Treatment of Detector Systematics via Likelihood-free Inference

NuXtract Workshop, Oct. 3rd, 2023

A paper by:

Alexandra Trettin, University of Manchester
Leander Fischer, DESY
Richard Naab, DESY

arXiv (accepted preprint): 2305.02257
GitHub: LeanderFischer/ultrasurfaces

DESY.

MANCHESTER 1824
The University of Manchester

Introduction: Monte-Carlo Forward-Folding

This nomenclature will be used all throughout the talk!

Produce MC

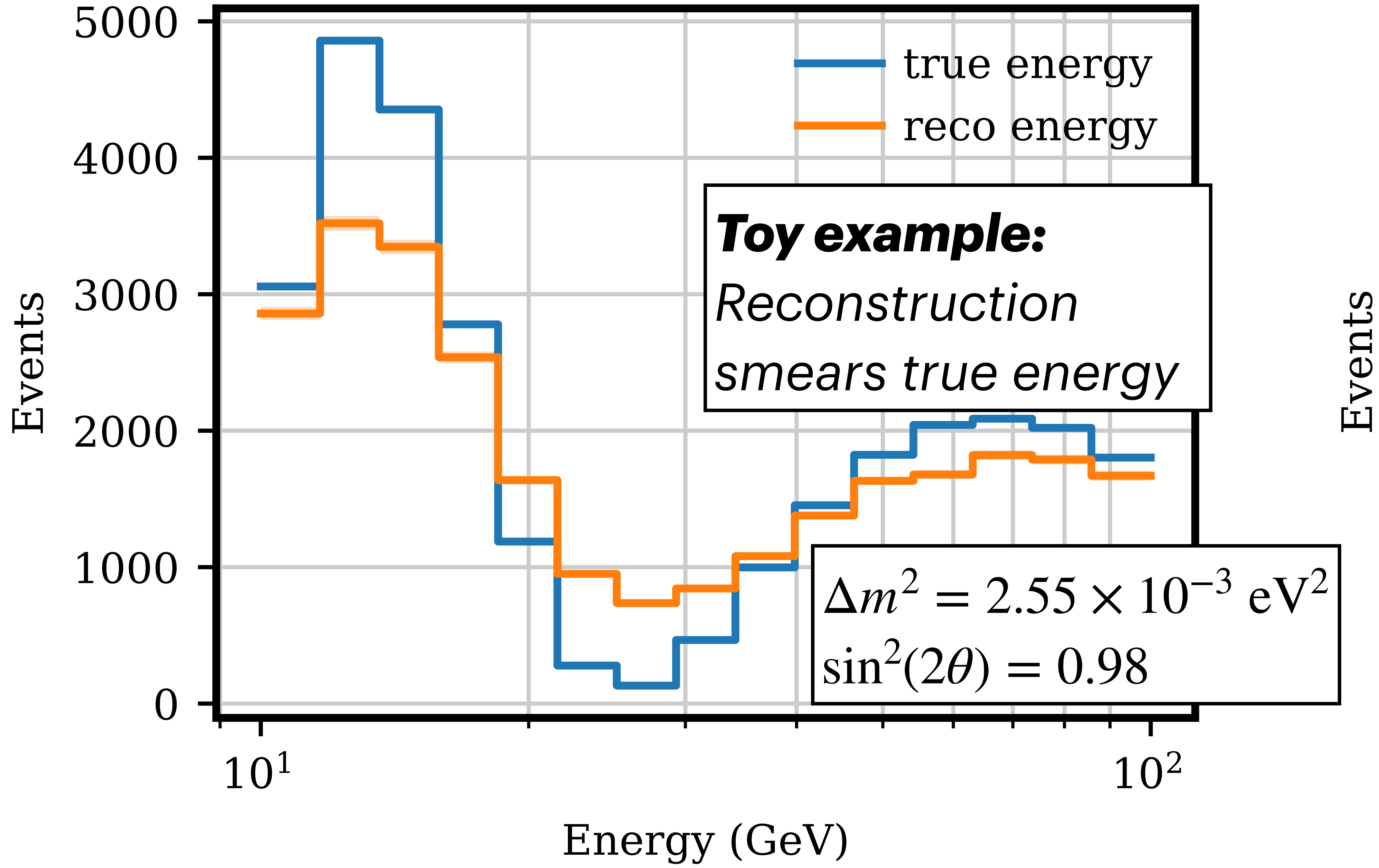
- $\mathbf{x}$ = true event properties (energy, direction, etc.)
- Simulated flux: $\Phi_{\mathrm{sim}}(\mathbf{x})$ (for example power law)

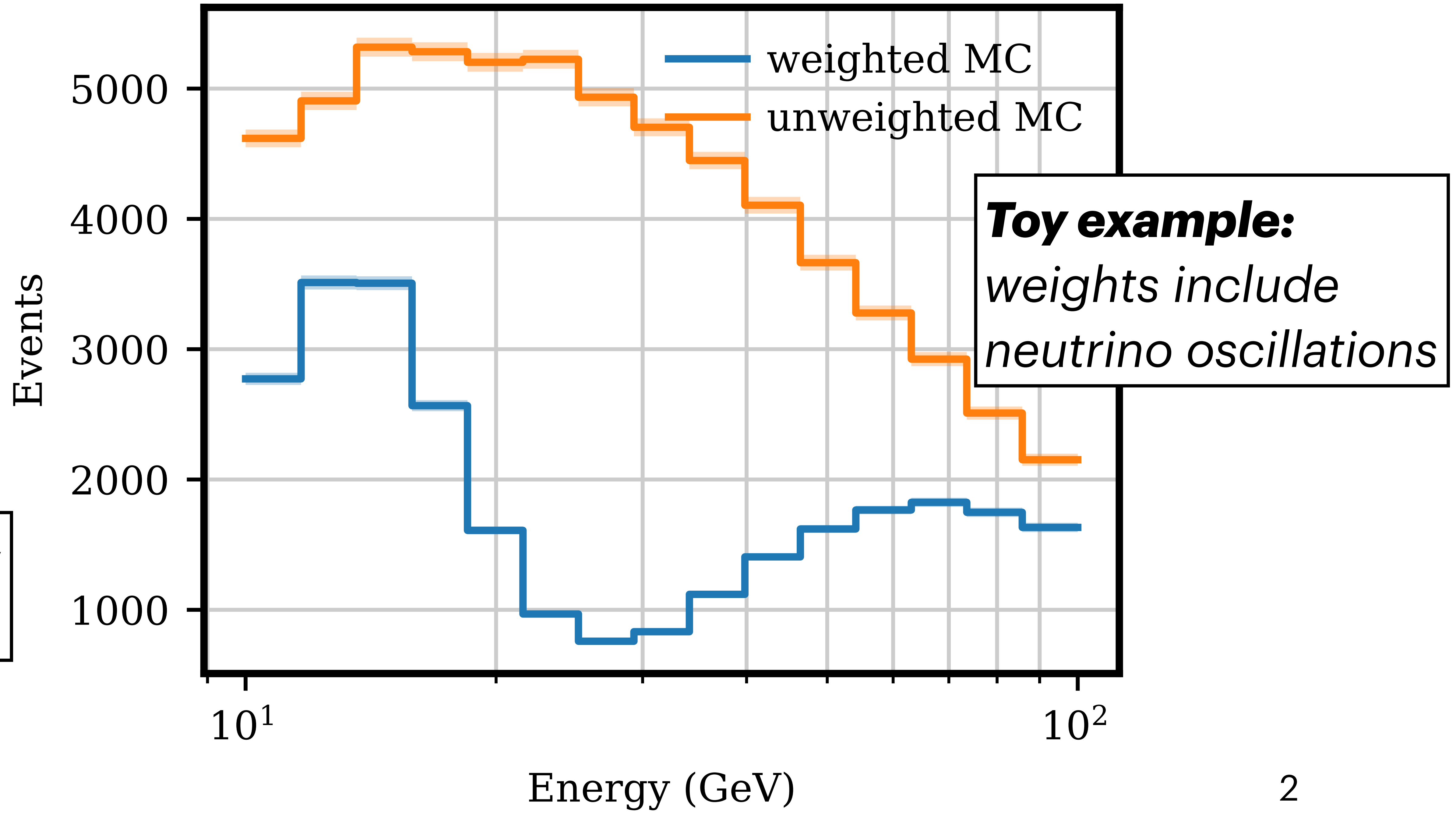
Detector Response

- $\mathbf{y}$ = reco. event properties
- α = detector properties
- $P(\mathrm{acc}\,|\,\mathbf{x}, \alpha)$ = acceptance
- $P(\mathbf{y}\,|\,\mathbf{x}, \alpha)$ = reconstruction

Weight Events

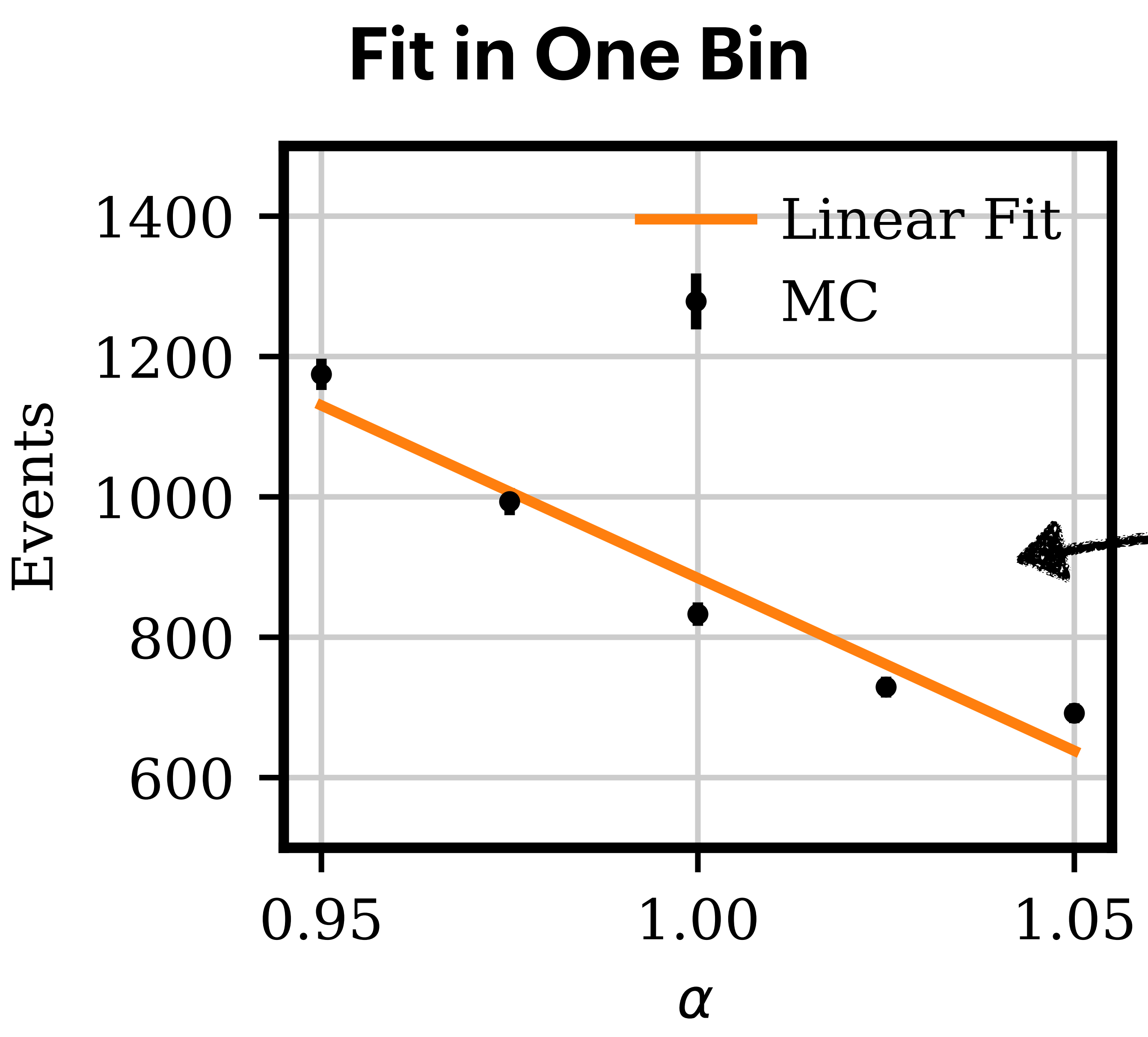
- $w = \Phi(\mathbf{x}\,|\,\theta)/\Phi_{\mathrm{sim}}(\mathbf{x})$
- θ = physics + nuisance parameters

