the classifier output, $f(\vec{x}) \in [0,1]$, to construct reweighting function

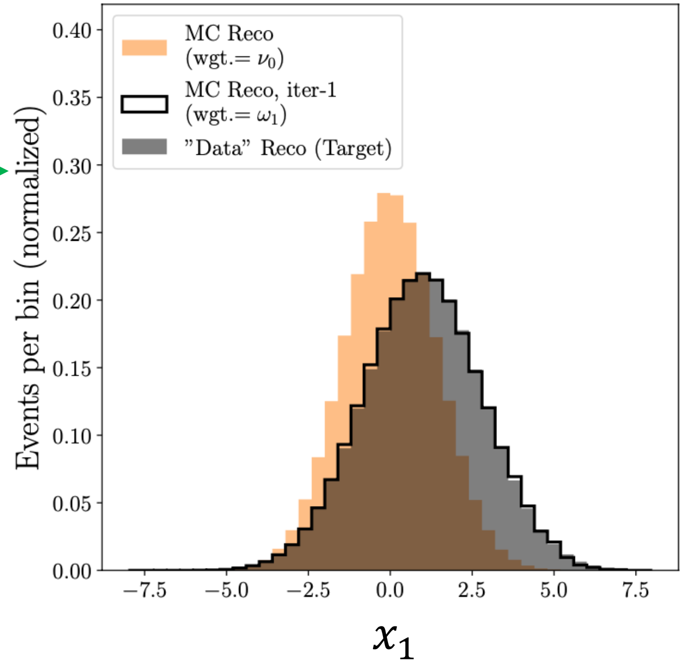
$$\Downarrow$$

$$\omega = \frac{f(\vec{x})}{1 - f(\vec{x})} \propto \frac{p(\vec{x}|\text{Data})}{p(\vec{x}|\text{MC})}$$

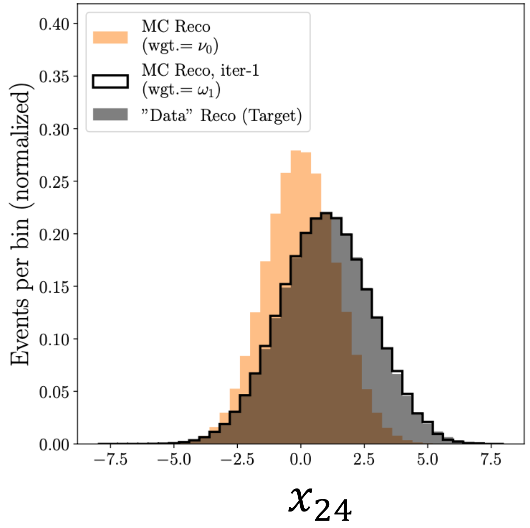
Higher weight if event is more "data-like"

Simultaneously reweights all 24 dimensions



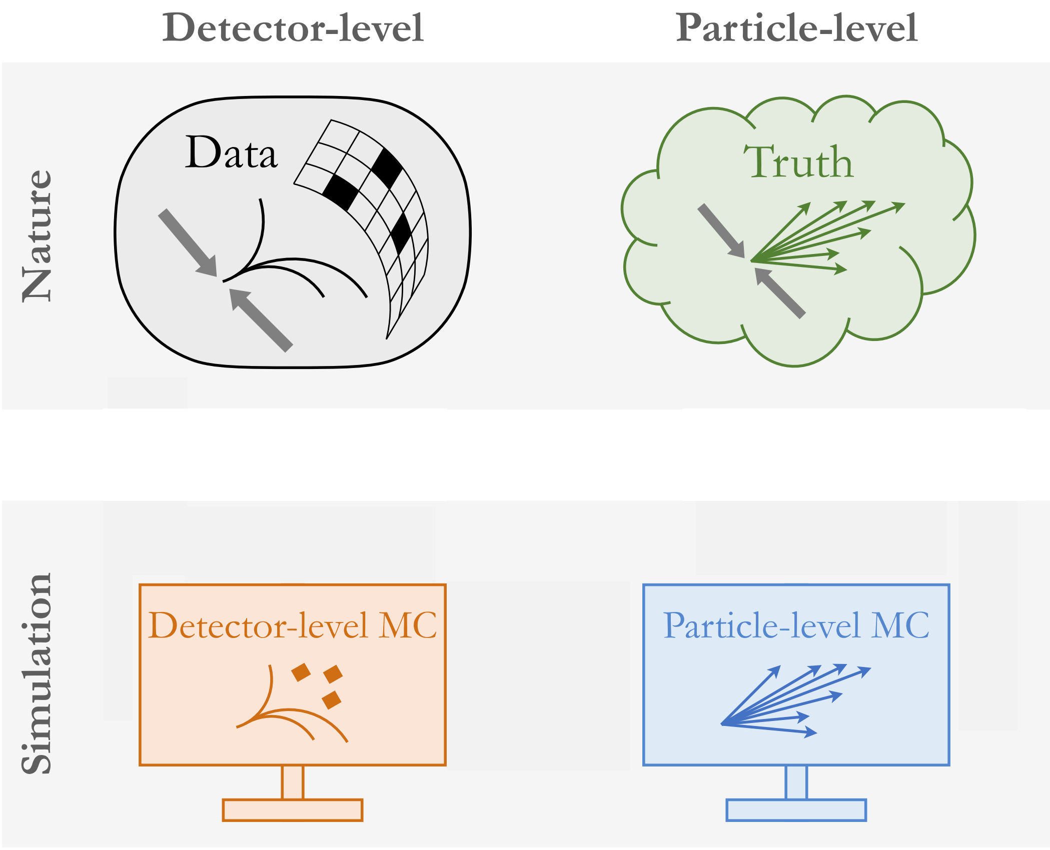


...



