Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Current status

Nikol Madzharova
22.05.2025

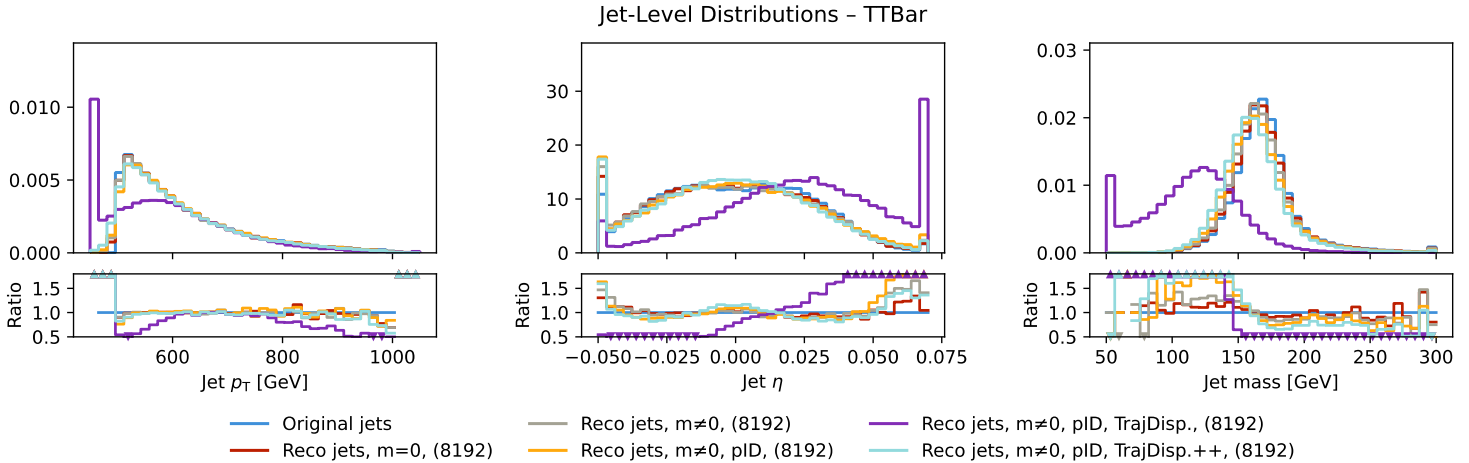
Foto: UHH/Lutsch

Feature Implementation Overview

Feature Name	Implemented
part_pt	✅ Yes
part_etarel	✅ Yes
part_phirel	✅ Yes
part_mass	✅ Yes
part_charge	✅ Yes
part_isChargedHadron	✅ Yes
part_isNeutralHadron	✅ Yes
part_isPhoton	✅ Yes
part_isElectron	✅ Yes
part_isMuon	✅ Yes
part_d0val	❌ No
part_dzval	❌ No
part_d0err	❌ No
part_dzerr	❌ No

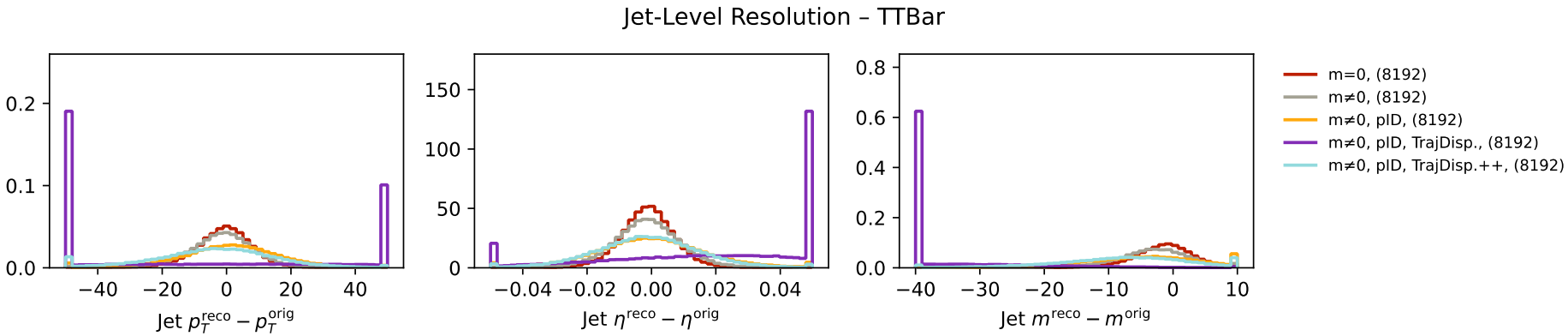
- The features marked **“No”** are those where **VQ-VAE struggles to correctly encode and decode** the values.
- Ensuring their **successful implementation** is the **main focus** of my master’s project work.

