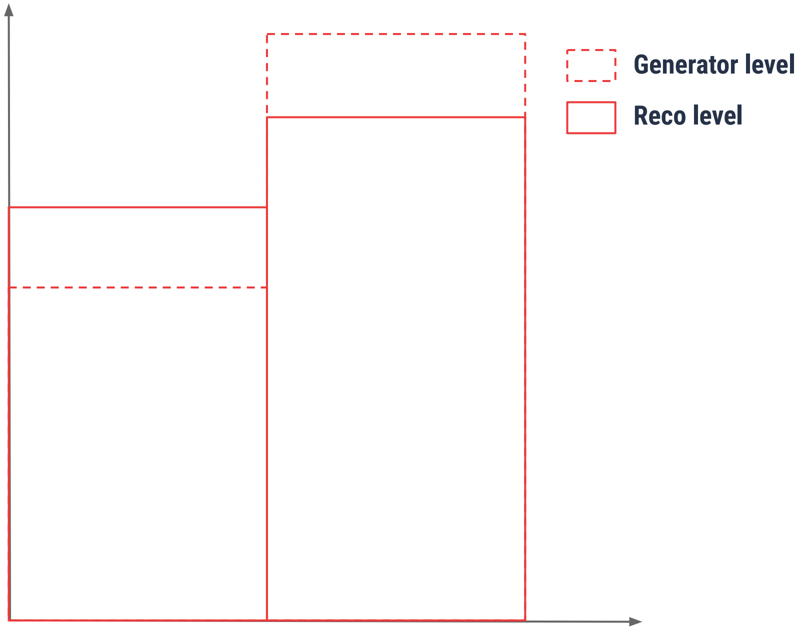
Machine learning-based unfolding with Omnifold and measurement of jet substructure with H1 data

Vinicius M. Mikuni

Unfolding

- We only have access to observables at **reconstruction level**, i.e after detector effects
- When comparing different theories, we want to compare observables before detector interaction (**generator level**):
  - Don’t require theorists to have expert detector knowledge to compare their predictions
  - Easier to maintain and incorporate new calibration routines for detector simulation
- What I’m **not** talking about today:
  - [IBU/D’Agostini method]
  - [SVD]
  - Matrix inversion
  - Other methods for unfolding using histograms



Unfolding

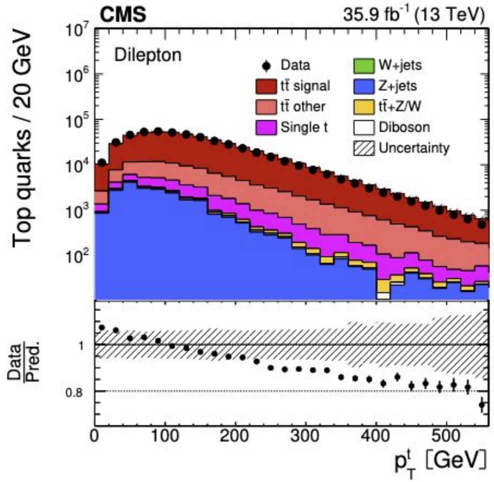
Traditional methods for unfolding are performed using **histograms**

- Well understood statistical properties
- Clear convergence criteria

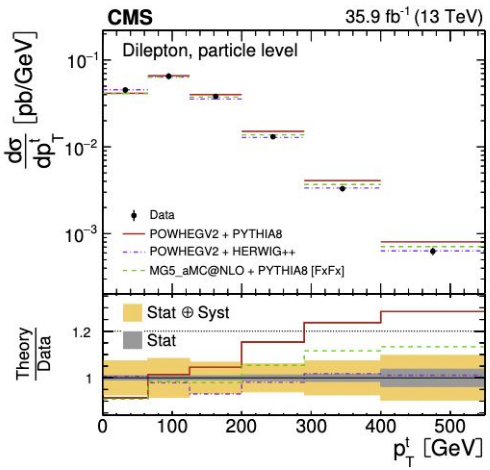
Limitations:

- Histograms need to be defined before unfolding.
  - If a different binning is required, the full unfolding routine needs to be redone
- Often able to address only 1 observable at a time
  - Multi-dimensional histograms are harder to deal with: curse of dimensionality

Reco level



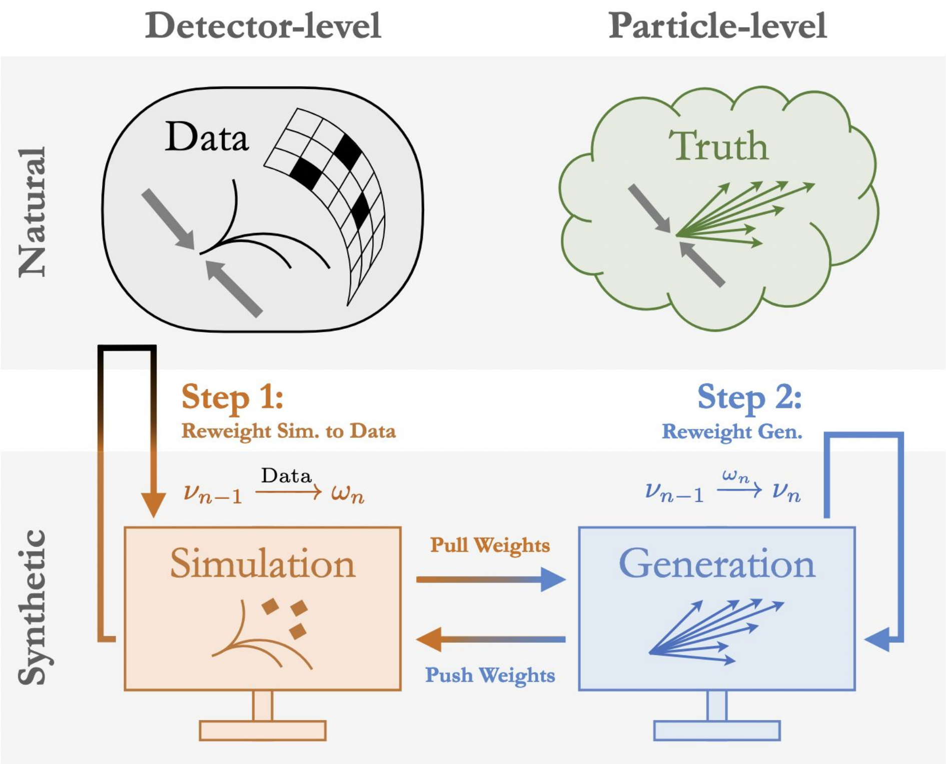
Generator level



J. High Energ. Phys. **2019,** 149 (2019).

Omnifold*

* Andreassen et al. PRL 124, 182001 (2020)
For unfolding using **invertible networks** see:
- SciPost Phys. 9 (2020) 074 e-Print: [2006.06685](https://arxiv.org/abs/2006.06685)



ML is used to define a method for unfolding that is unbinned and can use multiple distributions at a time

2 step iterative approach
- Simulated events after detector interaction are reweighted to match the data
- Create a "new simulation" by transforming weights to a proper function of the generated events

Machine learning is used to approximate **2** likelihood functions:
- **reco MC to Data** reweighting
- **Previous** and **new Gen** reweighting

* Andreassen et al. PRL 124, 182001 (2020)