The MultiFold method

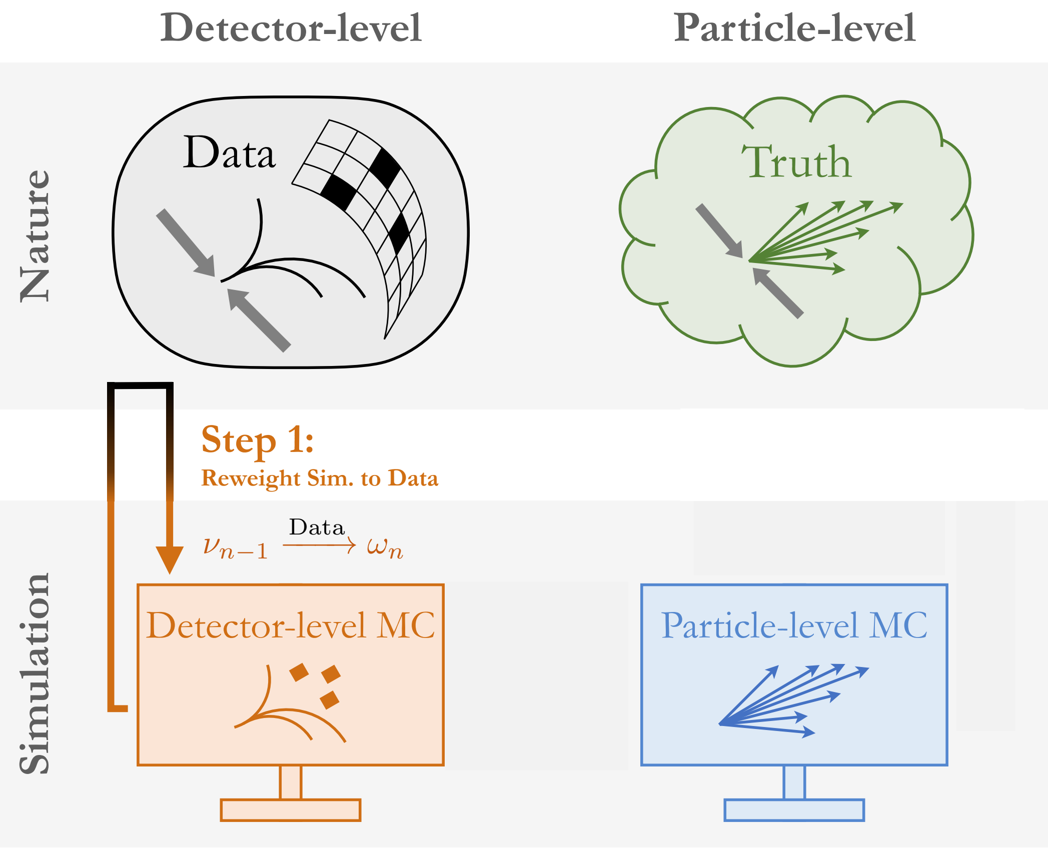
- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks



