Can you hear the shape of a jet?

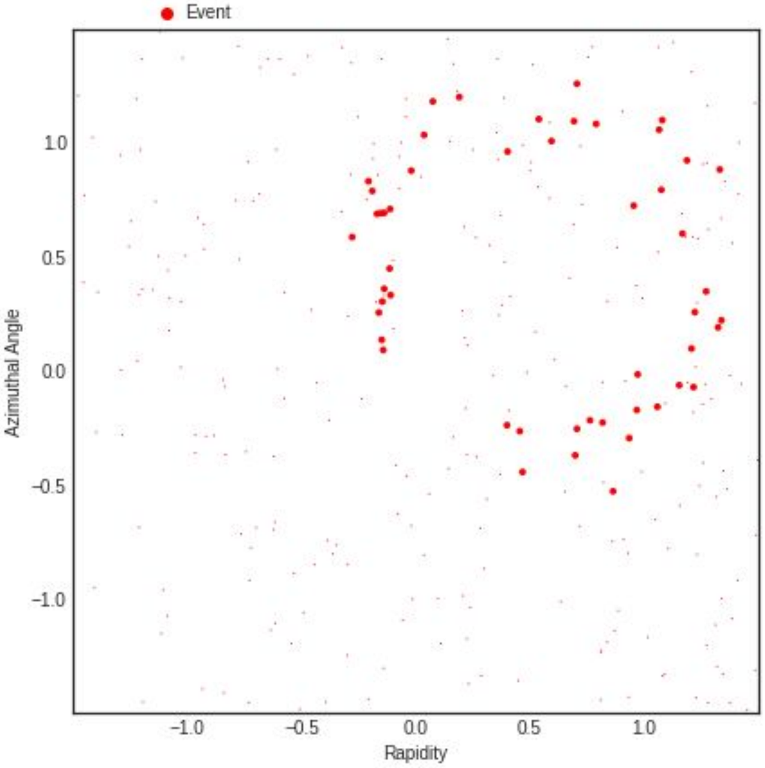
Rikab Gambhir

With Akshunna S. Dogra ( ),

Demba Ba ( ),

& Jesse Thaler ( )

Fundamental Question: What shape is this?

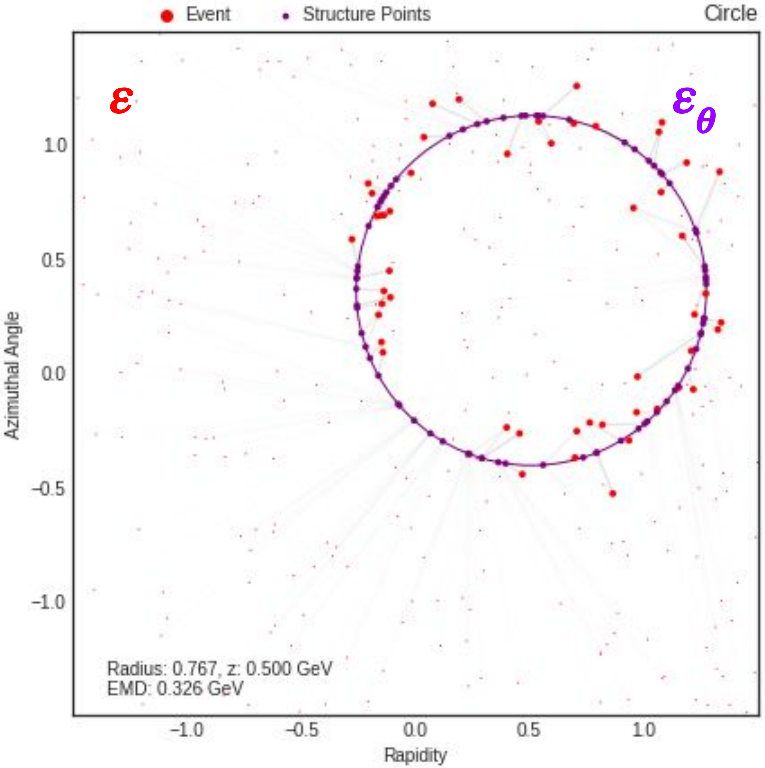


Pictured: (Fake) event that you might have measured at the LHC

Red dots are detector hits on a patch of the LHC cylinder, weighted by energy

Goal: Construct an observable $\mathcal{O}$ that generically answers this question!

Fundamental Question: What shape is this?



Using the **SHAPER** framework …

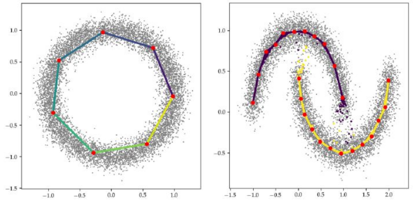
$$\mathcal{O}_{\mathcal{M}}(\mathcal{E}) = \min_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

$$\theta = \operatorname*{argmin}_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

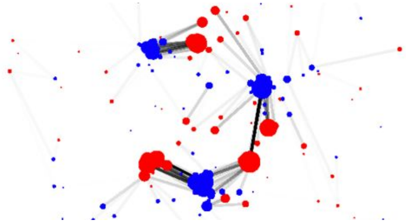
Circle with radius 0.767, center (0.50, 0.36) and a "circle-ness" value of 0.32.

Yes, you *CAN* hear the shape of a jet!