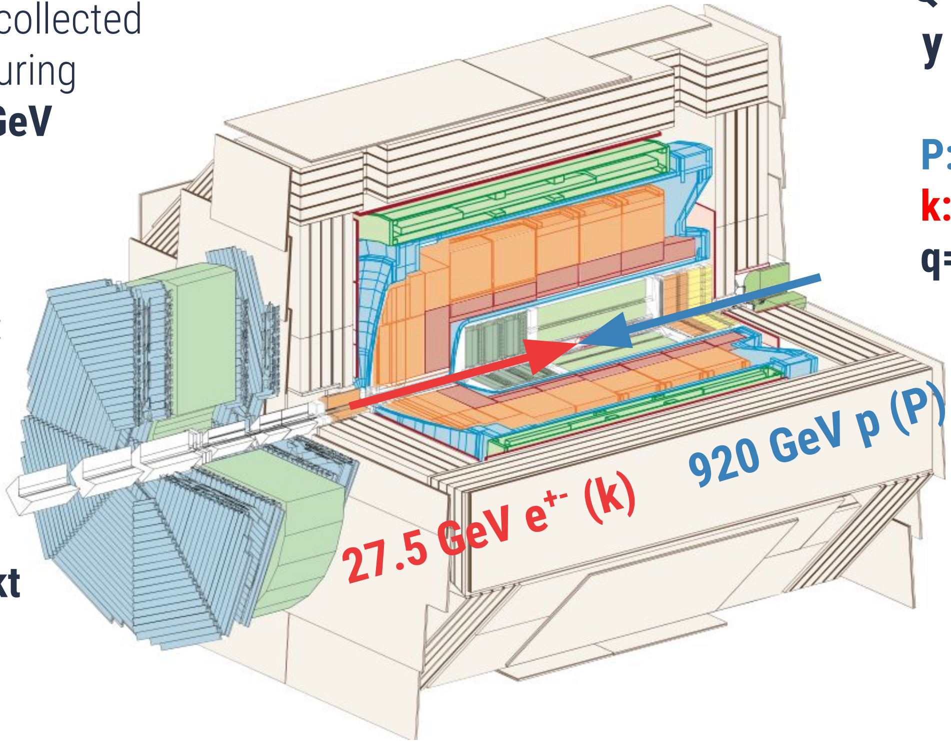
Experimental setup

Using **228 pb^{-1}** of data collected by the **H1 Experiment** during **2006** and **2007** at **318 GeV center-of-mass energy**

Phase space definition:

- $0.2 < y < 0.7$
- $Q^2 > 150$ GeV^2
- Jet $p_T > 10$ GeV
- $-1 < \eta_{lab} < 2.5$

Jets are clustered with **kt** algorithm with **R=1.0**



$Q^2 = -q^2$

$y = Pq / pk$

P: incoming proton 4-vector

k: incoming electron 4-vector

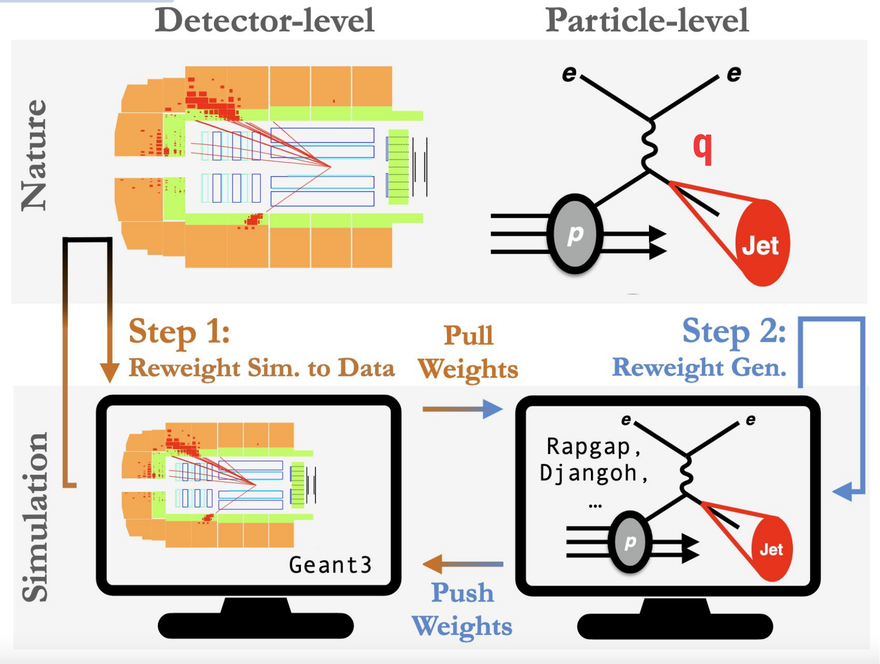
q=k-k' : 4-momentum transfer

Reconstructed hadrons using combined detector information: **energy flow algorithm**

TMD Measurement

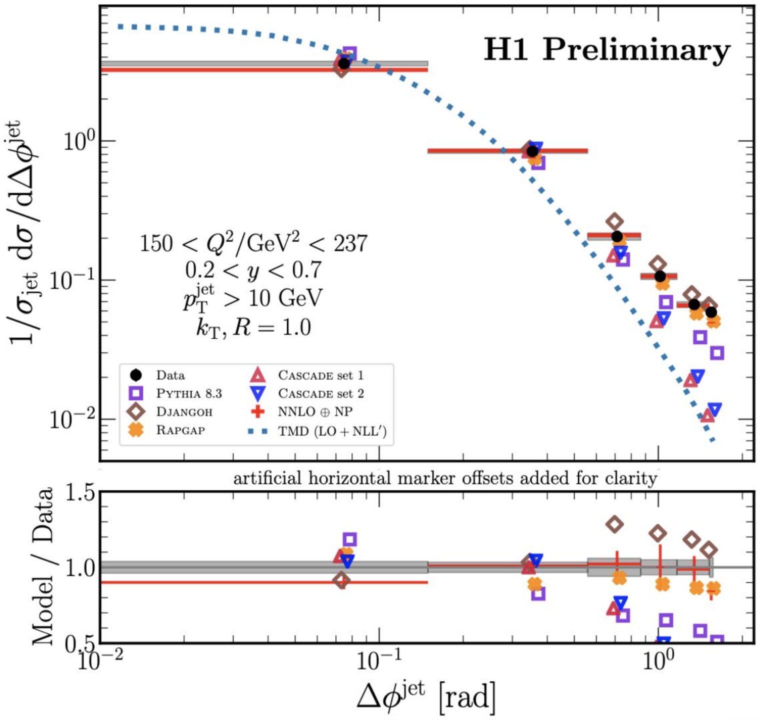
TMD sensitive observables are unfolded using jet and lepton information

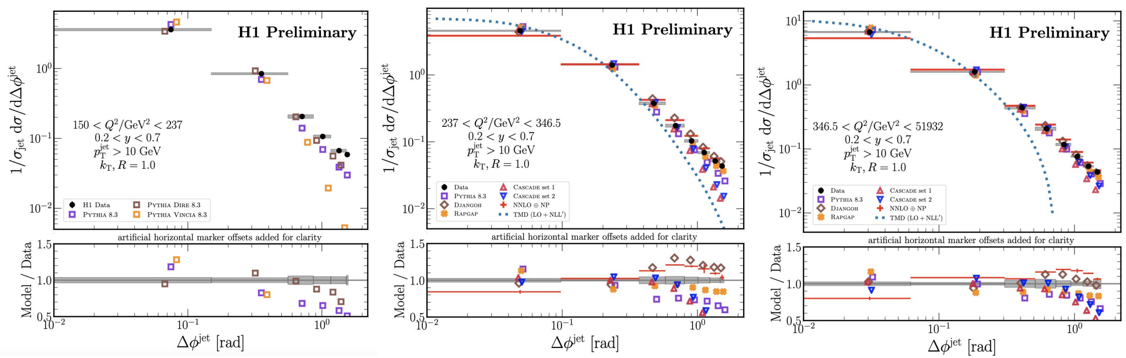
- Both jet and lepton kinematics are used as inputs to OmniFold
- Follow up work from [Phys. Rev. Lett. 128, 132002]
- New observables can also be calculated **after** the unfolding procedure!
- Study the evolution of the observables with energy scale $\mathbf{Q^2 = -q^2}$



