Modeling Hadronization with Machine Learning

ML4Jets2022, Rutgers University

MLHAD Team: Phil Ilten, Tony Menzo, Stephen Mrenna, **Manuel Szewc**, Michael Wilkinson, Ahmed Youssef, and Jure Zupan

University of CINCINNATI

MLHAD

In this talk

I hope to convey how **Hadronization** is a complicated and worthwhile target for **ML@HEP** practitioners with a two part talk:

- Brief introduction to Hadronization.
- Efforts to apply ML to Hadronization, both published and preliminary.

This talk is based on arxiv:2203.04983 by Phil Ilten, Tony Menzo, Ahmed Youssef, and Jure Zupan and ongoing preliminary work.

Hadronization

Image from Pythia 8.3 manual

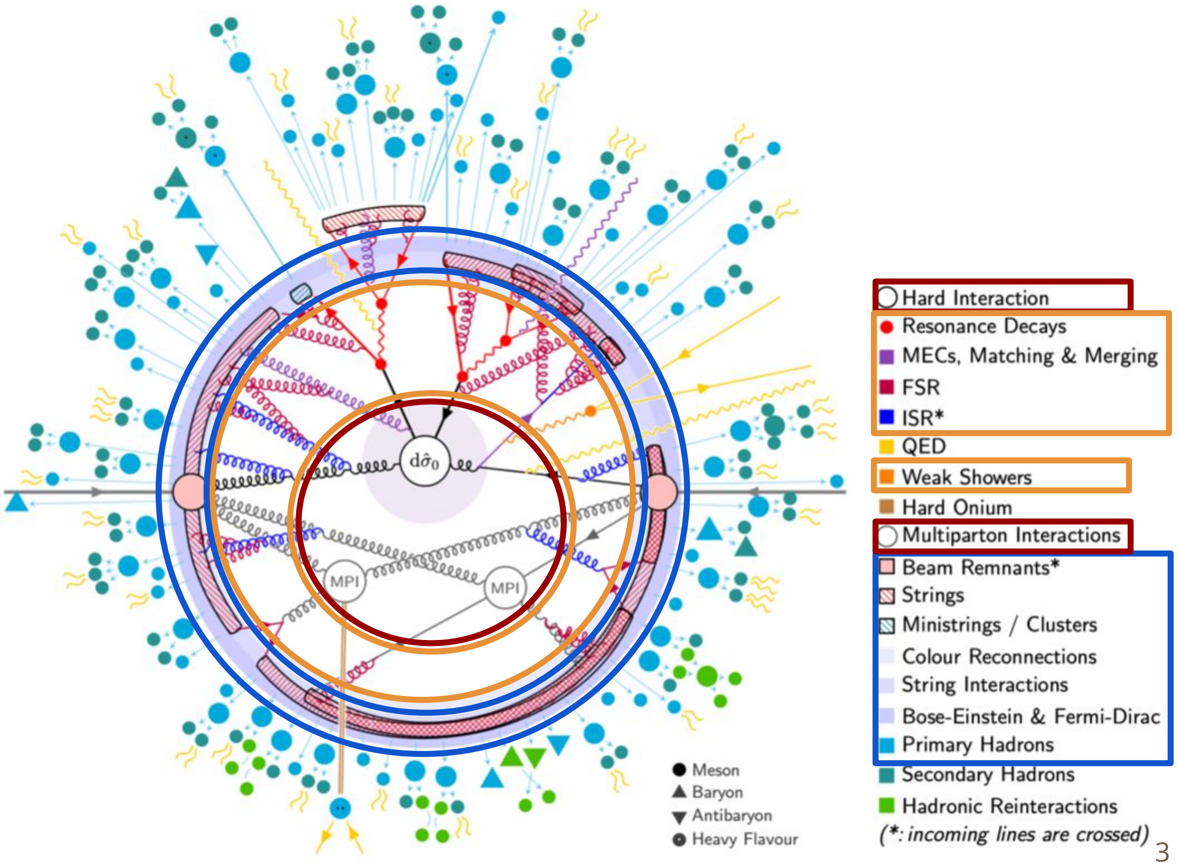
The radial coordinate is time

or 1/energy scale.

Hard process dσ, perturbatively calculated.

Perturbative evolution from hard to hadronization scale, also perturbative.

Hadronization: combining partons into hadrons. Non perturbative.