The MultiFold method

- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

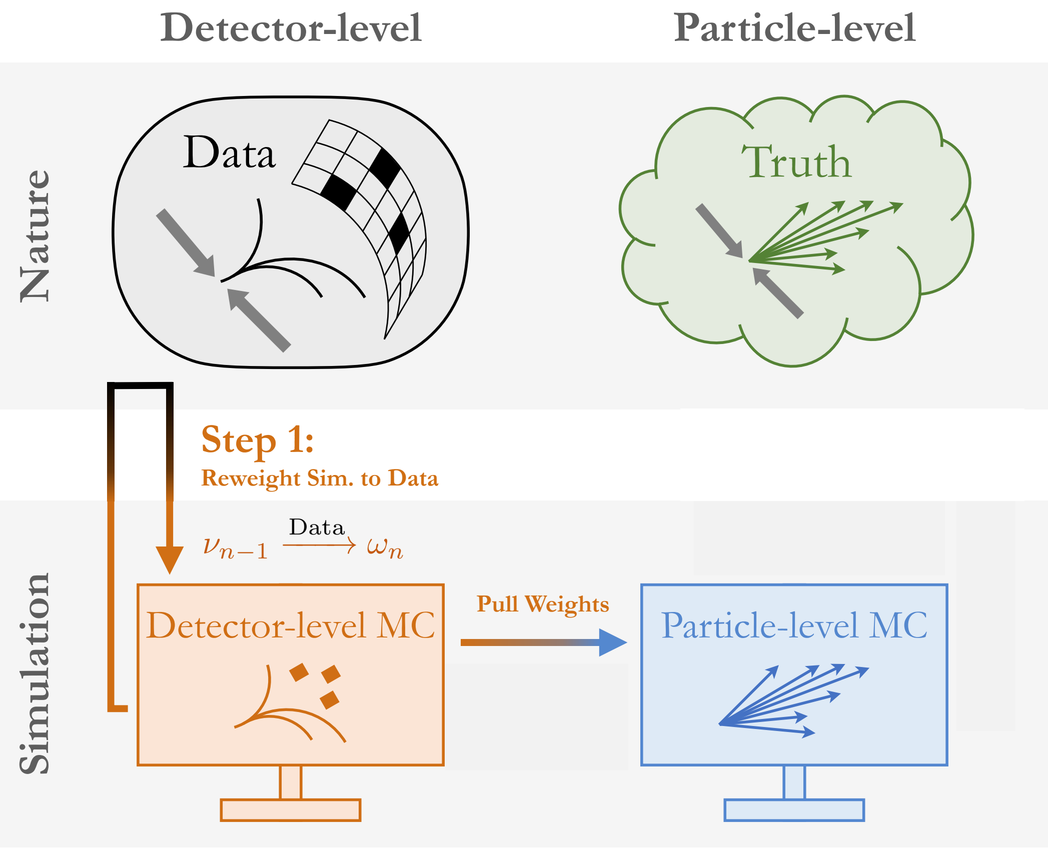


The MultiFold method

- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

Propagate the weights to the particle-level MC simulation



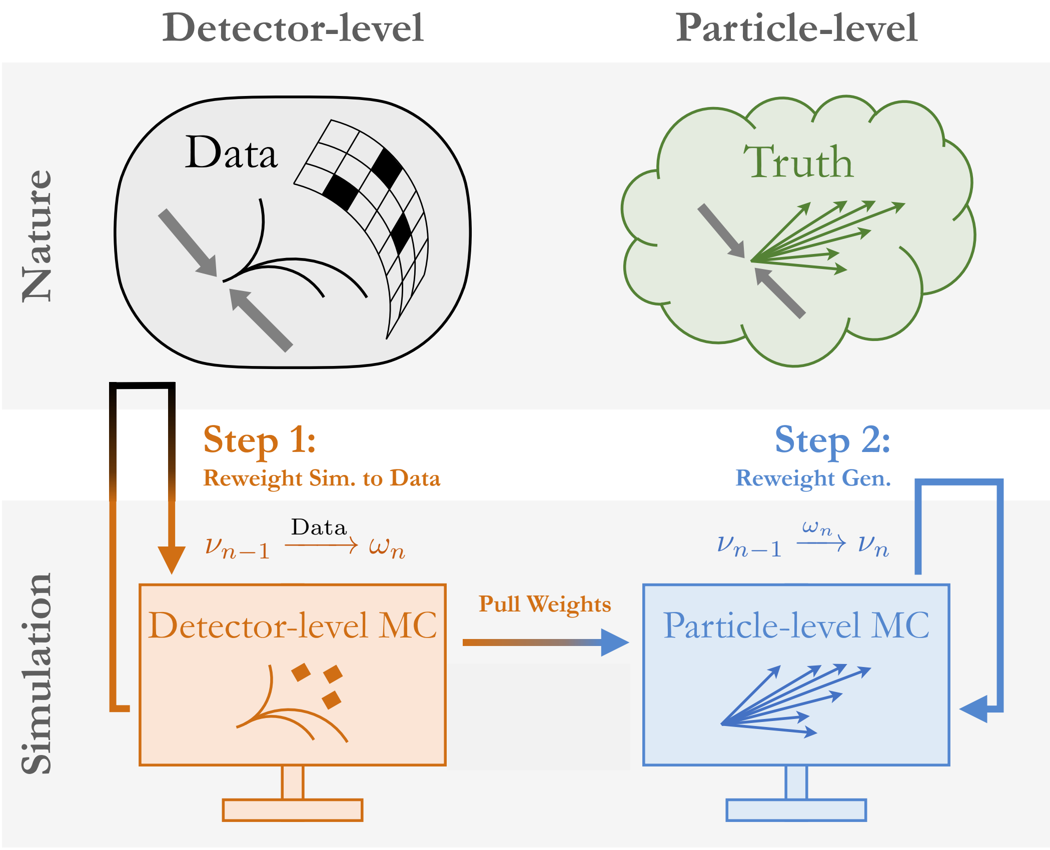
The MultiFold method

- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

Propagate the weights to the particle-level MC simulation

Step 2: Determine a reweighting function that replicates the reweighting in Step 1, but for the particle-level MC



The MultiFold method

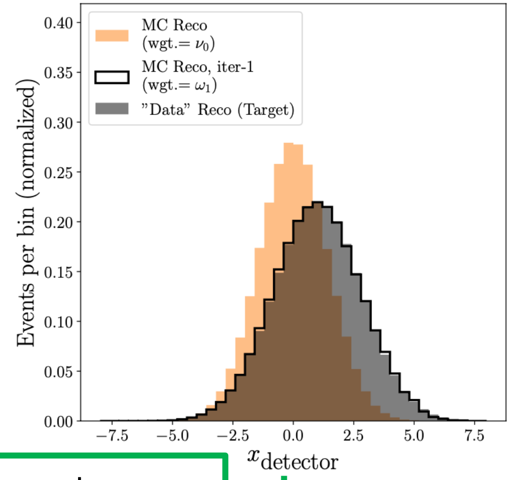
- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

