HADML - towards a deep learning model for hadronization

Andrzej Siódmok

in collaboration with

Aishik Ghosh

Xiangyang Ju

Ben Nachman

based on Physical Review D [arXiv: 2203.12660]

JAGIELLONIAN UNIVERSITY IN KRAKÓW

UCZELNIA BADAWCZA INICJATYWA DOSKONAŁOŚCI

NARODOWE CENTRUM NAUKI

NCN: 2019/34/E/ST2/00457

H7

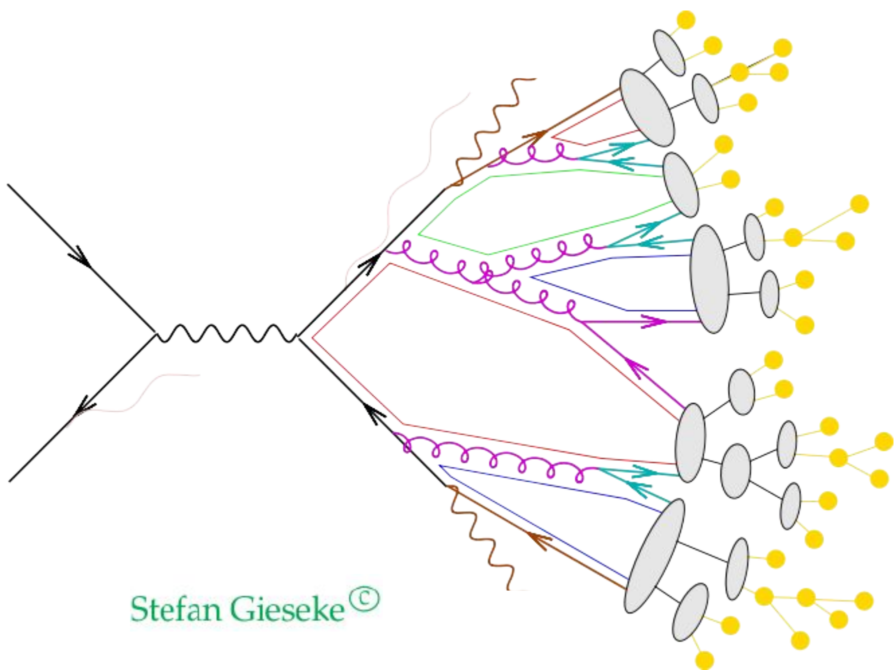
MCnet

Motivation - Monte Carlo Event Generators (MCEG)

QCD correctly describes strong interactions in each energy range but its complex mathematical structure makes it very difficult to obtain precise predictions (Millennium Prize Problem $1,000,000)

High energy
- **perturbative QCD**
- **in theory we know what to do**
- **in practice very difficult**

Low energy
- **non-perturbative QCD**
- **we don't know what to do**
- **phenomenological models (with many free parameters)**

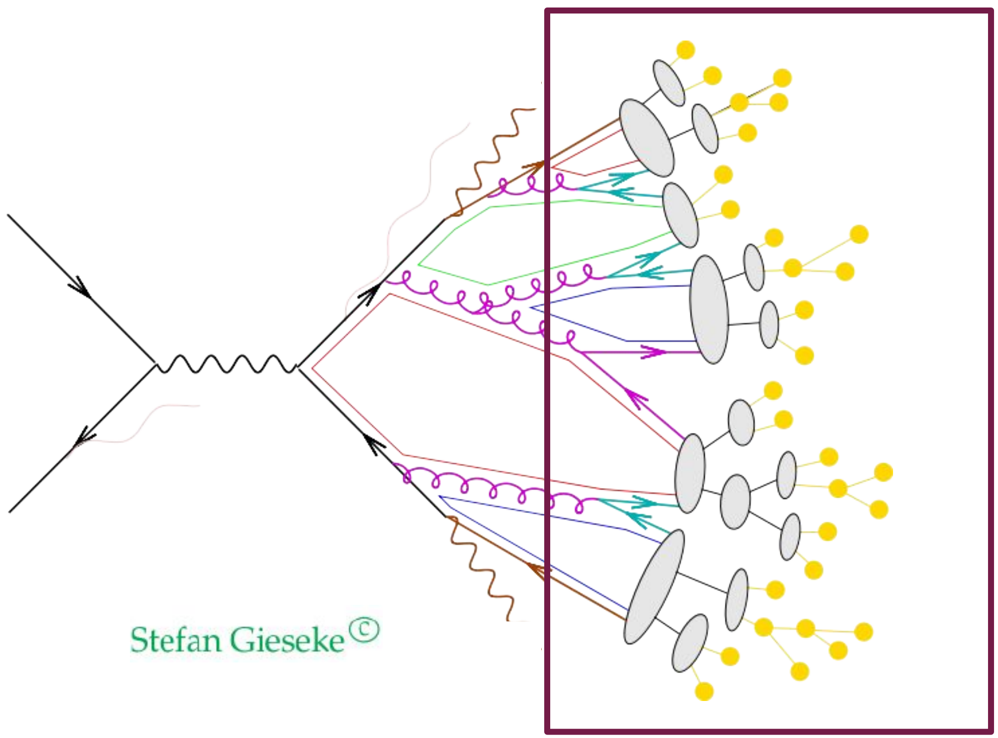


Motivation - Monte Carlo Event Generators (MCEG)

QCD correctly describes strong interactions in each energy range but its complex mathematical structure makes it very difficult to obtain precise predictions (Millennium Prize Problem $1,000,000)

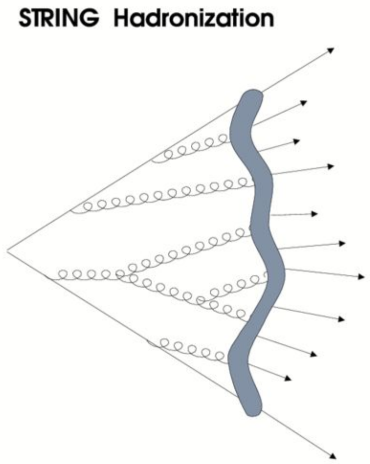
High energy
- **perturbative QCD**
- **in theory we know what to do**
- **in practice very difficult**

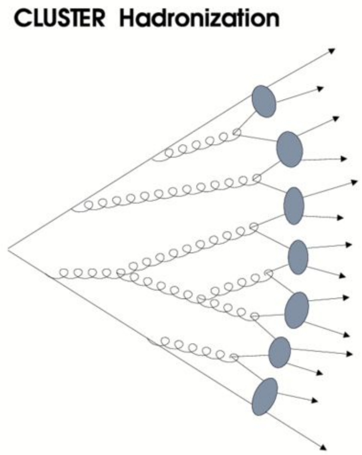
Low energy
- **non-perturbative QCD**
- **we don't know what to do**
- **phenomenological models (with many free parameters)**



Hadronization:
one of the least understood elements of MCEG

Hadronization:



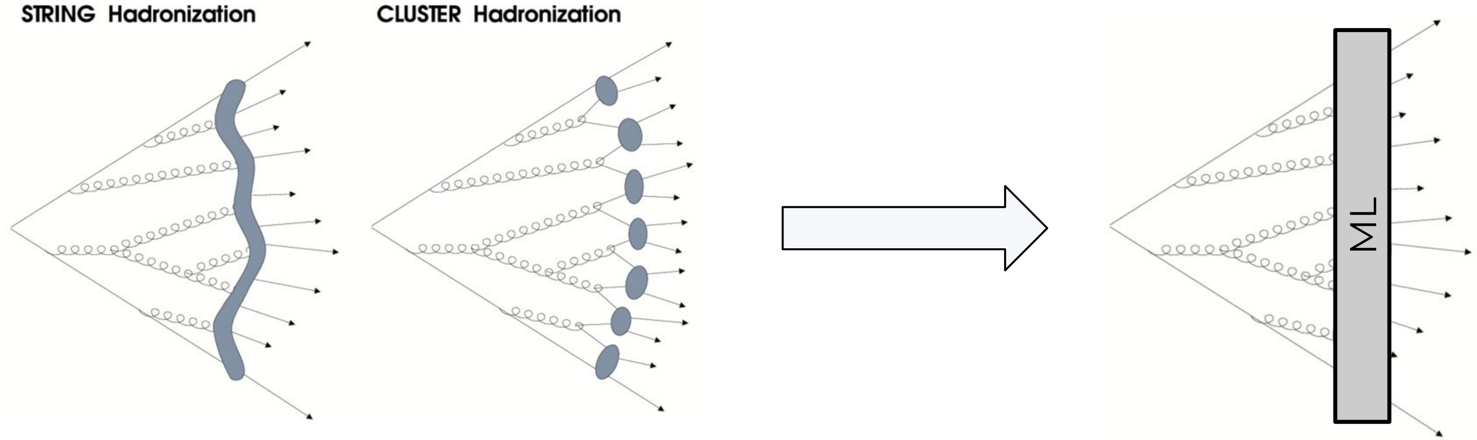


- ➔ Increased control of perturbative corrections ⇒ more often the precision of LHC measurements is limited by MCEG's non-perturbative components, such as hadronization.
- ➔ Hadronization (phenomenological models with many free parameters ~ 30 parameters)

Hadronization:

Early 1980’s
(since then very little development)

Early 2020’s
(lot of progress in ML)



- ➔ Increased control of perturbative corrections ⇒ more often the precision of LHC measurements is limited by MCEG’s non-perturbative components, such as hadronization.
- ➔ Hadronization (phenomenological models with many free parameters ~ 30 parameters)
- ➔ Hadronization is a fitting problem ML is proved to be well suited for such a problems.

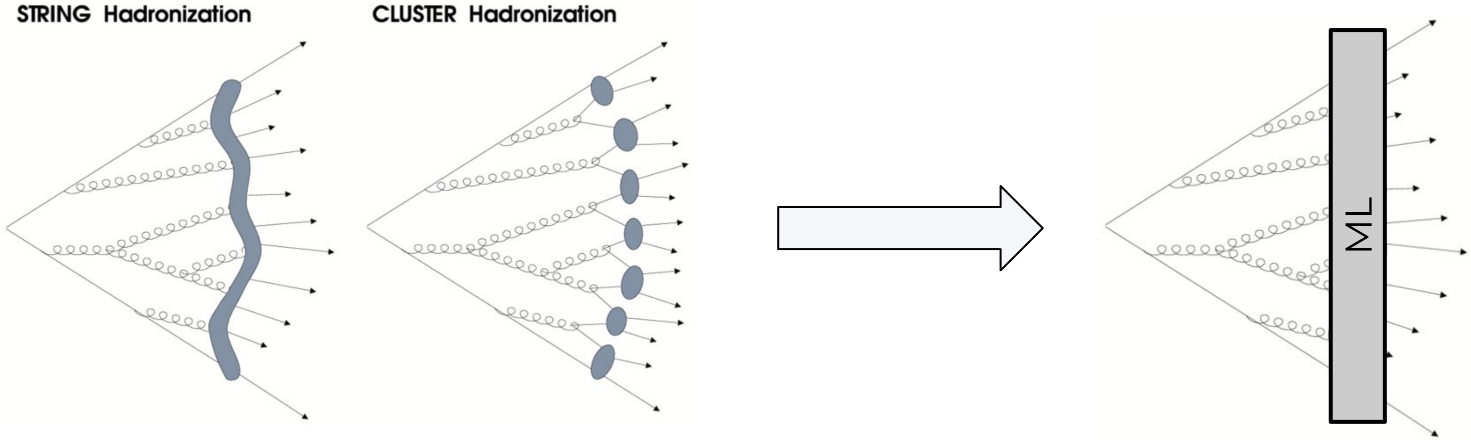
Idea of using Machine Learning (ML) to improve hadronization.

Non-perturbative QCD

Hadronization:

Early 1980's
(since then very little development)

Early 2020's
(lot of progress in ML)



- ➔ Increased control of perturbative corrections ⇒ more often the precision of LHC measurements is limited by MCEG's non-perturbative components, such as hadronization.
- ➔ Hadronization (phenomenological models with many free parameters ~ 30 parameters)
- ➔ Hadronization is a fitting problem ML is proved to be well suited for such a problems.

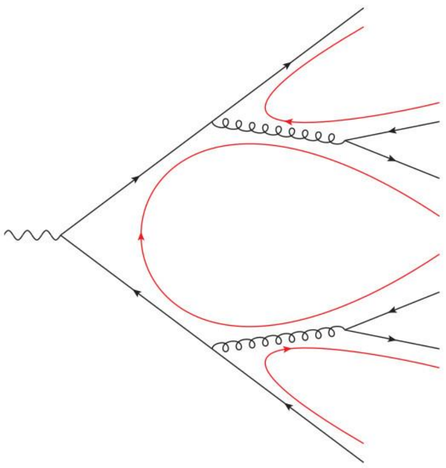
Idea of using Machine Learning (ML) to improve hadronization.

NNPDF used successfully ML to nonperturbative Parton Density Functions (PDF)
Hadronization is closely related to so-called fragmentation functions (FF). Early on, FFs were considered the counterpart of PDFs. While PDFs are understood as probability densities for finding partons, with a given momentum, inside colour-neutral particles, FFs (or hadronization) were understood as probability densities for finding colour-neutral particles from partons.

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:

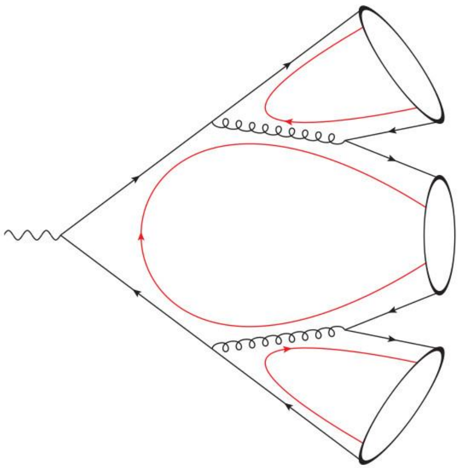


- QCD provide pre-confinement of colour

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:

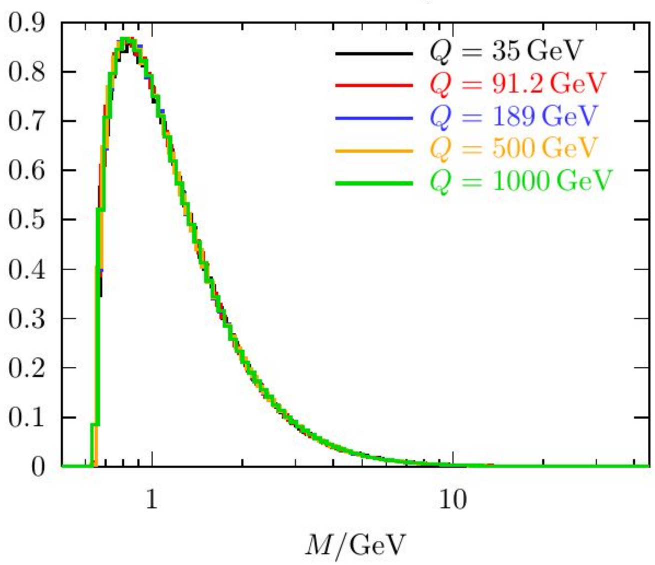


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters

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:

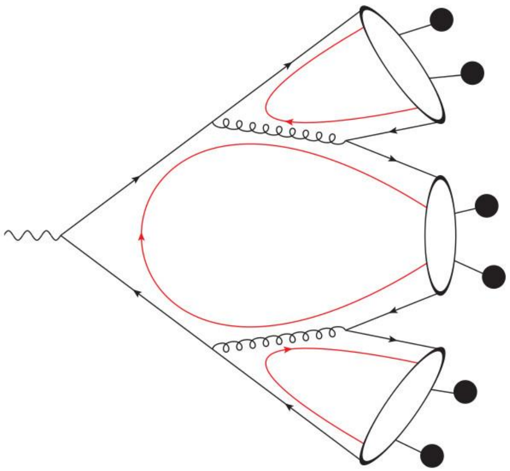


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision

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:

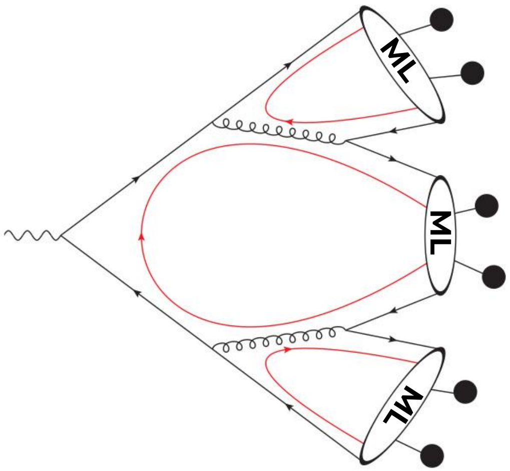


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically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

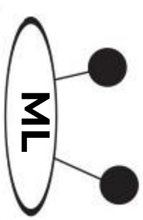
Cluster hadronization model

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

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



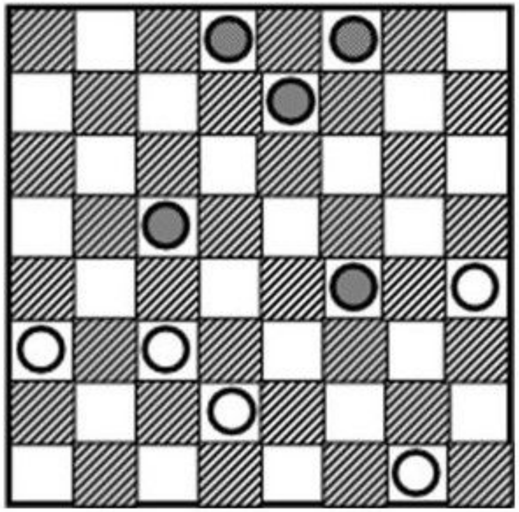
- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically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



- **How?**
  Use Generative Adversarial Networks (**GAN**)

Adversarial Networks

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

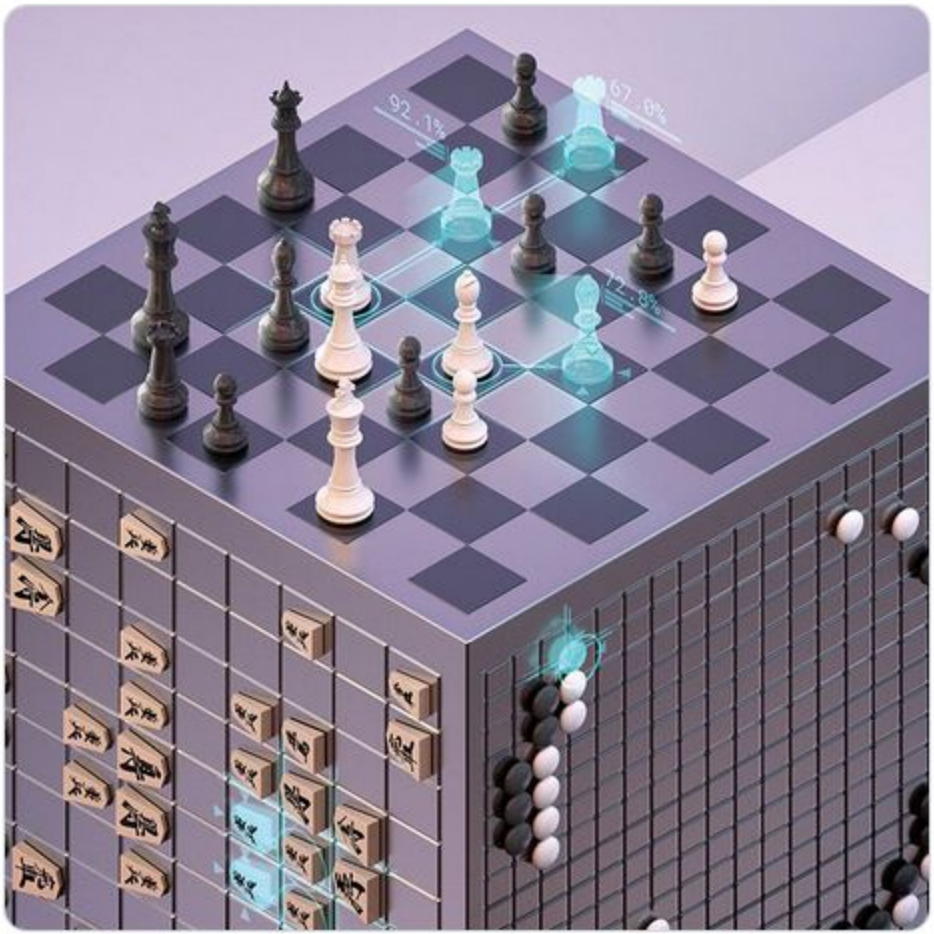


- He popularized the term **"machine learning"** in 1959.
- The program chose its move based on a **minimax** strategy, meaning it made the move assuming that the opponent was trying to optimize the value of the same function from its point of view.
- He also had it play thousands of **games against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

Adversarial Networks

DeepMind @DeepMind · Dec 6, 2018

The full peer-reviewed @sciencemagazine evaluation of #AlphaZero is here - a single algorithm that creatively masters chess, shogi and Go through self-play deepmind.com/blog/alphazero…



Demis Hassabis

CBE FRS FREng FRSA

By playing **games against itself**, AlphaGo Zero surpassed the strength of AlphaGo Lee in three days by winning 100 games to 0.

Adversarial Networks

You Retweeted

OpenAI
@OpenAI

We trained a neural network that solved two problems from the International Math Olympiad. openai.com/blog/formal-ma...

```
theorem imo_longlist_1990_p77
  (a * b + b * c + c * a)^3 ≤
    (a^2 + a * b + b^2) * (b^2 + b * c + c^2) *
    (c^2 + c * a + a^2)
begin
  let u : euclidean_space ℝ (fin 2) := ![a, b],
  let v : euclidean_space ℝ (fin 2) := ![b, c],
  have h₀ := real_inner_mul_inner_self_le u v,
  …
```