Modeling hadronization

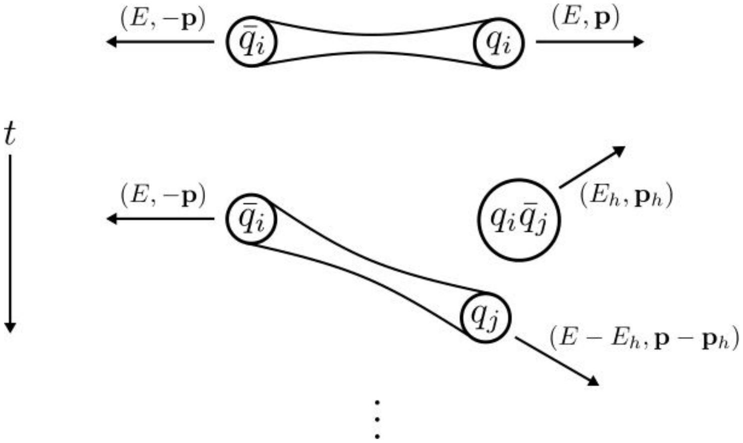
Hadronization is a inherently non-perturbative process → Empirical models for predictions.

Two main models: the **Lund String model (Pythia)** and the Cluster model (Herwig).

Lund String Model: Colored singlets + ~20 parameters → Hadrons

Each hadron is characterized by its four-momenta and its flavour. Translated into three variables of interest: **z, p_T, flavour.**

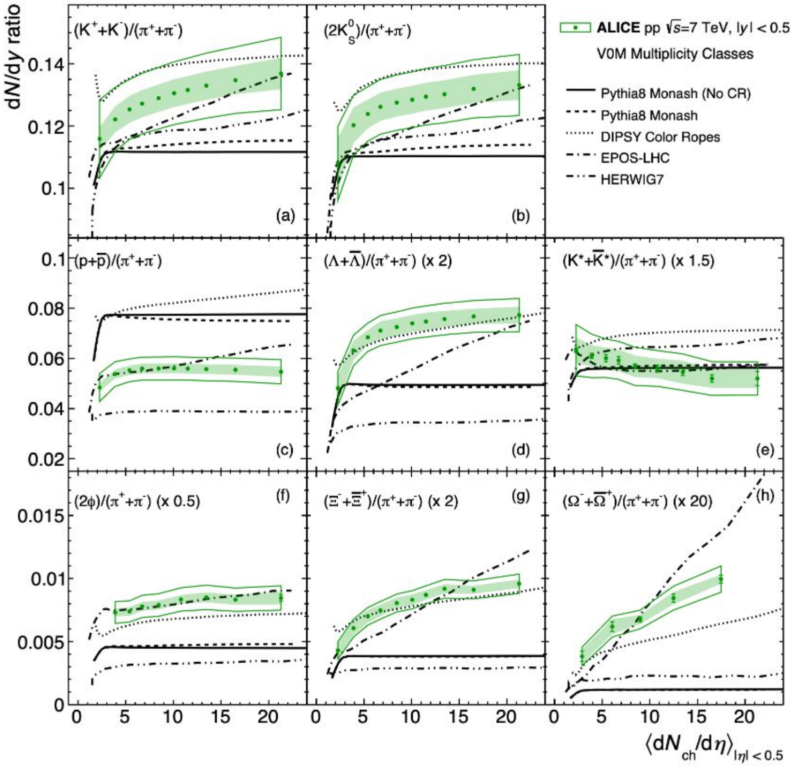
Simplified example from arxiv:2203.04983.



However...

Tuned Pythia is **very** successful. **However**, we are pushing the models to their limits.

Collective effects in general are tricky to recover e.g. heavy baryon production at high event multiplicities as in arxiv:1807.11321.



Machine Learning to the rescue?

Complex problem with **no full model flexible enough** and where **training is expensive**? → Machine Learning should be really useful here!

A lot of possible ways to attack this problem. The richness of the involved physics **forbids the use of any plug-and-play algorithms**.

Two recent papers on the subject: MLHAD (arxiv:2203.04983) and HADML (arxiv:2203.12660). Different generators (Pythia, Herwig) and different architectures (cSWAE, GAN) with different degrees of implementation.

Learning the Lund Fragmentation model

MLHAD learns the **Lund String first hadronization pdfs** for $e^+e^- \to q\bar{q}$ at various energies. Checks feasibility of the problem.

Introduce **inductive bias**. Improve over the existing empirical model by first mapping it to a learnable model.