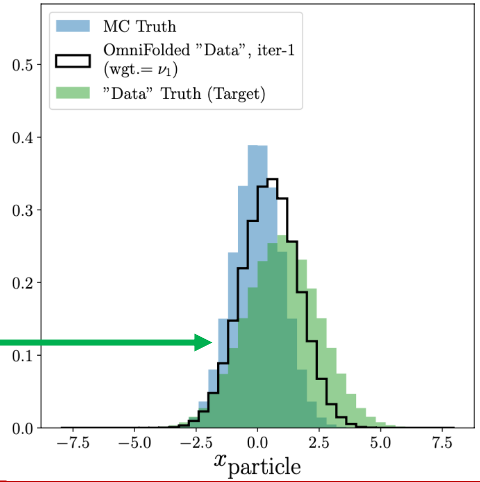
Propagate the weights to the particle-level MC simulation

Step 2: Determine a reweighting function that replicates the reweighting in Step 1, but for the particle-level MC

In the context of the previous example:



Propagate weights from ω to truth-level and replicate function for the truth-level quantities



Solid black line: result after Step 2, reweighting applied to the truth MC should approach the truth "Data"

The MultiFold method

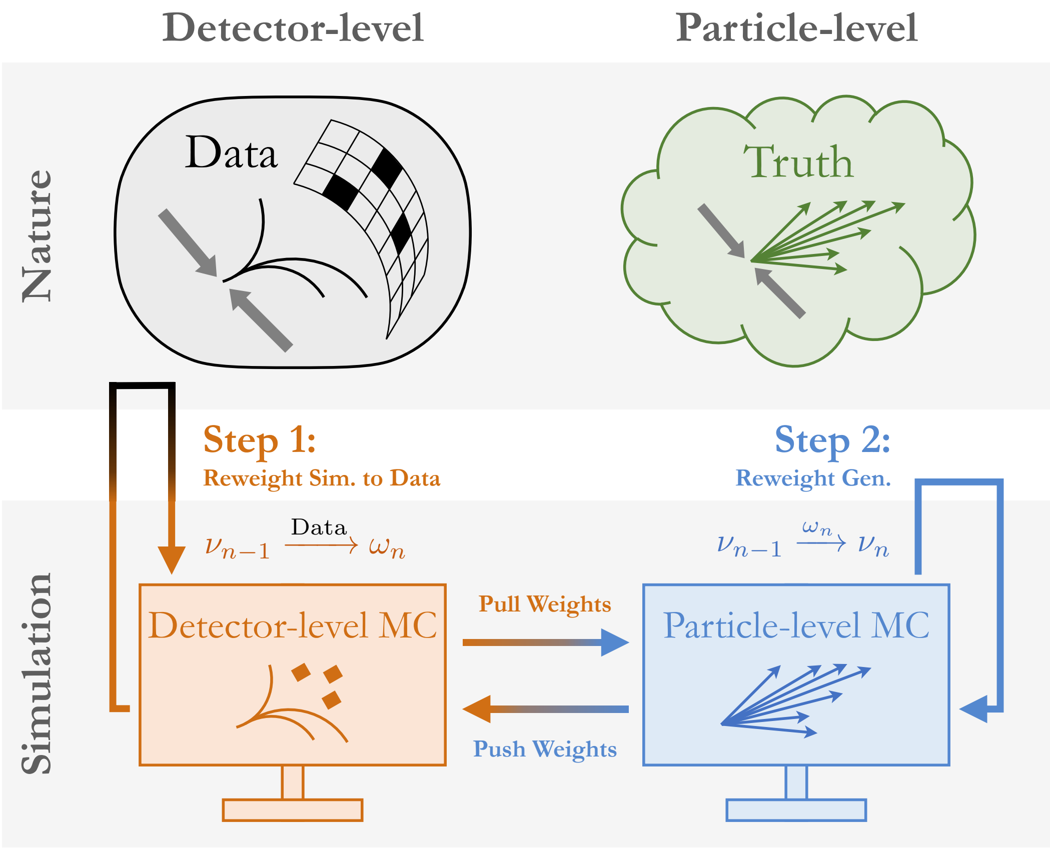
- Current unfolding methods present some challenges
    1. The data must be binned
    2. Can only unfold a small number of observables simultaneously
    3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

Propagate the weights to the particle-level MC simulation

Step 2: Determine a reweighting function that replicates the reweighting in Step 1, but for the particle-level MC

