Preamble

This talk was prepared following two of the many things that Marco though me

Preamble

This talk was prepared following two of the many things that Marco though me

1. When you have to speak at a talk or conference, you prepare the talk during the flight

Preamble

This talk was prepared following two of the many things that Marco though me

1. When you have to speak at a talk or conference, you prepare the talk during the flight
2. It is OK to cite Star Wars in a physics paper

B tagging with neural network: from LEP to JEDI-net and beyond

Maurizio Pierini

mPP

CERN

CMS
Compact Muon Solenoid

The origins of NN for b tagging

DELPHI Collaboration

DELPHI 92-20 PHYS 159
25 February 1992

B Tagging With Neural Networks
An Alternative Use of Single Particle Information for Discriminating Jet Events[1]

P. Branchini, M. Ciuchini

INFN - Sezione Sanità
Scuola del dottorato di ricerca - Università "La Sapienza" - Roma
Istituto Superiore di Sanità - Physics Laboratory

P. Del Giudice

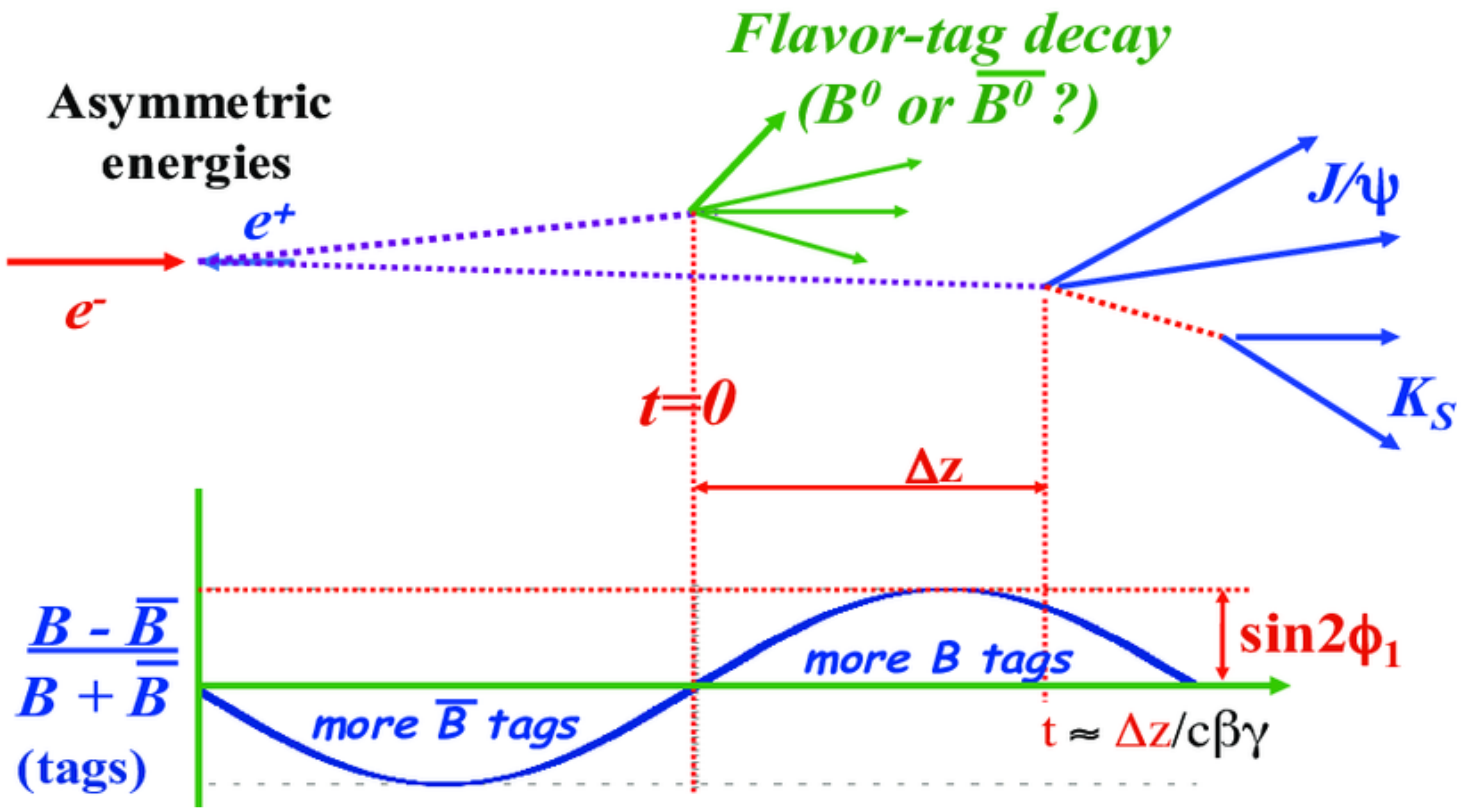
Istituto Superiore di Sanità - Physics Laboratory
INFN - Sezione Sanità

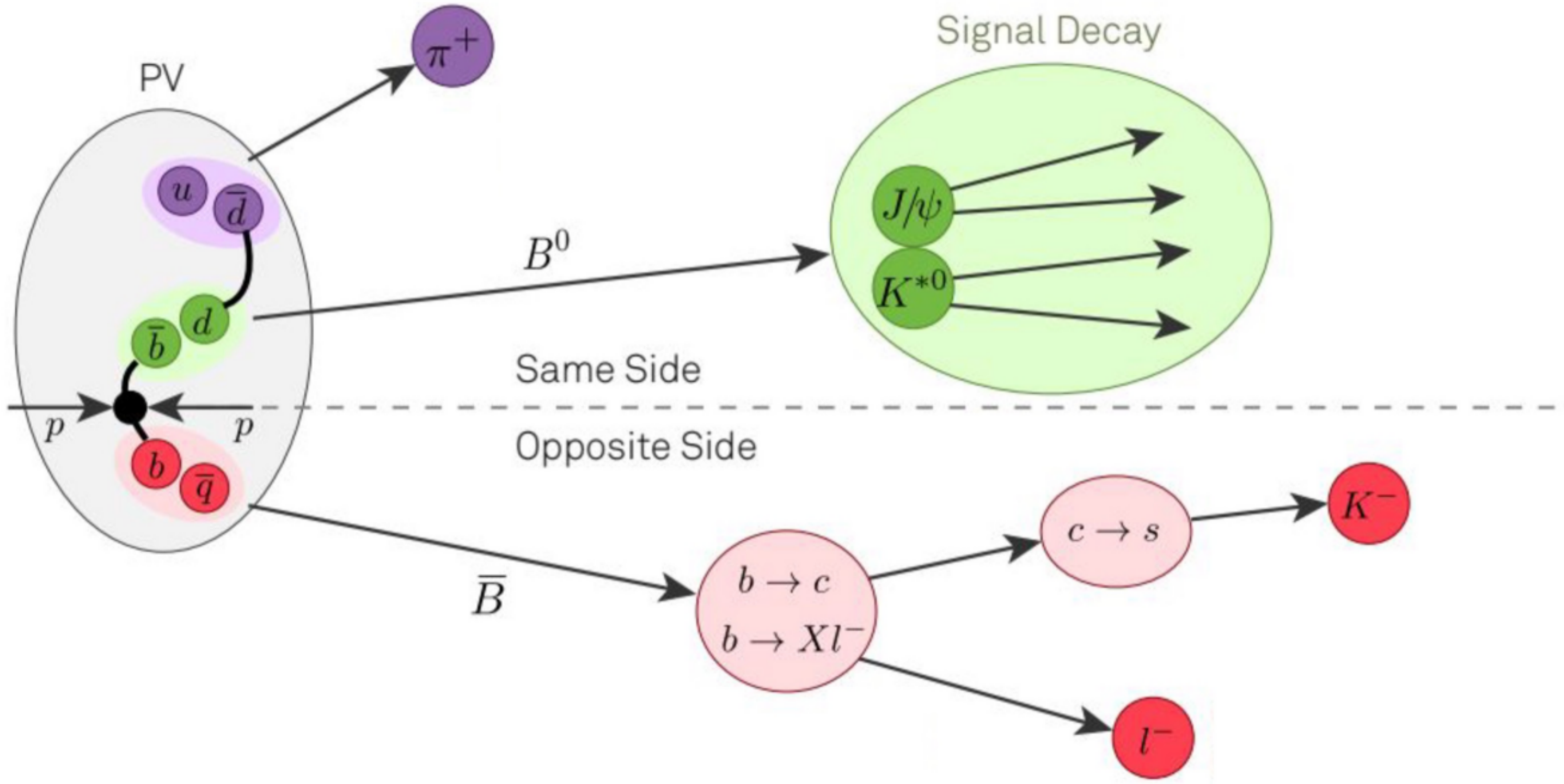
B flavor tagging and b-jet tagging

- *Two different problems:*
  - *B flavor tagging: tell the difference between a B and an anti-B*
  - *b-jet tagging: identify a jet from a b quark, differentiating it from a jet from gluons or light quarks*
- *Traditionally approached as two different problems*

B flavor tagging

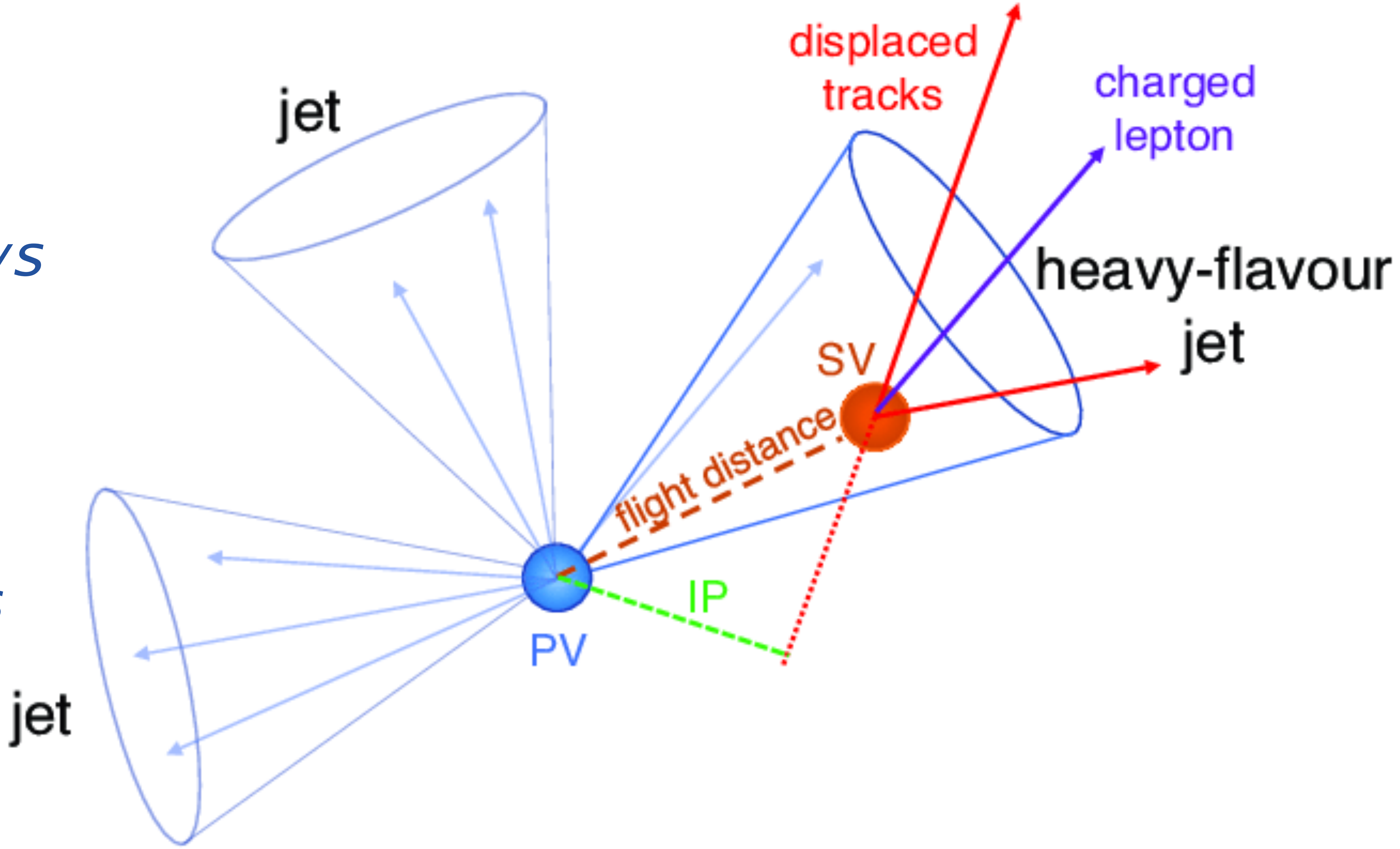
- *B flavour is the essential tool for CP violation studies at B physics experiments*
- *It is ultimately performed measuring the charge of specific particles that correlate to the B meson flavor*
  - *The charge of a lepton from the B vertex*
  - *The charge of a kaon from the B vertex*
  - …
- *With LEP, NNs were introduced to the task*
- *BaBar & Belle inherited this*
- *Similar approaches at hadron colliders, were also same-side tag matters*