The first hadronization pdf can be iteratively applied to get a full chain.

$$p_{\mathrm{Lund}}(x) \to p_{\mathrm{cSWAE}}(x) \qquad x = \{p_z, p_T\}$$

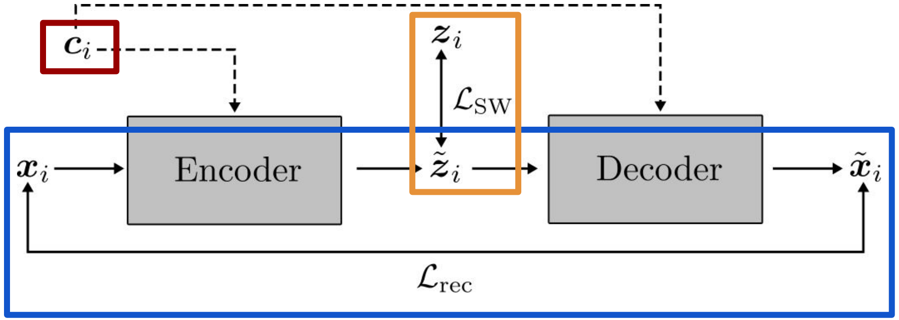
conditional Sliced Wasserstein AE

Probabilistic generative models are known to obtain **convincing physical observables** from **limited datasets** while retaining **flexibility** and **control of the output**.

Traditional Auto-Encoder with a **key difference:** Gaussian latent space → a more flexible distribution.

Achieved through the Sliced Wasserstein method for pdf distance computation.

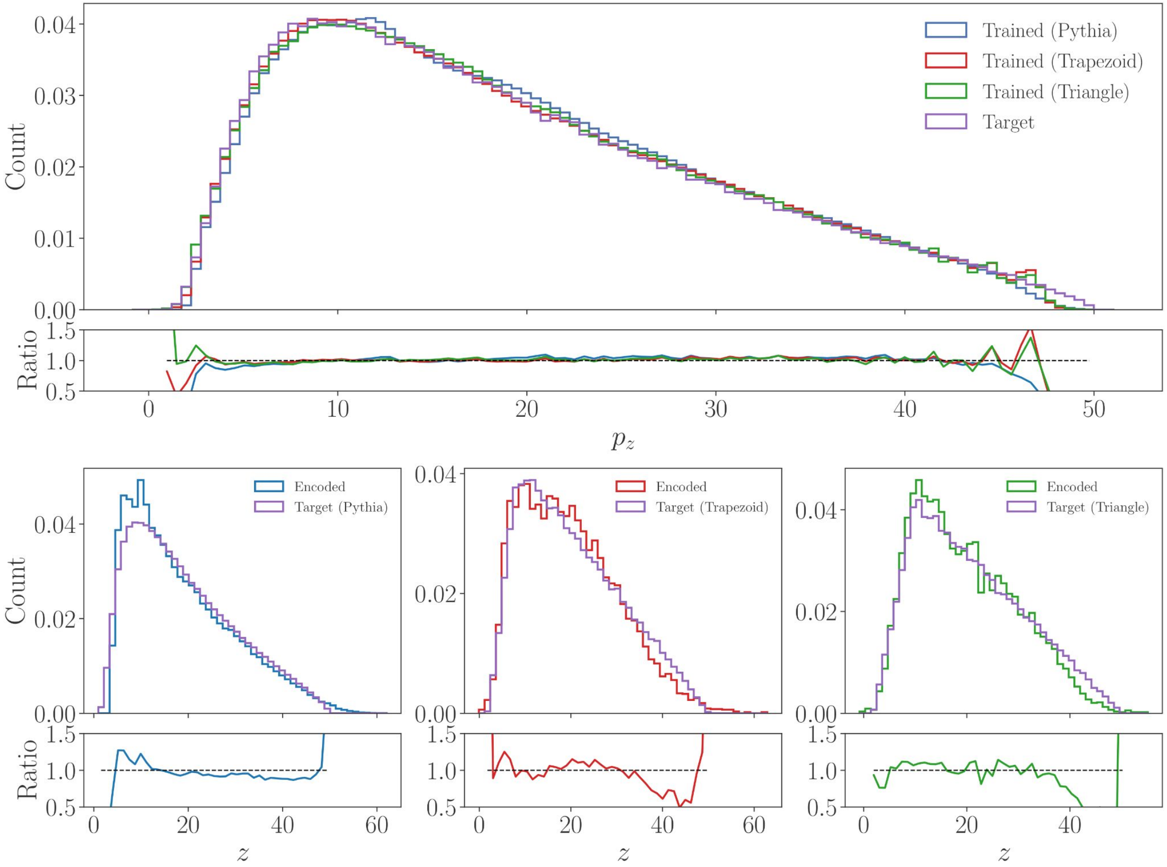
The energy of the string enters as a condition vector.