Jet-Level Distributions

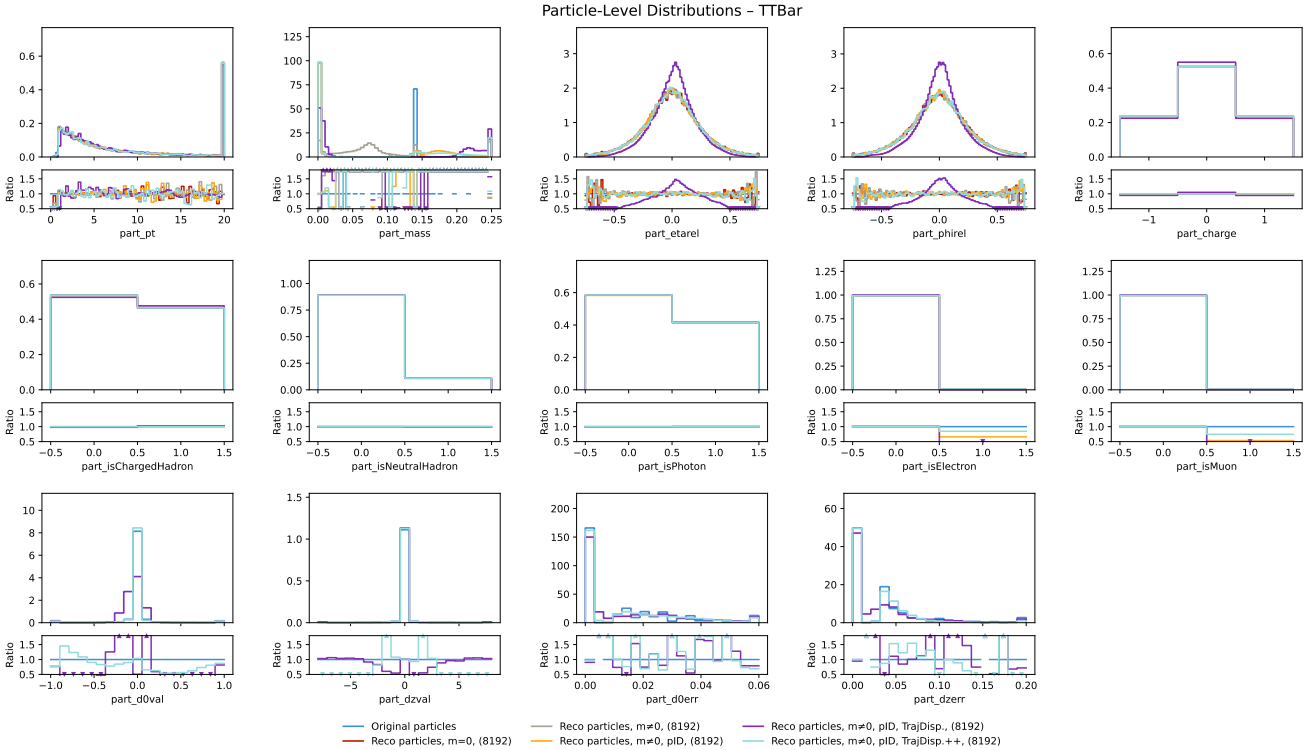


- The **purple line** (no preprocessing) shows severe distortion — preprocessing is clearly necessary.
- The **light blue line**, using tanh as a preprocessing step, improves the shape significantly.
- However, applying tanh introduces **numerical instability** —numpy may yield **NaN values** due to limited floating-point precision, especially when input magnitudes are large. (e.g. np.arctanh(np.tanh(100)) = ∞)