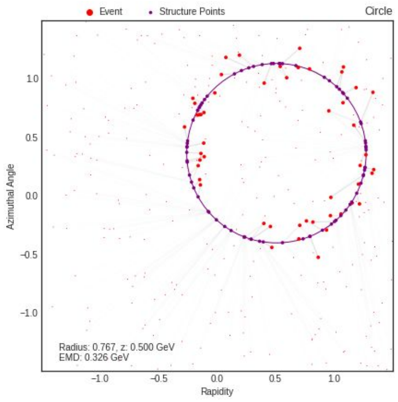
NSF AI Institute for Artificial Intelligence & Fundamental Interactions



Piecewise-Linear Manifold Approximation with K-Deep Simplices (KDS, 2012.02134)



Well-Defined Metric on Particle Collisions using Energy Mover's Distance (EMD, 2004.04159)



SHAPER: Learning the Shape of Collider Events

$$\mathcal{O}_{\mathcal{M}}(\mathcal{E}) = \min_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

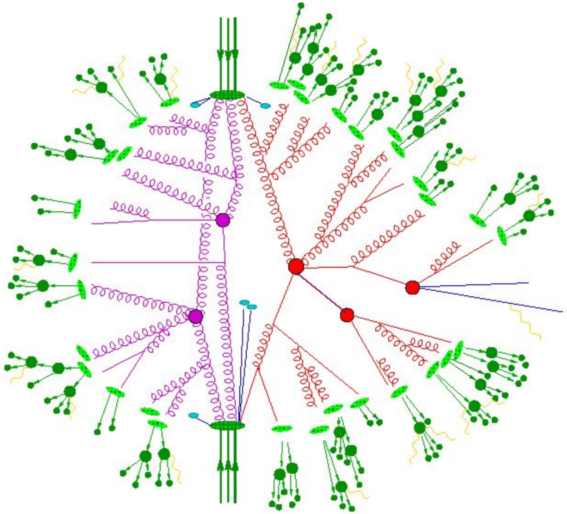
$$\theta = \operatorname*{argmin}_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

Framework for defining and calculating useful observables for collider physics!

Energy Flows

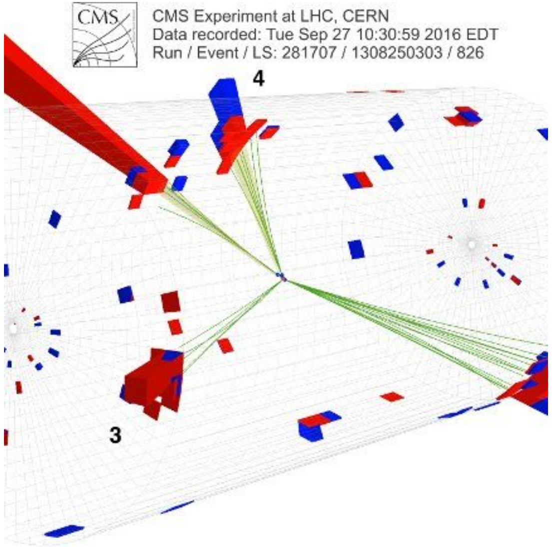
Robust Observables

[Enrico Bothmann et. al., 1905.09127;
CMS, 1810.10069]



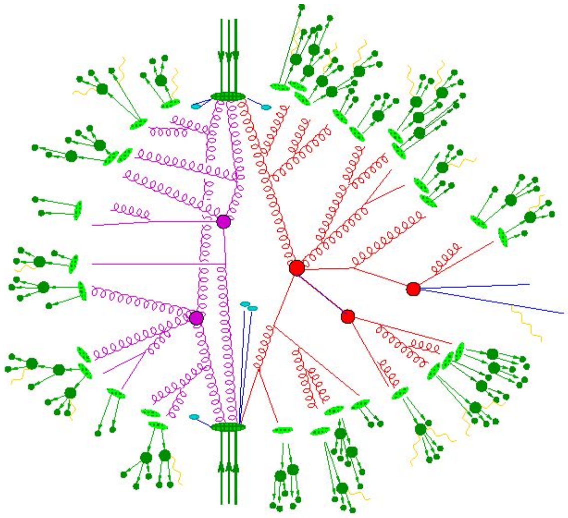
Theory

Experiment



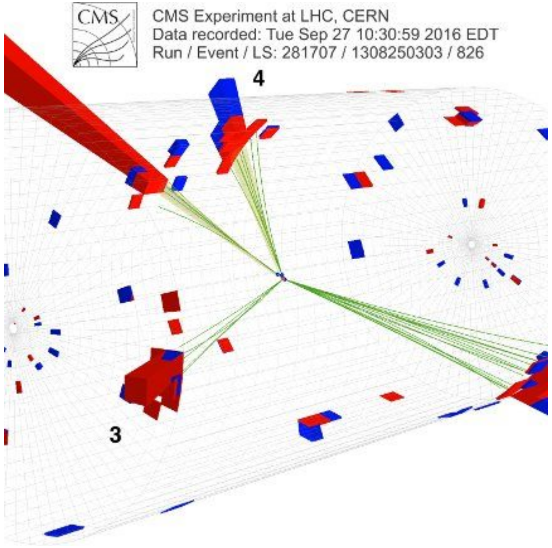
Robust Observables

[Enrico Bothmann et. al., 1905.09127;
CMS, 1810.10069]



Theory

Experiment

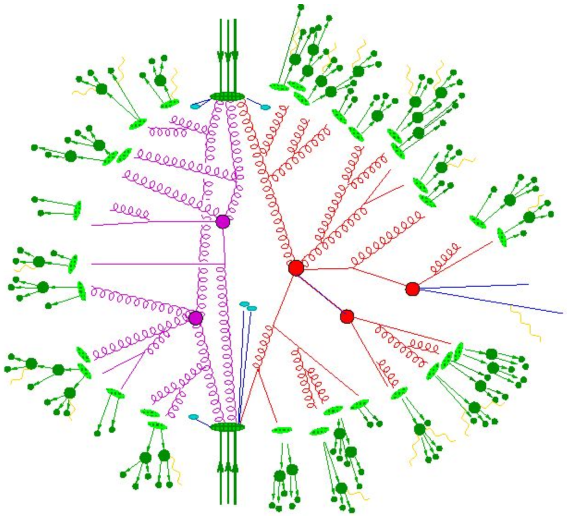


Perturbativity?
Hadronization?
Parton Shower Model?