$\boldsymbol{c}_i$
$\boldsymbol{z}_i$
$\mathcal{L}_{\mathrm{SW}}$
$\boldsymbol{x}_i$
Encoder
$\tilde{\boldsymbol{z}}_i$
Decoder
$\tilde{\boldsymbol{x}}_i$
$\mathcal{L}_{\mathrm{rec}}$

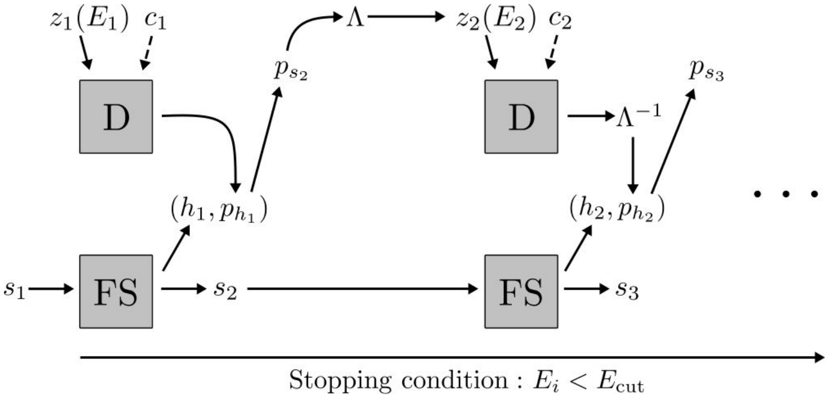
Learning Pythia

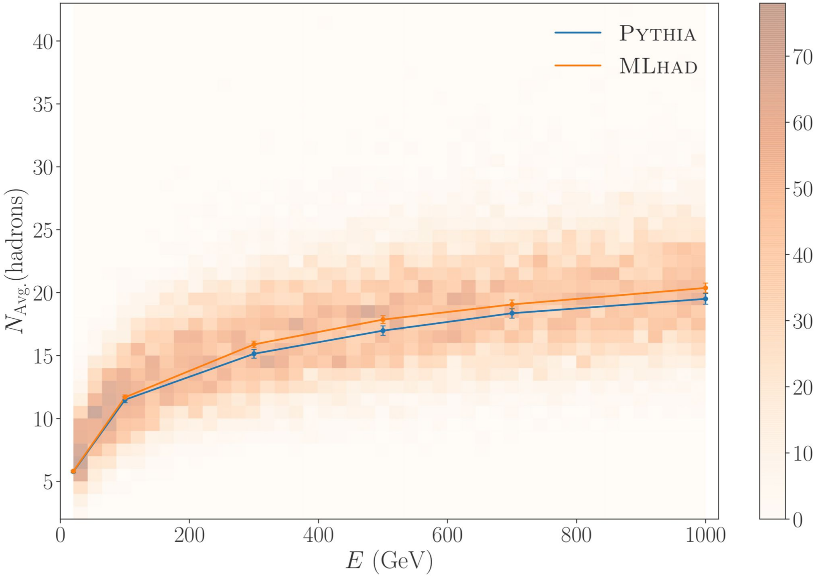
cSWAE learns the p_z and p_T spectrum for different string energies and different pdfs in the latent space.



Obtaining a full chain

Fragmentation chain of N successive hadronizations. cSWAE recovers the probabilistic distribution of the chain length conditioned on the initial energy of the string.





Limitations

cSWAE has only been trained in **each variable separately** and only for **simple strings producing pions**.

It assumes no correlation between hadronization steps.

Not a single hadronization generator: generate a batch and have to post-process them for a single chain generation.

Valid until a certain energy E_{cut}

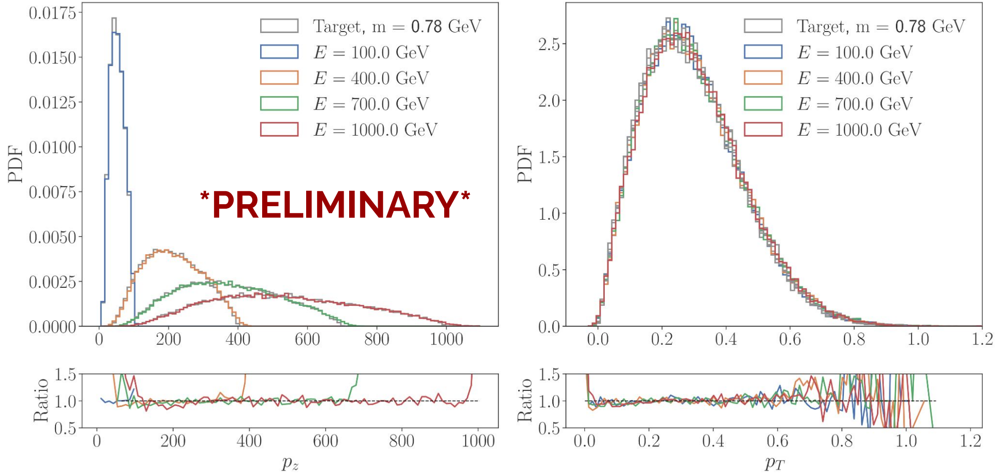
However, keep in mind this is a first crack at a very complicated problem.

