Modeling Hadronization using Machine Learning

AI4EIC

Tony Menzo

PhD candidate, University of Cincinnati

In collaboration with:

Phil Ilten, Stephen Mrenna, Manuel Szewc, Michael Wilkinson, Ahmed Youssef, and Jure Zupan

Based upon work done in 2203.04983

Goals and outlook

The overarching goal is to create a better simulator of collider events.

But also, more ambitiously, to promote a paradigm shift in the modeling of non-perturbative physics.

What has been done?: In **2203.04983** we showed that machine learning techniques can be used to implement a model of hadronization based on (artificial) data

Short term: Implement a machine learning-improved (i.e. data-improved) model of hadronization

Long term: Take what we've learned and develop BETTER theoretical models

Event Generators

1. Hard proccess

2. Parton Showers

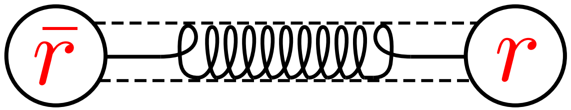
3. Hadronization

4. Unstable particle decay

Early 80s brought many non-perturbative models: Cluster, percolation, …

Lund String Model

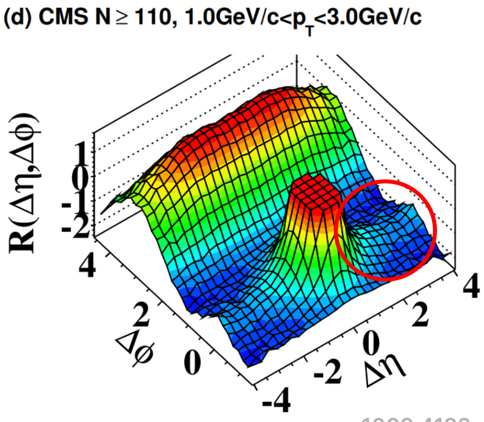
(currently implemented in Pythia)



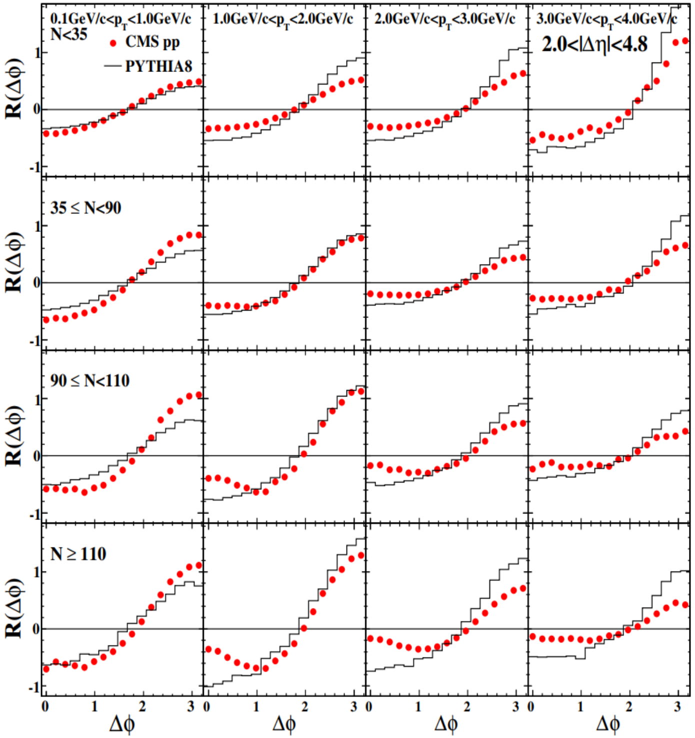
Remarkable agreement with data but some ~new disagreements for **high multiplicty events**...

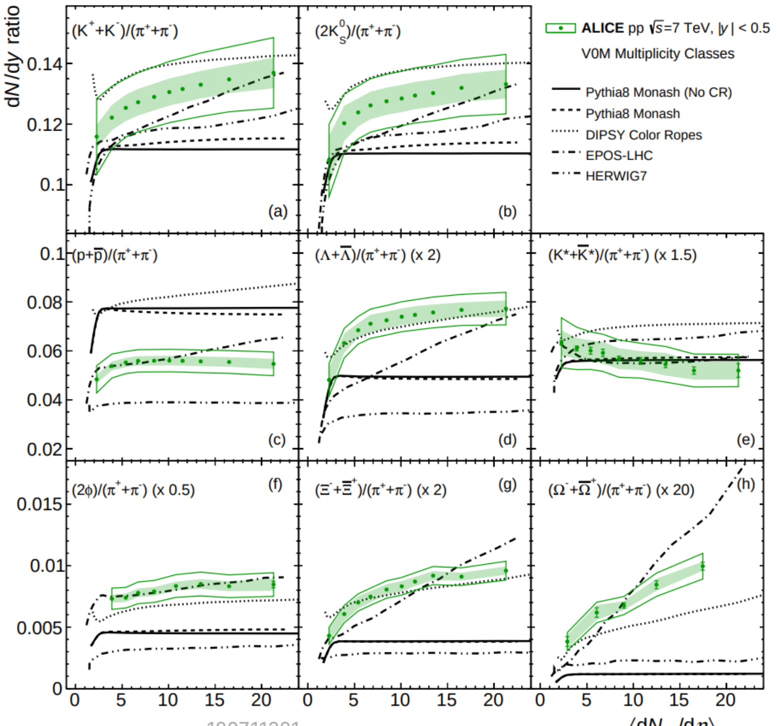
Similar properties to heavy ion collisions:

- "The ridge" i.e. enhanced particle production around the azimuthal angle of a trigger jet (CMS)
- Strangeness production increases as a function of event multiplicity (ALICE)



1009.4122





1807.11321

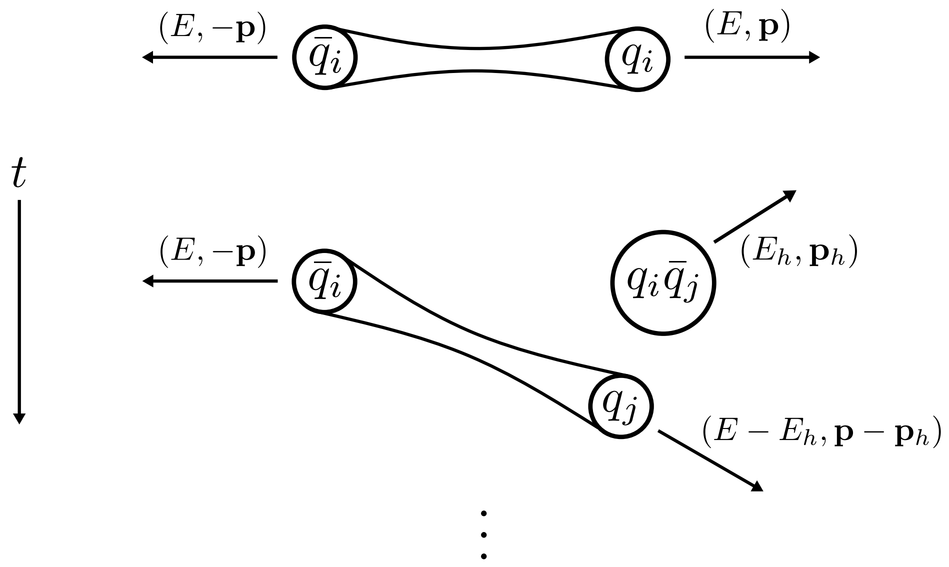
Stringy Hadronization

The momentum fraction z of each fragmenting hadron is sampled according to the

Lund fragmentation function

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_\perp^2}{z}\right)$$

$$z = \frac{p_z + E_h}{2E}$$



How to improve the generator: two* approaches

- Improve model
  - MPIs, rope hadronization, transverse mass supression, flavor asymmetries, hadronic rescattering, multiscale models (string → hydrodynamical), flavor selector, etc.
  - Utilize techniques from gauge-gravity duality

  Hard to come up with mathematically precise model without established calculational techniques

- Data-driven generator
  - Sample directly from global distributions

  Non-universal and extremely difficult to convert into representative particle flow data

*** or a combination of both (our approach)**

Hybrid approach

Hadronization models already do really well!

MODEL

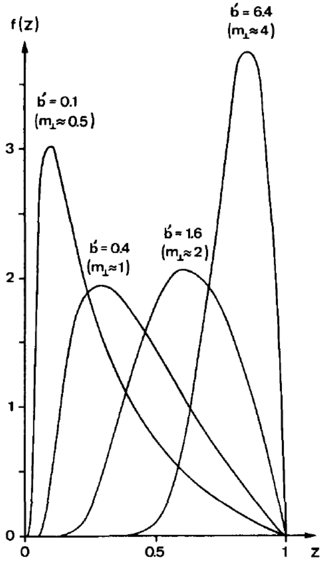
+

EXPERIMENTAL DATA

↓

COMPLETE (OR AT LEAST BETTER) PHENOMENOLOGICAL MODEL OF HADRONIZATION

For example, modify the fragmentation function f(z)...