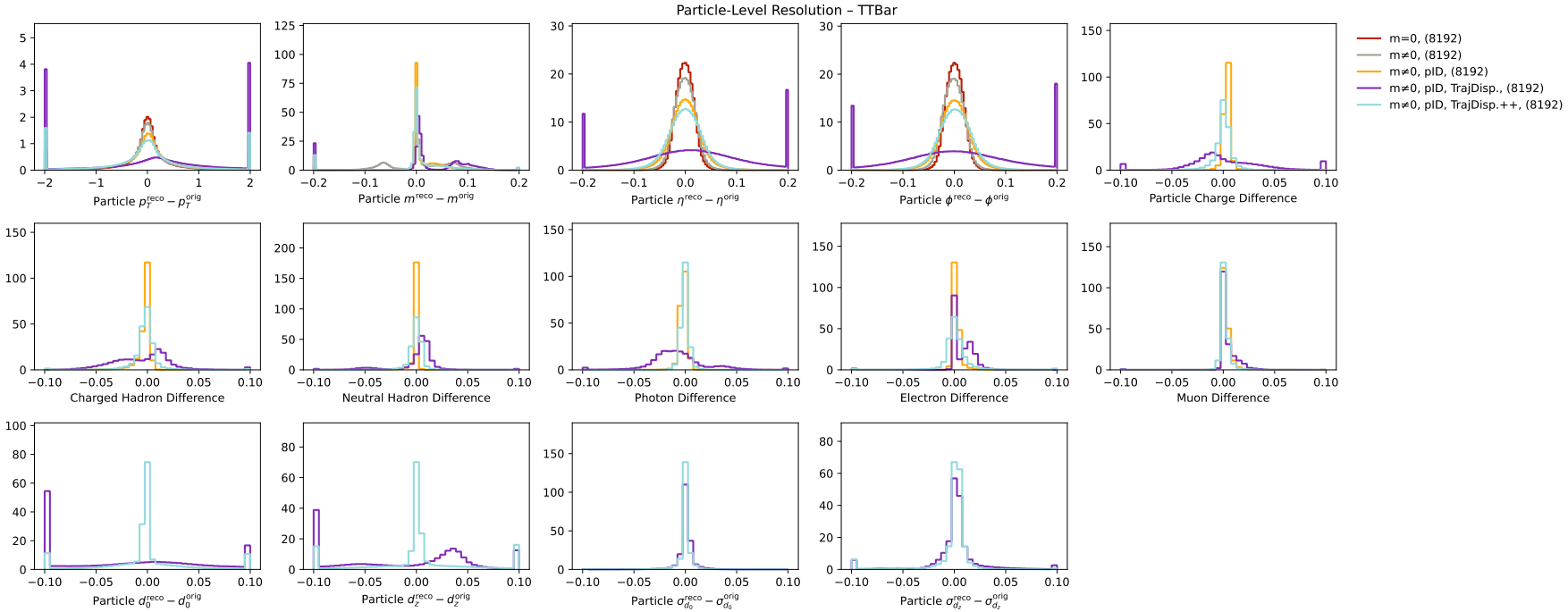
Jet-Level Resolution



Particle-Level Distributions



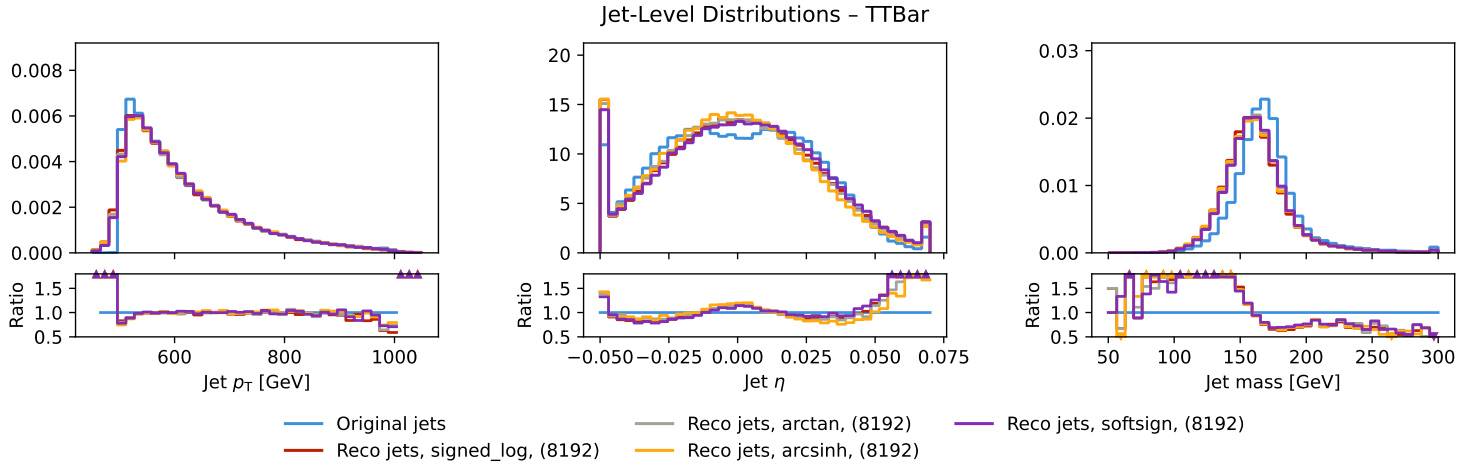
Particle-Level Resolution



Key Considerations for Choosing a Preprocessing Function

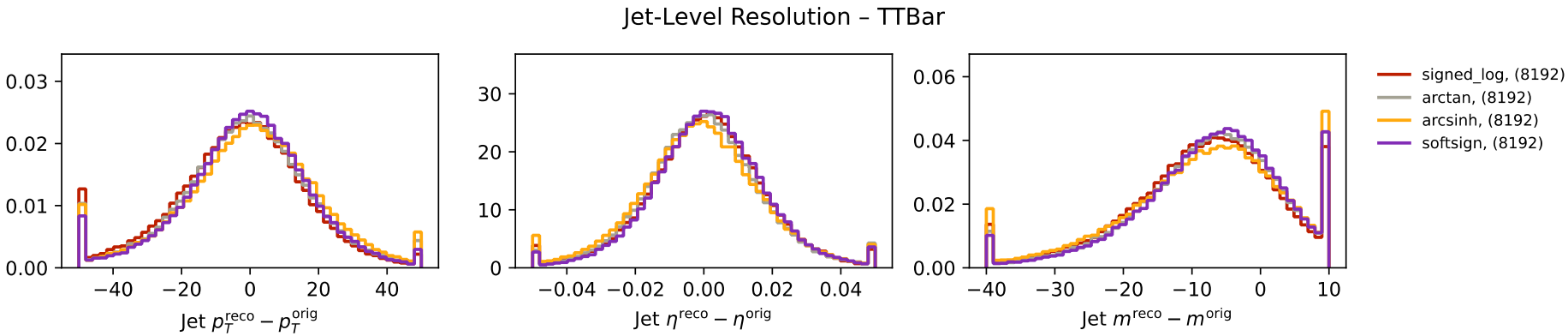
- ~52% of particles are neutral with **0 displacement,**
  big peak around 0
- **Handles long tails**: Values range up to ±1000
- **Invertible**:
  Needed for reconstruction in VQ-VAE → function and inverse must be numerically stable.