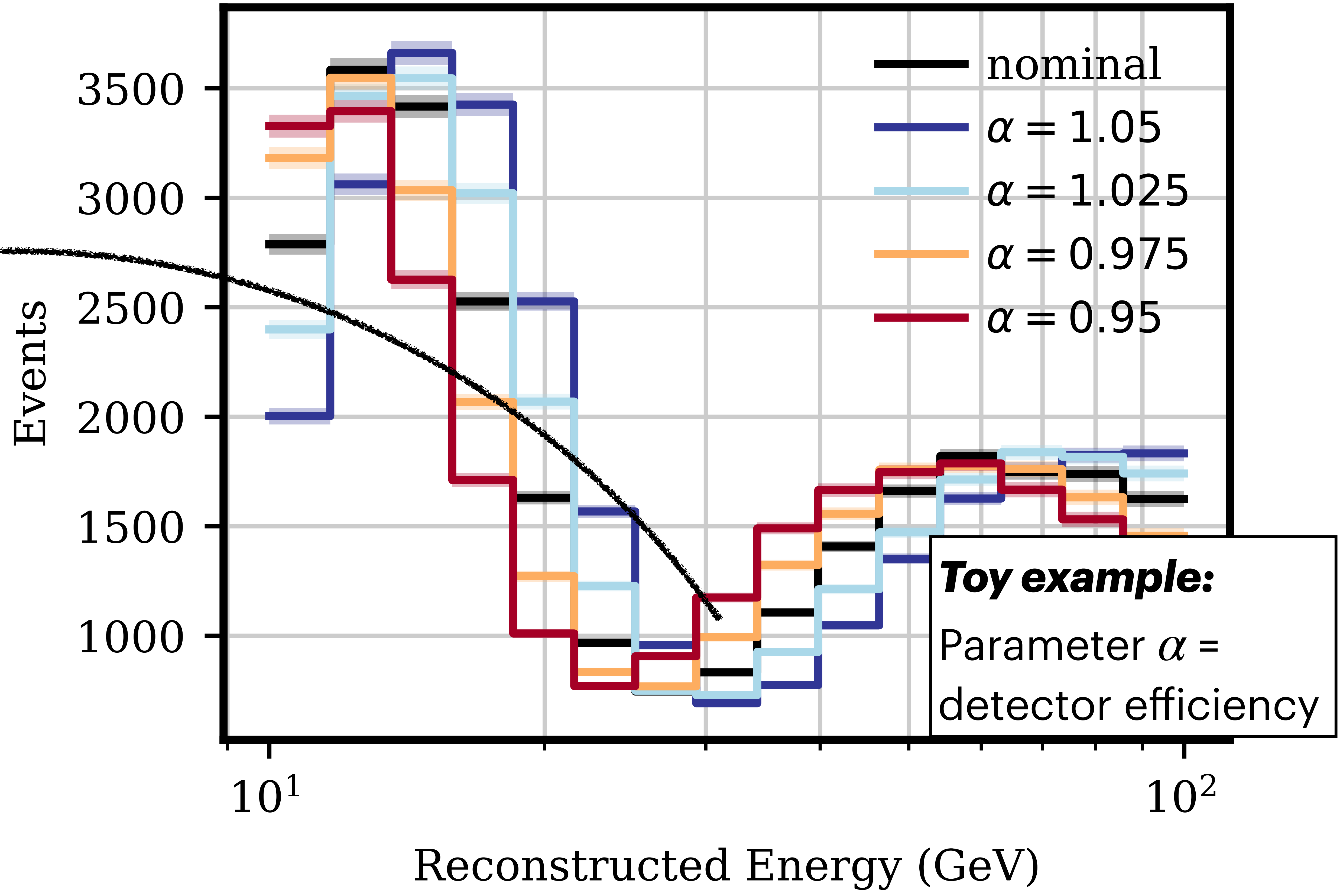




Modeling of Detector Effects

Bin-wise weighting method

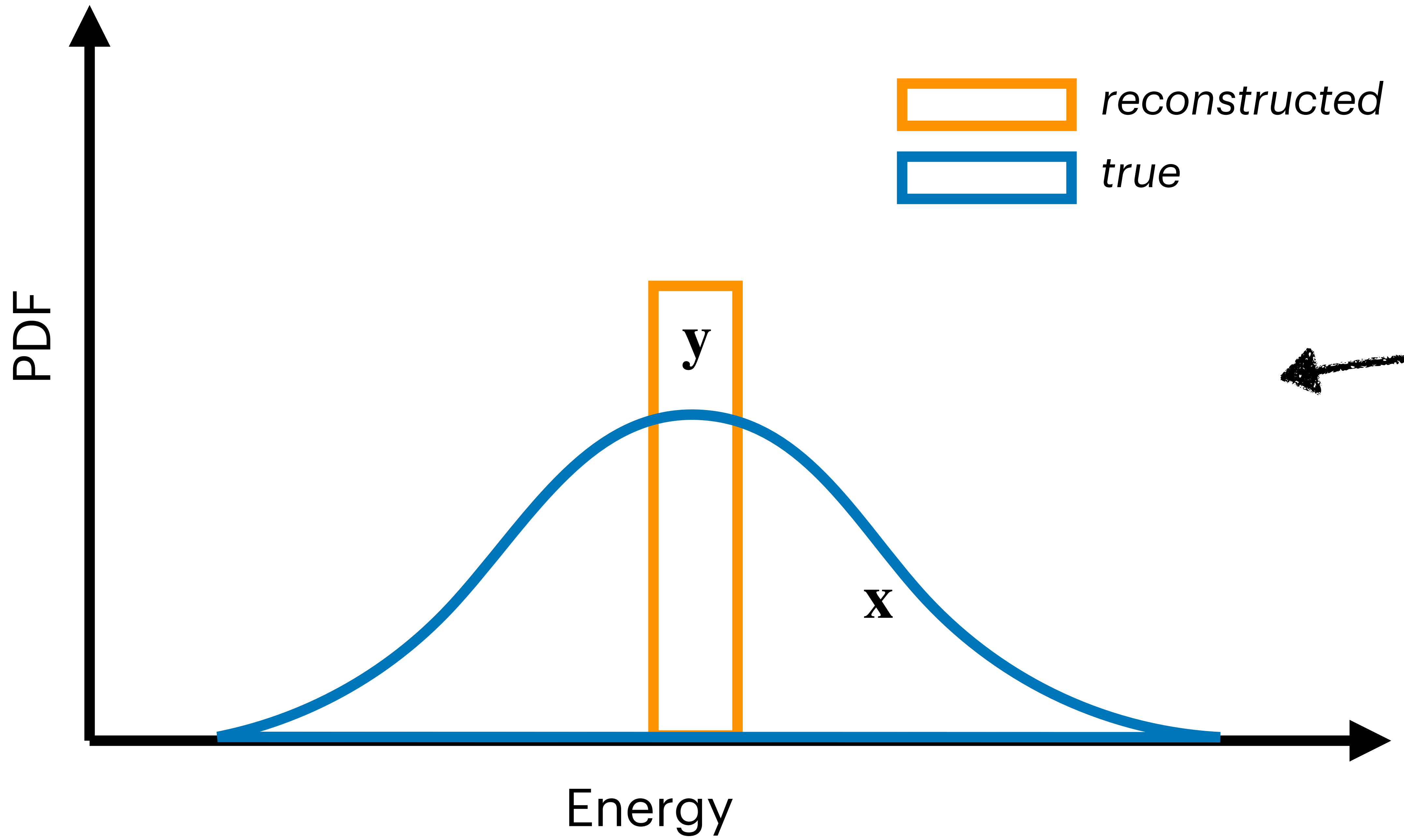


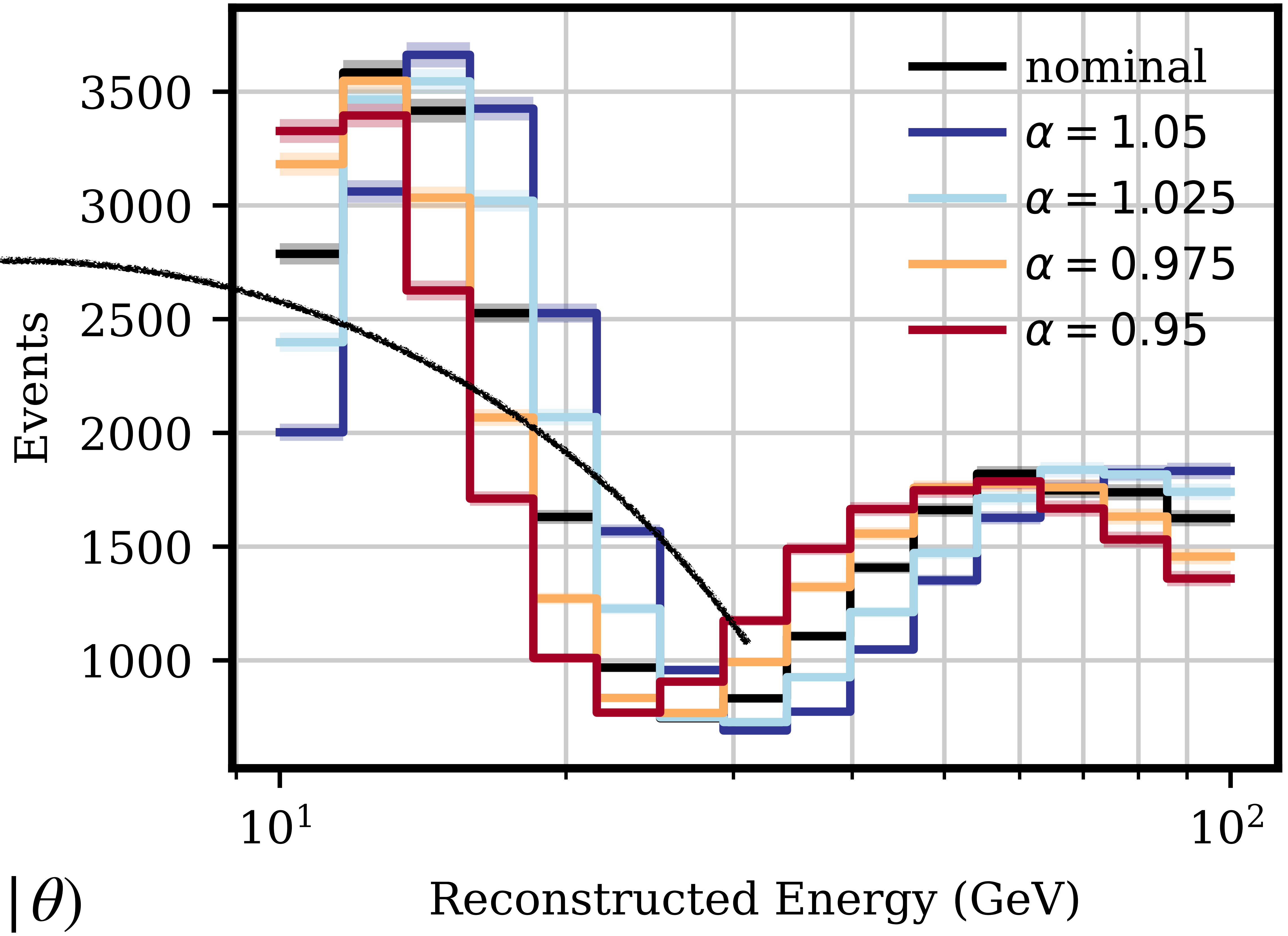


- **Simplest method:** Bin-wise gradients
- **Problem:** Depend on physics parameters θ

