Fitting of a Deep Generative Hadronization Model

Andrzej Siódmok

Towards a Deep Learning Model for Hadronization

Aishik Ghosh,[a,b] Xiangyang Ju,[b] Benjamin Nachman,[b,c] and Andrzej Siodmok[d]

[a] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[b] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[c] *Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA*
[d] *Jagellonian University, Krakow, Poland*

2203.12660

Fitting a Deep Generative Hadronization Model

Jay Chan,[a,b] Xiangyang Ju,[b] Adam Kania,[e] Benjamin Nachman,[b,c] Vishnu Sangli,[d,b] and Andrzej Siodmok[d]

[a] *Department of Physics, University of Wisconsin-Madison, Madison, WI 53706, USA*
[b] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[c] *Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA*
[d] *Department of Physics, University of California, Berkeley, CA 94720, USA*
[e] *Jagellonian University, Krakow, Poland*

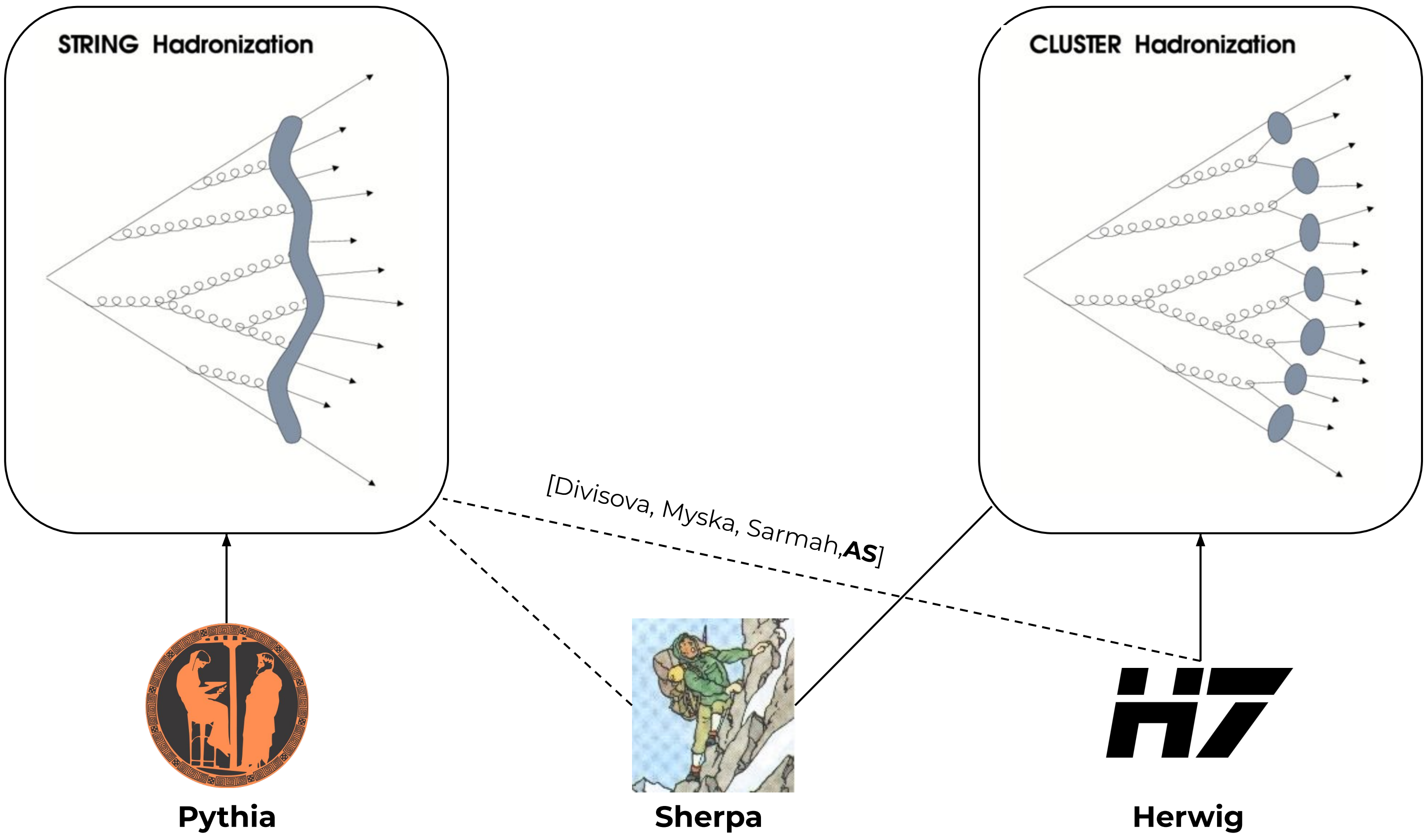
2305.17169

JAGIELLONIAN UNIVERSITY IN KRAKÓW
UCZELNIA BADAWCZA INICJATYWA DOSKONAŁOŚCI
NARODOWE CENTRUM NAUKI
NCN: 2019/34/E/ST2/00457
H7
MCnet

Hadronization:

➔ Increased control of perturbative corrections ⇒ more often LHC measurements are limited by non-perturbative components, such as hadronization.

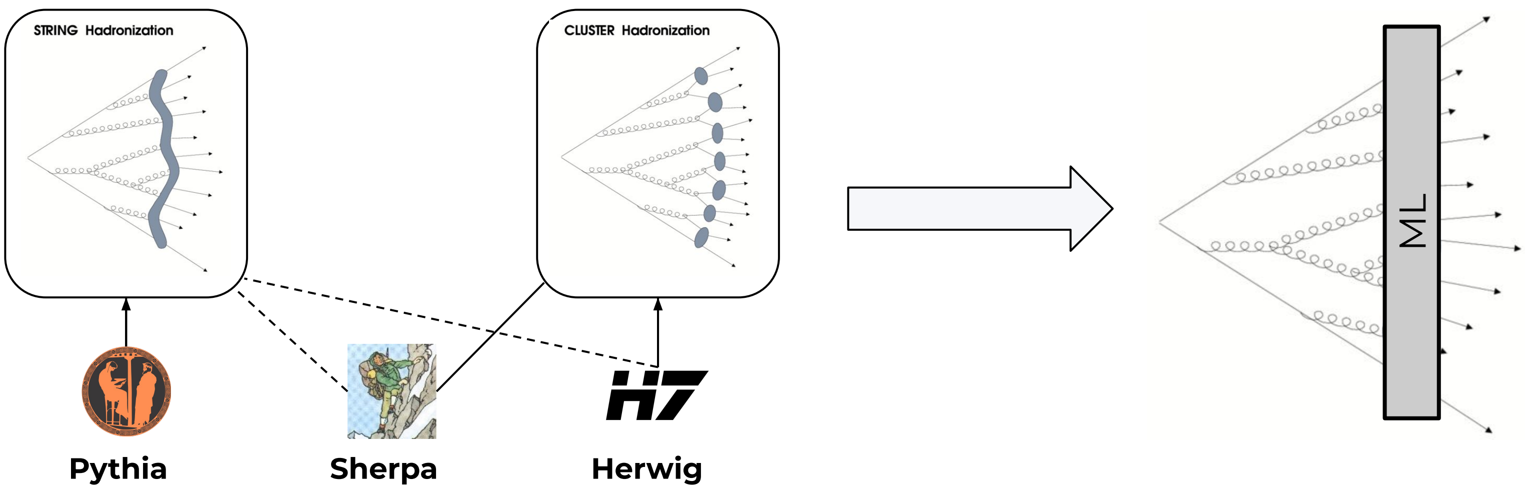
- W mass measurement using a new method [Freytsis at al. JHEP 1902 (2019) 003]
- Extraction of the strong coupling in [M. Johnson, D. Maître, Phys.Rev. D97 (2018) no.5]
- Top mass [S. Argyropoulos, T. Sjöstrand, JHEP 1411 (2014) 043]
- ...



Hadronization:

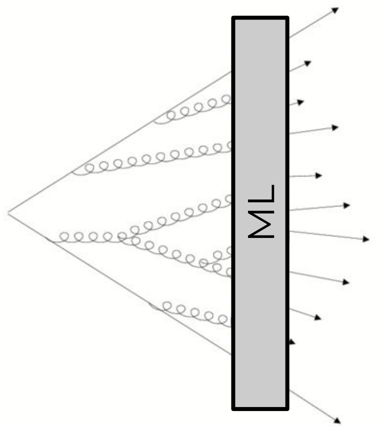
Early 1980's

Early 2020's
(lot of progress in ML)



Idea of using Machine Learning (ML) for hadronization.

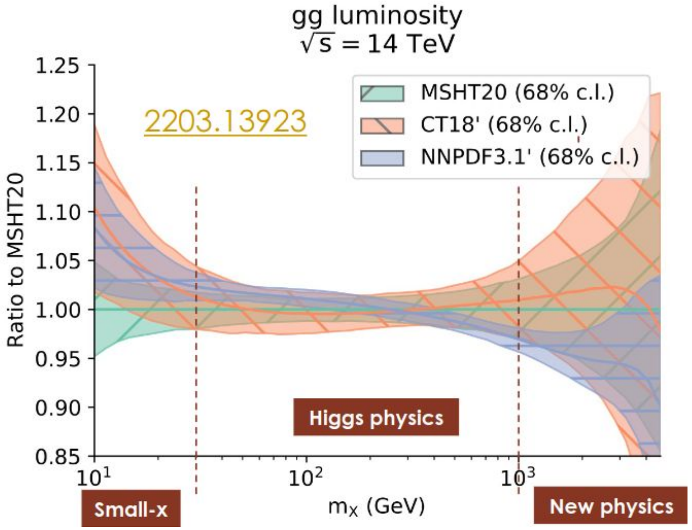
Idea of using Machine Learning (ML) for hadronization.



- Existing hadronization models are highly parameterized functions.
- Hadronization is a fitting problem
  - Can ML hadronization be more flexible to fit the data?
  - Can ML hadronization extract more information from the data? [can accommodate unbinned and high-dimensional inputs]

NNPDF

NNPDF used successfully ML to nonperturbative Parton Density Functions (PDF).
Hadronization is closely related to fragmentation functions (FF) which were considered the counterpart of PDFs.



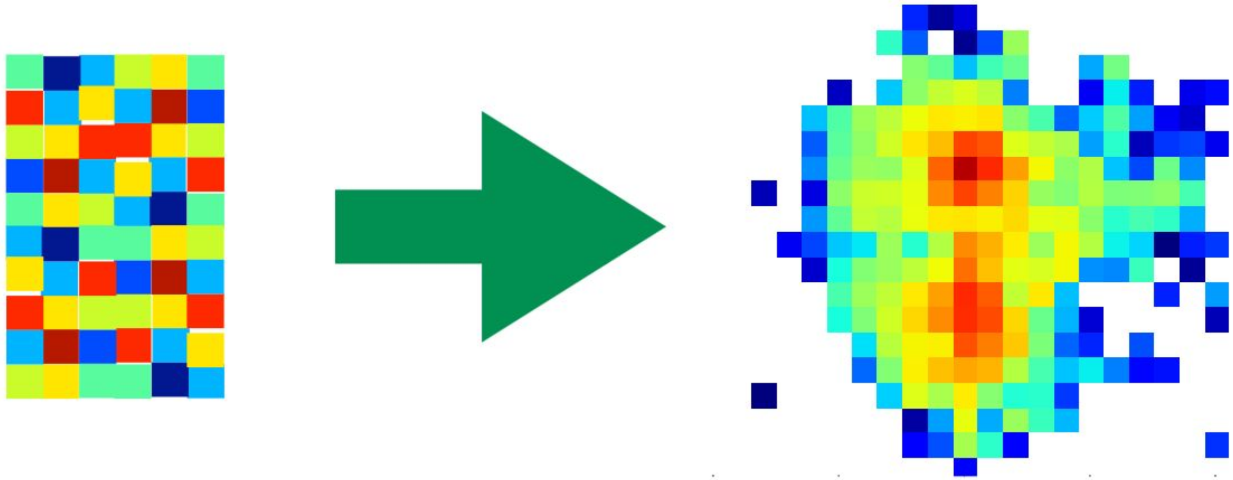
First steps for ML hadronization:

- HADML - [A. Ghosh, Xi. Ju, B. Nachman **AS**, *Phys.Rev.D* 106 (2022) 9]
- MLhad - [P. Ilten, T. Menzo, A. Youssef and J. Zupan, SciPost Phys. 14, 027 (2023)]

	MLhad	HADML
Deep generative model:	Variational Autoencoder	Generative Adversarial Networks
Trained on:	String model	Cluster model
Recent progress:	*“Reweighting Monte Carlo Predictions and Automated Fragmentation Variations in Pythia 8”* [Bierlich, Ilten, Menzo, Mrenna, Szewc, Wilkinson, Youssef, Zupan, 2308.13459] (see Christian’s talk)	*“Fitting a Deep Generative Hadronization Model”* [J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli and **AS**, JHEP 09 (2023) 084]

What is a deep generative model?