More details at [H1Prelim-22-031]

TMD Measurement





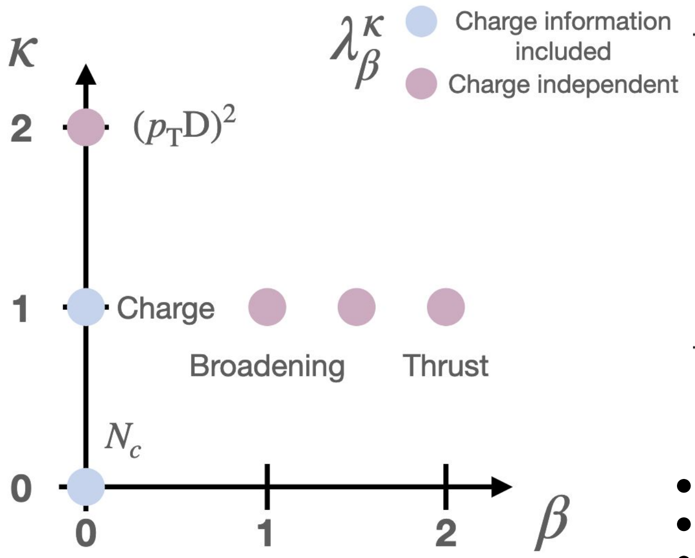
- $\mathbf{Q^2}$ is not directly unfolded, but can be calculated from the unfolded distributions!

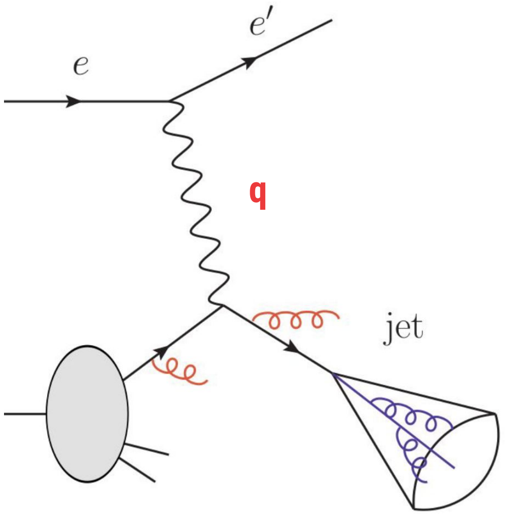
More details at [H1Prelim-22-031]

Jet angularities

Use jet observables to study different aspects of QCD physics:

- IRC safe λ^1_a, a = [0,0.5,1] and unsafe $\mathbf{p_TD}$ angularities
- Charge dependent observables: $\mathbf{Q_j}$ and $\mathbf{N_c}$
- Study the evolution of the observables with energy scale $\mathbf{Q^2 = -q^2}$

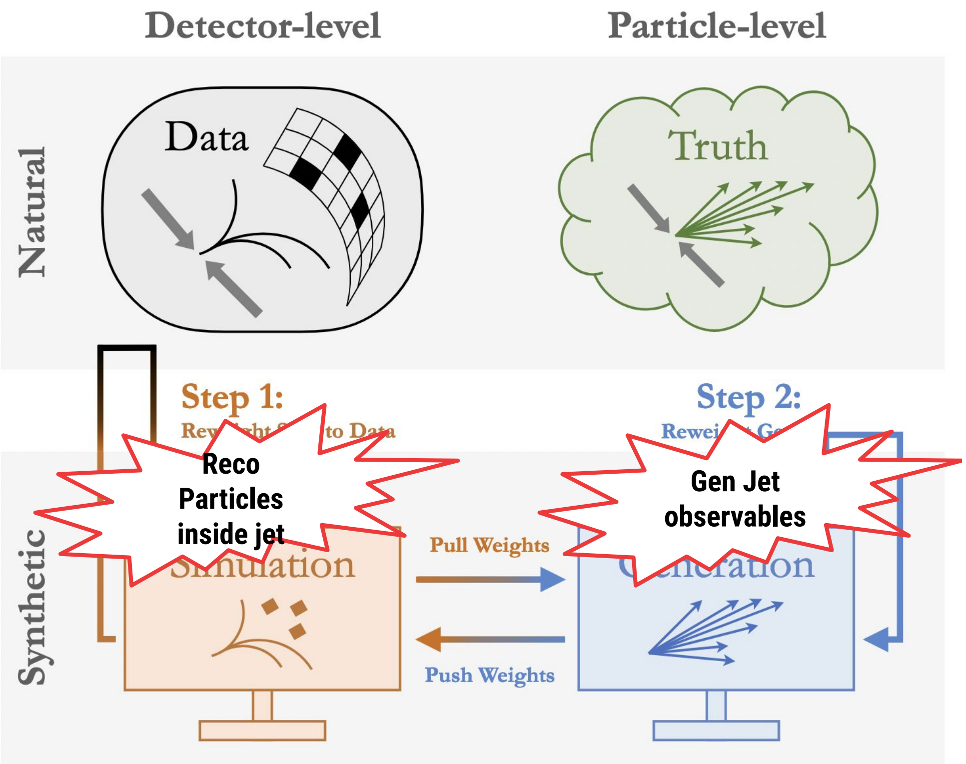




- z_i: longitudinal momentum fraction
- q_i: charge
- R_i distance from jet axis in (eta,phi)

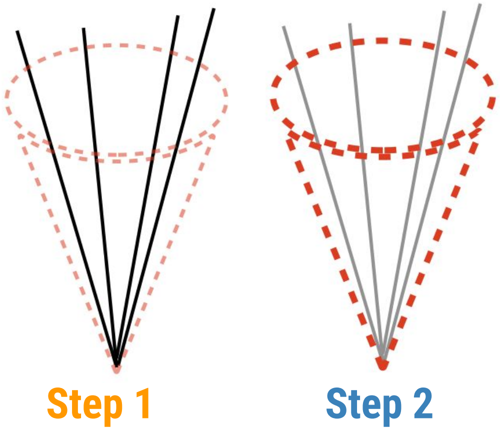
$$\lambda^\kappa_\beta = \sum_{i\in\text{jet}} z_i^\kappa \left(\frac{R_i}{R_0}\right)^\beta \qquad \tilde{\lambda}^\kappa_0 = Q_\kappa = \sum_{i\in\text{jet}} q_i \times z_i^\kappa .$$

