ZAF Meeting

# Jets in common ntuples

Florian Lorkowski

May 15, 2020

## Motivation

**Purpose of this talk**

- Fill holes in documentation of common ntuples
- Give instructions to reproduce jet types

**Not the purpose of this talk**

- Comment on validity of methods
- Give advice on which jet type to use

**Source:** Original Fortran code

- Phantom[1], Orange[2]
- Control card for `v08b` ntuples[3]
- Especially: `<orange>/jets/ora_jets.fpp`, `<phantom>/Jets/ktclus/`

[1] `/afs/desy.de/group/zeus.zsmsm/ZEUSSysSoft/Released/zeus/ZeusUtil/phantom/v2013a/src/`
[2] `/afs/desy.de/group/zeus.zsmsm/ZEUSSysSoft/Released/zeus/Programs/orange/v2013a/src/`
[3] `https://www-zeus.desy.de/ZEUS_ONLY/analysis/comntp/cards/v08b/control.cards.Data.0607p`

## Motivation

**State of jets in the common ntuples**

- Common ntuples contain 21 different kinds of $k_\perp$ jets
- Nine of which where added in `v08` and are so far undocumented[1]

**Why these have to be reproduced**

- Changing input particles requires rerunning of jet clustering procedure
- Using different parameters for jet clustering
- Evaluation of systematic uncertainties

[1] They also render any previous documentation obsolete, as already existing jet types where renamed.



## Contents

**Preparation**

Define the following (examples on next slide):

- $d_1(p)$: A function which measures the distance of a particle to the beam pipe.
- $d_2(p, p')$: A function which measures the distance of two particles.
- A method to combine to particles into one.

**Clustering** [1]

- Start with a list of input particles.
- Evaluate $d_1(p)$ for every particles and $d_2(p, p')$ for every pair of particles.
- Find smallest value in $\{d_1(p), d_2(p', p)\}$
  - If smallest value is $d_1(p_i)$: Call particle $i$ a jet and remove it from list of inputs.
  - If smallest value is $d_2(p_i, p_j)$: Combine particles $i$ and $j$.
- Repeat until no more input particles are left.

[1] S. Catani, Yu. L. Dokshitzer, M. H. Seymour, B. R. Webber. Longitudinally-invariant $k_\perp$-clustering algorithms for hadron-hadron collisions. Nucl. Phy. B 406 (1993). DOI: 10.1016/0550-3213(93)90166-m





**Most common choice for $d_1$, $d_2$**

$$d_1(p) = p_\perp^2 \cdot R^2$$

$$d_2(p, p') = \min\left(p_\perp^2, {p'_\perp}^2\right)\left((y - y')^2 + (\phi -^1 \phi')^2\right)$$

- $R^2$ is a constant input parameter (usually 1 for DIS)
- Literature often incorrectly uses pseudorapidity $\eta$ instead of rapidity $y$[2]

**A note on notation**

Definitions at ZEUS:

$$\vec{p}_\perp = \begin{pmatrix} p_x \\ p_y \\ 0 \end{pmatrix}$$

$$p_\perp = |\vec{p}_\perp| = |\vec{p}| \sin\theta$$

$$E_\perp = E \cdot \frac{|\vec{p}_\perp|}{|\vec{p}|} = E \sin\theta$$

Definition in some papers:[2]

$$E_\perp = |\vec{p}_\perp|$$

- Equivalent for massless jets
- Distinction important for massive jets

[1] Pay attention to periodicity of $\phi$.
[2] S. D. Ellis, D. E. Soper. Successive combination jet algorithm for hadron collisions. Phys. Rev. D 48 (1993). DOI: 10.1103/physrevd.48.3160

## The $k_\perp$ algorithm
### Examples: Recombination schemes

**$E$ scheme**

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_{\text{new}} = \begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_1 + \begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_2$$

→ Reconstructed jets will be massive

**Weighted $p_\perp$ scheme**

$$p_{\perp,\text{new}} = p_{\perp,1} + p_{\perp,2}$$

$$\eta_{\text{new}} = \frac{\eta_1 p_{\perp,1} + \eta_2 p_{\perp,2}}{p_{\perp,\text{new}}}$$

$$\phi_{\text{new}} = \frac{\phi_1 p_{\perp,1} +^1 \phi_2 p_{\perp,2}}{p_{\perp,\text{new}}}$$

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_{\text{new}} = \begin{pmatrix} p_\perp \cos\phi \\ p_\perp \sin\phi \\ p_\perp \sinh\eta \\ p_\perp \cosh\eta \end{pmatrix}_{\text{new}}$$

→ Reconstructed jets will be massless

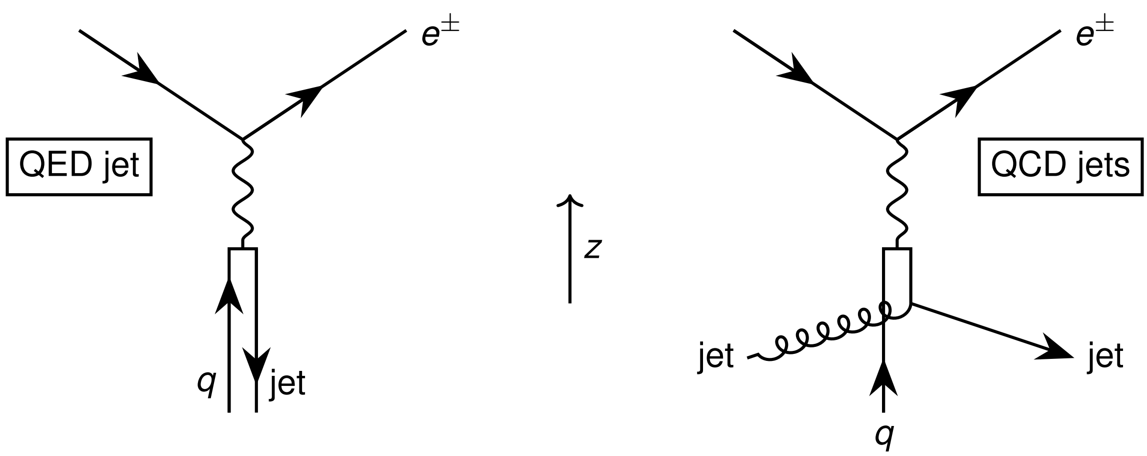
⇒ Sample implementation of $k_\perp$ algorithm in appendix 1

[1] Pay attention to periodicity of $\phi$. Use $\phi_{\text{new}} = \phi_1 + p_{\perp,2}/p_{\perp,\text{new}} f(\phi_2 - \phi_1)$, where $|f(x)| \leq \pi$.



## Breit reference frame
### Properties

- Single jets may arise purely from QED, which is uninteresting for studies of QCD
- In the Breit frame such events are suppressed, as the jet will be scattered on the 'beam pipe' ($z$ axis/photon axis)
- Also convenient from theoretical point of view, as it has useful properties related to factorization



# Breit reference frame
## Definition

### Definition

- Exchanged photon is purely space-like:

$$q^\mu = \begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -Q \\ 0 \end{pmatrix}$$

- Exchanged photon collides head-on with incoming quark
  $\rightarrow$ incoming proton momentum:

$$p^\mu = \begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ Q/(2x) \\ Q/(2x) \end{pmatrix}$$

### Two way to determine transformation

- Definition of Breit frame depends on $q$, which depends on momentum of outgoing electron
- Electron momentum may be reconstructed from
  - EL: position & energy ($\hat{=}$ electron method) or
  - DA: angles ($\hat{=}$ double angle method).

$\Rightarrow$ Sample code for Breit frame transformation in appendix 1



# Jet types in Ntuples
## Overview: ZUFO jets

| Jet type | Mass | Electron rejection | Frame: Found | Frame: Stored | Frame: Breit type | Cuts (lab): $E_{\perp,\text{min}}$ [GeV] | Cuts (lab): $\eta_{\text{min}}$ | Cuts (lab): $\eta_{\text{max}}$ |
|---|---|---|---|---|---|---|---|---|
| A | massive | no | Lab | Lab | – | 2.5 | -2.5 | 2.5 |
| B | massive | Sinistra | Lab | Lab | – | 2.5 | -2.5 | 2.5 |
| C | massive | Sinistra | Breit | Lab | EL | 2.5 | -2.5 | 2.5 |
| D | massive | Sinistra | Breit | Breit | EL | 2.5 | -2.5 | 2.5 |
| E | massless | Sinistra | Breit | Lab | EL | 2.5 | -2.5 | 4.0 |
| F | massless | Sinistra | Breit | Lab | DA | 2.5 | -2.5 | 4.0 |
| G | massless | Sinistra | Breit | Breit | EL | 2.5 | -2.5 | 4.0 |
| H | massless | Sinistra | Breit | Breit | DA | 2.5 | -2.5 | 4.0 |
| I | massless | no | Lab | Lab | – | 4.0 | -2.5 | 4.0 |
| J | massive | EM | Lab | Lab | – | 2.5 | -2.5 | 2.5 |
| K | massive | EM | Breit | Lab | EL | 2.5 | -2.5 | 2.5 |
| L | massive | EM | Breit | Breit | EL | 2.5 | -2.5 | 2.5 |
| M | massless | EM | Breit | Lab | EL | 2.5 | -2.5 | 4.0 |
| N | massless | EM | Breit | Lab | DA | 2.5 | -2.5 | 4.0 |
| O | massless | EM | Breit | Breit | EL | 2.5 | -2.5 | 4.0 |
| P | massless | EM | Breit | Breit | DA | 2.5 | -2.5 | 4.0 |