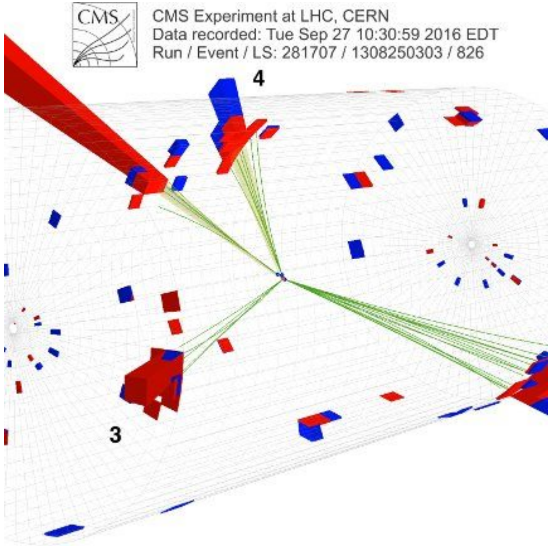
Finite Calorimeter Resolution Effects?
Different Resolution between Detectors?

Robust Observables

[Enrico Bothmann et. al., 1905.09127; CMS, 1810.10069]



Theory | Experiment



Perturbativity?
Hadronization?
Parton Shower Model?

Energy Flow!

Finite Calorimeter Resolution Effects?
Different Resolution between Detectors?

The Energy Flow

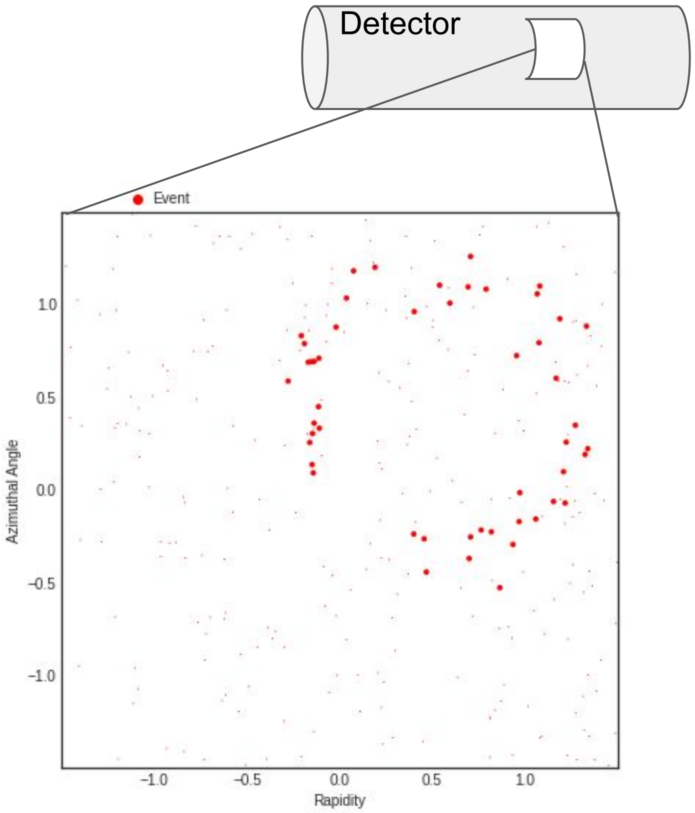
The Infrared and Collinear safe information about a state is contained within its **Energy Flow**:

$$\mathcal{E}(\vec{y}) = \sum_i z_i \delta(\vec{y} - \vec{y}_i)$$

Detector Coordinate (η, ϕ)

Energy Fraction E_j / E_{Tot}

Can be either **real** or **idealized**.

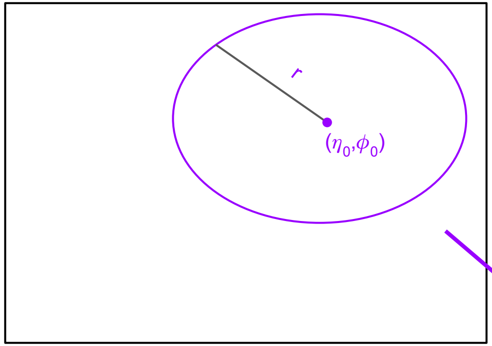


This plot *is* the energy flow for an event

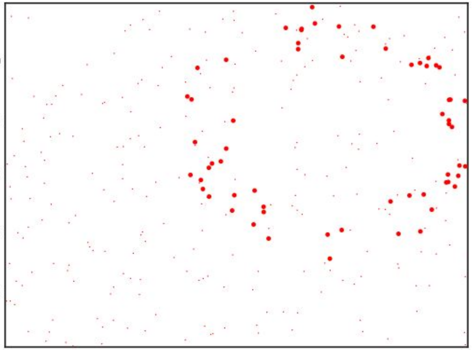
Shapes and the Wasserstein Metric

Shapes and the Energy Flows

Translate our question about shapes into energy flows!



≆



Idealized Energy Flow **Real Energy Flow**

What parameterized energy flow, $\boldsymbol{\varepsilon_\theta}$, *best* matches the observed energy flow $\boldsymbol{\varepsilon}$?

Shape Observables

Minimize over the manifold of parameterized energy flows to determine shape!

$$\mathcal{O}_{\mathcal{M}}(\mathcal{E}) = \min_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

$$\theta = \operatorname*{argmin}_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

Learns the "shapiness" $\mathcal{O}$ and the optimal shape parameters θ

Observables ⟺ Manifold of Parameterized Flows correspondence!

[P. Komiske, E. Metodiev, J. Thaler, 1902.02346;
see also T. Cai, J. Cheng, K. Craig, N. Craig, 2111.03670;
see also C. Zhang, Y. Cai, G. Lin, C. Shen, 2003.06777;
see also L. Hou, C. Yu, D. Samaras, 1611.05916;
see also M. Arjovsky, S. Chintala, L. Bottou, 1701.07875]

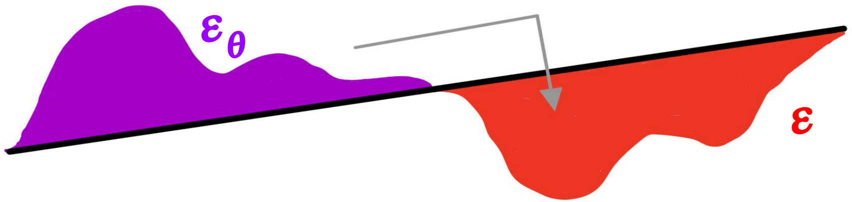
The Wasserstein Metric

There is a natural metric on probability distributions, the **Wasserstein Metric**

$$\mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta) = \sum_{i,j} f_{ij} \frac{|\vec{y}_i - \vec{y}'_j|}{R} + |E_i - E'_j|$$

$$\text{where} \sum_j f_{ij} = E_i, \sum_i f_{ij} = E'_j, \sum_{i,j} f_{ij} = \min(E_i, E'_j)$$

Also known as the Earth/Energy Mover's Distance (EMD) - geometric structure on events!



