Efficiency parametrization with NN

C. Badiali, F. A. Di Bello, G. Frattari, E. Gross, V. Ippolito, E. Loiacono, M. Kado, J. Shlomi

IML meeting
5/5/20

arxiv pre-print

SAPIENZA
UNIVERSITÀ DI ROMA

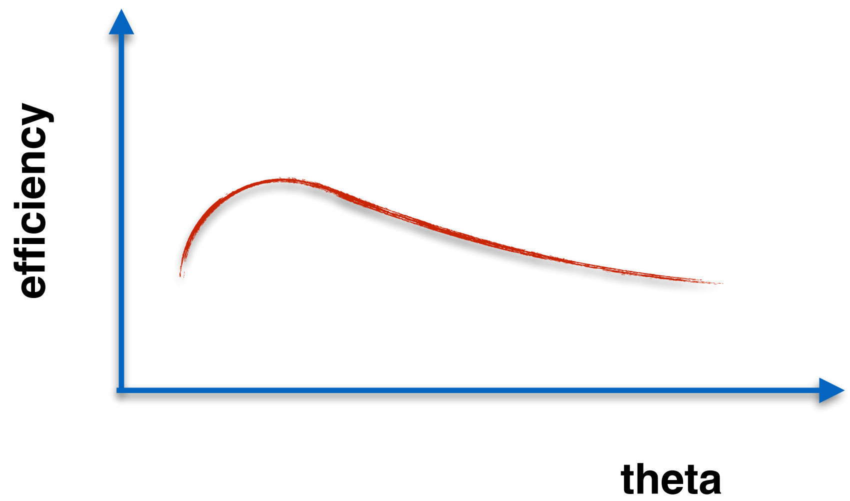
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

INFN
Istituto Nazionale di Fisica Nucleare

Parametrizing the efficiency

- Efficiency is defined as a ratio of two classes.

$$\epsilon_{\text{jet}}\left(\boldsymbol{\theta}\right) = \frac{N(f(\mathbf{x}) > \mathrm{T}_f|\boldsymbol{\theta})}{N(\boldsymbol{\theta})}$$



Smooth parametrization in multi-dimension

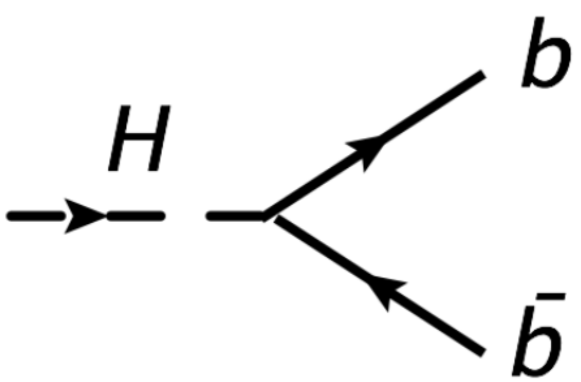
f(x): Tagger used for discrimination

x: Variables used for discrimination

theta: Variables that captures the efficiency dependencies

Leading motivation - MC stat

- Generating enough MC stat ,especially for the background of rare signals is a real issue at the LHC.
- Clear examples are Hbb/cc, (ttH, VH, etc..) analyses.

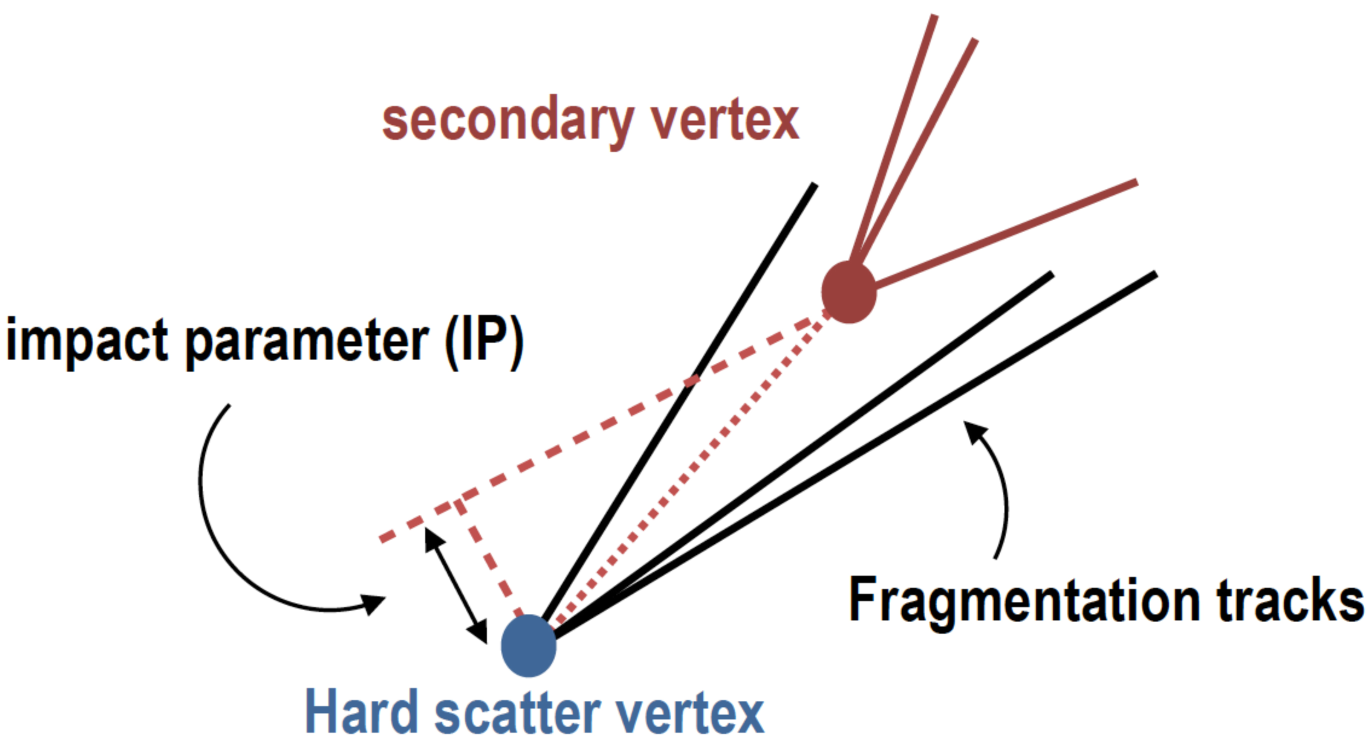


[Observation of Hbb coupling]

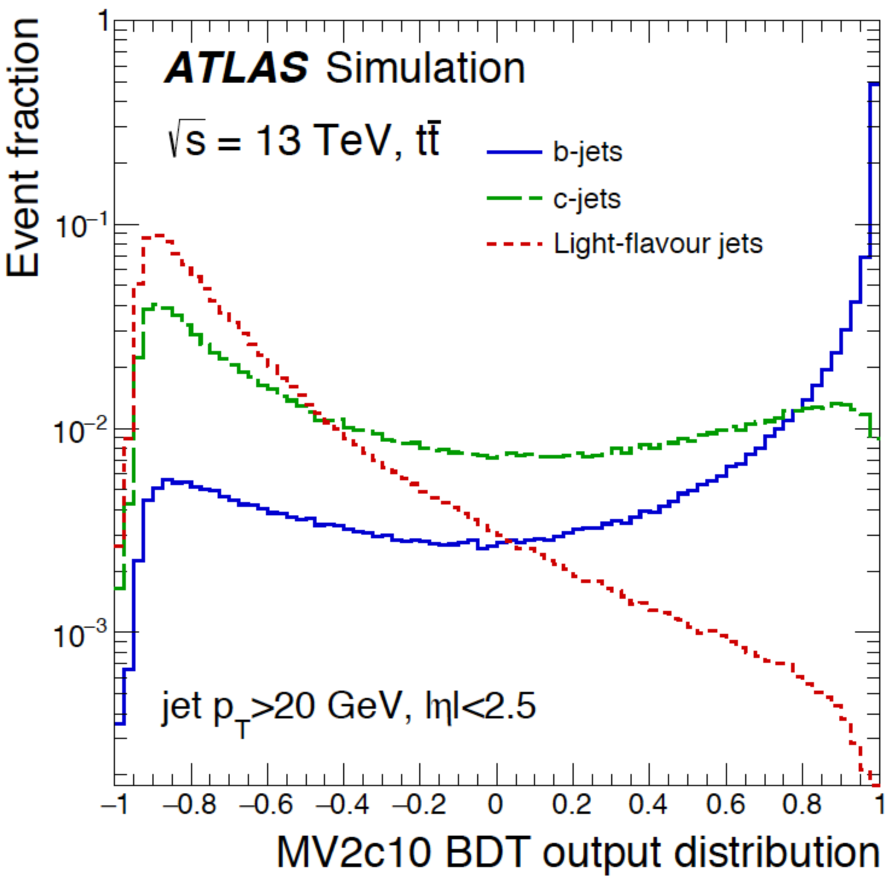
Source of uncertainty		σ_μ
Total		0.259
Statistical		0.161
Systematic		0.203
Experimental uncertainties		
Jets		0.035
E_T^{miss}		0.014
Leptons		0.009
b-tagging	b-jets	0.061
	c-jets	0.042
	light-flavour jets	0.009
	extrapolation	0.008
Pile-up		0.007
Luminosity		0.023
Theoretical and modelling uncertainties		
Signal		0.094
Floating normalisations		0.035
Z + jets		0.055
W + jets		0.060
$t\bar{t}$		0.050
Single top quark		0.028
Diboson		0.054
Multi-jet		0.005
MC statistical		0.070

Dominant
… since RUN 1 analyses!