A **generator** is nothing other than a function that maps random numbers to structure.

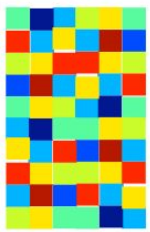


Deep generative models: the map is a deep neural network.

Our tool of choice: GANs

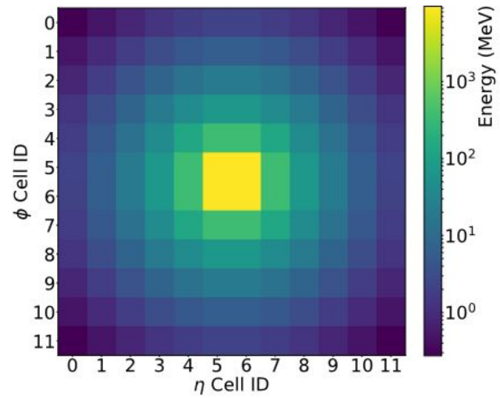
[Goodfellow et al. “Generative adversarial nets”. arxiv:1406.2661]

Generative Adversarial Networks (GANs):
*A two-network game where one **maps noise to structure** and one **classifies images as fake or real**.*



noise

G

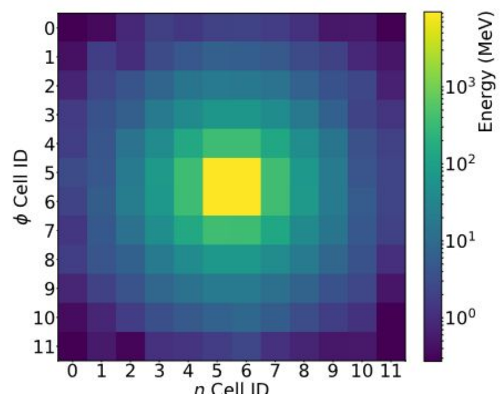


D

{real,fake}

D

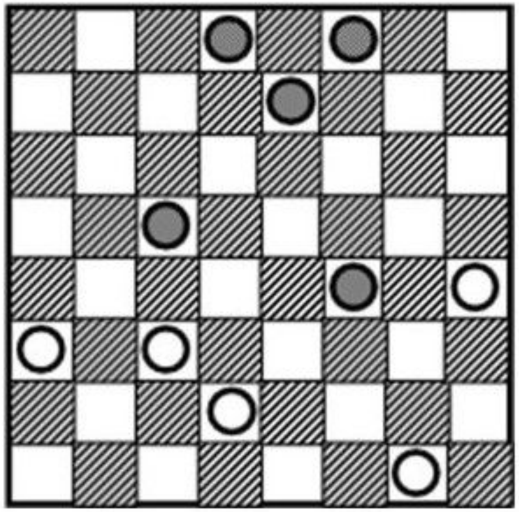
When **D** is maximally confused, **G** will be a good generator



Physics-based simulator or data

Adversarial Networks

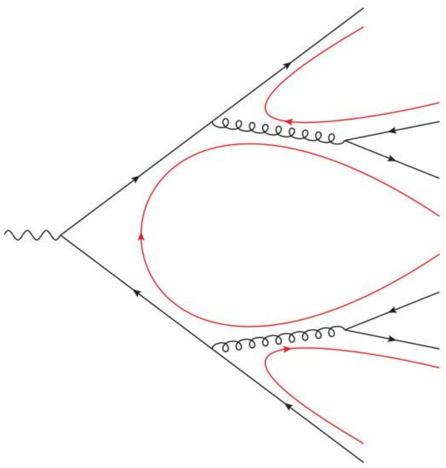
Arthur Lee Samuel (1959) wrote a program that learnt to play checkers well enough to beat him.



- He popularized the term **"machine learning"** in 1959.
- The program chose its move based on a **minimax** strategy, meaning it made the move assuming that the opponent was trying to optimize the value of the same function from its point of view.
- He also had it play thousands of **games against itself** as another way of learning.

The philosophy of the model: use information from perturbative QCD as an input for hadronization.

QCD **pre-confinement** discovered by Amati & Veneziano:

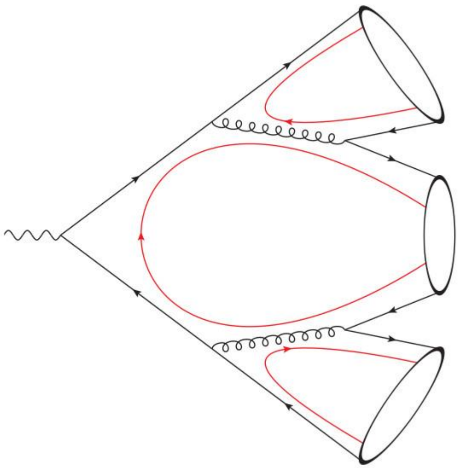


- QCD provide pre-confinement of colour

Cluster hadronization model

The philosophy of the model: use information from perturbative QCD as an input for hadronization.

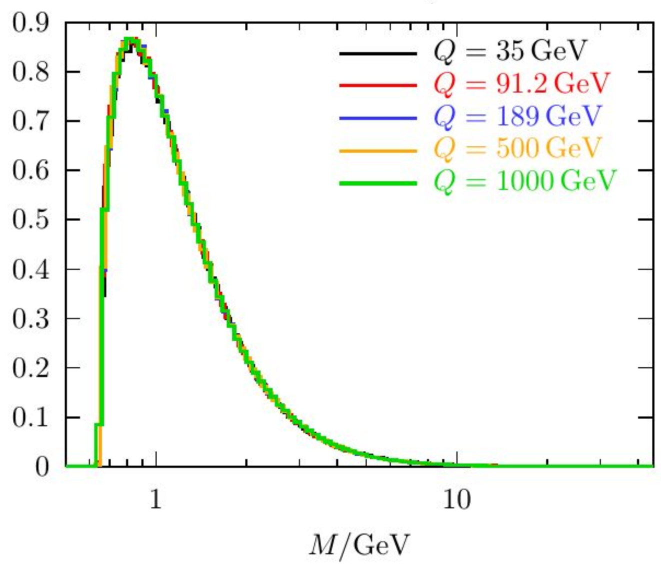
QCD **pre-confinement** discovered by Amati & Veneziano:



- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters

The philosophy of the model: use information from perturbative QCD as an input for hadronization.

QCD **pre-confinement** discovered by Amati & Veneziano:



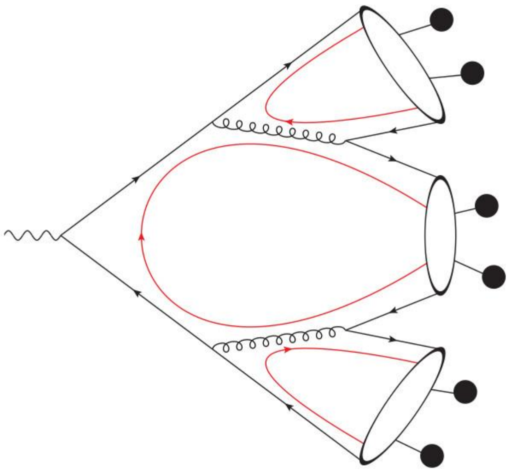
[S. Gieseke, A. Ribon, MH Seymour,
P Stephens, B Webber JHEP 0402 (2004) 005]

- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision

Cluster hadronization model

The philosophy of the model: use information from perturbative QCD as an input for hadronization.

QCD **pre-confinement** discovered by Amati & Veneziano:

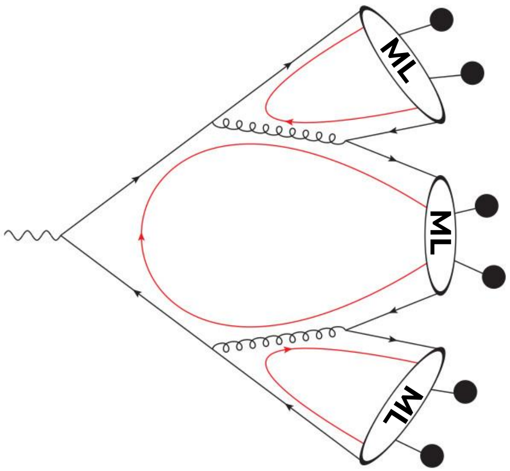


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically decay into 2 hadrons

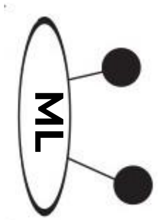
Cluster hadronization model

The philosophy of the model: use information from perturbative QCD as an input for hadronization.

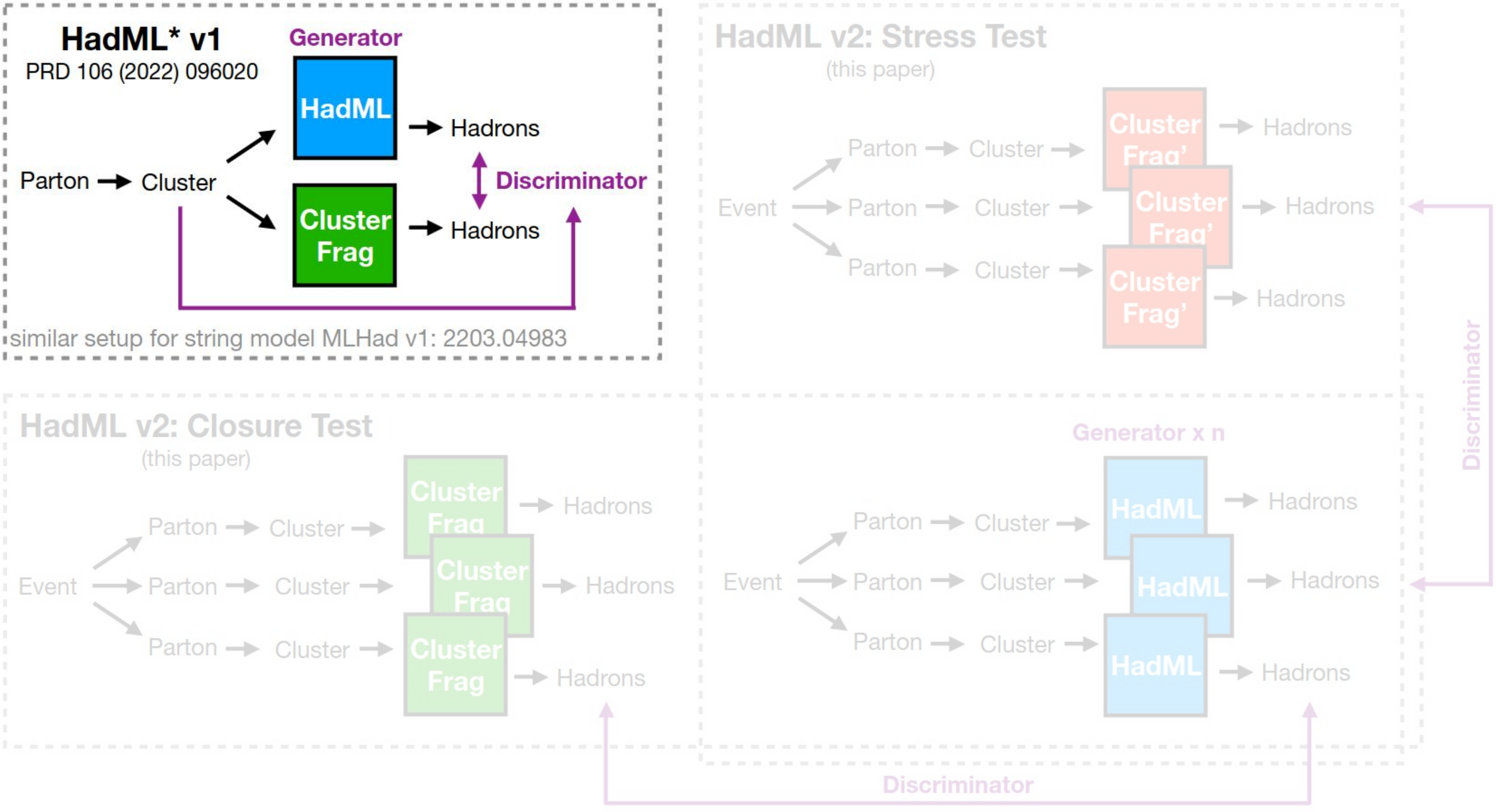
QCD **pre-confinement** discovered by Amati & Veneziano:



- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically decay into 2 hadrons
- **ML hadronization**
  1st step: generate kinematics of a cluster decay:



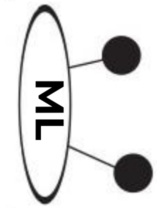
Road map for today



Towards a Deep Learning Model for Hadronization

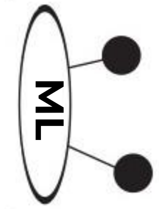
ML hadronization

1st step: generate kinematics of a cluster decay to 2 hadrons



ML hadronization

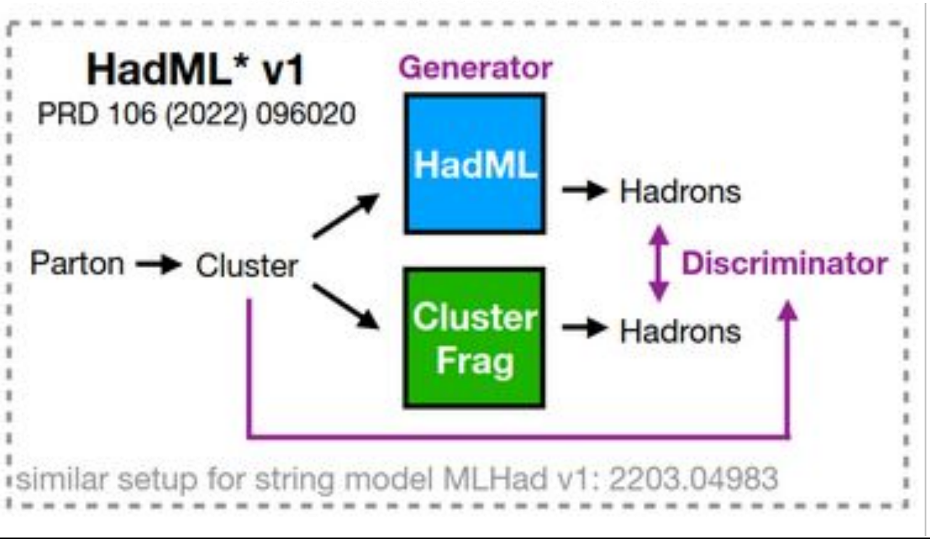
1st step: generate kinematics of a cluster decay to 2 hadrons



How?

We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.

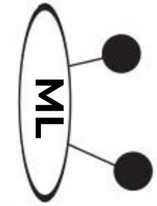
Generative Adversarial Net



Towards a Deep Learning Model for Hadronization

ML hadronization

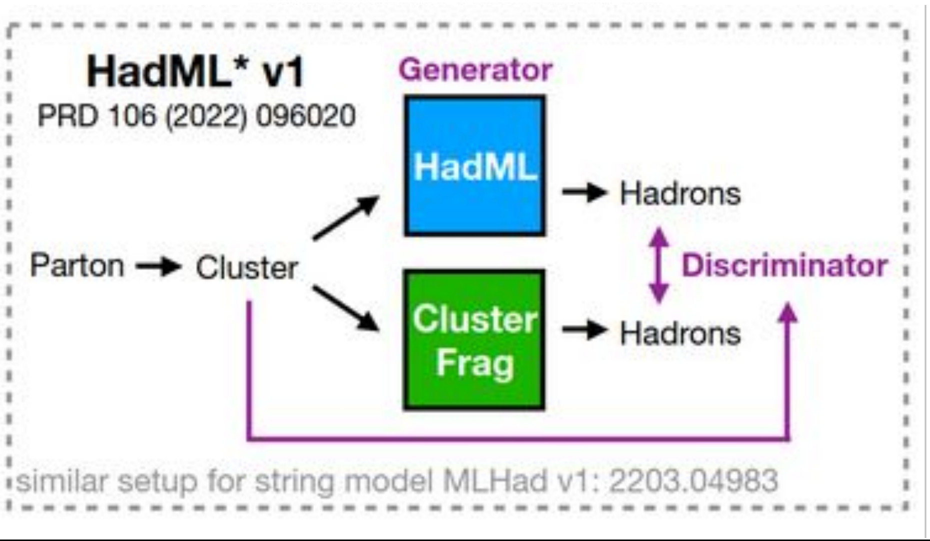
1st step: generate kinematics of a cluster decay to 2 hadrons



How?

Generative Adversarial Net

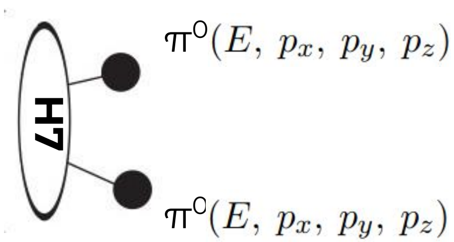
We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.



Training data:

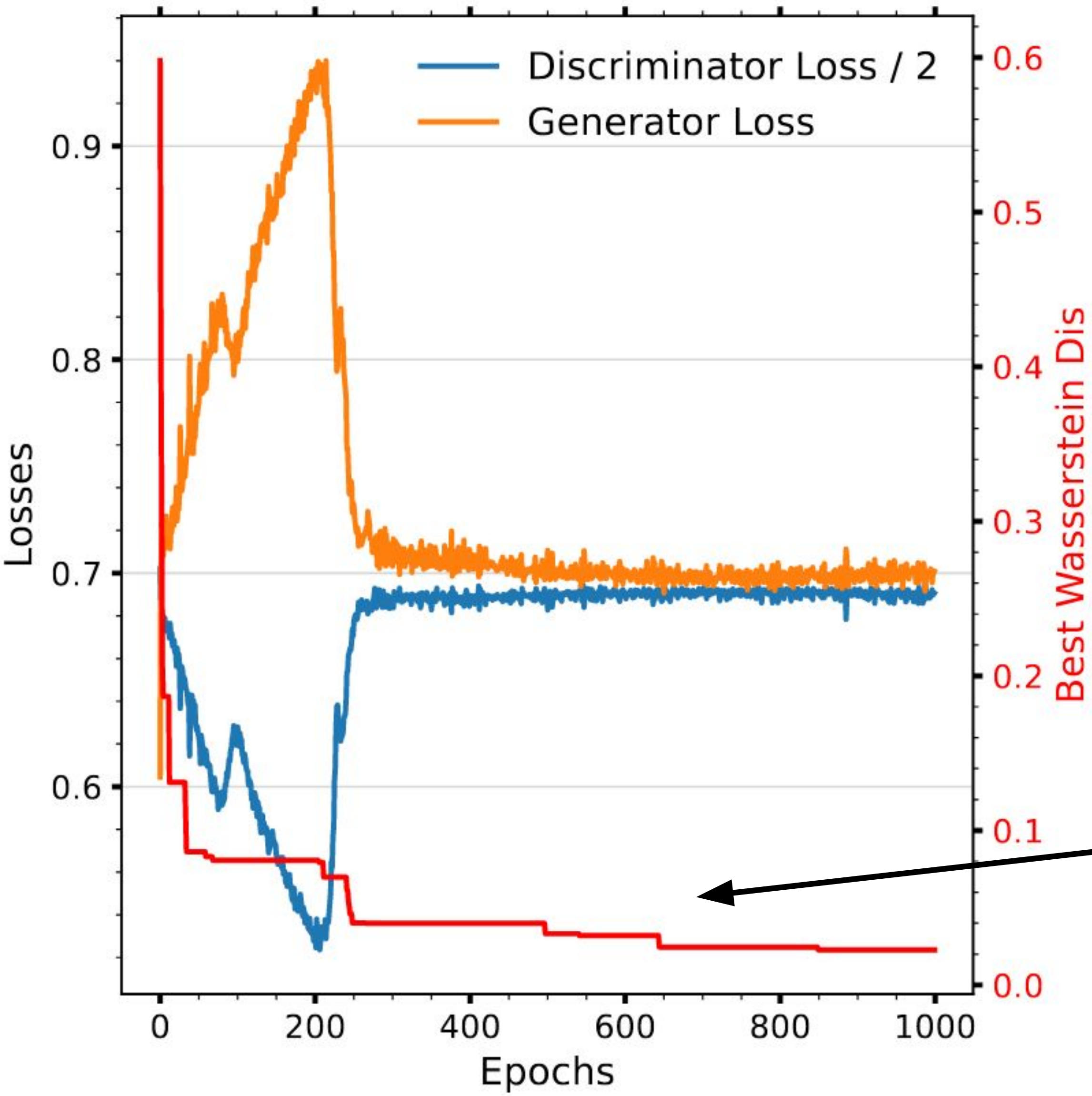
e^+e^- collisions at $\sqrt{s} = 91.2$ GeV

Cluster $(E,\ p_x,\ p_y,\ p_z)$



Simplification: considering only pions and generating two angles in the cluster rest frame.

Training HADML v1



We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.

Simplification: considering only pions and generating two angles in the cluster rest frame.