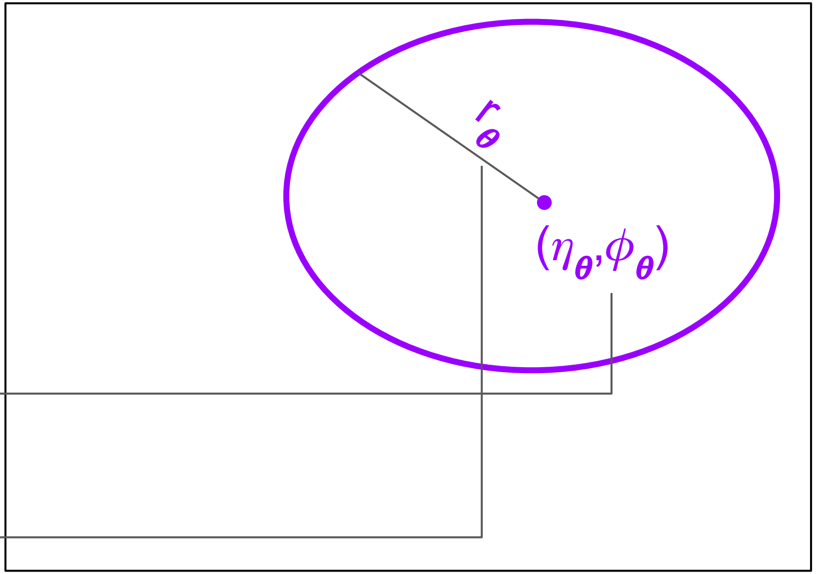
EMD = Work done to move "dirt" optimally

Defining Shapes

Define a *shape* as any parameterized energy flow

For example, for a circle with parameterized radius and center*

$$\mathcal{E}(\vec{y}) = \begin{cases} \frac{1}{2\pi r_\theta} & |\vec{y} - \vec{y}_\theta| = r_\theta \\ 0 & |\vec{y} - \vec{y}_\theta| \neq r_\theta \end{cases}$$



Our parameterized circle written as an energy flow

*Uniform prior by choice for simplicity. In principle, we can pick any parameterized normalized distribution.

Observable ⇔ Manifolds

[P. Komiske, E. Metodiev, and J. Thaler, 2004.04159;
J. Thaler, and K. Van Tilburg, 1011.2268;
I. W. Stewart, F. J. Tackmann, and W. J. Waalewijn, 1004.2489.;
S. Brandt, C. Peyrou, R. Sosnowski and A. Wroblewski, PRL 12 (1964) 57-61;
C. Cesarotti, and J. Thaler, 2004.06125]

Many existing observables have this form!

- *N*-subjettines ⇔ Manifold of *N*-point events
- *N*-jettiness ⇔ Manifold of *N*-point events with floating energy
- Thrust ⇔ Manifold of back-to-back point events
- Event Isotropy ⇔ Uniform distribution
- … and more!

All of the form "How much like **[shape]** does my **event** look like?"

We generalize this to build more observables!

The SHAPER Framework

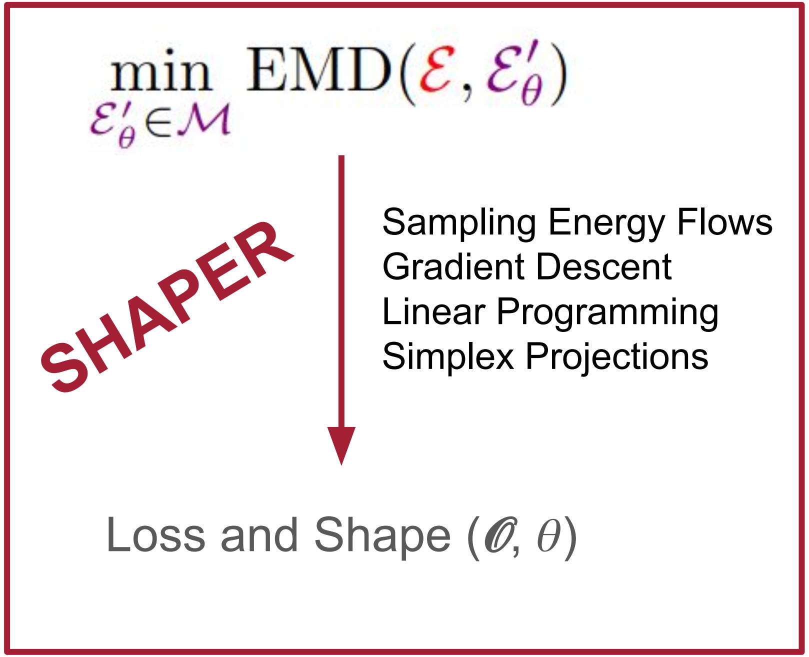
SHAPER

Shape-**H**unting **A**lgorithm using **P**arameterized **E**nergy **R**econstruction