A word about b-tagging



Discriminant f(x)



b-tagging taken as an example but the work can be applied to any efficiency estimation

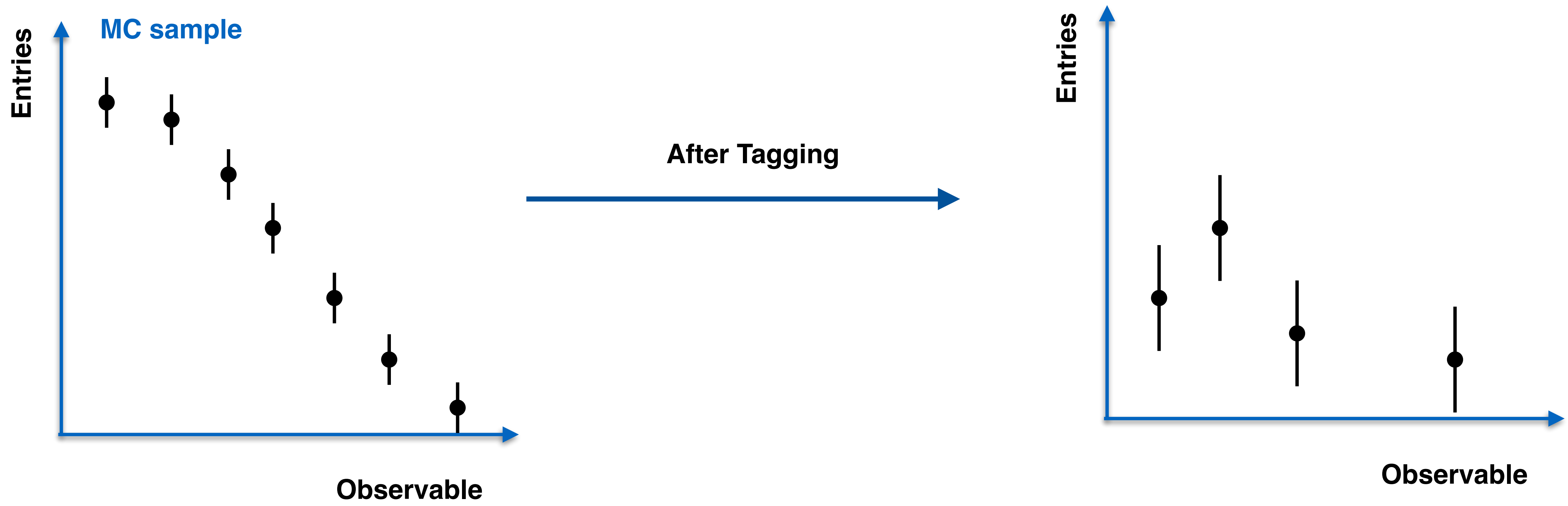
The issue of MC stat

identification efficiency is low for c- and light-jets:

For a b-jet efficiency of 70 %:

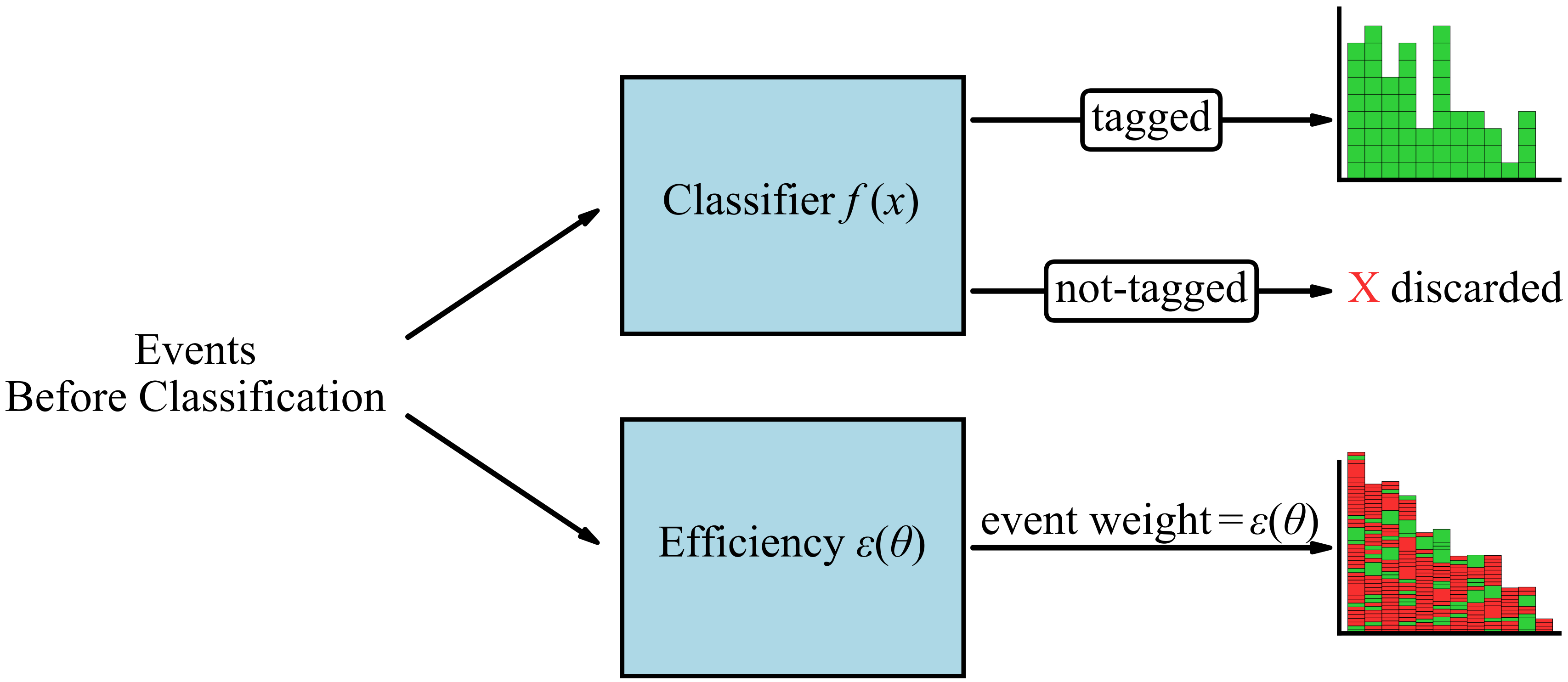
Light-Jets are classified as b-jets with a rate of around 1 %

c-jets are classified as b-jets with a rate of around 10 %



Truth Tagging - generalities

- TT is a weighting technique where all events are retained and weighted with the expected efficiency/mistag values



- x = input to the discriminant (IP properties, Secondary vertex etc)

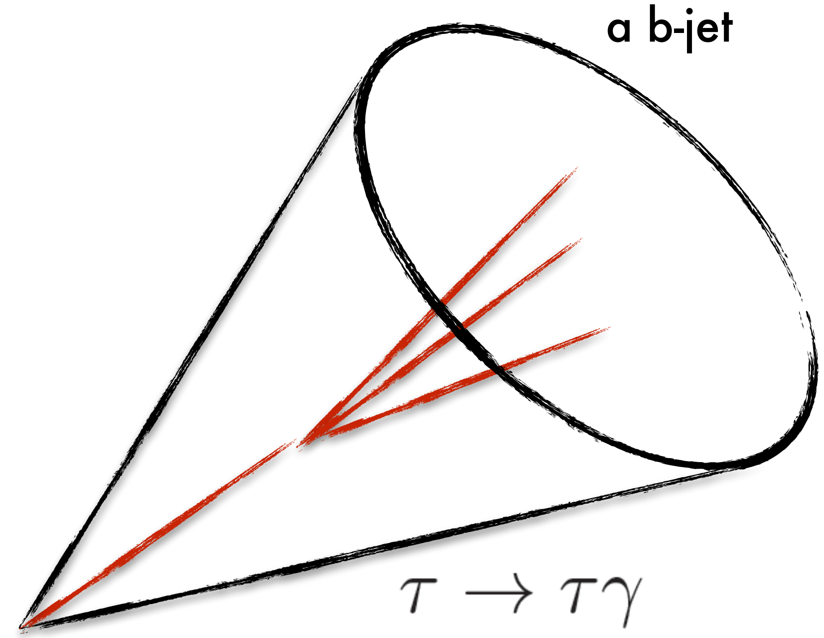
- theta = Parameters used to derive the parametrization (jet-pt,jet-eta, others...)

The choice of theta

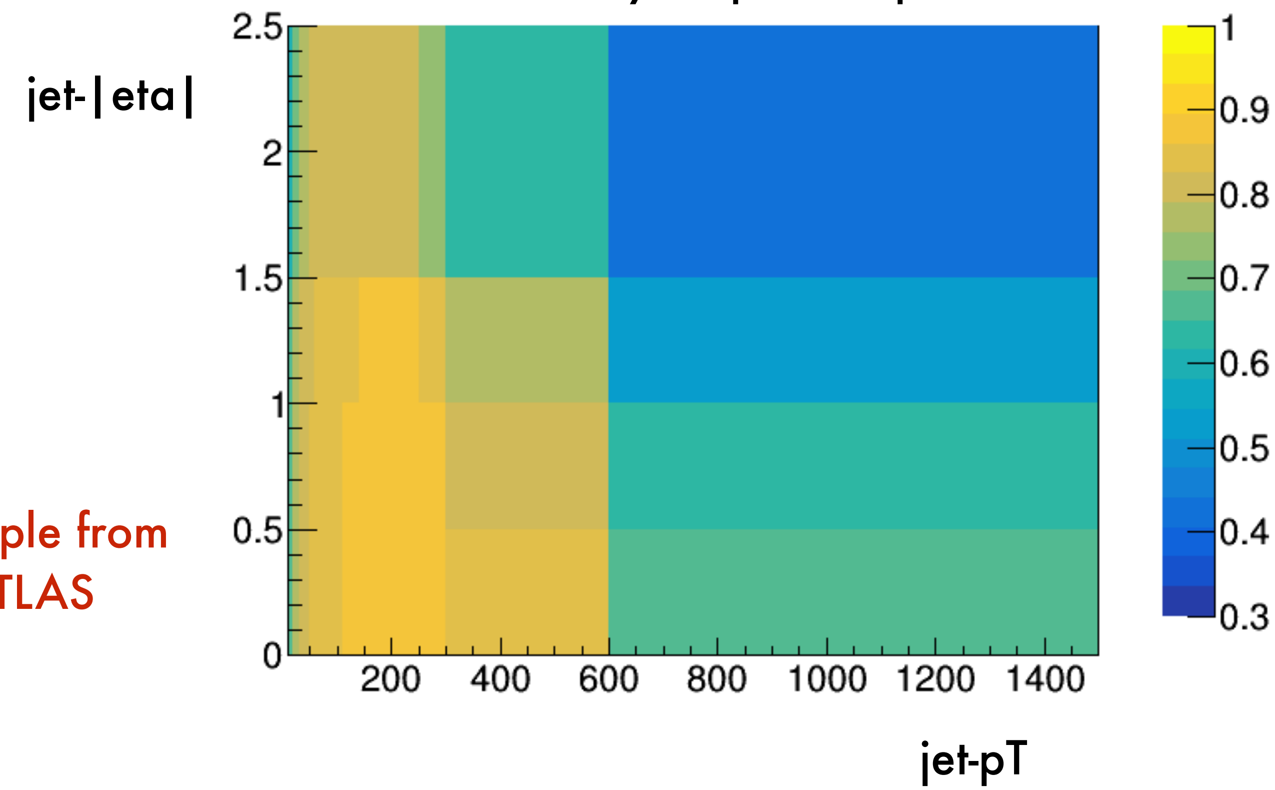
- The choice of theta is crucial and not trivial
- Theta is not typically fully known
- A set of variables correlated with the tagging decision driven by prior physics considerations
- Theta and x have typically no overlap
- In ATLAS the b-tag efficiency parametrization is carried out in 2D (jet-pT, jet-eta maps).