**Cone island jets**

| Jet type | Mass | Electron rejection | Frame | | | Cuts (lab) | | |
|---|---|---|---|---|---|---|---|---|
| | | | Found | Stored | Breit type | $E_{\perp,\text{min}}$ [GeV] | $\eta_\text{min}$ | $\eta_\text{max}$ |
| Q | massless | no | Lab | Lab | – | 4.0 | -3.0 | 3.0 |
| R | massless | EM | Lab | Lab | – | 4.0 | -3.0 | 3.0 |

**Jets from CAL cells**

| Jet type | Mass | Electron rejection | Frame | | | Cuts (lab) | | |
|---|---|---|---|---|---|---|---|---|
| | | | Found | Stored | Breit type | $E_{\perp,\text{min}}$ [GeV] | $\eta_\text{min}$ | $\eta_\text{max}$ |
| S | massless | no | Lab | Lab | – | 4.0 | -2.5 | 4.0 |
| T | massless | Sinistra | Breit | Breit | EL | 2.5 | -2.5 | 4.0 |
| U | massless | Sinistra | Breit | Lab | EL | 2.5 | -2.5 | 4.0 |





- Massive jet types use the $E$ scheme, massless ones $p_\perp$ scheme
- Cuts are always applied to $E_\perp$ and $\eta$, even for massive jets
- Cuts are always applied in the Lab frame, even for Breit frame jets
- Jets are always sorted by their $E_\perp$ in the Lab frame
- $R^2 = 1$ for all jet types
- At most 10 jets are saved per type
- ZUFOs and Cone islands are corrected for dead material, CAL cells are not; the same is true for corresponding jets

Massive ZUFOS:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_i = \begin{pmatrix} \texttt{Zufo[i][0]} \\ \texttt{Zufo[i][1]} \\ \texttt{Zufo[i][2]} \\ \texttt{Zufo[i][3]} \end{pmatrix}$$

Massless ZUFOS:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix} \rightarrow \begin{pmatrix} f \cdot p_x \\ f \cdot p_y \\ f \cdot p_z \\ E \end{pmatrix} \qquad f = \frac{E}{|\vec{p}|}$$

**Electron rejection**

- No electron rejection: Use all ZUFOs (without applying cuts/corrections)
- Sinistra rejection: Reject ZUFOs where `tufo[i][0]` in range $[3000, 3041]$ or $[7000, 7041]$
- EM rejection: Reject ZUFOs where `tufo[i][0]` in range $[4000, 4041]$ or $[7000, 7041]$





- Workflow for cone island jets:

  ZUFO islands → CorAndCut → Jet finder

- Uncorrected ZUFO islands are stored in ntuples; corrected ones are not
- Corrections applied by CorAndCut
  - Dead material
  - Overestimation of small energies
  - Super cracks
- Without reimplementing CorAndCut algorithm, cone island jets cannot be reproduced
- Left as exercise for the reader

# Input particles
## CAL cells

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_i = \begin{pmatrix} \texttt{Caltru\_pos[i][0]} - \texttt{Xvtx} \\ \texttt{Caltru\_pos[i][1]} - \texttt{Yvtx} \\ \texttt{Caltru\_pos[i][2]} - \texttt{Zvtx} \end{pmatrix}$$

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ E \end{pmatrix}_i = \begin{pmatrix} f \cdot x \\ f \cdot y \\ f \cdot z \\ \texttt{Caltru\_e[i]} \end{pmatrix} \qquad f = \frac{E}{|\vec{x}_i|}$$

Note: CAL cell particles are always massless. Clustered jets may be massive.

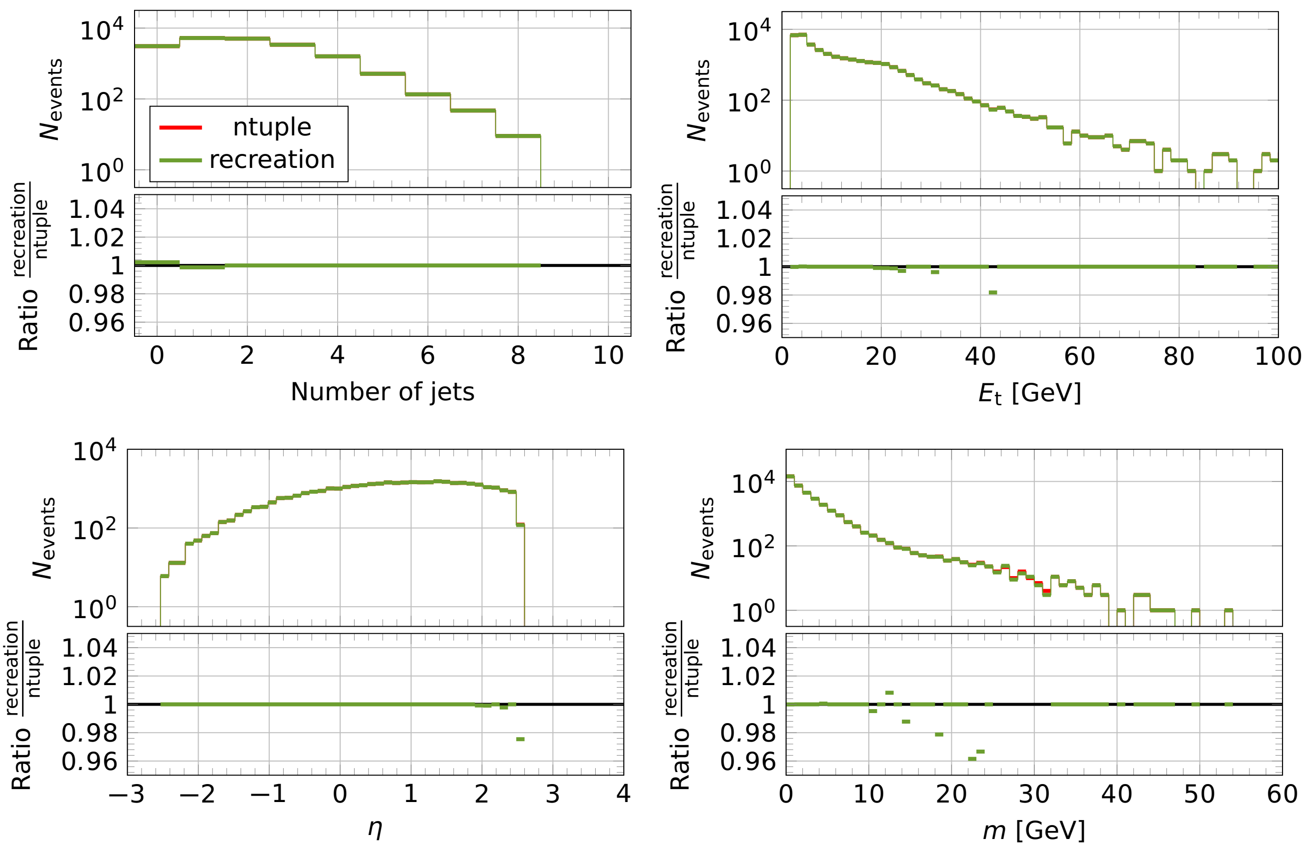
**Electron rejection**

- No electron rejection: Use all cells (without applying cuts/corrections)
- Sinistra/EM rejection: Reject all cells which are part of cell list of a given electron

⇒ Sample code for input particle construction in appendix 1

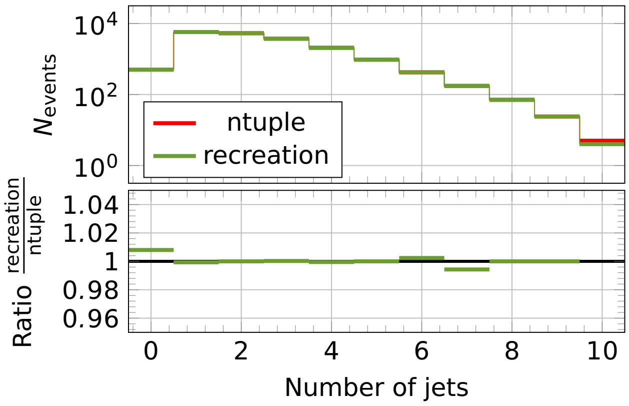


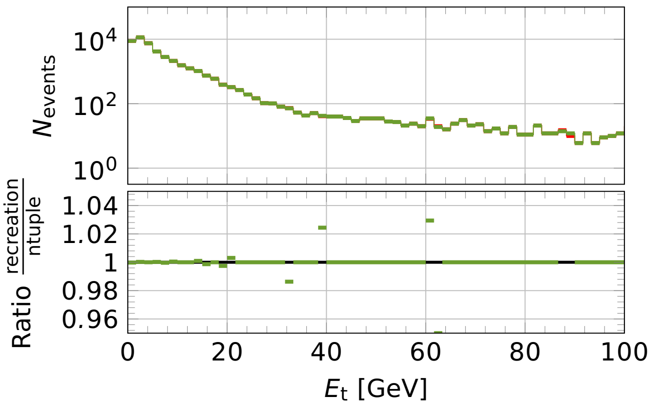
# Results of reproduction
## Control distribution: Jet type C (ZUFO jets in Lab frame)

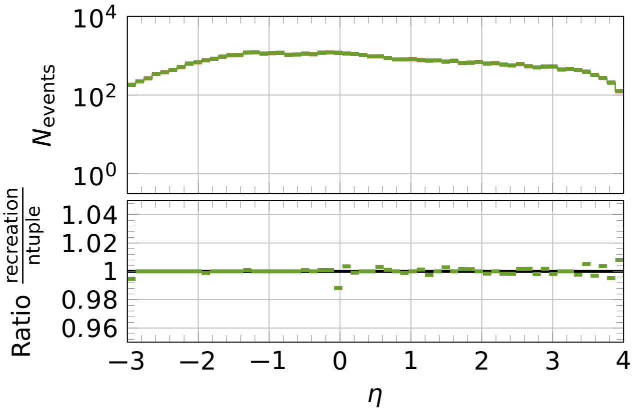


# Results of reproduction

## Control distribution: Jet type T (CAL jets in Breit frame)






# Results of reproduction

## Overall results

- Almost perfect agreement for all jet types
- Mismatch[1] for Lab frame jets: $< 10^{-5}$
- Mismatch for Breit frame jets: $\sim 10^{-4}$
- Deviations consistent with being due to numerics
- Increased numerical fluctuation when Breit frame boosts are involved
- Mass especially affected