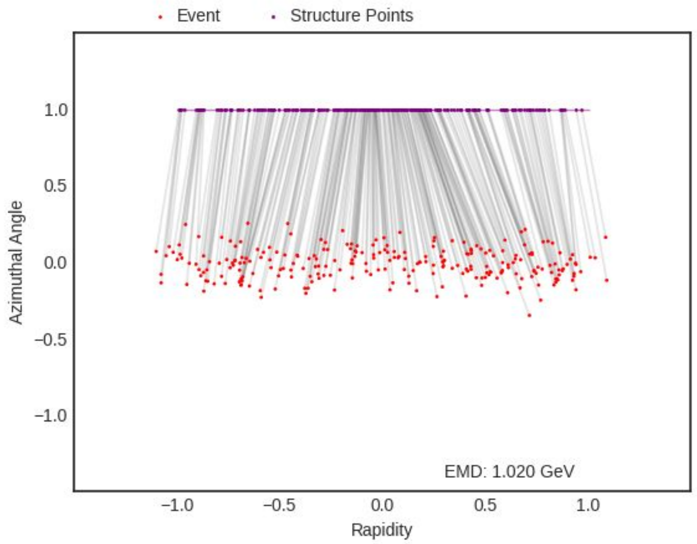
Framework for defining and building IRC-safe observables using parameterized objects

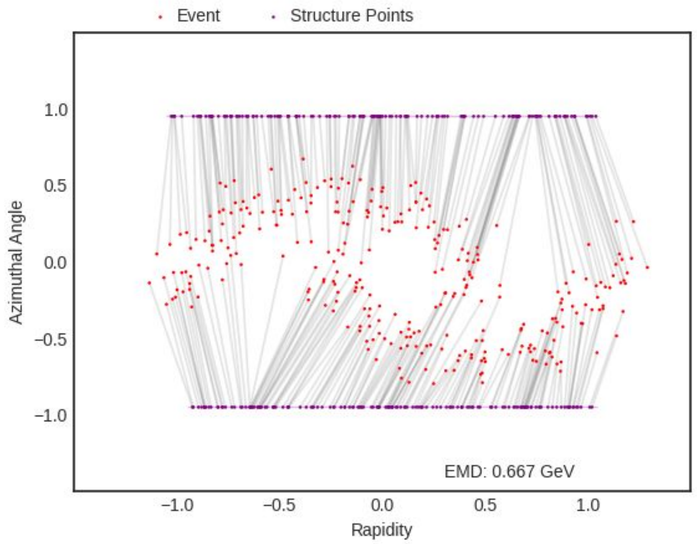
Easy to define new observables by specifying parameterization, or by combining shapes

Returns EMD distance and optimal shape parameters



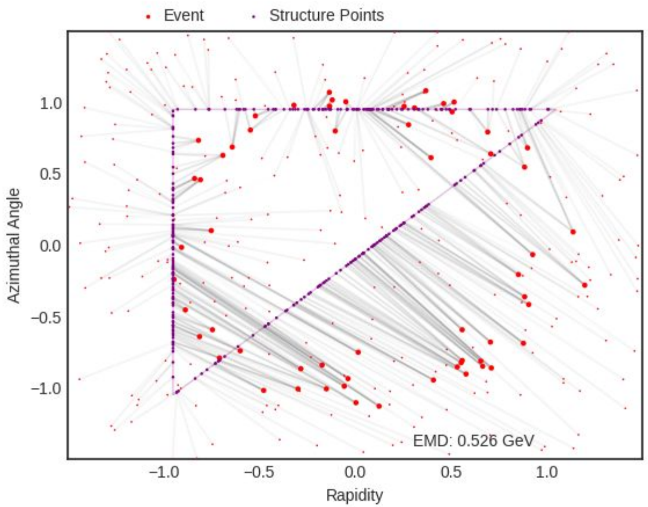
Fun Animations

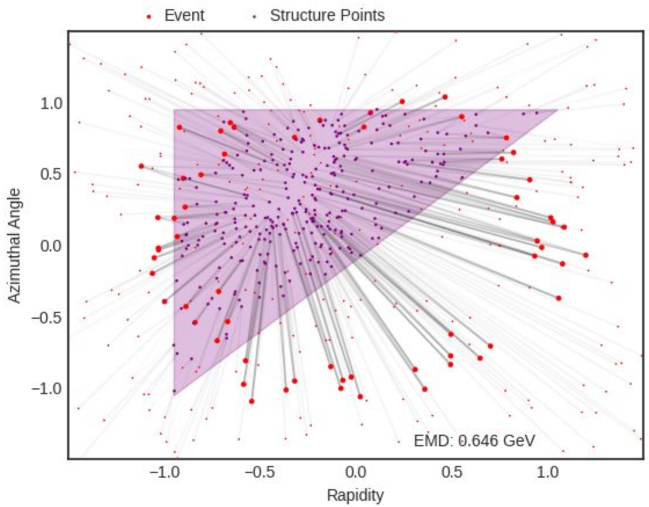




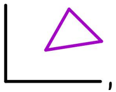
Red: Event Y
Purple: Shape ε_θ with structure points a_i
Grey: Matrix x_{ij} connecting y's and a_i's

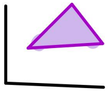
Fun Animations Cont'd





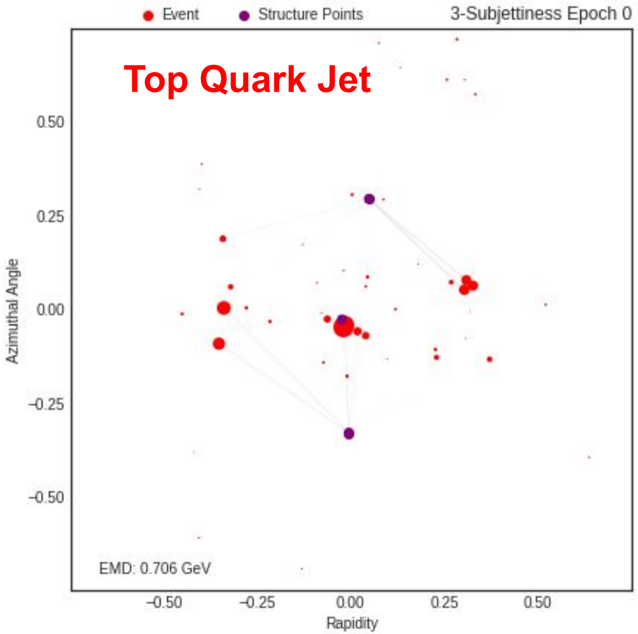
Red: Event Y
Purple: Shape $\mathcal{E}_\theta$ with structure points a_i
Grey: Matrix f_{ij} connecting y's and a_i's