Omnifold



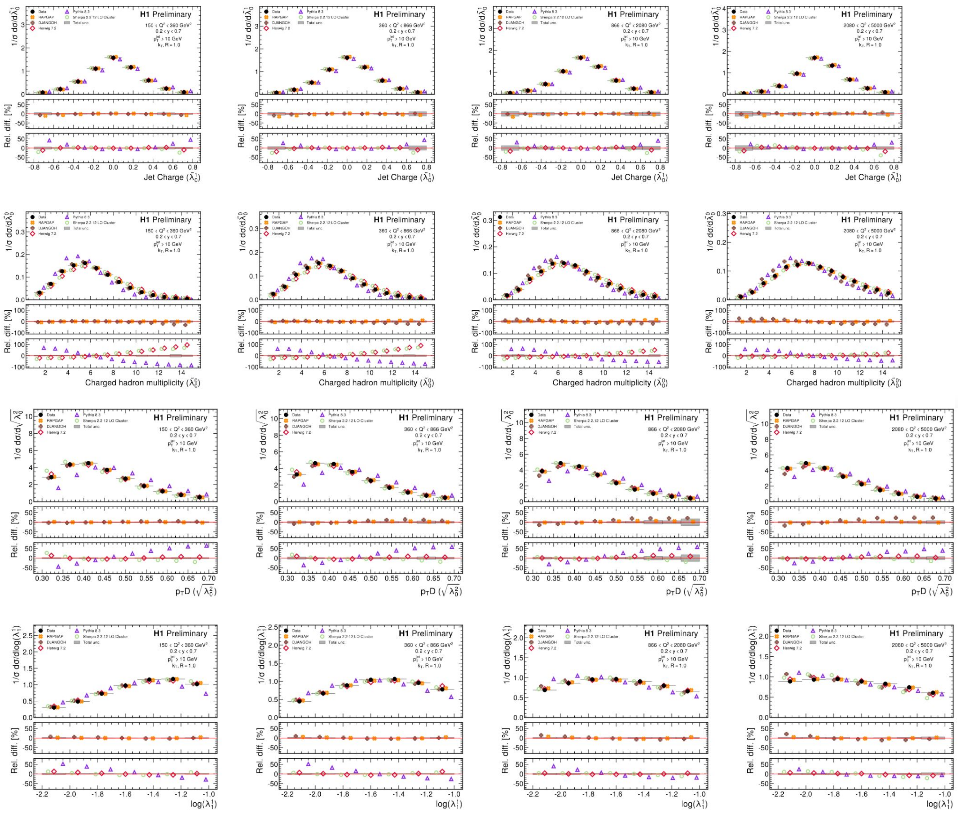
Different input levels for each step

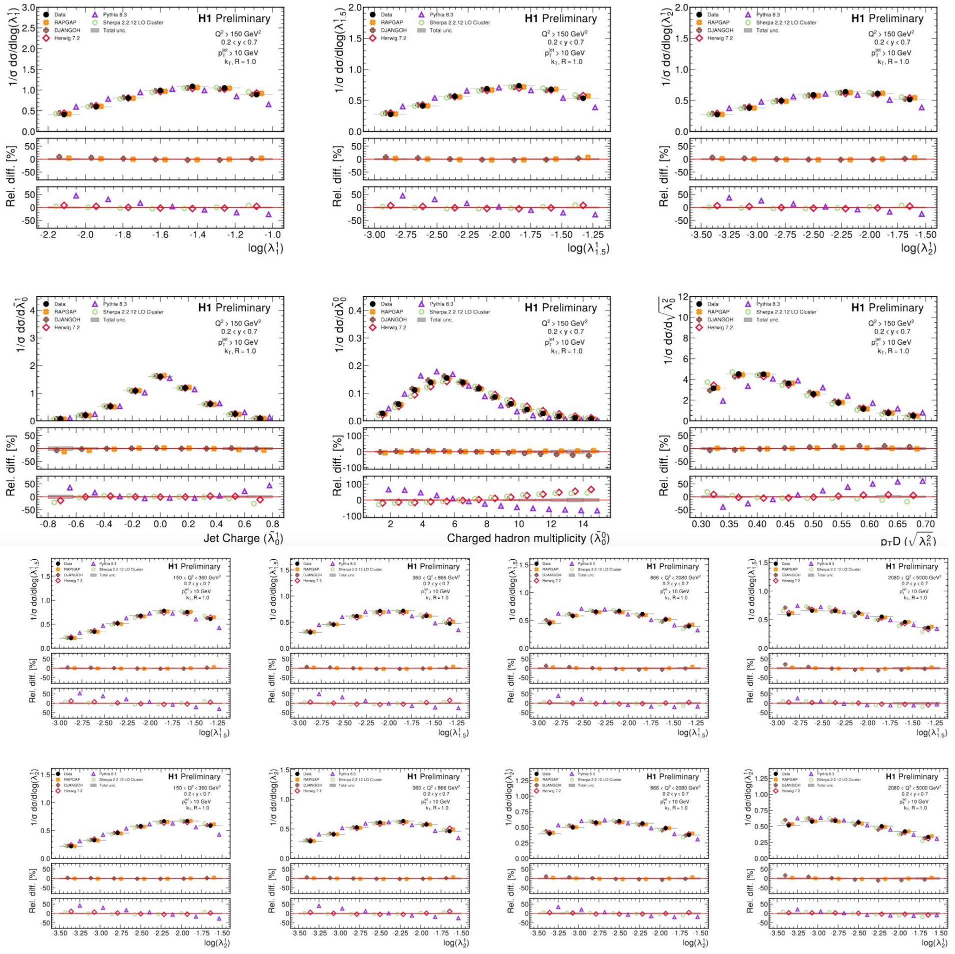
- **Step 1** particles are used as inputs
- **Step 2** uses the set of observables planned to unfold