bias observed and large syst associated with these maps

[VHbb RUN1 paper]



Efficiency map for b-jets

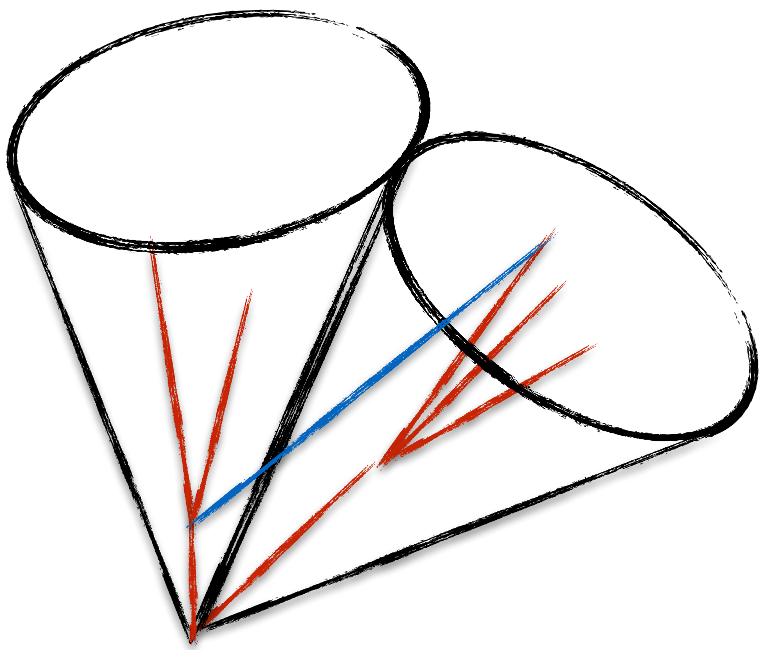


Example from ATLAS

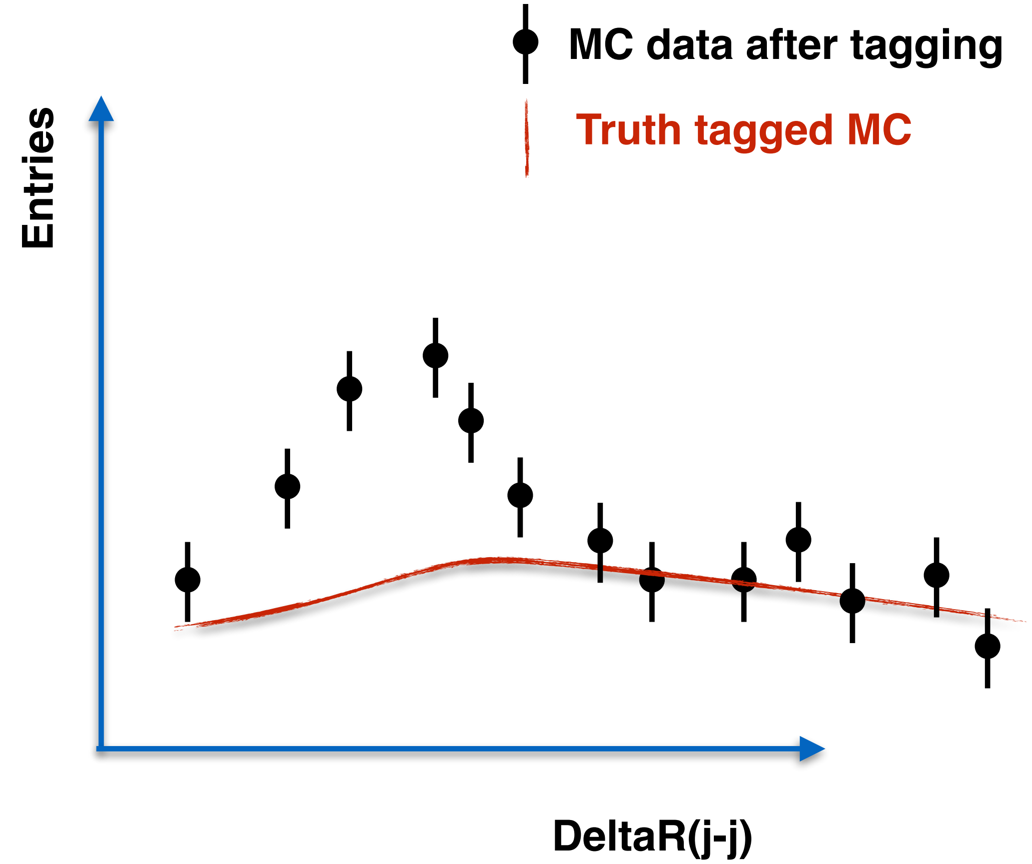
Truth Tagging in ATLAS

- A dependence on the jet-jet angular distance was observed in boosted regimes

- Also depends on the flavor of the close-by jet. Not trivial to cope with.