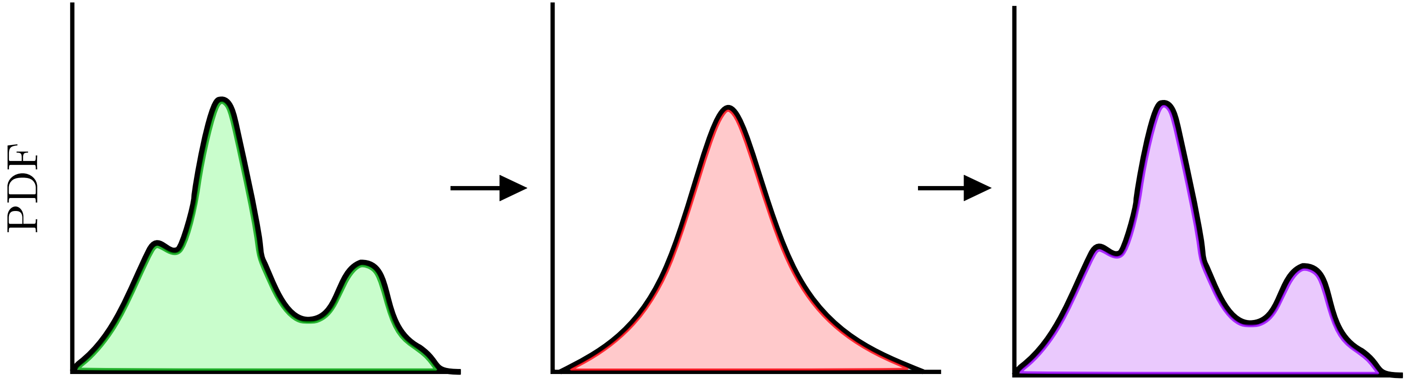
Physics Reports 97.2-3 (1983): 31-145.

$+\ \epsilon$

Why machine learning?

To make any headway we need a tool which will allow us to efficiently sample probability distributions whose analytic form is unknown.

Generative machine learning algorithms are the perfect tool!



Proof of concept (2203.04983)

Consider Pythia output as 'experimental data' and try to reproduce hadronization observables by training on single emission kinematics (~learn the fragmentation function f(z)).

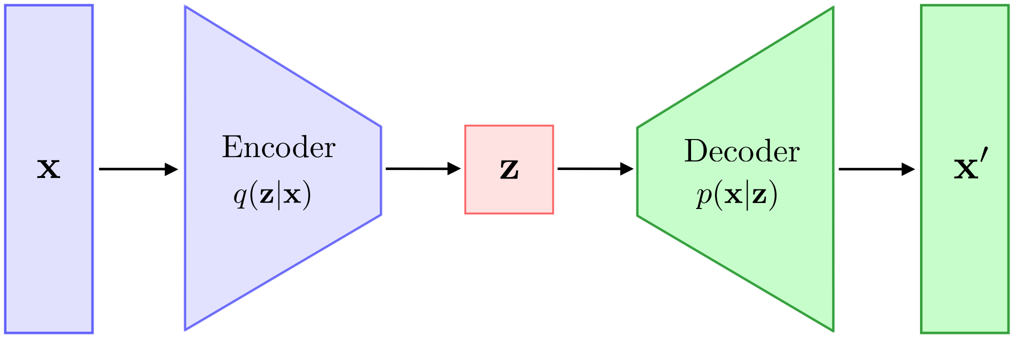
Start from simplest hadronizing system:

1. $q\bar{q} \rightarrow \pi$'s
2. Assume no correlations between emissions
3. E_{cut}~5 GeV (To avoid termination effects)

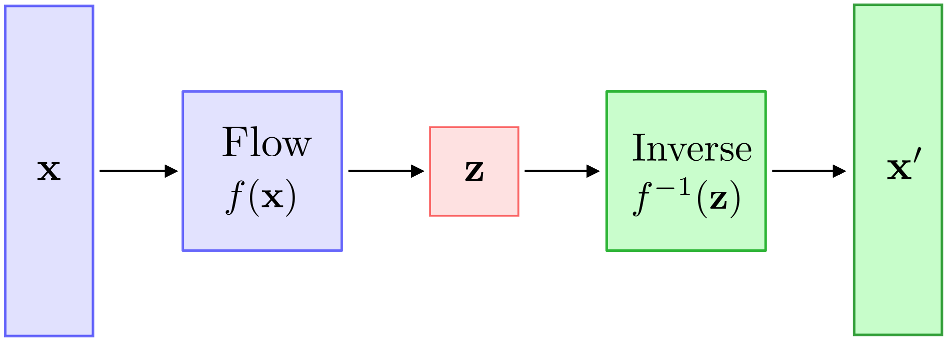
Train on p_z and p_T distributions of 1st emitted π