19:47 · 02 Feb 22 · Twitter Web App

99 Retweets **32** Quote Tweets **564** Likes

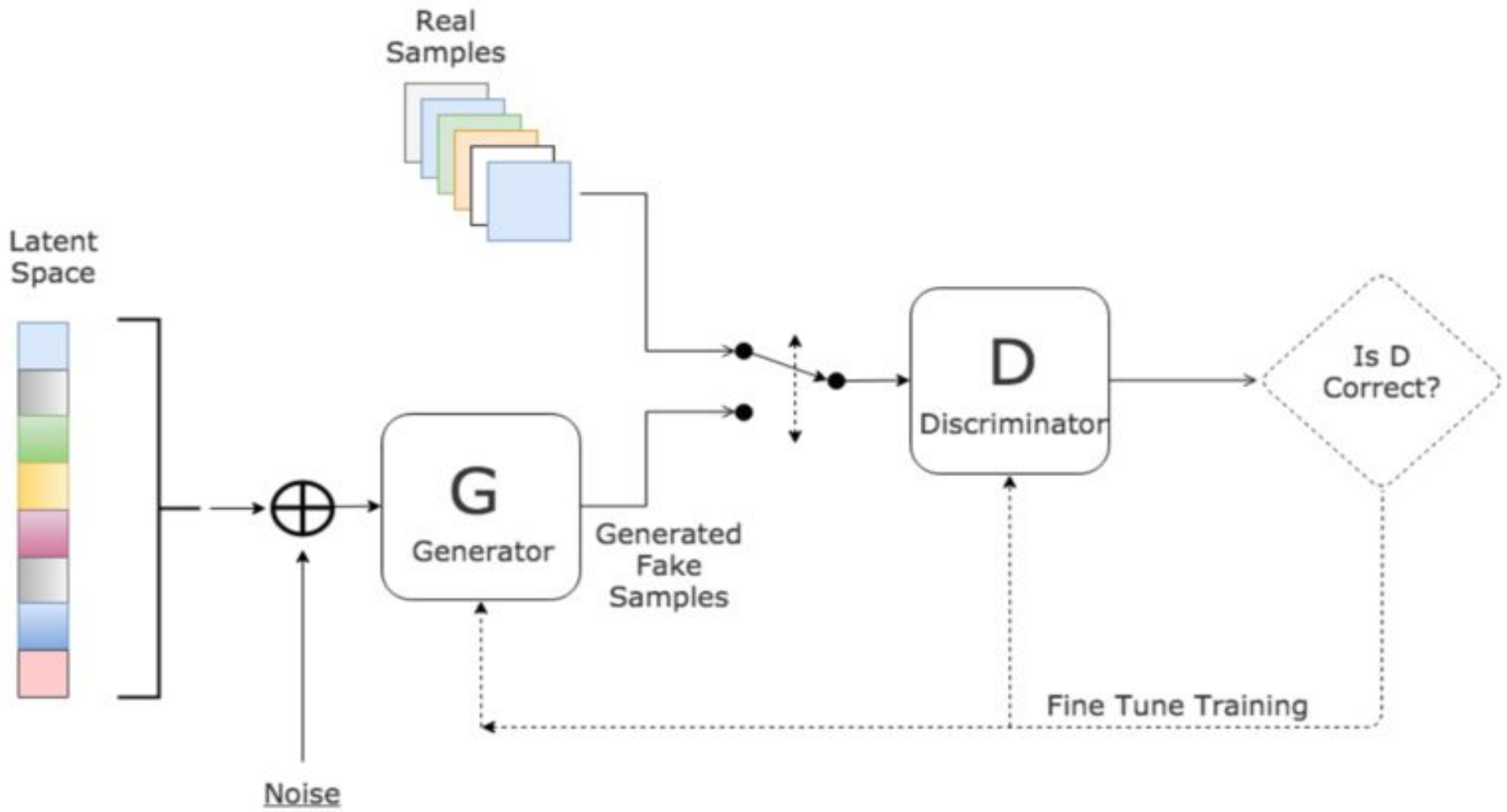
Demis Hassabis
CBE FRS FREng FRSA

Generative Adversarial Network (GAN)

thispersondoesnotexist.com

Generative Adversarial Network (GAN)

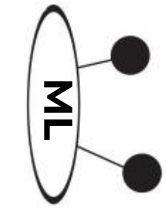
[Goodfellow et al. “Generative adversarial nets”. arxiv:1406.2661]



Towards a Deep Learning Model for Hadronization

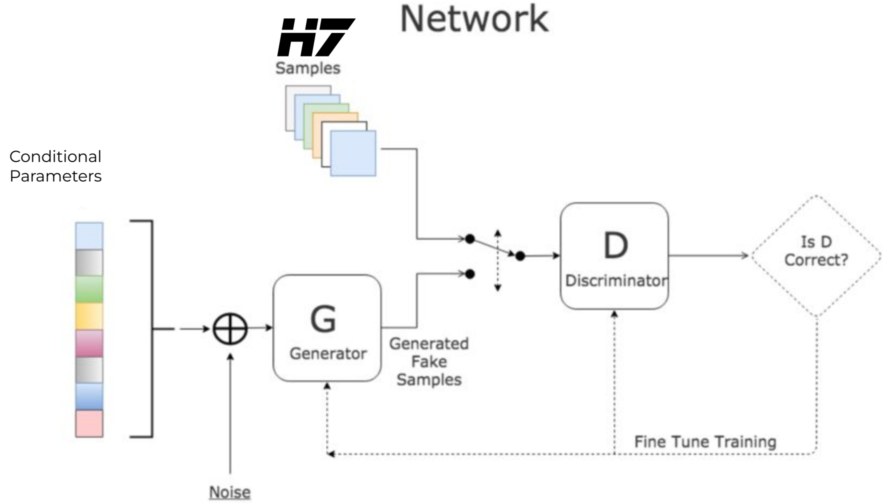
ML hadronization

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



How?

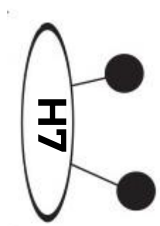
Generative Adversarial Network



Training data:

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

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

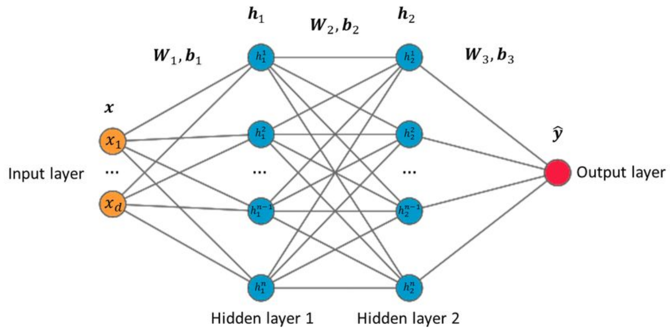


$\pi^0(E,\ p_x,\ p_y,\ p_z)$

$\pi^0(E,\ p_x,\ p_y,\ p_z)$

Pert = 0/1 memory of quarks direction

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

Wasserstein distance

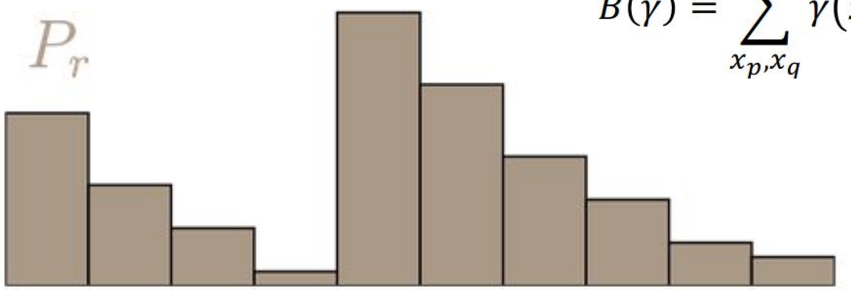
The Wasserstein distance

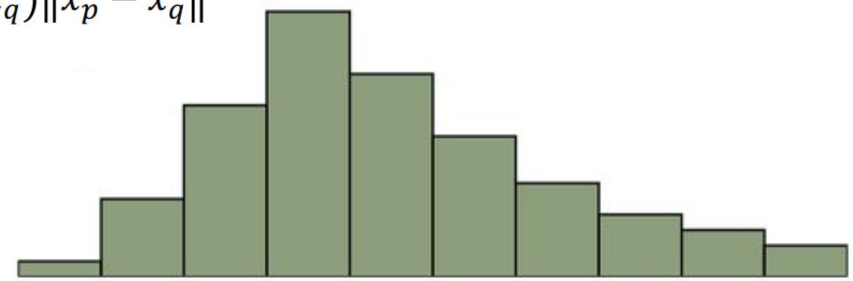
- For discrete probability distributions, the Wasserstein distance is called the earth mover's distance (EMD):
- EMD is the minimal total amount of work it takes to transform one heap into the other.

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

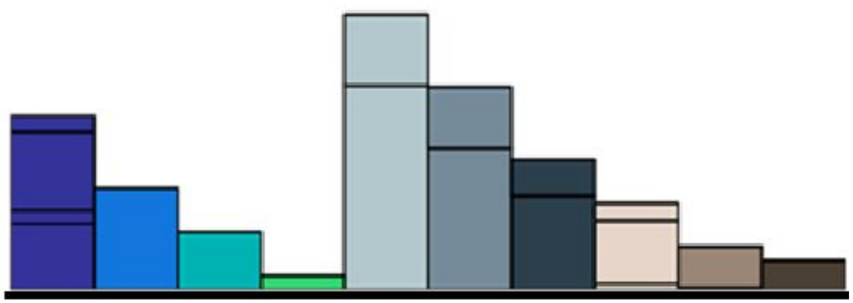
- Work is defined as the amount of earth in a chunk times the distance it was moved.

