Towards a Deep Learning Model for Hadronization

Andrzej Siódmok

in collaboration with

Aishik Ghosh

Xiangyang Ju

Ben Nachman

based on arXiv: 2203.12660

JAGIELLONIAN UNIVERSITY IN KRAKÓW

UCZELNIA BADAWCZA INICJATYWA DOSKONAŁOŚCI

NARODOWE CENTRUM NAUKI

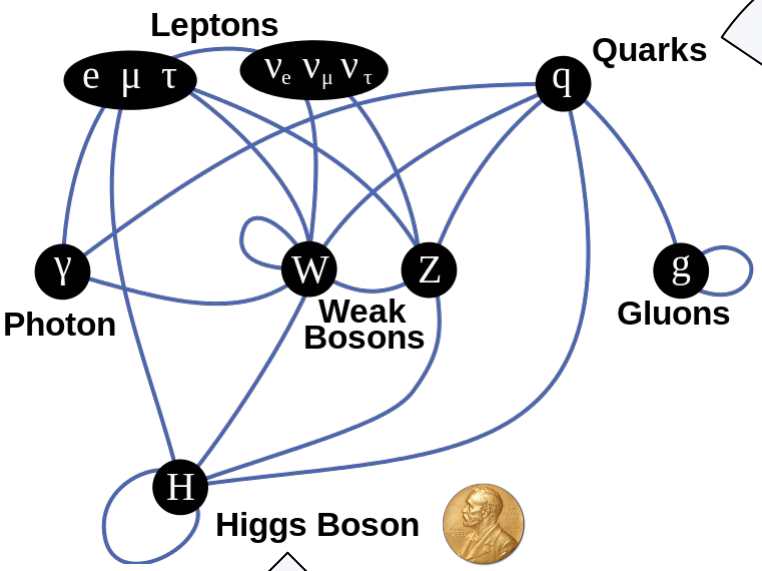
NCN: 2019/34/E/ST2/00457

MCnet

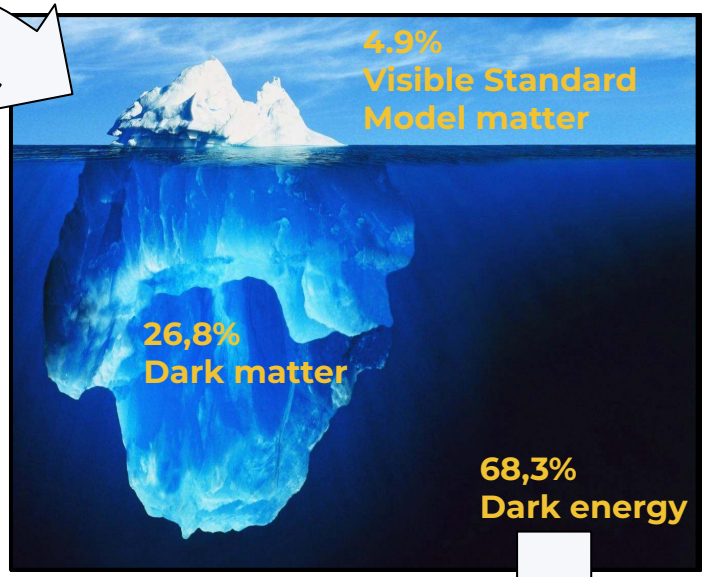
Motivation

Standard Model

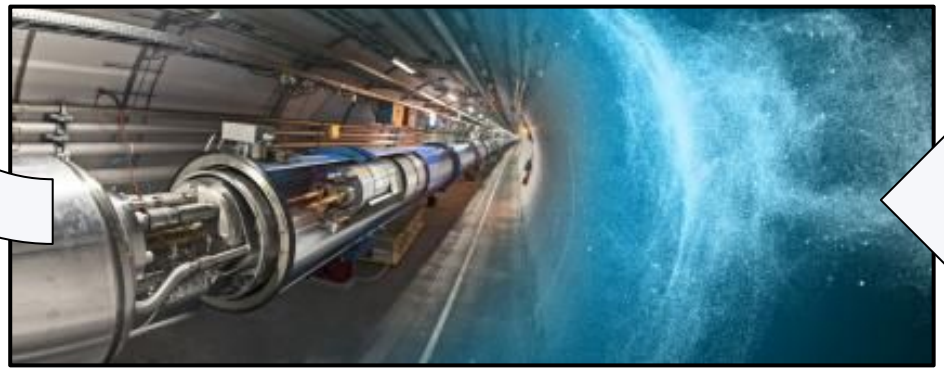
very successful theory



Physics beyond SM must exist



LHC



European Strategy for Particle Physics

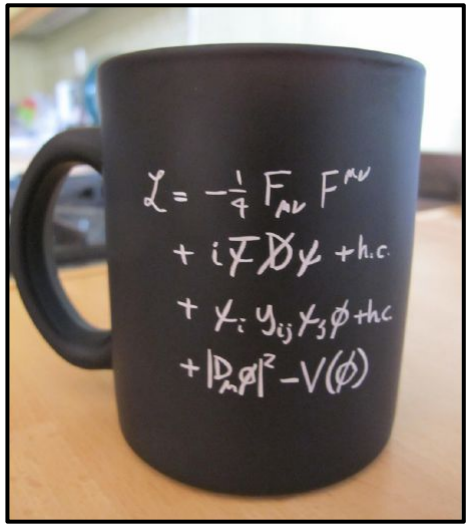
"Europe's top priority should be the exploitation of the full potential of the LHC"

Standard Model

There is a **huge gap** between a one-line formula of a fundamental theory, like the Lagrangian of the SM, and the experimental reality that it implies

Theory

Standard Model Lagrangian



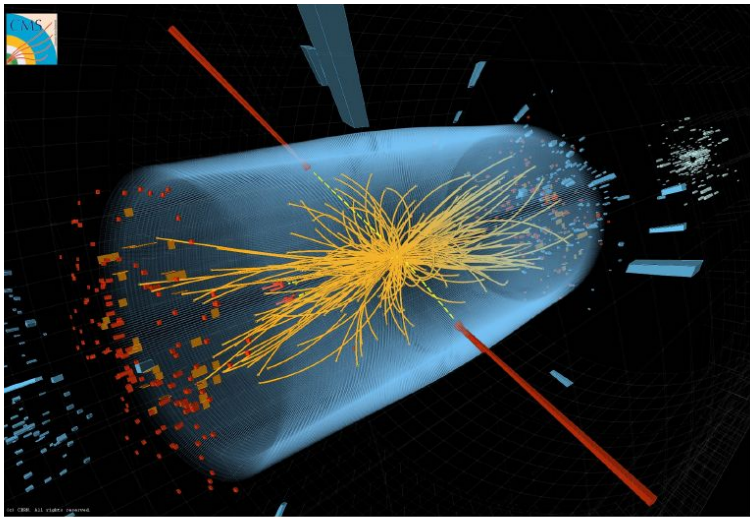
Data makes you smarter

It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong.

Richard P. Feynman

Experiment

LHC event

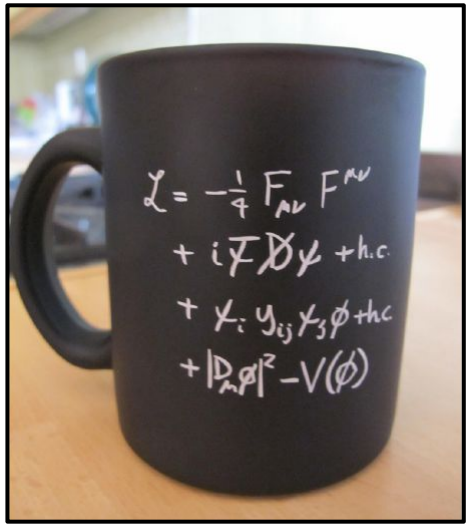


Standard Model

There is a **huge gap** between a one-line formula of a fundamental theory, like the Lagrangian of the SM, and the experimental reality that it implies

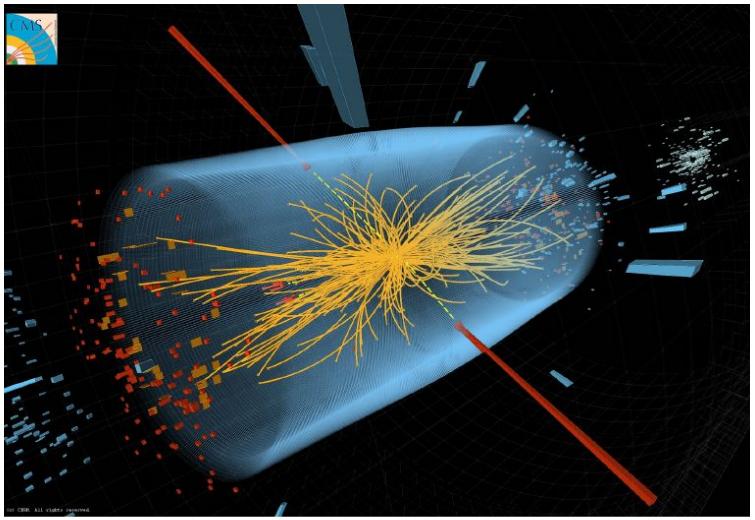
Theory

Standard Model Lagrangian



Experiment

LHC event



- MC event generators are designed to bridge that **gap**
- “Virtual collider” ⇒ Direct comparison with data

Almost all **HEP measurements and discoveries** in the modern era have **relied on MCEG**, most notably the discovery of the Higgs boson.

Published papers by ATLAS, CMS, LHCb: **2252**
Citing at least 1 of 3 existing MCEG: **1888** (**84%**)

Current Situation

No BSM physics at the LHC ⇨ Look for much more subtle effects ⇨ Very precise MC predictions ⇨ Limited by our understanding of QCD

Complex structure of Quantum Chromodynamics (QCD)

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)**

Imagine that the BSM physics signal is at the LHC but due to lack of QCD understanding we missed it

We need:

SIMPLIFICATIONS in perturbative QCD techniques

NEW IDEAS of treatment non-perturbative QCD

QCD ex-Machina

NARODOWE CENTRUM NAUKI
NCN: 2019/34/E/ST2/00457

Motivation - Monte Carlo Event Generators (MCEG)

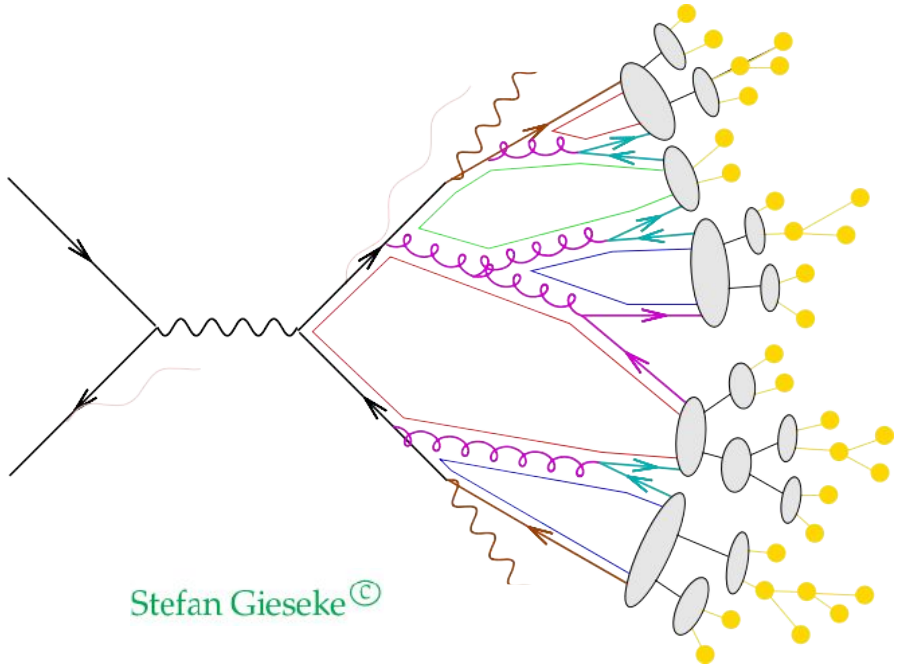
QCD correctly describes strong interactions in each energy range but its complex mathematical structure makes it very difficult to obtain precise predictions (Millennium Prize Problem $1,000,000)