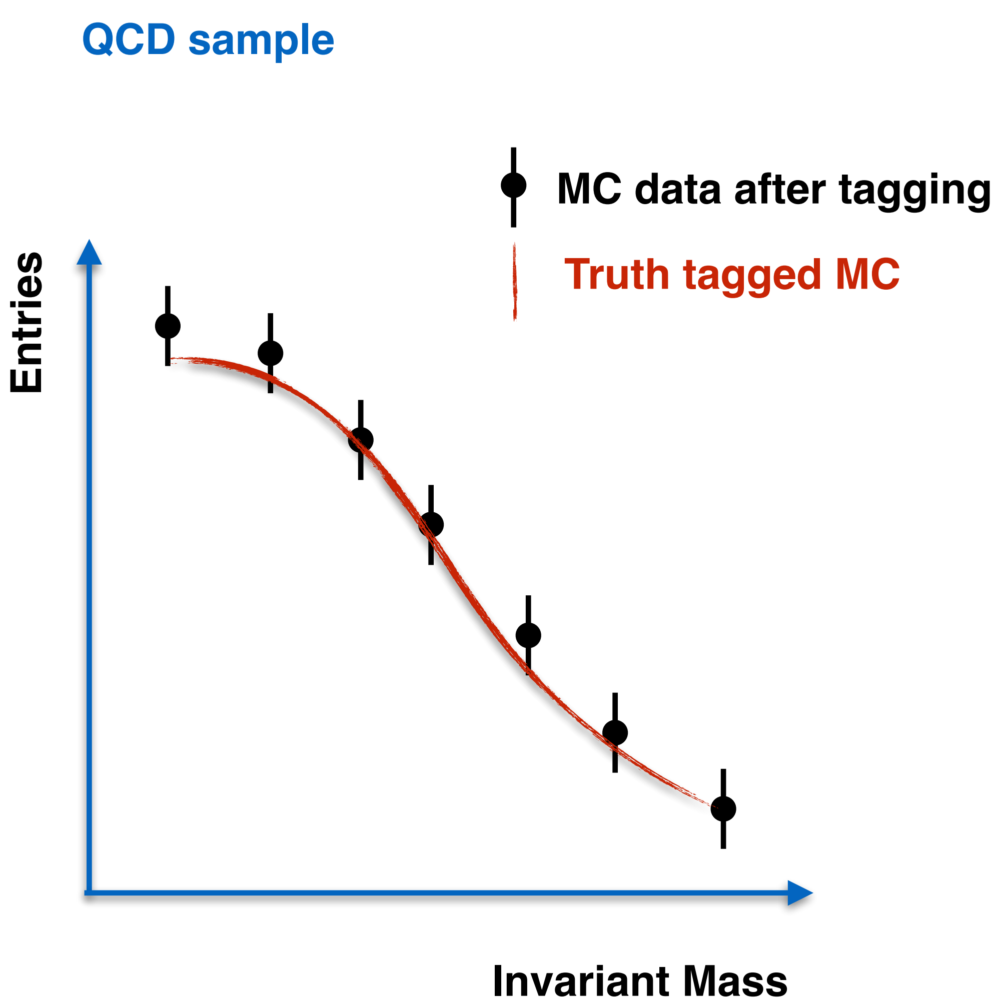
Wrongly associated track

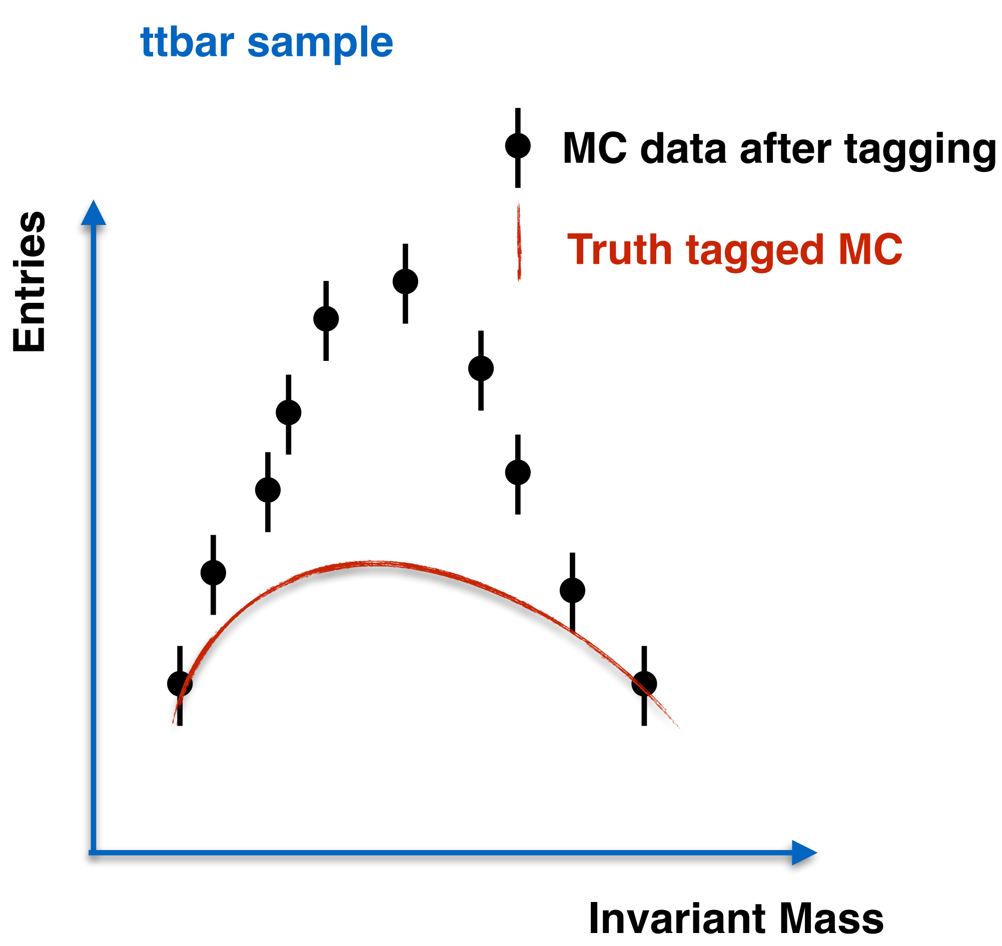


at small angles, the map-based approach

Generalization

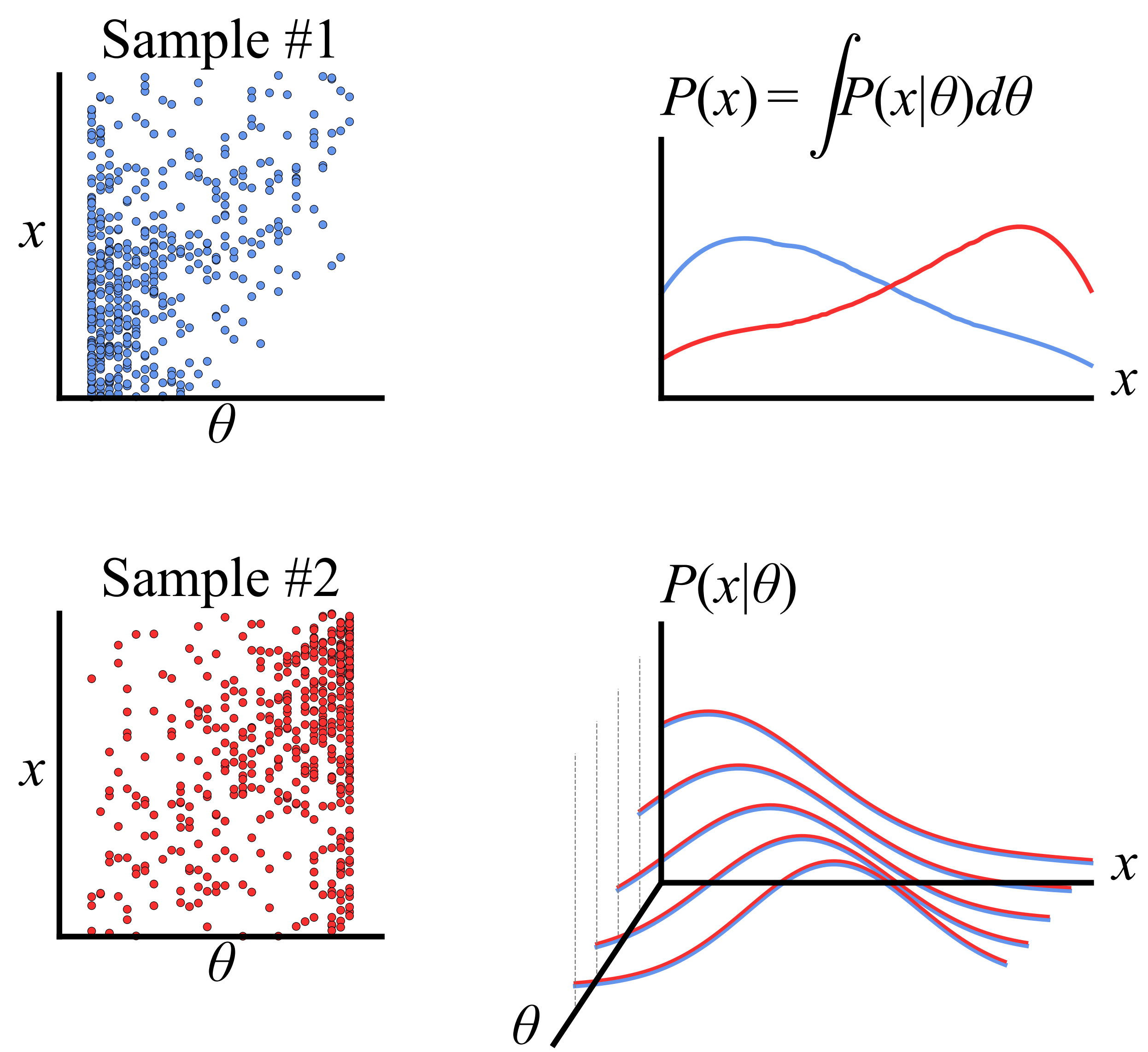
- A desirable feature of the parametrization is generalization, i.e. Is it applicable to different sample?
- Example: Consider a QCD and a ttbar sample.
    1. Parametrization is derived on the QCD sample and works well there.
    2. Would that parametrization work on a ttbar sample





Generalization

- Conditions for a "universal" parametrization (typically depends on the phase-space of interest for the analyses)



- The key is an appropriate choice of theta

- If theta becomes large, not easy to come with large stat fluctuation

The main limitations of map-methods

1. Theta typically not fully known. Do not generalize

2. Increase in dimension of the maps require appropriate smoothing in multi-dimension

3. One map for each flavor. Need to control several maps. This is impractical

Indeed never really worked...

We propose a NN approach to solve these issues

A NN approach

- Would be better to directly access multidimensional maps

- Let's start with a simple feed-forward NN with input theta* = (jet-pt, eta, deltaR, flavor of closest jet)

the * indicates that this set is a subset of theta which only accounts for the closest jet

Loss function:

gentle explanation

$$\frac{1}{N_{jets}} \sum_{j=1}^{N_{jets}} -(\mathrm{istag}_j)\log(\epsilon_{NN}(\theta^*)_j) - (1-\mathrm{istag}_j)\log(1-\epsilon_{NN}(\theta^*)_j) \longrightarrow \epsilon_{NN}(\theta^*) \approx \frac{p_{\mathrm{tag}}(\theta^*)}{p_{\mathrm{tag}}(\theta^*) + p_{\text{non-tag}}(\theta^*)}$$

- This is a density ratio estimation, the num and den remain independently unknown

Additional considerations:

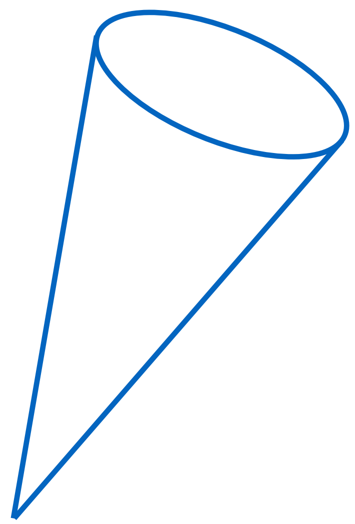
1. A priori the parameters theta are not fully known. Could be inferred directly during the training
2. We need to cope with jet-jet dependency in every event where the number of jets is not fixed

—> GNN

[GNN review]

Physical space

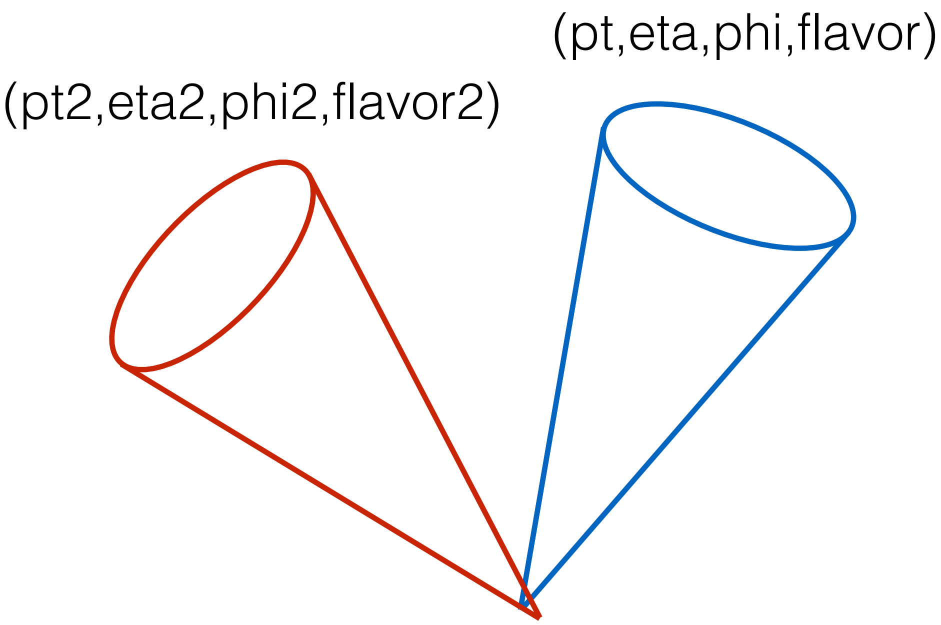
(pt,eta,phi,flavor)



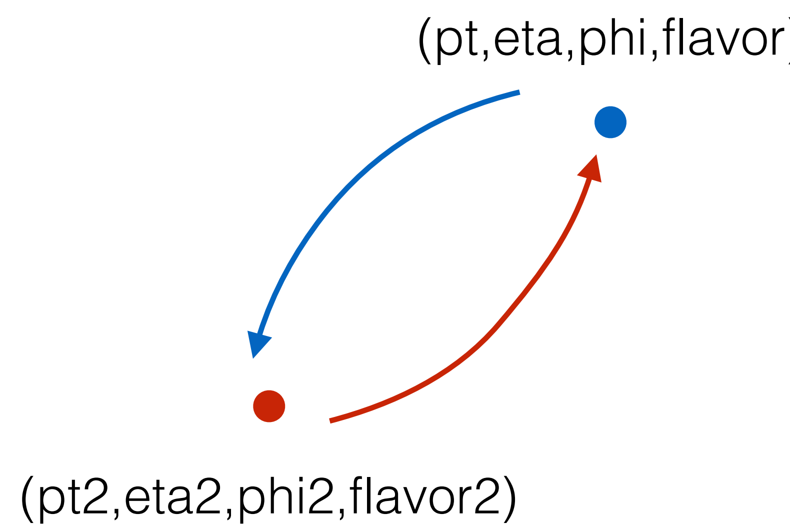
Graph

(pt,eta,phi,flavor)