Bin-wise Weighting Methods

Implicit Marginalization



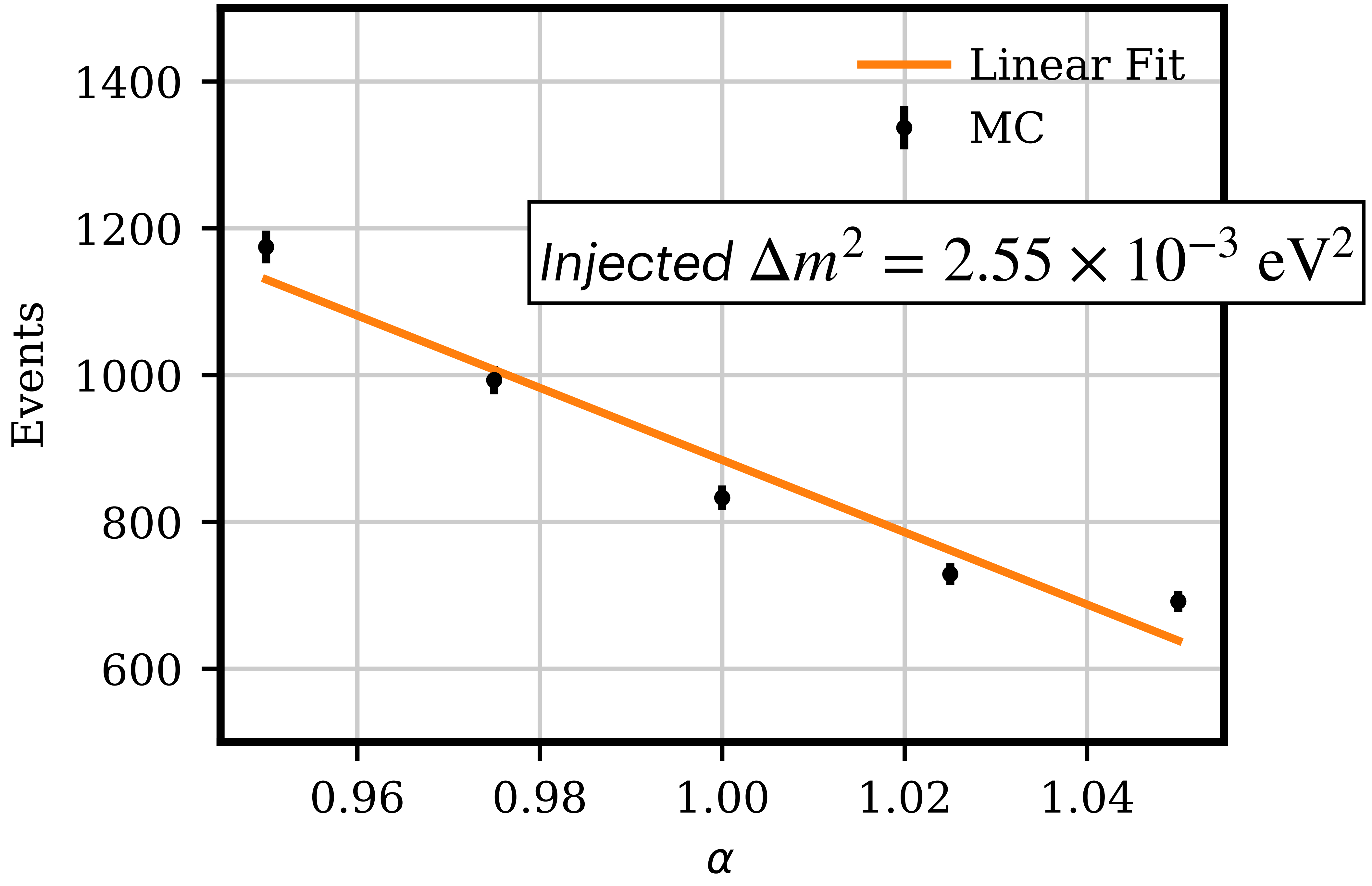


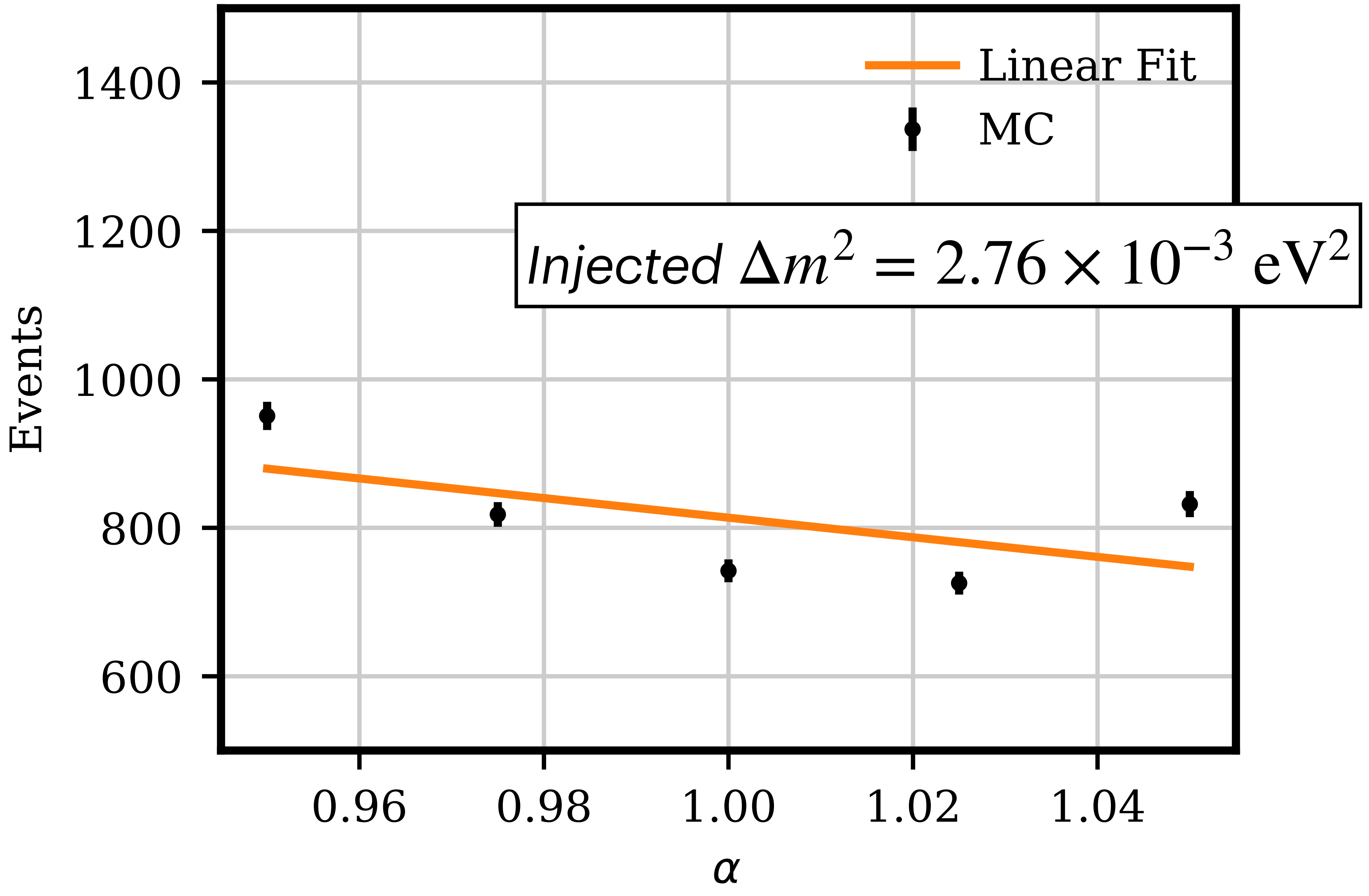
Expectation value in bin i:

$$\mu_i(\theta) = \int_{\boxed{\mathbf{y} \in \text{bin } i}} \mathrm{d}\mathbf{y} \int \mathrm{d}\mathbf{x} P(\mathbf{y} \,|\, \mathbf{x}, \alpha) P(\text{acc} \,|\, \mathbf{x}, \alpha) \frac{\Phi(\mathbf{x} \,|\, \theta)}{\Phi_{\text{sim}}(\mathbf{x})}$$

Bin-wise Weighting Methods

Gradients' dependence on Physics Parameters





Expectation value in bin i:

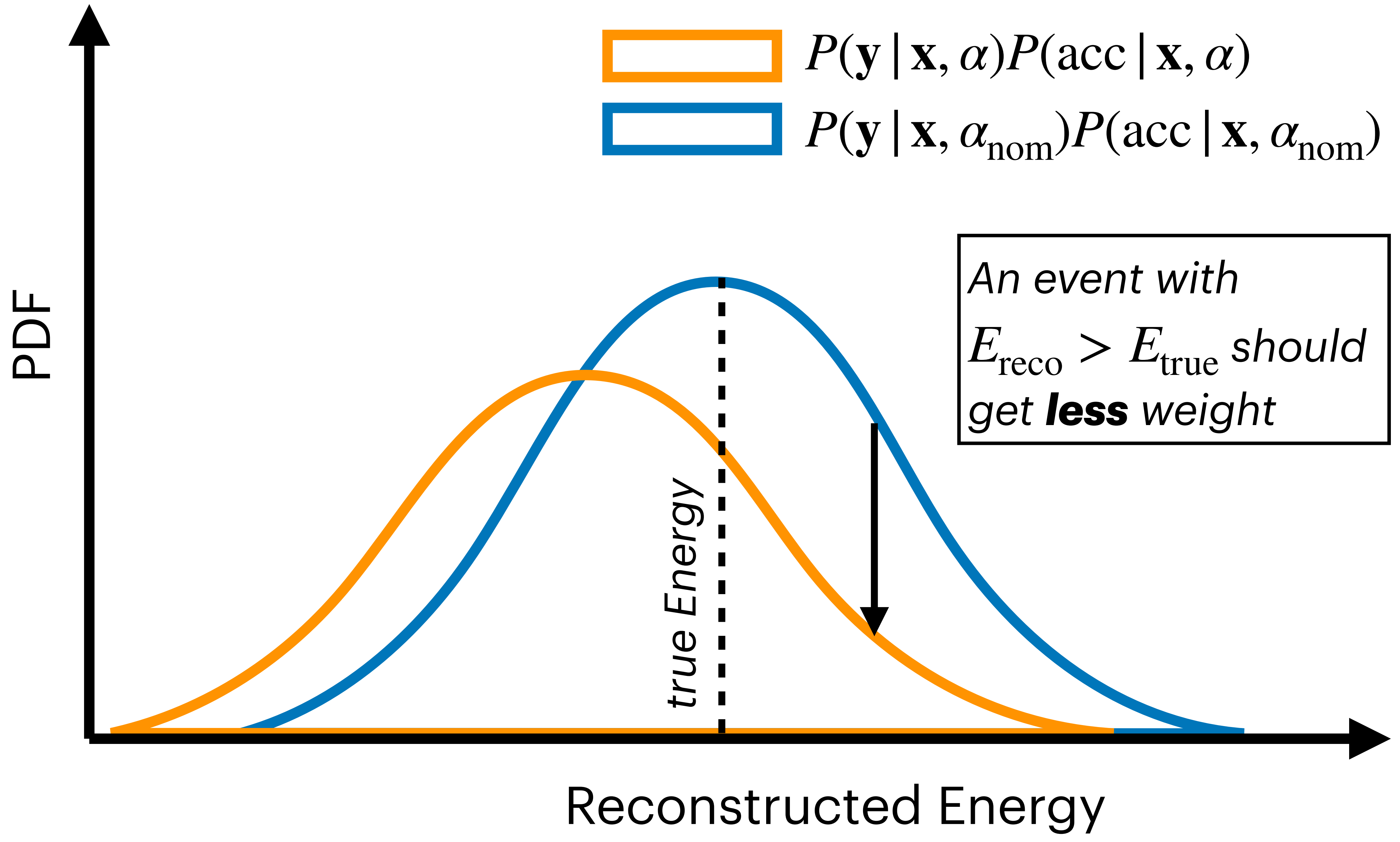
$$\mu_i(\theta) = \int_{\boxed{\mathbf{y} \in \mathrm{bin}\, i}} d\mathbf{y} \int d\mathbf{x} P(\mathbf{y} \,|\, \mathbf{x}, \alpha) P(\mathrm{acc} \,|\, \mathbf{x}, \alpha) \frac{\Phi(\mathbf{x} \,|\, \theta)}{\Phi_{\mathrm{sim}}(\mathbf{x})}$$

Gradient in one bin $\nabla_\alpha \mu_i$ requires integrating over true event properties

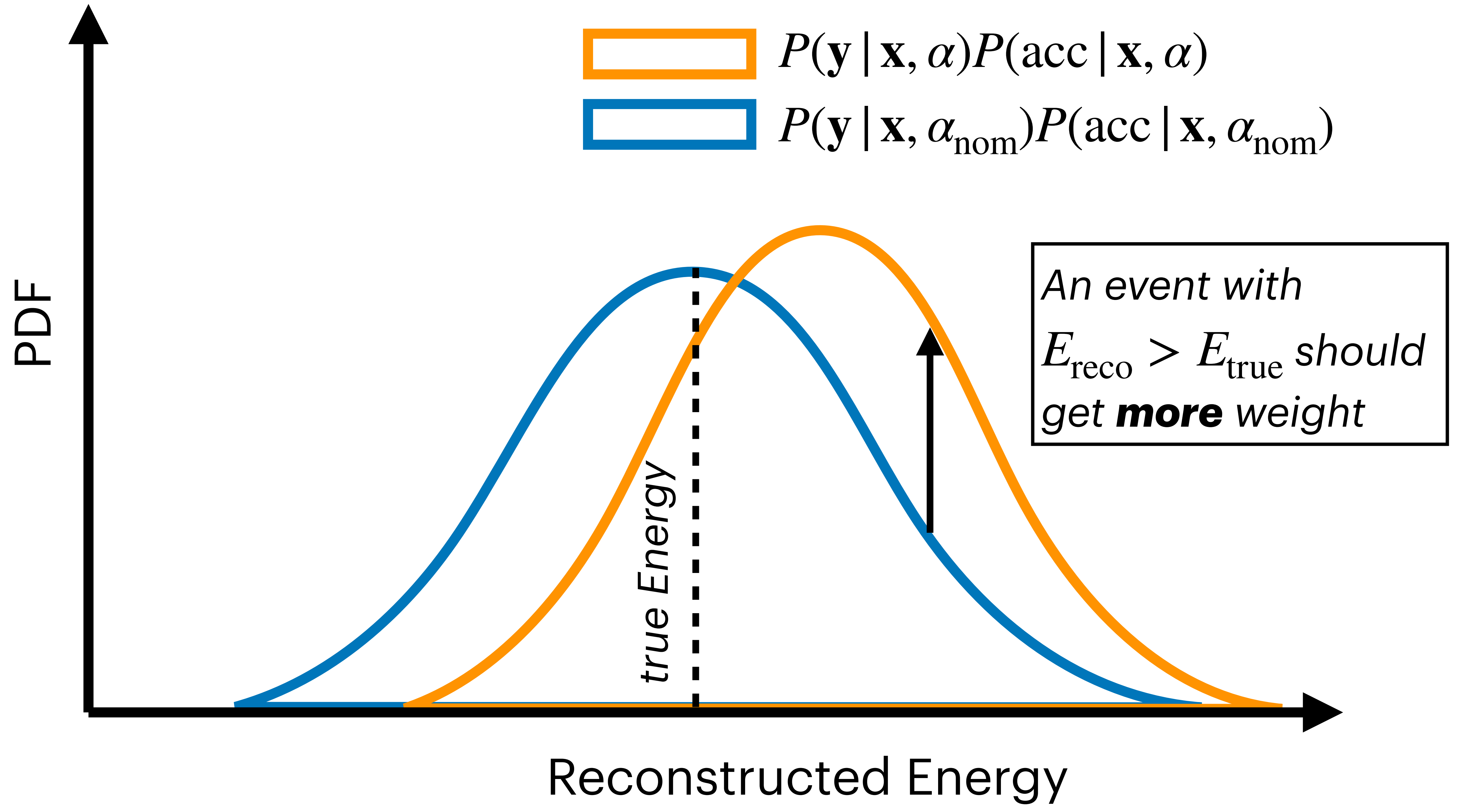
Decoupling Detector Effects

Event Weights Should be Independent from Initial Flux

Example: α = detector efficiency, $\alpha_{\text{nom}} = 1$



$\alpha < 1$: Reconstructed energy tends to be smaller than true energy



$\alpha > 1$: Reconstructed energy tends to be larger than true energy

Decoupling Detector Effects

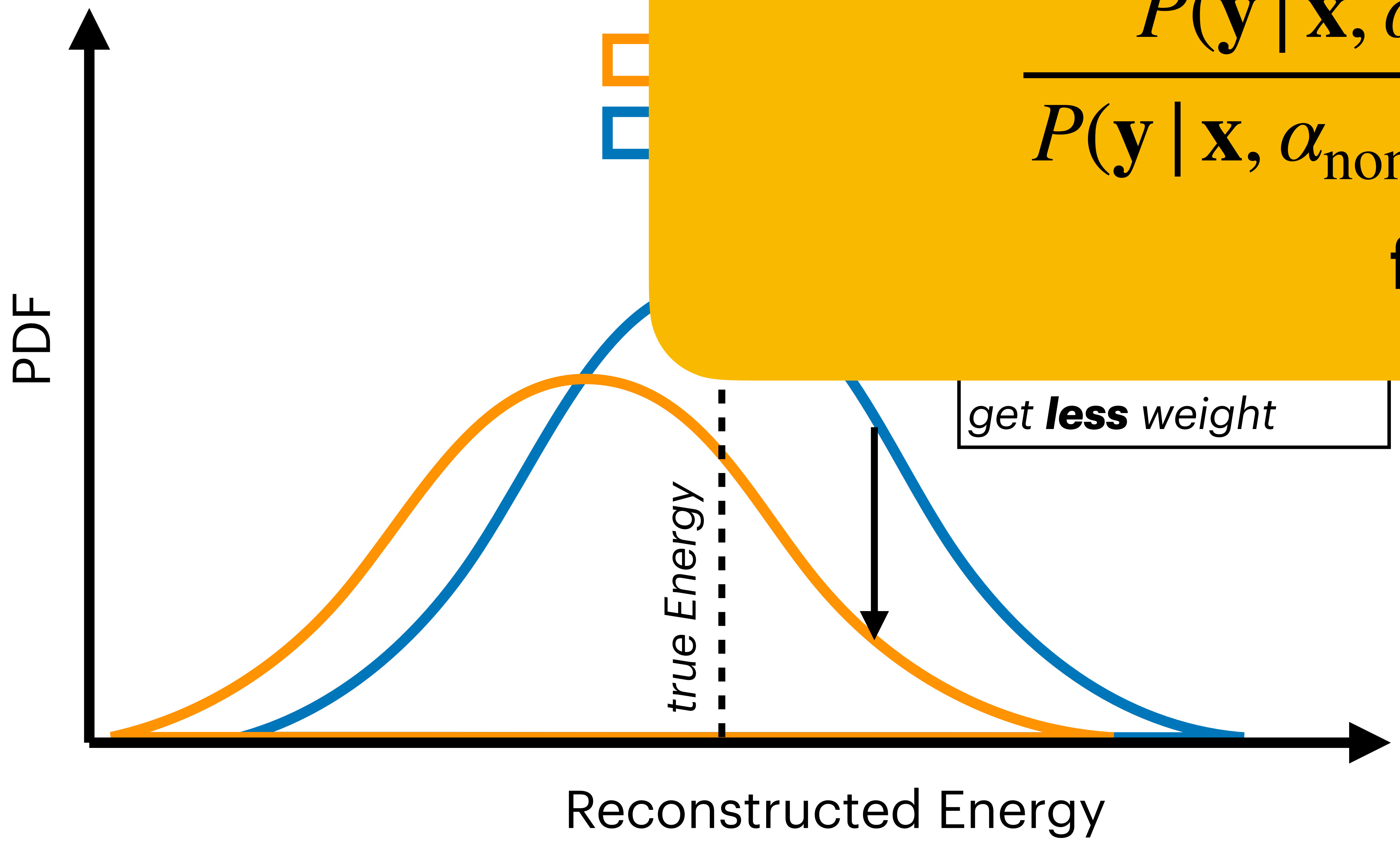
Event Weights Should be Independent from Initial Flux