$\Rightarrow$ Control distributions for all jet types in appendix 2

[1] Defined as (Number of jets missing + Number of jets too much)/(Expected number of jets)

## Summary

- Documentation for missing jet types provided
- Provided code samples for jet reconstruction (input particle selection, clustering, boosting)
- ZUFO and CAL cell jets can be reproduced very well
- Cone island jets cannot be reproduced, because implementation of CorAndCut algorithm is missing
- It is now possible to reconstruct jets using any combination of jet parameters (input, mass, frame, cuts, . . . )



## Code samples

`C++` **code samples:** (click file name to save file)

Contents:

- JetFinder: $k_\perp$ algorithm
- BreitFrame: Breit frame transformation
- InputParticles: Construction of input particle lists
- JetReconstruction: Combines everything to recreate ntuple jets
- Not included: Reading ntuples and drawing plots

Dataset:

- Single input file, which is part of `ariadne_high_Q2_NC_0607p` MC dataset
- `07p/v08b/mc/root/zeusmc.hfix627.e11911.nc.dja.ari.5d.07p.400.0074.root`

Cuts:

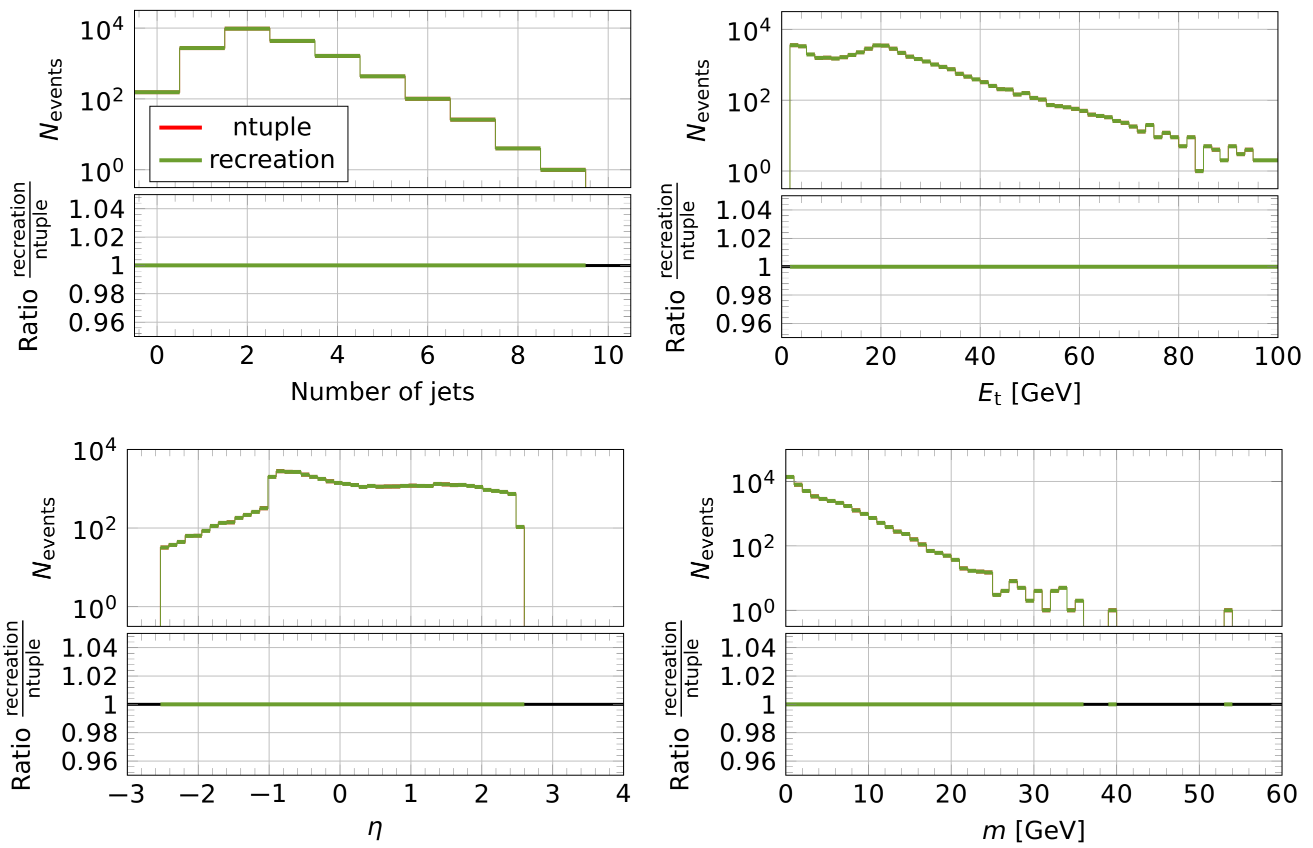
- Same set of cuts for all jet types, even though some types require fewer cuts
- At least one EM and Sinistra candidate (probability does not matter)
- $y_{\mathrm{jb}} < 1$ for most probable EM and Sinistra electron[1]

$\rightarrow$ 19063 events selected $\hat{=}$ $\sim$25 000 to 50 000 jets, depending on type

[1] In the ntuple code, there is a bug, where the DA variables do not get calculated, if the JB calculation failed. If the Breit frame reconstructed from angles, it will fail as well and some jets will be missing from the ntuples. This cut removes the affected events ($\sim$ 1.5%). This cut is only necessary, when DA variables are calculated in analysis. If they are taken from ntuples, they will correspond to the jets.