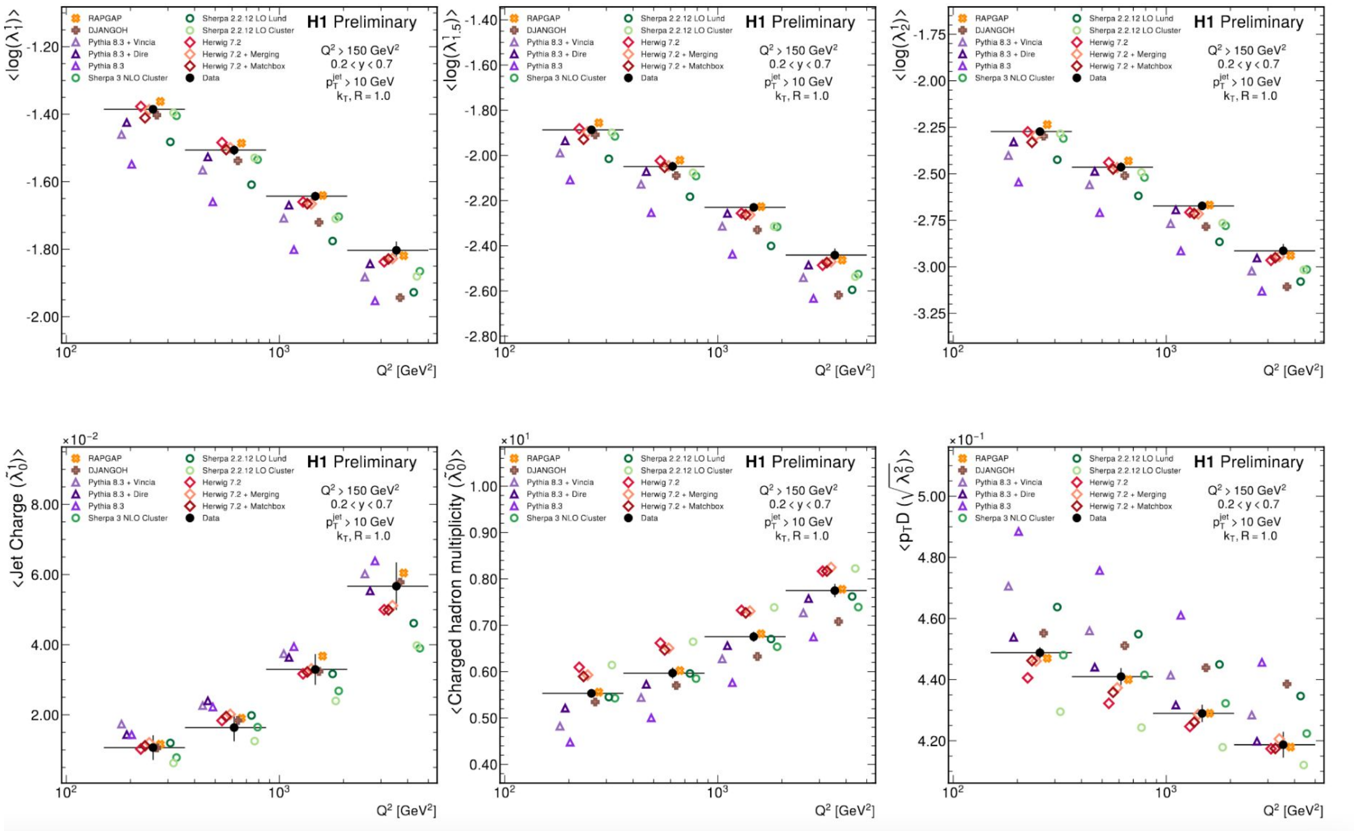
Multi-differential

All Q^2 intervals and **distributions** are unfolded **simultaneously!**



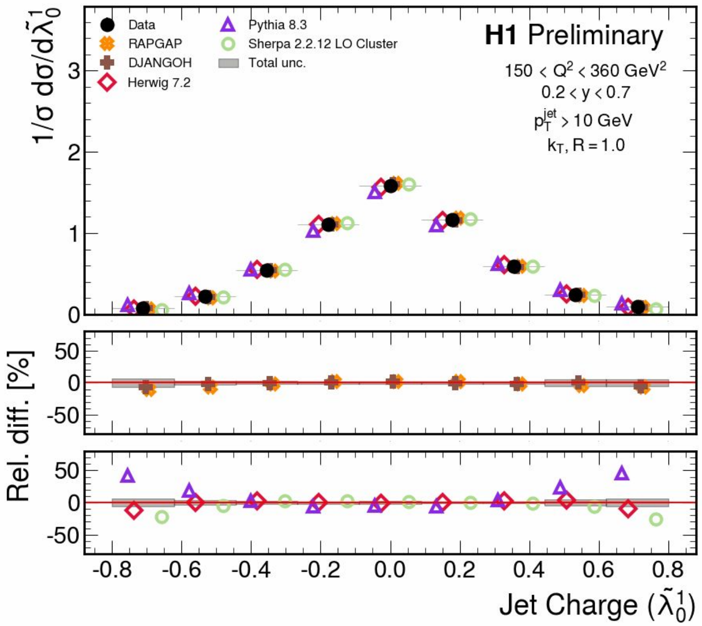


Multi-differential

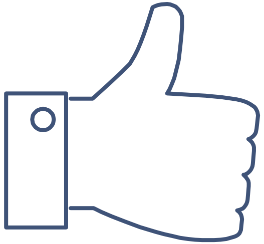


First moment (mean) of **all** jet substructure observables are measured **vs Q^2**

Conclusions and prospects



- ML based unfolding can drastically change the way we measure physics observables
  - Open the possibility for **generator level information before the measurement is performed**
  - More information is incorporated to the unfolding model, leading to **proper correlations** between inputs and **possibly lower detector uncertainties**
  - **Binning independent** and can be quickly changed after unfolding
  - Trivial to **estimate statistical properties** of the data after unfolding, such as moments of the distribution
- More info in the preliminary results available at: H1prelim-22-034, H1Prelim-22-031



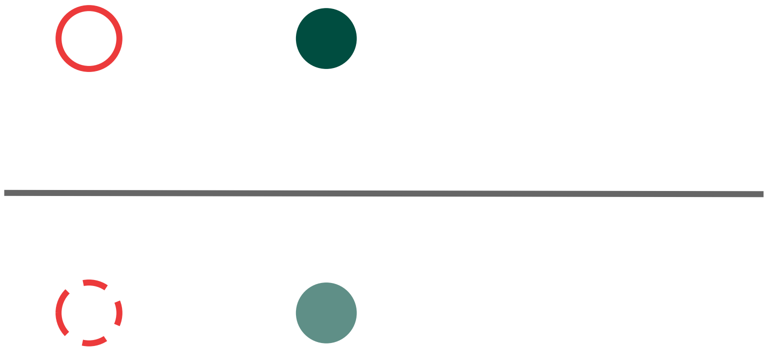
THANKS!

Any questions?

Backup

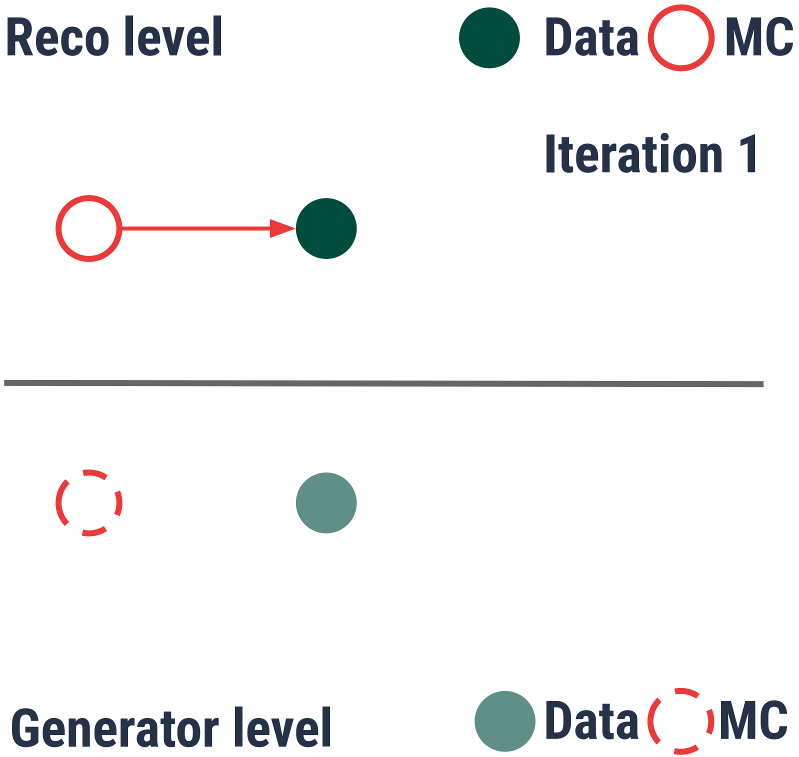
Omnifold

Reco level Data MC



Generator level Data MC

Omnifold



Step 1:

- Train a classifier to separate **data** from **MC** events
- Reweight **reco level MC** with weights:

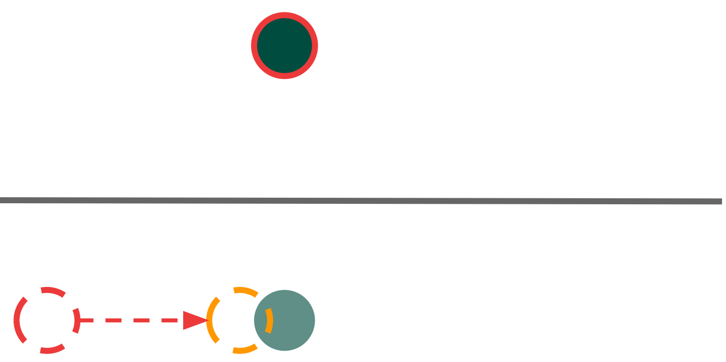
$$W(\mathrm{reco}) = p_{\mathrm{Data}}(\mathrm{reco})/p_{\mathrm{MC}}(\mathrm{reco})$$