Example: α = detector

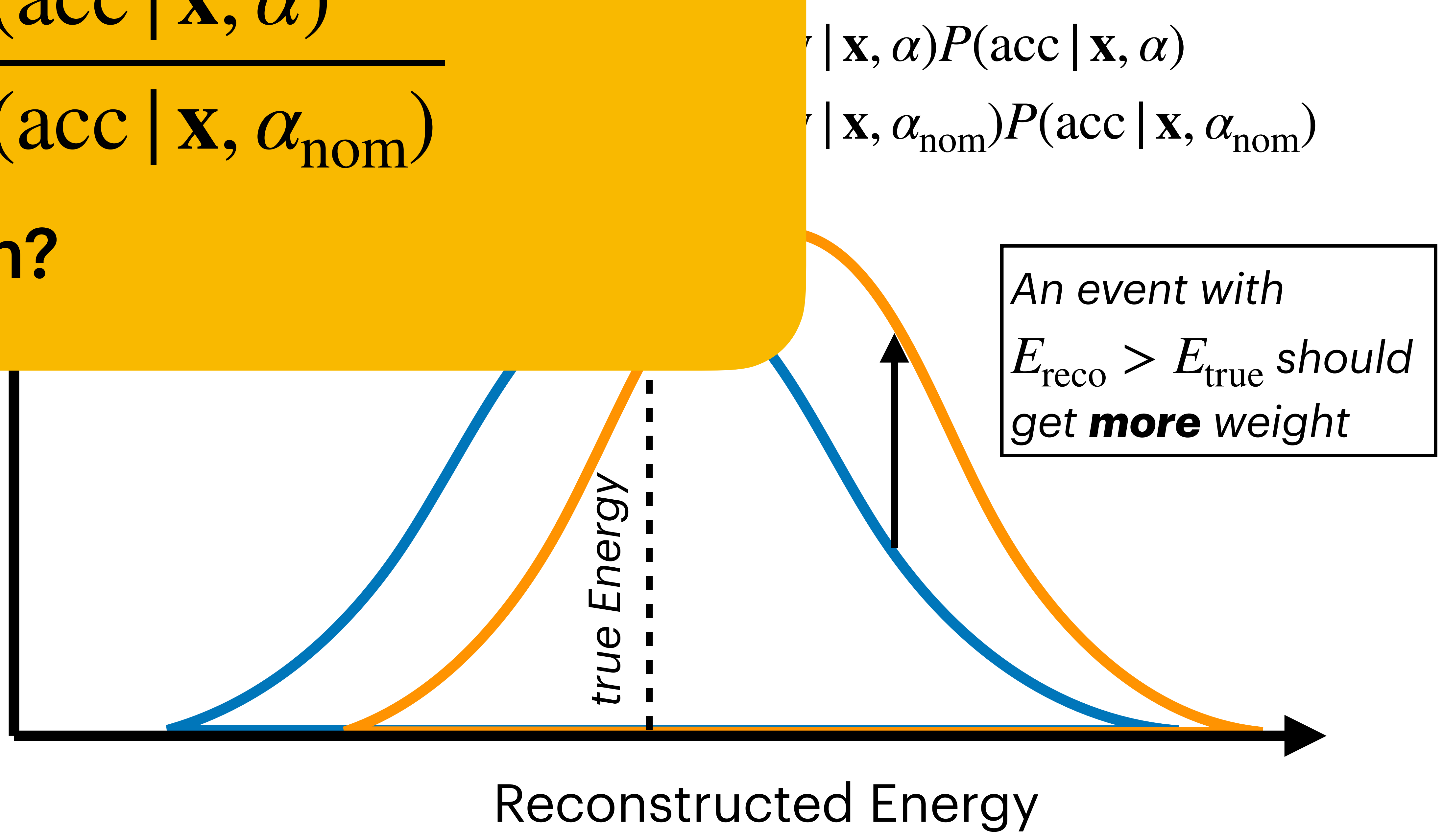
But where do we get

$$\frac{P(\mathbf{y}\,|\,\mathbf{x},\alpha)P(\text{acc}\,|\,\mathbf{x},\alpha)}{P(\mathbf{y}\,|\,\mathbf{x},\alpha_{\text{nom}})P(\text{acc}\,|\,\mathbf{x},\alpha_{\text{nom}})}$$

from?



$\alpha < 1$: Reconstructed energy tends to be smaller than true energy



$\alpha > 1$: Reconstructed energy tends to be larger than true energy

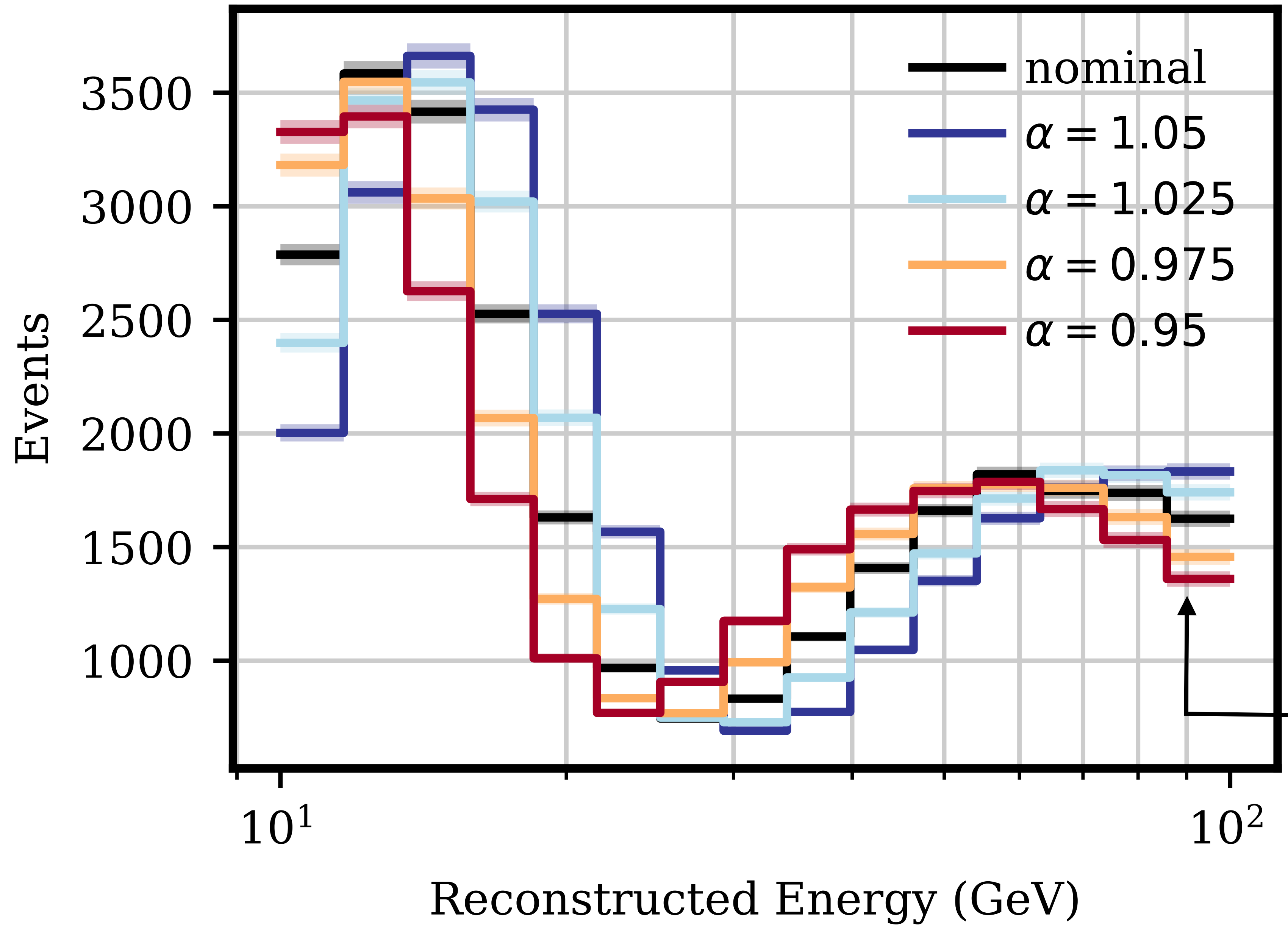
The Likelihood-free Inference Trick

Weighting from Nominal to Any Off-Nominal MC Set

Applying Bayes' Theorem:

index j = index of MC set

$$\frac{P(\mathbf{y}\,|\,\mathbf{x}, \alpha_k)P(\mathrm{acc}\,|\,\mathbf{x}, \alpha_k)}{P(\mathbf{y}\,|\,\mathbf{x}, \alpha_{\mathrm{nom}})P(\mathrm{acc}\,|\,\mathbf{x}, \alpha_{\mathrm{nom}})} = \frac{P(\alpha_k\,|\,\mathbf{x}, \mathbf{y})}{P(\alpha_{\mathrm{nom}}\,|\,\mathbf{x}, \mathbf{y})}$$



Treat each MC set as one discrete class

Just train a classifier to estimate posterior that an event with given $\mathbf{x}, \mathbf{y}$ belongs to set j!

