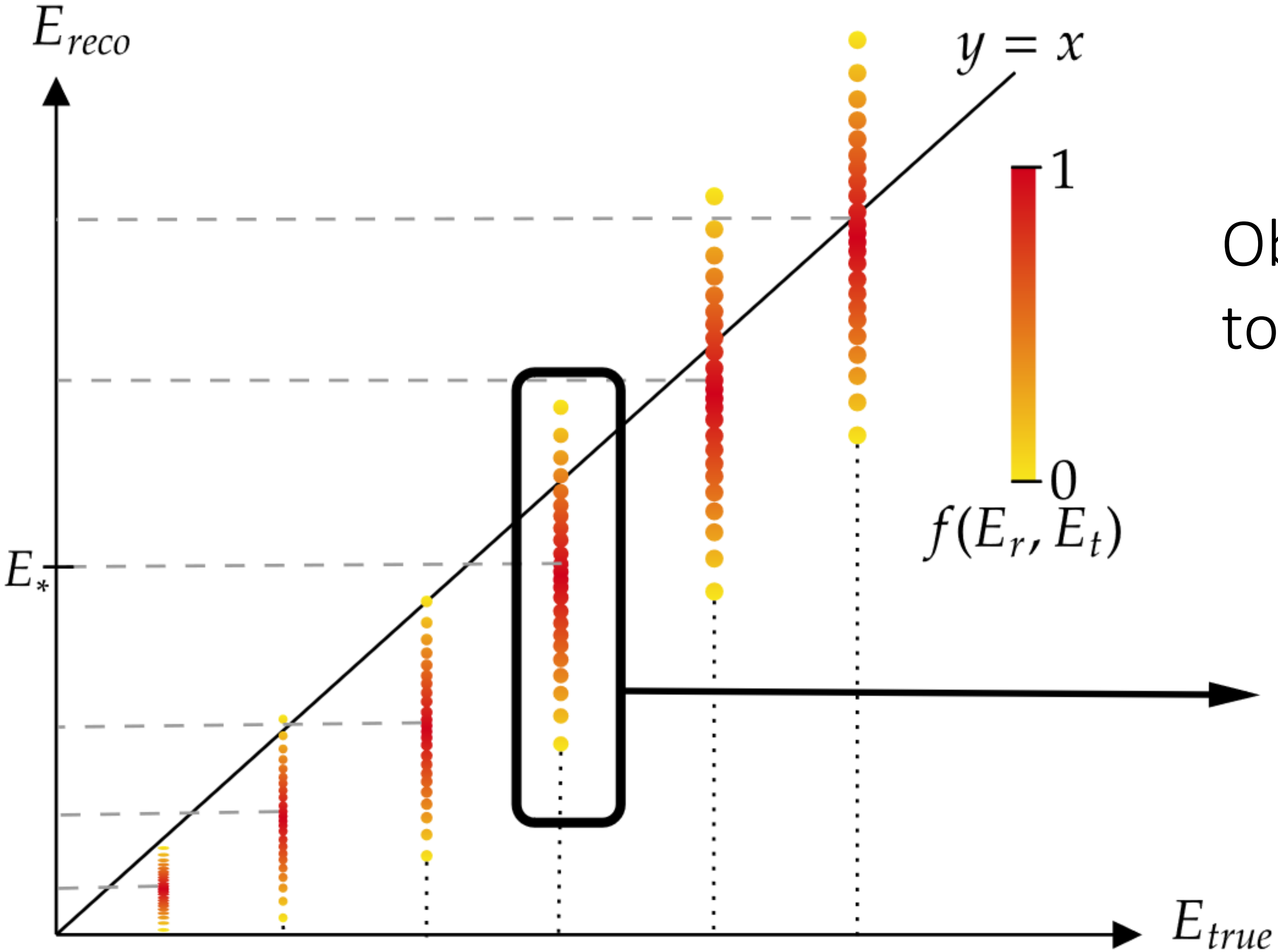
Solving a differential equation to calibrate the ATLAS detector using a neuronal newtork

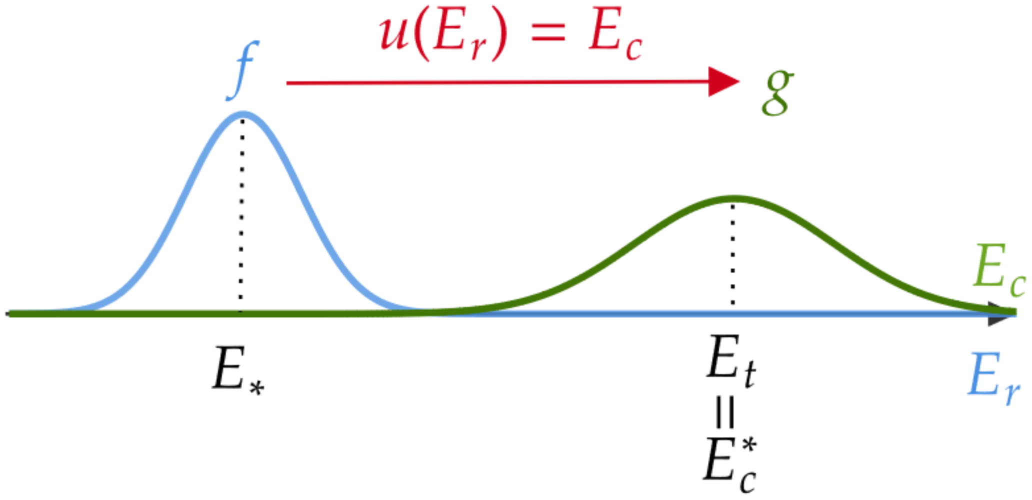
Matthieu Pélissier, Pierre-Antoine Delsart