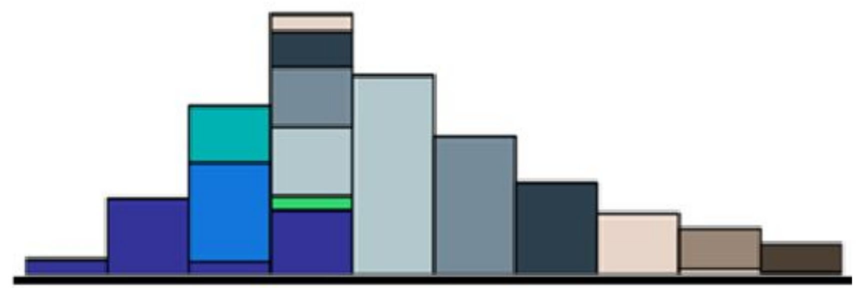
$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \| x_p - x_q \|$$





Best "moving plans" of this example

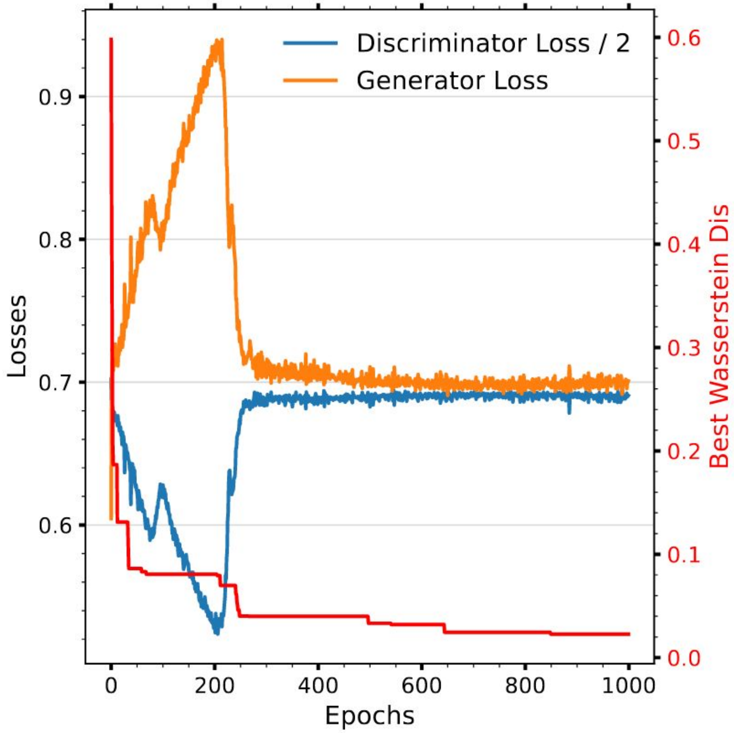




Training

- **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

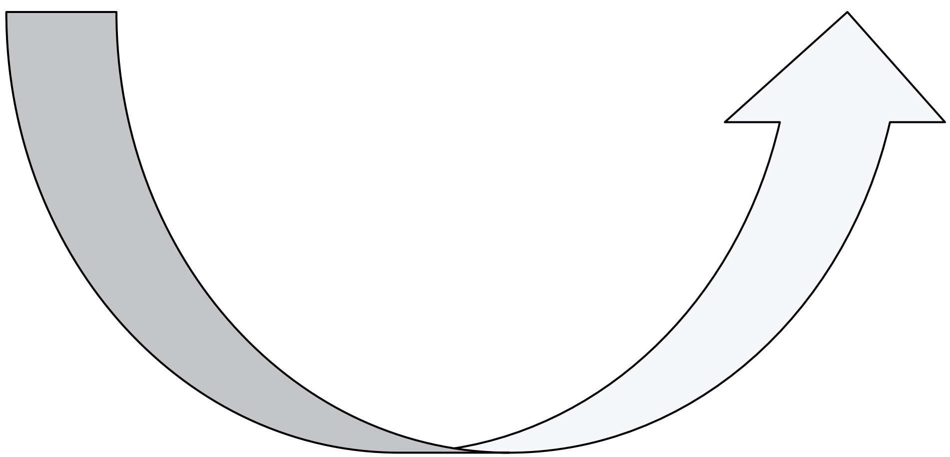


- **The best model** for events with partons of Pert = 0, is found at the epoch 849 with a total Wasserstein distance of 0.0228.

Integration into Herwig

Training

Event generation

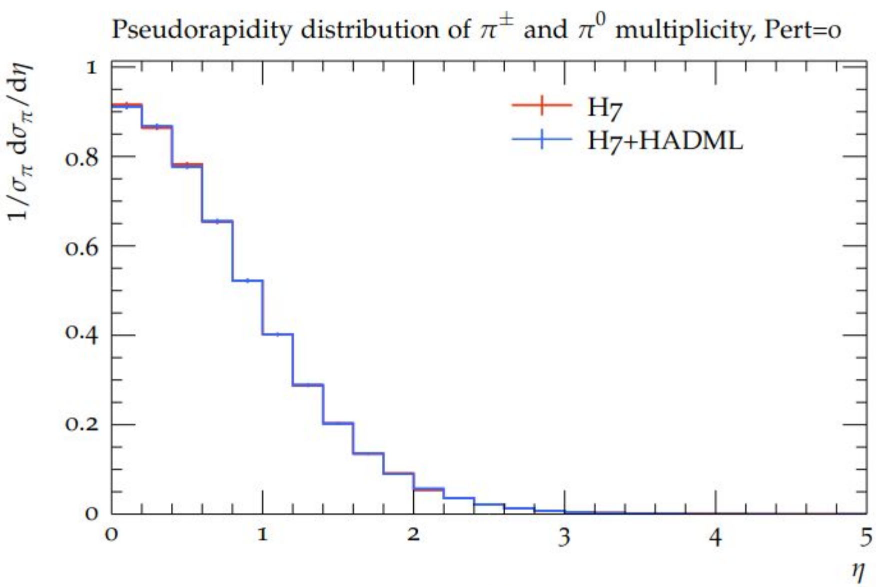


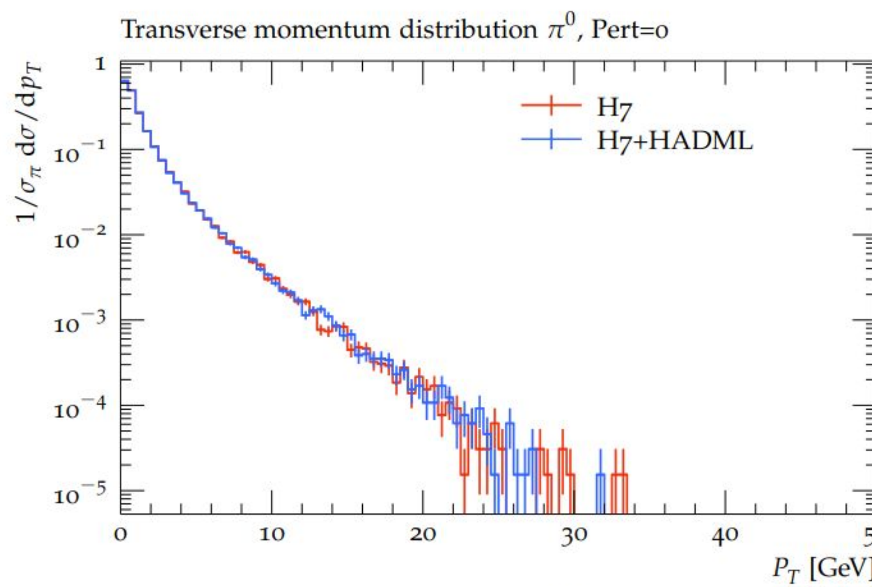
Results

Low-level Validation
(similar to training data)