Physical space



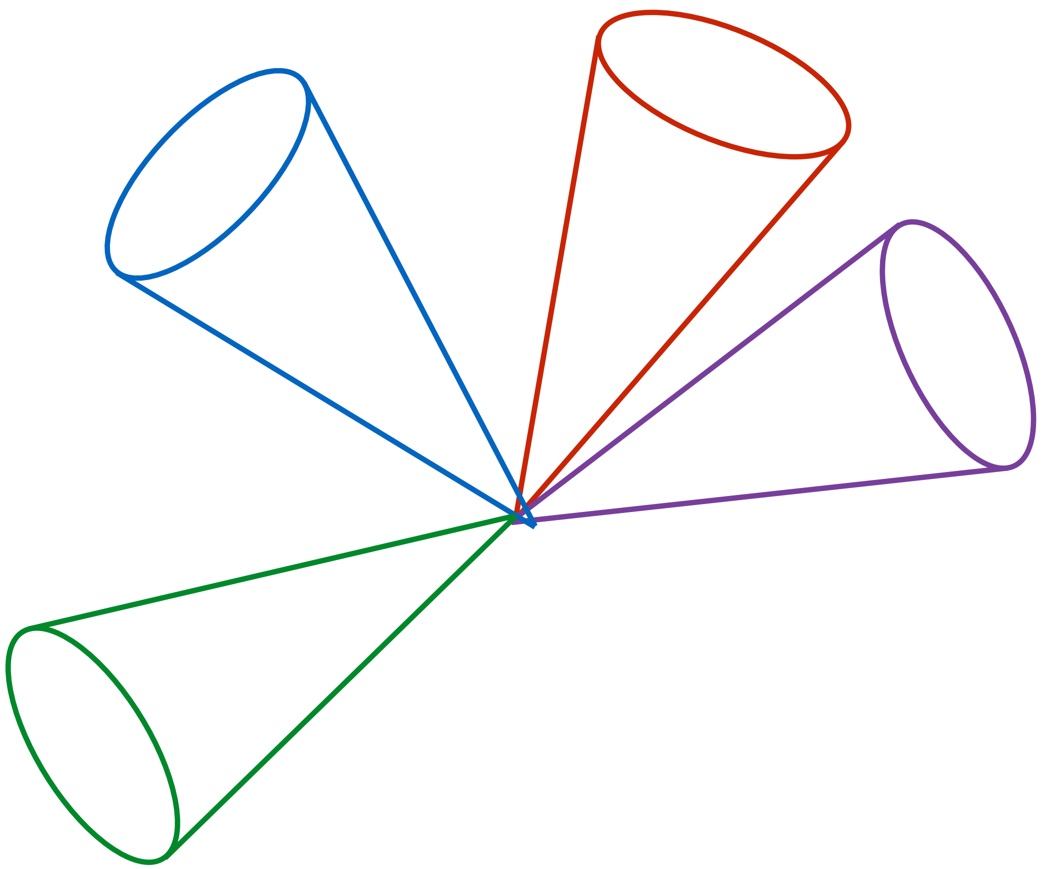
Graph



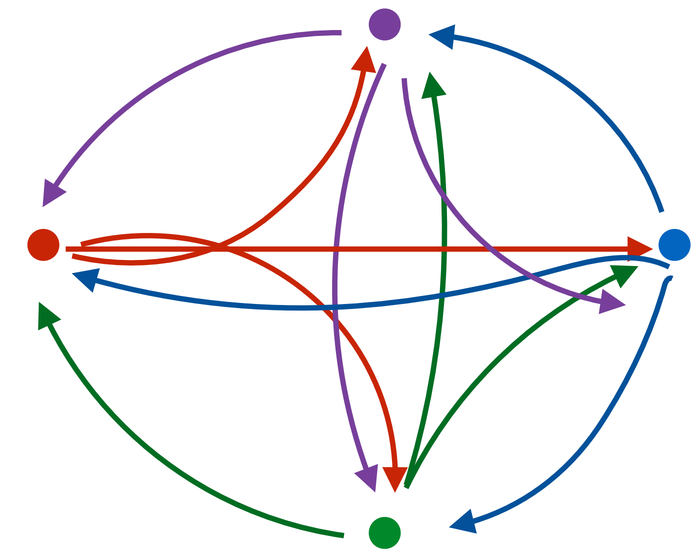
$$\Delta R(i, j) = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}.$$

- Note that (deltaR,fj) is not given explicitly but rather inferred during the training
- For this the input to the NN is not theta directly but a different set Theta is used

Physical space



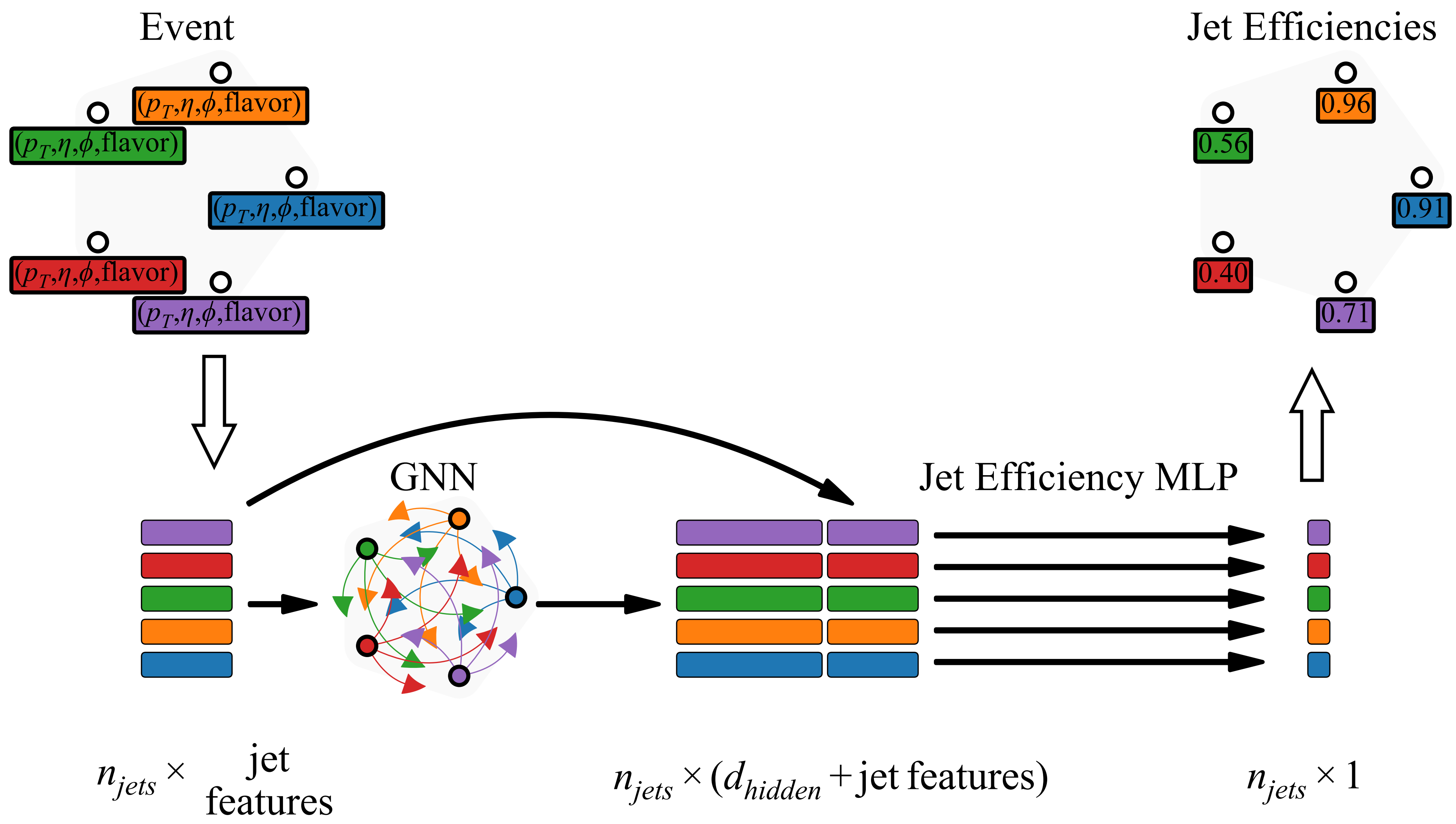
Graph



- learning efficiencies with inter-object correlations and object-environment dependencies
- Flexible in terms of number of jets
- Permutation invariant

The model

NN(Set of jets) —> eff_j, for each jets



The model

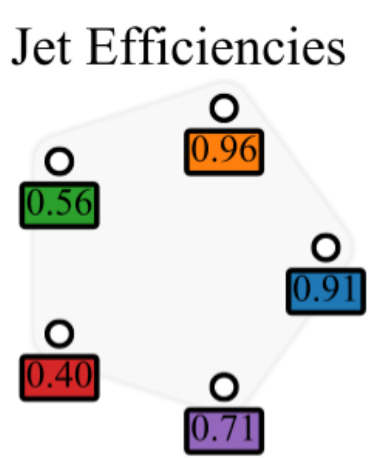
- theta = variables influencing the true efficiency
- Theta = Variables used as input to the NN - theta is a function of g(theta) inferred by the NN

Loss function:

$$\mathrm{BCE}_e = \frac{1}{N_{jets}} \sum_{e}^{N_{events}} \sum_{i}^{n_{jets_e}} \left[-(\mathtt{istag}_i)\log(\epsilon_{NN}(\mathbf{\Theta}_e)_i) - \mu(1-\mathtt{istag}_i)\log(1-\epsilon_{NN}(\mathbf{\Theta}_e)_i)\right],$$

downsampling factor for unbalanced dataset.

for GNN solved by enlarging batch size and fixing mu = 1



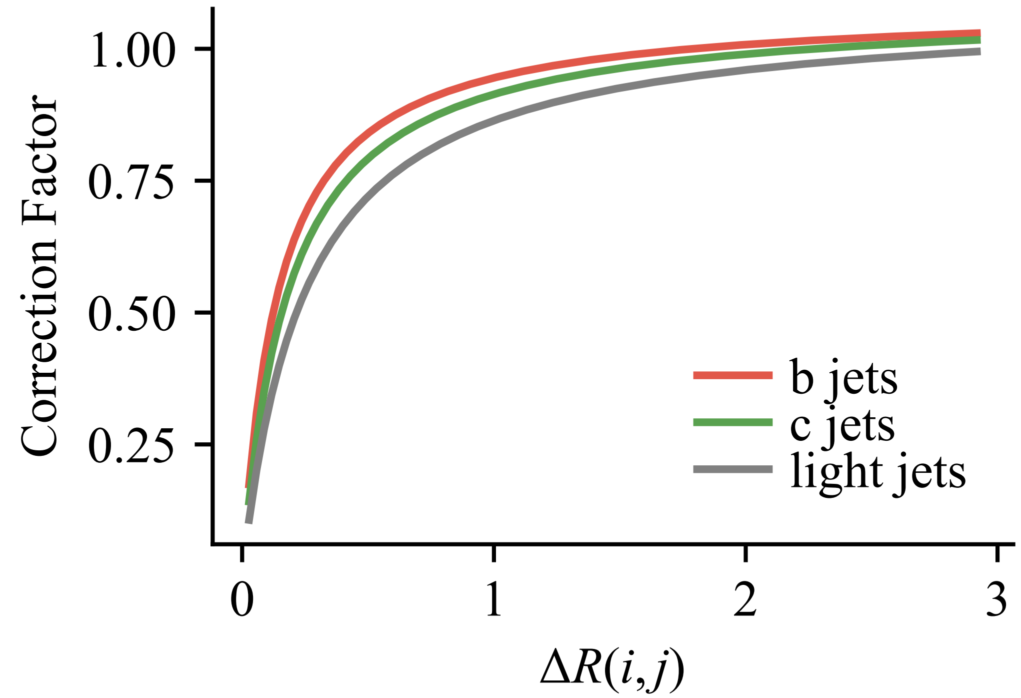
output in the event e for the i-th jet:

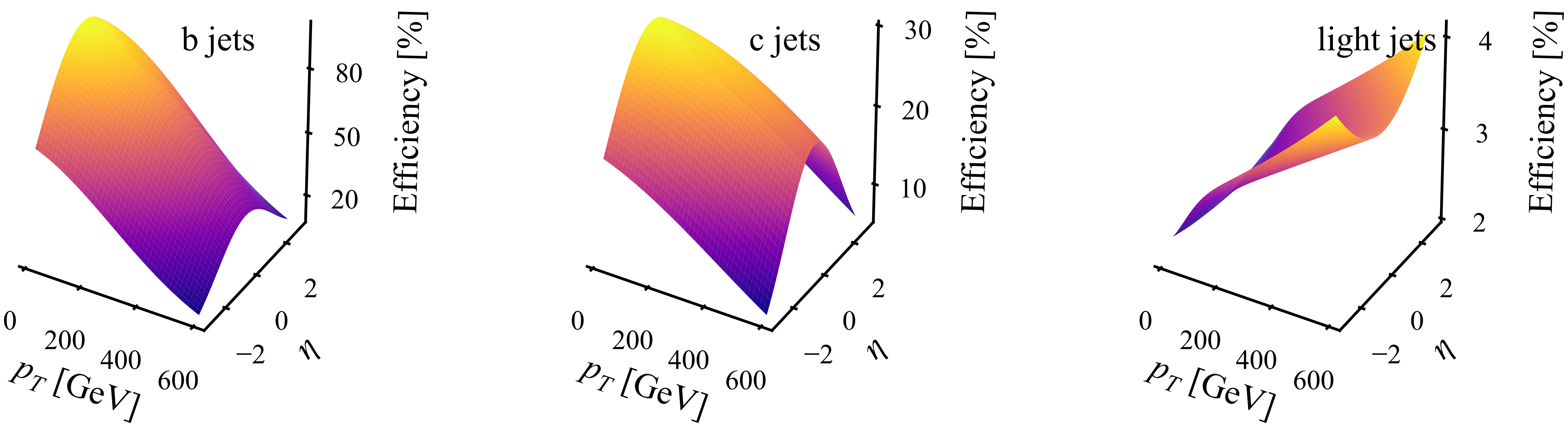
$$\epsilon_{NN}(\mathbf{\Theta}_e)_i \approx \frac{p_{\text{tag}}(g_i(\mathbf{\Theta}_e))}{p_{\text{tag}}(g_i(\mathbf{\Theta}_e)) + p_{\text{non-tag}}(g_i(\mathbf{\Theta}_e))} \approx \epsilon(g_i(\mathbf{\Theta}_e)) \approx \epsilon(\theta_i) = \epsilon_i$$

The Toy Model

- pt, eta and phi distribution sampled using functions
- True efficiency given by the following formula:

$$\epsilon_{\mathrm{jet}_i} = \epsilon_{f_i}(p_{\mathrm{T}}, \eta) \cdot \prod_j \hat{\epsilon}_{ij}\left(\Delta R(i,j), f_j\right),$$