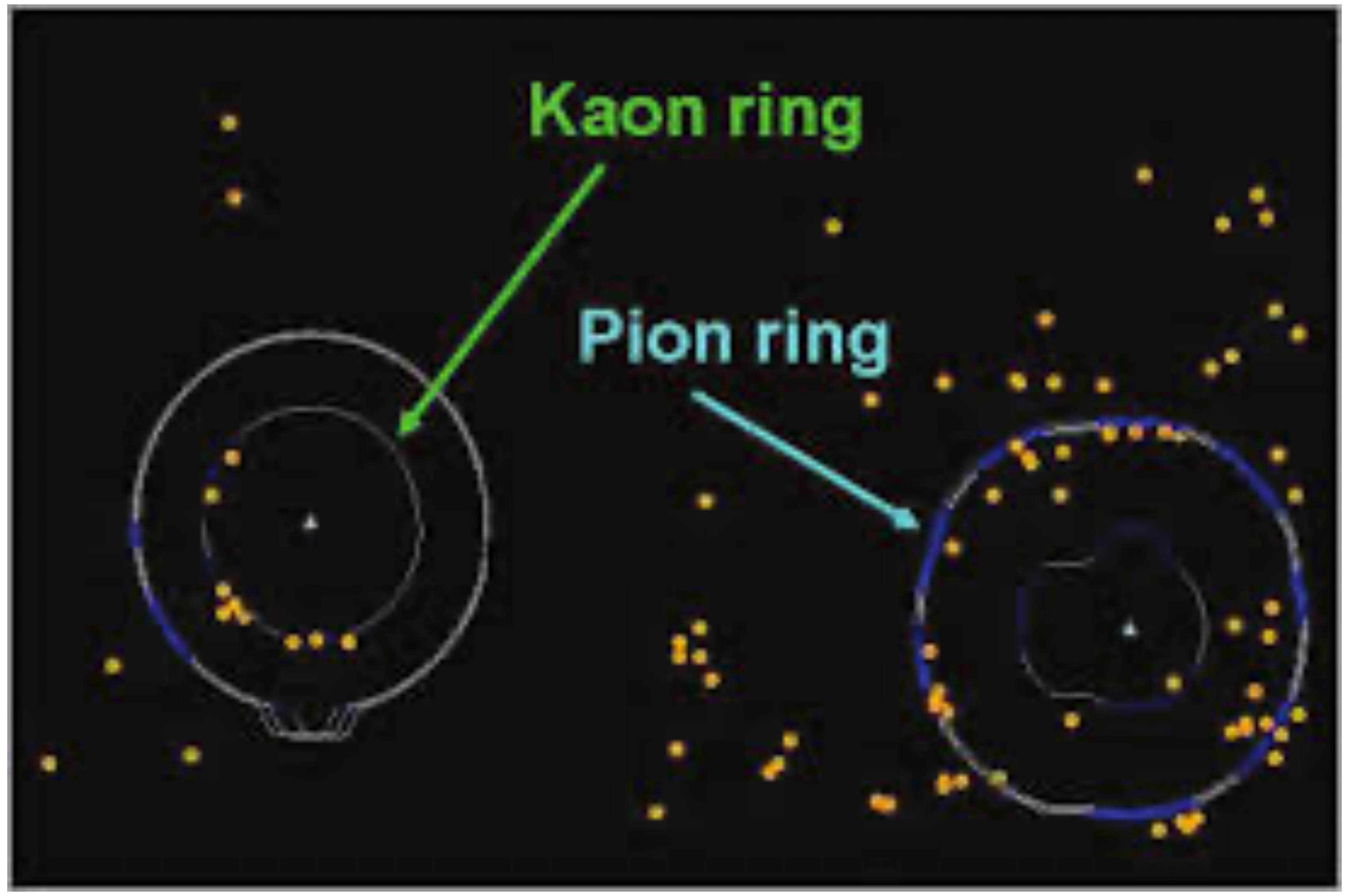
b-jet tagging

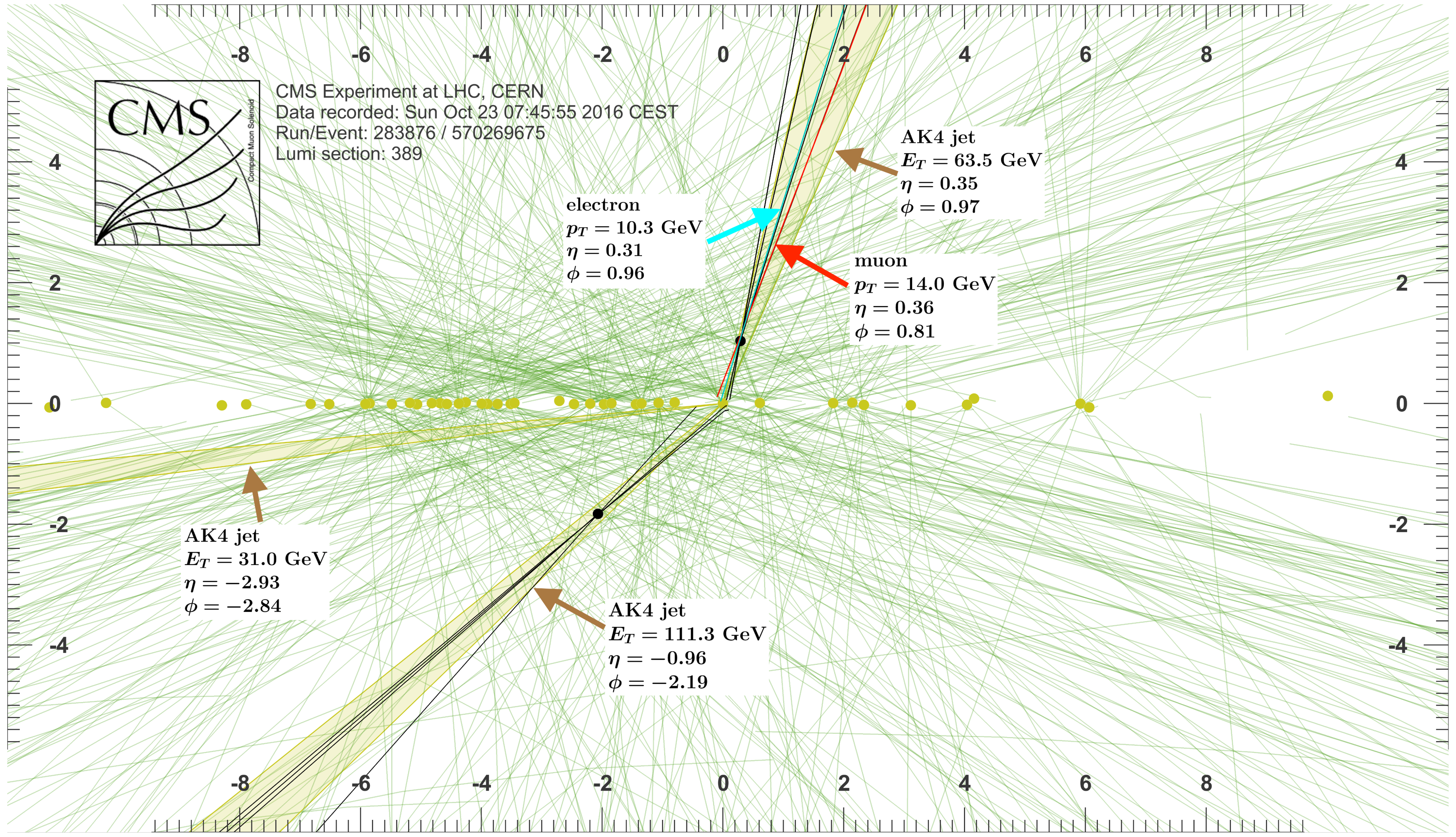
- *Tagging b-jets implies exploiting the secondary vertex*
  - *b fly before decaying*
  - *separation between primary and secondary vertex is unique of b vs gluon of u,d,s jets (charm is in the middle)*
- *Several features are computed from the primary vertex to quantify this signature*
  - *Correlated, but not 1-to-1*
  - *Usually combined in a multivariate analysis (likelihood ratio, neural networks, BDT)*