Omnifold

Reco level

Data MC

Iteration 1



Generator level

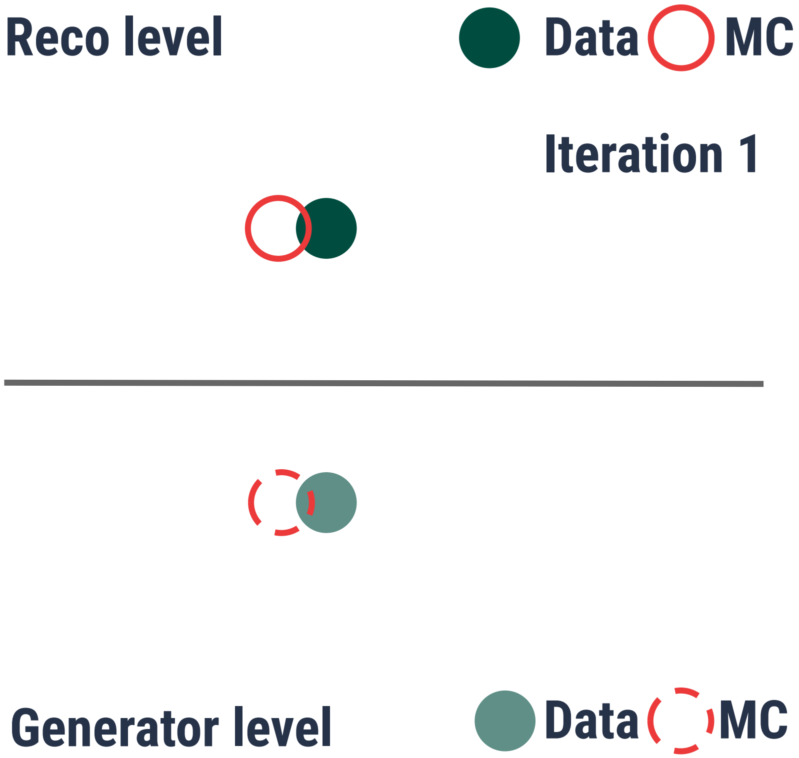
Data MC MC reweighted

Step 2:

- **Pull weights** from **step 1** to generator level events
- Train a classifier to separate **initial MC at gen level** from **reweighted MC** events
- Define a **new simulation** with weights that are a **proper function of gen level kinematics**

$$W(gen) = p_{weighted\ MC}(gen)/p_{MC}(gen)$$

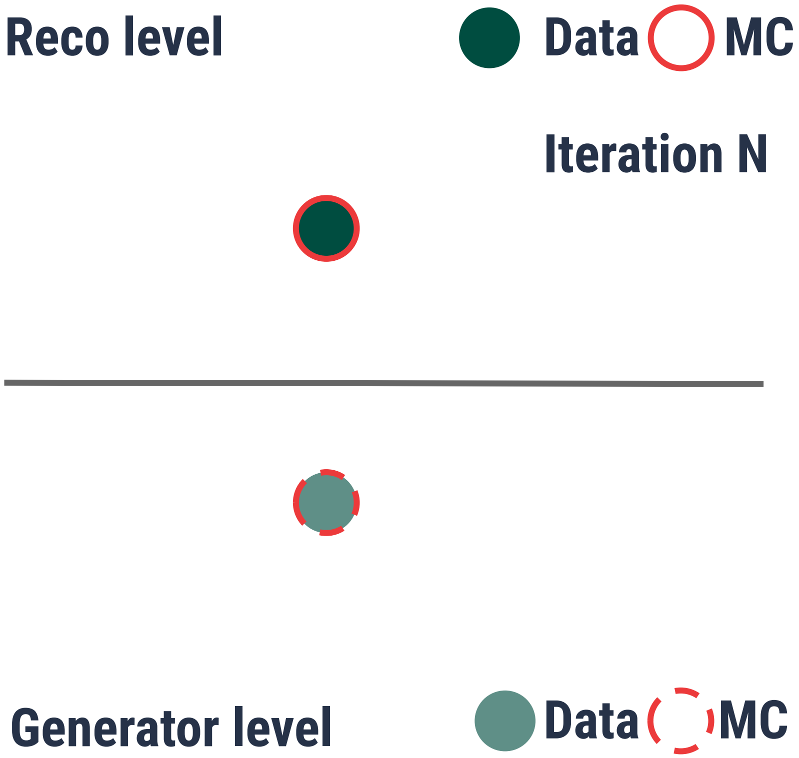
Omnifold



Start again from **step 1** using the **new simulation** after **pushing** the weights from **step 2**

- Guaranteed convergence to the maximum likelihood estimate of the generator-level distribution when number of iterations go to infinite
- In practice, less than 10 iterations are enough to achieve convergence

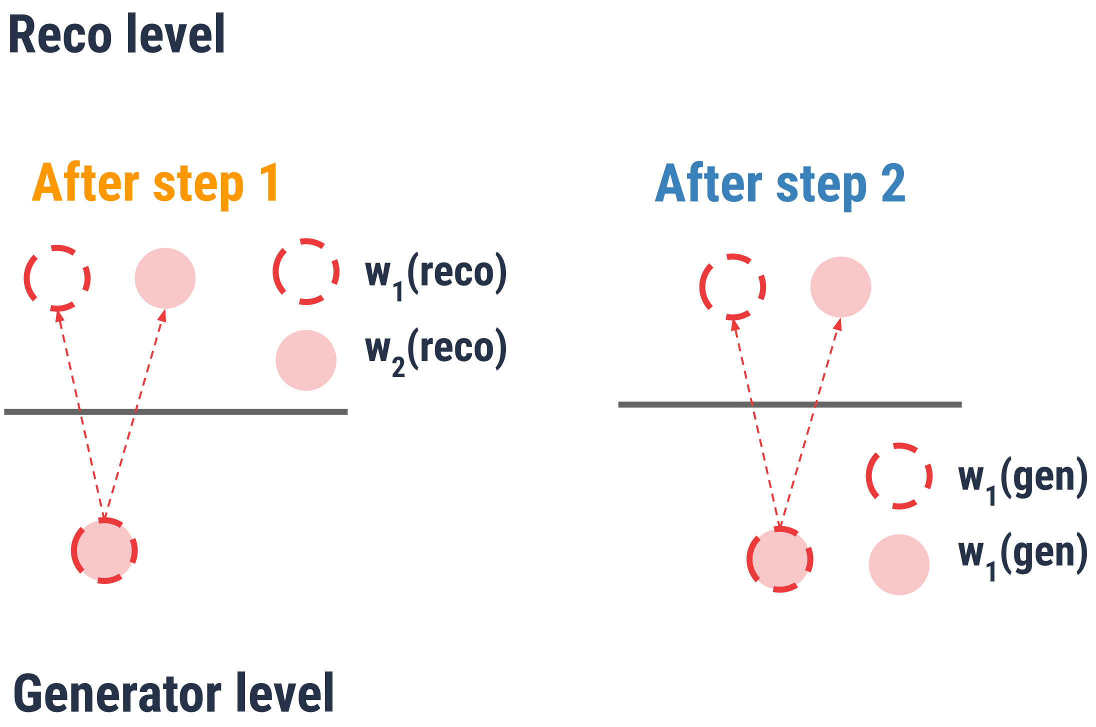
Omnifold



Start again from **step 1** using the **new simulation** after **pushing** the weights from **step 2**

- **Guaranteed convergence** to the maximum likelihood estimate of the generator-level distribution when number of iterations goes to infinite
- In practice, **less than 10 iterations** are enough to achieve convergence