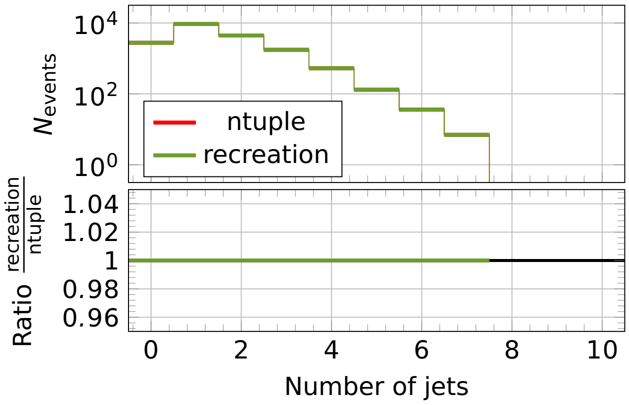


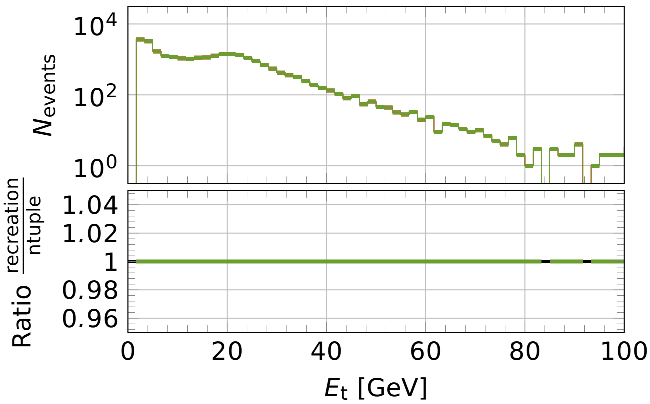


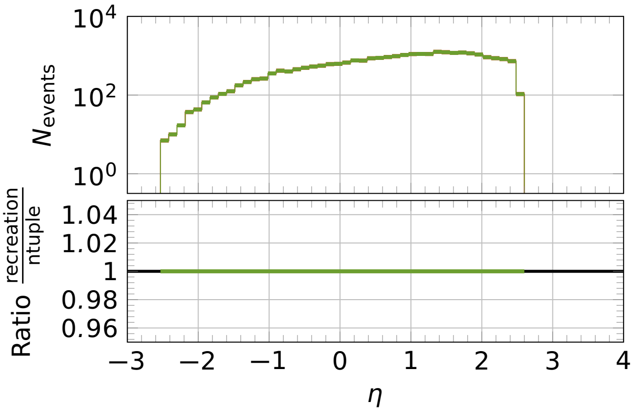


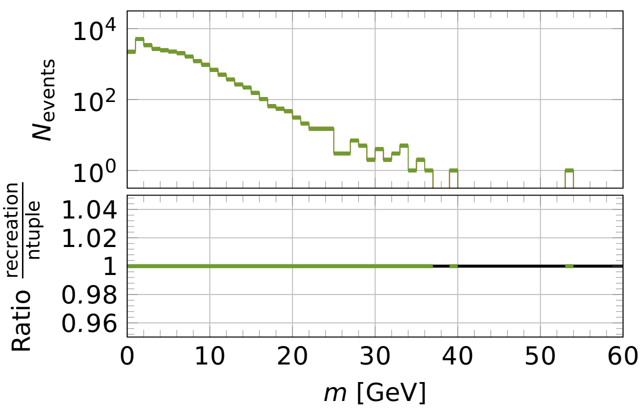
# Control distributions

## Jet type B



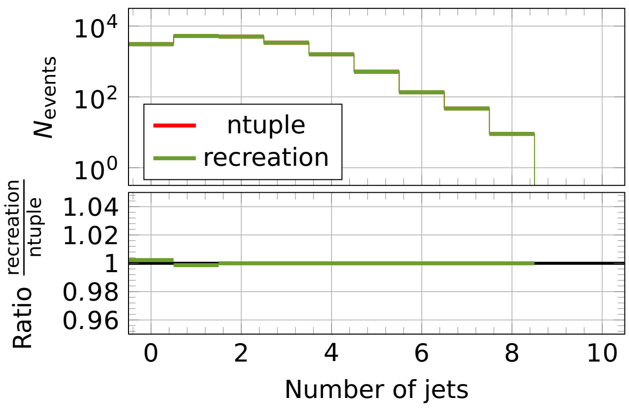



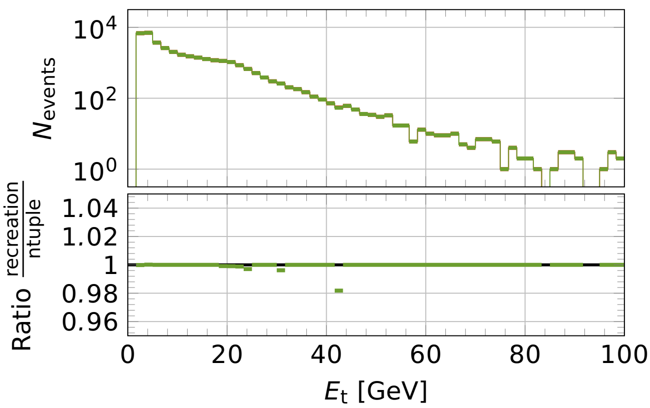



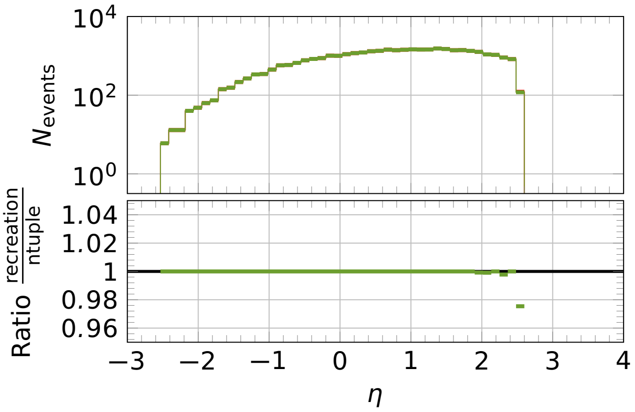


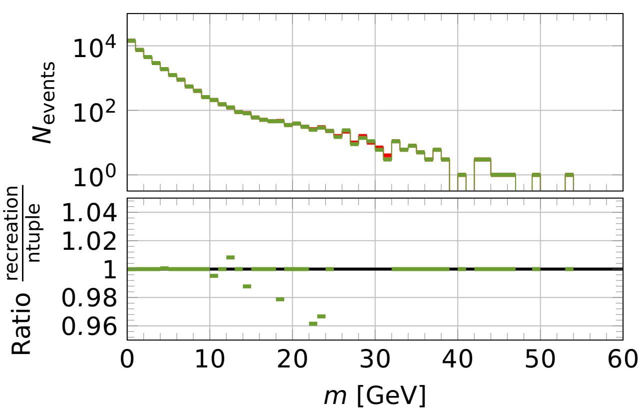

# Control distributions

## Jet type C

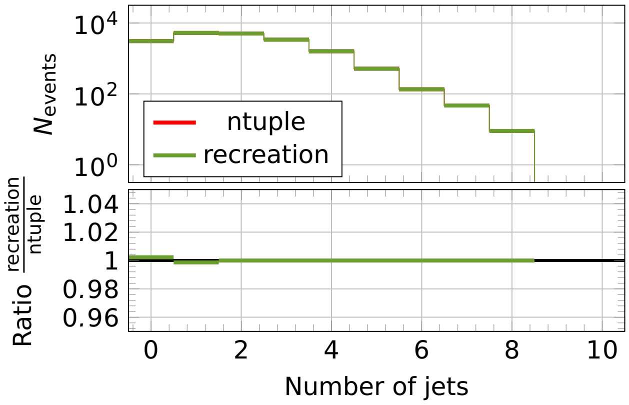


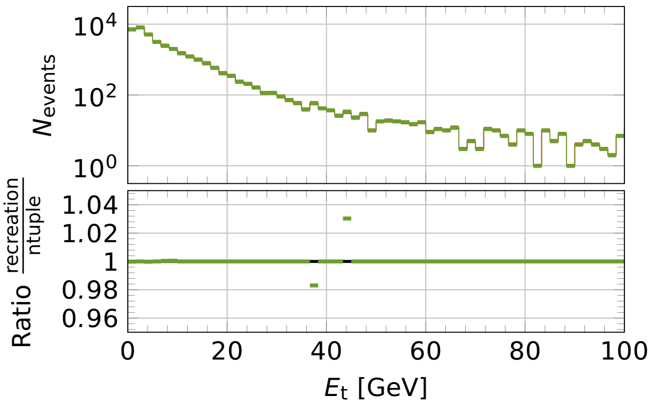



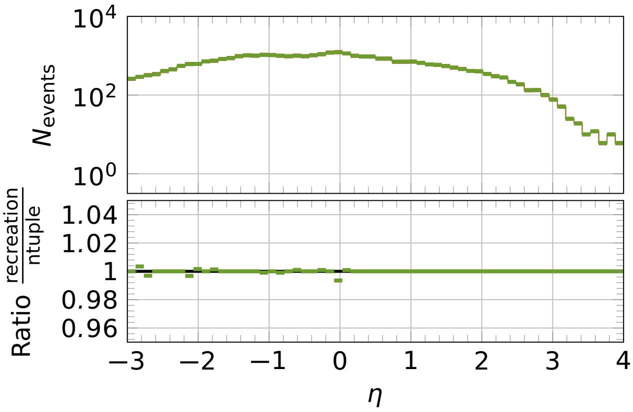