This is a typical learning curve for GAN training

Integration into Herwig

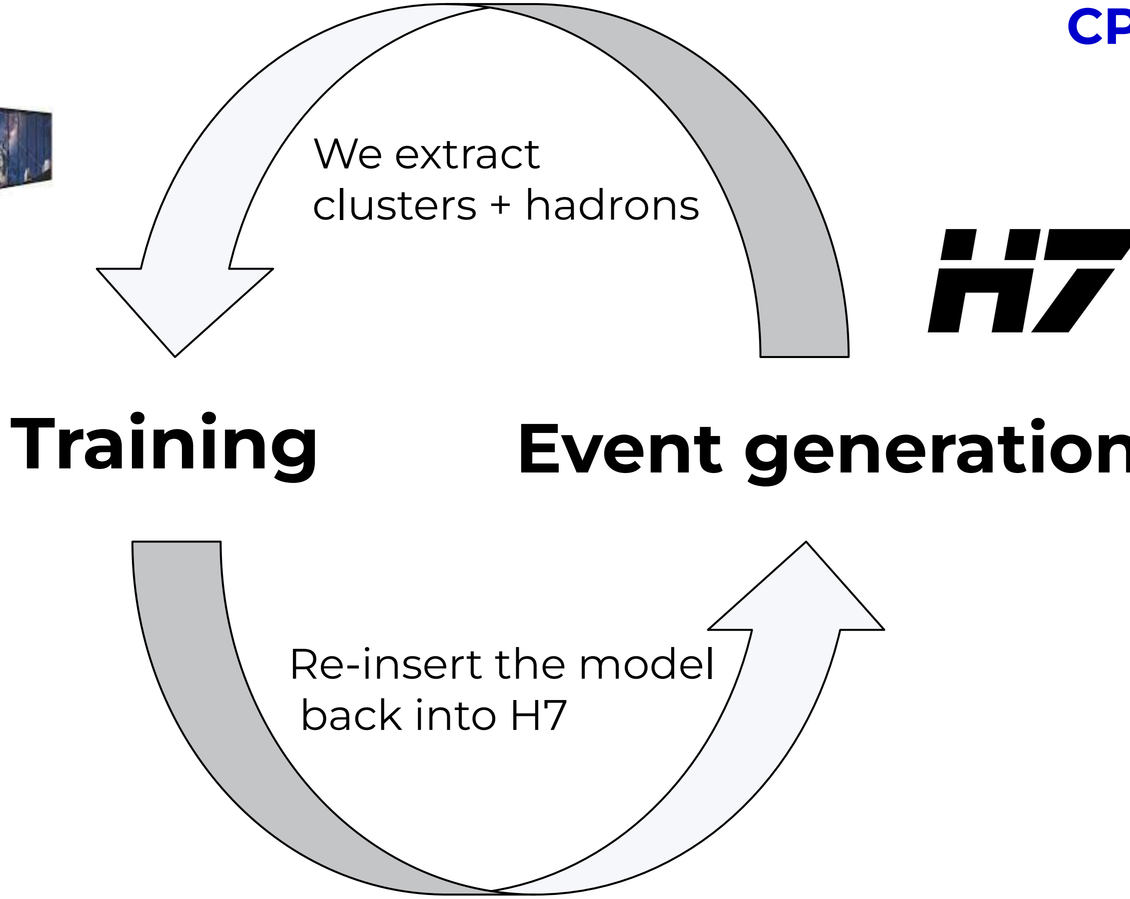
GPU

TensorFlow

python™

CPU

H7

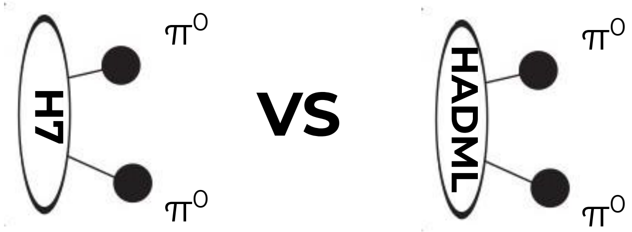


ONNX RUNTIME

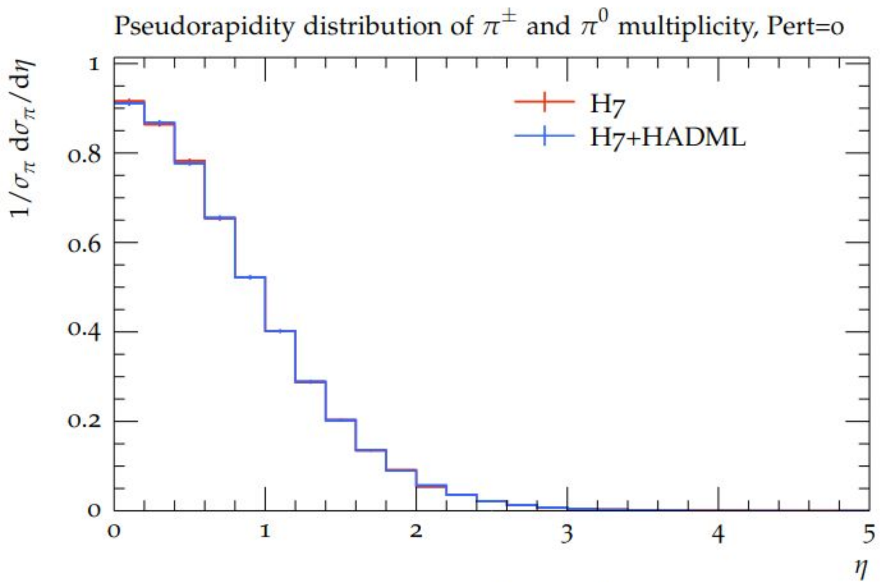
This then allows us to run a full event generator and produce plots

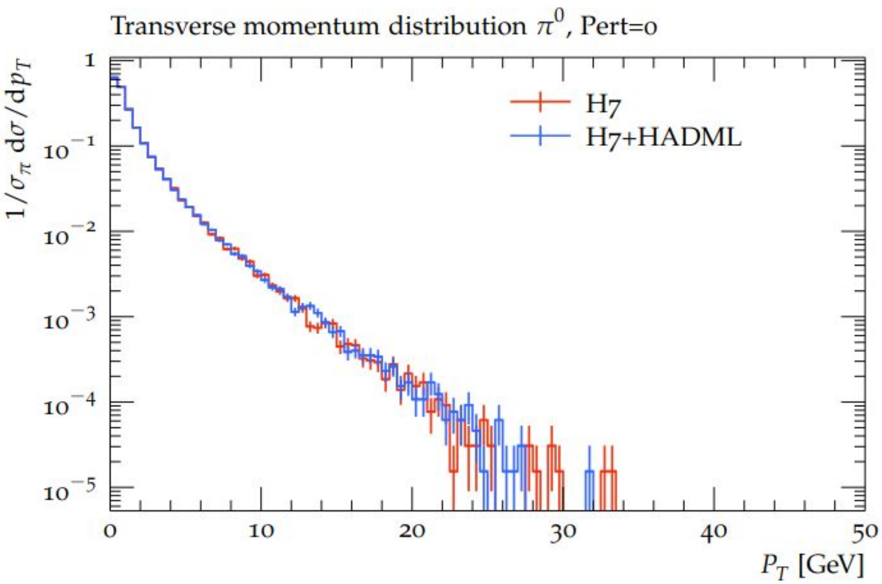
Low-level Validation
(similar to training data)

e^+e^- collisions at
$\sqrt{s} = 91.2$ GeV



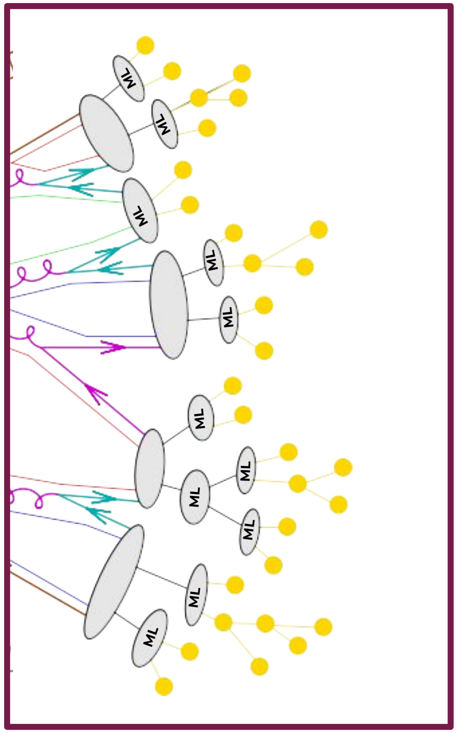
π^0 kinematic variables

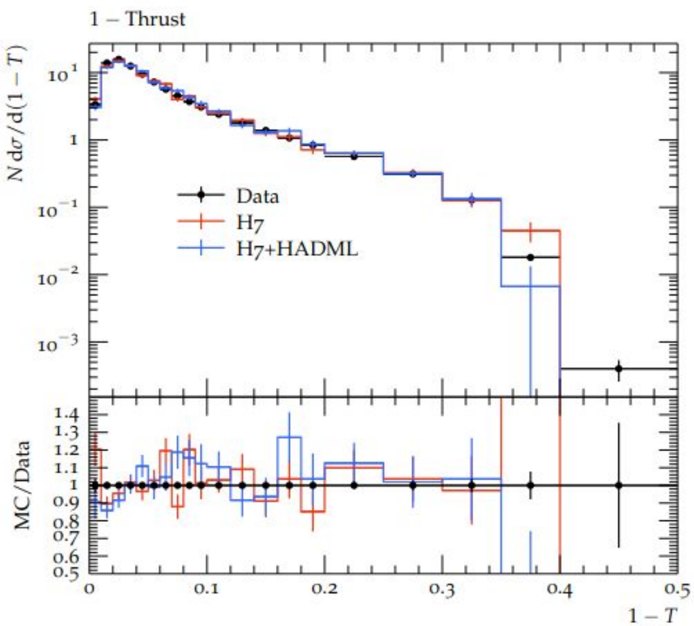


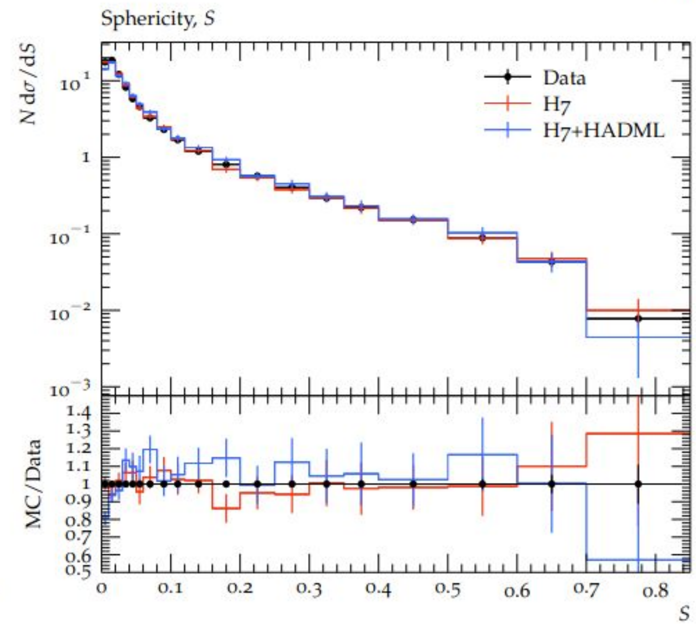


With a "full" model, we can compare directly to data!