Propagate the weights to the detector-level MC simulation



The MultiFold method

- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks

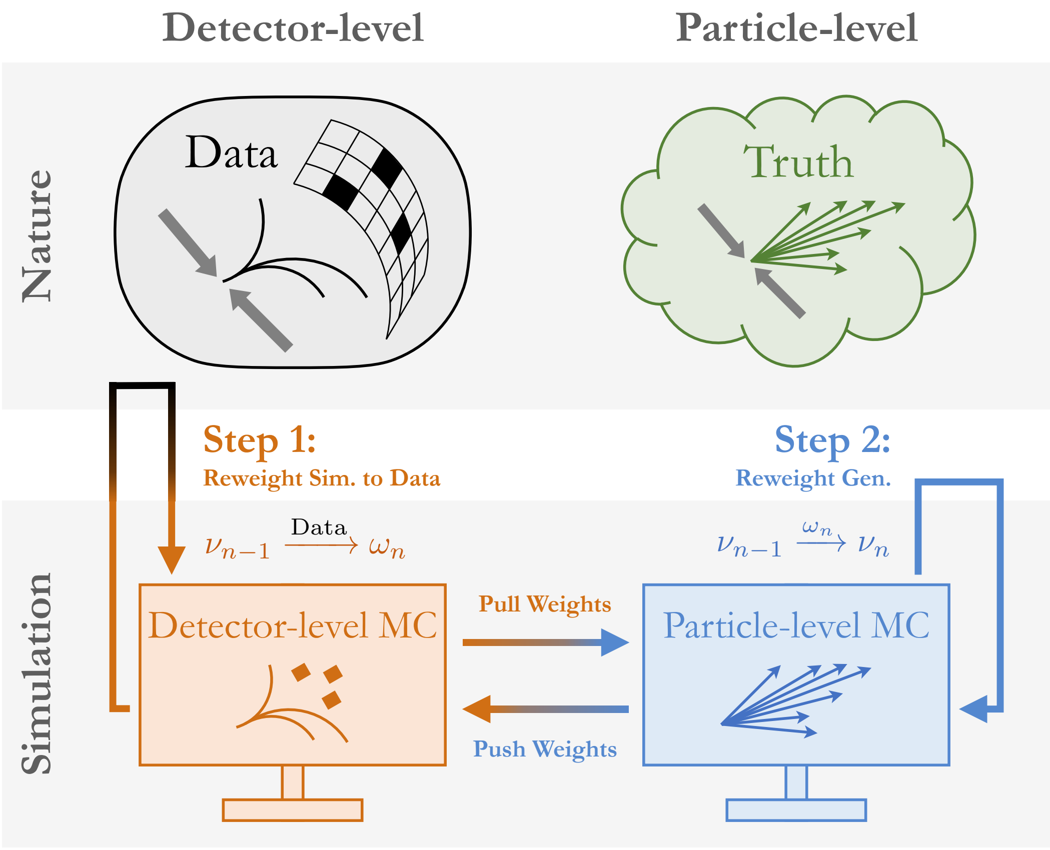
Step 1: Use a neural network to determine a reweighting function that will transform the detector-level MC to match the data for all observables (what we just did!)

Propagate the weights to the particle-level MC simulation

Step 2: Determine a reweighting function that replicates the reweighting in Step 1, but for the particle-level MC

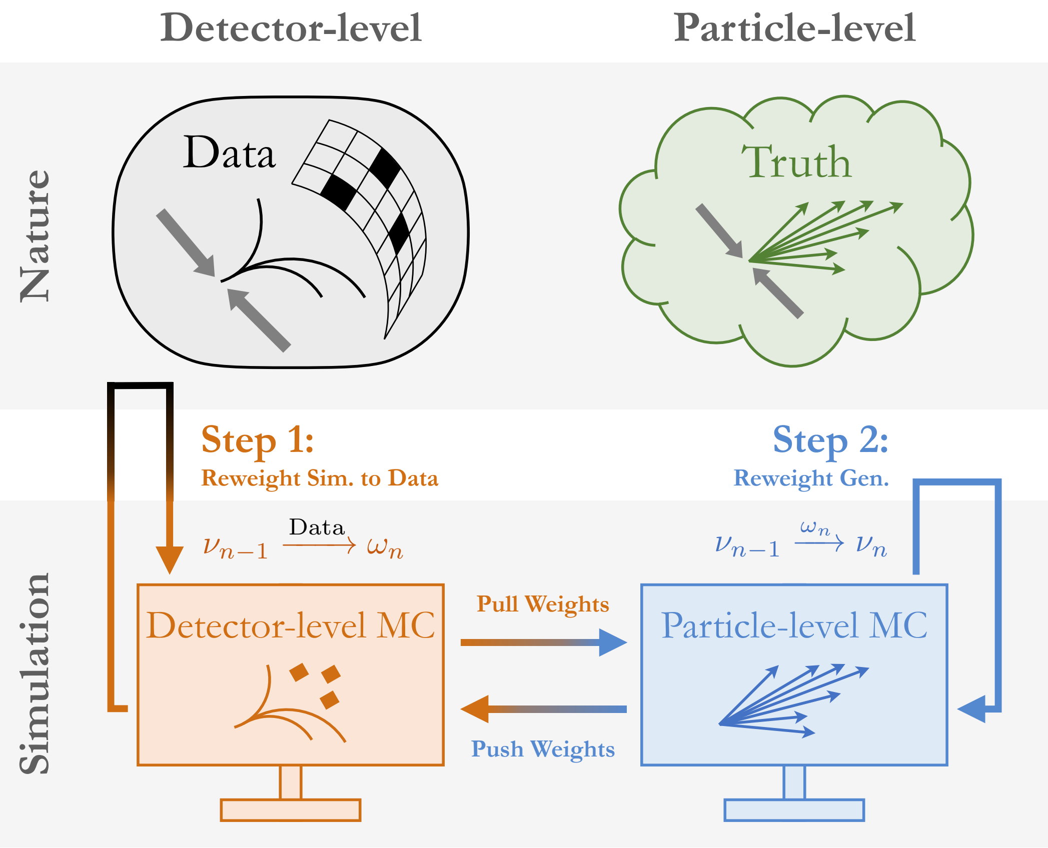
Propagate the weights to the detector-level MC simulation