Architectures

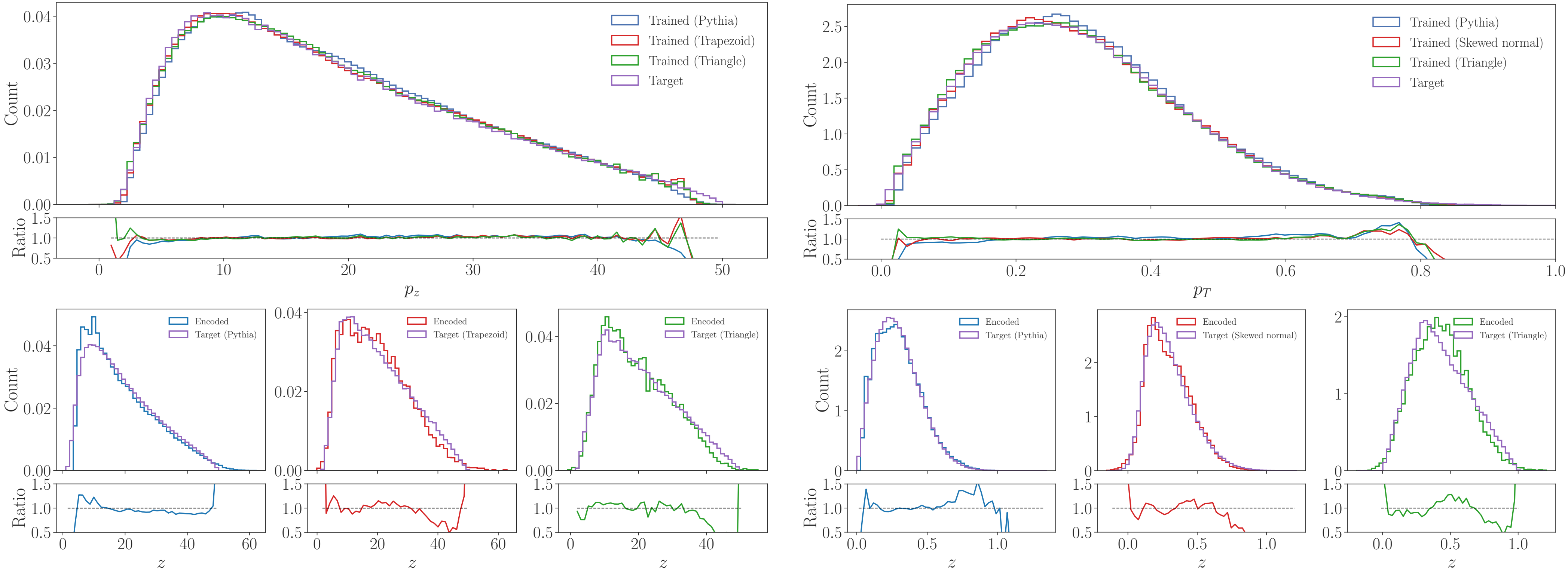
Conditional sliced-Wasserstein Autoencoder (cSWAE)



Conditional normalizing flow (cNF)



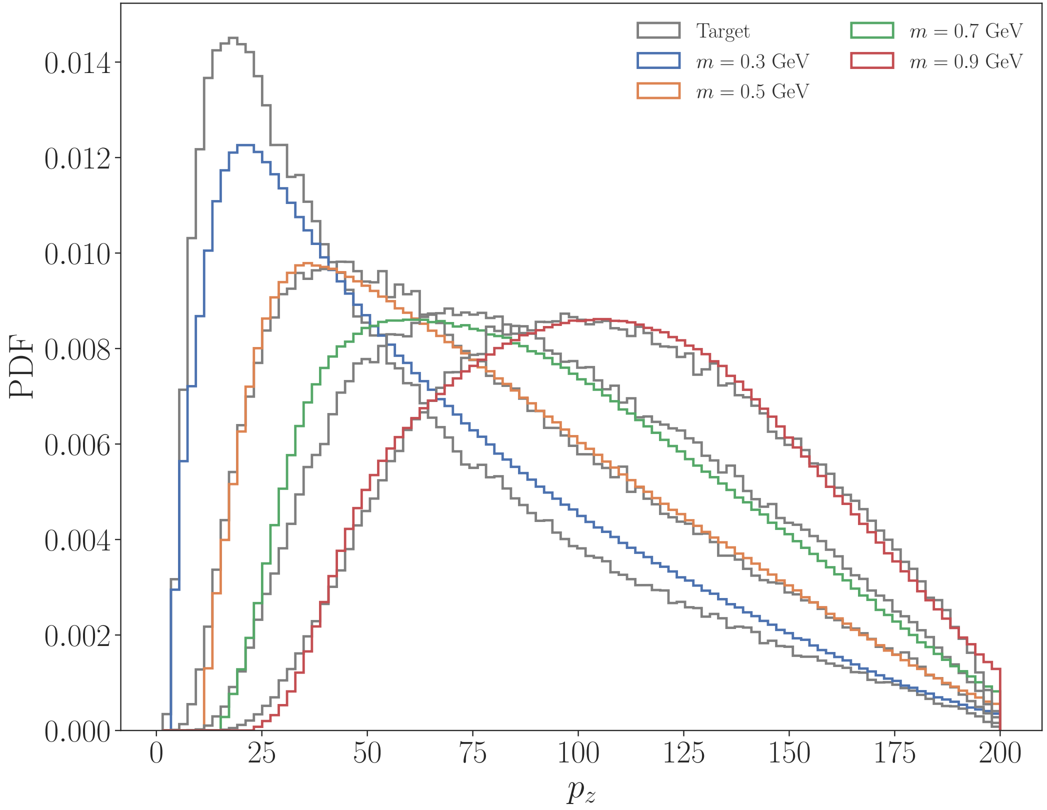
Training Results (cSWAE)