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

π^0 kinematic variables





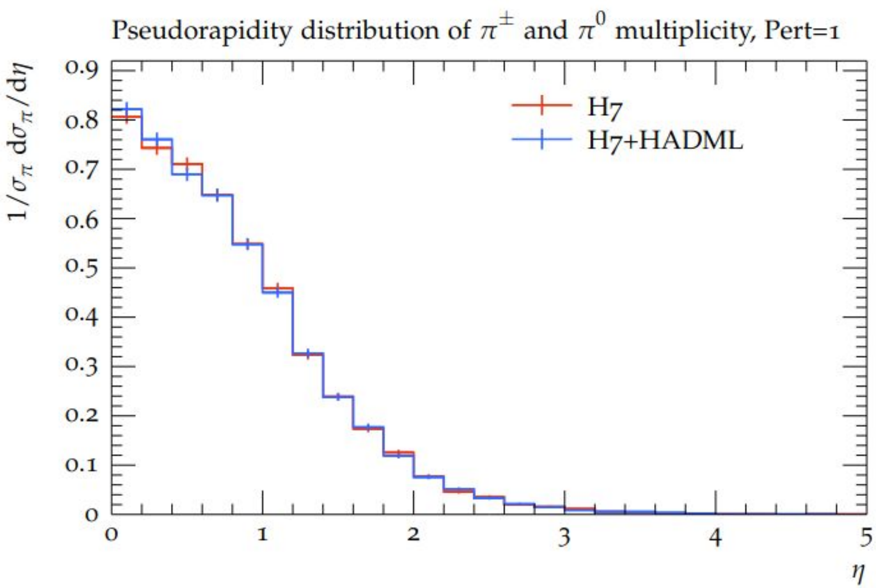
Pert = 0 (no memory of quark kinematics)

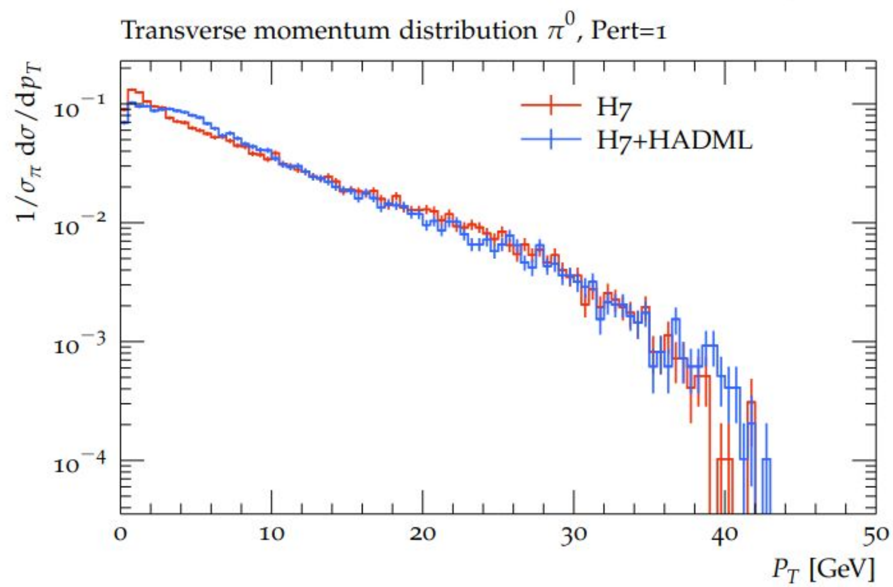
Results

Low-level Validation
(similar to training data)

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

π^0 kinematic variables

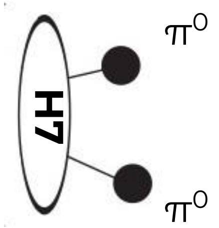




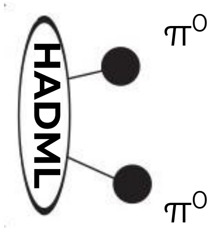
Pert = 1 (memory of quark kinematics)