Next directions: Improve upon first version

Sophistication of the full Machine Learning approach: different architectures (**Normalizing Flows**), more flavours, more colour topologies... Maybe dispense of the String altogether?

NFs can train on both variables at the same time + single hadronization pdf

It’s working very well conditioned on flavour and energy!



PRELIMINARY

Next directions: embed our model in Pythia

Pythia already handles different string topologies and their recursive splittings to produce hadrons→ Let’s take advantage of it.

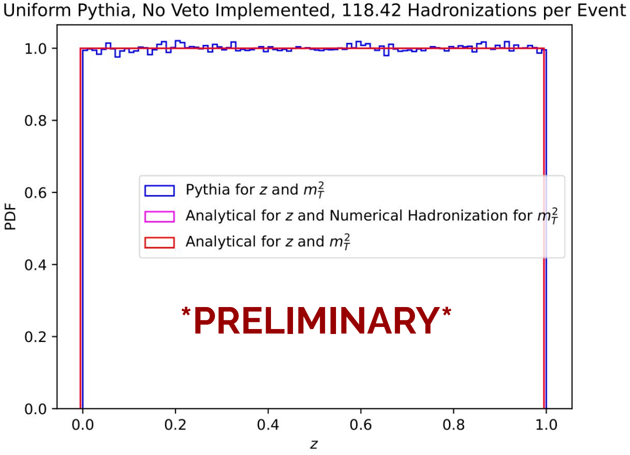
Think in modules → **Replace the fragmentation functions for z, p_T and flavour.**

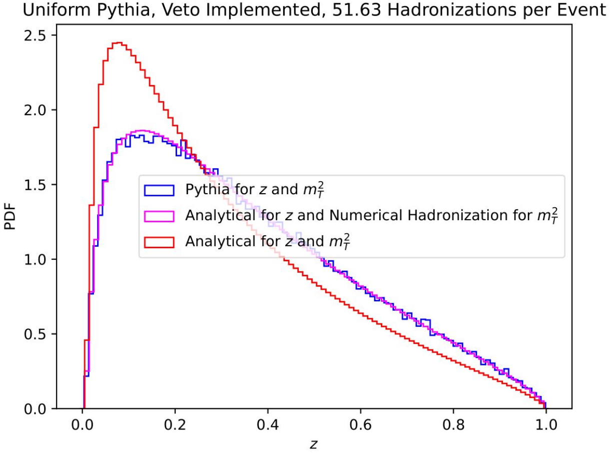
Perform Rejection Sampling with UserHooks in Pythia and reduce the problem to learning a **re-weighting function**

$$w(z, p_T, \text{Flavour}) = \frac{p_{\text{Data}}(z, p_T, \text{Flavour})}{p_{\text{Pythia}}(z, p_T, \text{Flavour})}$$

Preliminary results

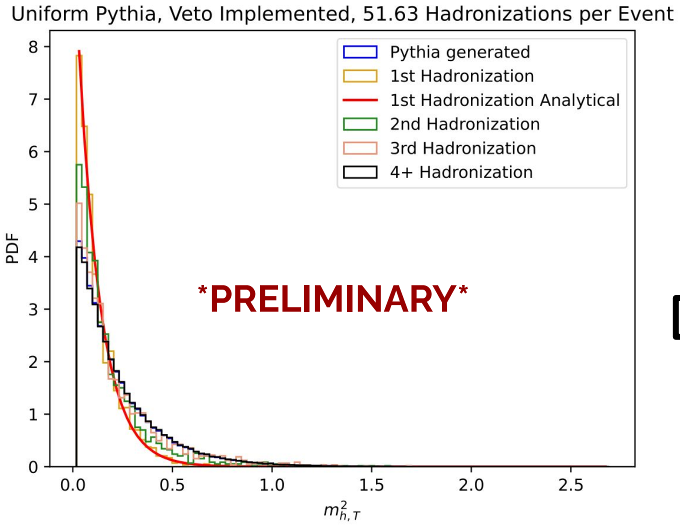
We re-compiled Pythia to have uniform distributions and morph them to the Lund String distributions.