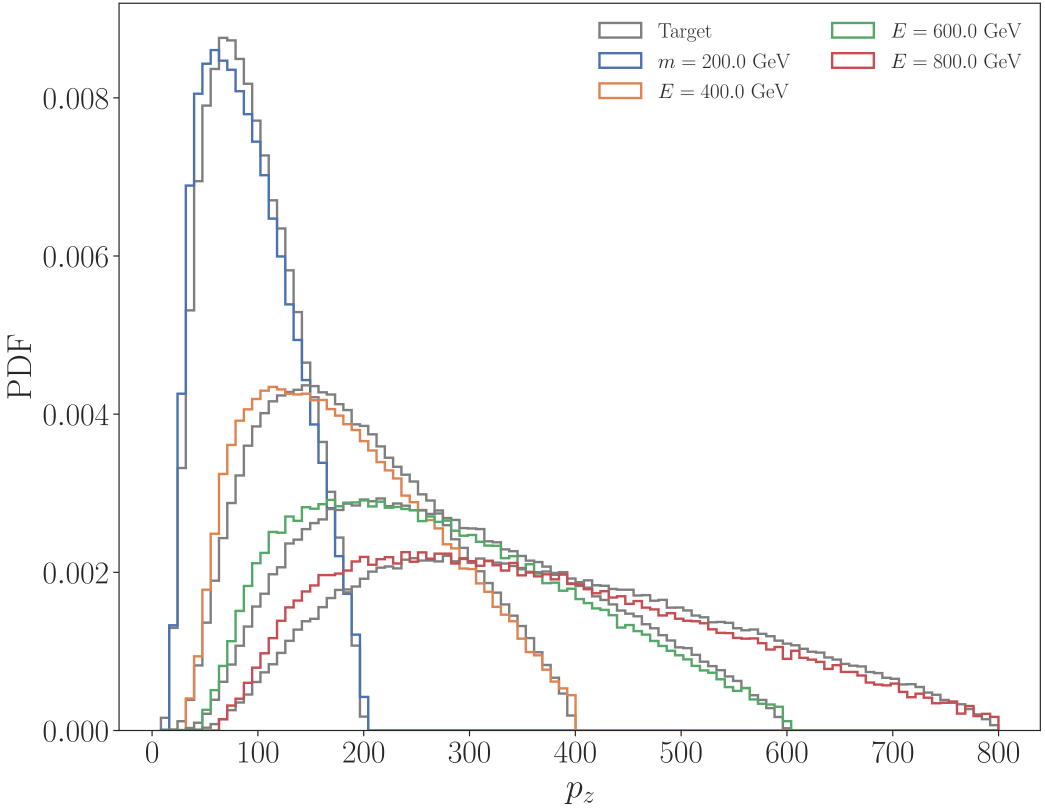


Training Results

(cSWAE with labels and boundaries)