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

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

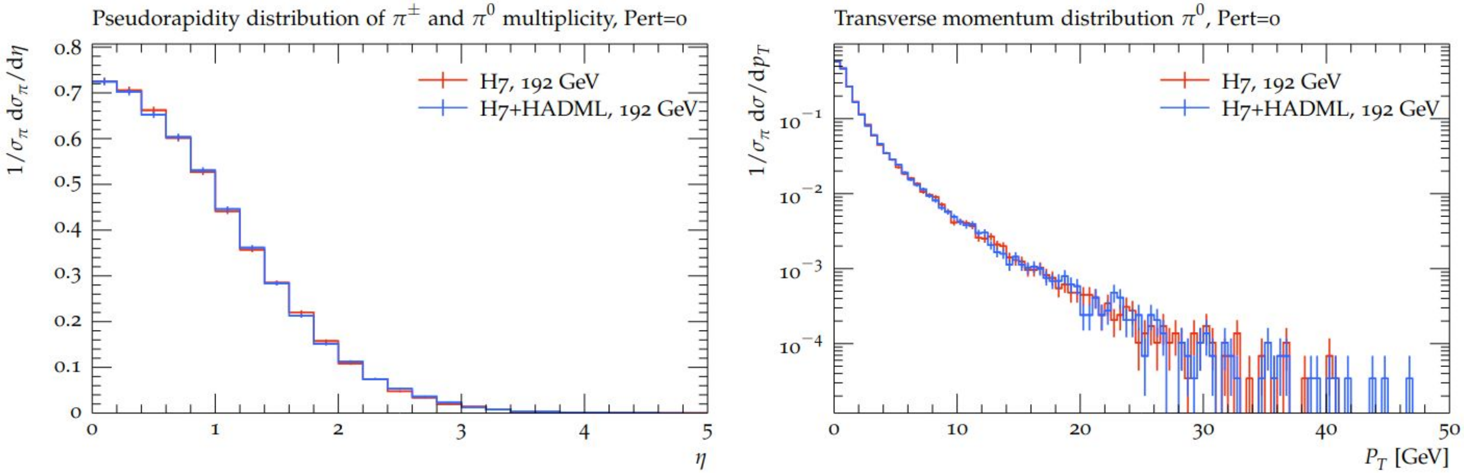


VS

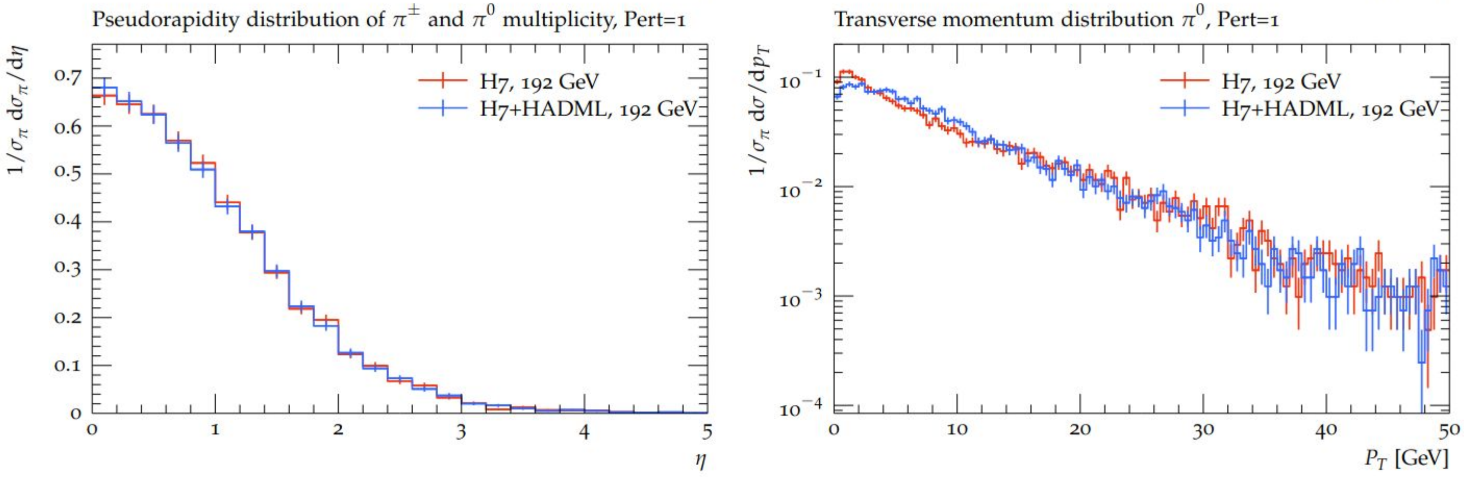


π^0 kinematic variables

Pert = 0



Pert = 1



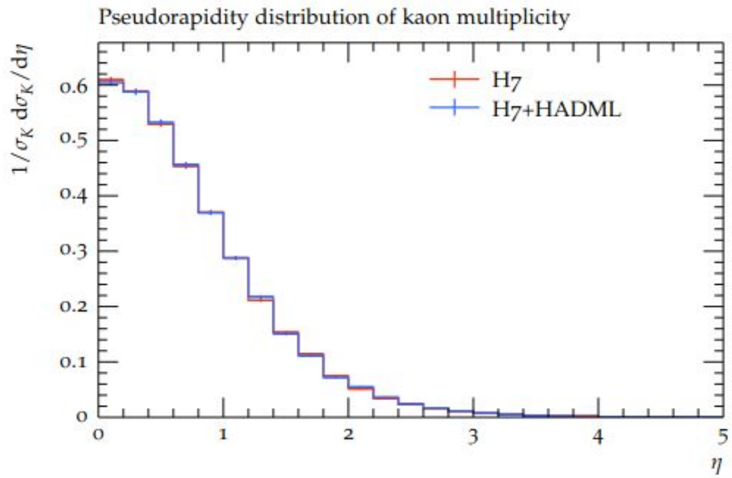
Results

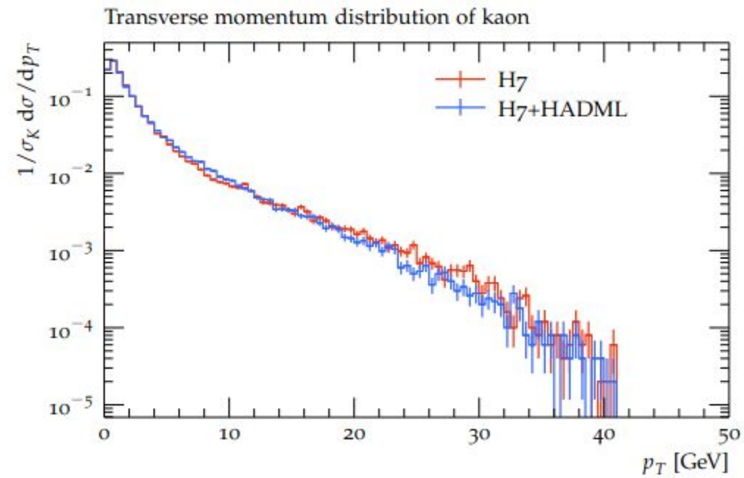
Low-level Validation
(**beyond** training data **different hadrons**)