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)**

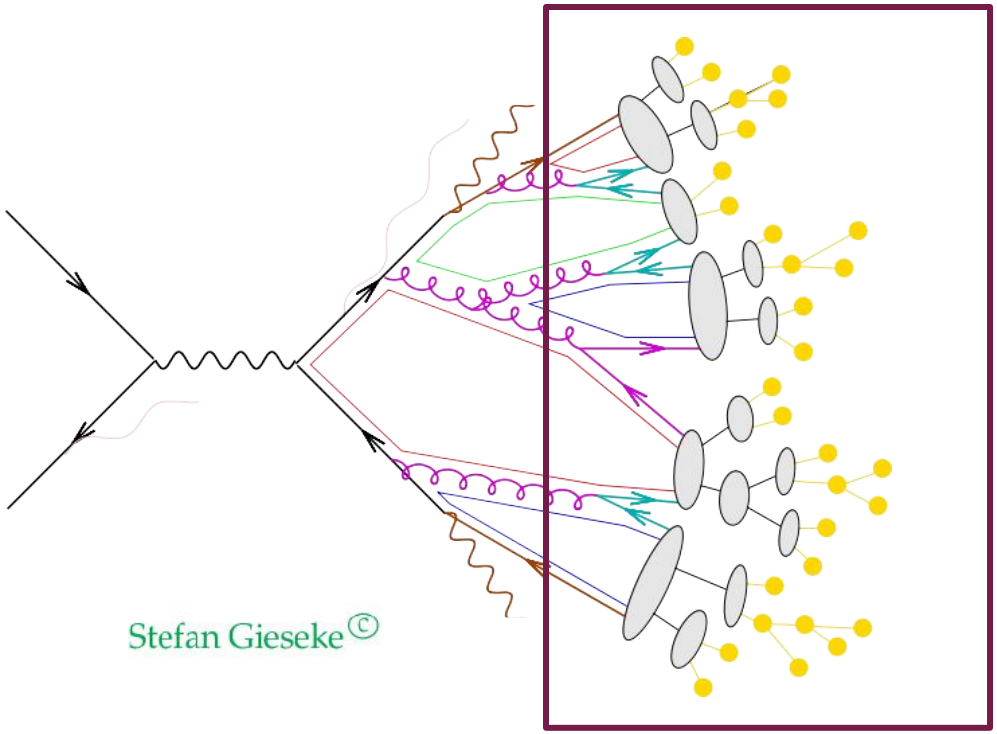


Motivation - Monte Carlo Event Generators (MCEG)

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)

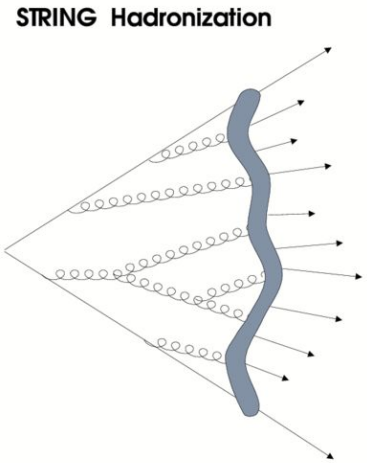
High energy
- **perturbative QCD**
- **in theory we know what to do**
- **in practice very difficult**

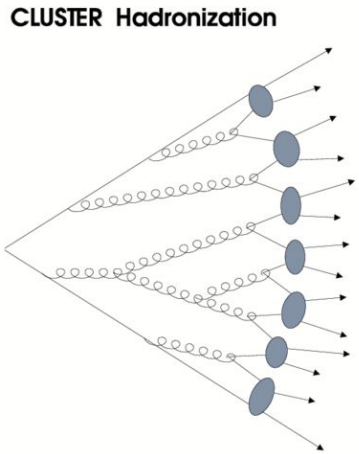
Low energy
- **non-perturbative QCD**
- **we don't know what to do**
- **phenomenological models (with many free parameters)**



Hadronization:
one of the least understood elements of MCEG

Hadronization:



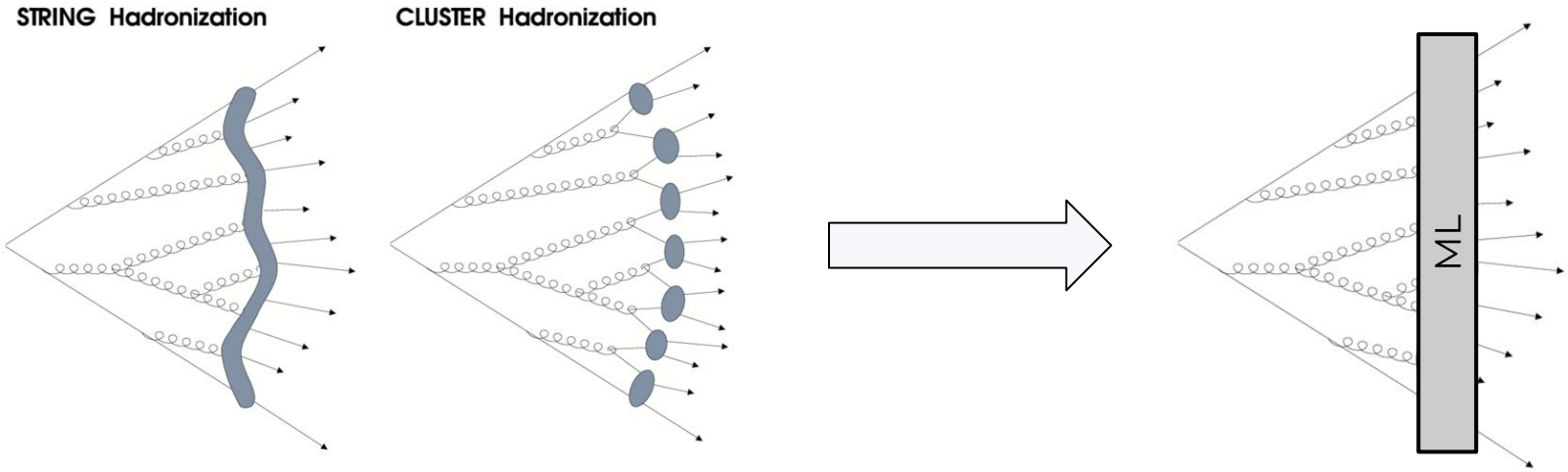


- ➔ Increased control of perturbative corrections ⇒ more often the precision of LHC measurements is limited by MCEG's non-perturbative components, such as hadronization.
- ➔ Hadronization (phenomenological models with many free parameters ~ 30 parameters)

Hadronization:

Early 1980's
(since then very little development)

Early 2020's
(lot of progress in ML)



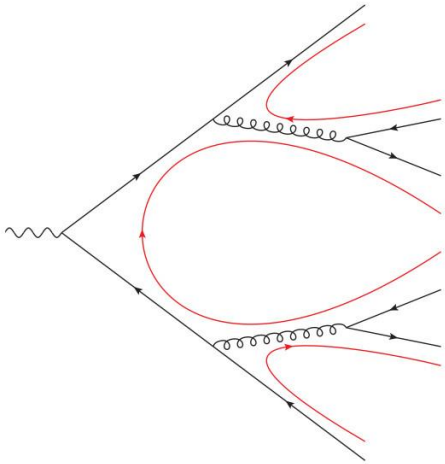
- ➔ Increased control of perturbative corrections ⇒ more often the precision of LHC measurements is limited by MCEG's non-perturbative components, such as hadronization.
- ➔ Hadronization (phenomenological models with many free parameters ~ 30 parameters)
- ➔ Hadronization is a fitting problem ML is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

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

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:

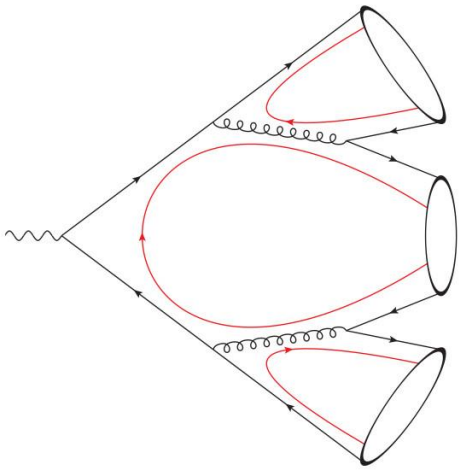


- QCD provide pre-confinement of colour

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:

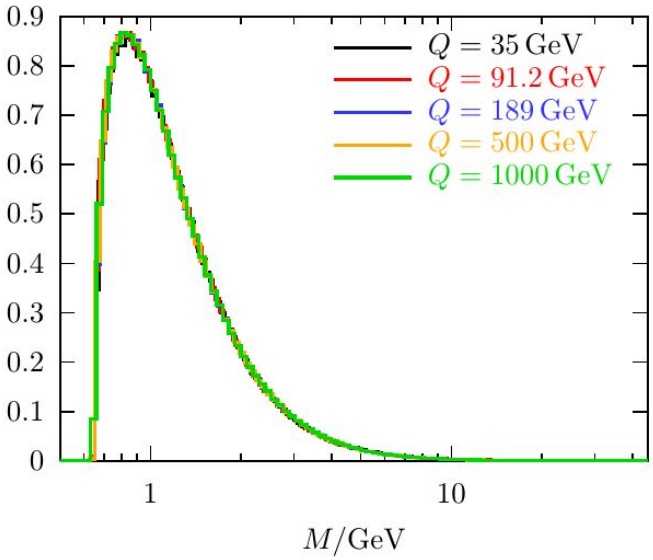


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters

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:

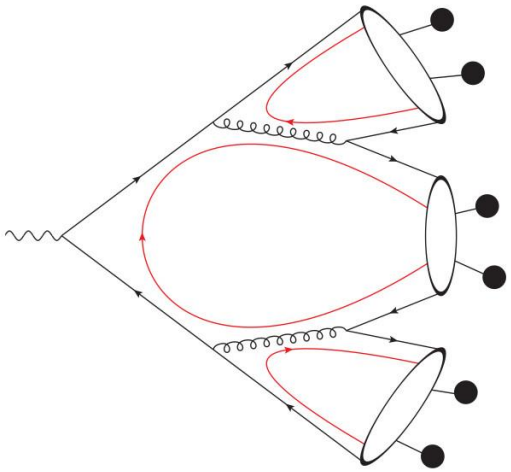


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision

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:

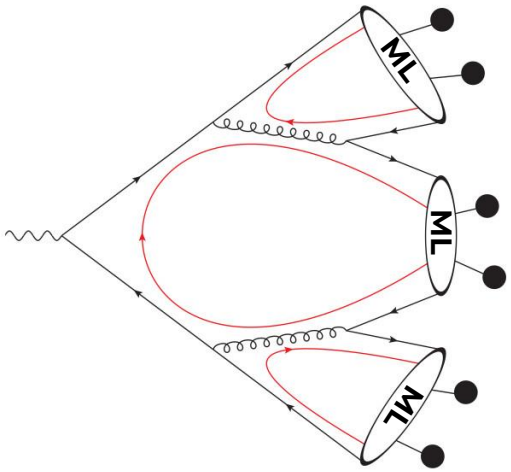


- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically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