***Preliminary**

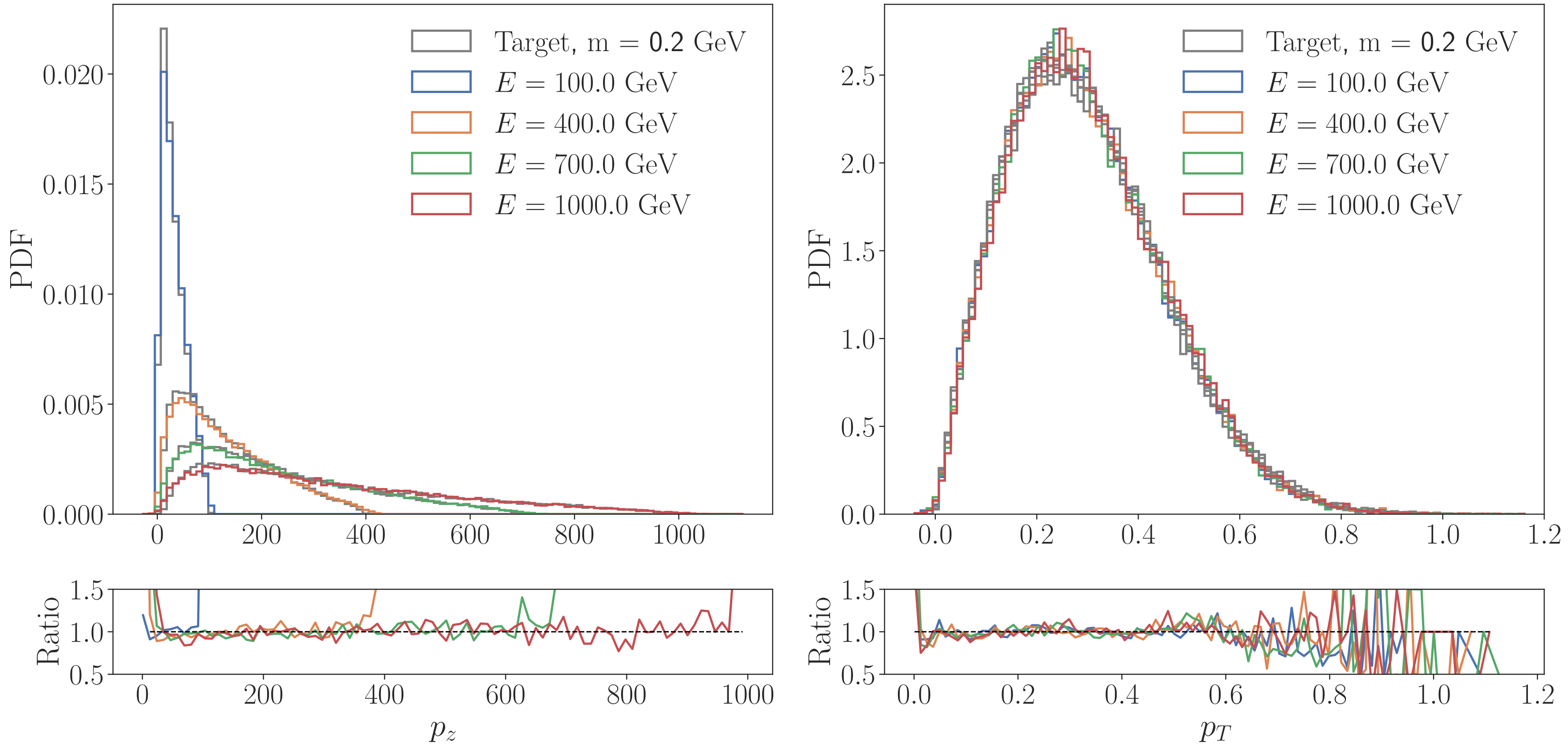




Training Results

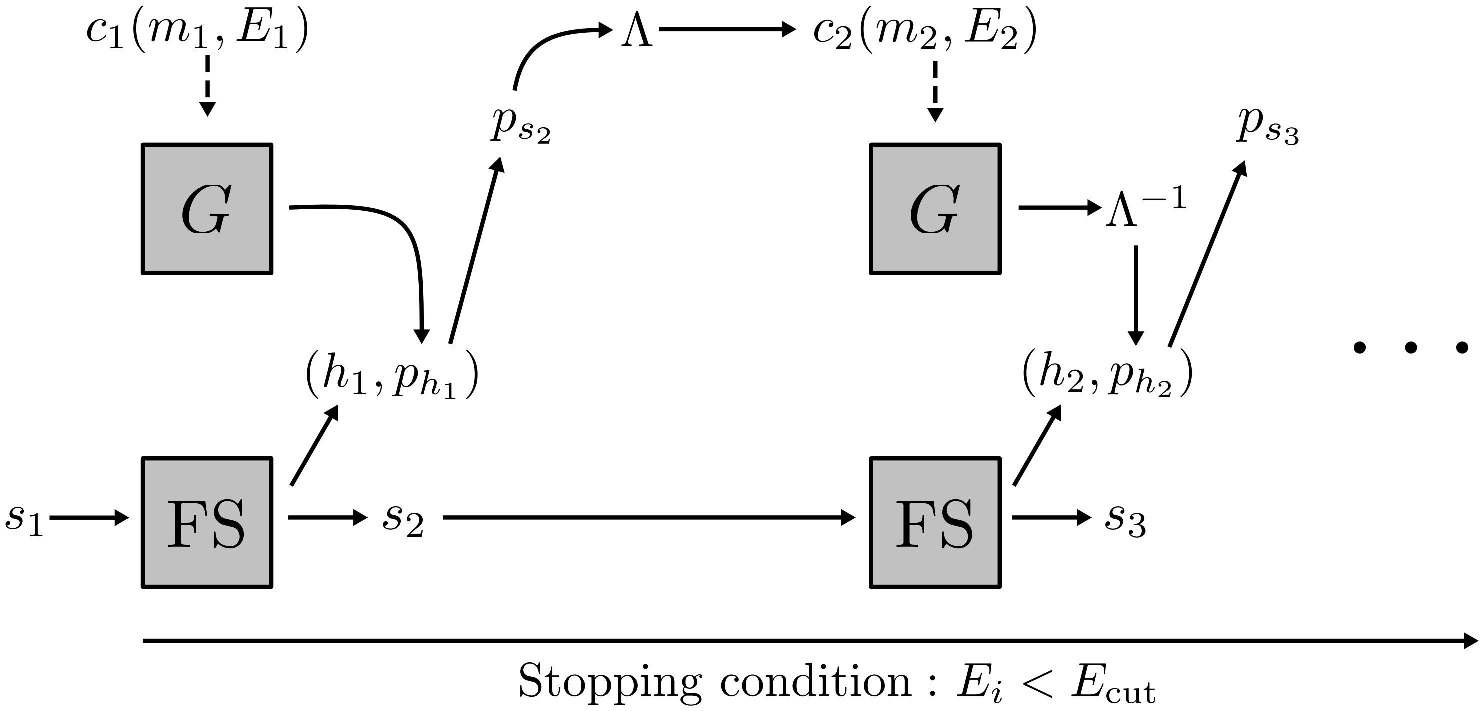
(cNF with labels)