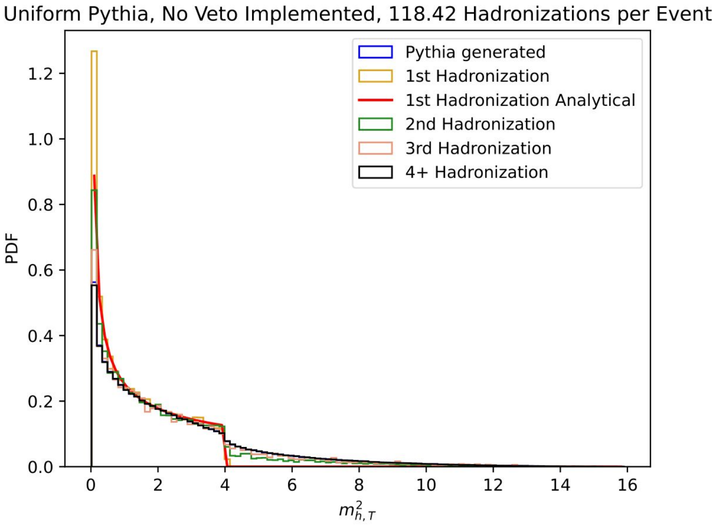
An example of an observable for training:

How to train? We have a **non-differentiable output + goodness-of-fit metric**. Several options: Reinforcement Learning, Simulation Based Inference, Nested Sampling.Advantages and disadvantages to all of them...



Uniform Pythia, Veto Implemented, 51.63 Hadronizations per Event

PRELIMINARY



Uniform Pythia, No Veto Implemented, 118.42 Hadronizations per Event

Conclusions

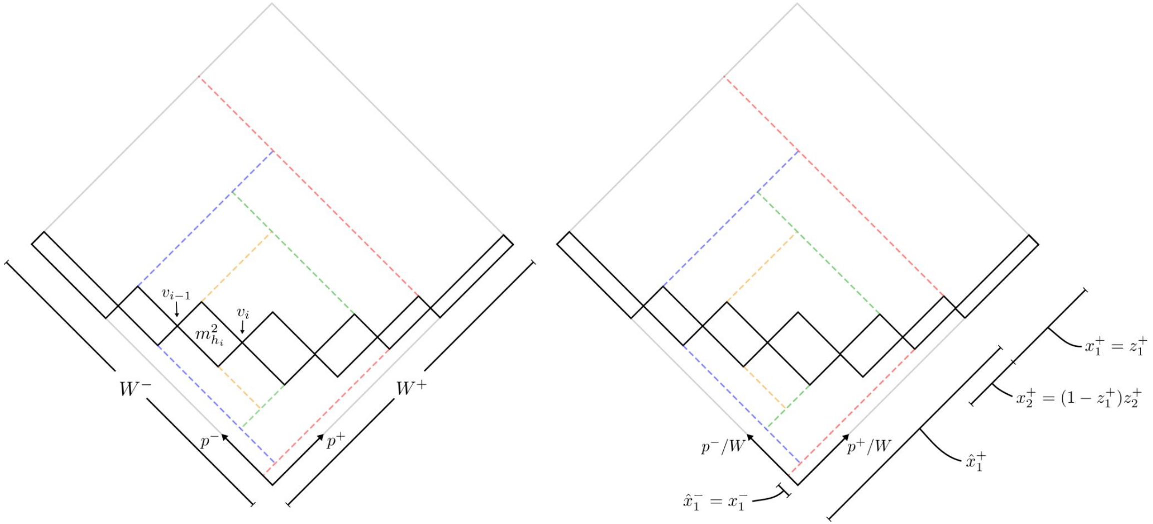
Hadronization is the type of problem you dream of if you want to work in ML for HEP: **physically meaningful** and **complicated enough** that it is not simply a case of plug-and-play with any ML algorithm.

The variety of colour topologies and correlations between hadronizations pose a challenge to represent in an appropriate manner for any learnable algorithm.

Training itself is an issue! Development of Simulation Based Inference or Nested Sampling could be really useful.

Backup slides

Momentum space for finding next hadronization



Limitations of the Lund String model

O(20%) to O(50%) discrepancies between proton-proton and ion-ion collisions

Heavy particle composition as a function of event multiplicity is mismodelled at high event multiplicities

Mismodelling of the mass dependence of the average transverse momentum

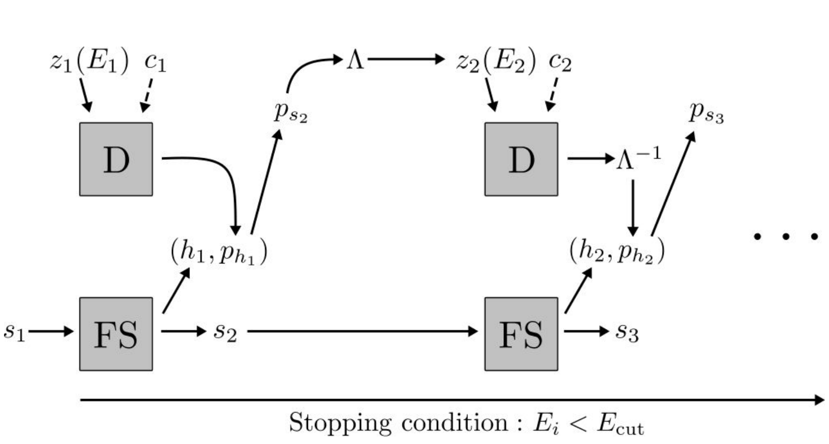
Minimum bias description can be incompatible because of low transverse momentum mismodelling