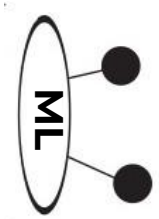
Cluster hadronization model

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

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



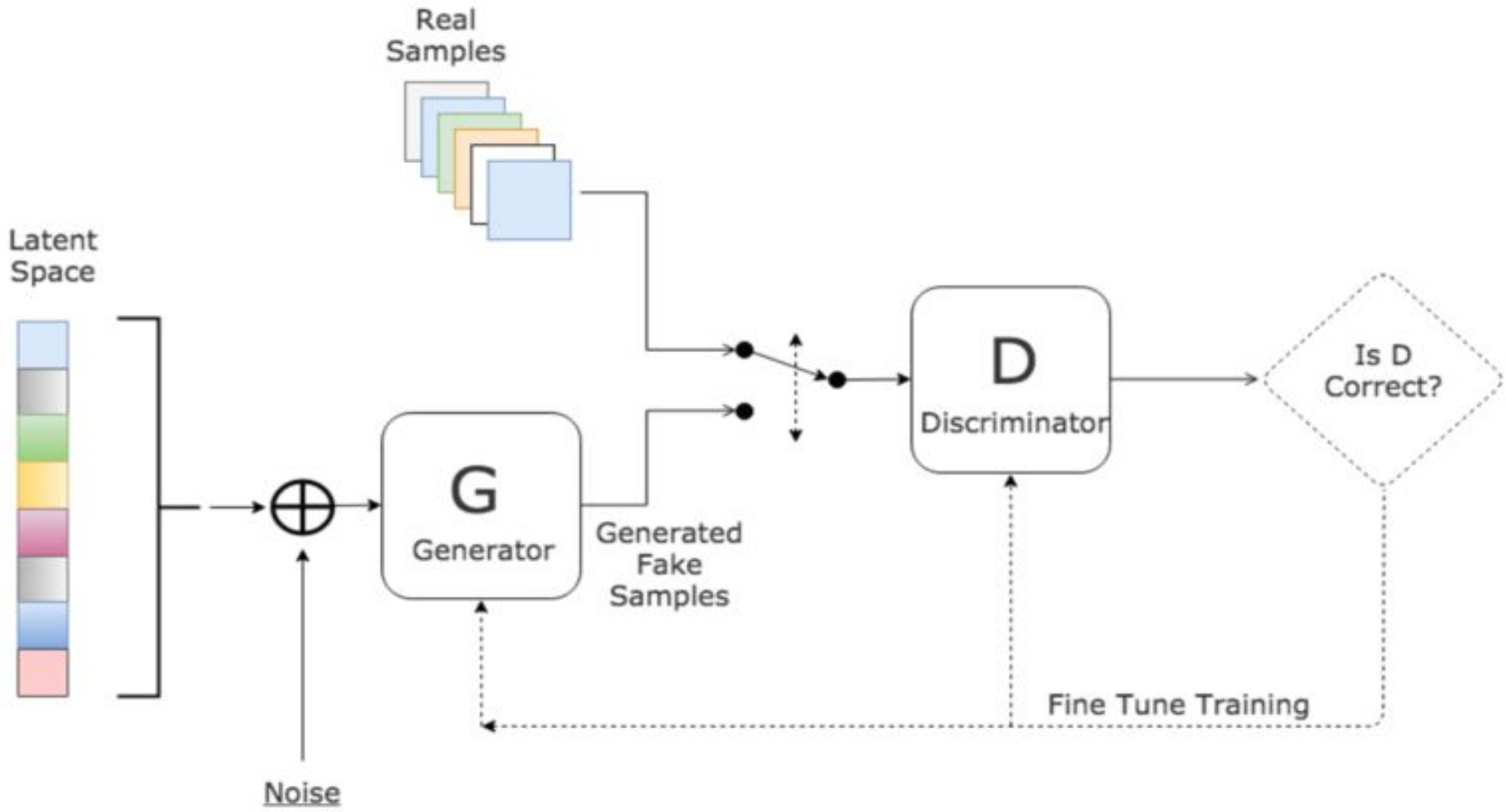
- QCD provide pre-confinement of colour
- Colour-singlet pair end up close in phase space and form highly excited hadronic states, the clusters
- Pre-confinement states that the spectra of clusters are independent of the hard process and energy of the collision
- Peaked at low mass (1-10 GeV) typically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



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

Generative Adversarial Network (GAN)

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



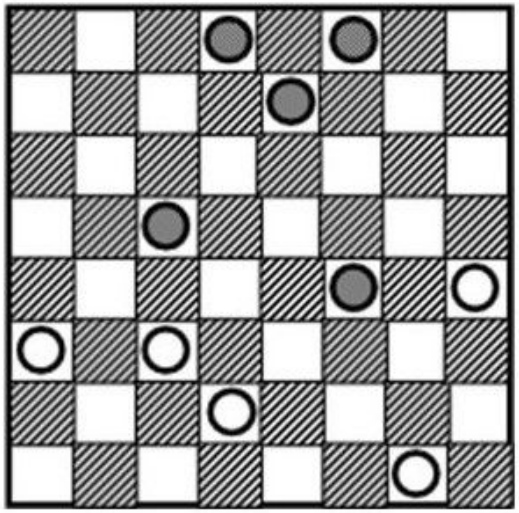
Generative Adversarial Network (GAN)

thispersondoesnotexist.com



Adversarial Networks

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

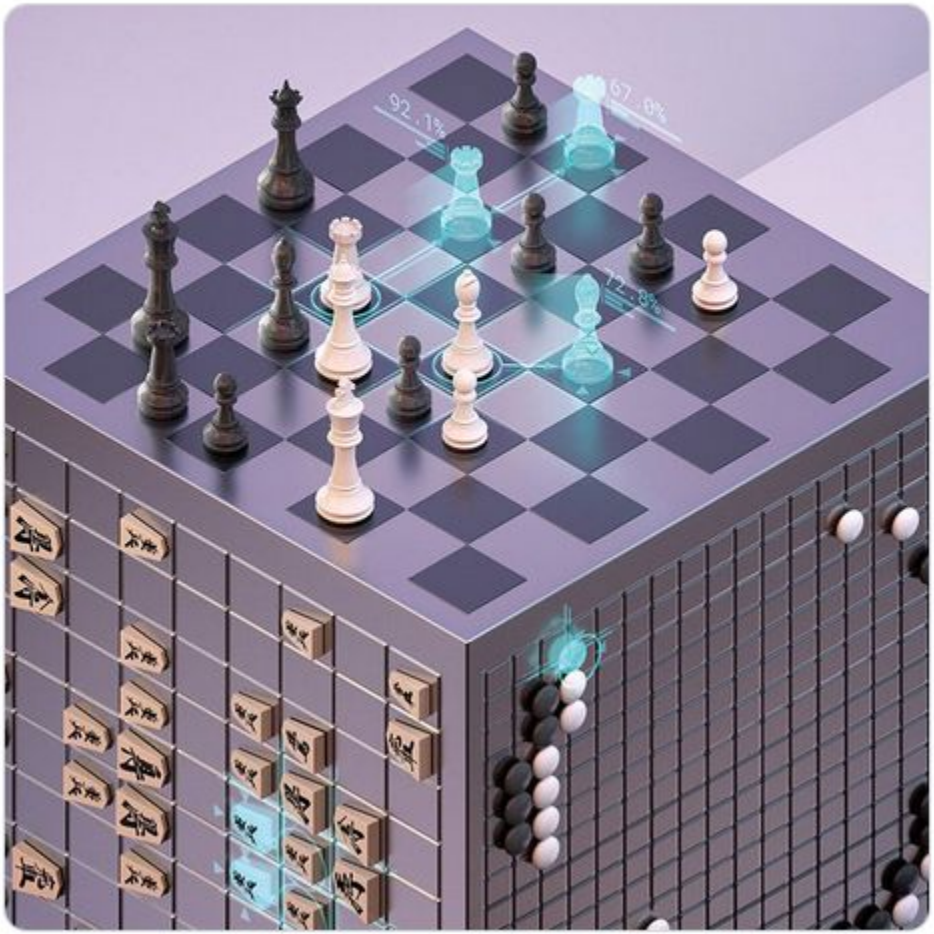


- He popularized the term **"machine learning"** in 1959.
- The program chose its move based on a **minimax** strategy, meaning it made the move assuming that the opponent was trying to optimize the value of the same function from its point of view.
- He also had it play thousands of **games against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



By playing **games against itself**, AlphaGo Zero surpassed the strength of AlphaGo [vs Lee Sedol 4:1] 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

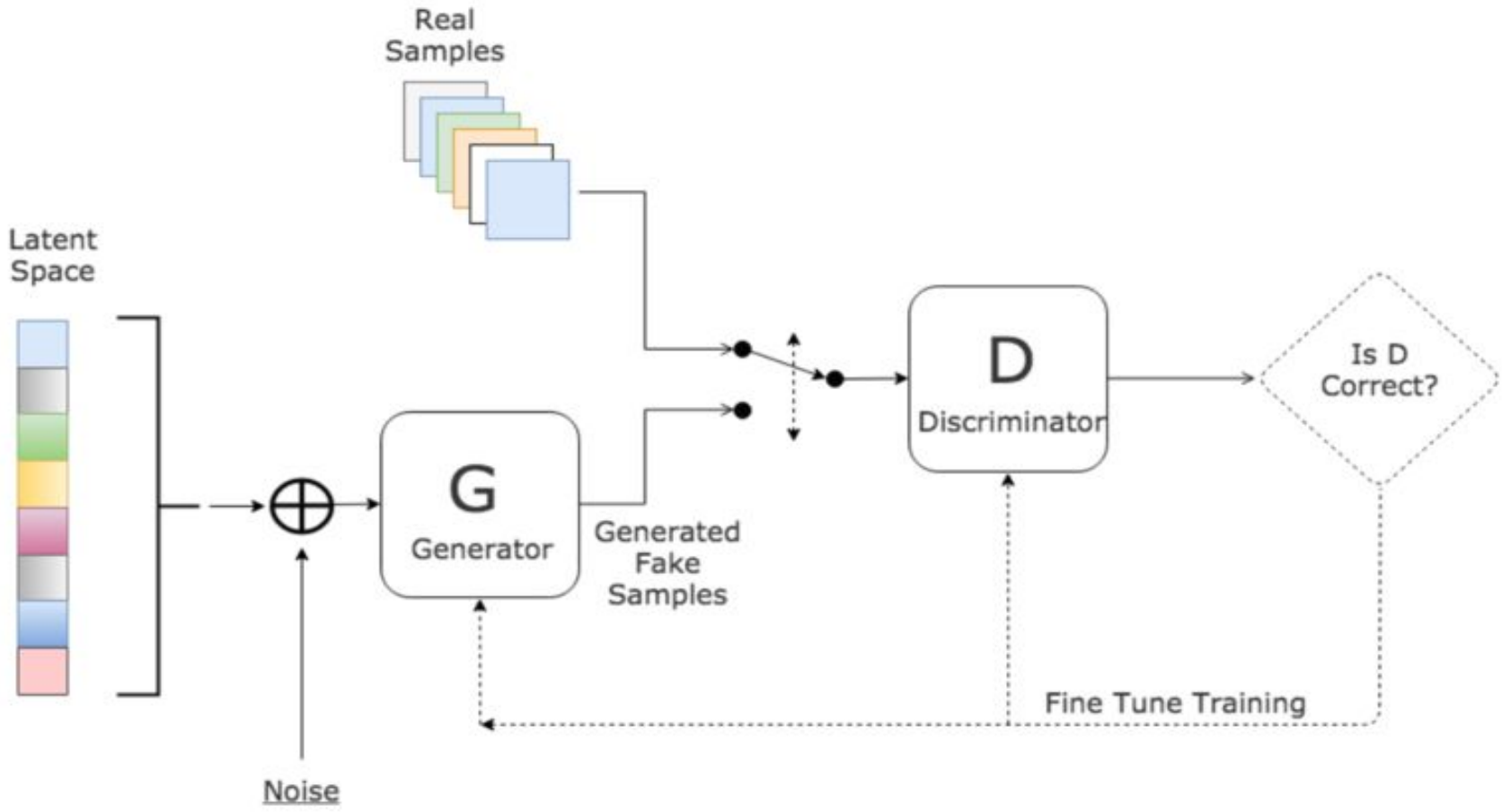
Demis Hassabis
CBE FRS FREng FRSA

Wasserstein GAN (WGAN)

[Martin Arjovsky, Soumith Chintala, and Leon Bottou, arxiv:1701.07875, Dec 2017]

How do you capture the difference between two distributions in GAN loss functions?
This question is an area of active research.