Different problems, different detector requirements

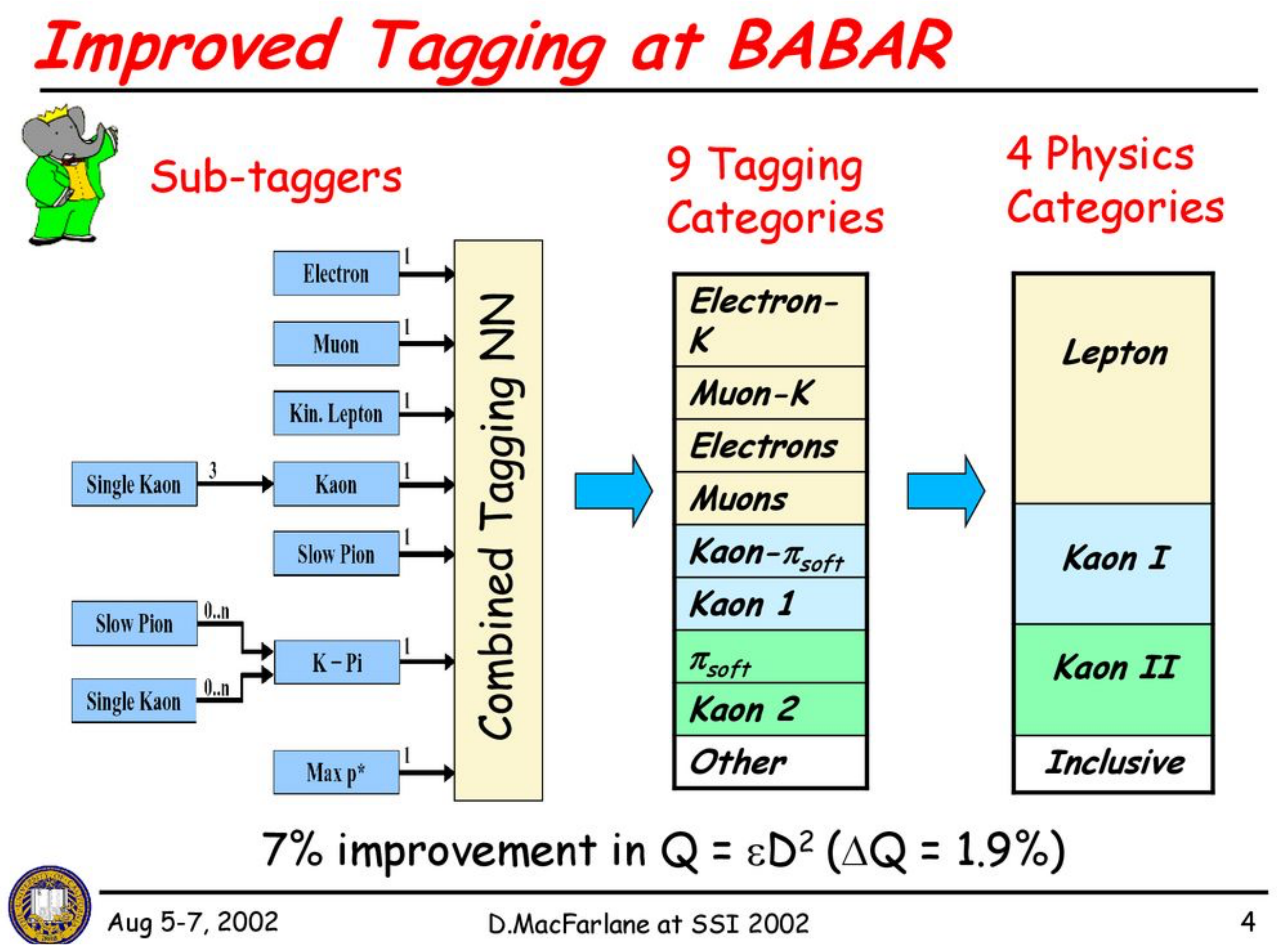
- For a good B flavor tagging one needs a particle identification (e.g., kaon vs pion)
    - LHCb, B factories etc. used Cherenkov detectors
    - ATLAS and CMS don't have PID (yet. It will come @HL-LHC with the timing detectors)
- For a good b-jet tagging one needs a good secondary-vertex resolution, which is also relevant to measure oscillations
    - All modern detectors have a pixel-based inner tracker

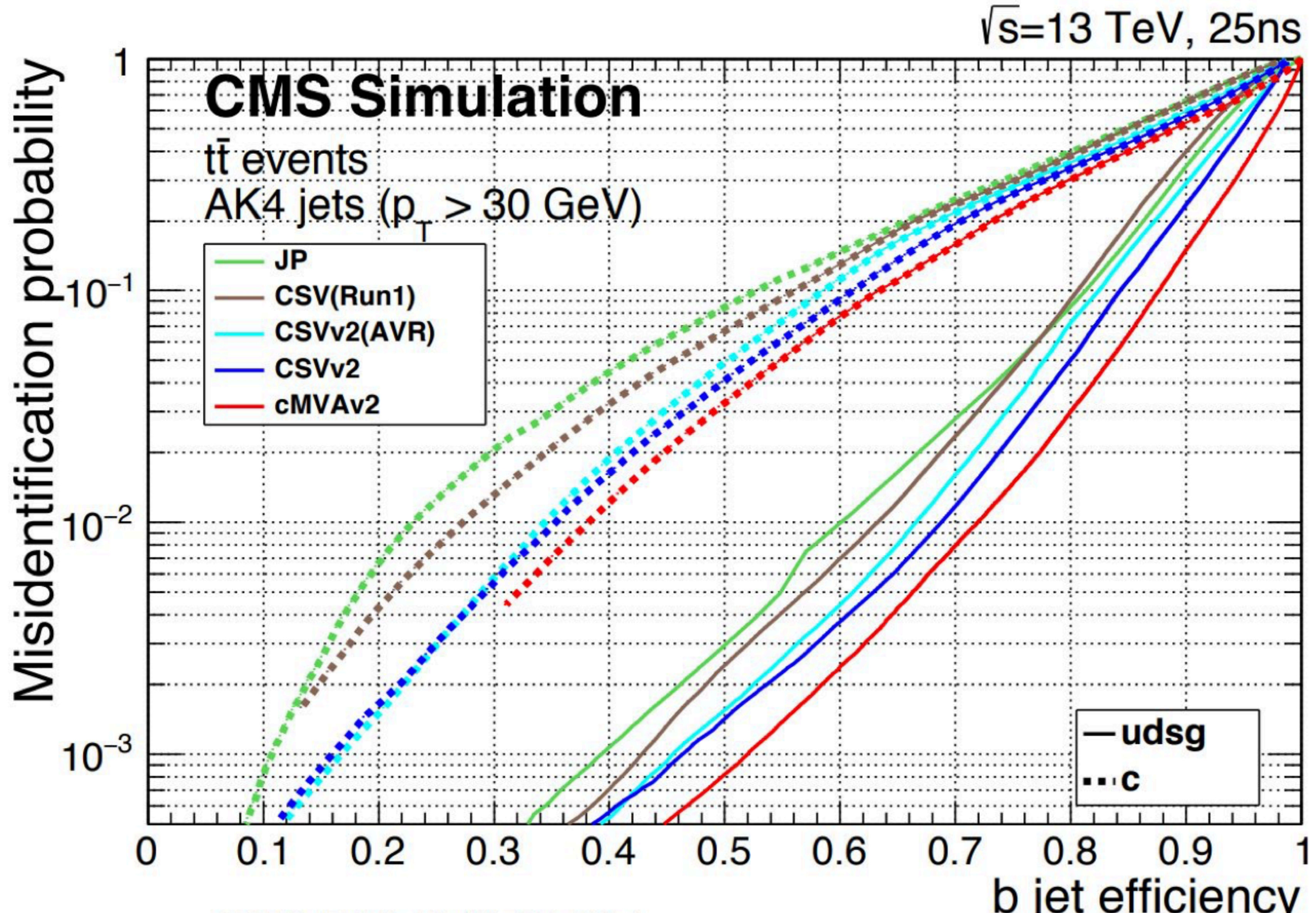




Two problems, one solution

- These two problems have many common points
  - Both problems are binary classifications
  - One engineers several quantities to address each problem separately
  - A multivariate approach exploits the correlations between these variables
- The solutions to these problems evolved according to the same pattern
  - NNs as a first MVA attempt
  - BDTs took over
  - NNs back with Deep Learning
  - Several architectures tried, until solution converged to graph networks