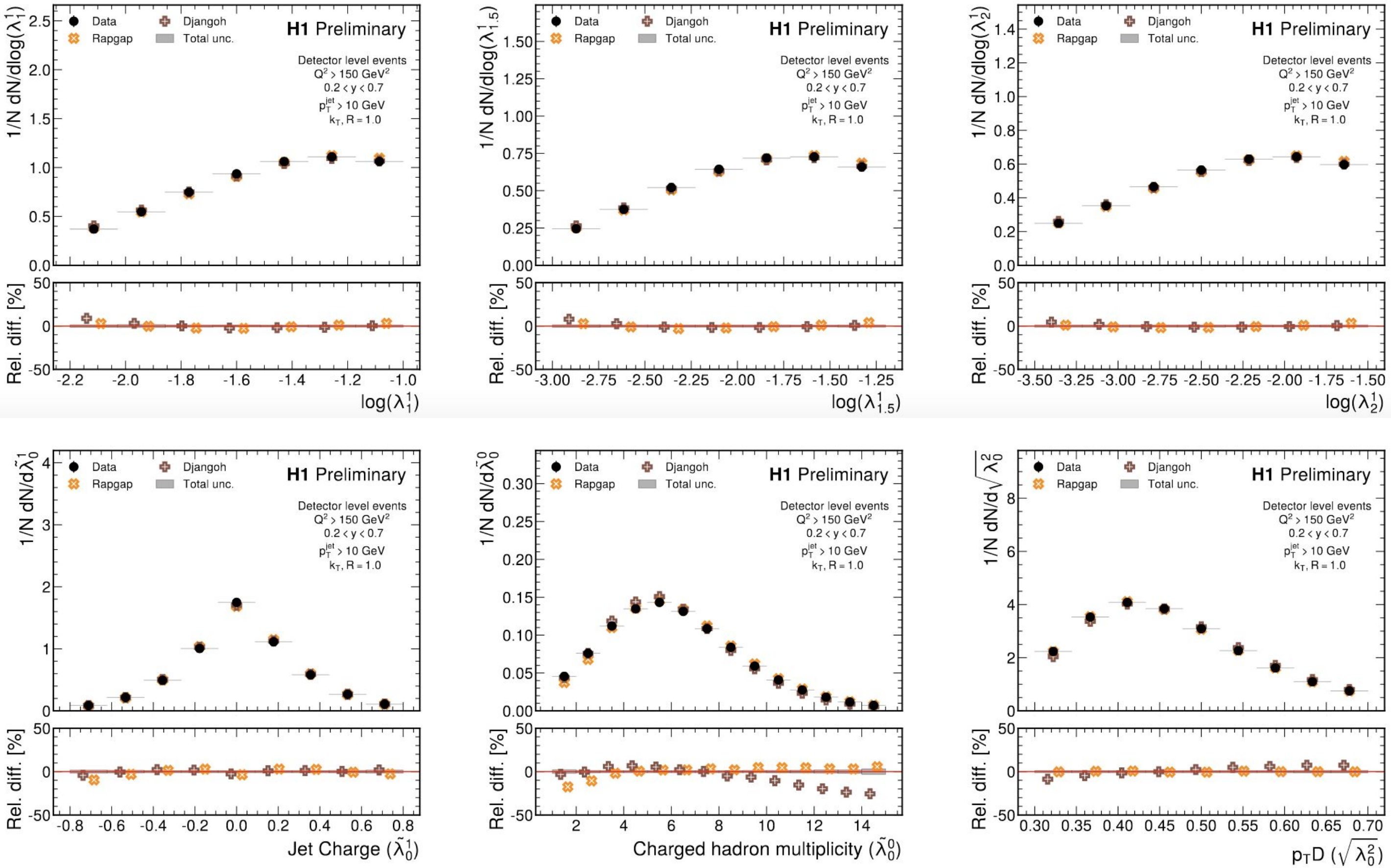
Why doesn't omnifold converge in a single iteration



We always need to push and pull weights that are often calculated using different kinetic information: **reco** and **particle level**

- After **step 1**, the same Gen event can give rise to distinct reco events (the process is **stochastic**!)
- That is fixed by **step 2**, which acts as a **regularizer**, but since data is not used, the unfolded response **moves away** from the maximum likelihood

Total experimental uncertainty at reconstruction level at the **% level**!



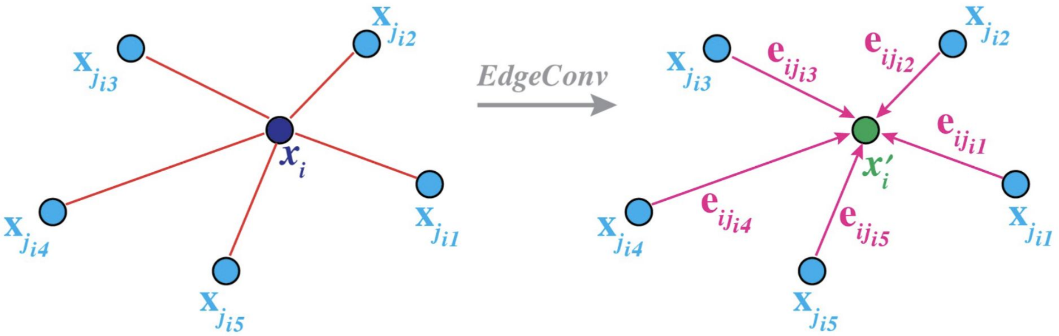
Systematic uncertainties

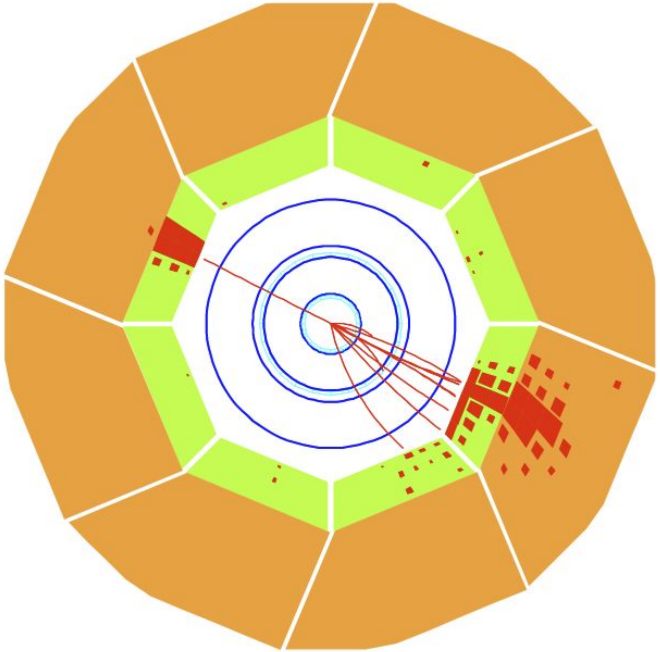
Systematic uncertainties currently considered

- **HFS energy scale:** +- 1%
- **HFS azimuthal angle:** +- 20 mrad
- **Lepton energy:** +- 0.5% (mainly affects Q^2)
- **Lepton azimuthal angle:** +- 1 mrad (mainly affects Q^2)
- **Model uncertainty:** differences in unfolded results between Djangoh and Rapgap
- **Non-closure uncertainty:** Differences between the expected and obtained values of the closure test
- **Statistical uncertainty:** Standard deviation of 100 bootstrap samples with replacement

Extracting particle information

- Particle information is extracted using a **Point cloud transformer*** model
- Model takes **kinematic properties** of particles as inputs
- Connect **k=10** nearest neighbors in η-φ to learn the relationship between particles.
- Built in symmetries: **permutation invariance**
- Consider up to **30** particles per jet