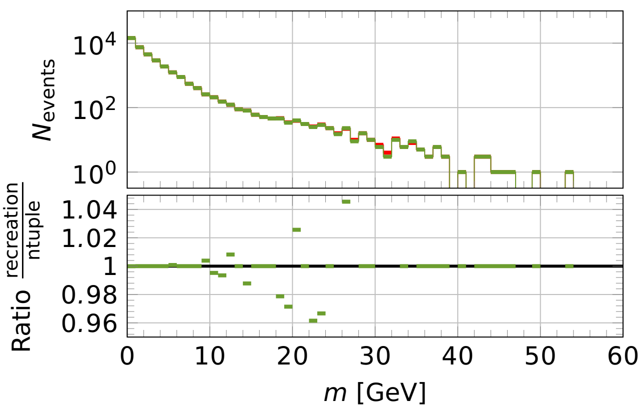



## Jet type D



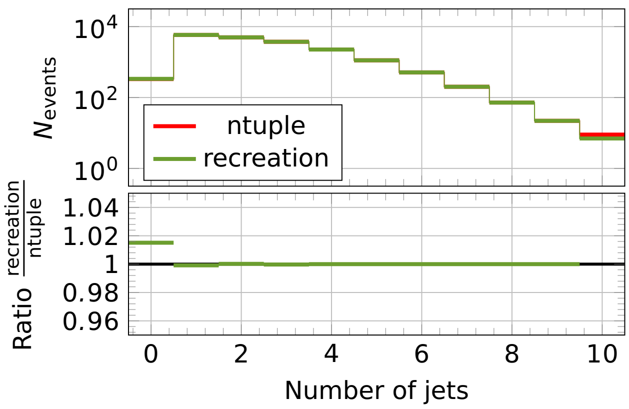



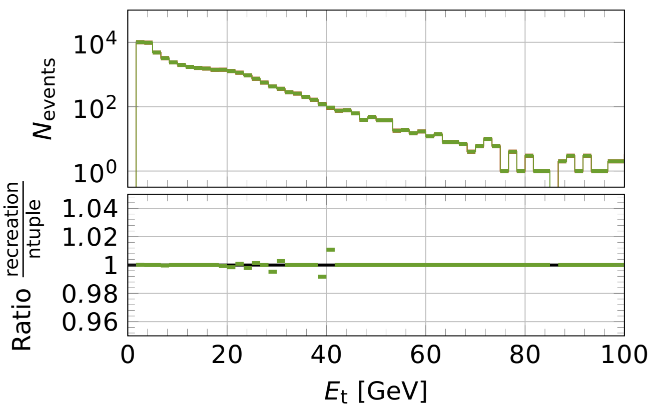



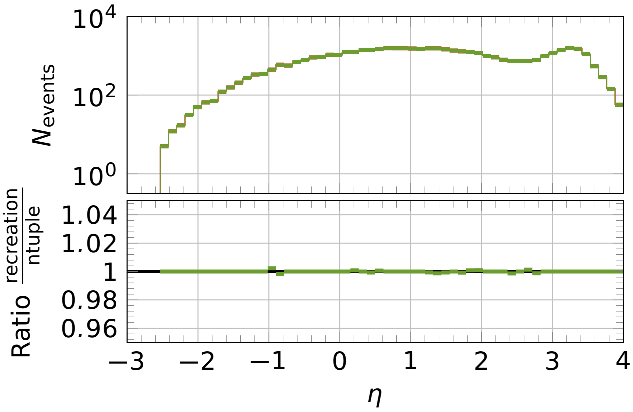



Control distributions

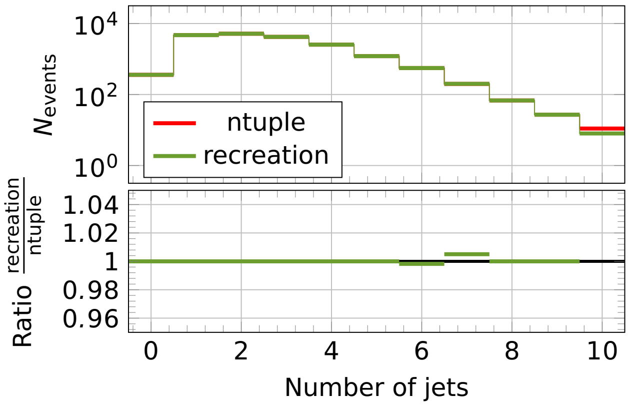
## Jet type E

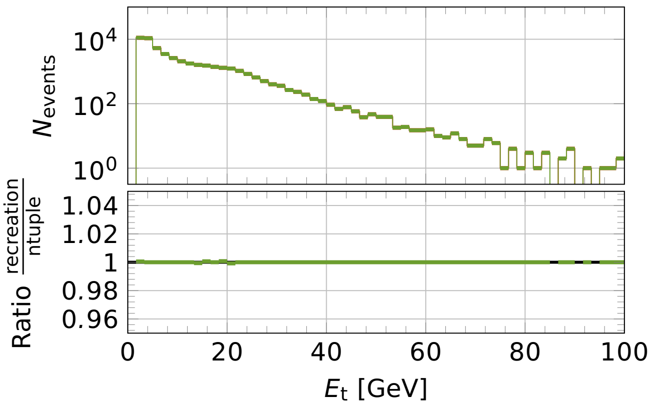


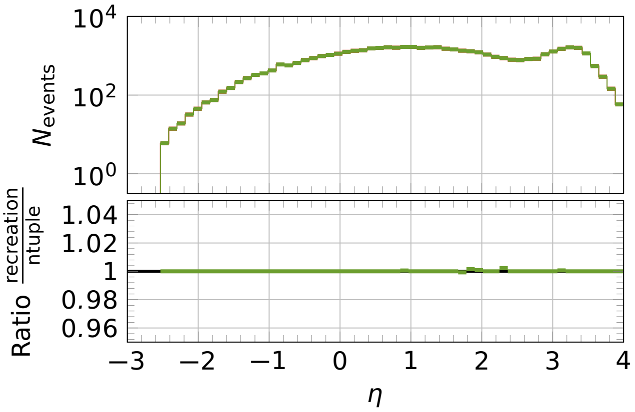



## Control distributions
### Jet type F

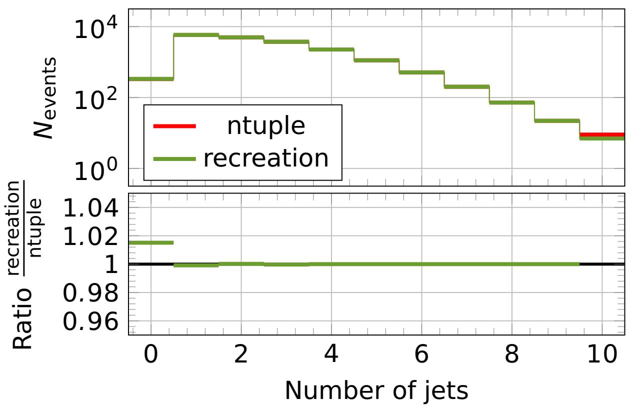


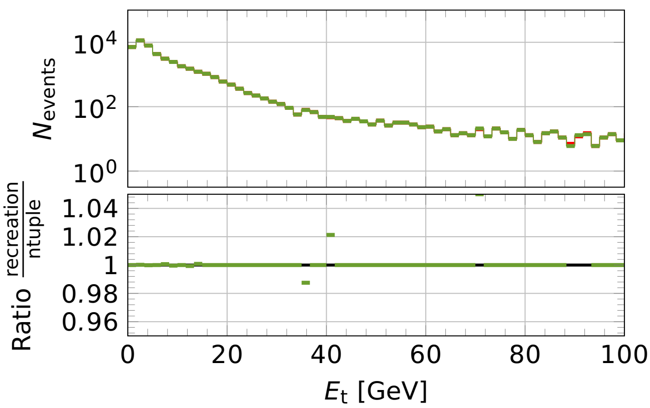



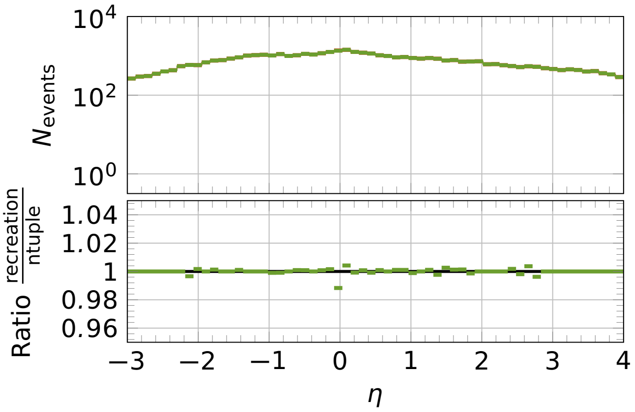