- GAN: **minimax loss**

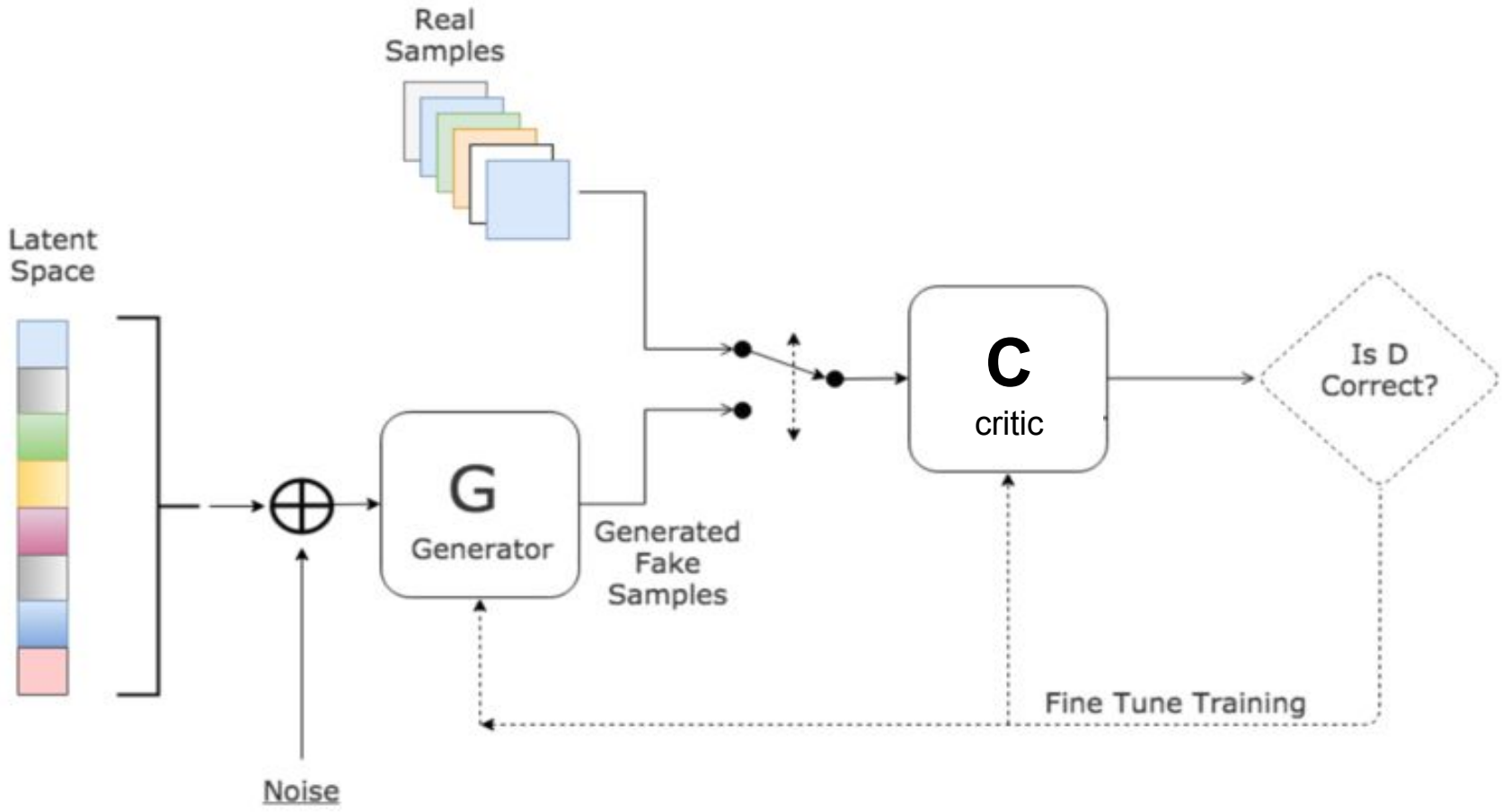


Wasserstein GAN (WGAN)

[Martin Arjovsky, Soumith Chintala, and Leon Bottou, arxiv:1701.07875, Dec 2017]

How do you capture the difference between two distributions in GAN loss functions?
This question is an area of active research.

- WGAN: **Wasserstein loss**

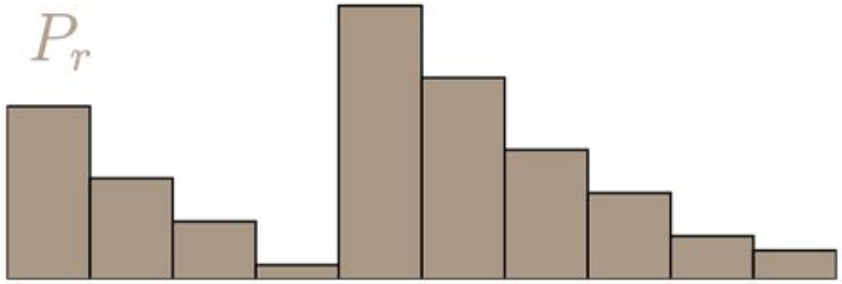


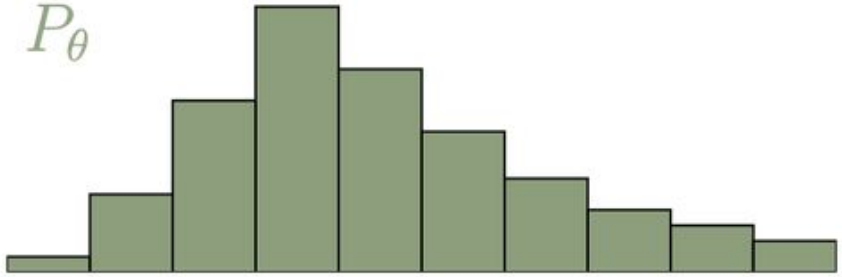
Wasserstein GAN (WGAN)

The Wasserstein distance

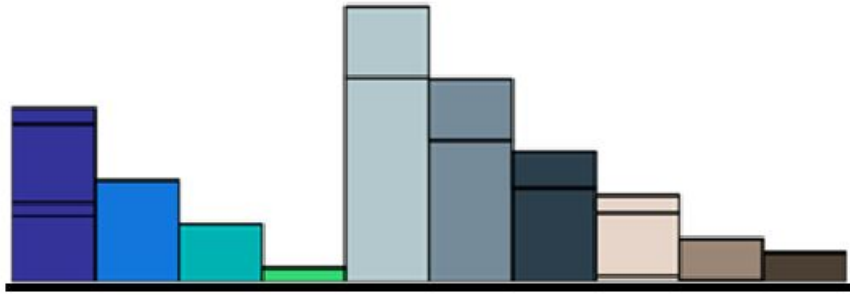
- For discrete probability distributions, the Wasserstein distance is called the earth mover's distance (EMD):
- EMD is the minimal total amount of work it takes to transform one heap into the other.
- $$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$
- Work is defined as the amount of earth in a chunk times the distance it was moved.

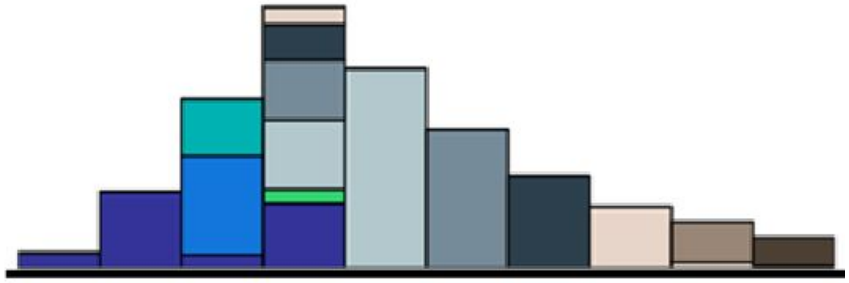
$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$





Best "moving plans" of this example





Wasserstein GAN (WGAN)

WGAN: **Wasserstein loss**

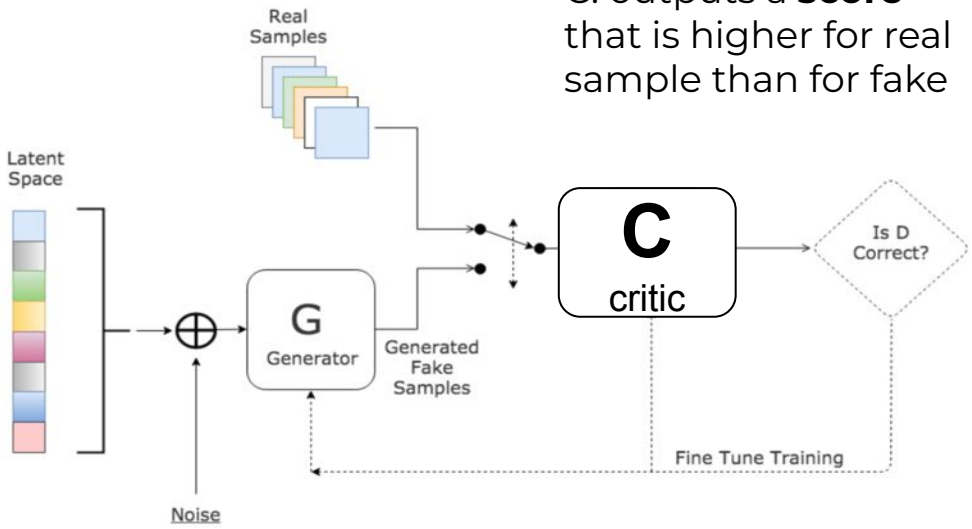
- Critic training just tries to make the output bigger for real instances than for fake instances:

  Critic Loss: D(x) - D(G(z))
  The Critic tries to maximize this function.

- The generator tries to maximize the discriminator's output for its fake instances

  Generator Loss: D(G(z))

C: outputs a **score** that is higher for real sample than for fake



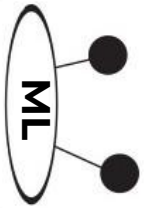
In these functions:

- D(x) is the critic's output for a real instance.
- G(z) is the generator's output when given noise z.
- D(G(z)) is the critic's output for a fake instance.
- The output of critic D does *not* have to be between 1 and 0.
- The formulas derive from the EMD between the real and generated distributions.

Towards a Deep Learning Model for Hadronization

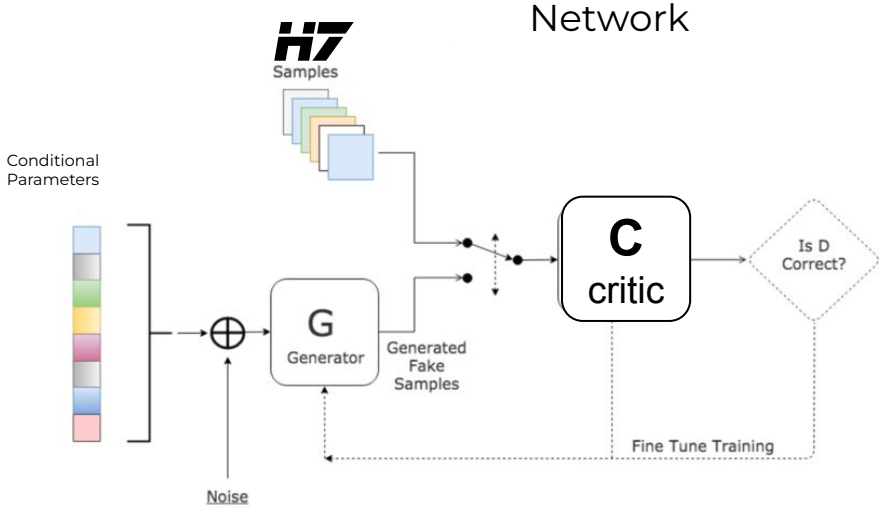
ML hadronization

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



How?