I - Context

Energy calibration of hadron jets required

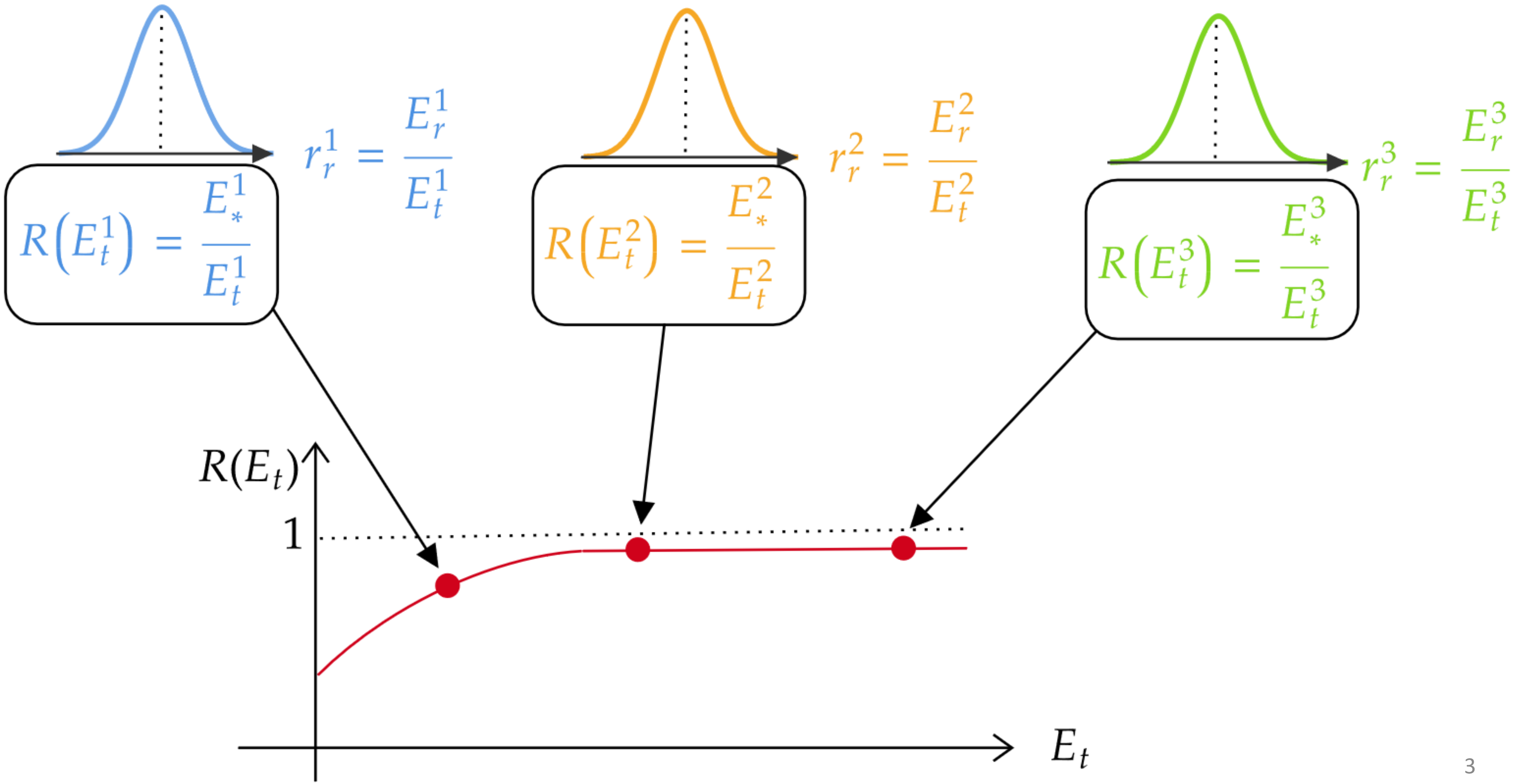


Objective : find calibration function u to go from f to g.



I - Context

Energy response: links the maximum of the measured and calibrated probability density.