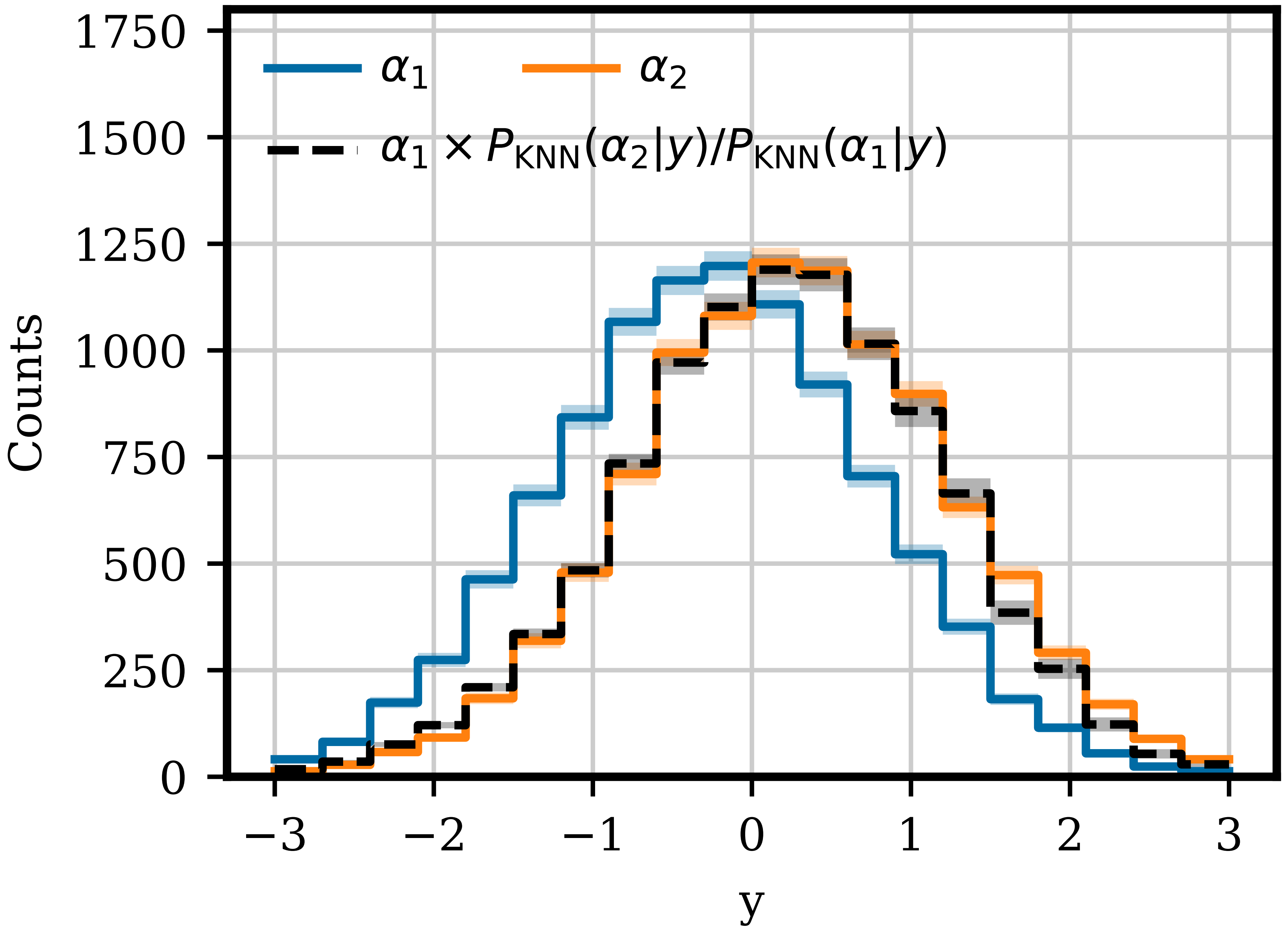
KNN Classifier Example

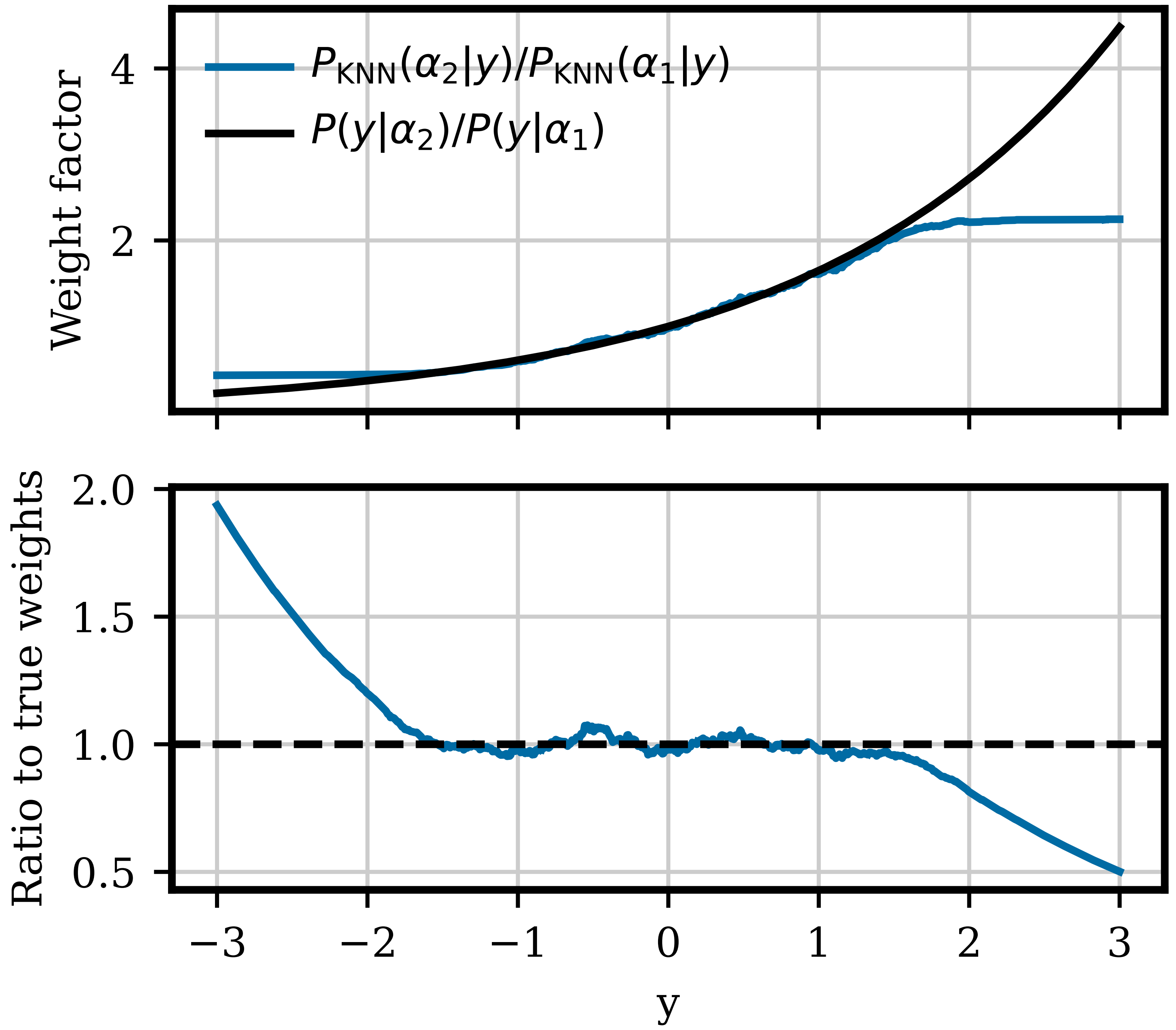
Simple and Robust Posterior Estimate

Sum over indices in neighborhood around $(\mathbf{x}, \mathbf{y})$ belonging to set k

KNN Classifier Equation:

$$P(\alpha = \alpha_k \,|\, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{j \in \mathcal{N}_k(\mathbf{x},\mathbf{y})} 1$$





Making Event-Wise Gradients
Interpolating between Discrete MC Sets

- **Probability estimate** using softmax to normalize

$$\hat{P}(\alpha_k \,|\, \mathbf{x}_j, \mathbf{y}_j) = \text{softmax}\left(\mathbf{g}_j A\right) = \frac{\exp(\sum_n g_{jn} A_{nk})}{\sum_{k'} \exp(\sum_n g_{jn} A_{nk'})}$$

g_{jn} = gradient w.r.t. α_n for event j

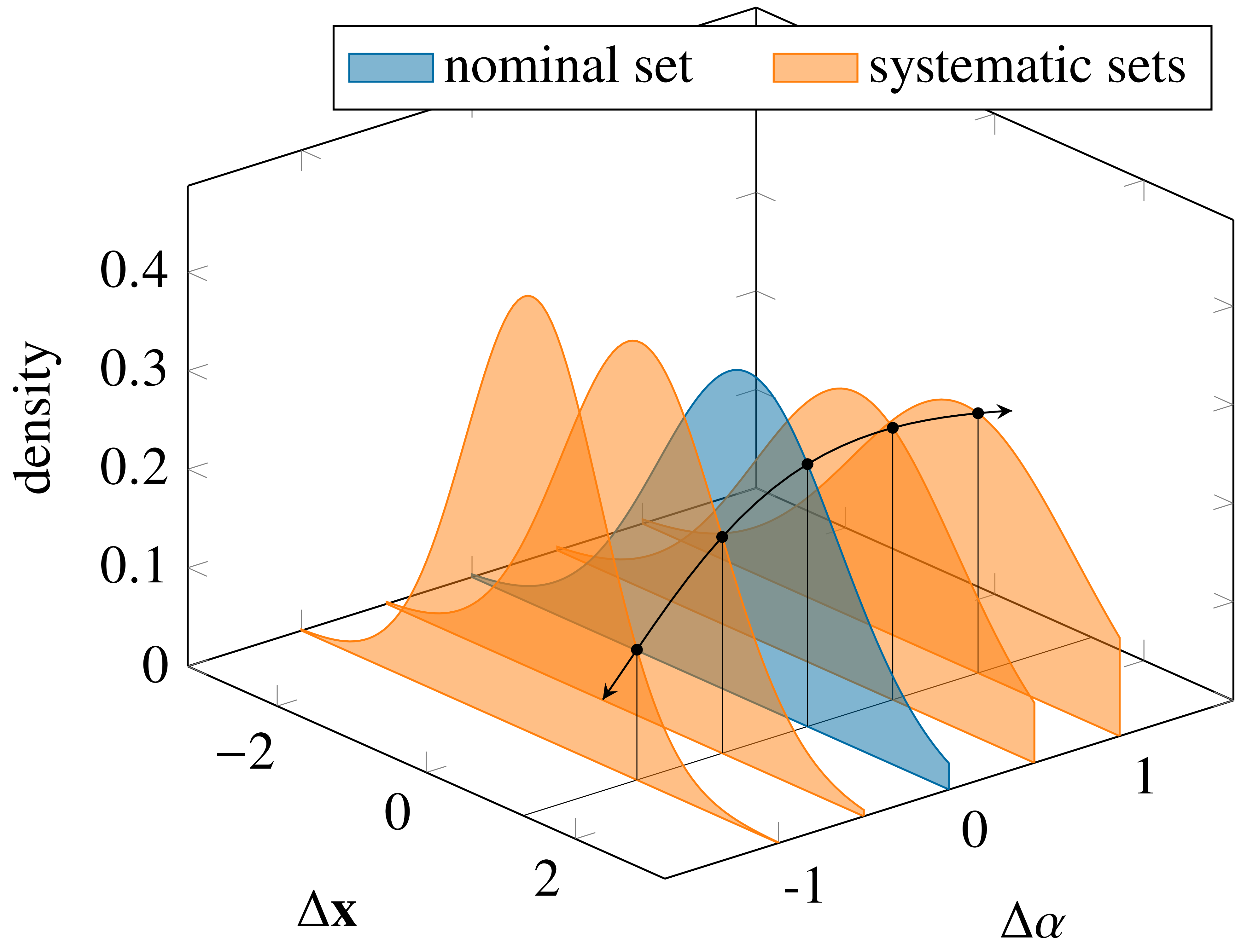
$$A_{nk} = \alpha_{n,k} - \alpha_{n,\,\text{nom}}$$

- **Loss function** to fit gradients g_{jn} is cross-entropy:

$$H_j = -\sum_k \log(\hat{P}(\alpha_k \,|\, \mathbf{x}_j, \mathbf{y}_j)) P_{\text{KNN}}(\alpha_k \,|\, \mathbf{x}_j, \mathbf{y}_j)$$

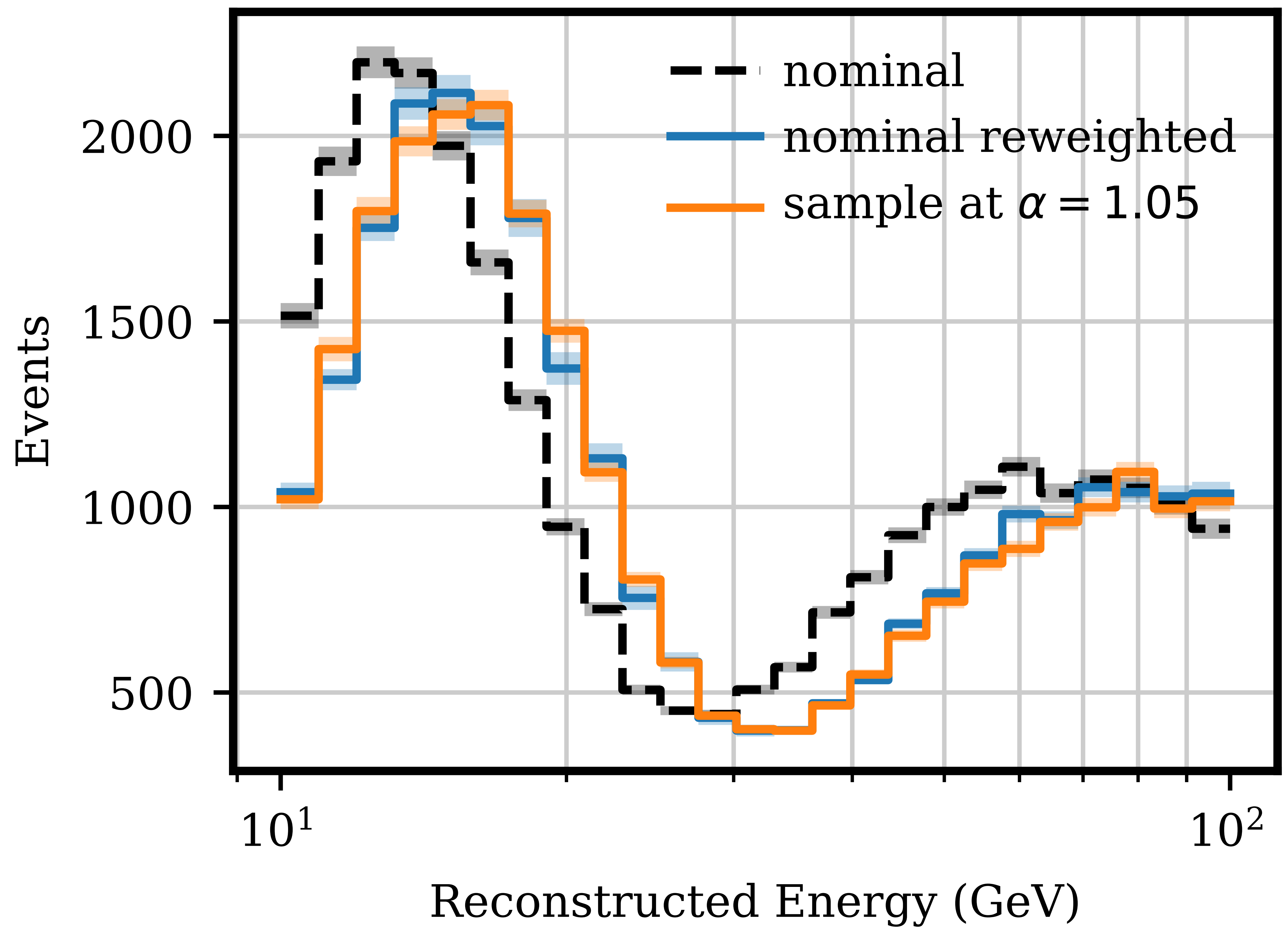
- **Interpolated weight** for every MC event during evaluation

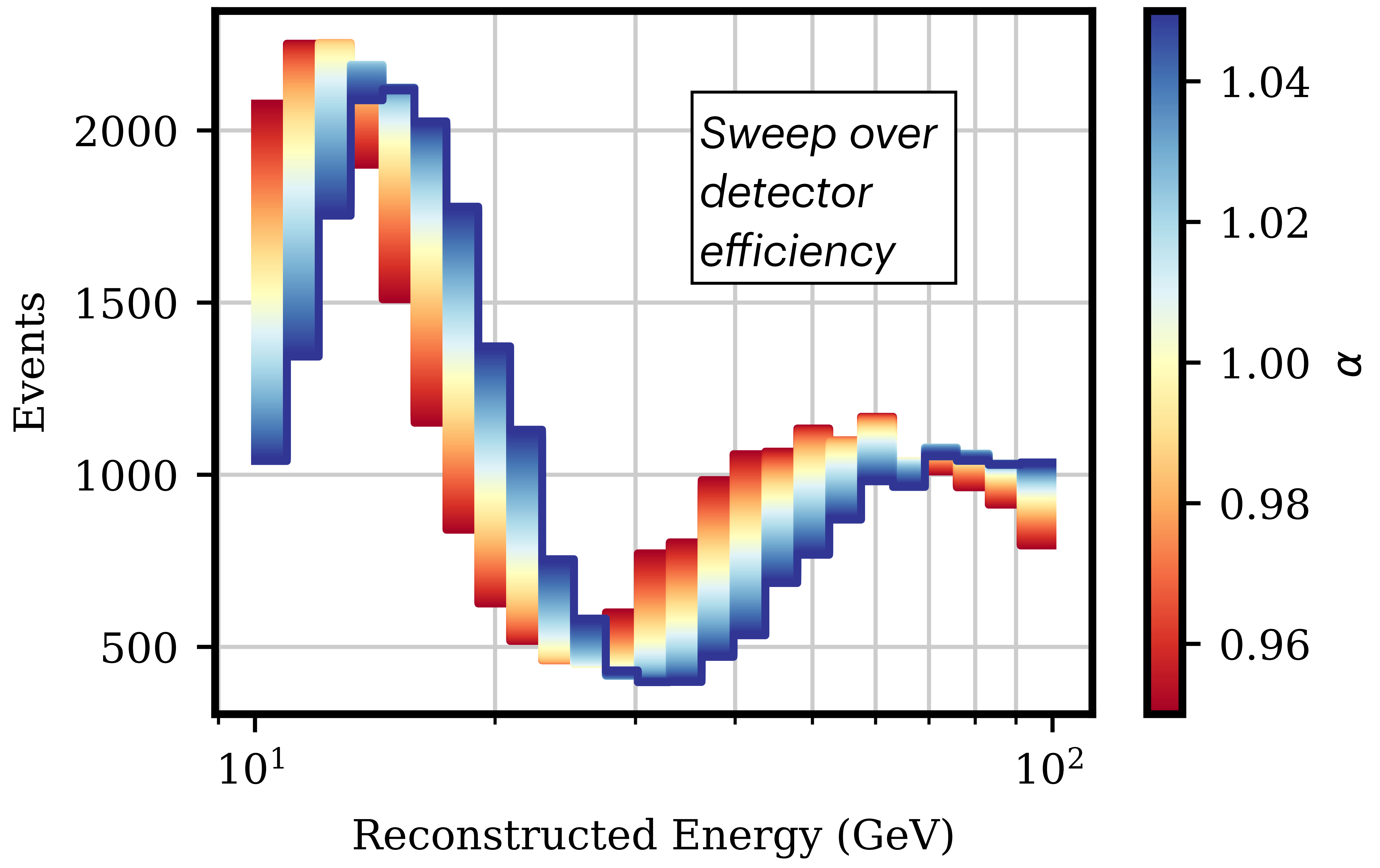
$$\hat{r}_j(\alpha) = \frac{\hat{P}(\alpha \,|\, \mathbf{x}_j, \mathbf{y}_j)}{\hat{P}(\alpha_{\text{nom}} \,|\, \mathbf{x}_j, \mathbf{y}_j)} = \exp\left(\sum_n g_{jn}(\alpha_n - \alpha_{n,\,\text{nom}})\right)$$