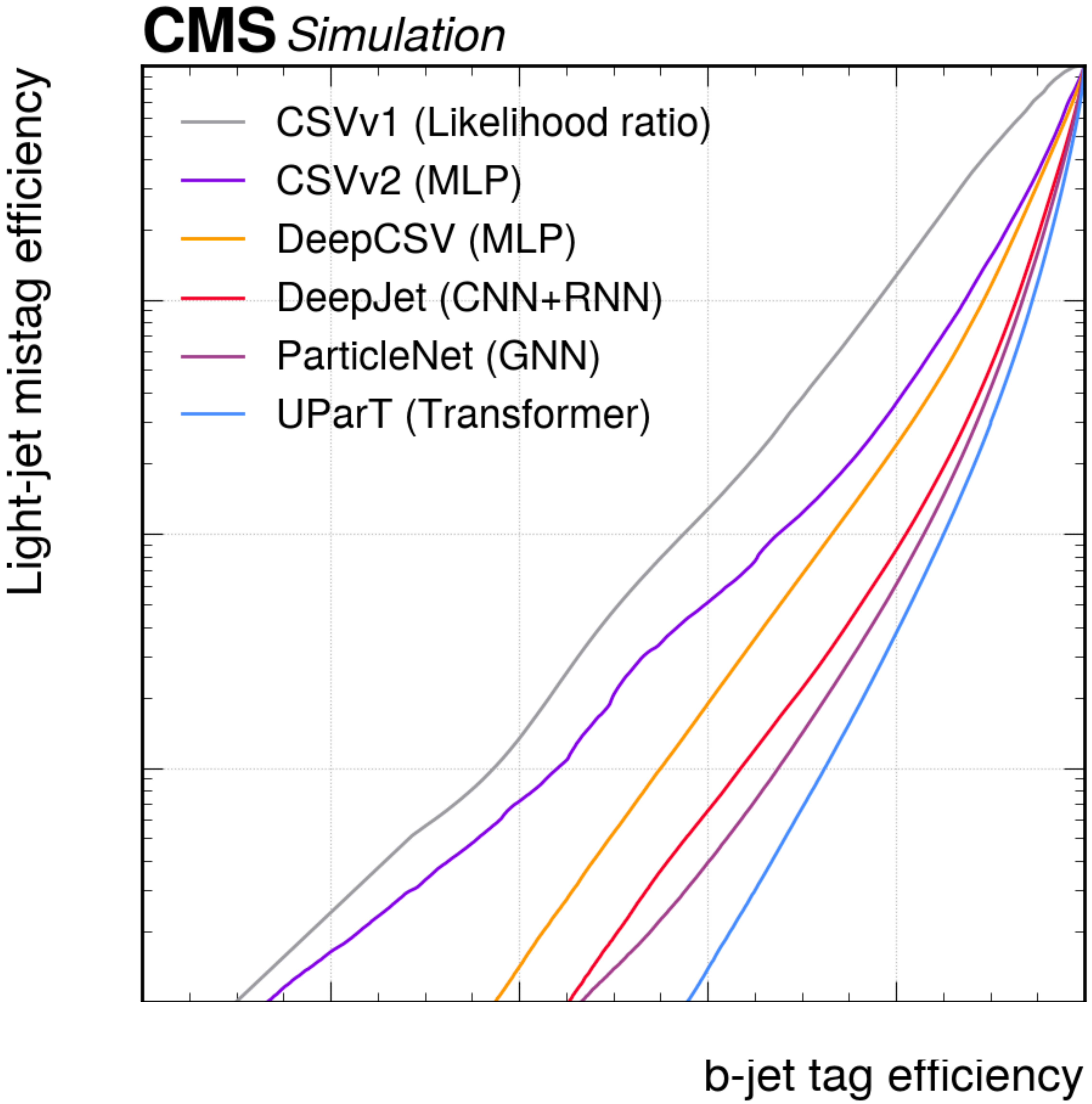




CMS PAS BTV-15-001

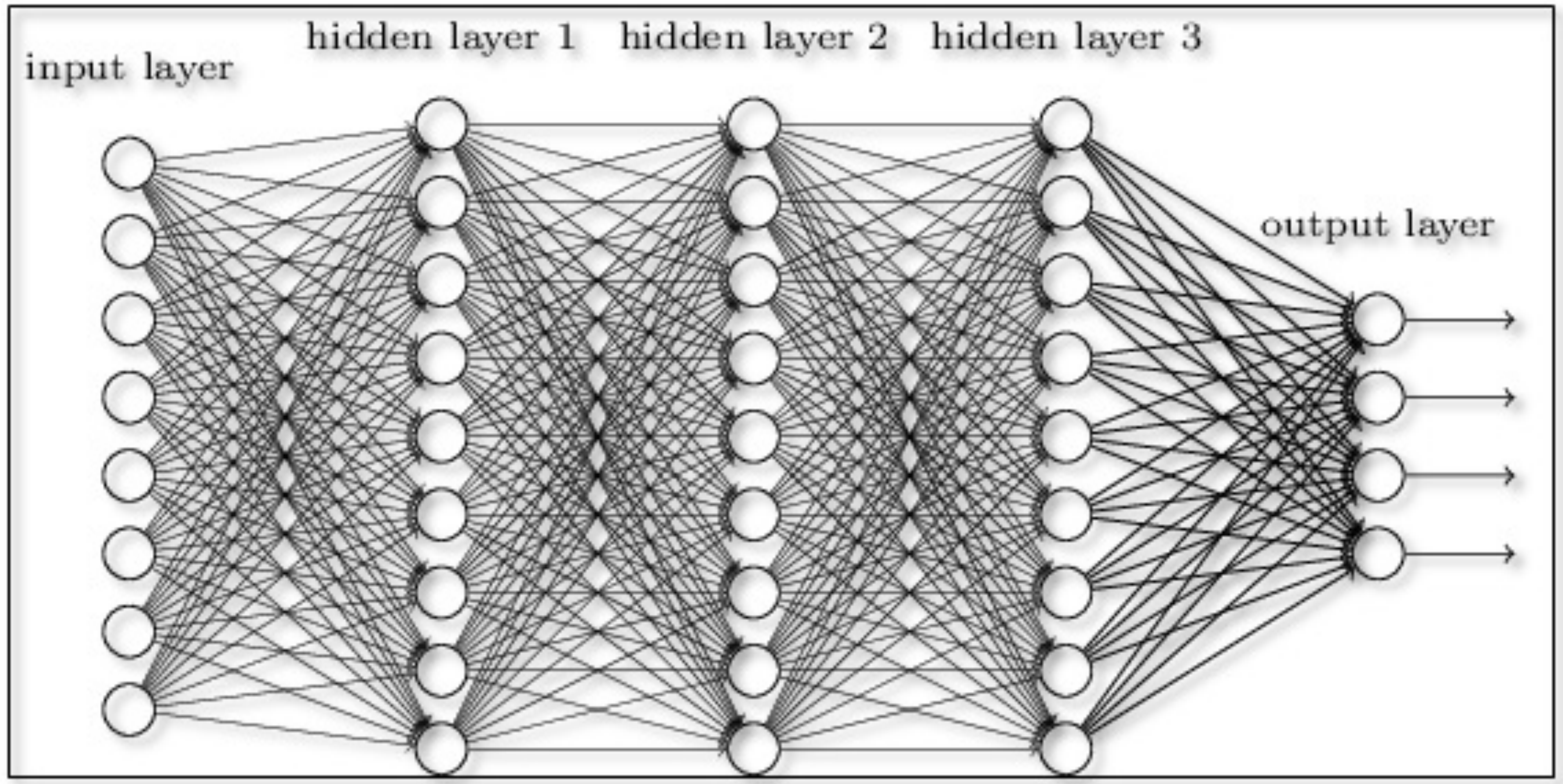
The evolution of b-jet tagging

- *During Run-1, CMS used a combined secondary vertex (a likelihood ratio), similar to what was used at Delphi*
- *Then ML was introduced*
  - *NNs were used early in the game*
  - *There was a BDT parenthesis*
  - *A Deep Neural Network*
  - *A Recurrent network*
  - *The ultimate solution: a graph network*



Deep Neural Networks

- In a feed-forward chain, each node processes what comes from the previous layer
- The final result (depending on the network geometry) is K outputs, given N inputs

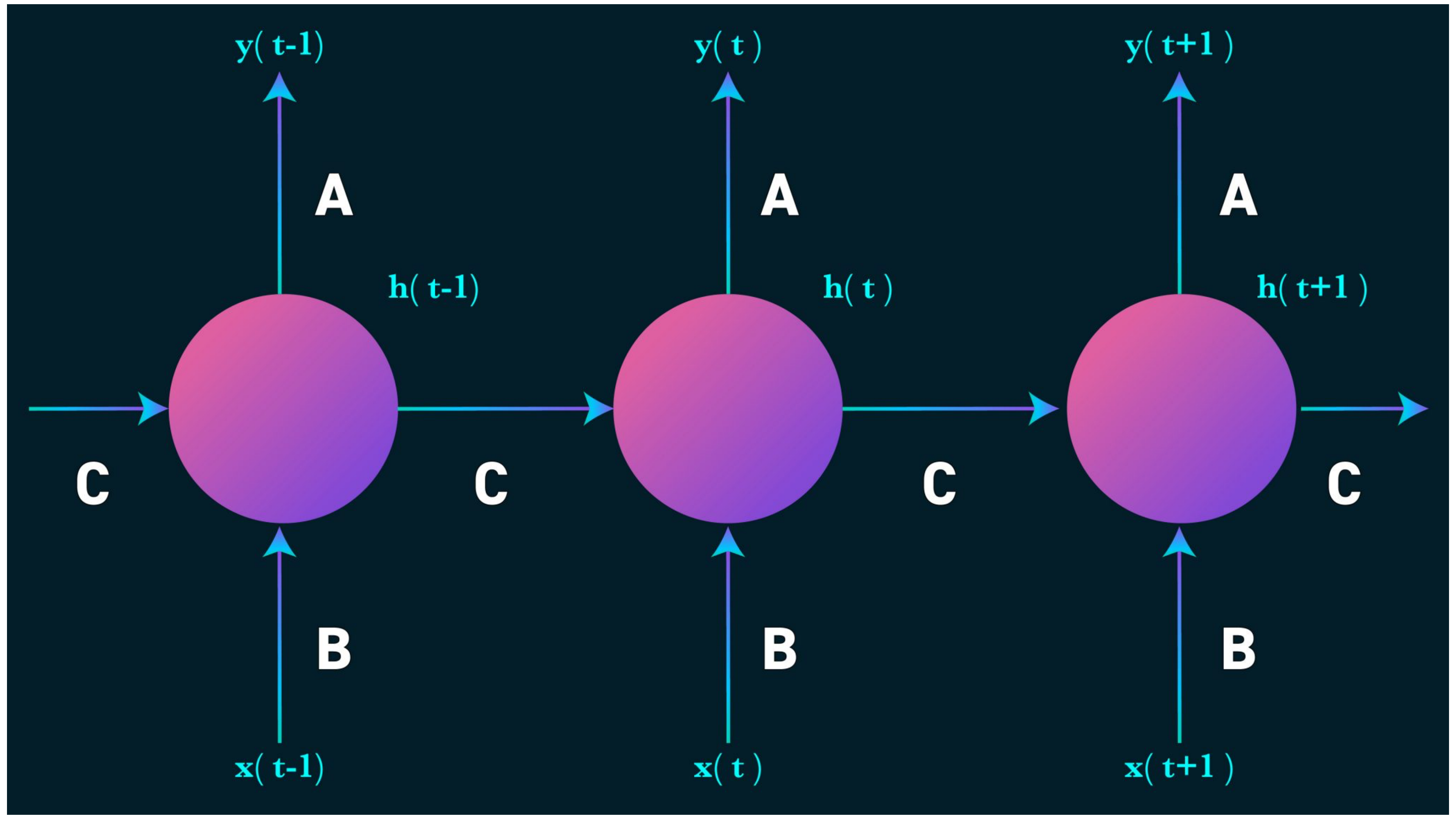


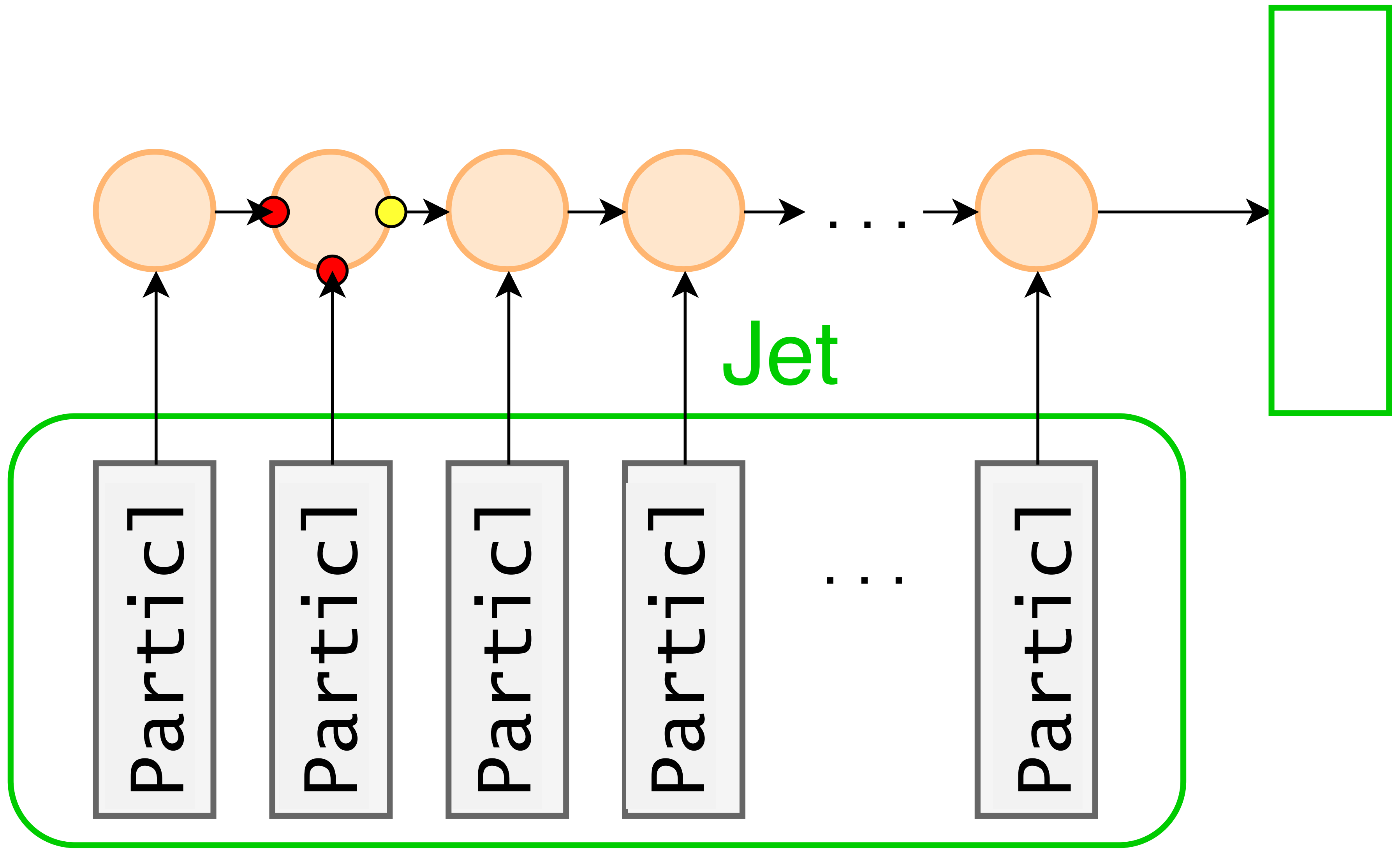
$$y_j = f^{(3)}(\Sigma_l w^{(3)}_{jl} f^{(2)}(\Sigma_k w^{(2)}_{lk} f^{(1)}(\Sigma_i w^{(1)}_{ki} x_i + b^{(k)}_k) + b^{(2)}_l) + b^{(3)}_j)$$