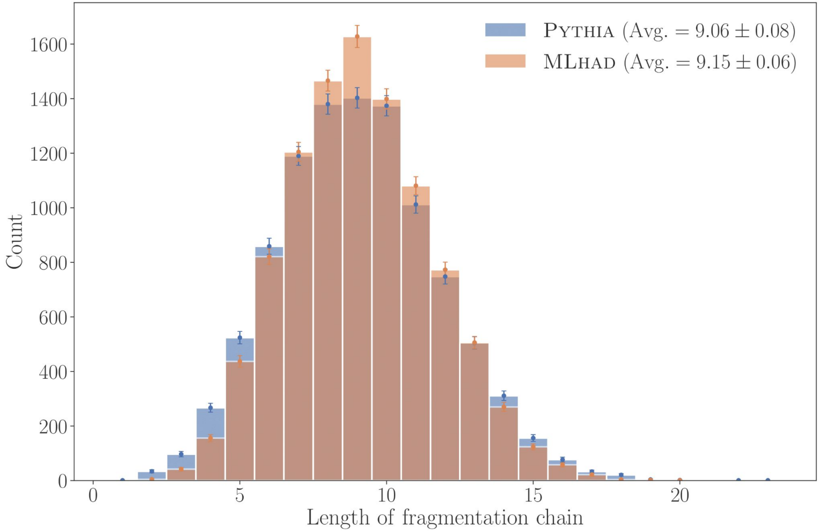
Ridge in pp collisions missing in Pythia (and in general long range correlations are hard to model)

Charged particle multiplicity spectrum is very sensitive to color reconnections and MPI modelling

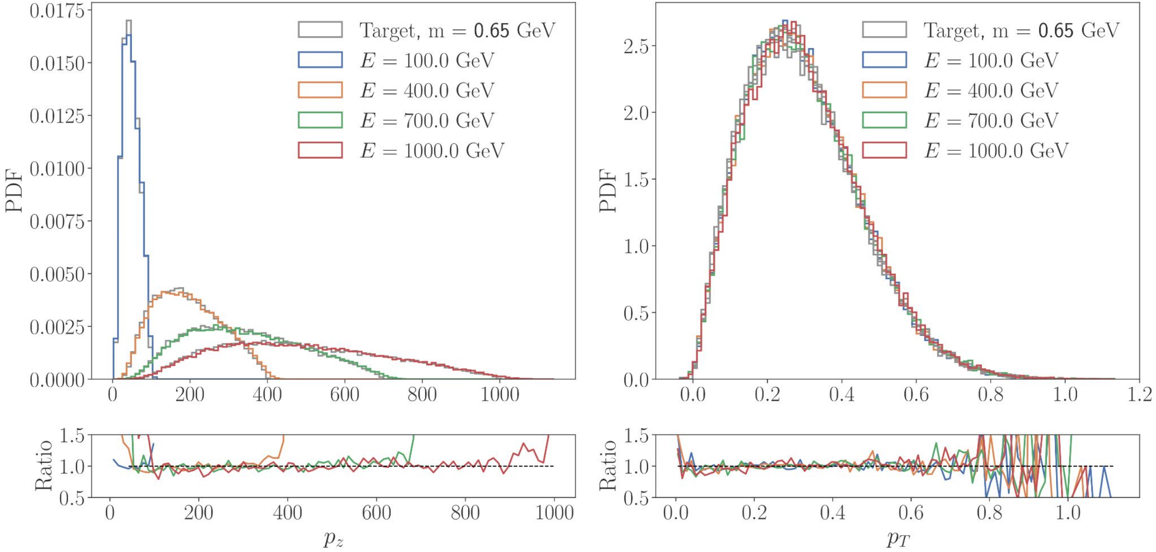
Obtaining a full chain

Fragmentation chain of N successive hadronizations.