Toy MC Example

Reweighting between and in between MC sets

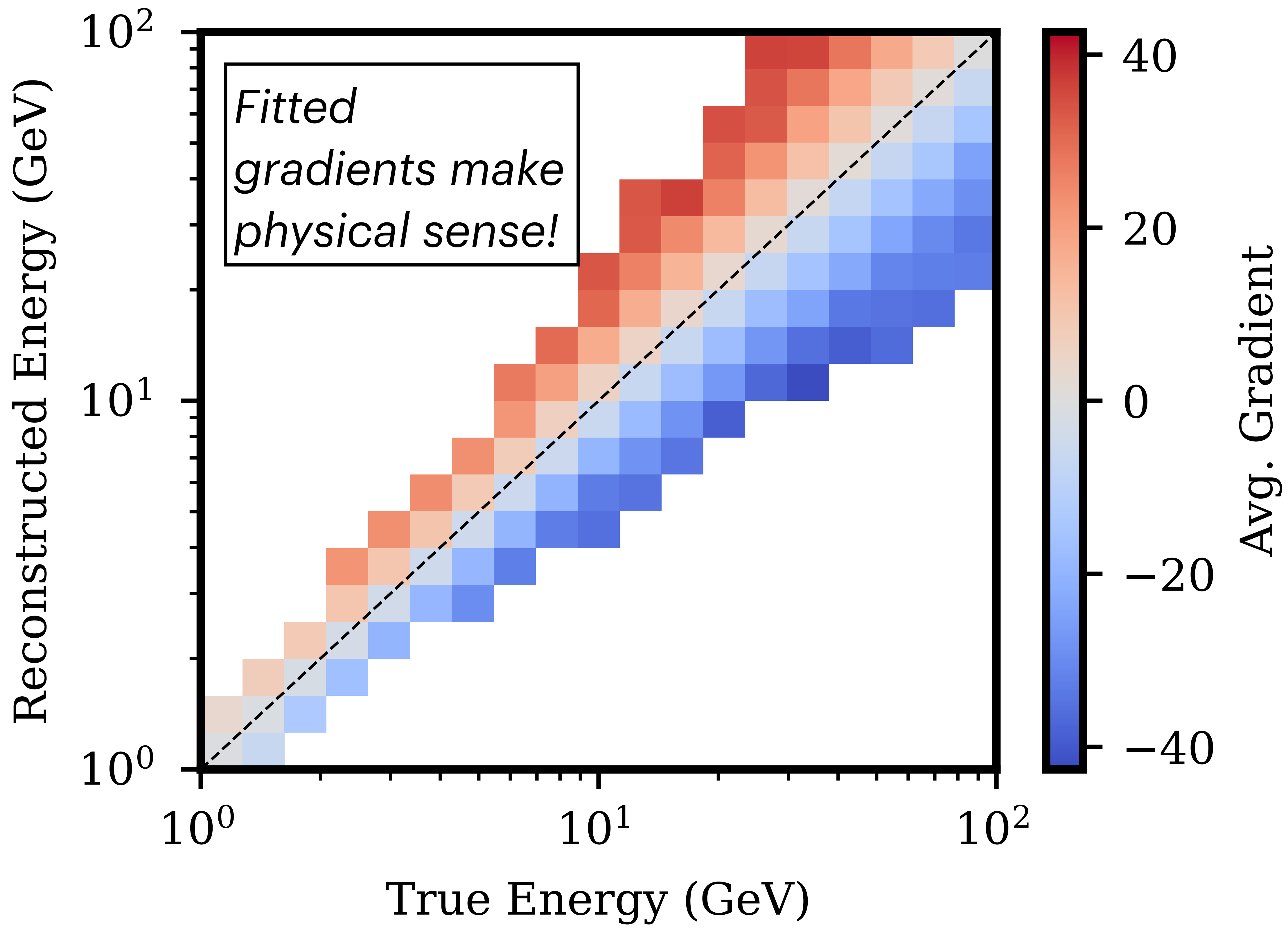


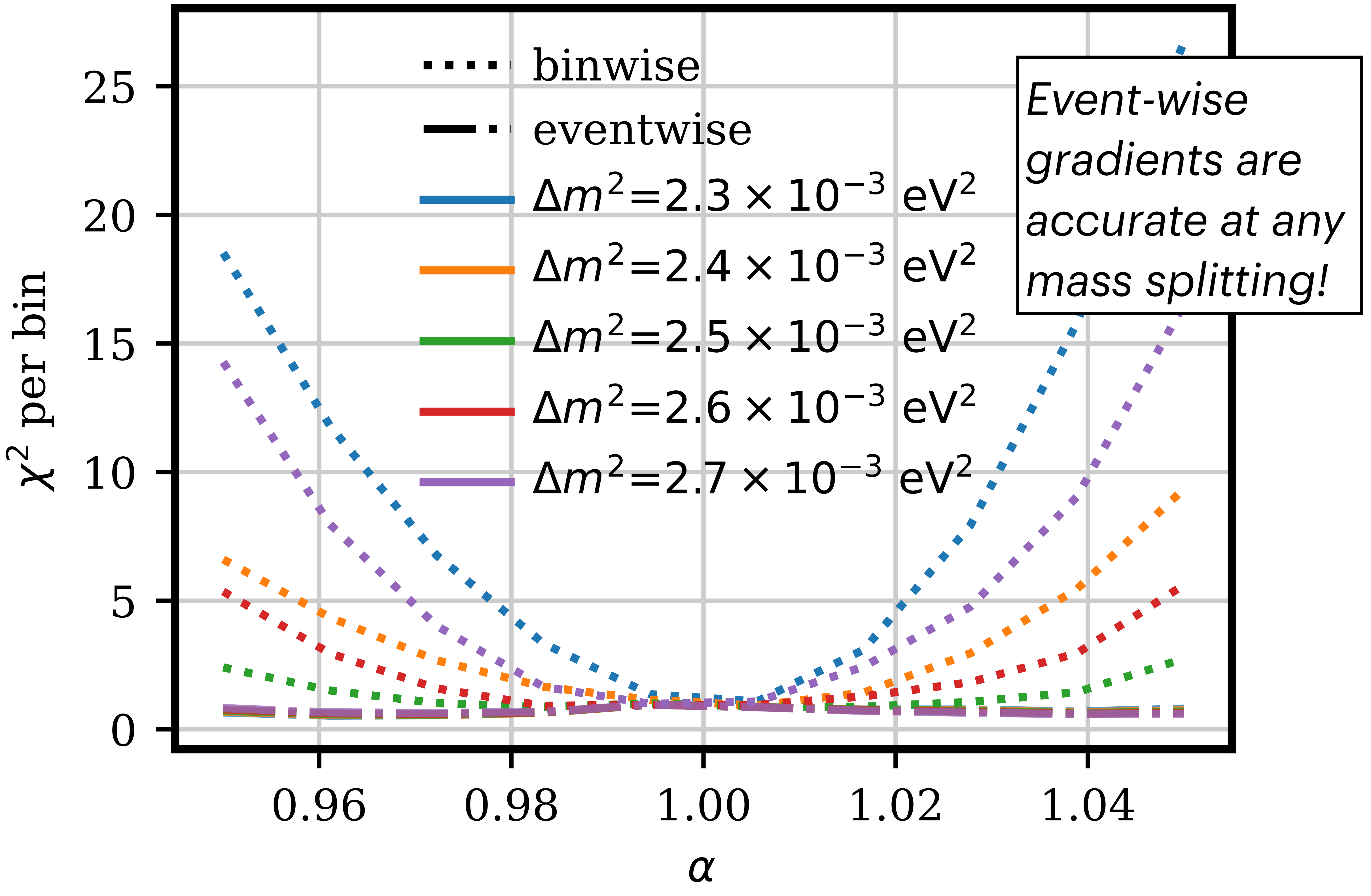


Note: The classifier was trained on the *unweighted* events!