## Control distributions
### Jet type G

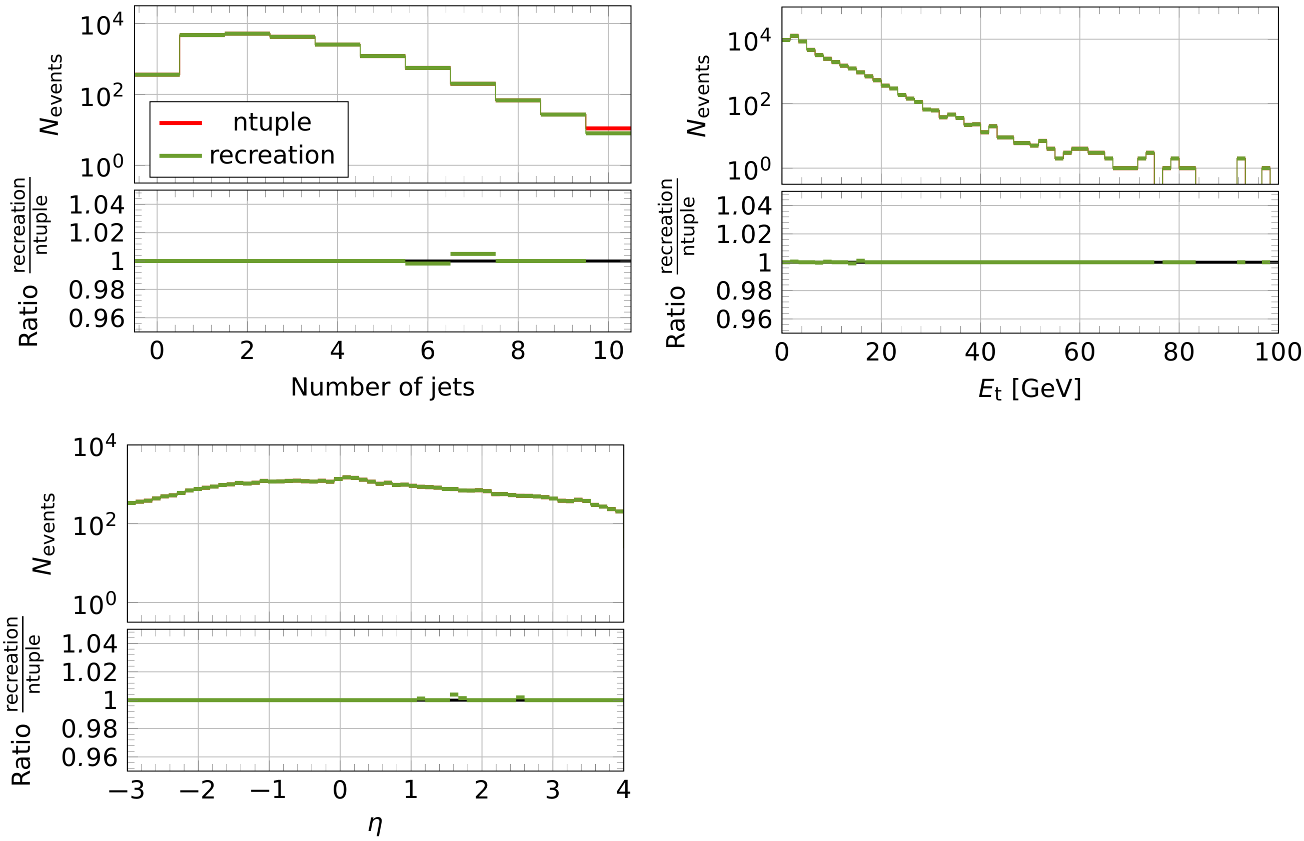


# Control distributions

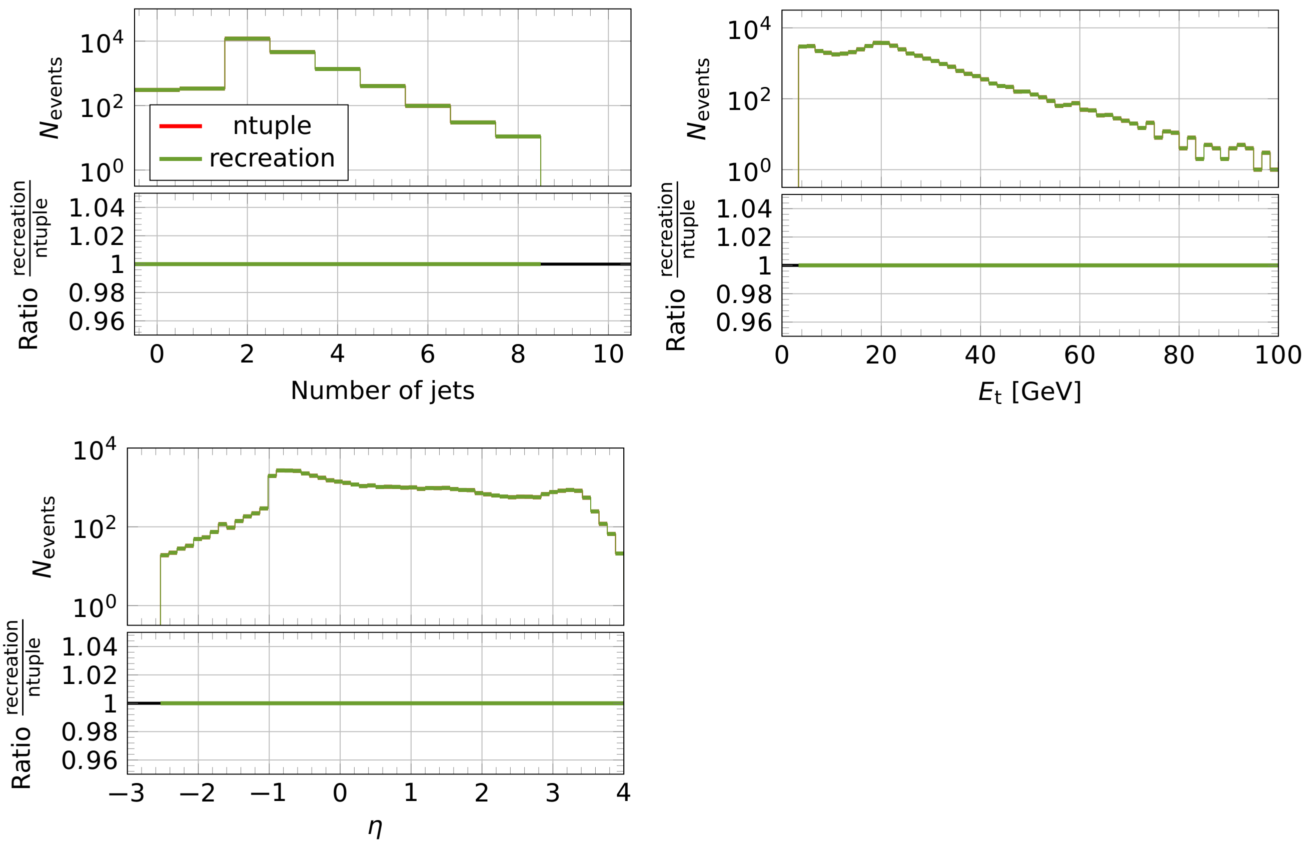
## Jet type H





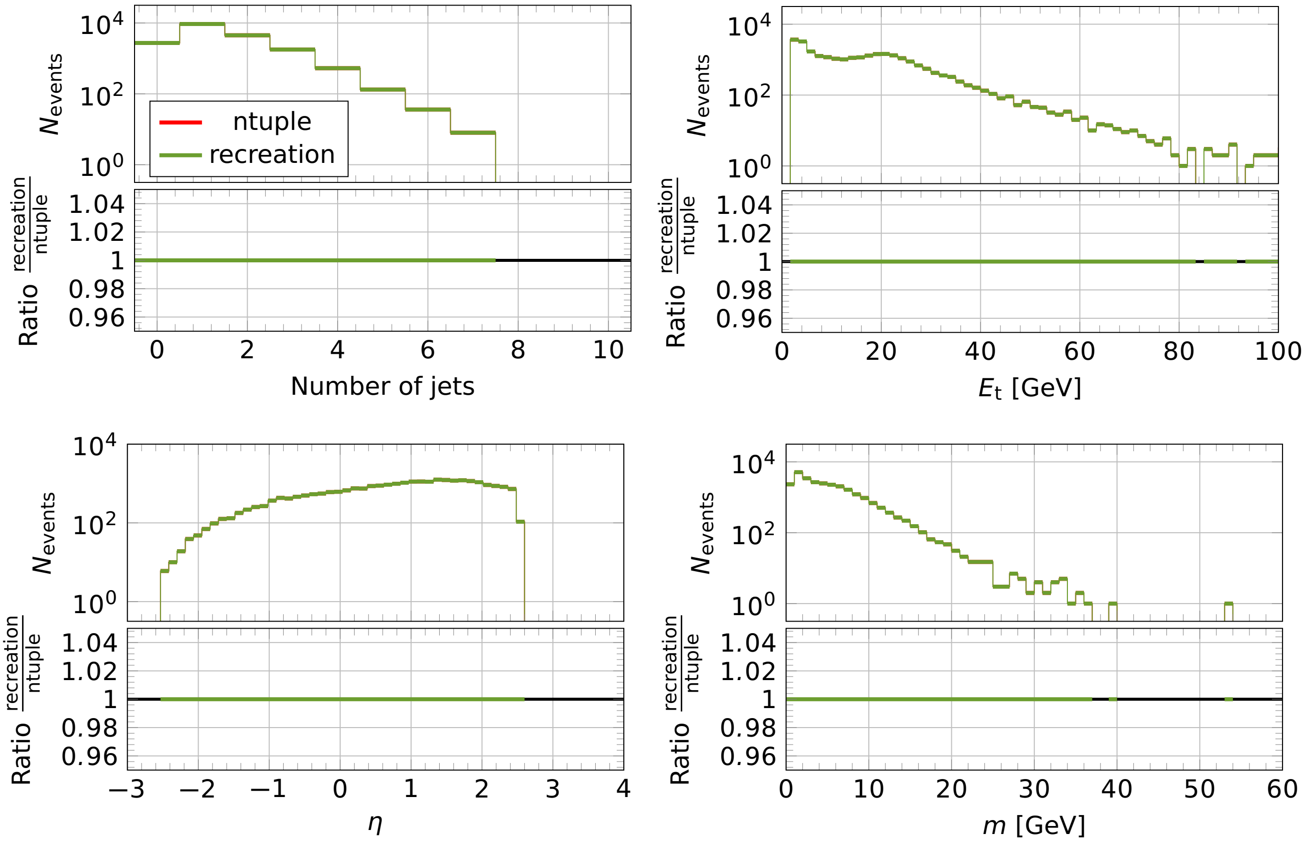

# Control distributions

## Jet type I



# Control distributions

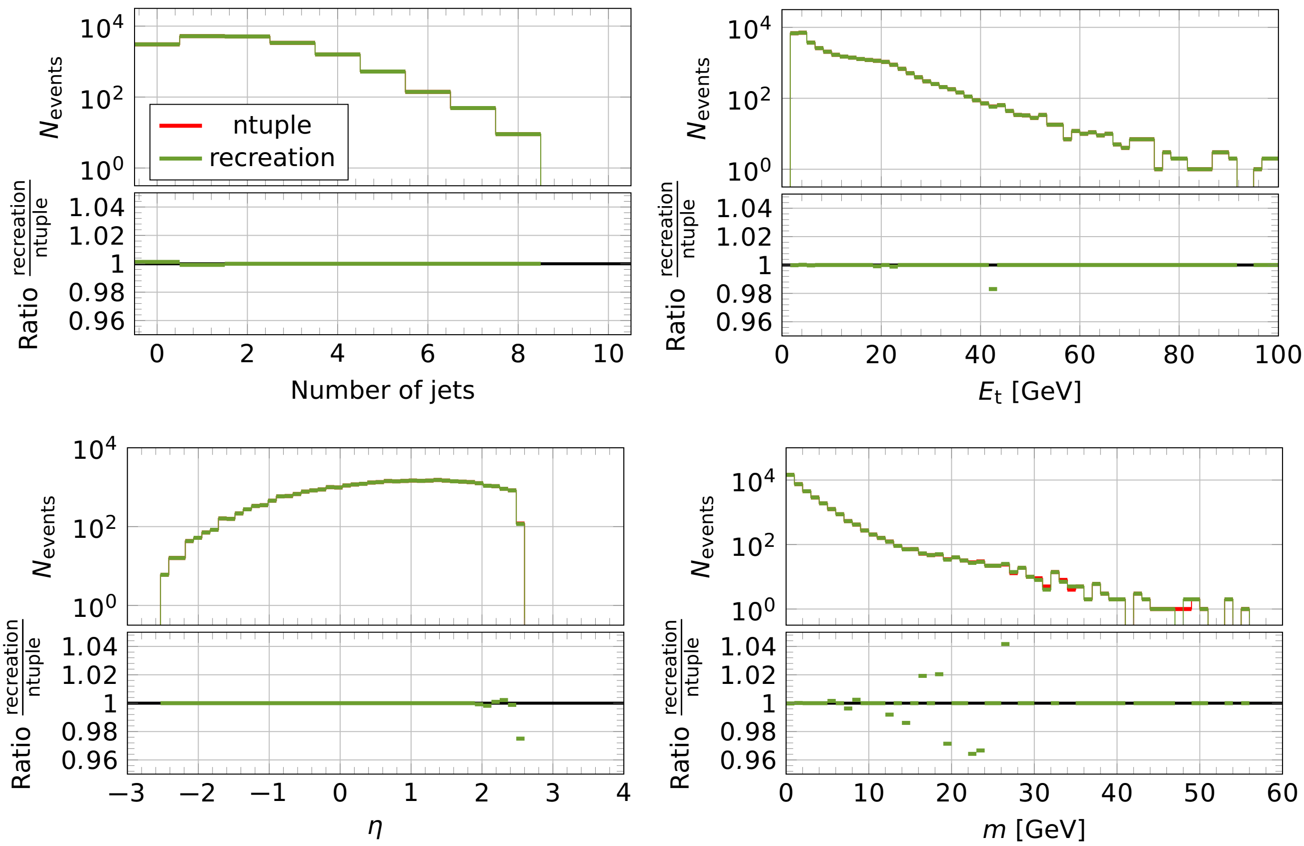
## Jet type J





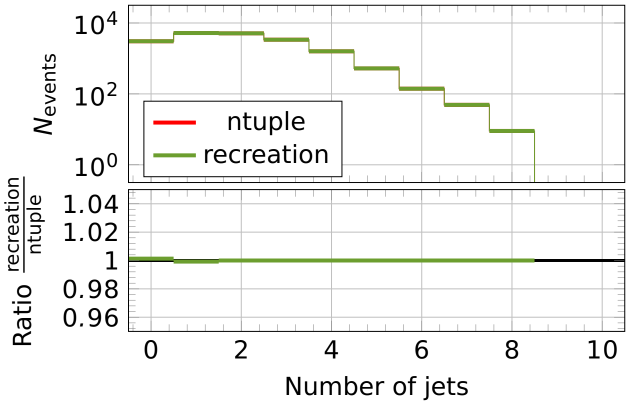

# Control distributions

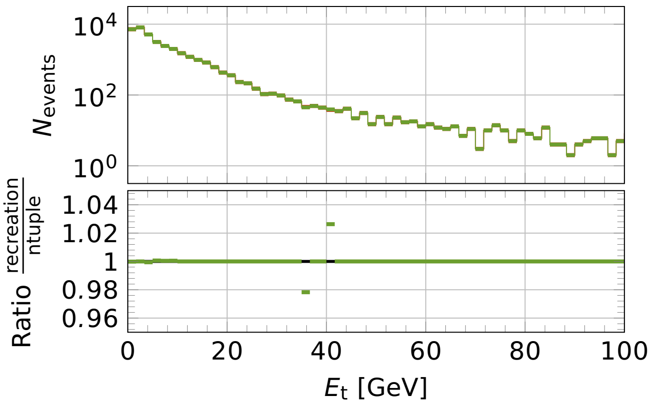
## Jet type K

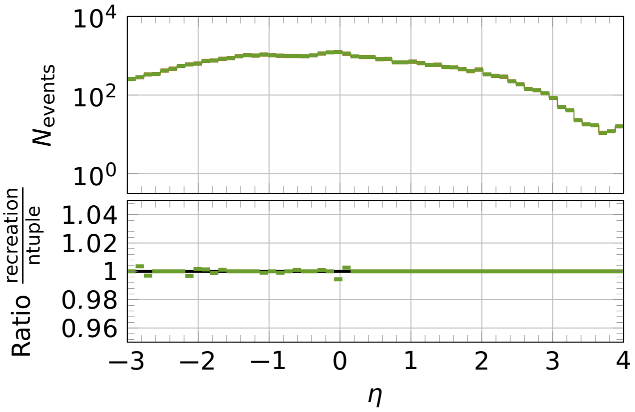


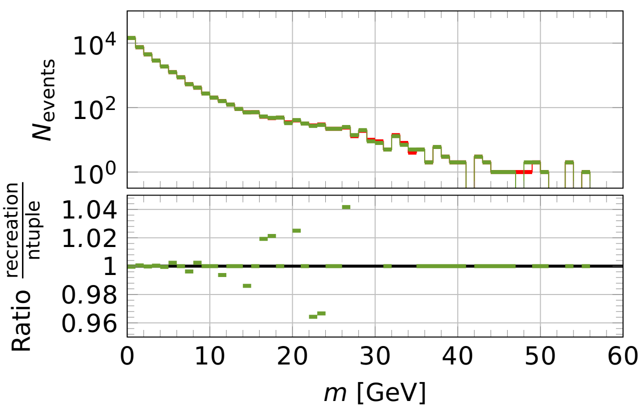