***Preliminary**

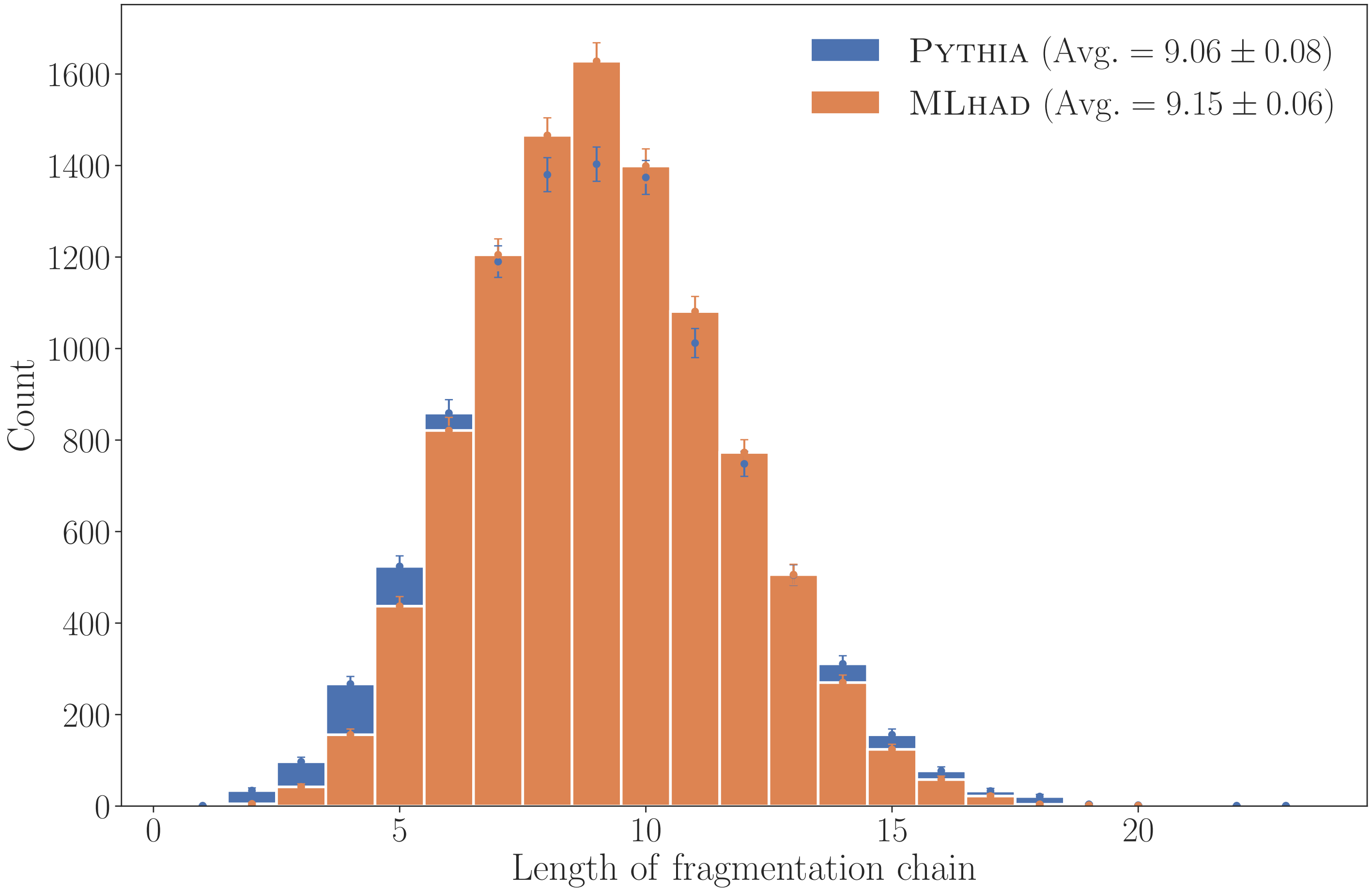


Hadronization (kinematics + flavor selector)