Performance on Toy MC

Gradients make Sense and Produce Accurate Predictions





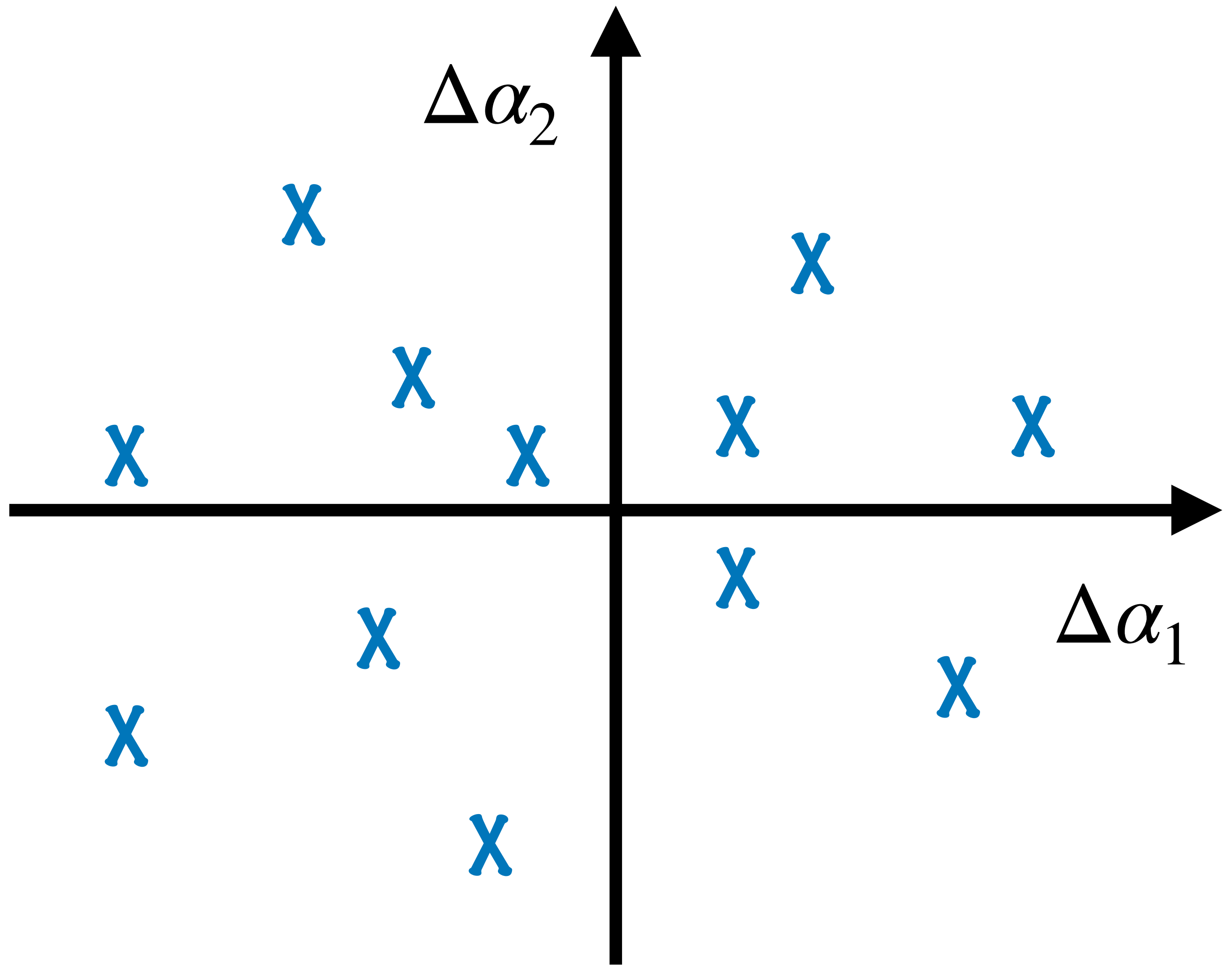
Summary of the Procedure

How you can apply this in your analysis

Summary of the Procedure

How you can apply this in your analysis

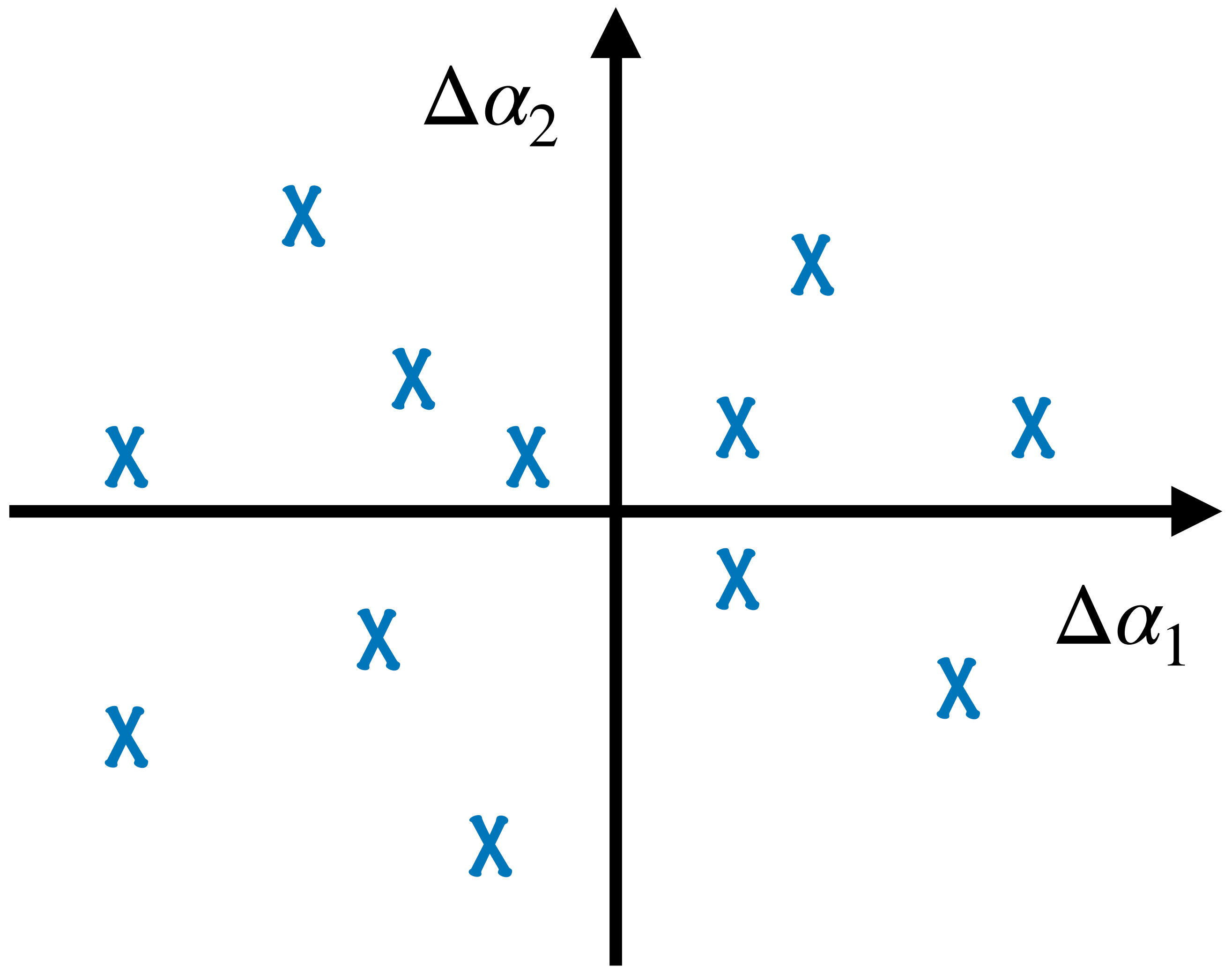
1. Generate MC sets at various realizations of the detector
   - Systematic sets can be much smaller than nominal set!



Summary of the Procedure

How you can apply this in your analysis

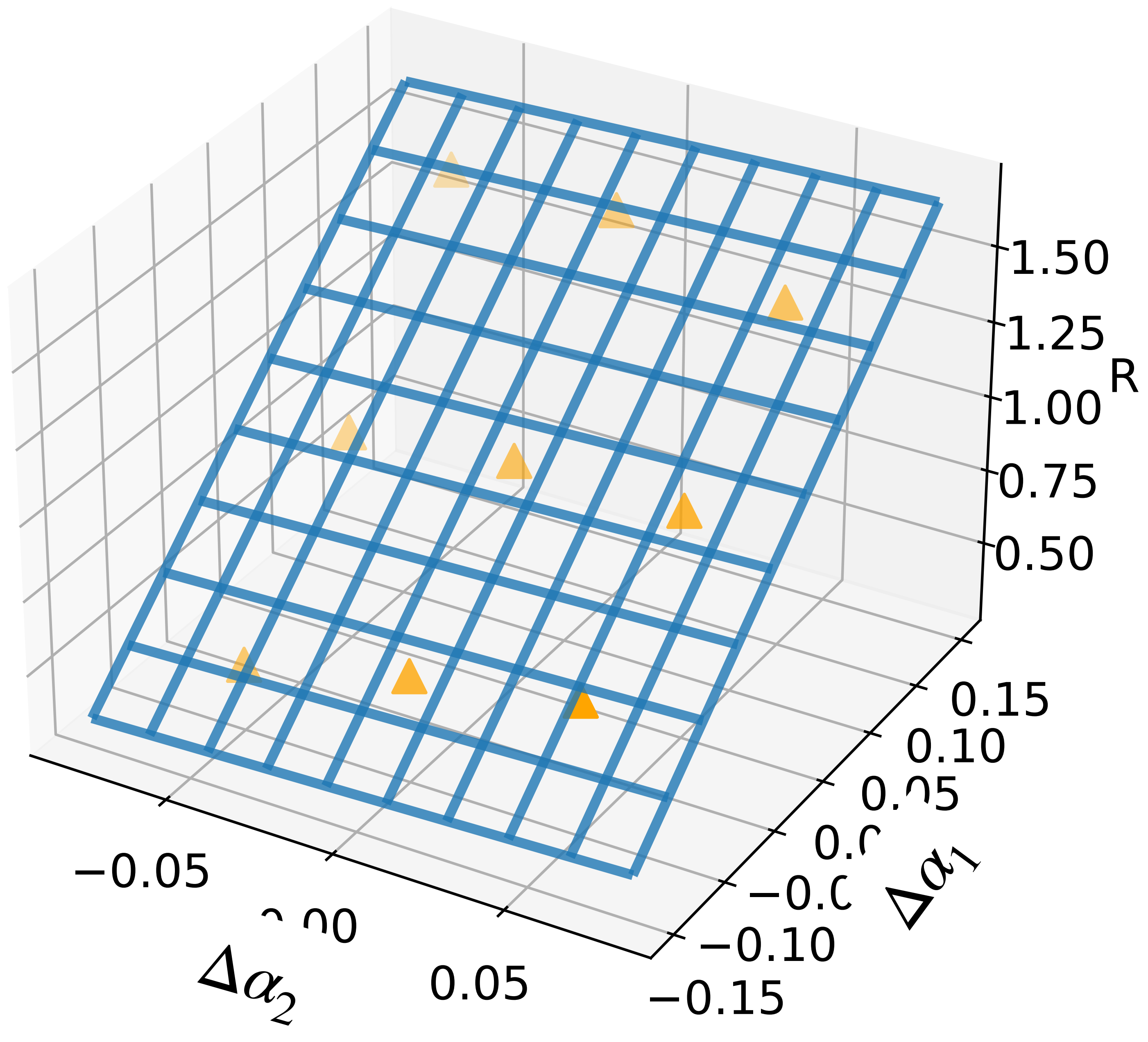
1. Generate MC sets at various realizations of the detector
   - Systematic sets can be much smaller than nominal set!
2. Fit the classifier
   - Any classifier giving *calibrated* posteriors may be used



Summary of the Procedure

How you can apply this in your analysis

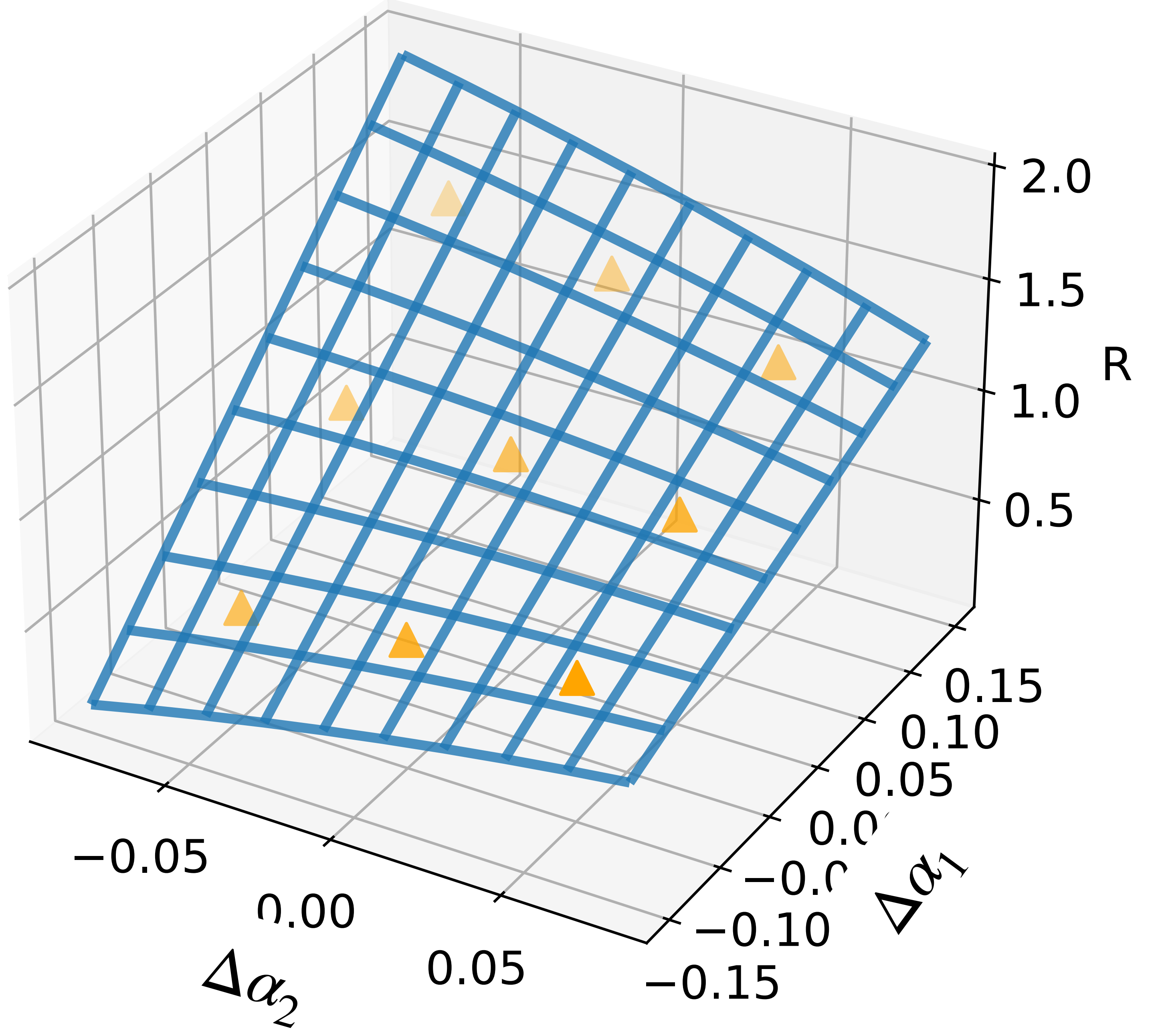
1. Generate MC sets at various realizations of the detector
    - Systematic sets can be much smaller than nominal set!
2. Fit the classifier
    - Any classifier giving *calibrated* posteriors may be used
3. Fit gradients for all nominal MC events
    - Polynomial features of parameters may be used



Summary of the Procedure

How you can apply this in your analysis

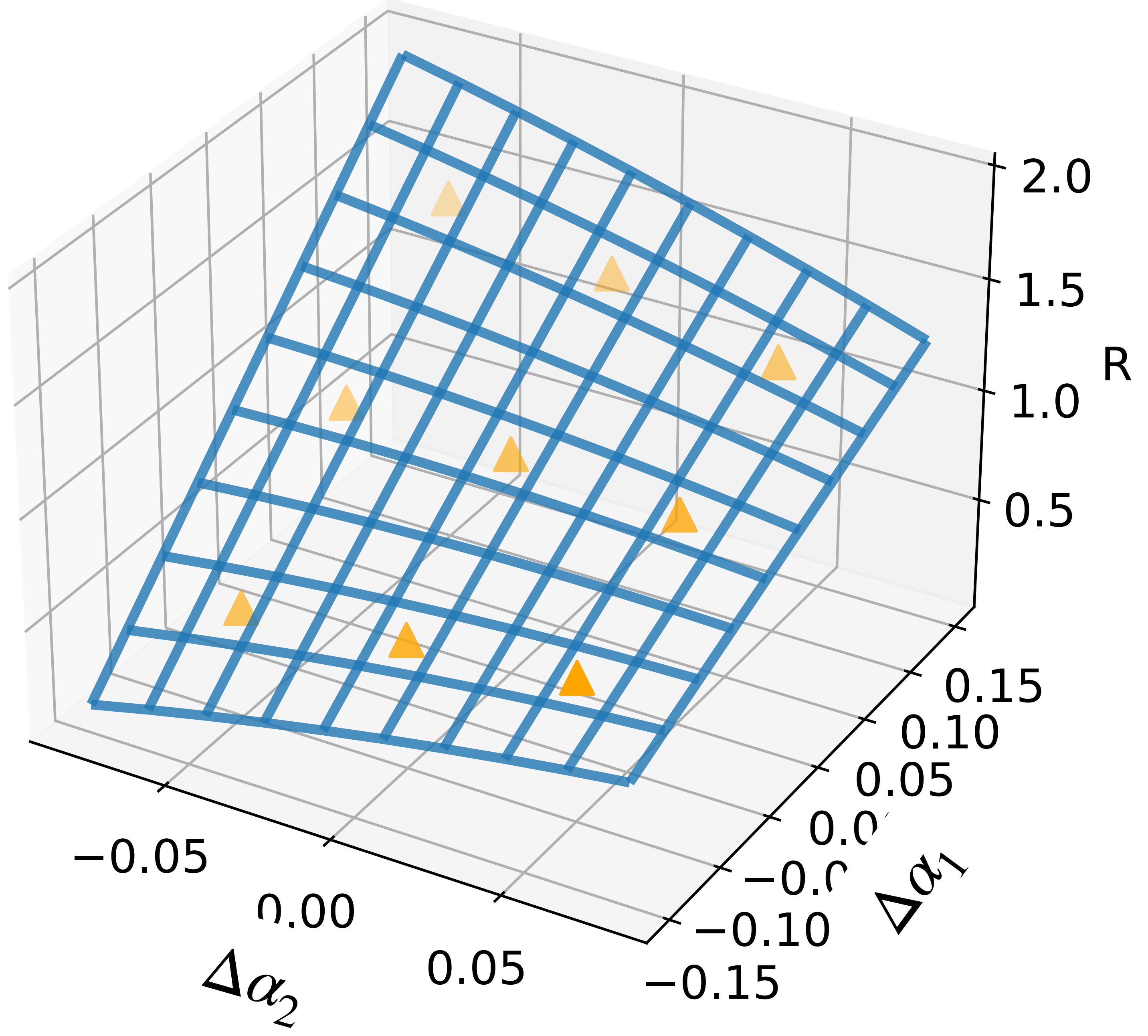
1. Generate MC sets at various realizations of the detector
   - Systematic sets can be much smaller than nominal set!
2. Fit the classifier
   - Any classifier giving *calibrated* posteriors may be used
3. Fit gradients for all nominal MC events
   - Polynomial features of parameters may be used