And iterate…



The MultiFold method

- Current unfolding methods present some challenges
  1. The data must be binned
  2. Can only unfold a small number of observables simultaneously
  3. Do not consider the full phase space and so may miss hidden dependencies
- MultiFold aims to rectify this by performing **unbinned** and **highly-dimensional** unfolding using neural networks



- **Final result after n iterations**: a reweighting function that transforms the particle-level MC into the "unfolded data"

- Measurement is an event sample instead of a 1D histogram like in a usual unfolding algorithm
  - Unbinned
  - Can calculate observables after unfolding
- For this analysis: 24 observables + weight as input and output (i.e. 24 dimensional unfolding)

Part 3: Results

Technical details

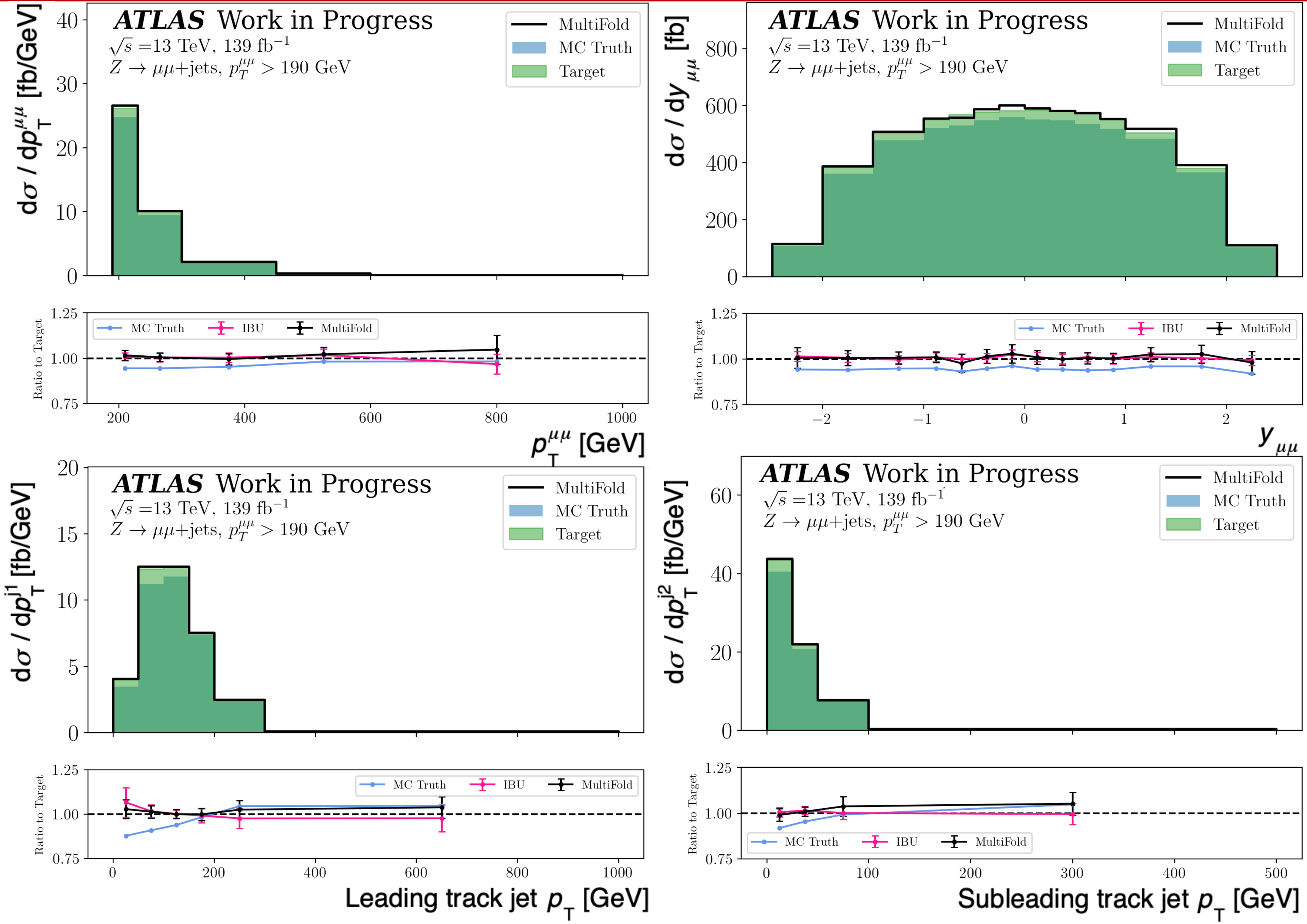
- Results shown here are using pseudodata (fake data with a known answer)
- Differential cross section: related to the rate of events as a function of an observable