# Control distributions

## Jet type L



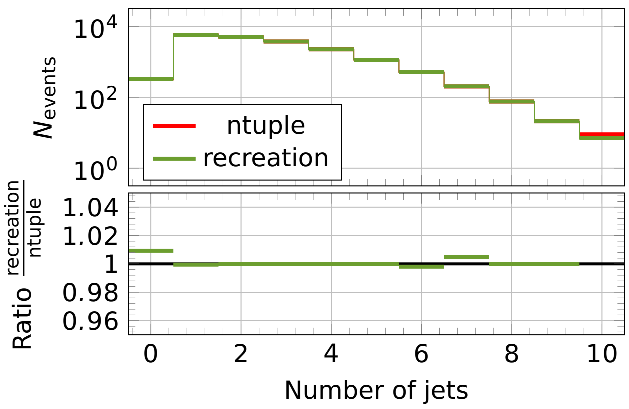



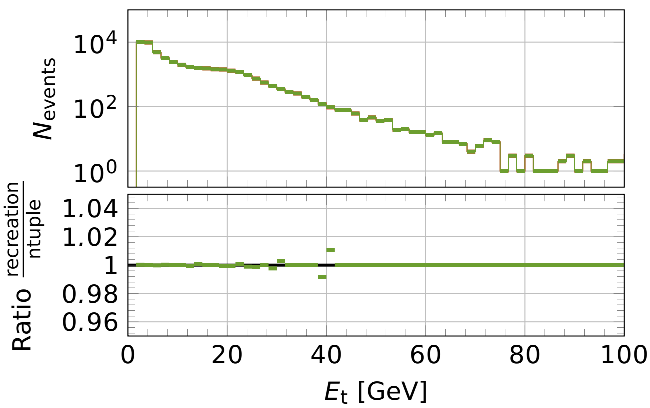



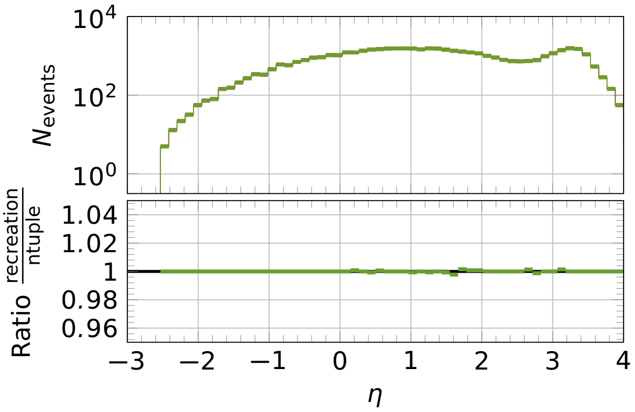



# Control distributions

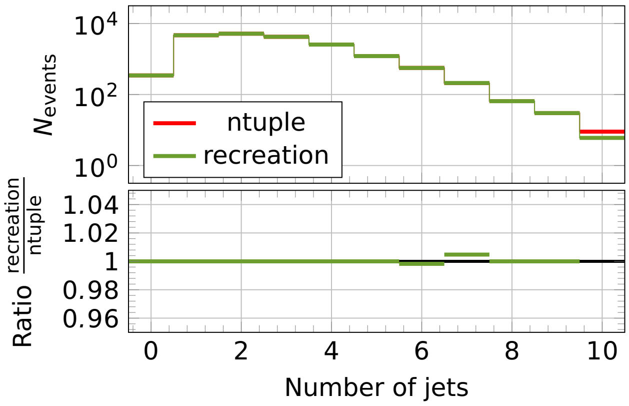
## Jet type M

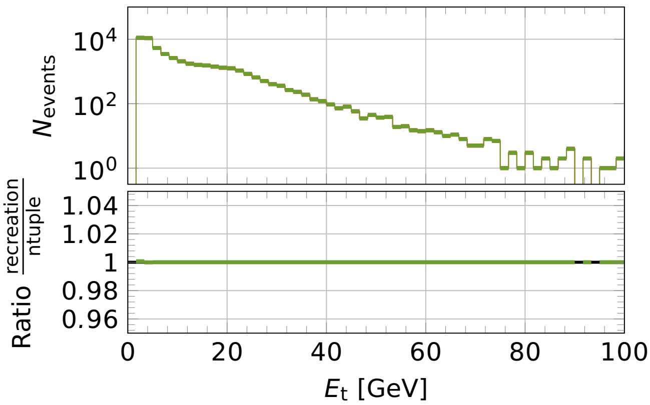


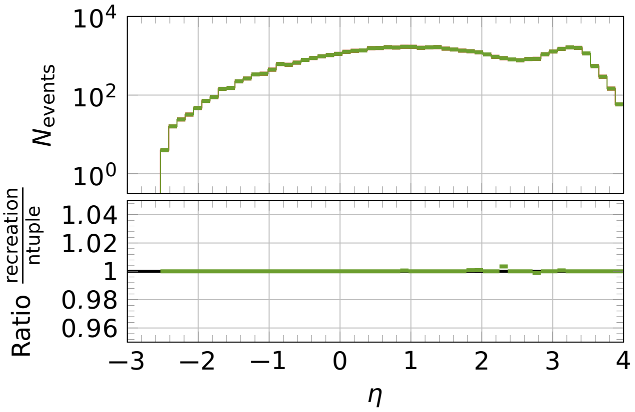



# Control distributions
## Jet type N

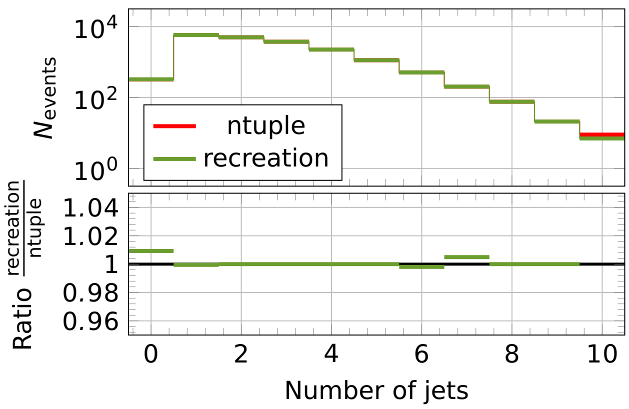


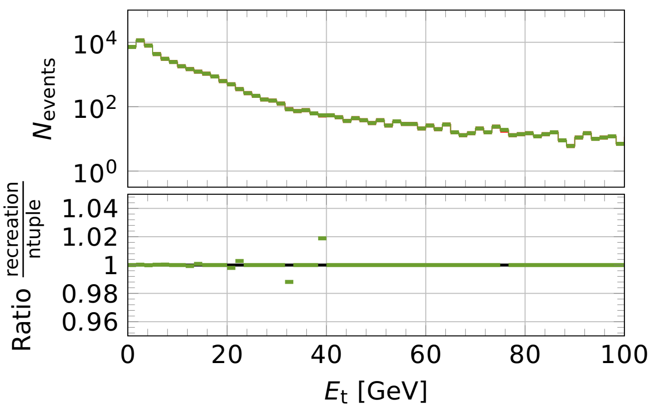



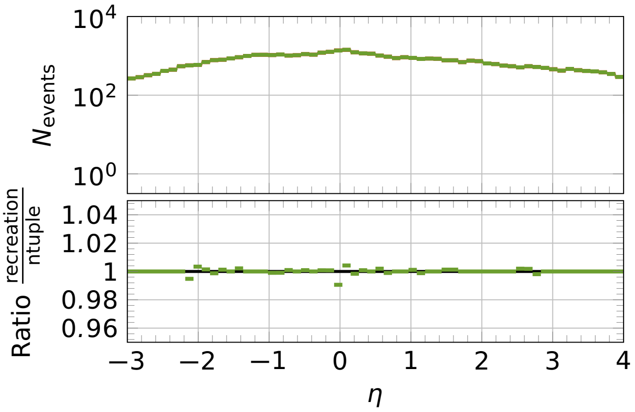






# Control distributions
## Jet type O

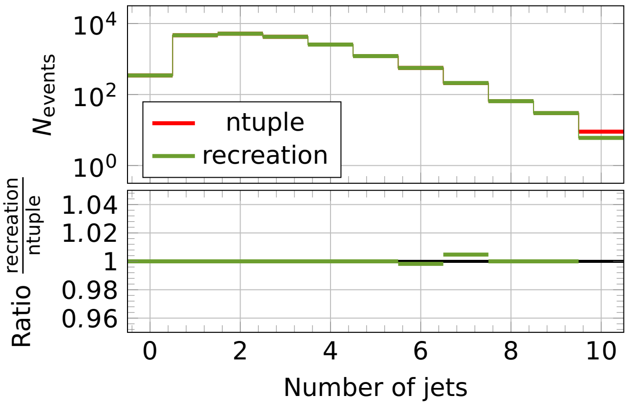