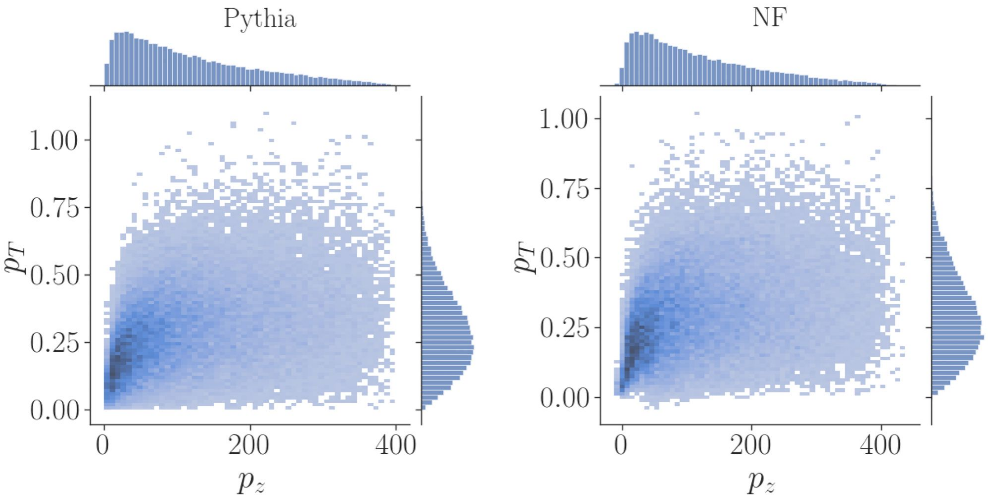




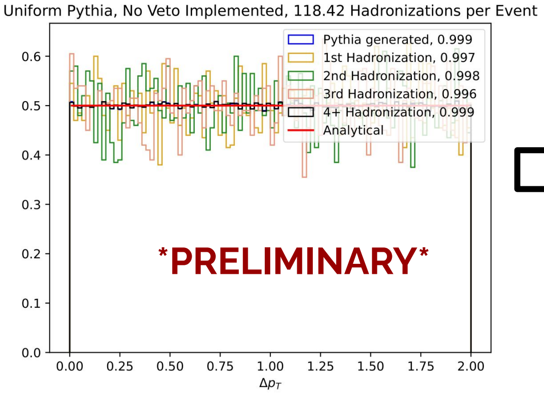
Conditional Normalizing Flows *PRELIMINARY*



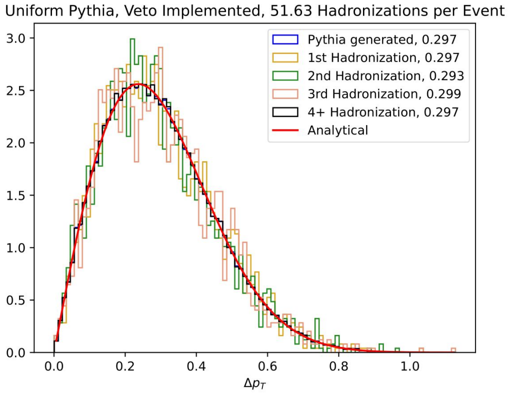
Conditional Normalizing Flows *PRELIMINARY*



The p_T distribution

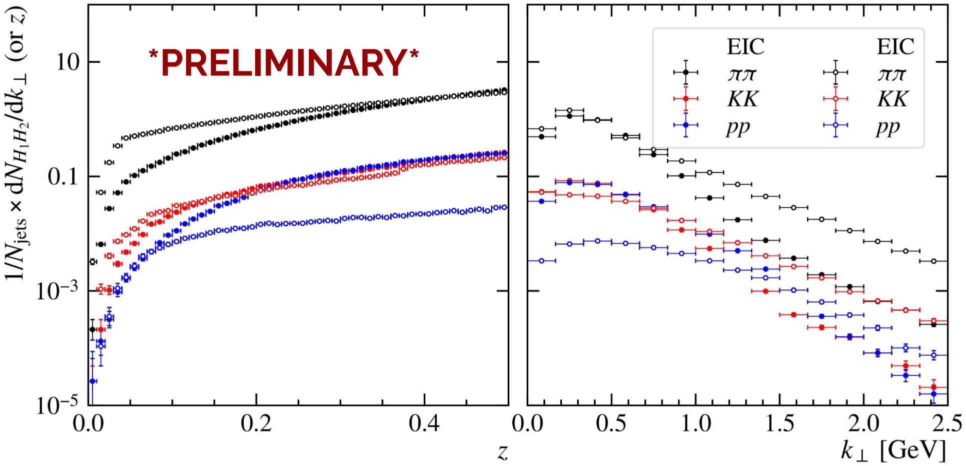

Uniform Pythia, No Veto Implemented, 118.42 Hadronizations per Event

PRELIMINARY


Uniform Pythia, Veto Implemented, 51.63 Hadronizations per Event

Next directions: Observable choices

Definition of better observables for training. We need observables sensitive to differences in the hadronization models


PRELIMINARY
$1/N_{\text{jets}} \times dN_{H_1H_2}/dk_\perp$ (or z)
10^{-3}
10^{-5}
z
$k_\perp$ [GeV]
EIC
$\pi\pi$
KK
pp
EIC
$\pi\pi$
KK
pp