- One can show that such a mechanism allows to learn generic $\mathbb{R}^N \rightarrow \mathbb{R}^K$ functions

Recurrent Neural Networks

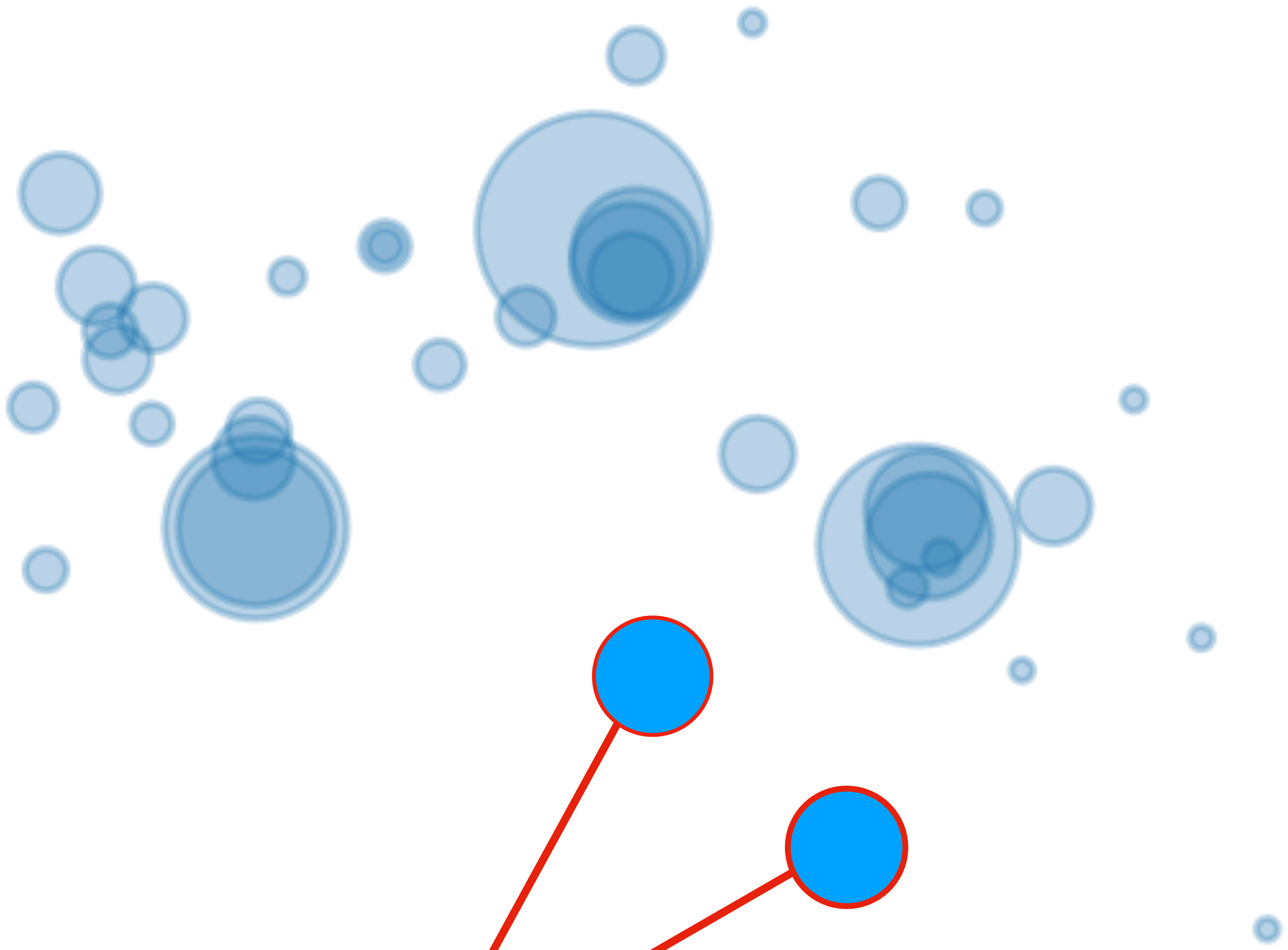
- Recurrent architectures are designed to process sequences of data
- Then idea is to have information flowing in the network while the sequence is sequentially processed
- Through this idea, recurrent networks mimic memory persistence
- It takes as input directly the "raw data" (particle momenta) and it engineers features by itself

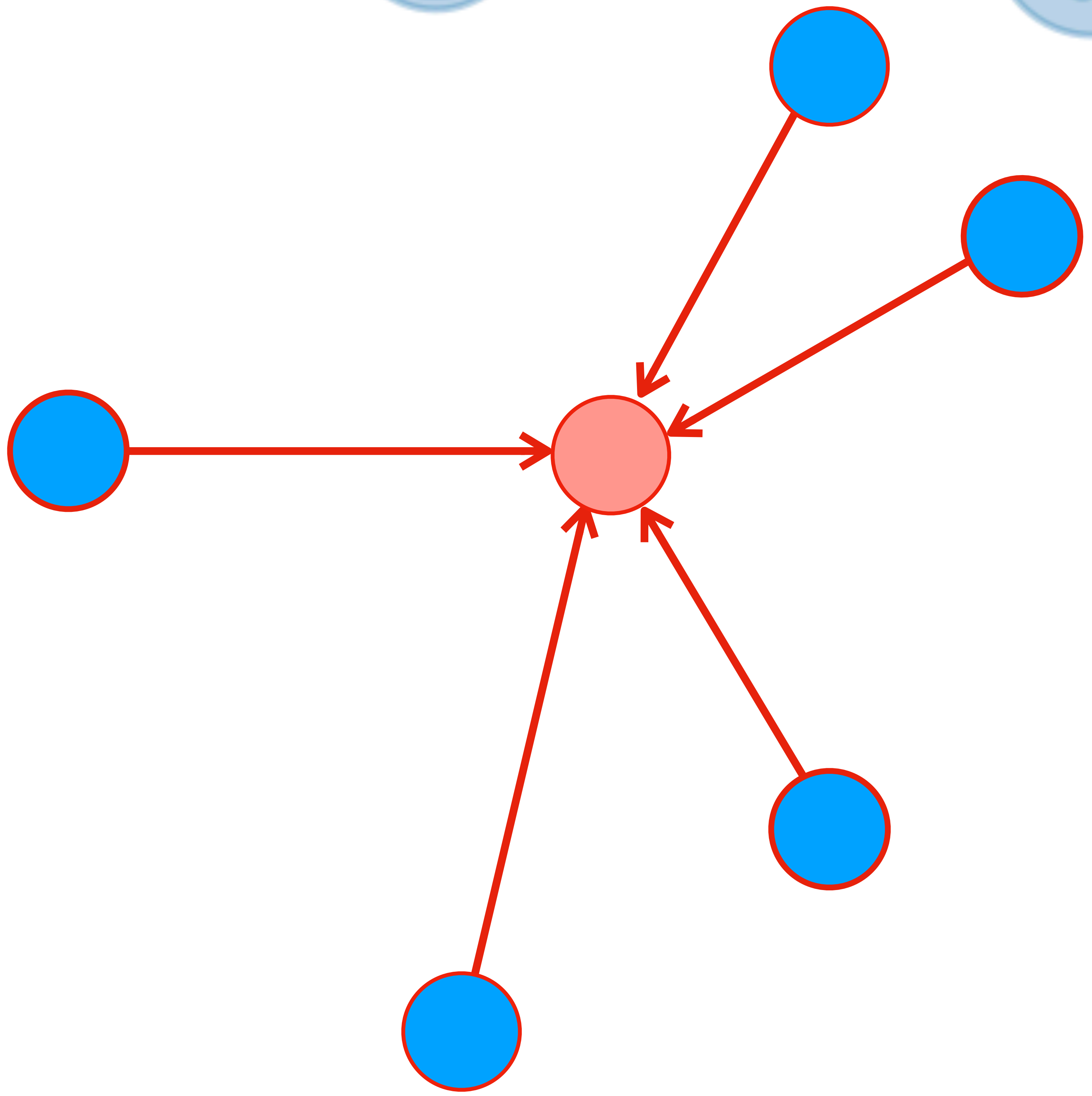




Point clouds and graph nets

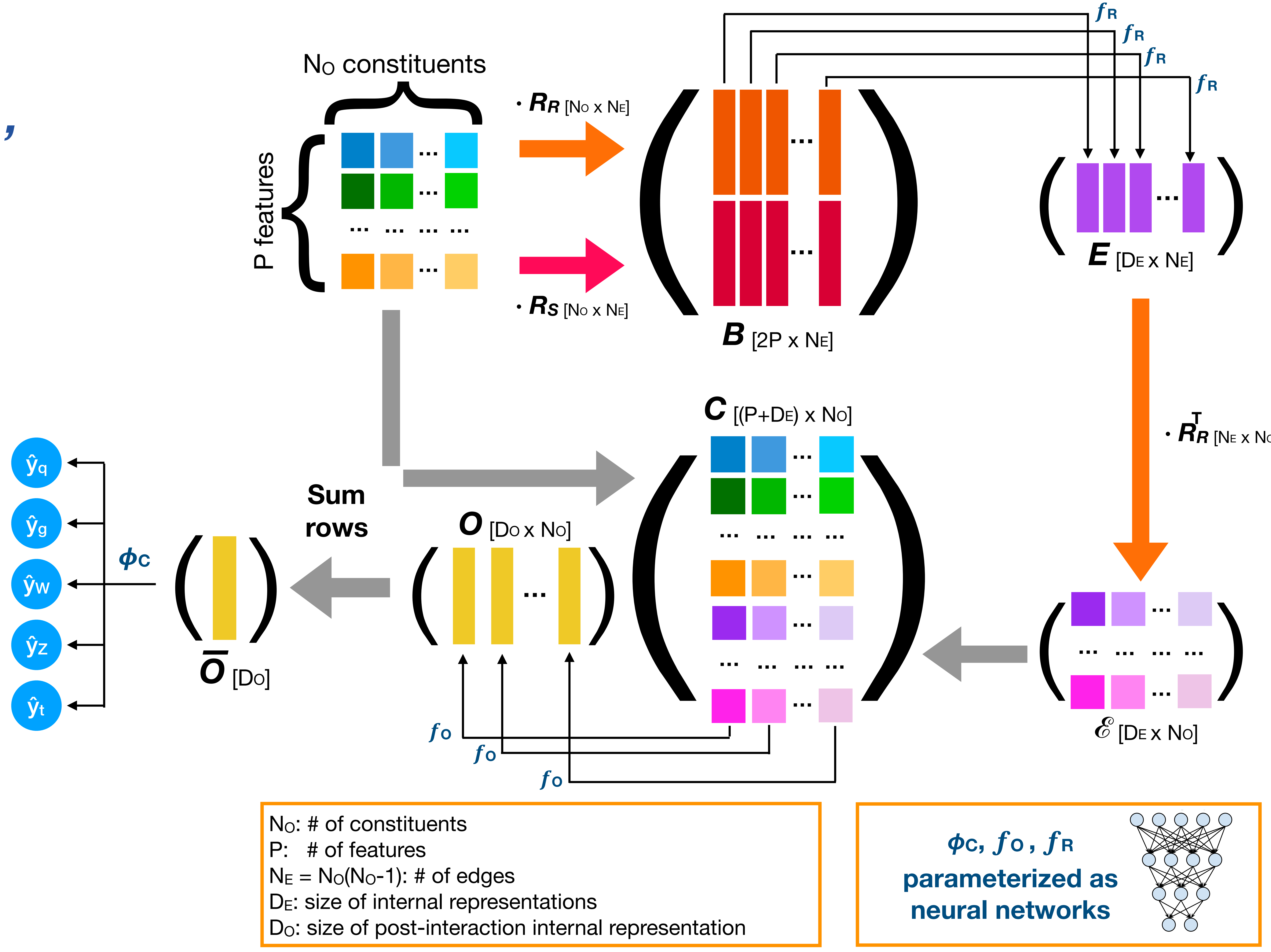
- *Graphs Nets are architectures based on an abstract representation of a given dataset*
  - *Each example in a dataset is represented as a set of vertices*
  - *Each vertex is embedded in the graph as a vector of features*
  - *Vertices are connected through links (edges)*
  - ***Messages are passed through links and aggregated on the vertices***
  - *A new representation of each node is created, based on the information gathered across the graph*