Left: $\mathcal{E}_\theta$ =  , EMD = 0.245

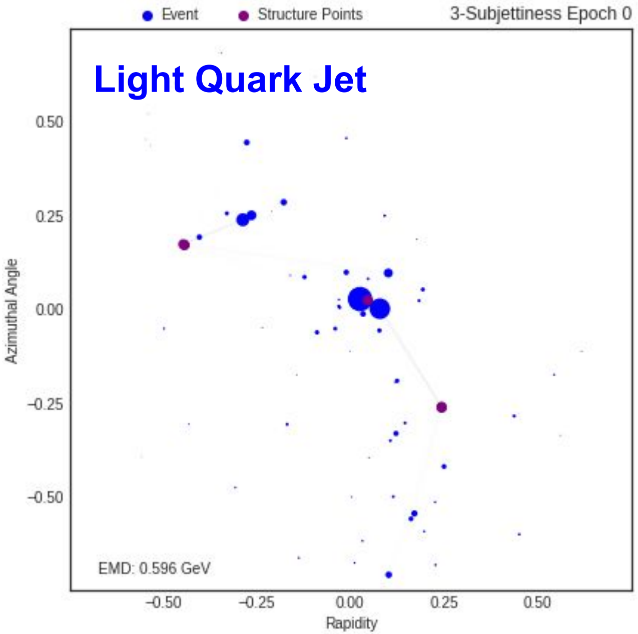
Right: $\mathcal{E}_\theta$ =  EMD = 0.279

N-Subjettiness

Easy to compute classic jet observables!



τ_3 = 0.214



τ_3 = 0.343

[see also L., B. Nachman, A. Schwartzman, C. Stansbury, 1509.02216; see also B. Nachman, P. Nef, A. Schwartzman, M. Swiatlowski, C. Wanotayaroj, 1407.2922; see also M. Cacciari, G. Salam, 0707.1378]

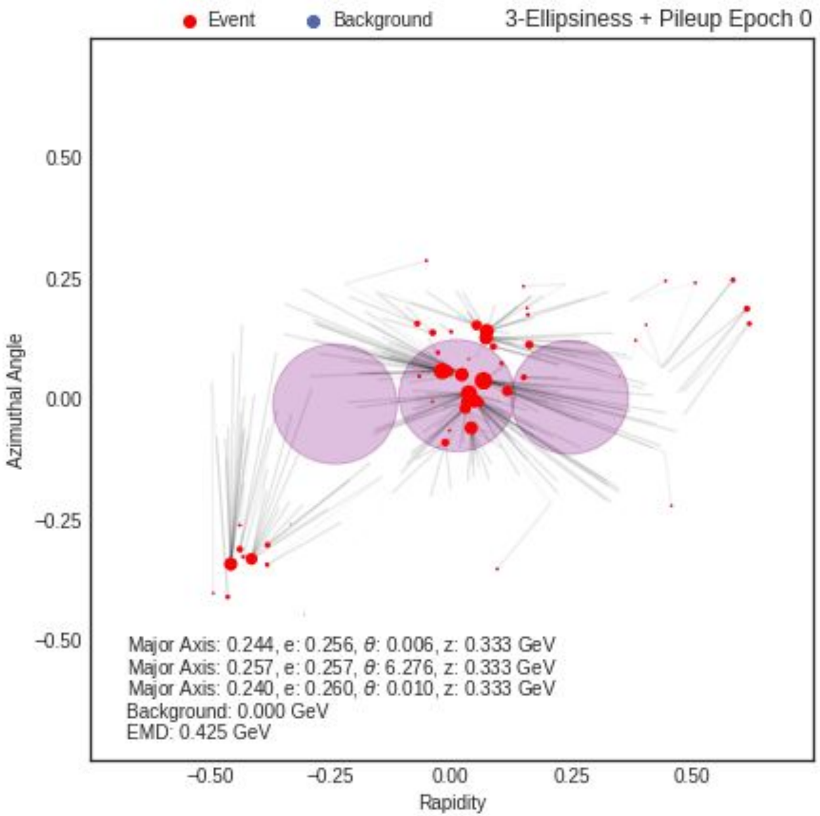
New IRC-Safe Observables

The **SHAPER** framework makes it easy to invent new jet observables!

***N*-Ellipsiness+Pileup** as a jet algorithm.

- Learn jet centers
- Dynamic jet radii (no *R* hyperparameter)
- Dynamic eccentricities and angles
- Dynamic jet energies
- Uniform Pileup Subtraction
- Learned parameters for discrimination

Can design custom specialized jet algorithms to learn jet substructure!



New IRC-Safe Observables

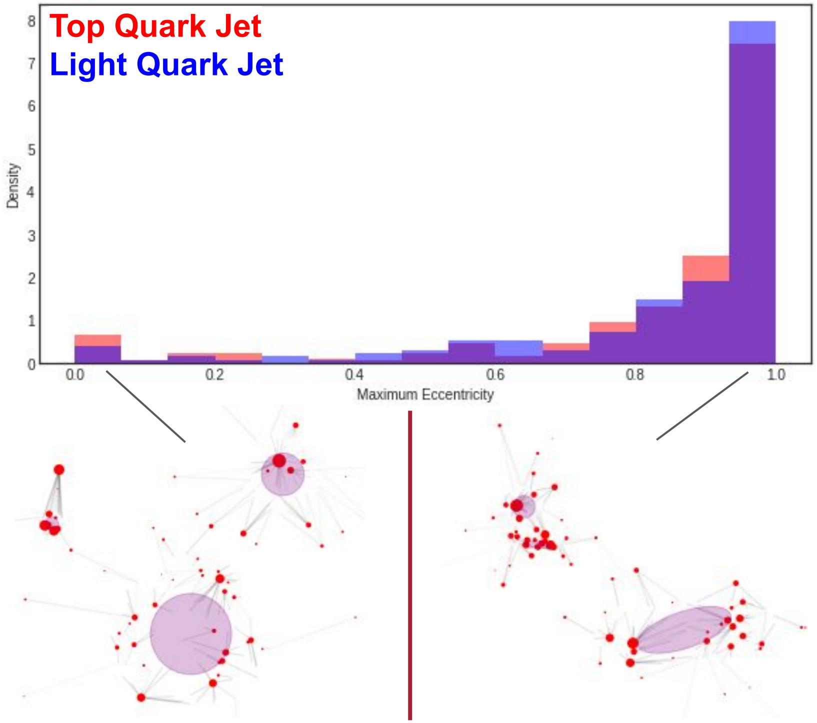
The **SHAPER** framework makes it easy to invent new jet observables!