Wasserstein Generative Adversarial Network

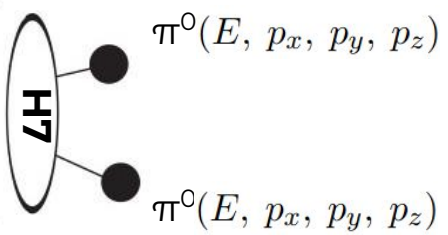


Training data:

H7

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

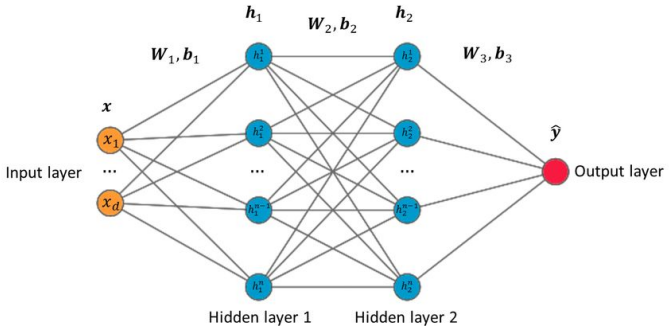
Cluster $(E,\ p_x,\ p_y,\ p_z)$



Pert = 0/1 memory of quarks direction

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)

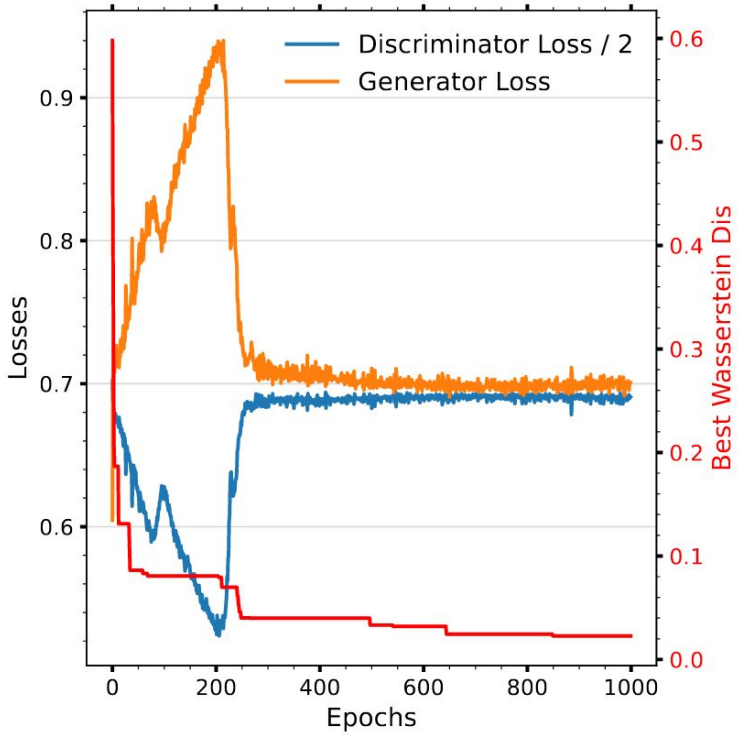
Output

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)

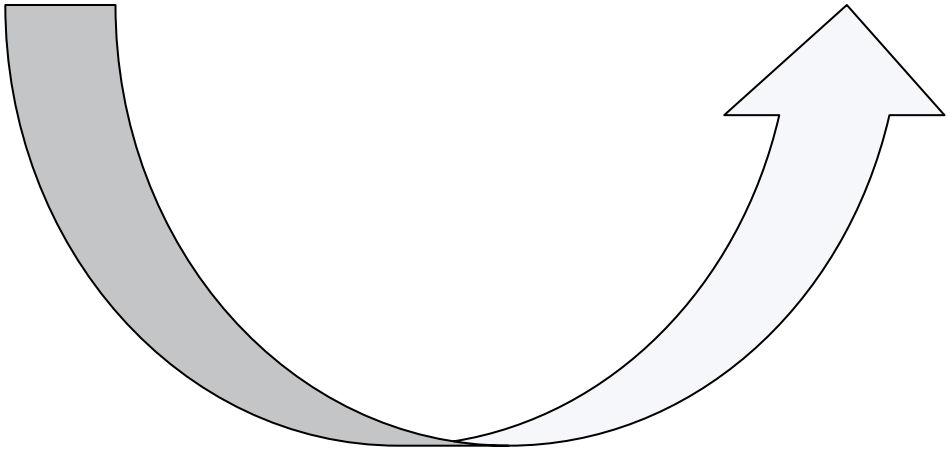
- **Discriminator** and the **Generator** are trained separately and alternately by two independent Adam optimizers with a learning rate of 10^{-4}, for 1000 epochs



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

Training

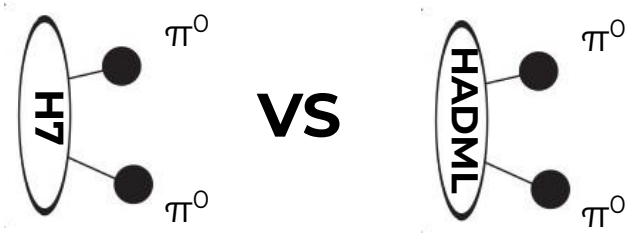
Event generation



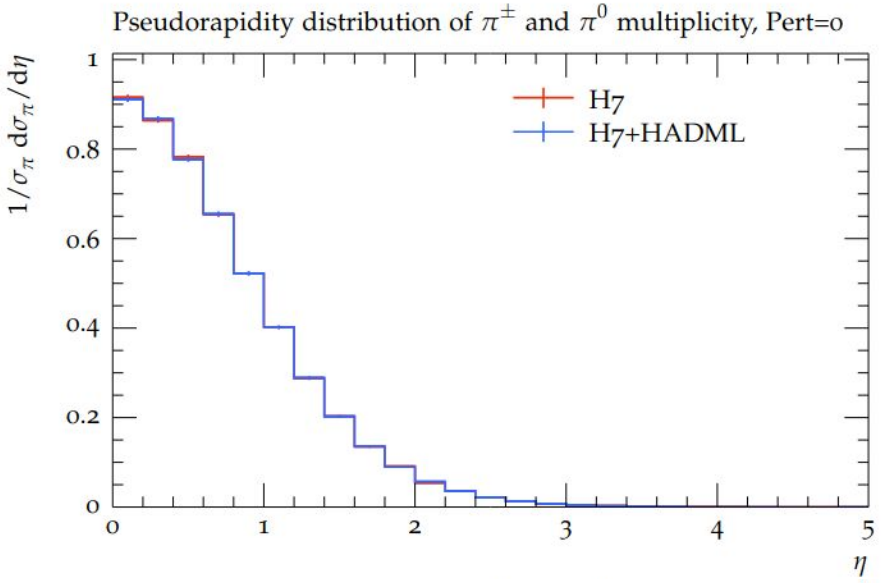
Results

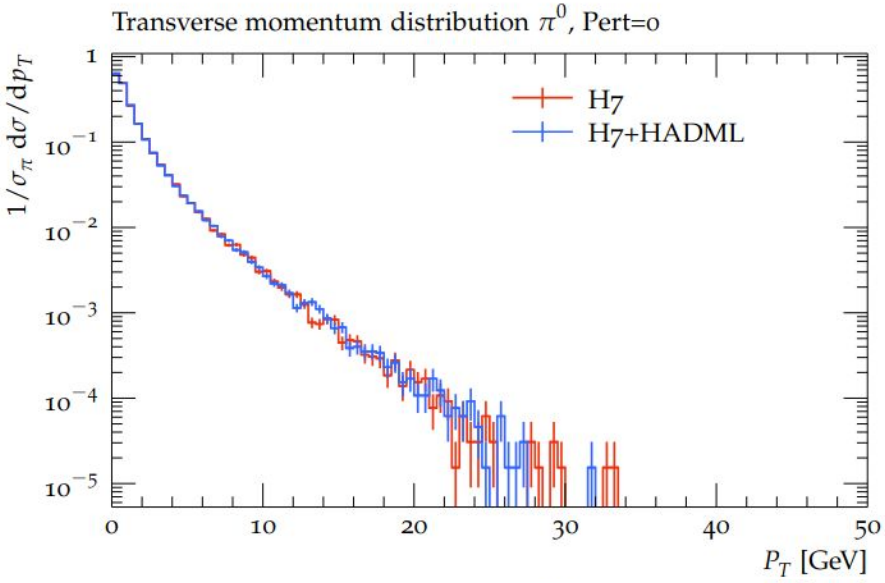
Low-level Validation
(similar to training data)

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



π^0 kinematic variables



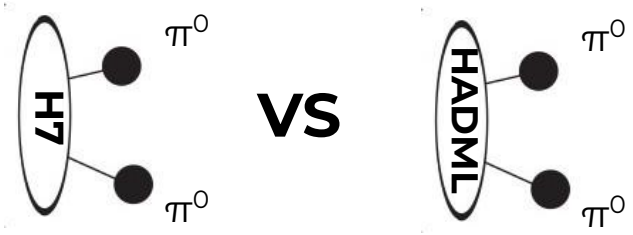


Pert = 0 (no memory of quark kinematics)

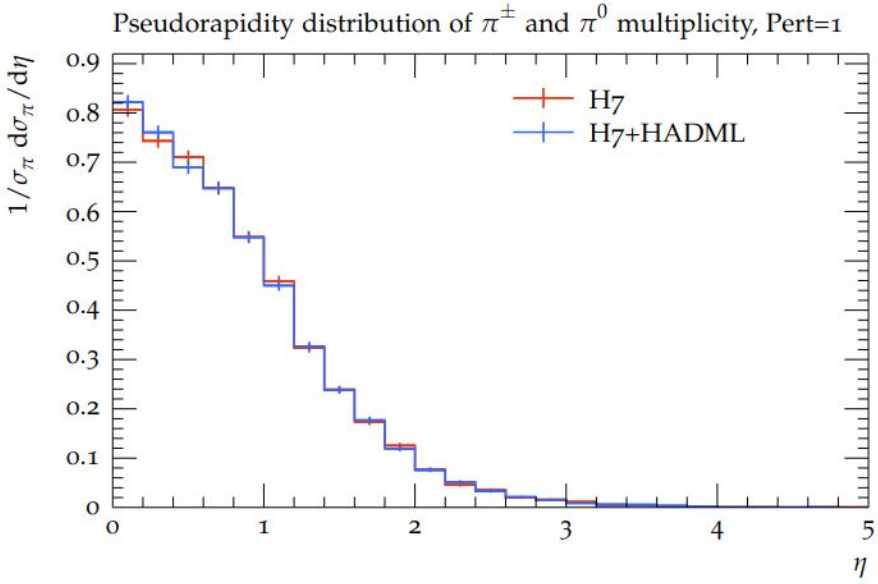
Results

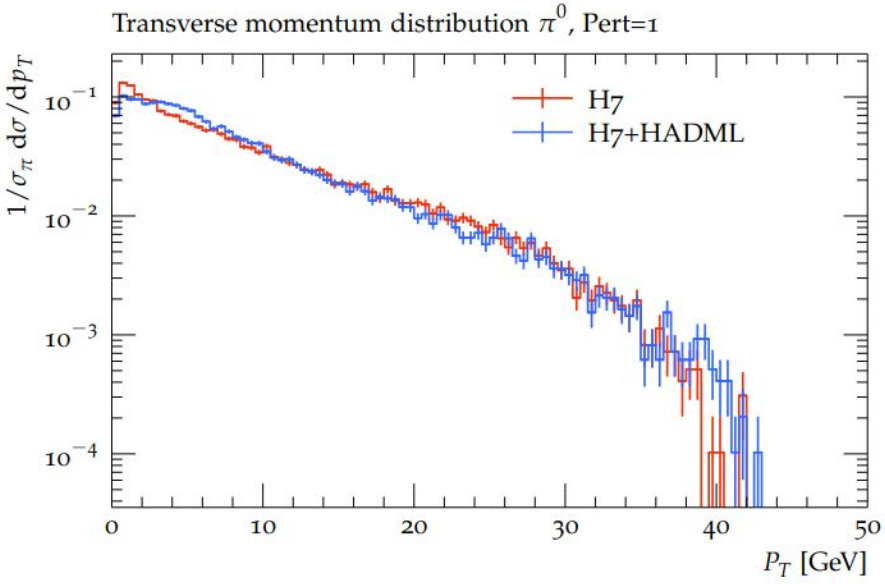
Low-level Validation
(similar to training data)