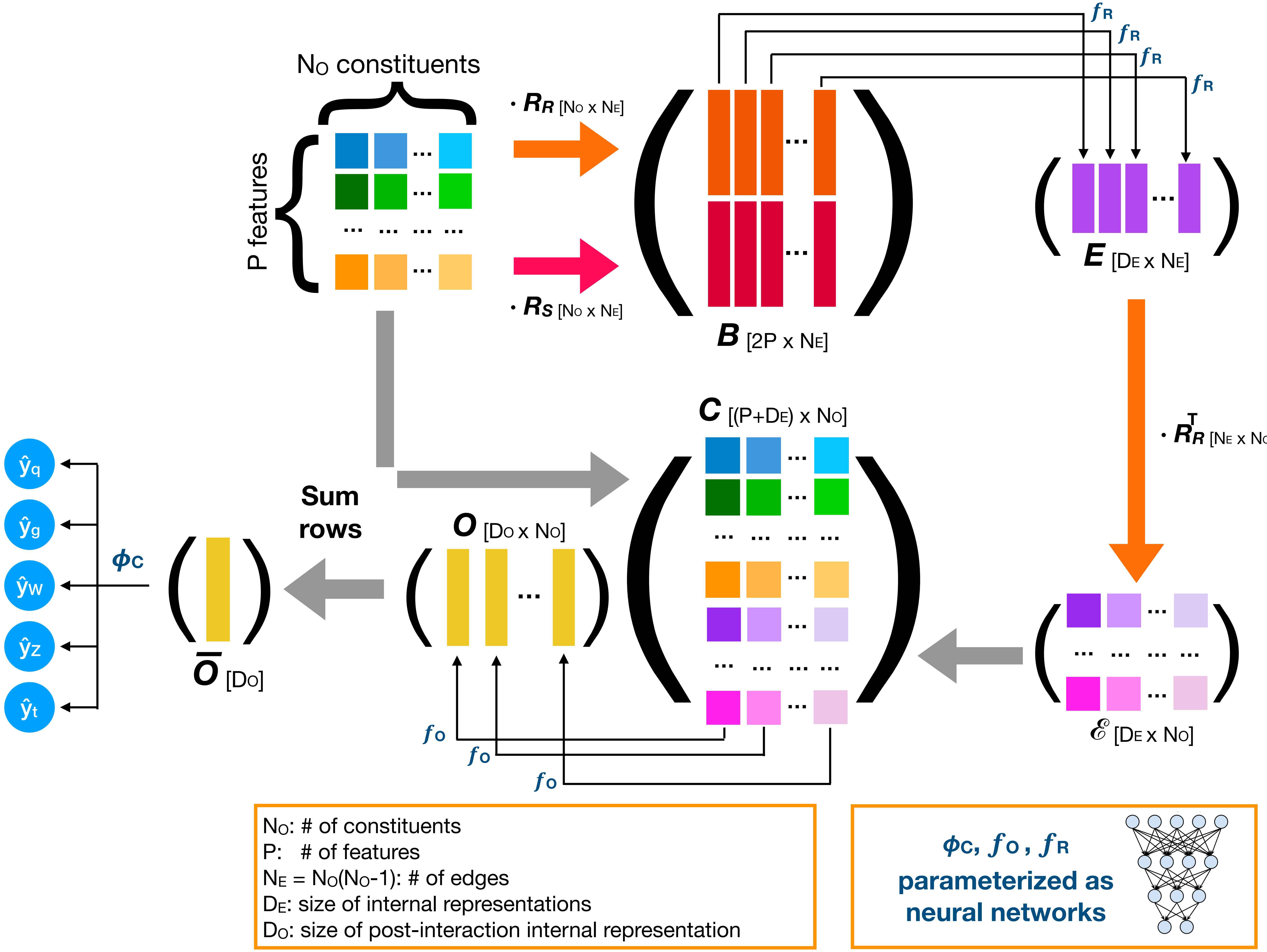
JEDI-net

- INs process a list of No x P inputs in pairs, through Receiving and Sending matrices
- The effect of the interaction is learned by fR and combined wit the input to learn (through fo) a post-interaction representation
- The procedure can then be *iterated* to produce further steps i the interactions