***N*-Ellipsiness+Pileup** as a jet algorithm.

- Learn jet centers
- Dynamic jet radii (no *R* hyperparameter)
- Dynamic **eccentricities** and angles
- Dynamic jet energies
- Uniform Pileup Subtraction
- Learned parameters for discrimination

Can design custom specialized jet algorithms to learn jet substructure!

Max Eccentricity



Low Max Eccentricity (.001)   High Max Eccentricity (.972)

Outlook

SHAPER, a machine learning framework for calculating robust observables for collider physics based on IRC-safety and Wasserstein geometry!

Playground for defining and building custom observables and jet algorithms!

$$\mathcal{O}_{\mathcal{M}}(\mathcal{E}) = \min_{\mathcal{E}'_\theta \in \mathcal{M}} \text{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

$$\theta = \underset{\mathcal{E}'_\theta \in \mathcal{M}}{\text{argmin}}\, \text{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

Outlook

SHAPER, a machine learning framework for calculating robust observables for collider physics based on IRC-safety and Wasserstein geometry!

Playground for defining and building custom observables and jet algorithms!

$$\mathcal{O}_{\mathcal{M}}(\mathcal{E}) = \min_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

$$\theta = \operatorname*{argmin}_{\mathcal{E}'_\theta \in \mathcal{M}} \mathrm{EMD}(\mathcal{E}, \mathcal{E}'_\theta)$$

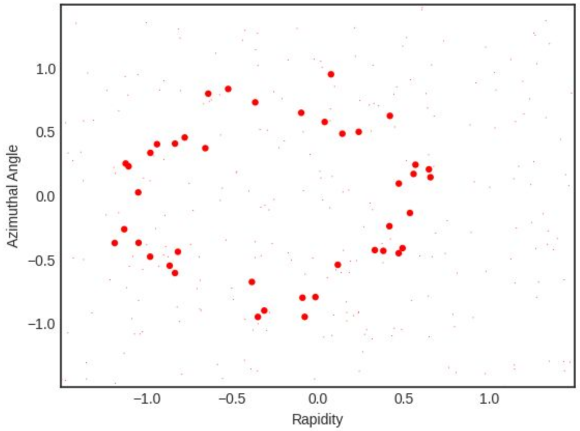
Using **SHAPER**, you ***CAN*** hear the shape of a jet!

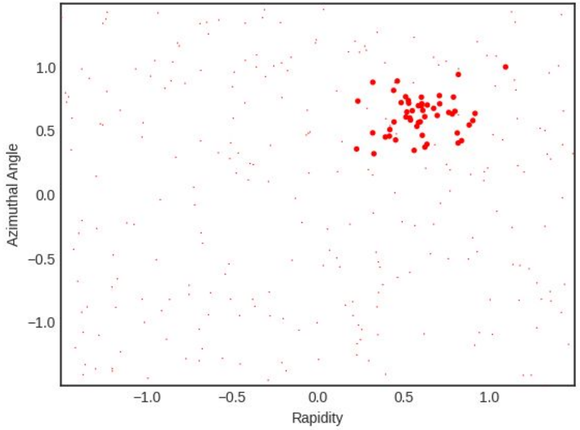
Appendices

Toy Analysis

Define the following two types of events:

- Type 1: Ring-Like
  - Pois(50) “signal” particles, uniformly making up 70%-90% of event energy
  - Arranged in a ring with radius *r ~ N(0.75, 0.25)*, width = 0.1
  - Pois(250) “background” particles making up remaining energy
- Type 2: Disc-Like
  - Pois(50) “signal” particles, uniformly making up 70%-90% of event energy
  - Arranged in a disc with radius *r ~ N(0.50, 0.25)*
  - Pois(250) “background” particles making up remaining energy





Observables

In analogy with *N-jettiness*, define the *A-shapeliness* of an event as the value of the loss when evaluated on the shape *A*.

For this toy analysis, define:

- Shape A_1: Filled-in triangle, parameterized by its vertices
- Shape A_2: Boundary of a triangle, parameterized by its vertices (∂A1)

Both shapes have *R = 0.25*

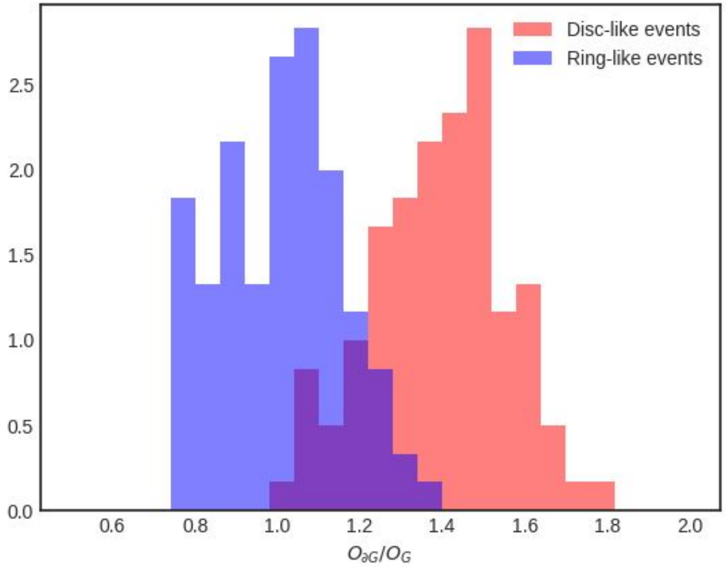
The ratio of the shapeliness values should be a proxy for if the event is ring-like or disc-like