LEP DELPHI **Data**

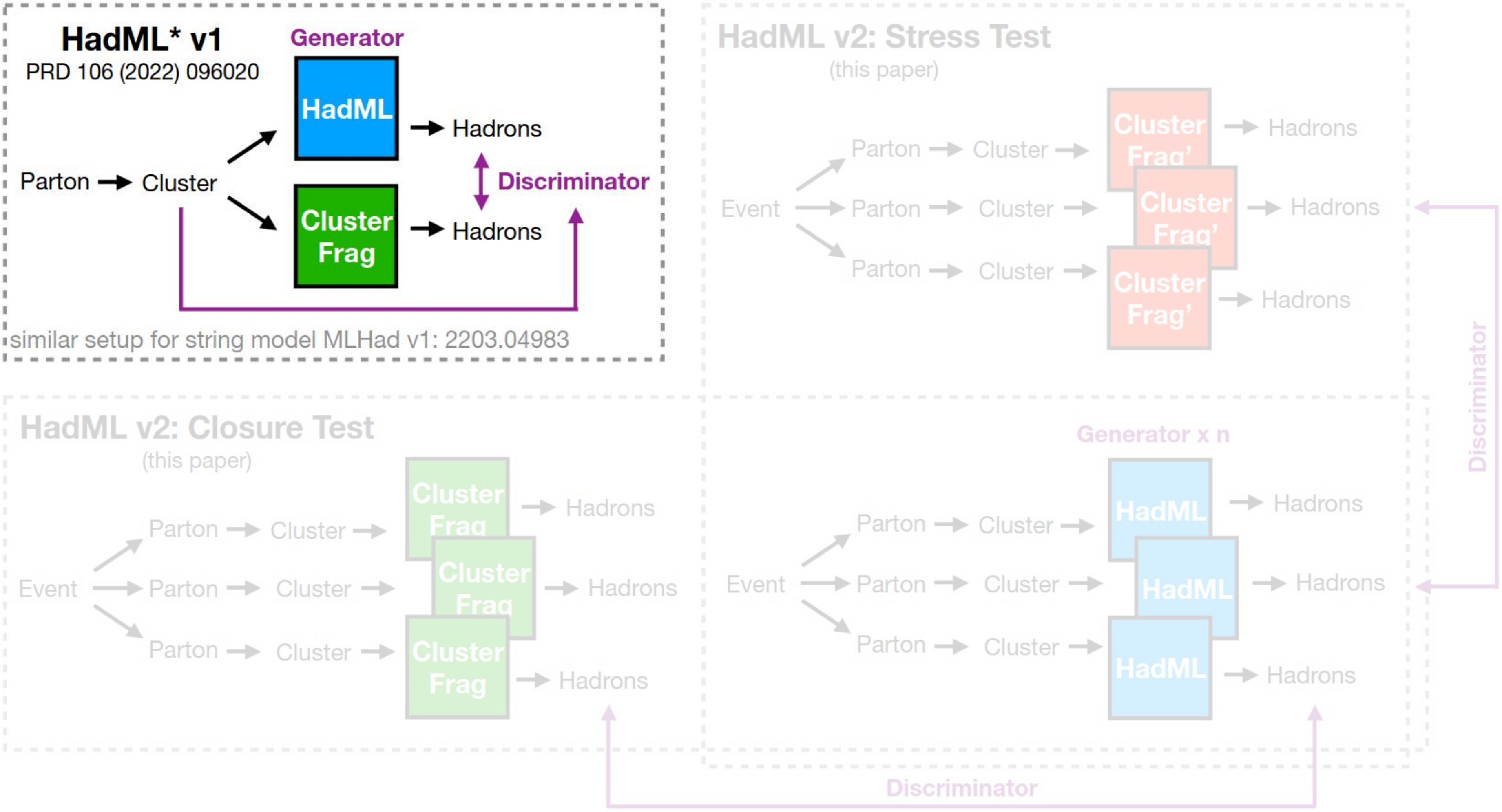




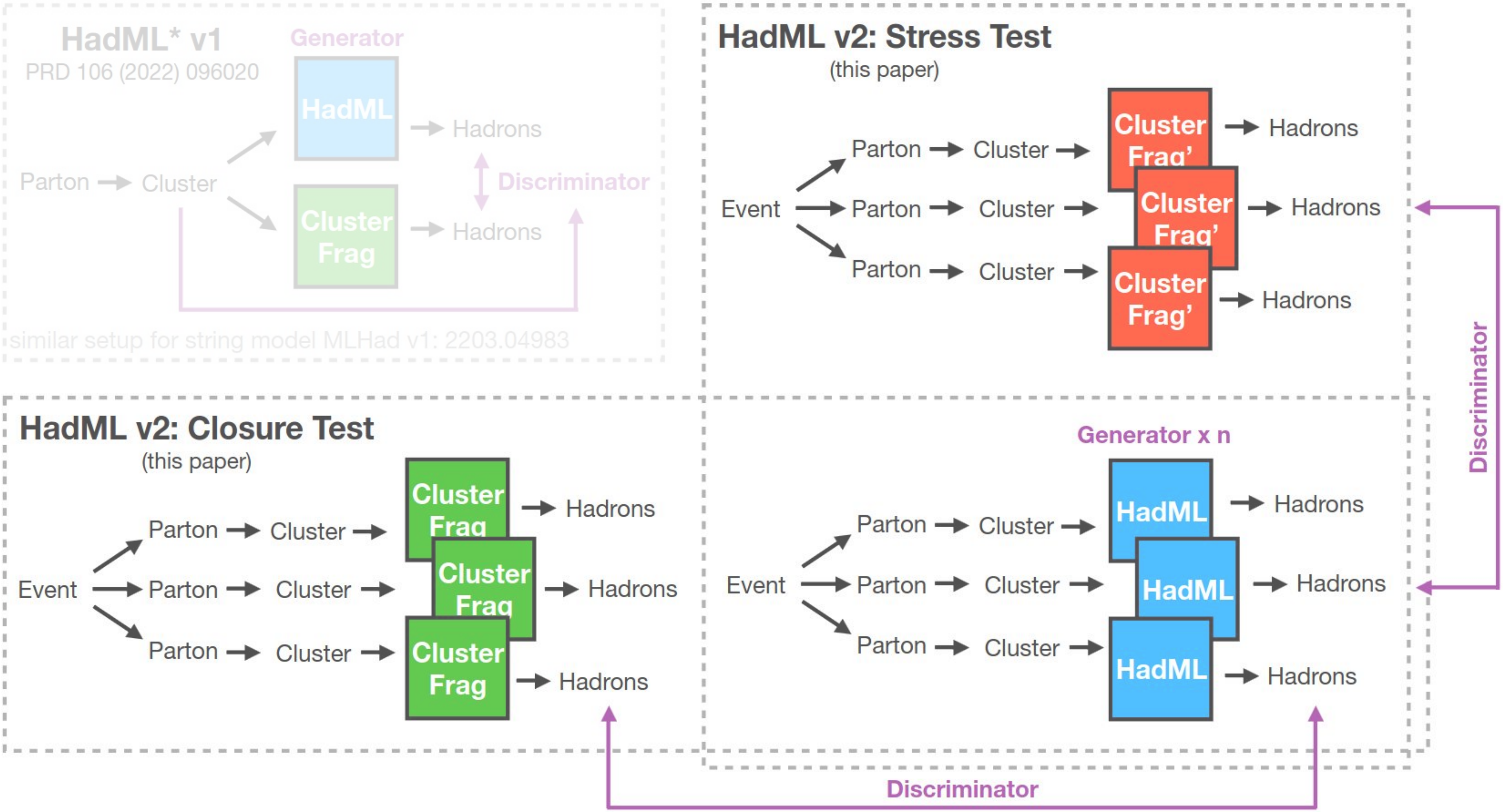


N.B. we have trained on H7, so we don't expect to be any better than it at modeling the data.

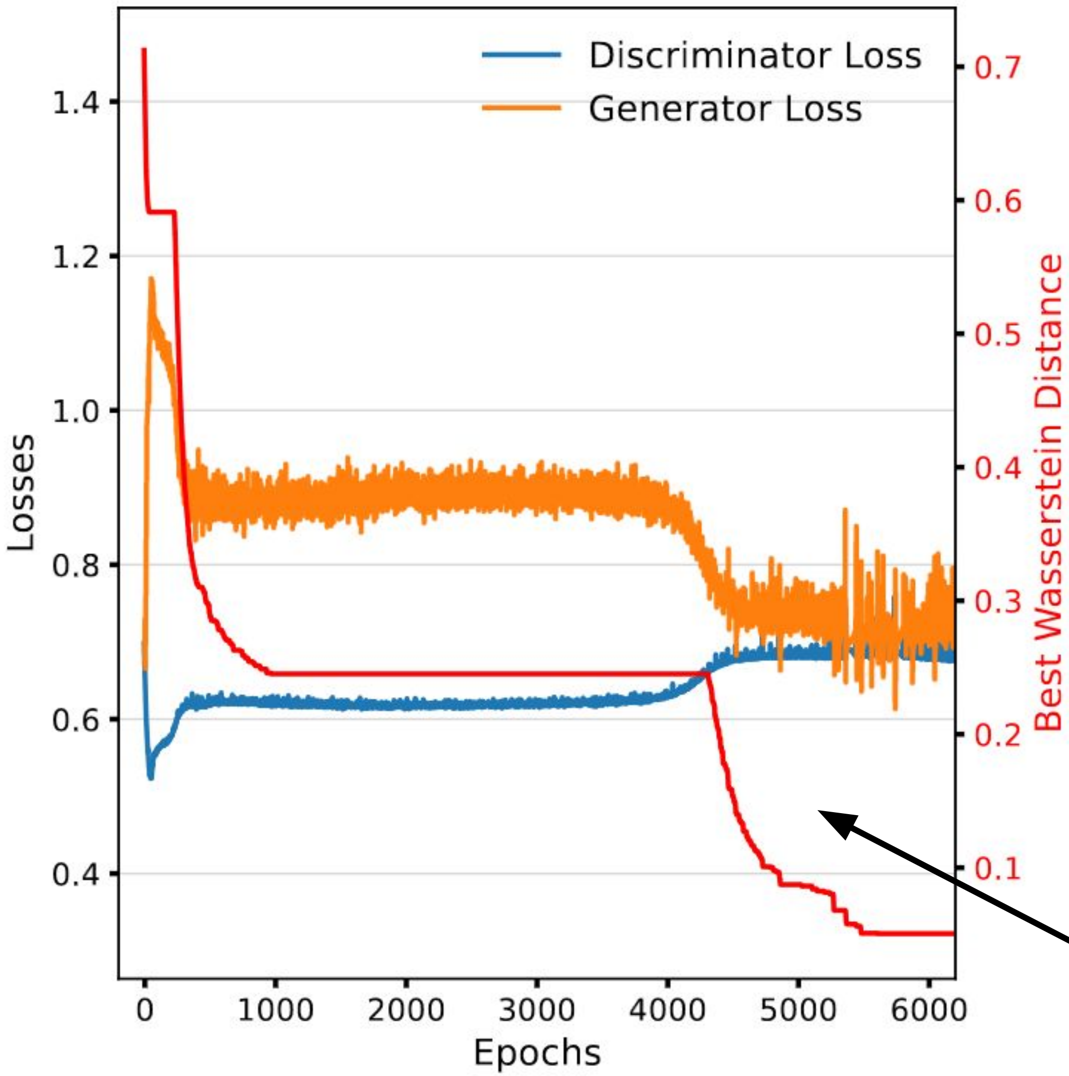
Road map for today



Road map for today



Protocol for fitting a deep generative hadronization model in a realistic data setting, where we only have access to a set of hadrons in data.