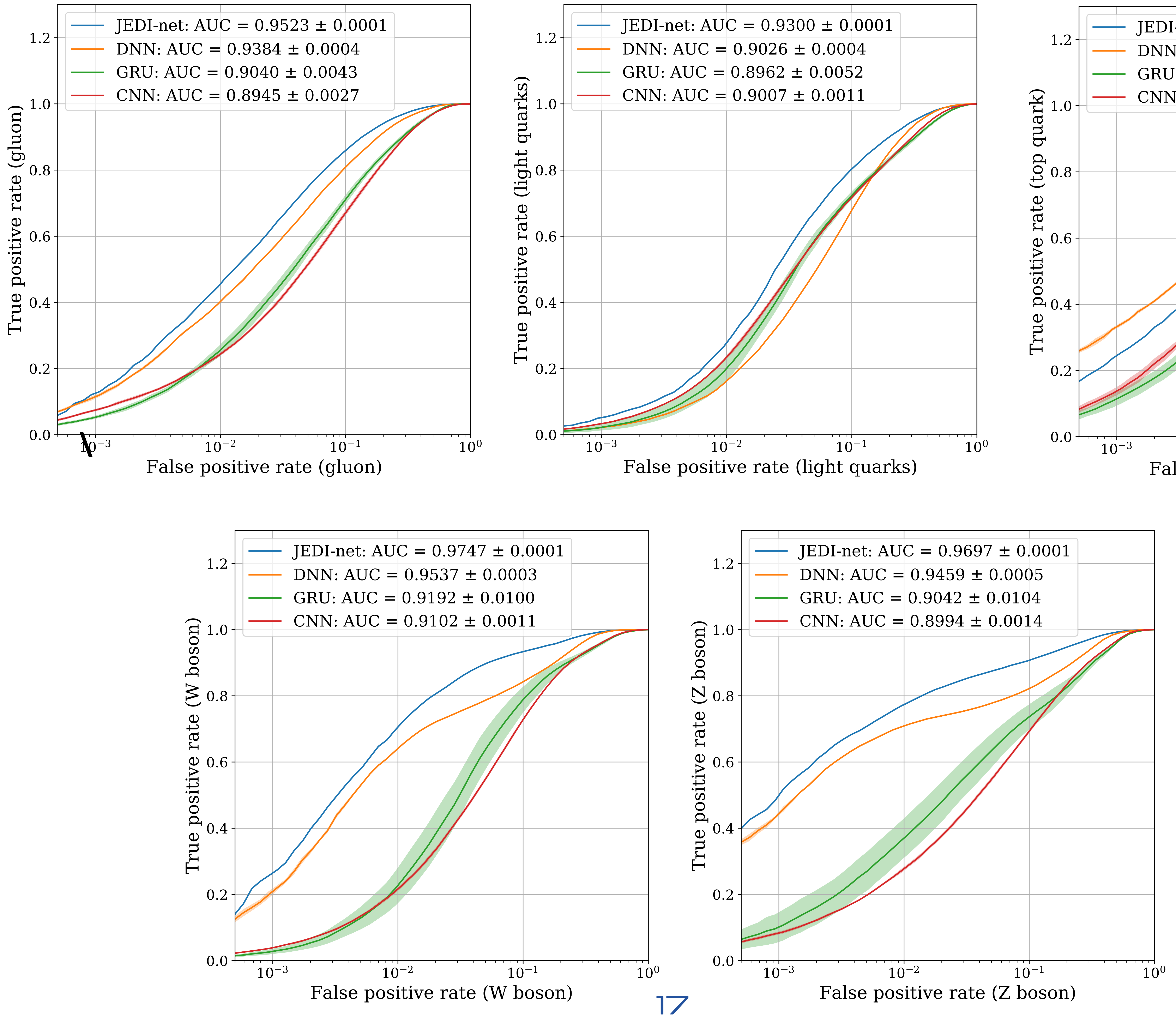


JEDI-net

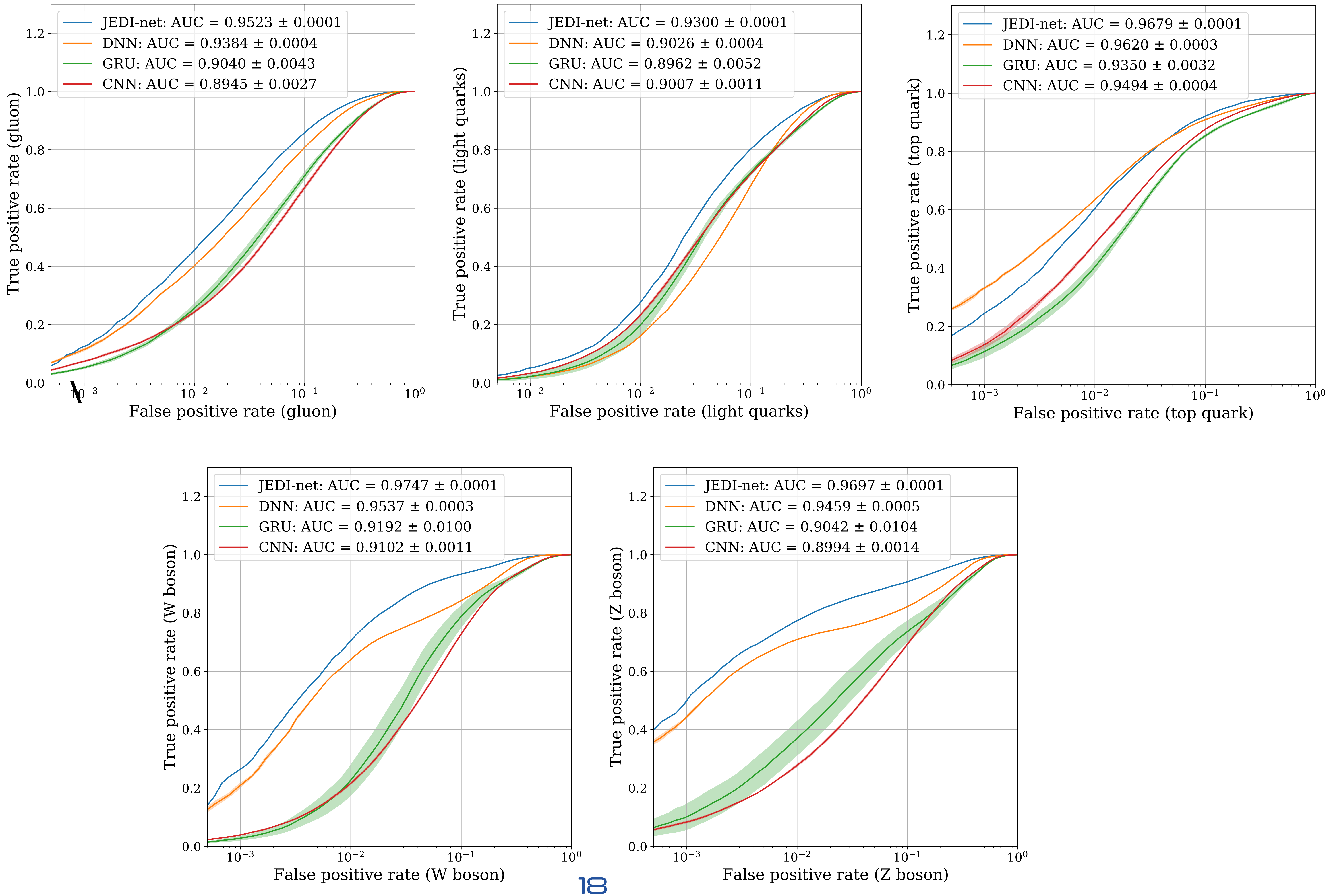
- *In this case, there is no system update needed (i.e., no cycle)*
- *It is sufficient to use the post-interaction representation as input to a classifier that returns the jet category*
- *The three networks are simultaneously optimized: the learned representation is chosen to help the classification*



A comparison

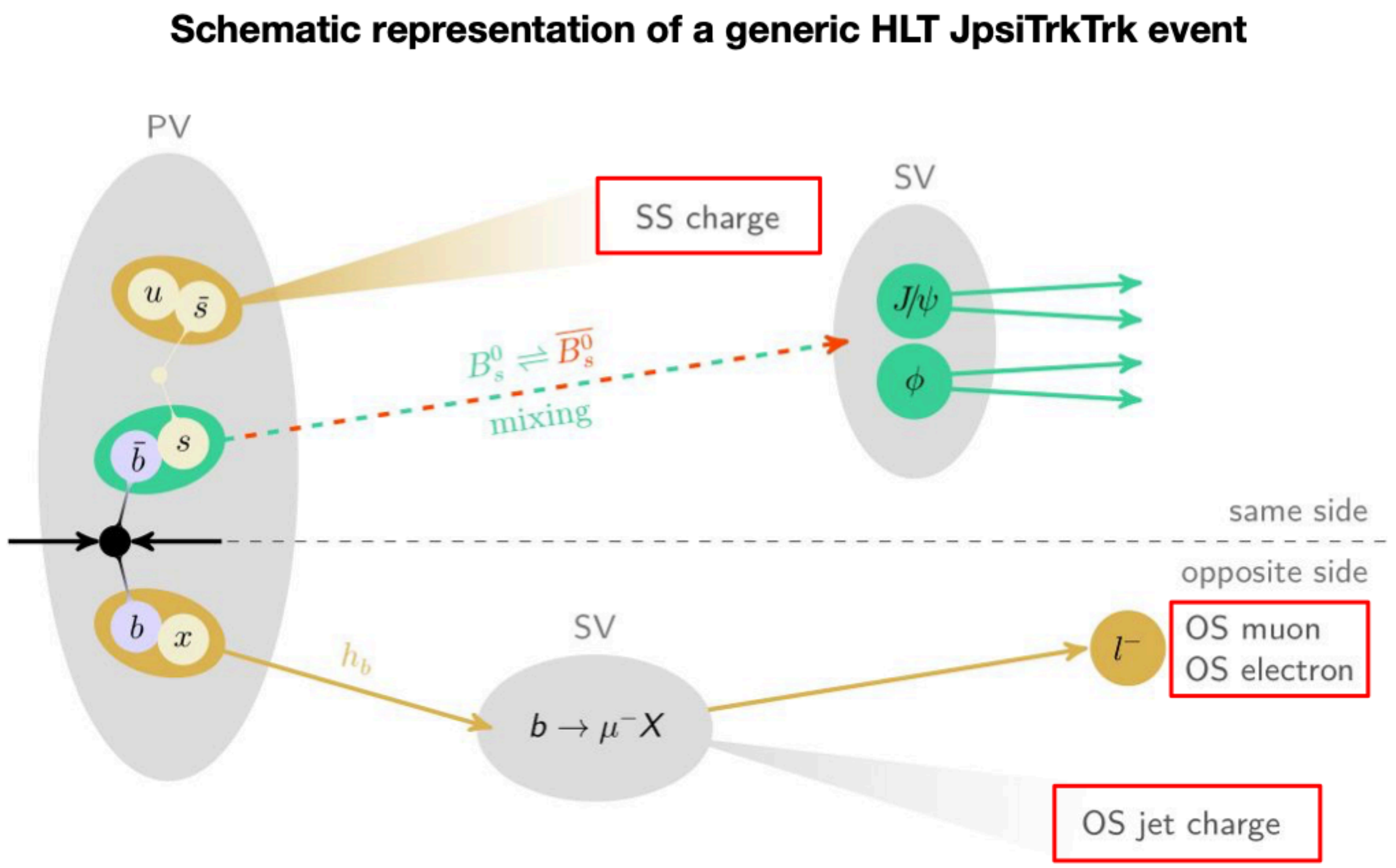


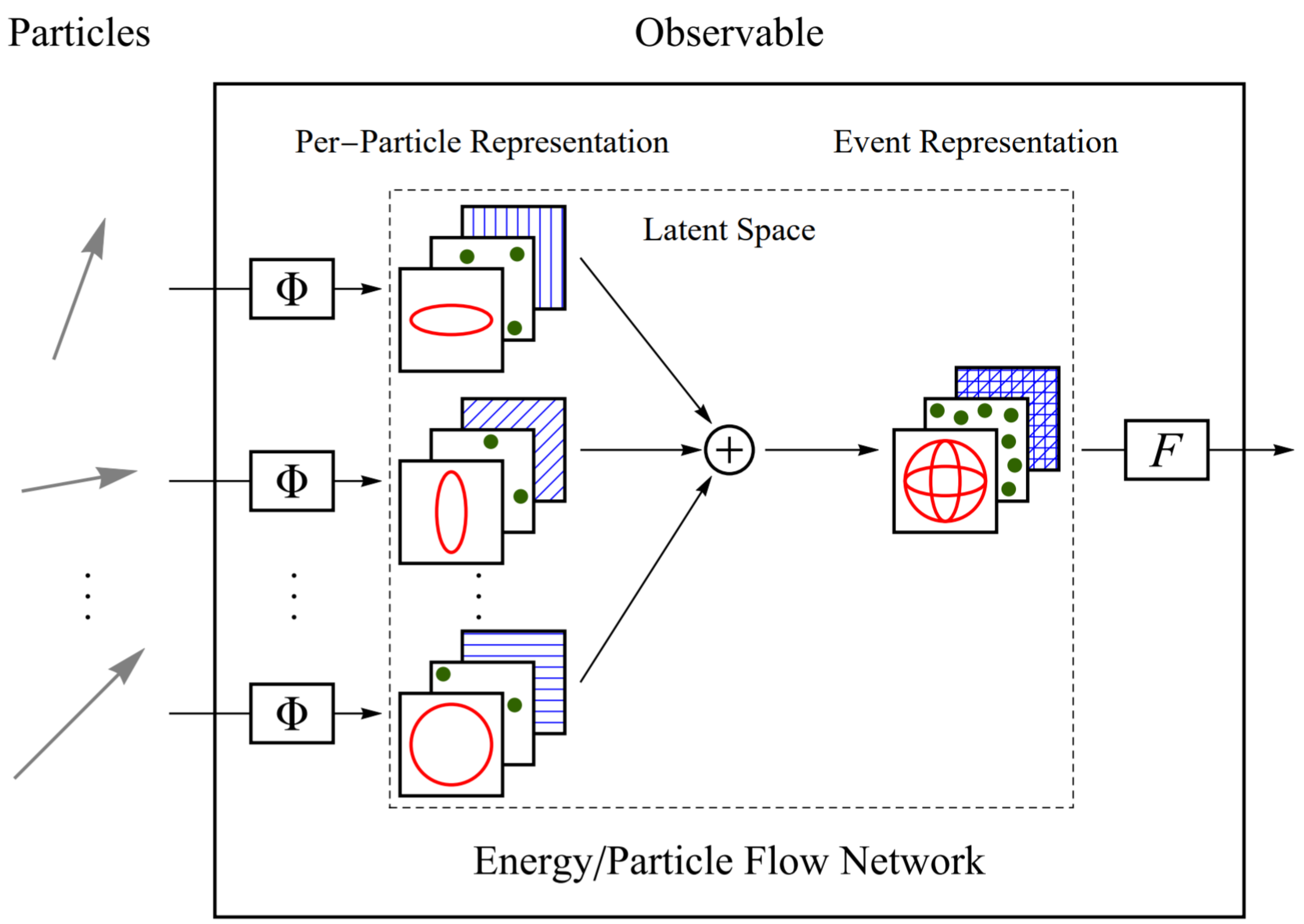
A comparison



Particle Clouds from b-jet to B-flav

- *Recently, CMS applied the same paradigm shift to B-flavor tagging*
  - *Same side (SS): exploits the Bs fragmentation*
    1. SS tagger: leverages charge asymmetries in the Bs fragmentation
  - *Opposite side (OS): exploits decay products of the other*
    1. *b-hadron in the event*
    2. *OS muon: leverages b ➜ μ-X decays*
    3. *OS electron: leverages b ➜ e-X decays*
    4. *OS jet: capitalizes on charge asymmetries in the OS b-jet*
  - *All algorithms are based on DeepSets trained on simulations and calibrated in B+ ➜ J/ψ K+ with special precautions to reduce systematic effects*