Jet-Level Distributions

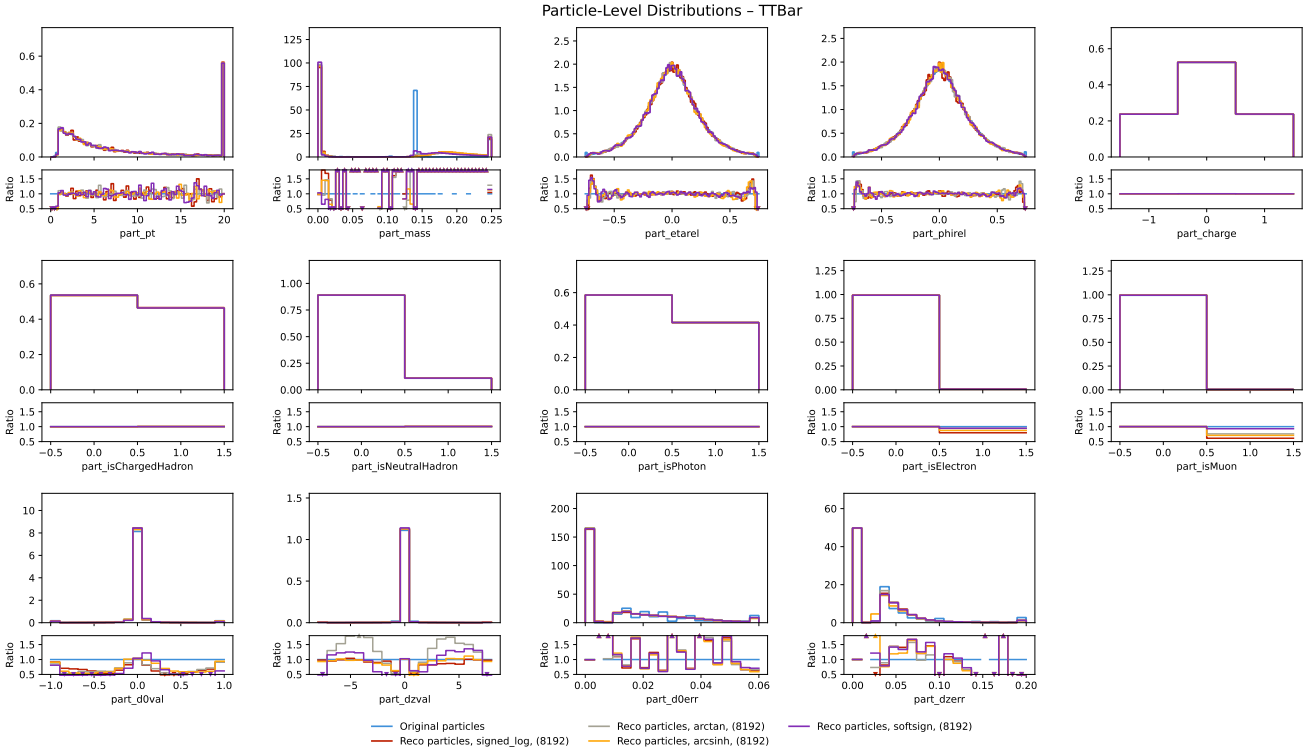


- signed_log = sgn(x) * ln(1 + |x| + epsilon), where epsilon = 10^-5
  signed_exp(x) = sgn(x) * ((e^|x|) - 1 - epsilon)
- Softsign= x / (1 + |x|)
  softsign_inv(y) = y / (1 - |y|)

Jet-Level Resolution



Particle-Level Distributions



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Particle-Level Resolution