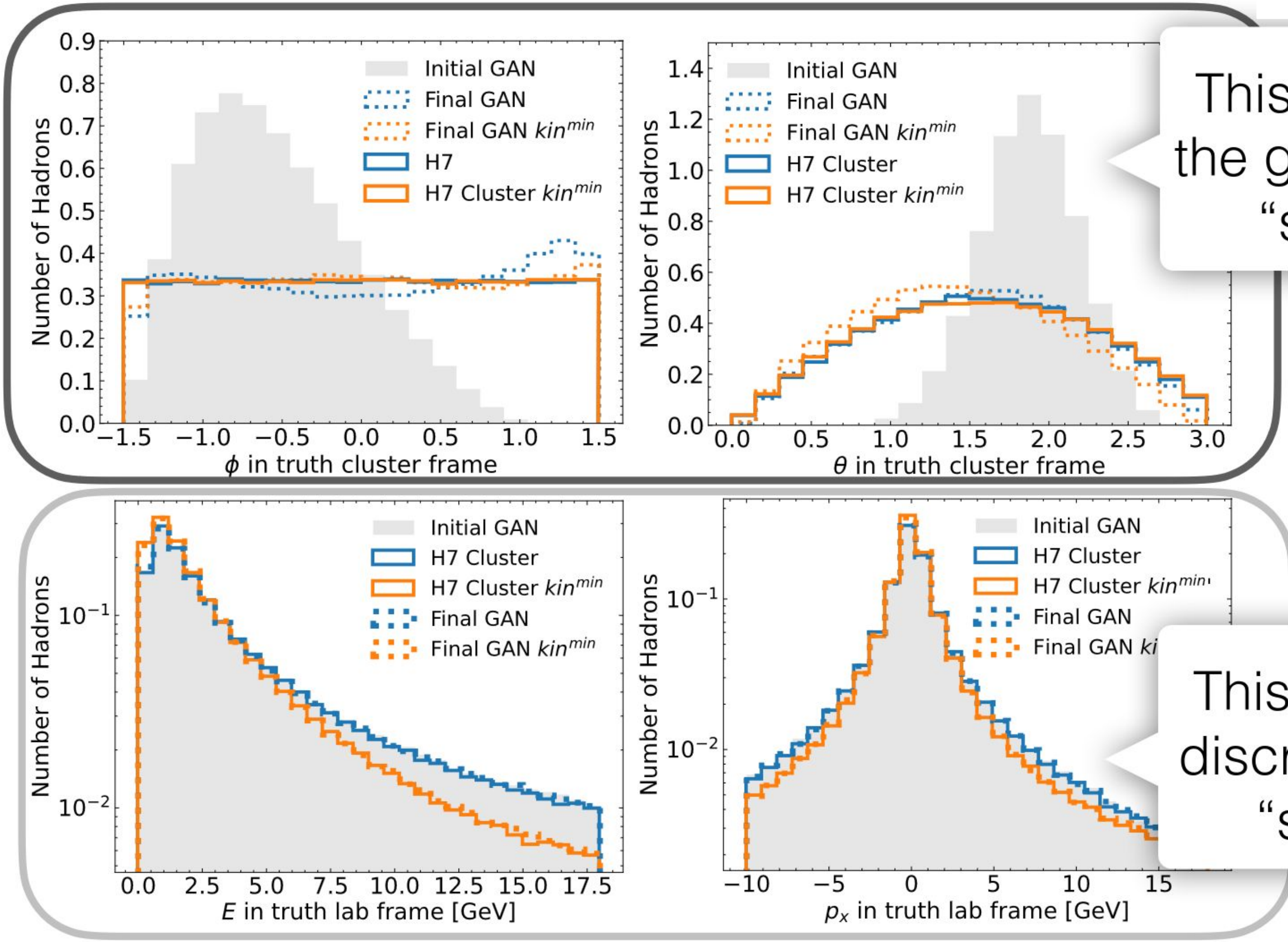


Now, the generator is local (per cluster), but the discriminator is global (whole event).

Discriminator is a permutation-invariant architecture called Deep Sets.

Simplification only Pions

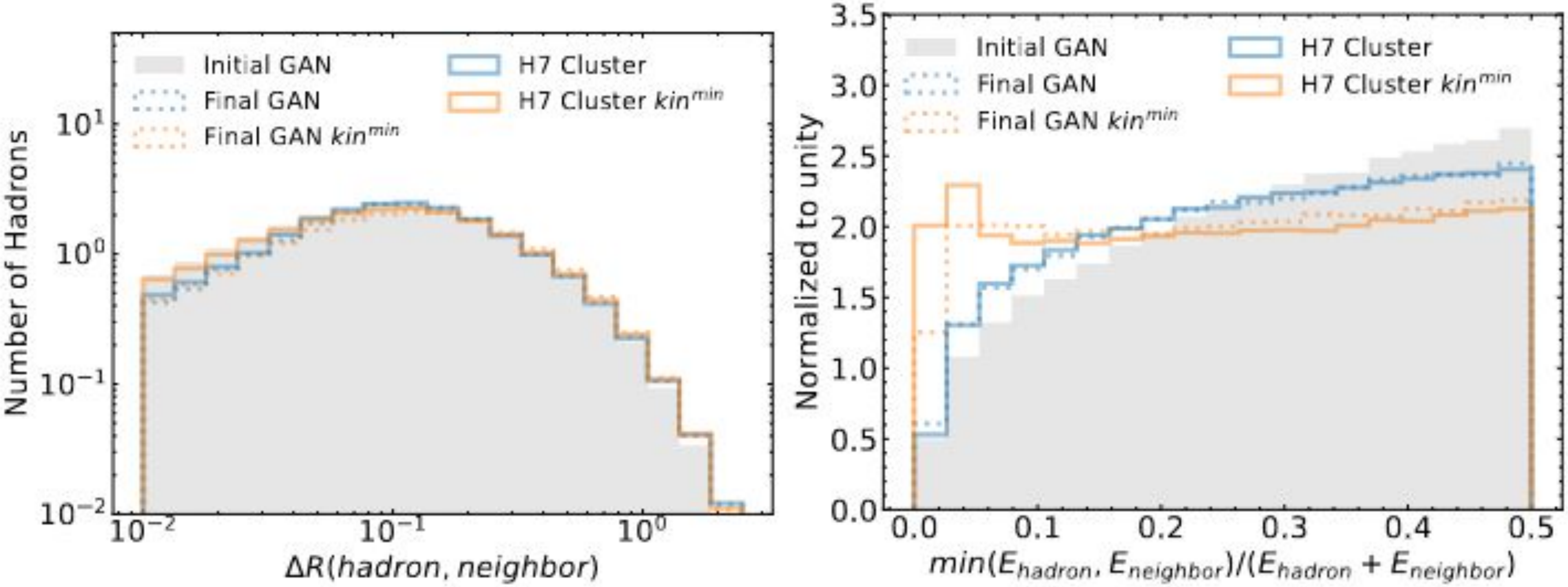
Still works !



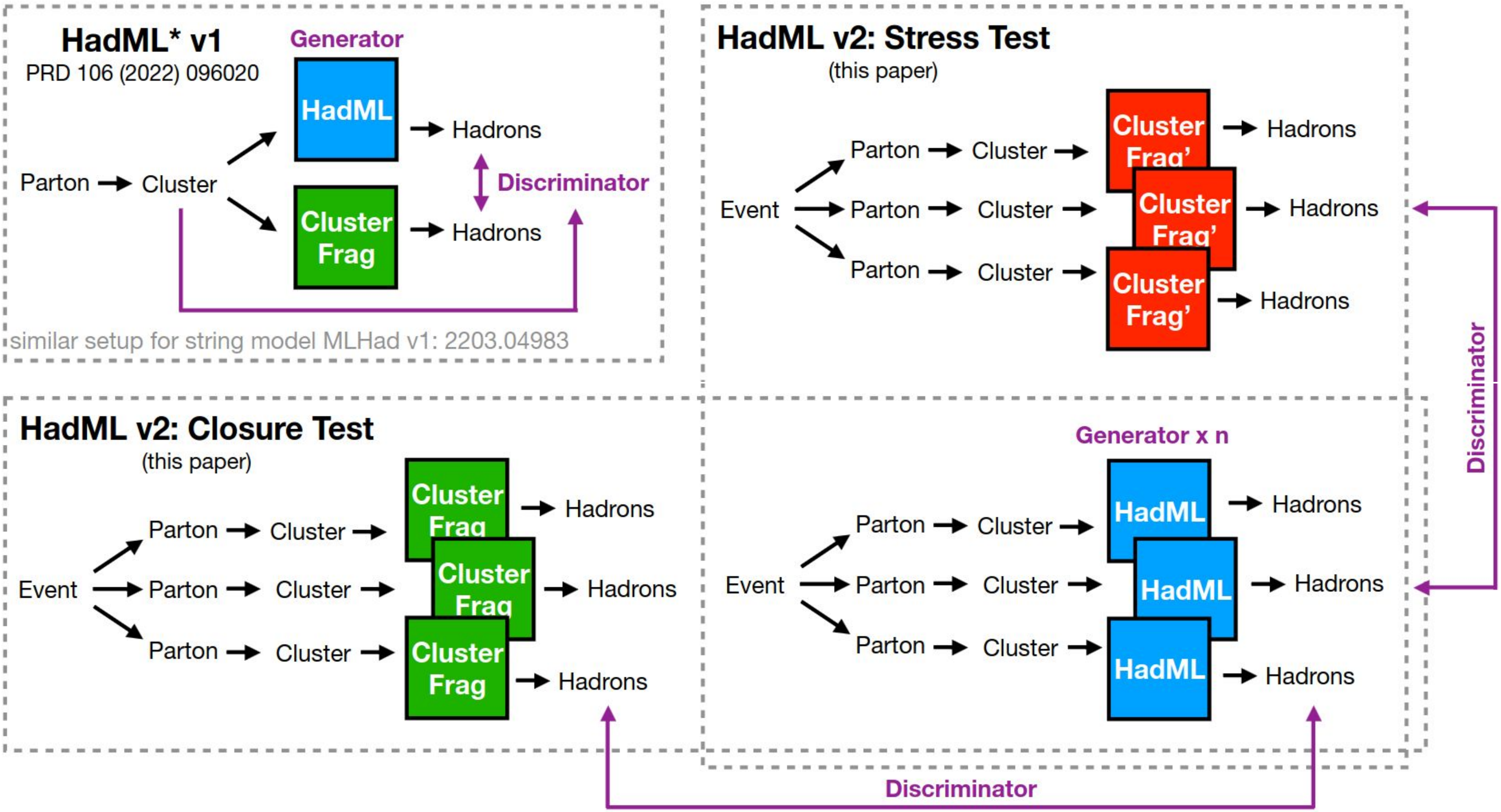
This is what the generator “sees”

This is what discriminator “sees”

Performance: going beyond inputs and outputs



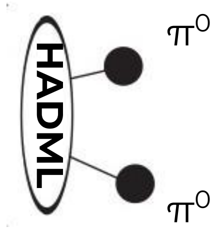
MINIMAL $\Delta R^2 = \Delta\phi^2 + \Delta\eta^2$



A key advantage of this fitting protocol over other methods is that it can accommodate unbinned and high-dimensional inputs.

The approach could also be used to tune (without binning) data to a parametric physics model (for example cluster) as well. However, this would require making the cluster model differentiable.

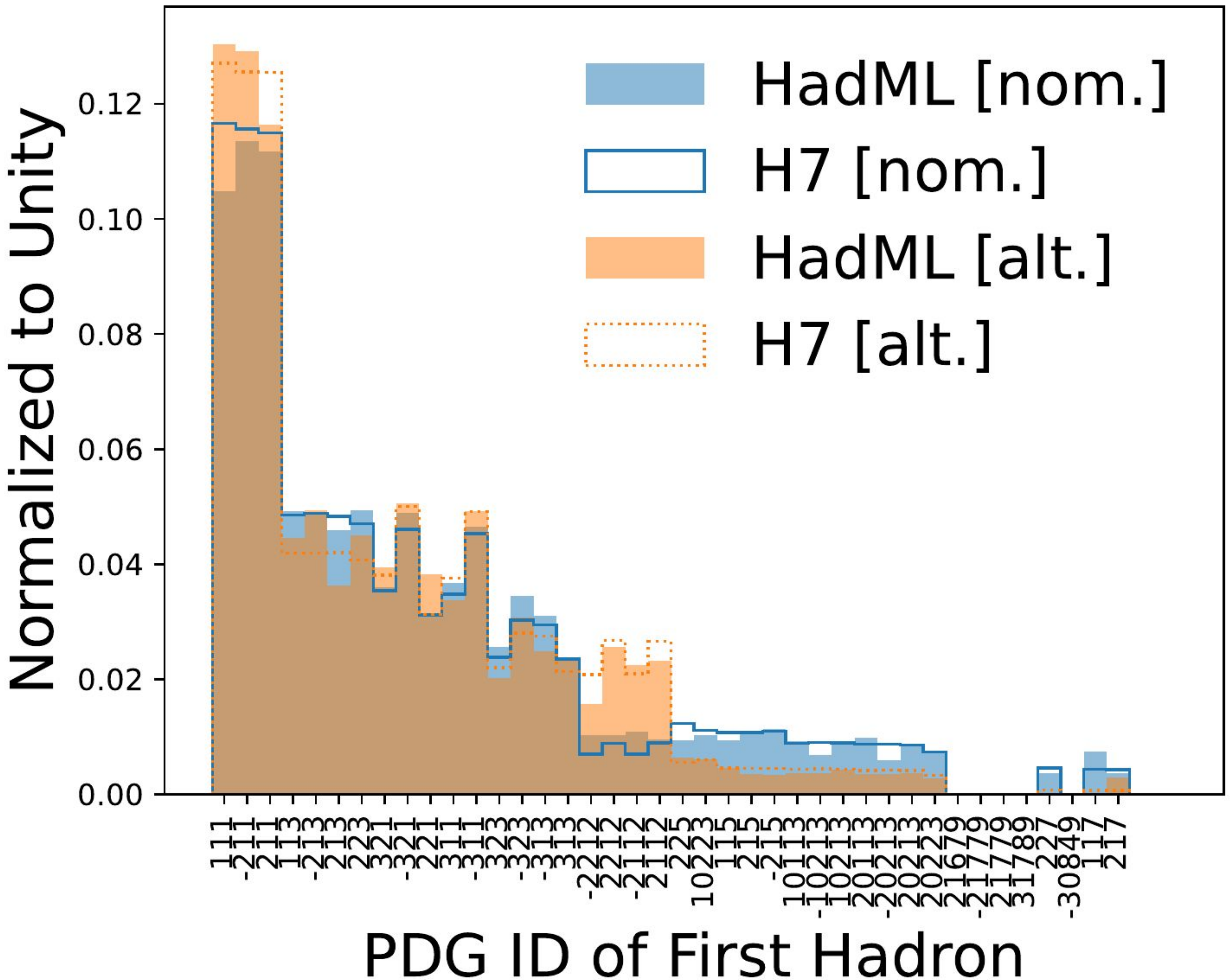
- For HADML, we have made significant progress, but there are still multiple steps to build and tune a full-fledged hadronization model.