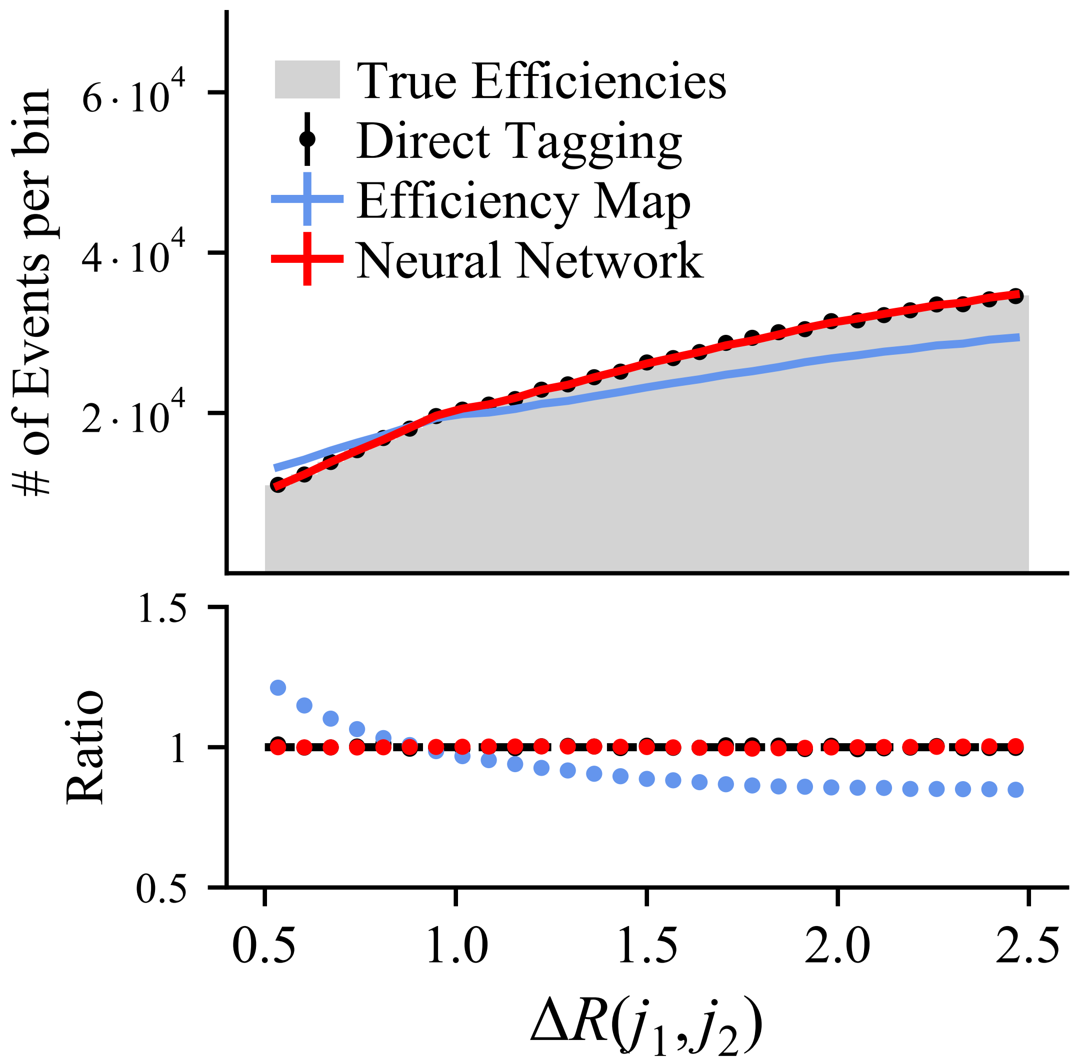


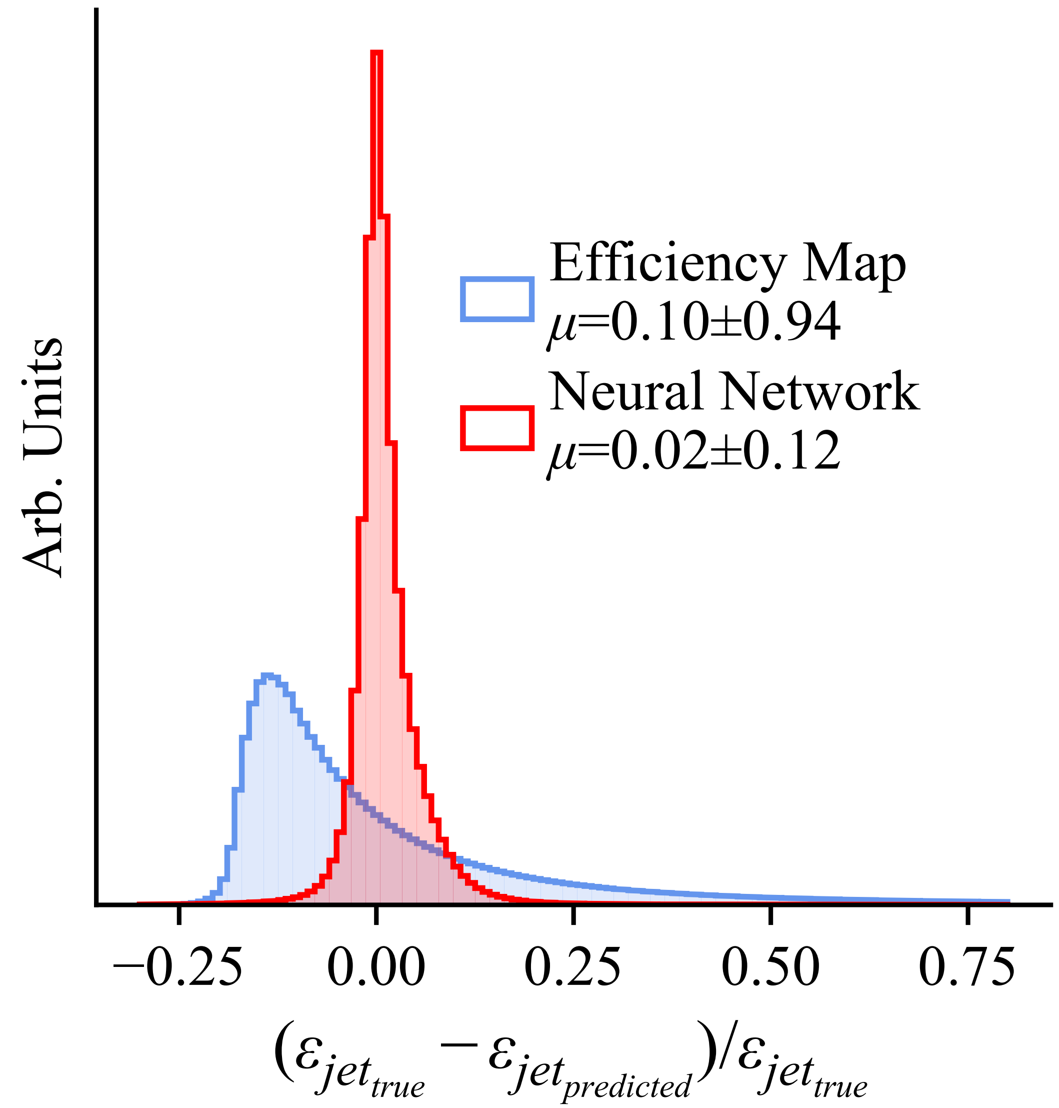


Results 1

- 1tag region, leading jet is b-tagged

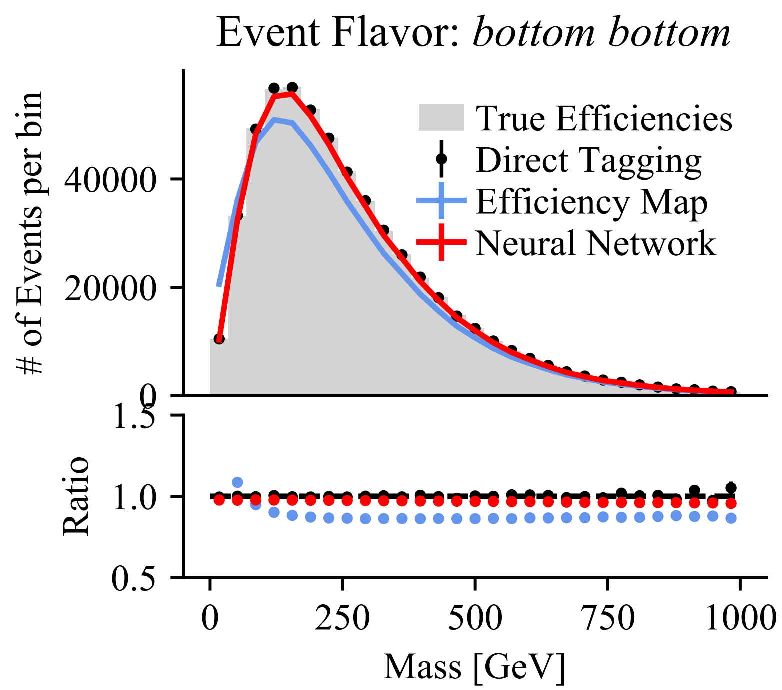
- In the efficiency map approach the jet-jet dependencies are marginalized out $\epsilon_{map}(p_T, \eta) = \int \epsilon(p_T, \eta, \Delta R) d(\Delta R)$
Leaving a sample poor deltaR modeling as well as a sample dependency