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

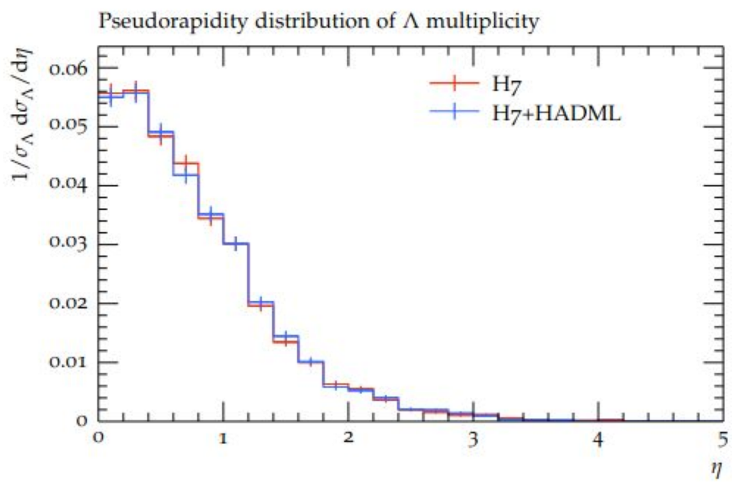
h kinematic variables

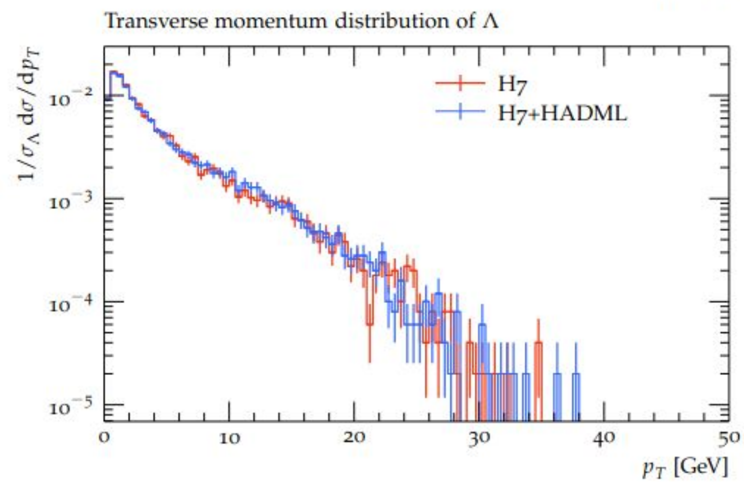
Kaons





Lambda

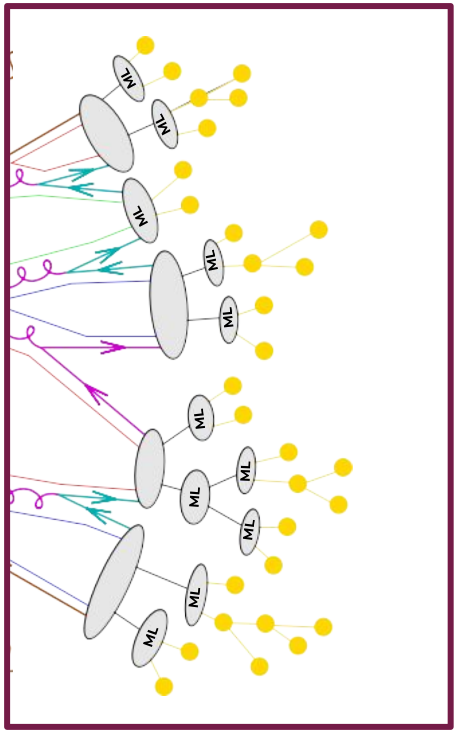


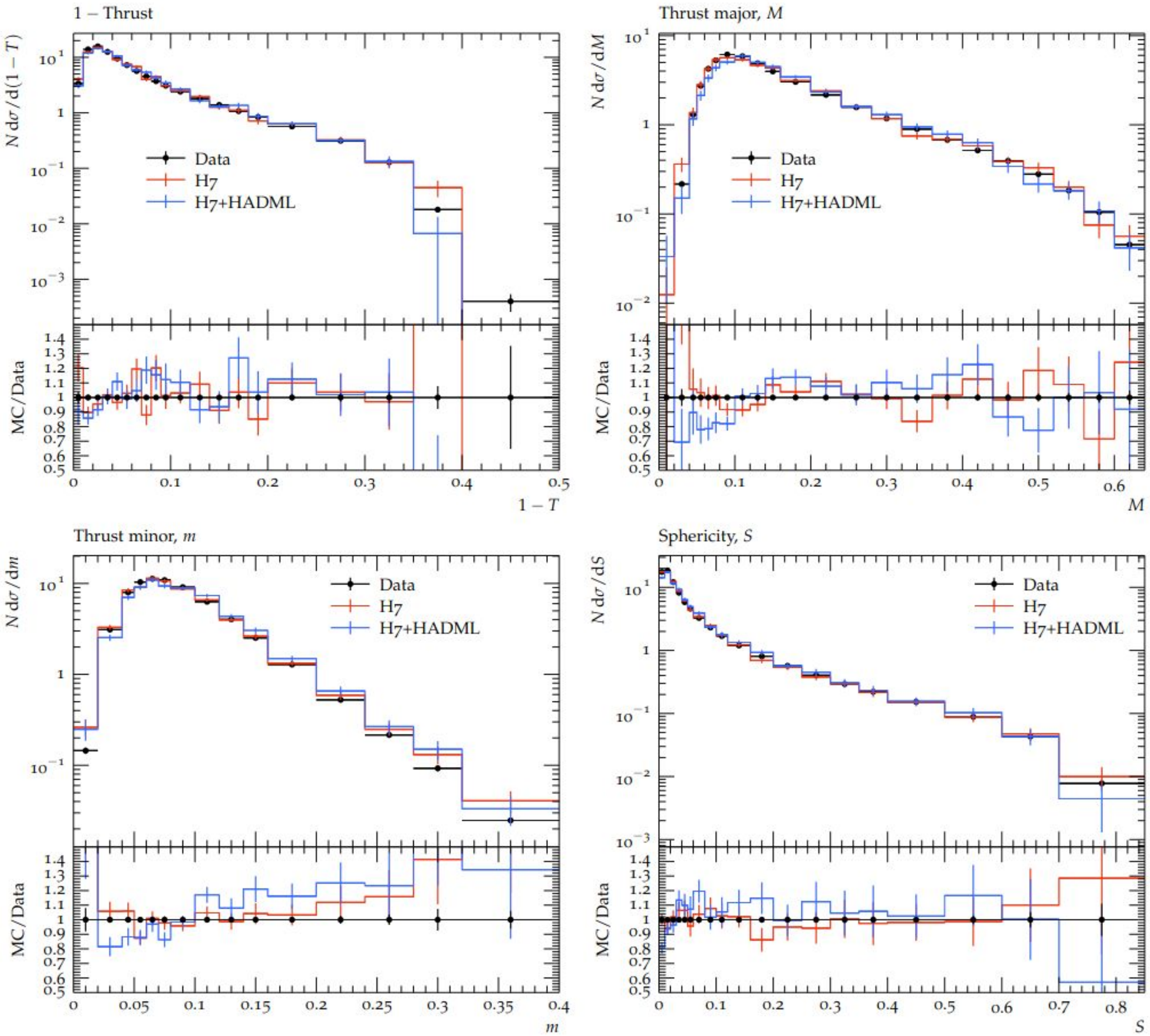


Full-event Validation

(Full events using HADML integrated into Herwig 7)

LEP DELPHI **Data**

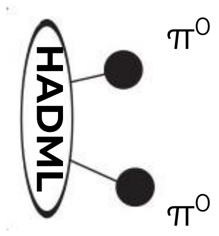




Summary and Outlook

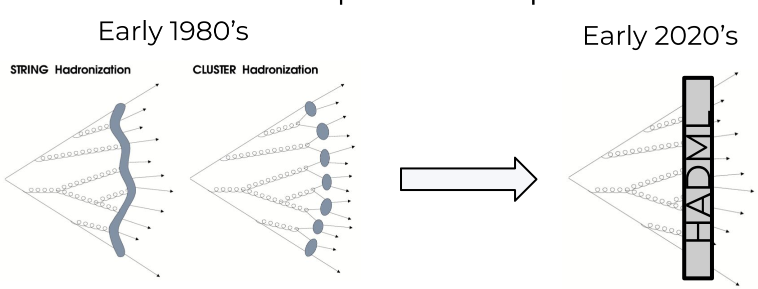
Summary

- We presented **first step** on the path towards a **neural network-based hadronization model**
- We emulated cluster hadronization model from Herwig with a GAN (**HADML**)
- HADML is designed to reproduce the two-body decay of clusters into pions
- The kinematic properties of other hadrons are emulated using the pion model and conservation of energy.
- HADML is able to reproduce Herwig's light cluster decays
- Integrated with the full Herwig simulation is able to reproduce results from LEP data



Outlook

- The ultimate goal of is to train the ML model directly on data to improve hadronization models
- Number of technical and methodological step needed:
  - ➔ Directly accommodate multiple hadron species with their relative probabilities
  - ➔ Heavy cluster decays
  - ➔ Hyperparameter optimization, including the investigation of alternative generative models
  - ➔ Methodological innovation is required to explore how to tune the model to data



Advertisement

2 postdoc in ML/HEP positions openings

JAGIELLONIAN UNIVERSITY
IN KRAKÓW

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

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:

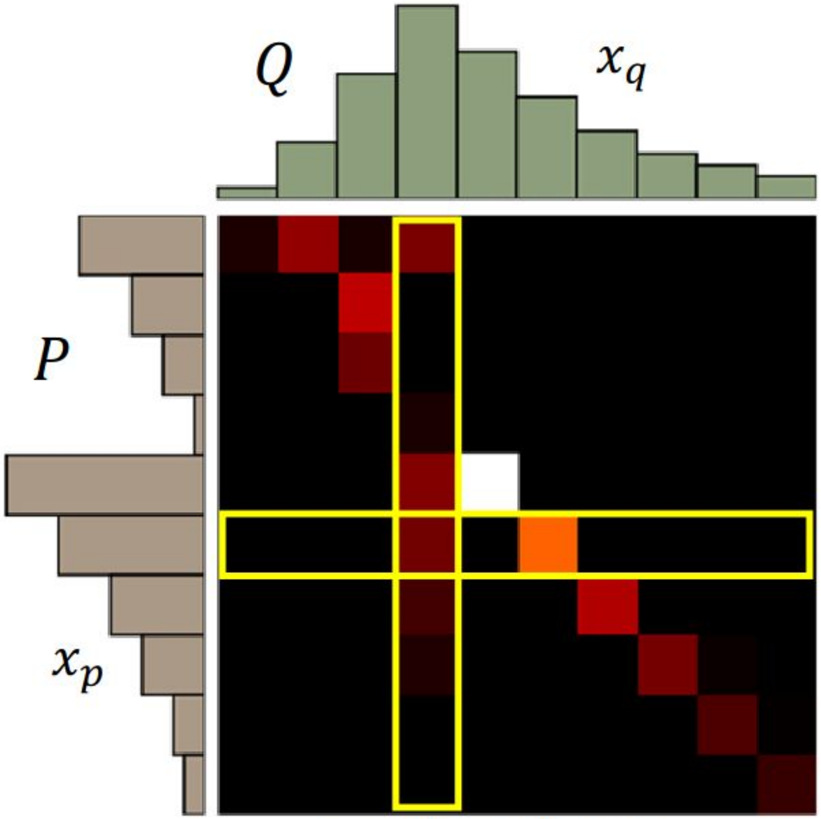
- `D(x)` is the discriminator's estimate of the probability that real data instance x is real.
- E_x is the expected value over all real data instances.
- `G(z)` is the generator's output when given noise z.
- `D(G(z))` is the discriminator's estimate of the probability that a fake instance is real.
- 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)))`.

AlphaGo

- AlphaGo's victory against Lee Sedol was a major milestone in artificial intelligence research.
- Go had previously been regarded as a hard problem in machine learning that was expected to be out of reach for the technology of the time.
- Most experts thought a Go program as powerful as AlphaGo was at least five years away;some experts thought that it would take at least another decade before computers would beat Go champions. Most observers at the beginning of the 2016 matches expected Lee to beat AlphaGo.
- Netflix document

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