What is next?

- Number of technical and methodological step needed:

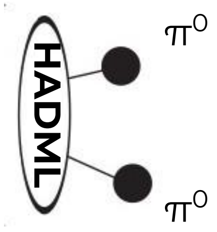
    ➔ Directly accommodate multiple hadron species with their relative probabilities


HadML [nom.]
H7 [nom.]
HadML [alt.]
H7 [alt.]
Normalized to Unity
0.00
0.02
0.04
0.06
0.08
0.10
0.12
PDG ID of First Hadron

Outlook

- For HADML, we have made significant progress, but there are still multiple steps to build and tune a full-fledged hadronization model.

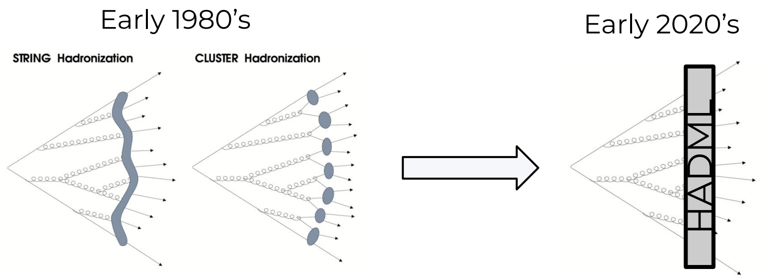
- HADML is naturally suited for GPUs



What is next?

- Number of technical and methodological step needed:
    - ➔ Directly accommodate multiple hadron species with their relative probabilities
    - ➔ Include heavy clusters (so far done by Herwig)
    - ➔ Hyperparameter optimization, including the investigation of alternative generative models
    - ➔ More flexible model with a capacity to mimic the cluster or string models and beyond.
    - ➔ Tune to the LEP data

There is still a multi-year program ahead of us, but it will be worth it!



So Stay tuned!

A postdoc in ML/HEP position

JAGIELLONIAN UNIVERSITY
IN KRAKÓW

If you are interested please contact me:
andrzej.siodmok@cern.ch

HadML v1

The loss function:

$$L = -\sum_{\lambda \sim \textsc{Herwig},\, z \sim p(z)} \left(\log\left(D\left(\tau\left(\lambda\right)\right)\right) + \log\left(1 - D\left(G\left(z, \lambda\right)\right)\right)\right)$$

HadML v2

The discriminator function is modified, we parameterize is as a Deep Sets model

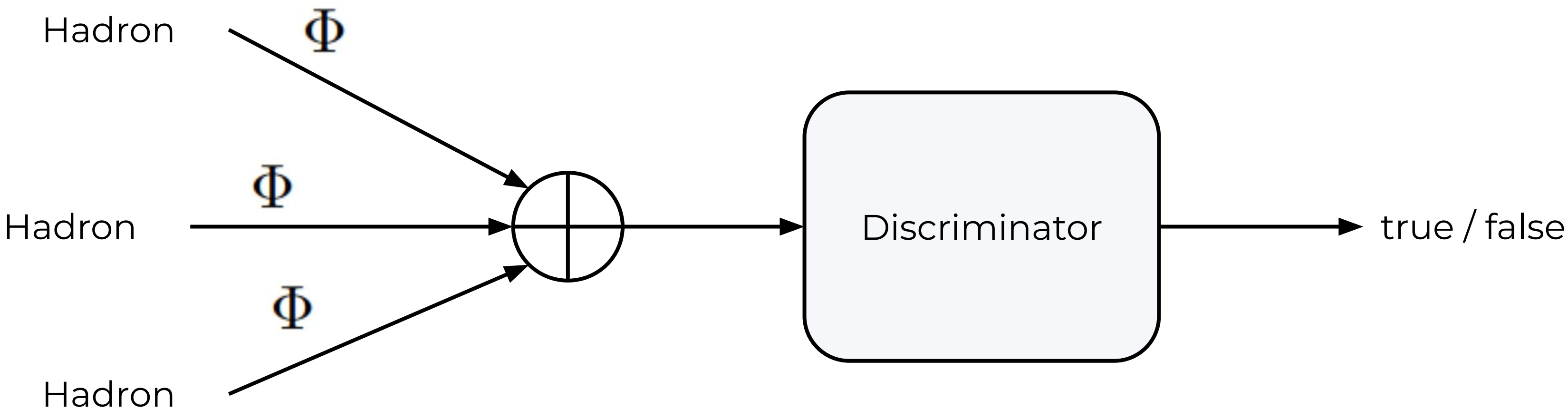
$$D_E\left(x\right) = F\left(\frac{1}{n}\sum_{i=1}^{n} \Phi\left(h_i, \omega_{D_\Phi}\right), \omega_F\right)$$

invariant under permutations of hadrons

Φ embeds a set of hadrons into a fixed-length latent space and F acts on the average

$$L = -\sum_{x \sim \text{data}} \log\left(D_E\left(x\right)\right) - \sum_{\{G\} \sim \textsc{Herwig},\, z \sim p(z)} \log\left(1 - D_E\left(\{G\left(z, \lambda\right)\}\right)\right)$$

The approach could also be used to fit (without binning) data to a parametric physics model (for example cluster) as well. However, this would require making the cluster model differentiable.



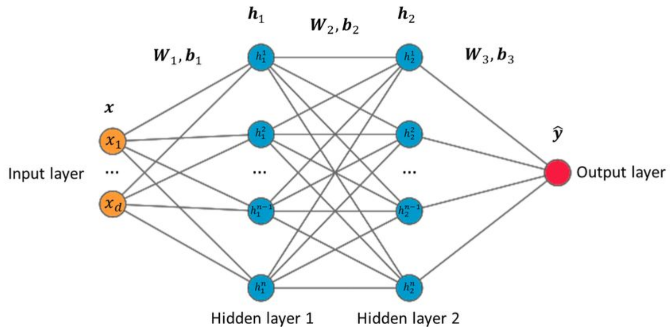
The discriminator function is modified, we parameterize is as a Deep Sets model

$$D_E(x) = F\left(\frac{1}{n}\sum_{i=1}^{n}\Phi\left(h_i, \omega_{D_\Phi}\right), \omega_F\right)$$

invariant under permutations of hadrons

Architecture: conditional GAN

Generator and the Discriminator are composed of two-layer perceptron
(each a fully connected, hidden size 256, a batch normalization layer, LeakyReLU activation function)



Generator

Input

Cluster $(E,\ p_x,\ p_y,\ p_z)$ and 10 noise features sampled from a Gaussian distribution

Output (in the cluster frame)

ϕ - polar angle
θ - azimuthal angle

we reconstruct the four vectors of the two outgoing hadrons

Discriminator

Input

ϕ and θ labeled as signal (generated by Herwig) or background (generated by Generator)

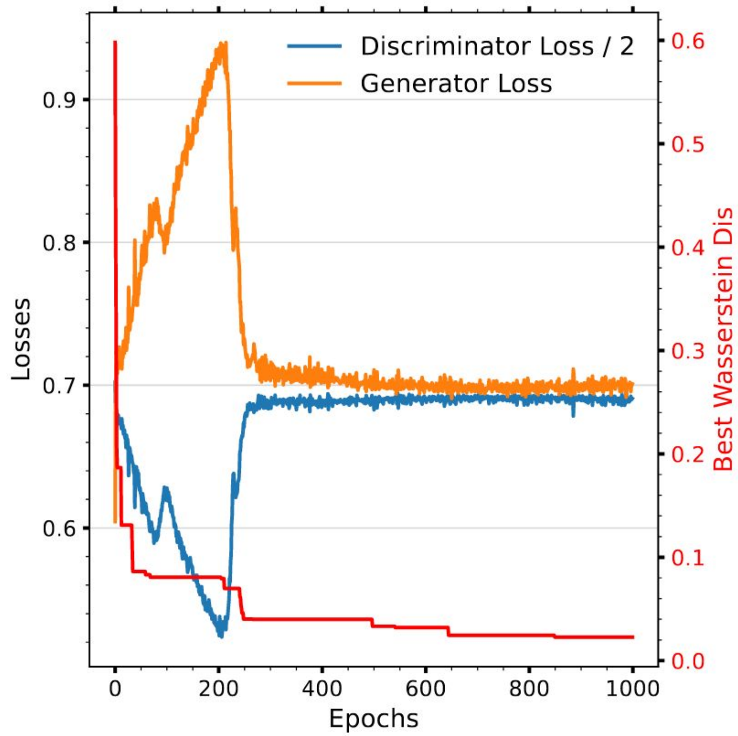
Output

Score that is higher for events from Herwig and lower for events from the Generator

Training

- **Data normalization:**
  cluster's four vector and angular variables are scaled to be between -1 and 1 (tanh activation function as the last layer of the Generator)

- **Discriminator** and the **Generator** are trained separately and alternately by two independent Adam optimizers with a learning rate of 10^{-4}, for 1000 epochs



- **The best model** for events with partons of Pert = 0, is found at the epoch 849 with a total Wasserstein distance of 0.0228.