* V. Mikuni and F. Canelli 2021 *Mach. Learn.: Sci. Technol.* **2** 035027

Extracting particle information

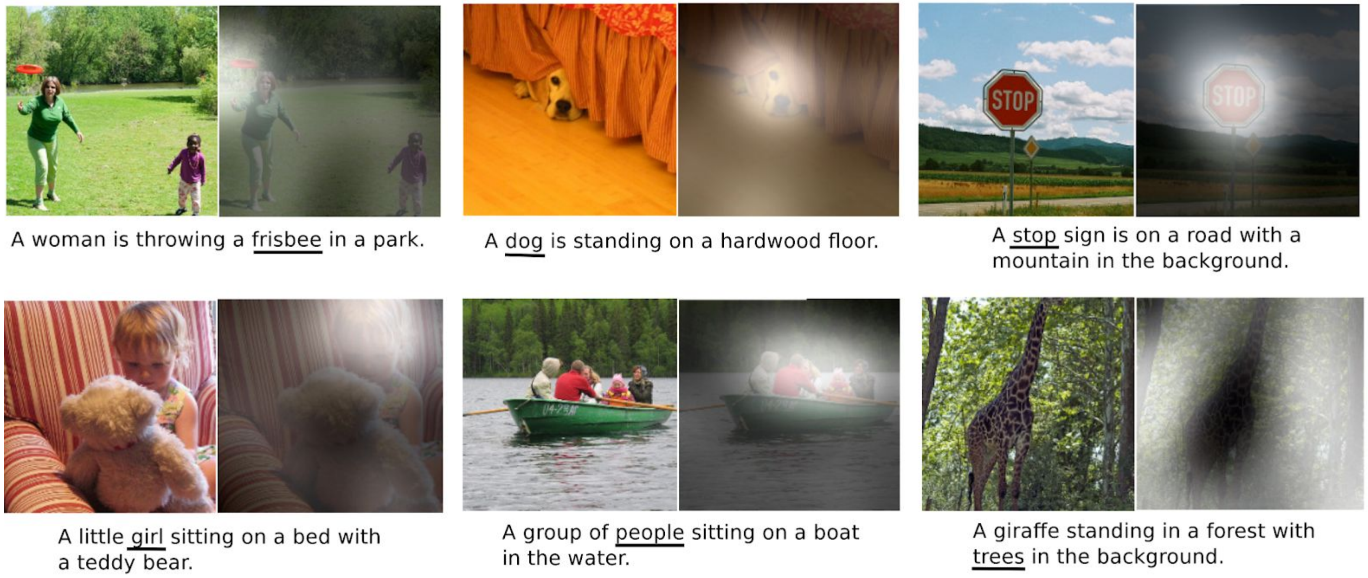


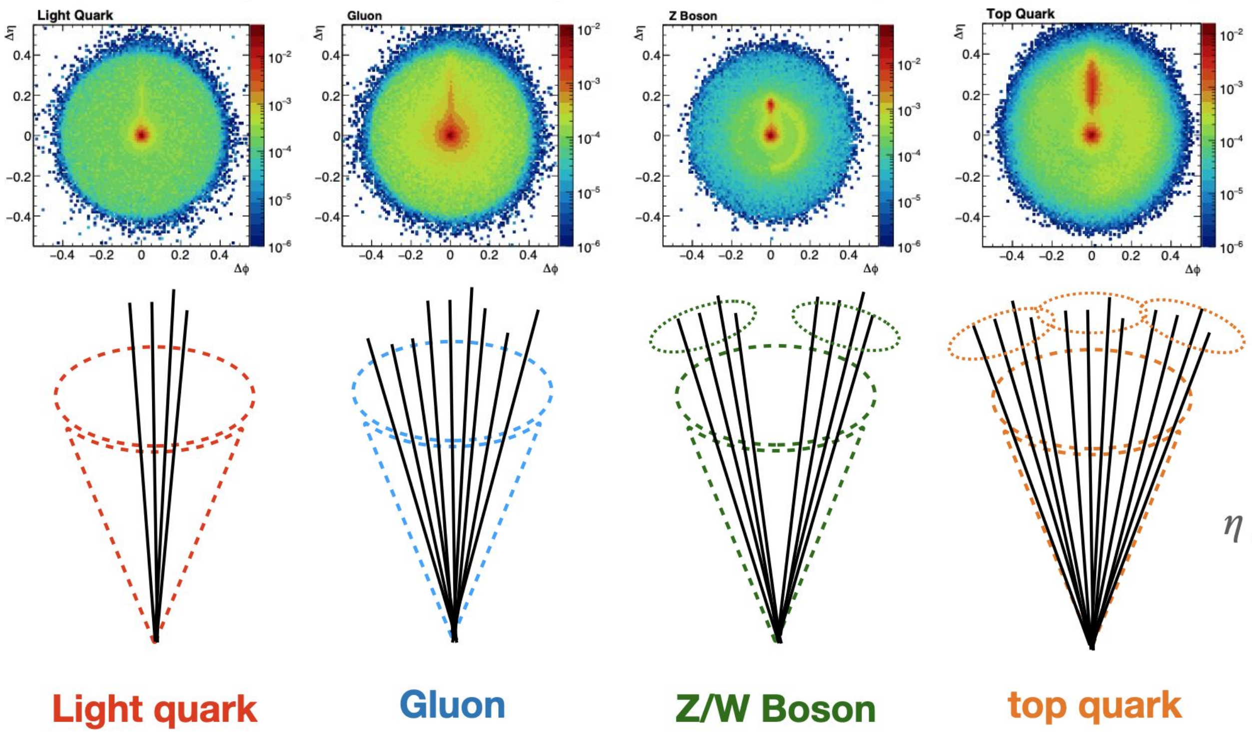
Figure from arXiv:1502.03044

Attention was introduced in the paper: [Attention is all you need]()

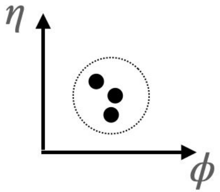
Attention for graphs introduced in paper [Graph Attention Networks]()

Transformers are the state-of-the-art for **NLP**, **computer vision** and also **graphs**!

PCT: Results



- For jet tagging, we can investigate what the **transformer considers important**
- **Top images:** Where is the particle with **highest attention** w.r.t the most energetic particle in the jet?



PCT: Results

- At the time of publication, leading performance in **light quark vs gluon task**

	Acc	AUC	$1/\epsilon_B$ ($\epsilon_S = 0.5$)	$1/\epsilon_B$ ($\epsilon_S = 0.3$)
ResNeXt-50 [17]	0.821	0.9060	30.9	80.8
P-CNN [17]	0.827	0.9002	34.7	91.0
PFN [33]	-	0.9005	34.7±0.4	-
ParticleNet-Lite [17]	0.835	0.9079	37.1	94.5
ParticleNet [17]	0.840	0.9116	39.8±0.2	98.6±1.3
ABCNet [18]	0.840	0.9126	42.6±0.4	**118.4±1.5**
SPCT	0.815	0.8910	31.6±0.3	93.0±1.2
PCT	**0.841**	**0.9140**	**43.2±0.7**	118.0±2.2

PCT: Results

- Similarly good when trained using a different dataset containing **different particles**

Algorithm	Gluon	Light quark	W boson	Z boson	Top quark
		AUC			
DNN	0.9384	0.9026	0.9537	0.9459	0.9620
GRU	0.9040	0.8962	0.9192	0.9042	0.9350
CNN	0.8945	0.9007	0.9102	0.8994	0.9494
JEDI-net	0.9529	0.9301	0.9739	0.9679	0.9683
JEDI-net with $\sum O$	0.9528	0.9290	0.9695	0.9649	0.9677
SPCT	0.9537	0.9326	0.9740	0.9779	0.9693
PCT	**0.9623**	**0.9414**	**0.9789**	**0.9814**	**0.9757**

Different graph architecture