- MultiFold results are compared with results obtained with iterative Bayesian unfolding
- IBU is a commonly used algorithm within the ATLAS experiment
  - It comes with the drawbacks mentioned with respect to current unfolding methods

- Note that although results are shown in binned one-dimensional histograms, the MultiFold result is completely unbinned and multidimensional

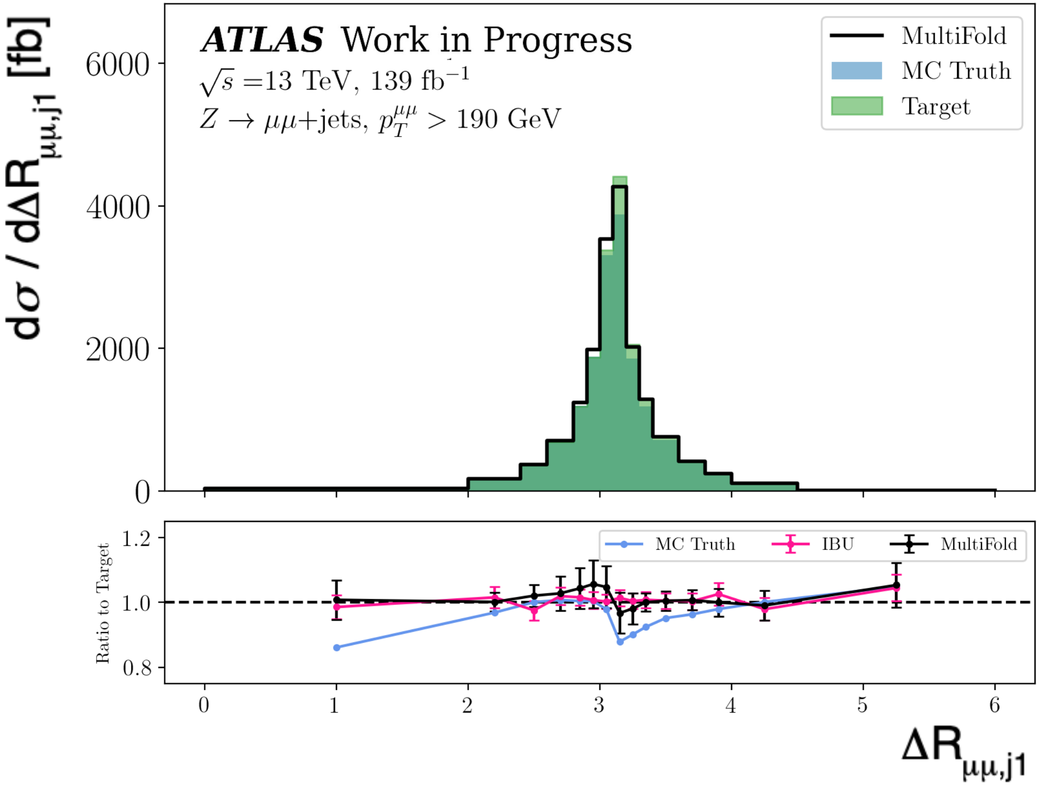
Results: Dimuon and track jet kinematics

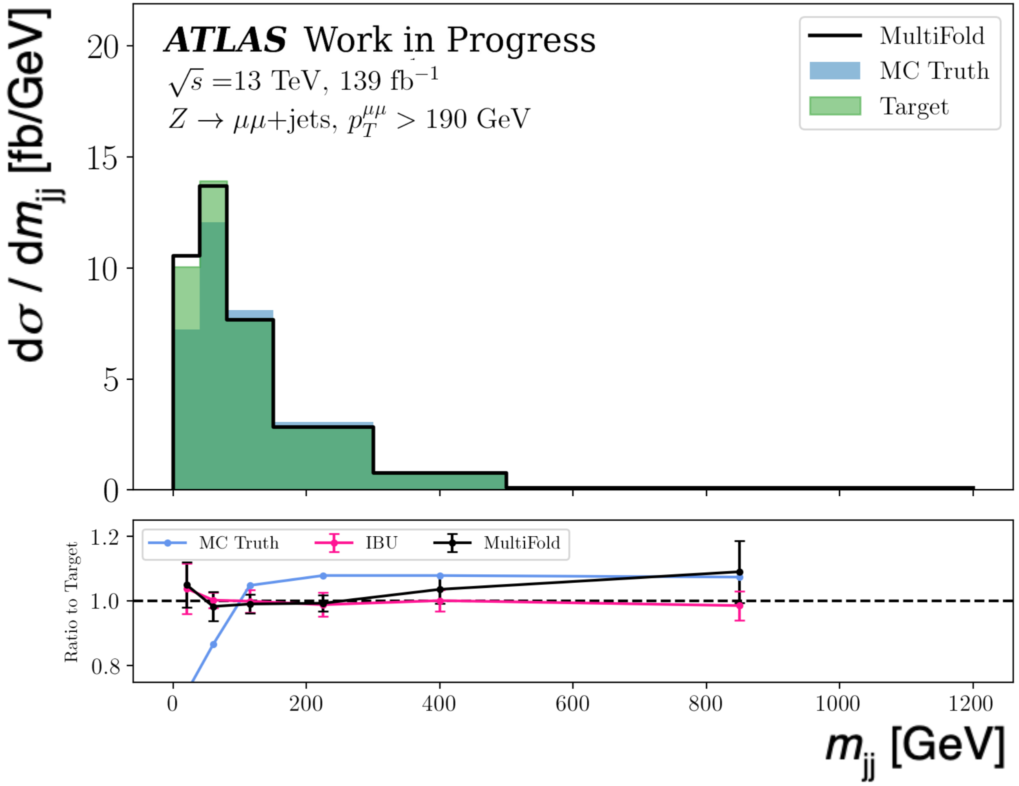
- MultiFold result agrees well with the target
- Performs comparably with IBU, but with all the additional benefits