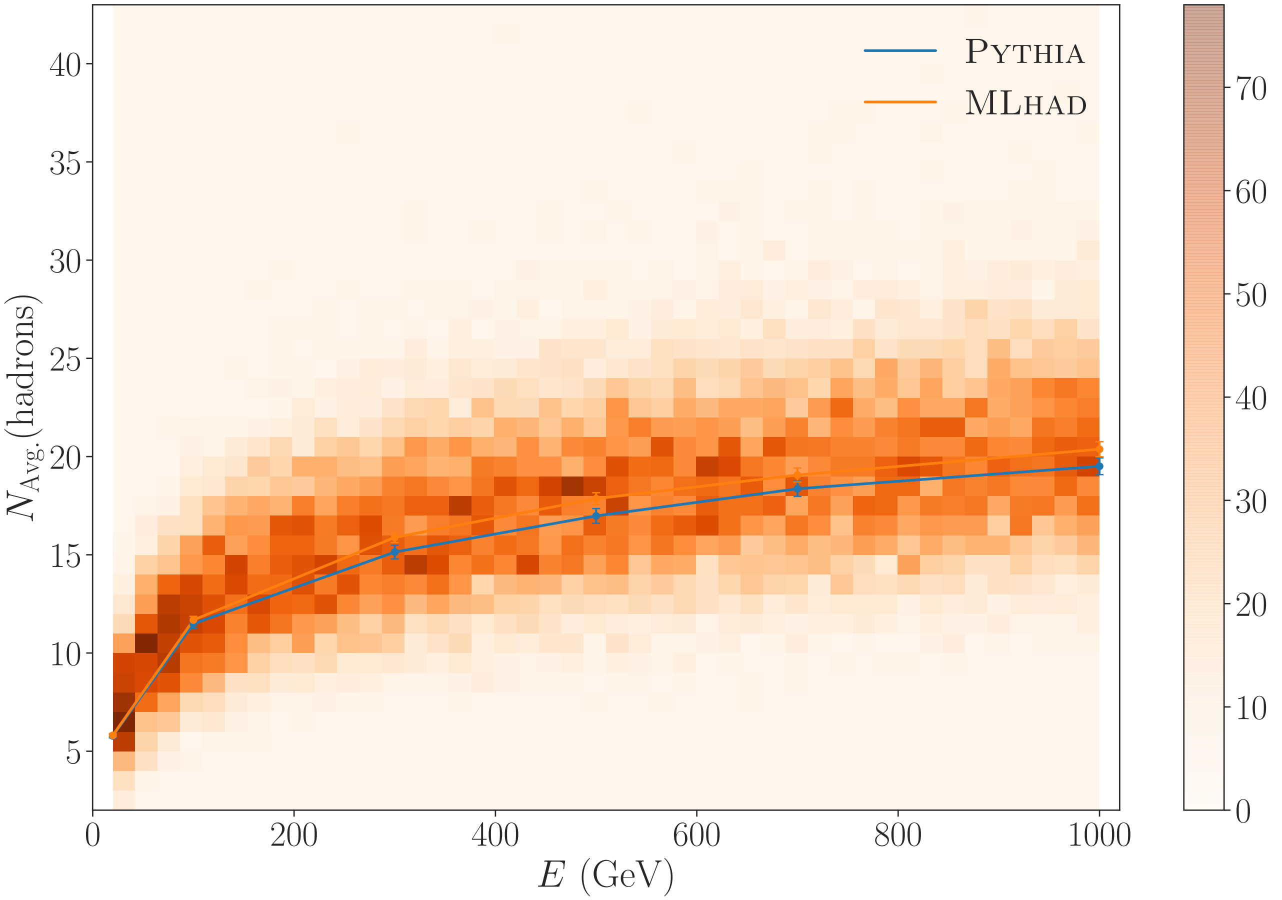
The trained model distributions now need to be integrated into a chain of fragmentations



Global observable (Hadron multiplicity cSWAE)



Global scaling (Hadron multiplicty vs string energy cSWAE)



Conclusion

Model + machine learning methods CAN be used to implement hadronization.

What's next:

- ML-improved (data-improved) model of hadronization
- ML flavor selector
- Error estimation
- Much more ☺

Check out our code!

MLHAD

https://gitlab.com/uchep/mlhad

Check out our paper!

(Recently accepted for publication in SciPost Physics)

arXiV: 2203.04983