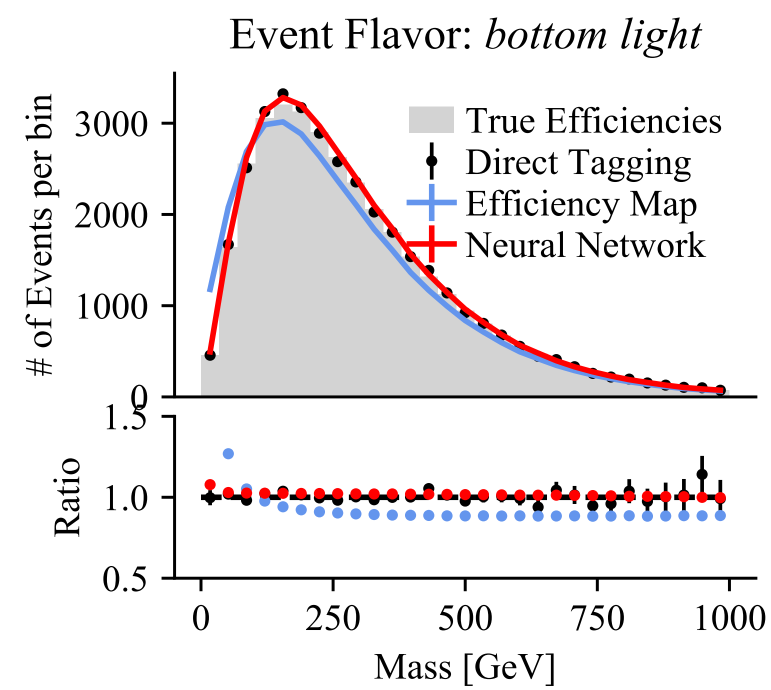


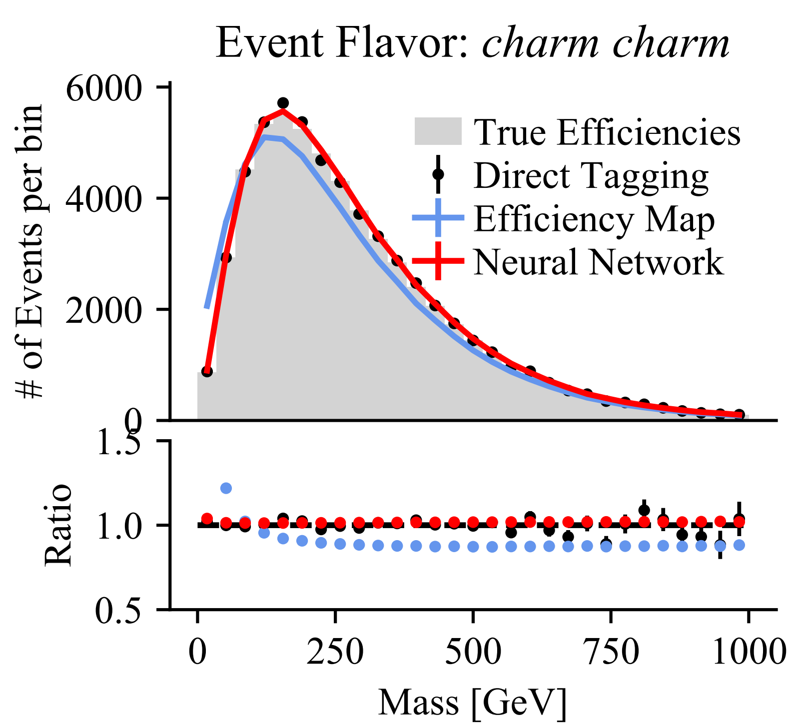


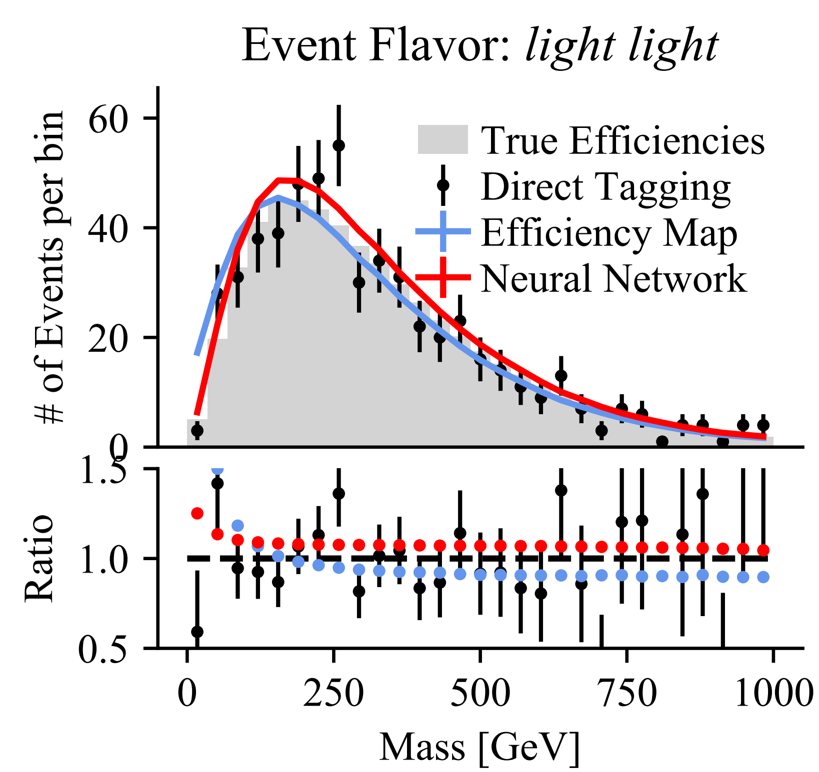
Result 2

- 2tag region, leading and sub-leading jets are b-tagged $\epsilon_1 \epsilon_2$



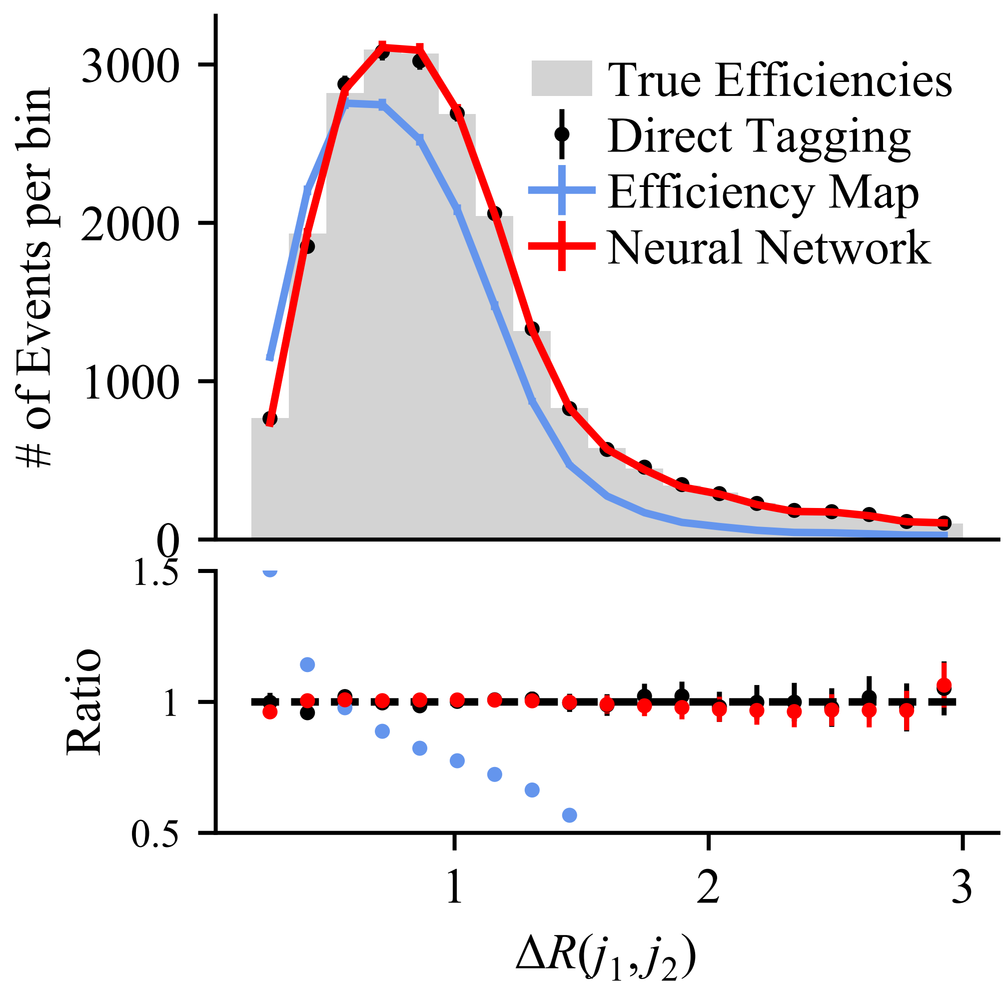


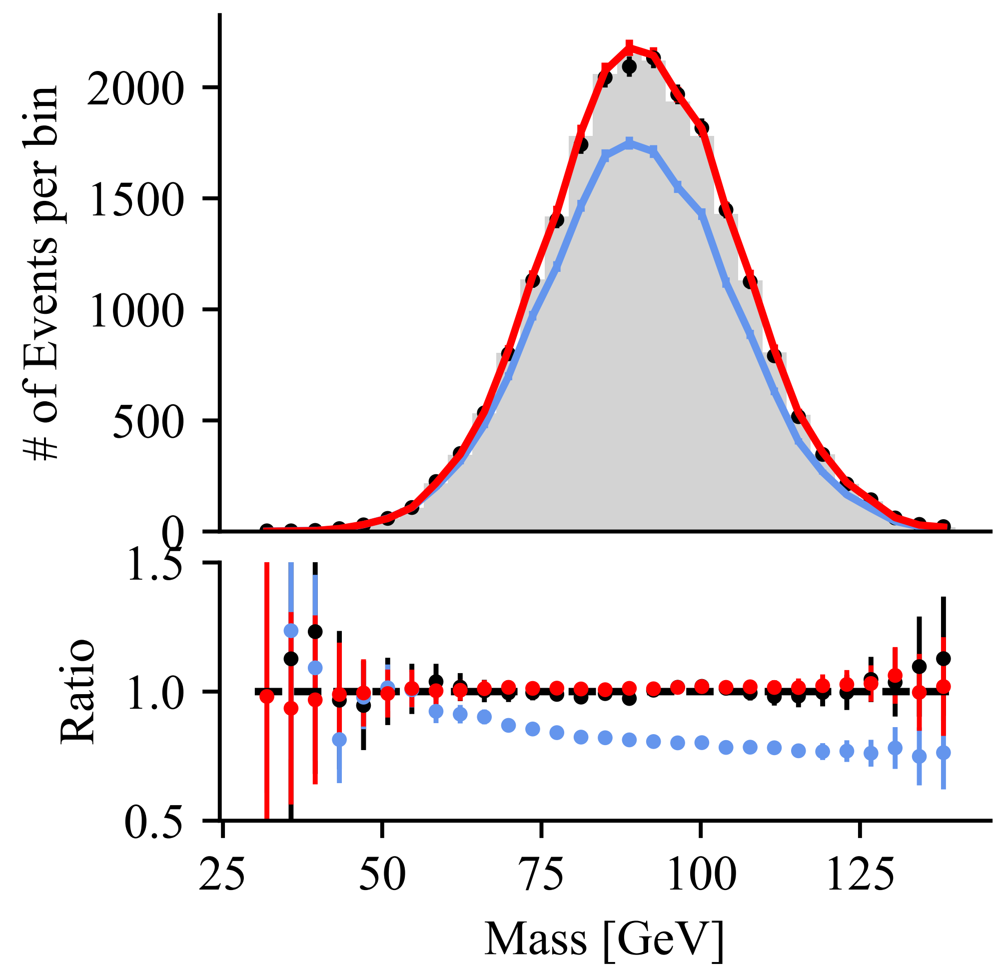




Results 3 - Generalization

- Is the model able to generalize to a different sample?
- Changed generation, scalar particle (m0 = 90 GeV) decaying into two quarks





Conclusions

- We propose a new method to parametrize efficiency in a multidimensional space

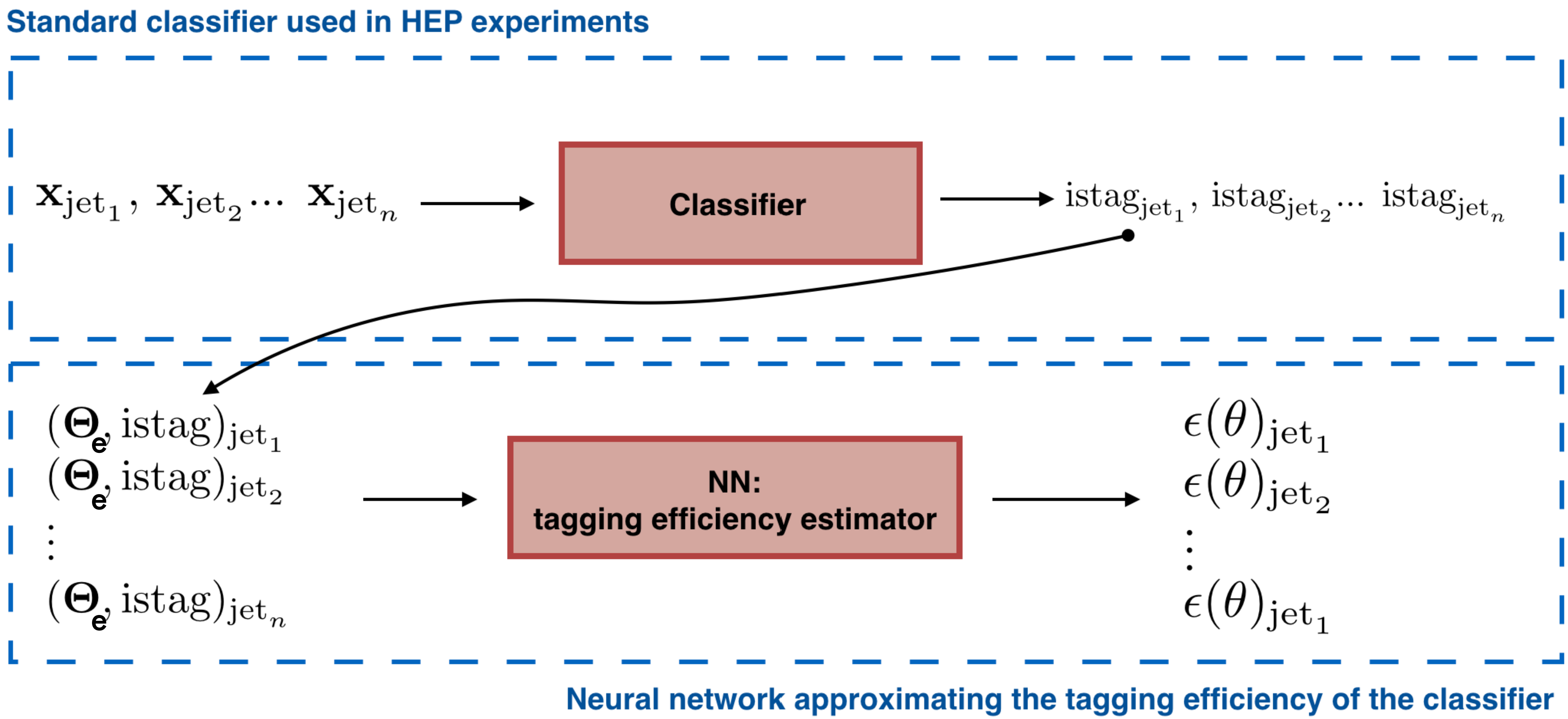
- The main advantages are:

    1. In one go have a parametrization for b-, c- and light jets instead of dealing with several binned maps
    2. Flexible in terms of event efficiency obtained combining different single-jet efficiencies
    3. No need to know a priori the relevant parameter theta
    4. Generalize well to sample on which it was not trained

- Looking to implement this in ATLAS, this could be used for cases where a variable number of objects as well as inter-objects dependencies needs to be taken into account.

Recap of the nomenclature and methodology

- x = input variable to the discriminant - not used for the parametrization
- theta = variables influencing the true efficiency
- Theta = Variables used as input to the NN - theta is a function of g(theta) inferred by the NN



Discussion

The size of $\boldsymbol{\theta}$ We used a relatively small number of variables that control the efficiency and required the network to only infer the variable $\Delta R(i, j)$. In real-life applications, $\boldsymbol{\theta}$ may include more variables and the related inference may be more complex in higher dimensions. To cope with this, the inputs variables $\boldsymbol{\Theta}$ needs to be enlarged using additional variables. Neural networks are a particularly suitable tool to perform this task due to their flexibility to cope with higher dimensions. Any variables potentially correlated with the tagging decision could be used to ensure that all correlations are captured.

The functional form of $\epsilon(\boldsymbol{\theta})$ We assumed a relatively simple efficiency in Eq. 3.1. In principle, the neural network can learn any function, no matter how complex the functional form is, as shown in Ref. [12]. This method in scenarios where the form of $\epsilon(\boldsymbol{\theta})$ may present more complex dependencies between the efficiency and the relevant variables $\boldsymbol{\theta}$.

Systematic uncertainties In the applications of the simple efficiency maps, the insufficient capture of the existing underlying correlations requires the introduction of systematic uncertainty. This method is aimed at avoiding this systematic error, it will, however, require thorough checks to ensure that its estimates are accurate.

Generalization of the method In the proposed approach we have focused our studies to approximate efficiency, i.e. density ratios between two complementary classes. The method can also be generalized to approximate ratios between two separate classes[3]. A multidimensional ratio between two classes could be used in a variety of different applications, such as to derive multi-dimensional scale factors from data to correct the tagging efficiency in Monte Carlo simulation.

Inside the GNN

- Three GN block are concatenated
- Relu and tanh activations functions
- Stochastic gradient descent
- Batch size of 5000 events - relevant especially for light jets
- dout is 256
- Used 20 NN with random weight initialization