Summary of the Procedure

How you can apply this in your analysis

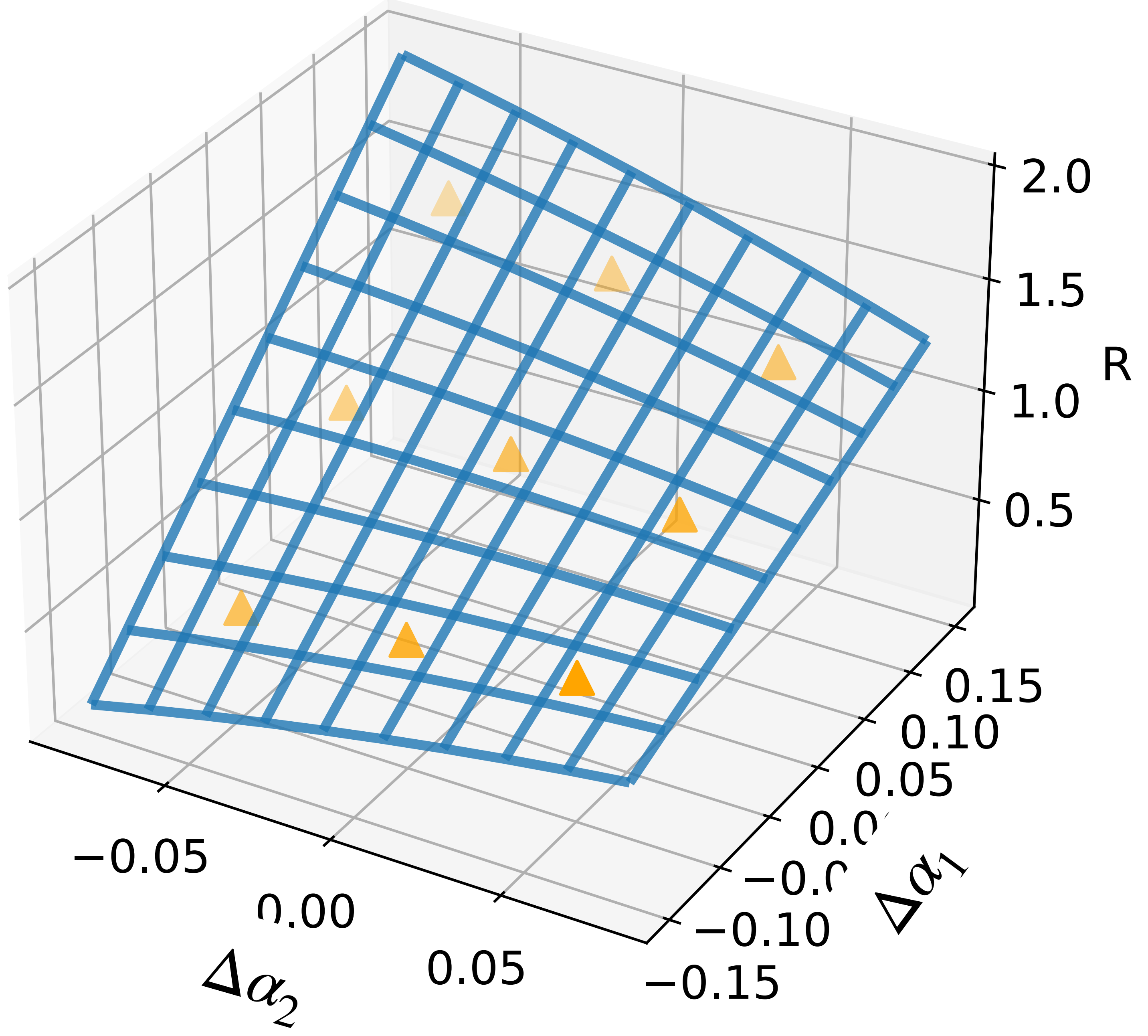
1. Generate MC sets at various realizations of the detector
   - Systematic sets can be much smaller than nominal set!
2. Fit the classifier
   - Any classifier giving *calibrated* posteriors may be used
3. Fit gradients for all nominal MC events
   - Polynomial features of parameters may be used



4. Weight your MC by $w_j = \exp\left(\sum_n g_{jn}(\alpha_n - \alpha_{n,\,\mathrm{nom}})\right)$ to get expectation values for any detector realization!

Summary of the Procedure

How you can apply this in your analysis



1. Generate MC sets at various realizations of the detector
   - Systematic sets can be much smaller than nominal set!
2. Fit the classifier
   - Any classifier giving *calibrated* posteriors may be used
3. Fit gradients for all nominal MC events
   - Polynomial features of parameters may be used
4. Weight your MC by $w_j = \exp\left(\sum_n g_{jn}(\alpha_n - \alpha_{n,\,\mathrm{nom}})\right)$ to get expectation values for any detector realization!

Analyzers need only the last step!

Summary

Summary

- Event-wise weights including true *and* reconstructed event properties can decouple detector effects from other weights such as flux, oscillation

Summary

- Event-wise weights including true *and* reconstructed event properties can decouple detector effects from other weights such as flux, oscillation
- Training a classifier to distinguish between systematic MC sets provides weights via likelihood-free inference trick

Summary

- Event-wise weights including true *and* reconstructed event properties can decouple detector effects from other weights such as flux, oscillation
- Training a classifier to distinguish between systematic MC sets provides weights via likelihood-free inference trick
- Can fit gradients for each MC event using a parametrization with the “softmax” function that interpolate between MC sets

Summary

- Event-wise weights including true *and* reconstructed event properties can decouple detector effects from other weights such as flux, oscillation
- Training a classifier to distinguish between systematic MC sets provides weights via likelihood-free inference trick
- Can fit gradients for each MC event using a parametrization with the “softmax” function that interpolate between MC sets
- Gradients allow for simple re-weighting of nominal MC to model any realization of the detector without requiring assumptions about linearity, fixed binning, etc.

Summary

- Event-wise weights including true *and* reconstructed event properties can decouple detector effects from other weights such as flux, oscillation
- Training a classifier to distinguish between systematic MC sets provides weights via likelihood-free inference trick
- Can fit gradients for each MC event using a parametrization with the “softmax” function that interpolate between MC sets
- Gradients allow for simple re-weighting of nominal MC to model any realization of the detector without requiring assumptions about linearity, fixed binning, etc.

Thank You!

Backup

MC Event Weighting

How we get an expectation value in each bin

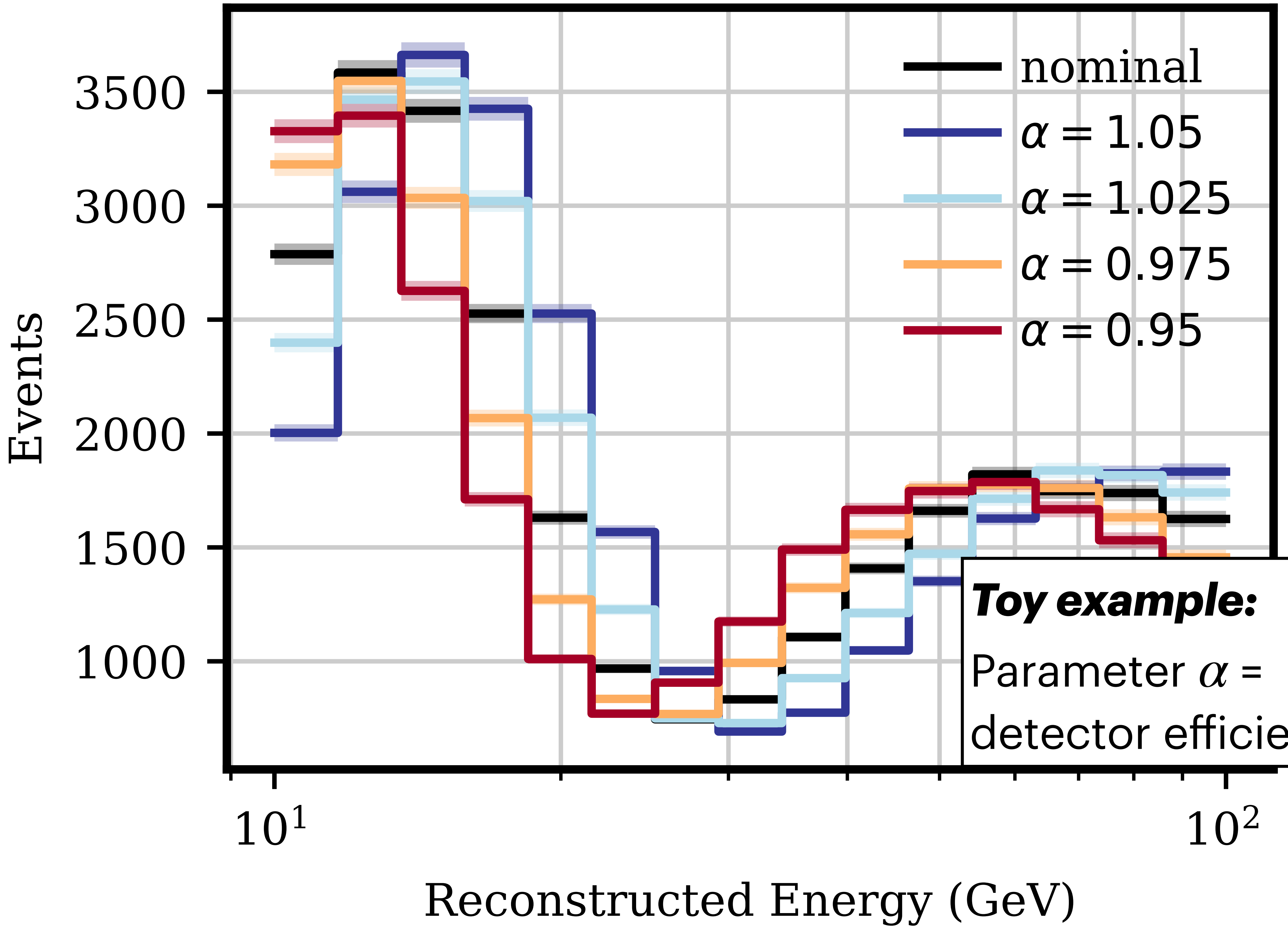
Full expression for expectation in each bin i:

$$\mu_i(\theta) = \int_{\mathbf{y}\in \text{bin } i} \mathrm{d}\mathbf{y} \int \mathrm{d}\mathbf{x} P(\mathbf{y}\,|\,\mathbf{x},\alpha) P(\text{acc}\,|\,\mathbf{x},\alpha) \frac{\Phi(\mathbf{x}\,|\,\theta)}{\Phi_{\text{sim}}(\mathbf{x})}$$

- **Flux, cross-sections, oscillations**:
  - Estimate bin count by weighting events:

$$\hat{\mu}_i(\theta) = \sum_j I(\mathbf{y}_j \in \text{bin } i) \frac{\Phi(\mathbf{x}_j\,|\,\theta)}{\Phi_{\text{sim}}(\mathbf{x}_j)}$$

- **Uncertainties of detector properties**:
  - How can we get $P(\text{acc}\,|\,\mathbf{x},\alpha)P(\mathbf{y}\,|\,\mathbf{x},\alpha)$?



Toy example:
Parameter α = detector efficiency

Goal of this Work

Decoupling Detector Response Weight from Physics Parameters

Basic intuition: Detector response should not depend on initial particle flux!

➡ Detector reacts to final state of each particle, doesn't know about flux or cross-sections

➡ Detector properties determine relationship between true and reconstructed variables

➡ If we knew $P(\mathbf{y} \mid \mathbf{x}, \alpha)P(\mathrm{acc} \mid \mathbf{x}, \alpha)$, we should be able to get the correct expectation value *independently* from θ

$$\hat{\mu}_i(\theta, \alpha) = \sum_j I(\mathbf{y}_j \in \mathrm{bin}\, i) \boxed{\frac{P(\mathbf{y}_j \mid \mathbf{x}_j, \alpha)P(\mathrm{acc} \mid \mathbf{x}_j, \alpha)}{P(\mathbf{y}_j \mid \mathbf{x}_j, \alpha_{\mathrm{nom}})P(\mathrm{acc} \mid \mathbf{x}_j, \alpha_{\mathrm{nom}})}} \frac{\Phi(\mathbf{x}_j \mid \theta)}{\Phi_{\mathrm{sim}}(\mathbf{x}_j)}$$

Event weight independent of θ

Benefits of our Method

Benefits of our Method

- ✓ Change your binning! Change your Physics! Gradients stay valid!
  - ➡ **Caveat:** Re-binning and selection changes may only use variables that were used as inputs into the classifier

Benefits of our Method

- ✓ Change your binning! Change your Physics! Gradients stay valid!
  - ➡ **Caveat**: Re-binning and selection changes may only use variables that were used as inputs into the classifier
- ✓ Allow anyone to use detector effects by adding gradients to MC data-release

Benefits of our Method

- ✓ Change your binning! Change your Physics! Gradients stay valid!
  - ➡ **Caveat**: Re-binning and selection changes may only use variables that were used as inputs into the classifier
- ✓ Allow anyone to use detector effects by adding gradients to MC data-release
- ✓ Automatically smooth over poor statistics in varied MC sets
  - ➡ Tune your classifier for the ideal balance of smoothness and over-fitting

Benefits of our Method

- ✓ Change your binning! Change your Physics! Gradients stay valid!
  - ➡ **Caveat**: Re-binning and selection changes may only use variables that were used as inputs into the classifier
- ✓ Allow anyone to use detector effects by adding gradients to MC data-release
- ✓ Automatically smooth over poor statistics in varied MC sets
  - ➡ Tune your classifier for the ideal balance of smoothness and over-fitting
- ✓ No assumption of linearity of detector effects

Benefits of our Method

- ✓ Change your binning! Change your Physics! Gradients stay valid!
  - ➡ **Caveat:** Re-binning and selection changes may only use variables that were used as inputs into the classifier
- ✓ Allow anyone to use detector effects by adding gradients to MC data-release
- ✓ Automatically smooth over poor statistics in varied MC sets
  - ➡ Tune your classifier for the ideal balance of smoothness and over-fitting
- ✓ No assumption of linearity of detector effects
- ✓ No assumption about how you space out your MC sets