Technical Aspects

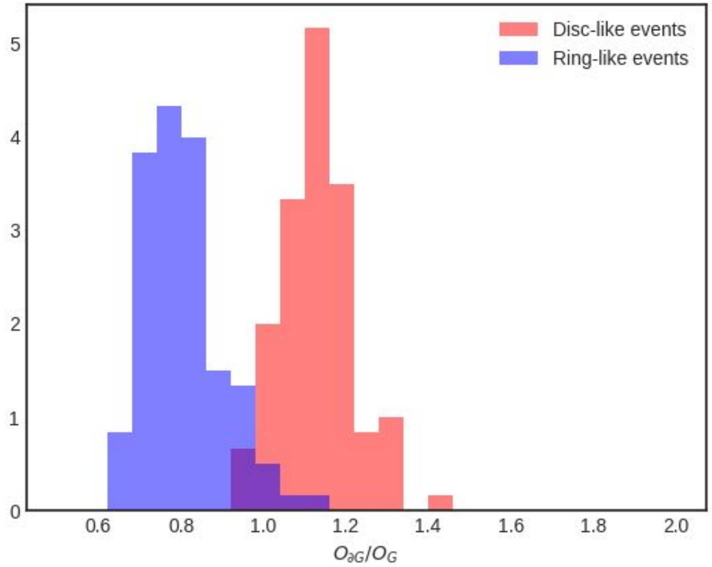
- 125 sample points defining shapes
- Adam optimizer, lr = 0.05
- 125 epochs with *z0* frozen, then 125 epochs with *z0* unfrozen
  - Early stopping if loss has not improved after 10 epochs
- Triangles initialized at (-1, -1), (-1, 1), (1,1)

With these settings, each observable takes about 2-3 seconds per event.

Results



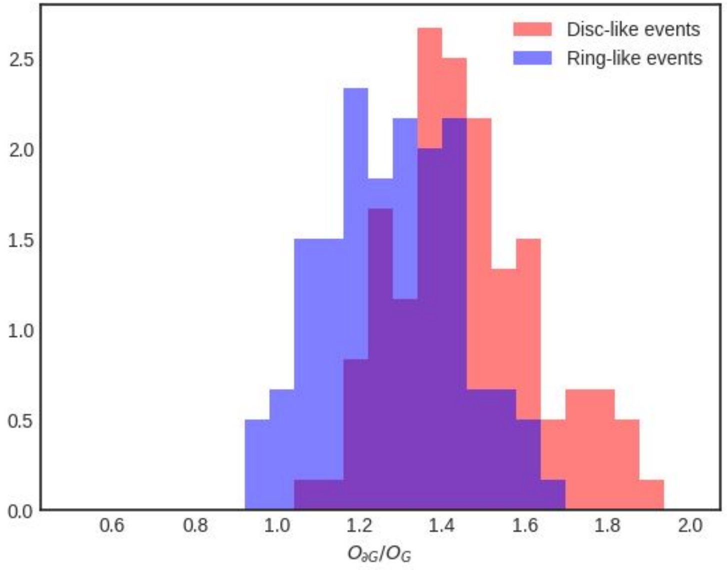
Without unclustered term



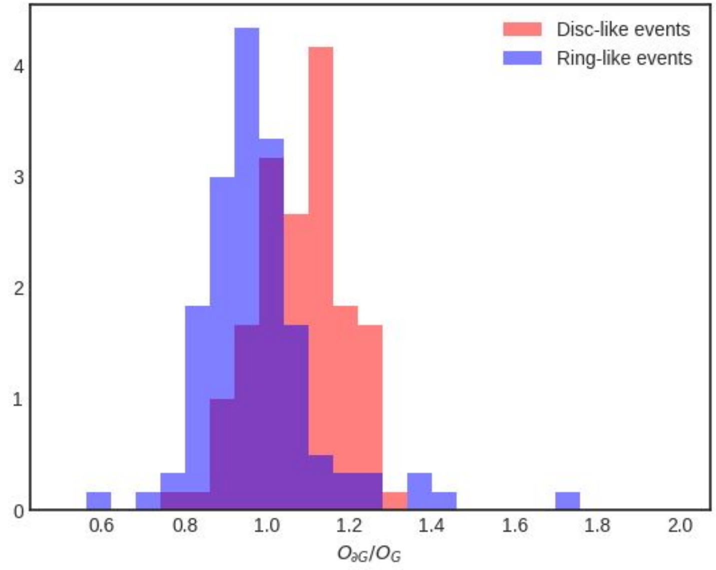
With unclustered term

Rikab Gambhir

Results - High Background (40%-60%)



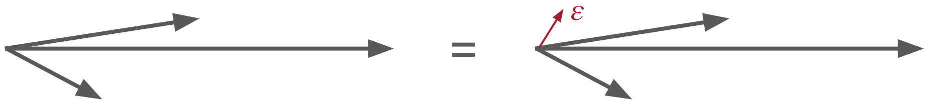
Without unclustered term



With unclustered term

IRC Safety

Infrared Safety: An observable is unchanged under a soft emission



Collinear Safety: An observable is unchanged under a collinear splitting

Implementation

To practically determine this minimum …

1. Initialize *A* by random sampling the energy flow
2. Freeze *A*. Calculate the Wasserstein Metric loss *L*, and the corresponding transport matrix f_{iy}
3. Freeze *f*. Calculate the gradients of *L* with respect to *A* [ignoring dependencies on *f*]
4. Gradient update *A*
5. Freeze *f* and *A*. Gradient update weights *z* by numerical derivative
6. Repeat 2-5 until convergence.