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



π^0 kinematic variables



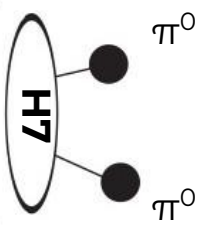


Pert = 1 (memory of quark kinematics)

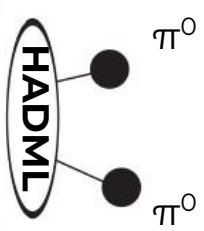
Results

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

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

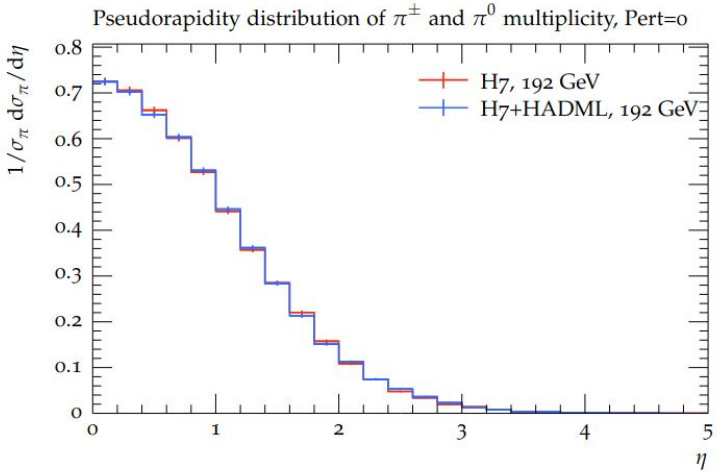


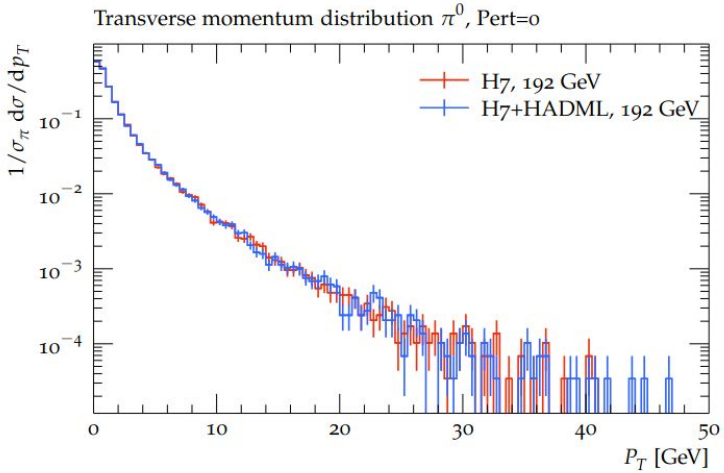
VS



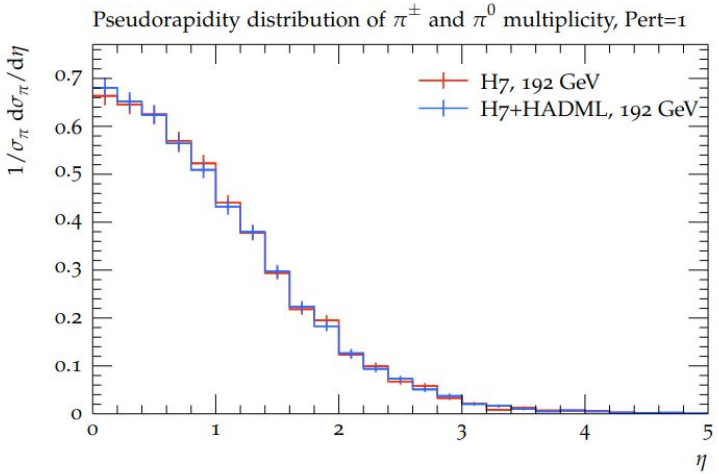
π^0 kinematic variables

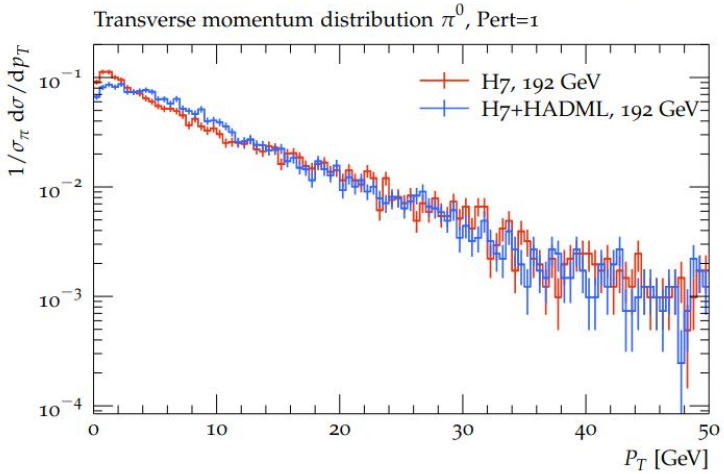
Pert = 0





Pert = 1

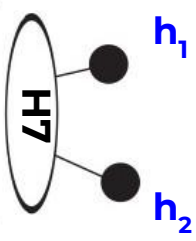




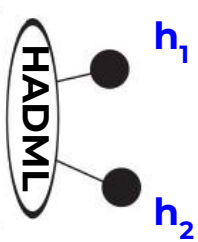
Results

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

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

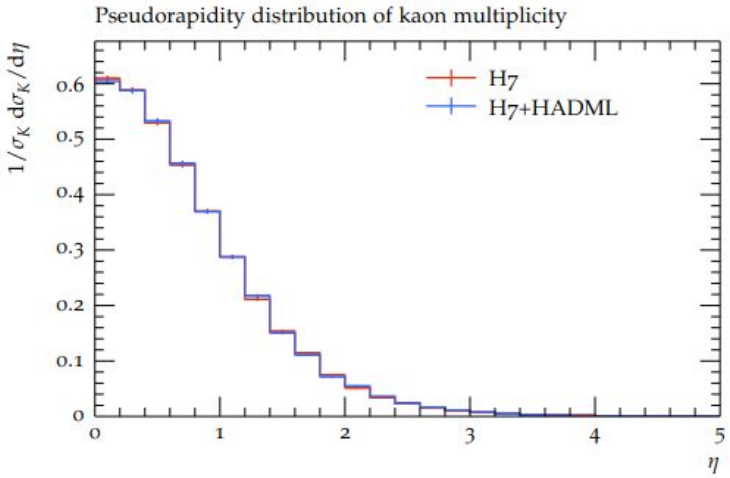


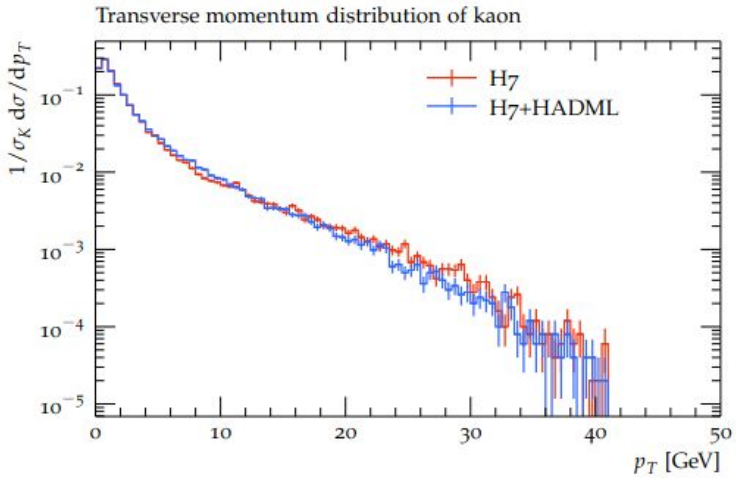
VS



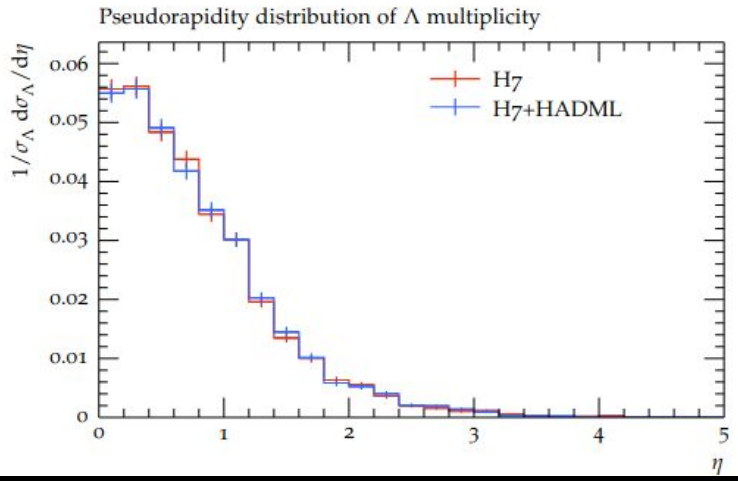
h kinematic variables

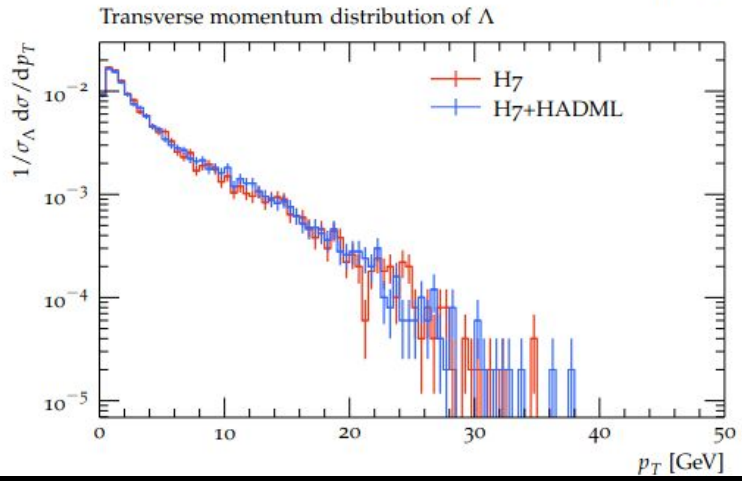
Kaons





Lambda

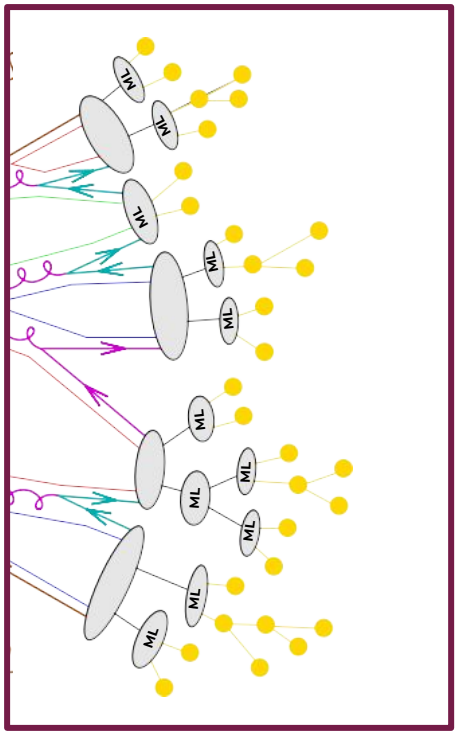


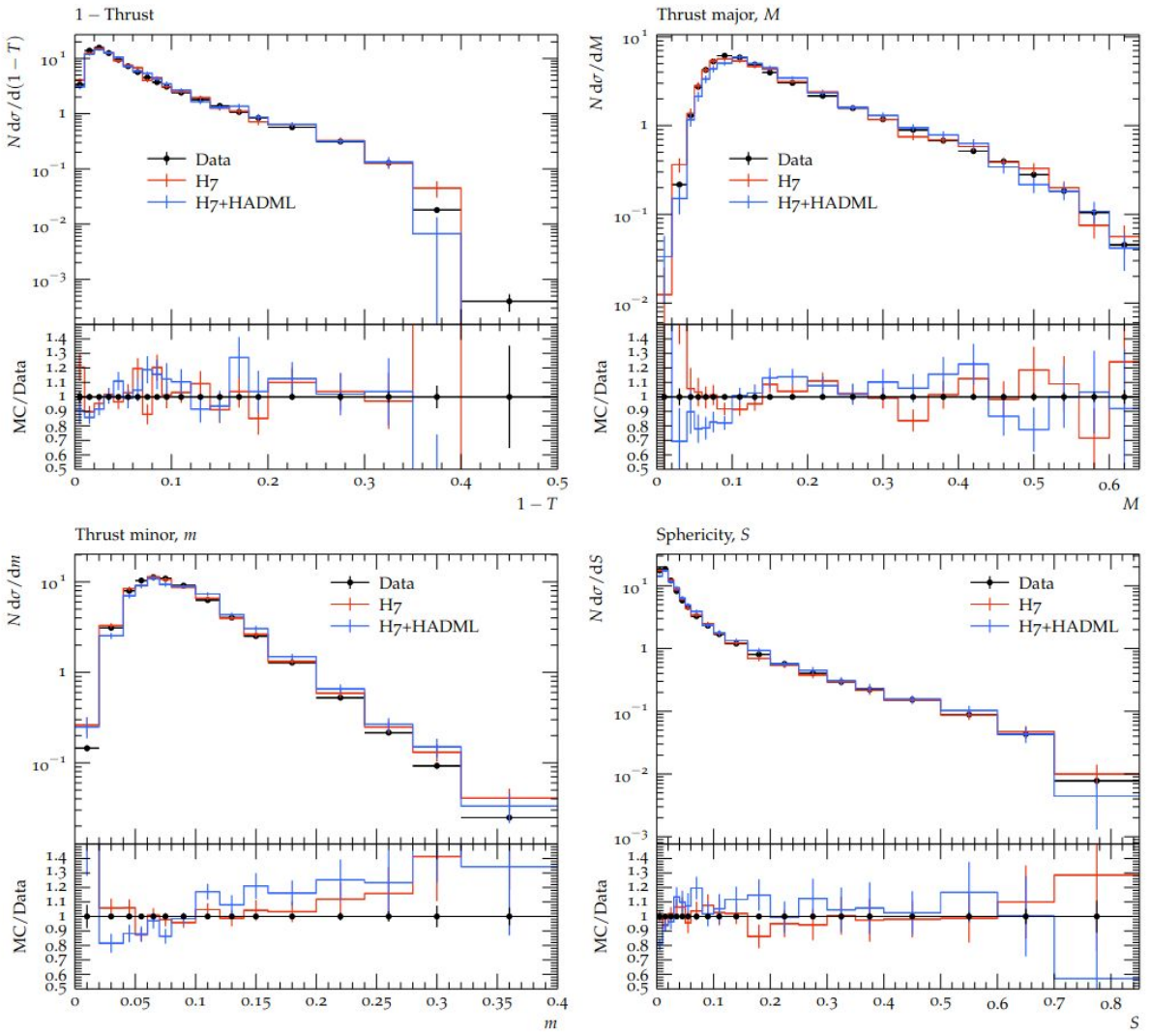


Results

Full-event Validation
(Full events using HADML integrated into Herwig 7)

LEP DELPHI **Data**

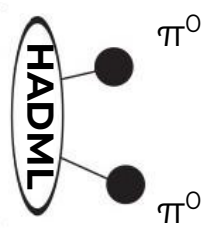




Summary and Outlook

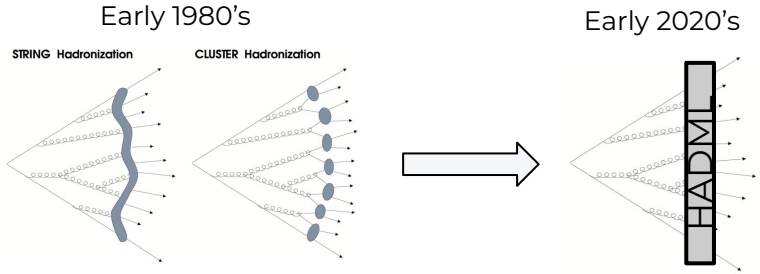
Summary

- 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

UCZELNIA BADAWCZA
INICJATYWA DOSKONAŁOŚCI

JAGIELLONIAN UNIVERSITY
IN KRAKÓW

If you are interested in **positions** or **joint ML projects** please contact me:
andrzej@cern.ch

Advertisement

MCnet Summer School and Cracow School of Theoretical Physics

Jun 19 – 25, 2022
Zakopane, Poland
Europe/Warsaw timezone

Enter your search term

- Overview
- Timetable
- Registration
- Cracow School of Theoretical Physics
- MCnet Schools
- Accommodation
- Travel

Local Organisers
zakopane@th.if.uj.edu.pl

The 15th MCnet school and 62nd Cracow School of Theoretical Physics will be held in Zakopane, near Cracow, in Poland.