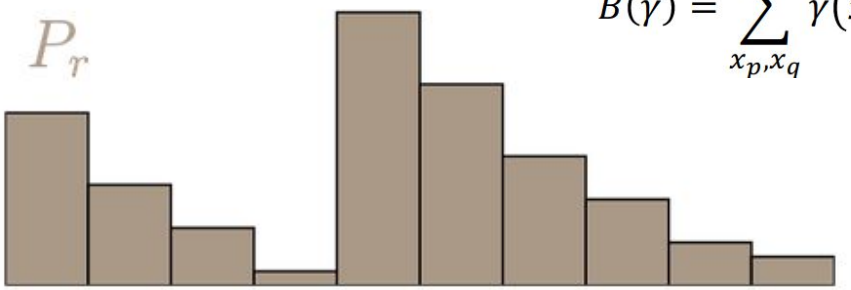
The Wasserstein distance

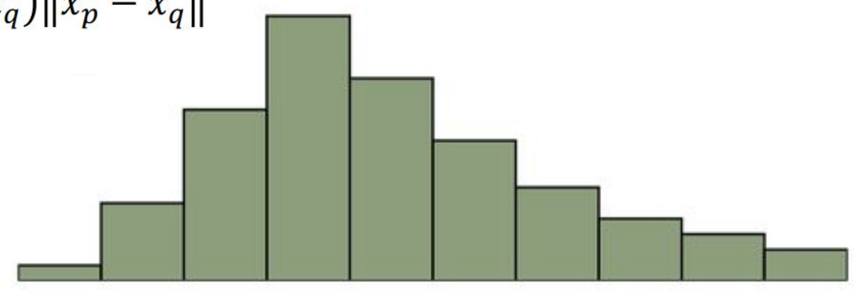
- For discrete probability distributions, the Wasserstein distance is called the earth mover's distance (EMD):
- EMD is the minimal total amount of work it takes to transform one heap into the other.

$$W(P,Q) = \min_{\gamma \in \Pi} B(\gamma)$$

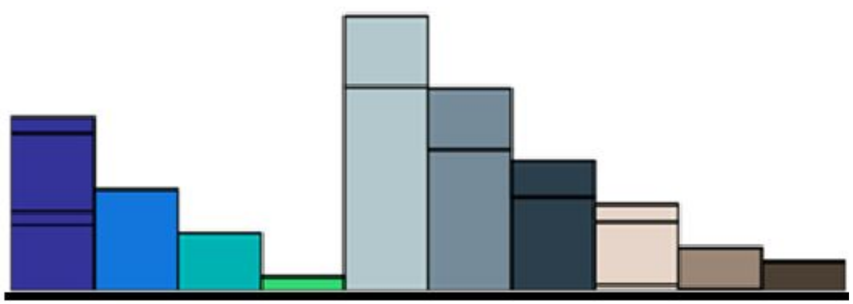
- Work is defined as the amount of earth in a chunk times the distance it was moved.

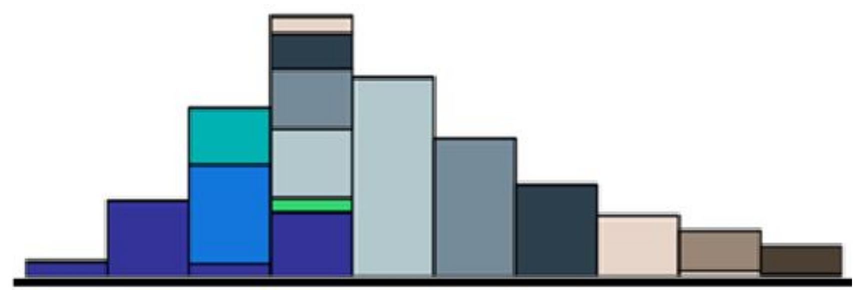
$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$



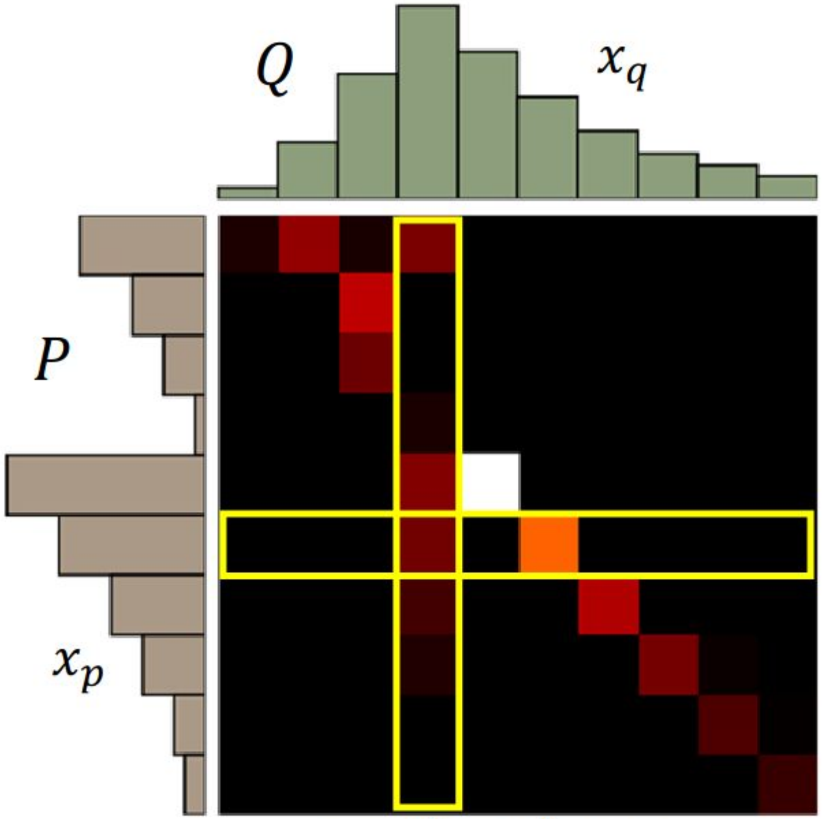


Best "moving plans" of this example





Wasserstein distance



moving plan γ

All possible plan Π

A "moving plan" is a matrix

The value of the element is the amount of earth from one position to another.

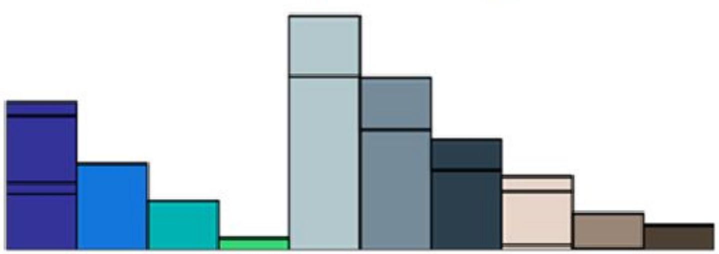
Average distance of a plan γ:

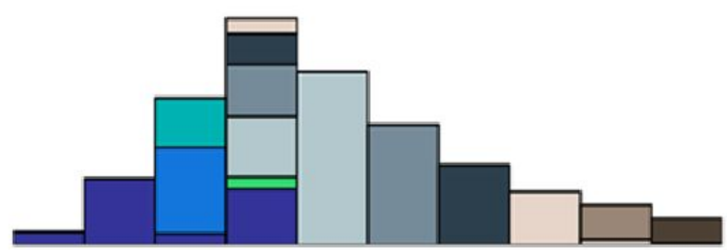
$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$

Earth Mover's Distance:

$$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$

The best plan





Minimax Loss

In the paper that introduced GANs, the generator tries to minimize the following function while the discriminator tries to maximize it:

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

In this function:

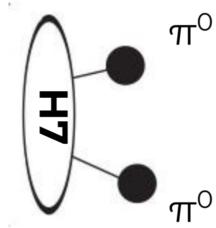
- `D(x)` is the discriminator's estimate of the probability that real data instance x is real.
- E_x is the expected value over all real data instances.
- `G(z)` is the generator's output when given noise z.
- `D(G(z))` is the discriminator's estimate of the probability that a fake instance is real.
- E_z is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances G(z)).
- The formula derives from the cross-entropy between the real and generated distributions.

The generator can't directly affect the `log(D(x))` term in the function, so, for the generator, minimizing the loss is equivalent to minimizing `log(1 - D(G(z)))`.

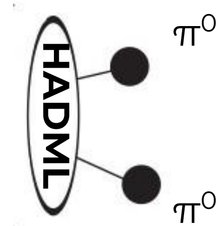
Performance: Energy of the collisions

Low-level Validation
(**beyond** training data **different energy**)

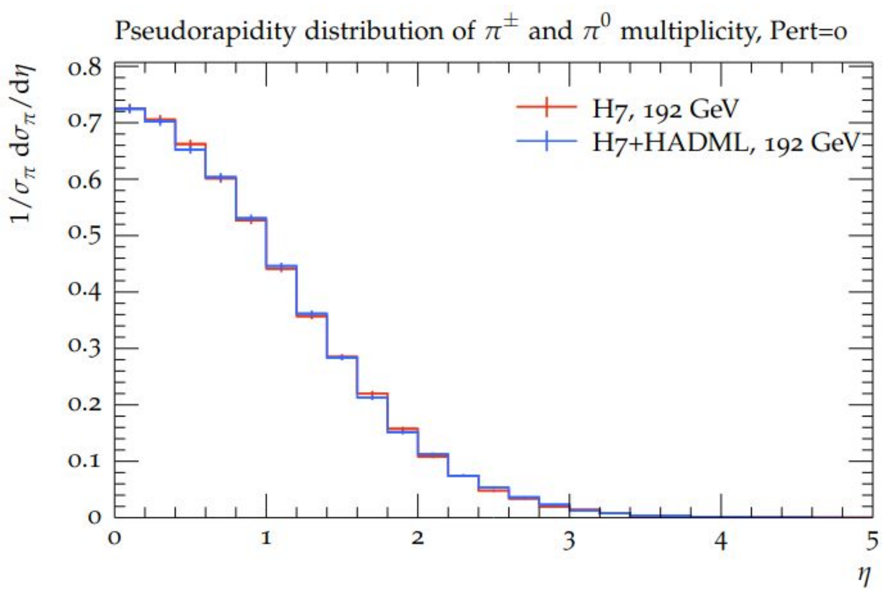
e^+e^- collisions at $\boxed{\sqrt{s} = 192\ \mathrm{GeV}}$

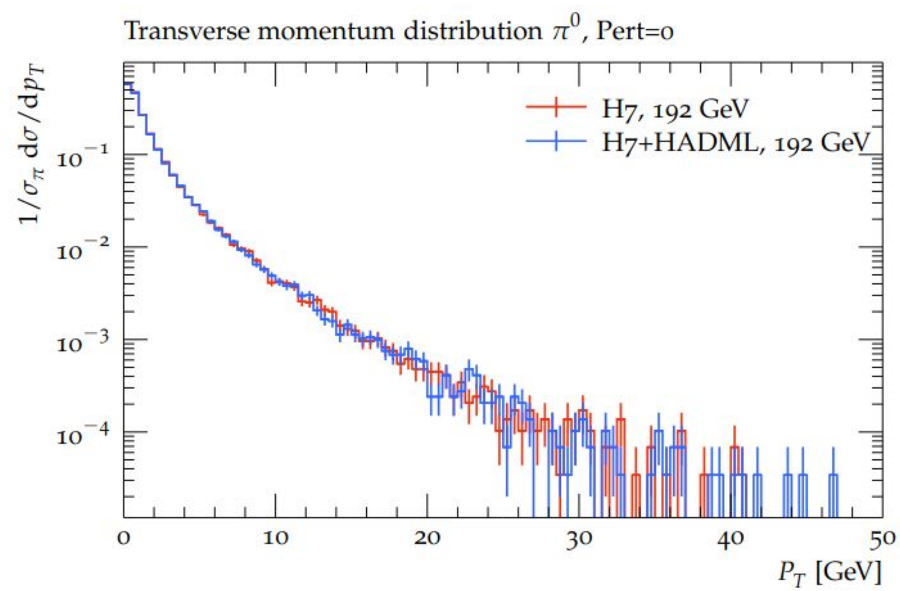


VS



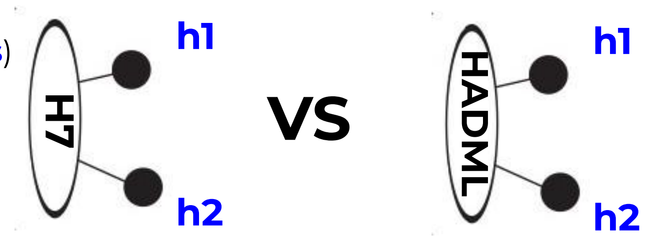
π^0 kinematic variables





Low-level Validation
(**beyond** training data **different hadrons**)

e^+e^- collisions at $\sqrt{s} = 91.2$ GeV



h kinematic variables

As a crude "full" model, we simply take the PIDs from Herwig and the kinematics from the GAN.

Kaons

Lambda