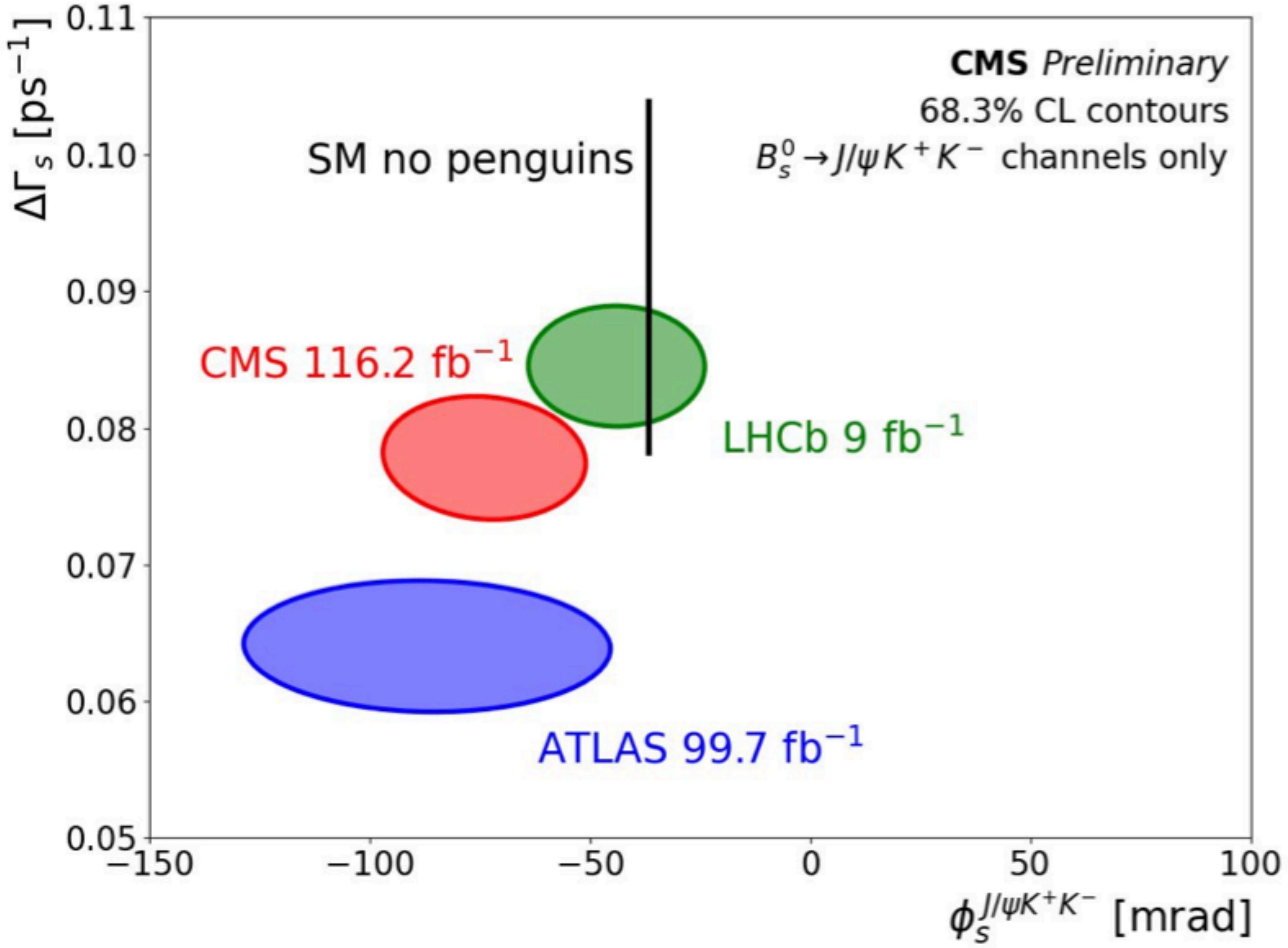


Impact on ϕs measurement

- CMS used this new technology for its measurement of ϕs
- Despite lack of any PID, best tagging performance at hadron collider ever
- Lack of PID is compensated by the information gathered from particles surrounding the B (through point-cloud approach), both on OS and SS
- As a result, very competitive result derived
  - First evidence of CP violation in this channel

Parameter	Fit value	Stat. uncer.	Syst. uncer.
ϕ_s [mrad]	-73	± 23	± 7
$\Delta\Gamma_s$ [ps^{-1}]	0.0761	± 0.0043	± 0.0019
Γ_s [ps^{-1}]	0.6613	± 0.0015	± 0.0028
Δm_s [$\hbar$ps^{-1}]	17.757	± 0.035	± 0.017
$\lvert\lambda\rvert$	1.011	± 0.014	± 0.012
$\lvert A_0\rvert^2$	0.5300	± 0.0016	± 0.0044
$\lvert A_\perp\rvert^2$	0.2409	± 0.0021	± 0.0030
$\lvert A_S\rvert^2$	0.0067	± 0.0033	± 0.0009
$\delta_\parallel$	3.145	± 0.074	± 0.025
$\delta_\perp$	2.931	± 0.089	± 0.050
$\delta_{S\perp}$	0.48	± 0.15	± 0.05

Comparison with other LHC experiments



Outlook

- *B flavor tagging and b-jet tagging are the ultimate examples of how NNs are changing particle physics*
  - *Two problems with clear experimental signature, that any physicist would try to solve from first principles*
  - *Still, NNs have been a game changer in terms of performance and had shown that there is much more information to exploit than the "obvious" experimental signature*
- *Most of the (young) physicists involved think that this is a ten years old revolution, while instead it started 30 years ago, with the work of Marco and others on LEP data*
  - *Most of the successes of modern applied DL, including the synergy between theory and experiments*

The first paragraph

This work deals with the problem of tagging the quark b in jet events in the DELPHI experiment with neural network techniques. The b tagging problem has proved to be a hard one, since no strategy just based on cuts on some variable characterizing the event seems viable. It seems necessary to exploit the full multidimensional data structure and multi-variable correlations are likely to play a major role. Neural networks have often proved capable to successfully cope with this kind of problems. There are pros and cons in the way they do this; the first include the possibility of using general purpose architectures and algorithms to solve problems which in principle would require careful analysis of multidimensional correlations in an "automatic" way. On the other hand this lack of detailed insight into the the solution developed by the network is an obvious drawback of this approach and it can be only partially overcome by an a posteriori analysis of the network configurations and outputs.

- *Human engineered vs artificially engineered features*
- *The importance of exploiting correlations*
- *The "black block problem" and explainable AI*

The second and third paragraphs

The problem of b-tagging has received attention in the last couple of years and some attempts to approach it with neural networks have already produced interesting results [1,5], even if their applicability in an actual experimental situation is not completely established. All the works have much in common from the point of view of the chosen neural network architectures and the learning algorithms used for training (for some interesting alternatives see ref. [2-4]); on the other hand, they significantly differ in the set of variables that feed into the network to give it information about the events to be classified.

In our present paper we focus on selecting input variables appropriate for this problem, regarding particularly how to handle single particle variables. We use a feed forward network trained with realistic input variables obtained from the DELPHI Monte Carlo set up in such a way that a satisfactory agreement is obtained on the MC distributions with the 1990 data set. Results on the performance of the network in terms of tagging efficiencies and purity as well as some analysis of the underlying strategy developed by the network are presented.

- *Delphes-base papers vs real-life applications*
- *Importance of working with real data from the collaborations*
- *Application-oriented development*

Not much different than what we do today

- A "shallow" neural network with 15 hidden nodes
  - Trained with backpropagation
  - Some hyperparameter scan
  - No GPUs back then
- The kind of work one would have done in 2016 at the LHC

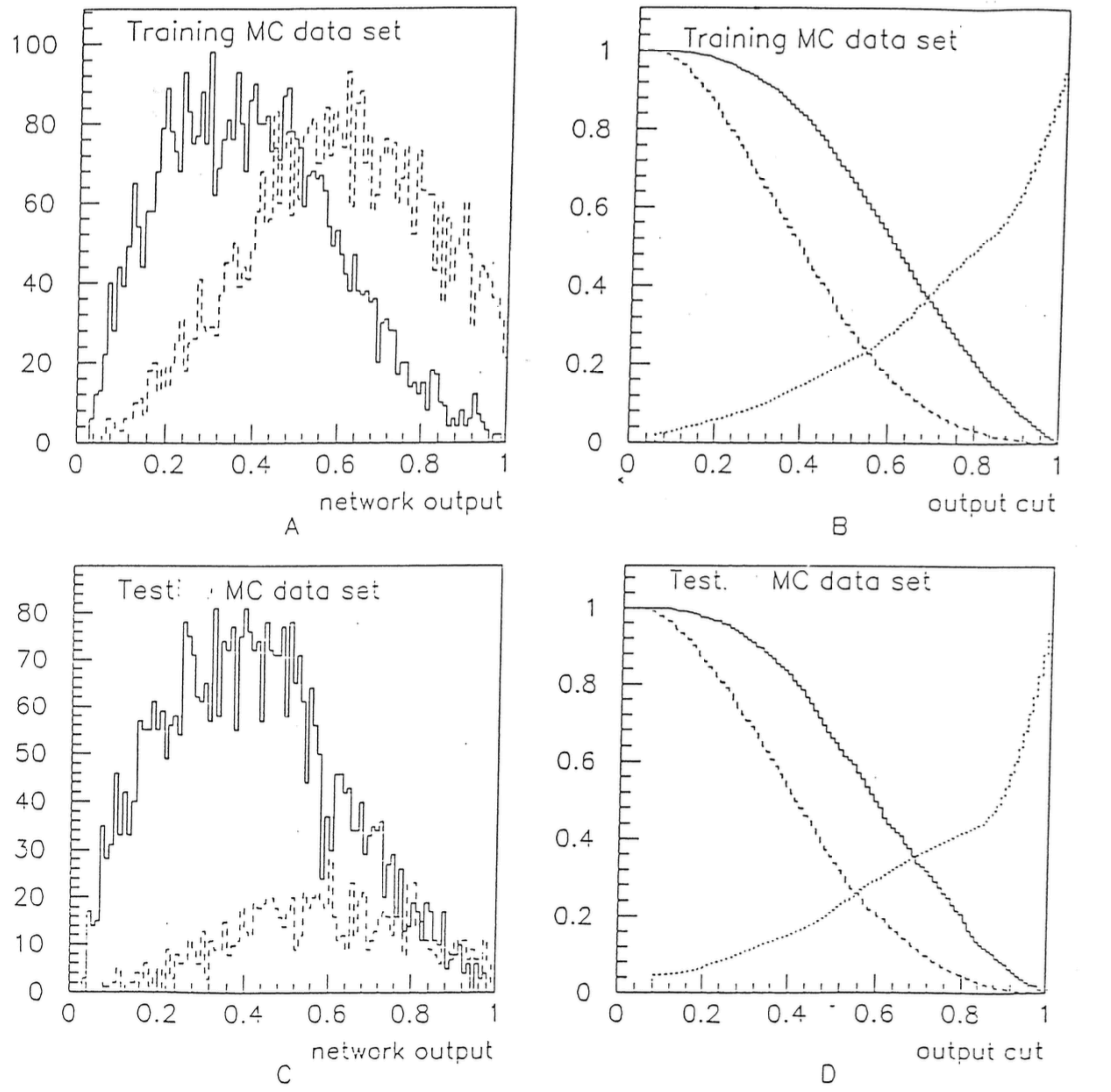


Figure 2 Network output for beauty (dashed line) and background (solid line) events in the training (A) and test (C) data set on the left column; signal efficiency (solid line), background efficiency (dashed line) and purity of b sample (dotted line) in the training (B) and test (D) data set on the right column.

But something got lost with time