The school provides a five day course of training in the physics and techniques used in modern Monte Carlo event generators via a series of lectures, practical sessions, and discussions with event-generator authors. The school is aimed at advanced doctoral students and early-career postdocs.

Our core sessions comprise a series of introductory lectures on the physics of event generators, further lectures on a wider range of associated topics, a series of hands-on tutorials using all of the MCnet event generators for LHC physics, and evening discussion sessions with Monte Carlo authors.

The full list of lectures is:

- **Introduction to Event Generators** - Leif LÖNNBLAD (Lund University)
- **Parton Shower, Matching and Merging** - Simon PLÄTZER (University of Graz)
- **The future and challenges of HEP** - Michelangelo MANGANO (CERN)
- **Aspects of the EW Standard Model** - Jonas LINDERT (The University of Sussex)
- **Monte Carlo simulation of FCCee physics** - Staszek JADACH (IFJ PAN Krakow)
- **Model-independent measurements** - Jon BUTTERWORTH (University College London)
- **Highlights from Run 2 of the LHC** - Pawel BRÜCKMAN DE RENSTROM (IFJ PAN Krakow)
- **Machine Learning in HEP** - Ramon WINTERHALDER (CP3, Louvain-la-Neuve)
- **Industrial applications** - Albrecht KYRIELEIS, (Jacobs Manchester)

Tutorial:

- **Tutorial coordinator** - Christian GUTSCHOW (University College London)

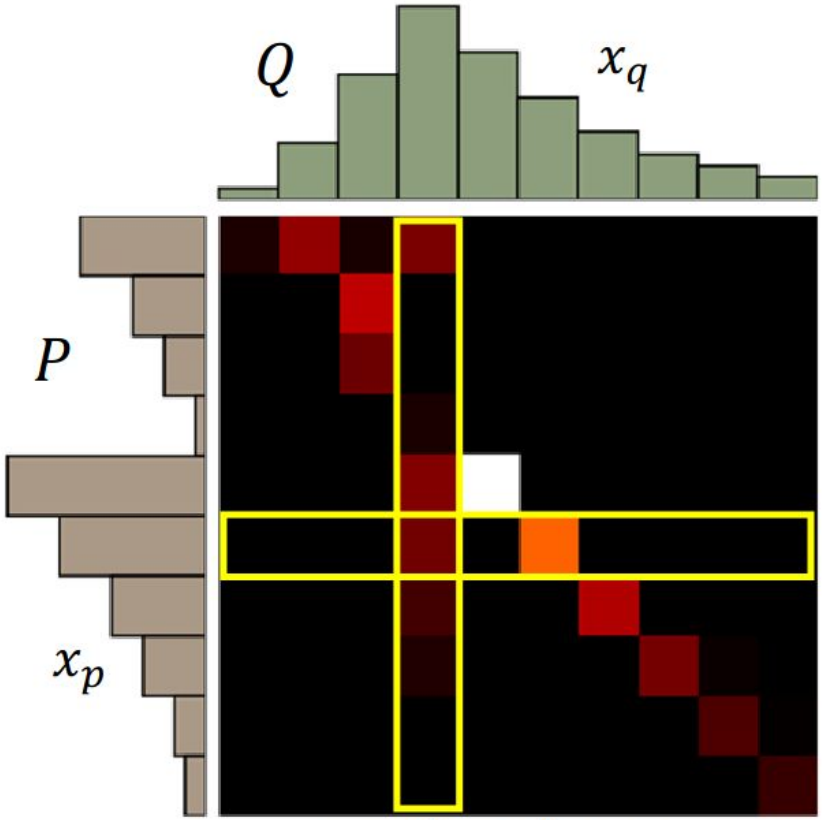
Organised by:

MCnet — Cracow School of Theoretical Physics

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.

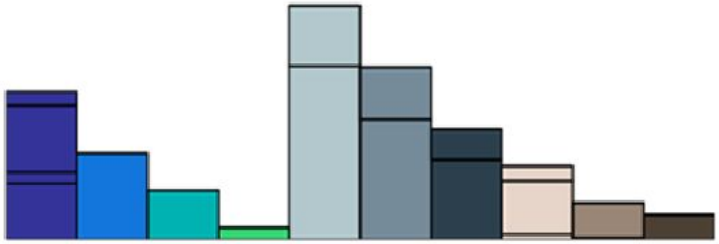
Average distance of a plan γ:

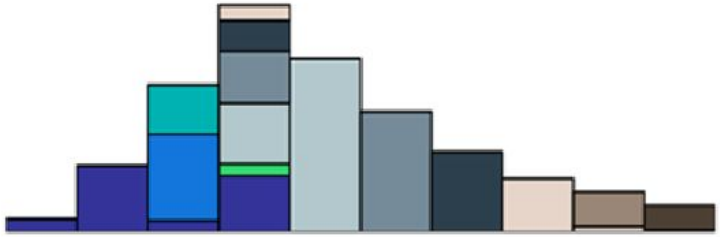
$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$

Earth Mover's Distance:

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

The best plan





Minimax Loss

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

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

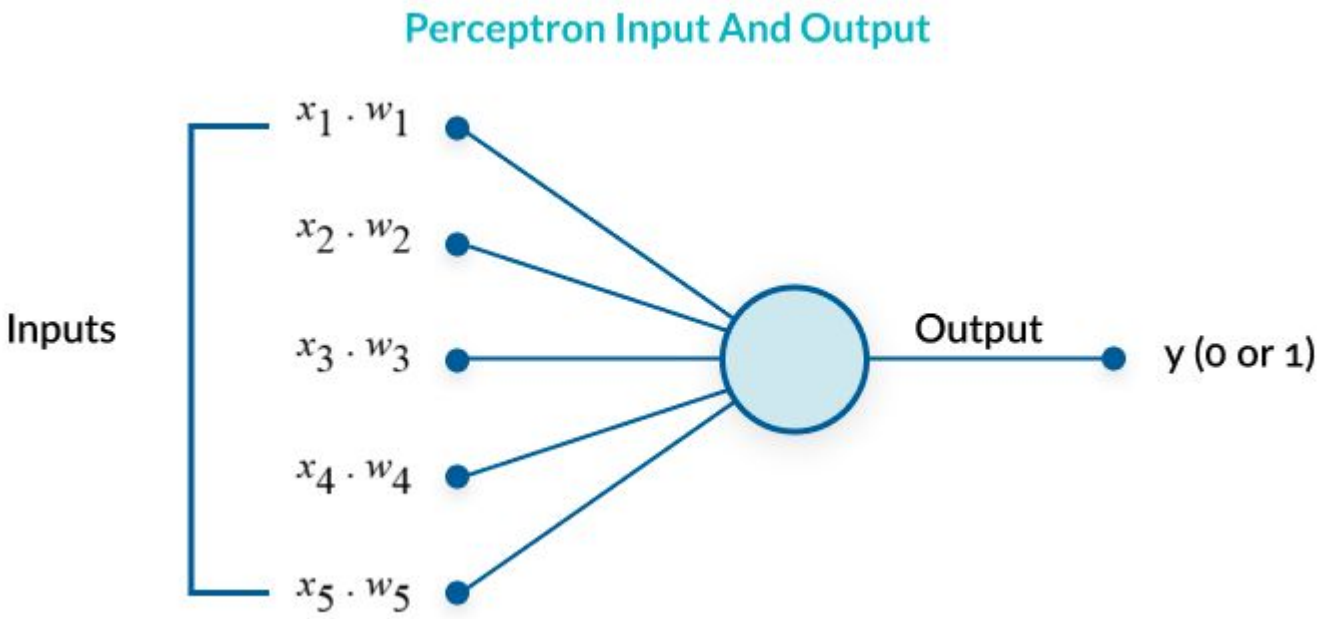
In this function:

- `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)))`.