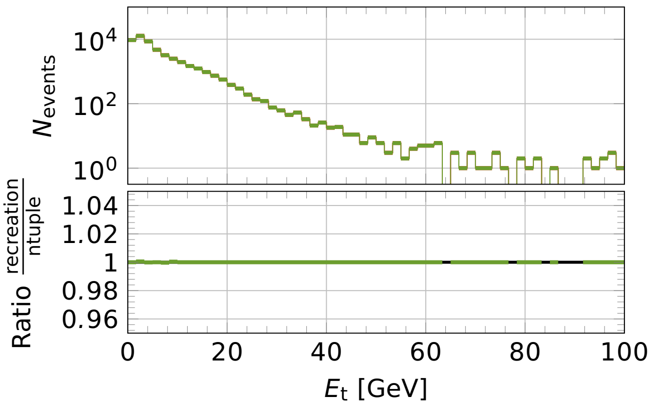



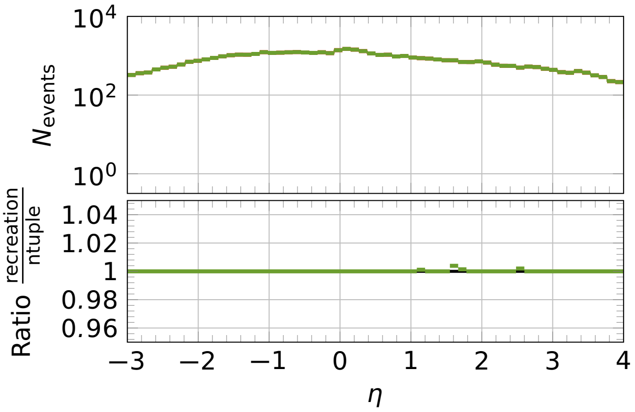



# Control distributions

## Jet type P



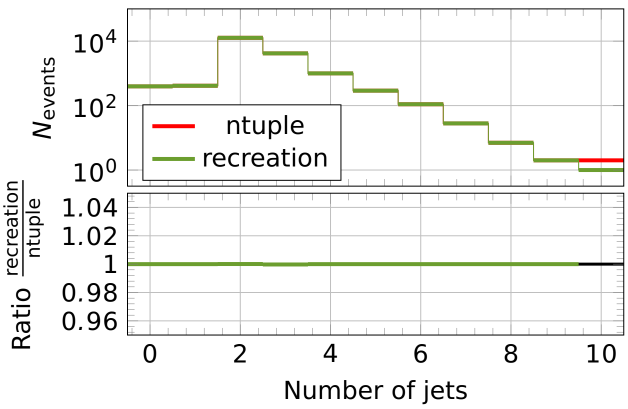



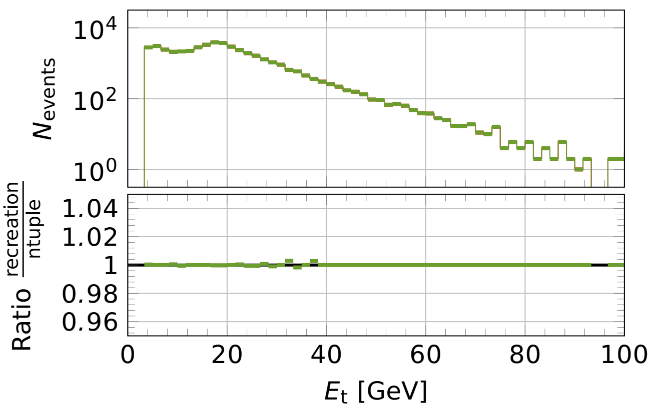


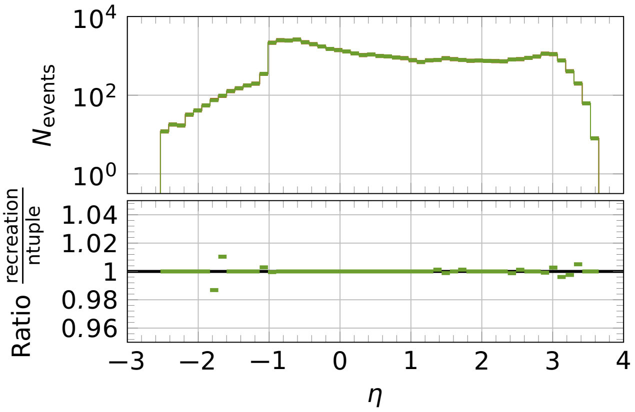

DESY.

# Control distributions

## Jet type S

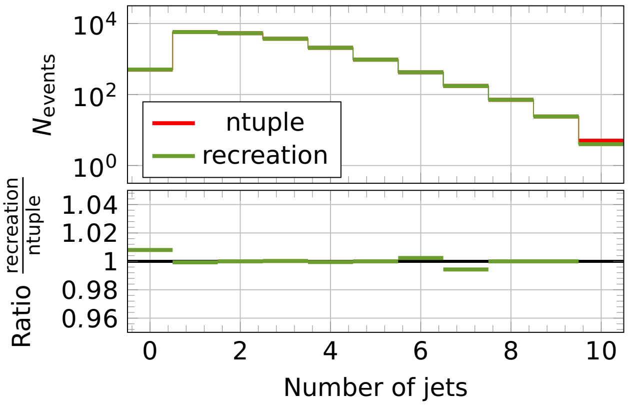


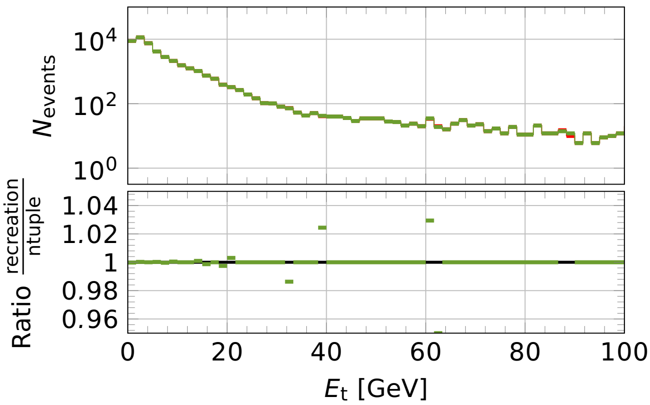



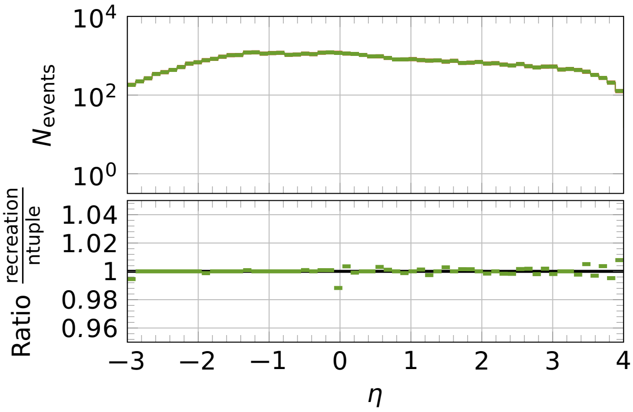



# Control distributions

## Jet type T






# Control distributions

## Jet type U