Results: Derived observables

- One of the main benefits of MultiFold is that **new observables can be calculated after unfolding** without the need to rerun anything
- Requires the quantities needed to construct the observable are available
  - Given that we have included many kinematic quantities, this leaves plenty of options open





- To stress, MultiFold was not trained on these observables but good agreement is still achieved

Conclusion

- Performing a **precision measurement** of $Z(\rightarrow \mu\mu)$+jets production at the LHC

- Unfolding is a procedure to correct measurements for detector effects
  - Needed to compare results with other experiments or theory predictions
- **MultiFold** is a new method that can perform **unbinned** and **highly dimensional** unfolding
  - Basic principle built on reweighting samples using density ratio estimation
  - Unbinned: final result is a list of events with a weight, user can construct any binning
  - High-dimensional: 24 observables available in events
- Applying to ATLAS data for the first time

- **Results shown for pseudodata have good agreement with the target**
- **Observables derived from the MultiFold results but not seen by the algorithm also perform well (exciting!)**

Thanks for listening! Questions?

Backup Slides

Detailed event selection

- Event must be of good quality
- Pass single muon trigger
- Muons: at least 2 muons with opposite charge, $m_{\mu\mu}$ within 10 GeV of the Z boson mass
- High $p_{\mathrm{T}}^{\mu\mu}$ cut at 190 GeV

- Object selection
  - Muons: $p_{\mathrm{T}} > 25\ \mathrm{GeV}$ and $|\eta| < 2.4$, pass quality and isolation criteria
  - Tracks: $p_{\mathrm{T}} > 500\ \mathrm{MeV}$, pass quality and track to vertex association, not from a muon
  - Track jets: input tracks pass above cuts

Full observable list

- Dimuon transverse momentum and rapidity: $p_{\mathrm{T}}^{\mu\mu}$, $y_{\mu\mu}$
- Leading and subleading muon transverse momentum, pseudorapidity, and azimuthal angle: $p_{\mathrm{T}}^{\mu 1}$, $p_{\mathrm{T}}^{\mu 2}$, $\eta_{\mu 1}$, $\eta_{\mu 2}$, $\phi_{\mu 1}$, $\phi_{\mu 2}$
- Leading and subleading track jet transverse momentum, rapidity, azimuthal angle and mass:
  - For the leading track jet: p_{T}^{j1}, y_{j1}, ϕ_{j1}, m_{j1}
  - For the subleading track jet: p_{T}^{j2}, y_{j2}, ϕ_{j2}, m_{j2}
- Leading and subleading track jet substructure:
  - The number of tracks (constituents) in the jet: n_{ch}^{j1}, n_{ch}^{j2}
  - N-subjettiness variables: τ_1^{j1}, τ_2^{j1}, τ_3^{j1}, τ_1^{j2}, τ_2^{j2}, τ_3^{j2}
    - τ_N gives a measure of how likely it is that the jet in question is made up of N subjets

Measuring the differential cross section

- The fiducial cross section for a particular process, X, is related to the probability that such a process will occur inside a defined phase space

$$\sigma_X = \frac{N_X}{\mathcal{L}}$$

Number of events observed

Total integrated luminosity

- If we lived in a world with a perfect detector, that would be that
- Unfortunately, have detector effects to contend with, **this is where unfolding comes in**

$$\sigma_{Z\rightarrow\mu\mu} = \frac{\mathcal{U}(N_{Z\rightarrow\mu\mu})}{\mathcal{L}}$$

Number of events after correcting for detector effects using an unfolding algorithm

- In the following, will express results as a **differential cross section** with respect to a certain observable, $\mathcal{O}$:

$$\frac{d\sigma^i_{Z\rightarrow\mu\mu}}{d\mathcal{O}} = \frac{\mathcal{U}(N^i_{Z\rightarrow\mu\mu})}{\mathcal{L}\Delta\mathcal{O}_i}$$

where $\Delta\mathcal{O}_i$ represents the bin width of the ith bin of the histogram for observable $\mathcal{O}$

Full results