Perceptron

A perceptron is a simple binary classification algorithm, proposed by Cornell scientist Frank Rosenblatt. It helps to divide a set of input signals into two parts—“yes” and “no”. But unlike many other classification algorithms, the **perceptron was modeled** after the essential unit of the human brain—the **neuron.**



Perceptron

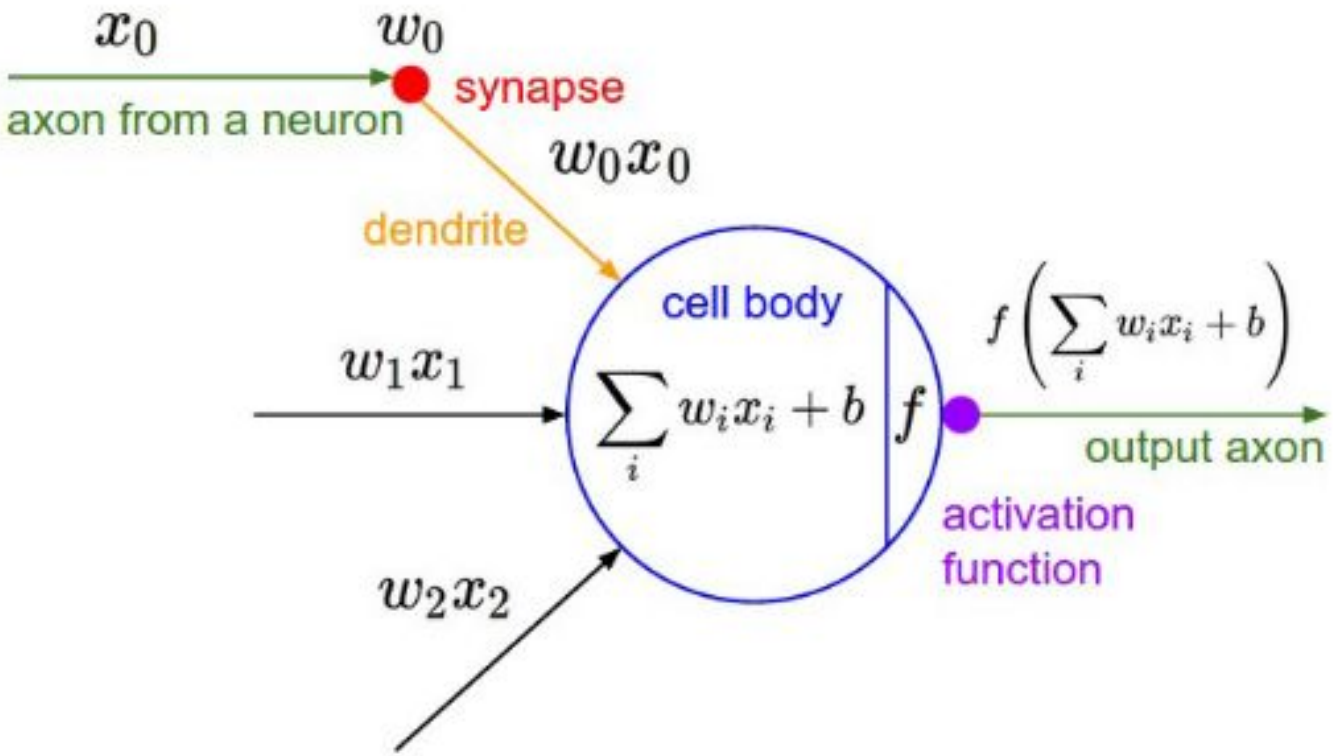
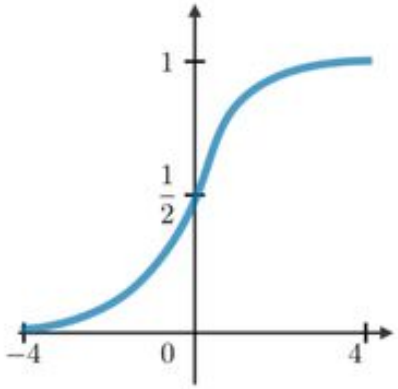
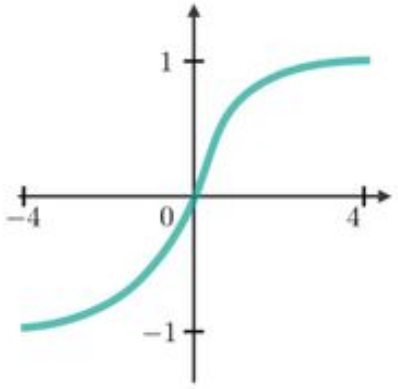
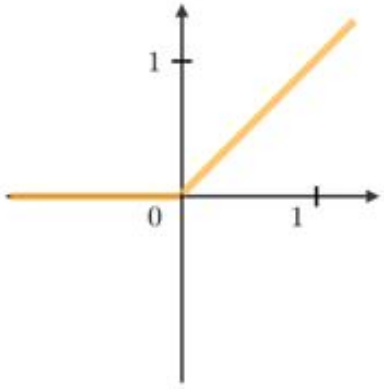
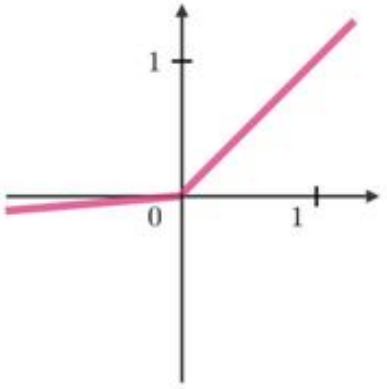


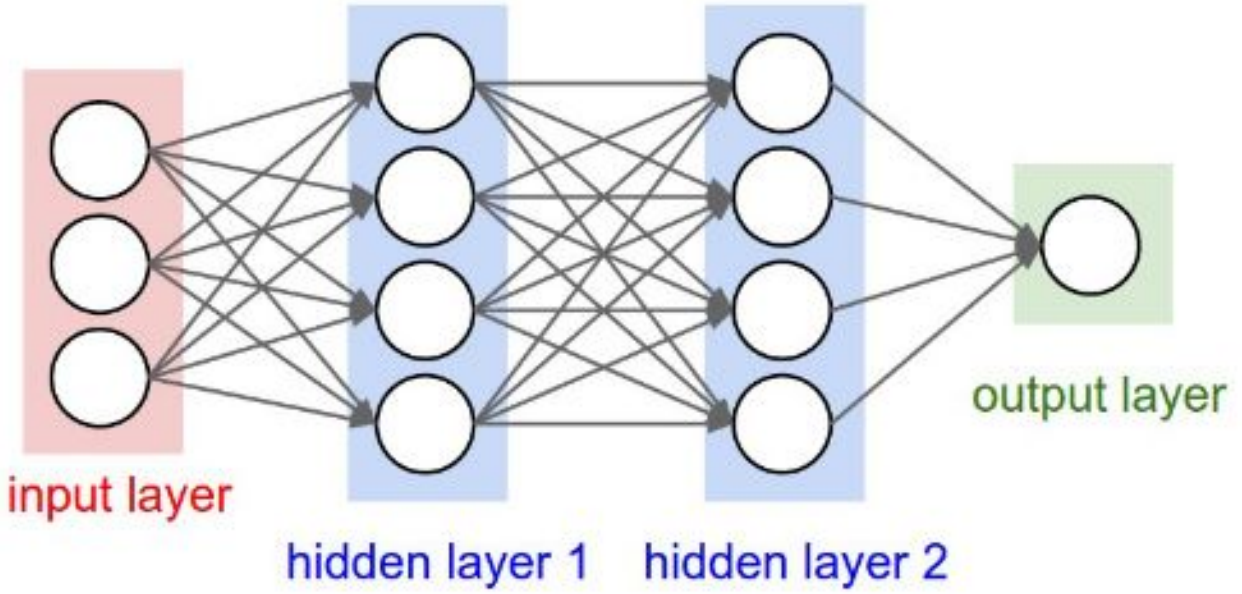
Figure : A mathematical model of the neuron in a neural network

Activation functions

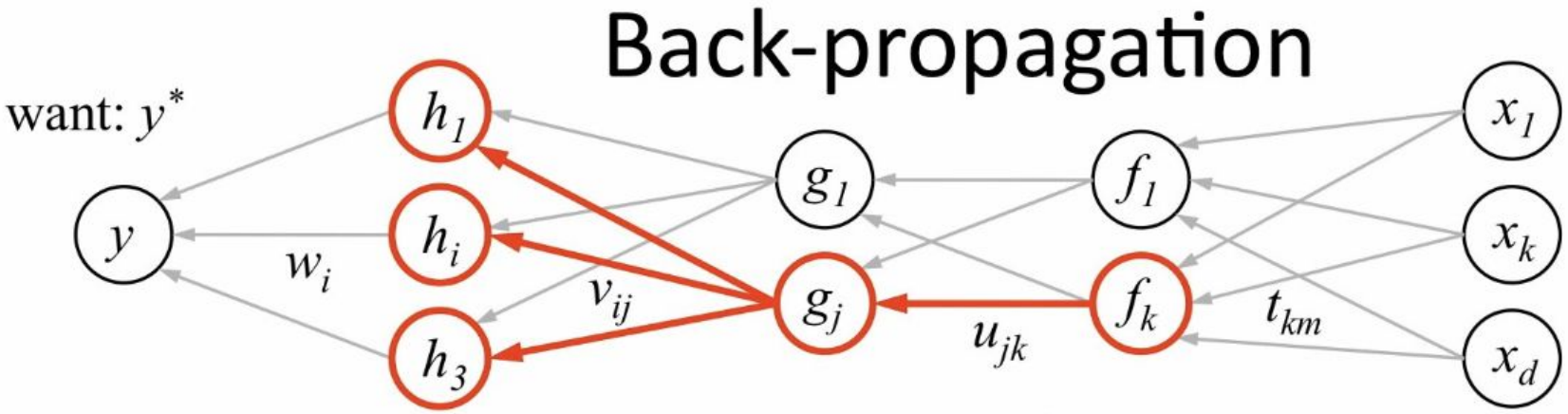
Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$
			

Multilayer Perceptron

A multilayer perceptron (MLP) is a perceptron that teams up with additional perceptrons, stacked in several layers, to solve complex problems.



One difference between an MLP and a neural network is that in the classic perceptron, the decision function is a step function and the output is binary. In neural networks that evolved from MLPs, **other activation functions can be used** which result in outputs of real values, usually between 0 and 1 or between -1 and 1.



1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target y^*
2. **feed forward:** for each unit g_j in each layer 1…L
   compute g_j based on units f_k from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction y and error $(y\text{-}y^*)$
4. **back-propagate error:** for each unit g_j in each layer L…1

(a) compute error on g_j

$$\frac{\partial E}{\partial g_j} = \sum_i \sigma'(h_i) v_{ij} \frac{\partial E}{\partial h_i}$$

- $\frac{\partial E}{\partial g_j}$: should g_j be higher or lower?
- $\sigma'(h_i) v_{ij}$: how h_i will change as g_j changes
- $\frac{\partial E}{\partial h_i}$: was h_i too high or too low?

(b) for each u_{jk} that affects g_j

(i) compute error on u_{jk}

$$\frac{\partial E}{\partial u_{jk}} = \frac{\partial E}{\partial g_j} \sigma'(g_j) f_k$$

- $\frac{\partial E}{\partial g_j}$: do we want g_j to be higher/lower
- $\sigma'(g_j) f_k$: how g_j will change if u_{jk} is higher/lower

(ii) update the weight

$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$