$r_r^1 = \frac{E_r^1}{E_t^1}$

$R\left(E_t^1\right) = \frac{E_*^1}{E_t^1}$

$r_r^2 = \frac{E_r^2}{E_t^2}$

$R\left(E_t^2\right) = \frac{E_*^2}{E_t^2}$

$r_r^3 = \frac{E_r^3}{E_t^3}$

$R\left(E_t^3\right) = \frac{E_*^3}{E_t^3}$

II – Differential equation on the calibration function

System to solve

$$\begin{cases} g(E_c, E_t) = f(v(E_c), E_t)v'(E_c) \\ \frac{dg(E_c)}{dE_c}(E_t, E_c = E_t) = 0 \end{cases}$$

Gaussian measurement approximation

$$f(E_r, E_t) = C_{norm} e^{-\frac{1}{2\sigma(E_t)^2}(v(E_c) - E_t R(E_t))^2}$$

Differential equation on the inverse calibration function

$$v''(E_t)\sigma(E_t)^2 - v'(E_t)^2 \left(v(E_t) - E_t R(E_t)\right) = 0$$

Main quantities

E_r	Measured energy
E_c	Calibrated energy
E_t	Energy of the maximum of $g(E_c, E_t)$
$g(E_c, E_t)$	Probability density of calibrated energy
$f(E_r, E_t)$	Probability density of measured energy
$R(E_t) = \frac{E_r^*}{E_t}$	Energy response
$v(E_c) = E_r$	Inverse calibration fonction

II – Differential equation on the calibration function

Differential equation on the inverse calibration function

$$v''(E_t)\sigma(E_t)^2 - v'(E_t)^2\left(v(E_t) - E_t R(E_t)\right) = 0$$

Normalisation

$$E \rightarrow x = \frac{E}{E_{norm}}$$

Link u and v

$$u(v(x)) = x$$

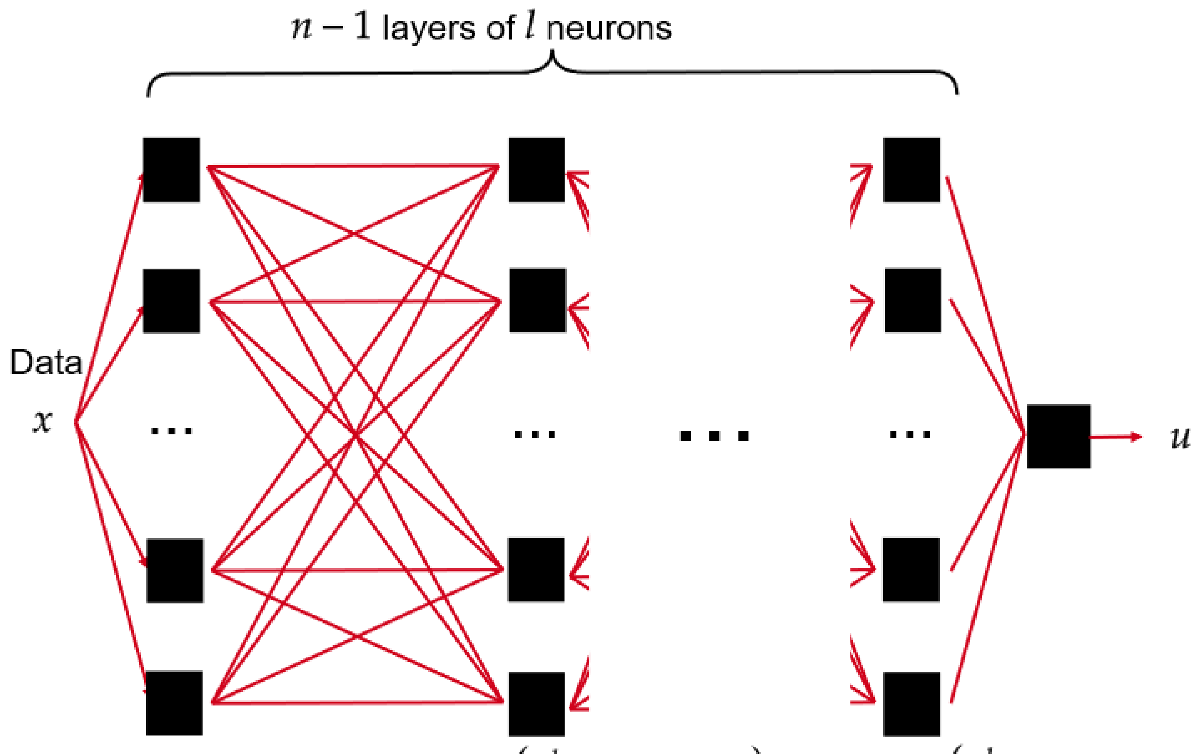
Differential equation on the calibration function

$$\boxed{u''(x)\sigma(x)^2 + u'(x)(x - u(x)R(u(x))) = 0}$$

Main quantities

E_r	Measured energy
E_c	Calibrated energy
x	Normalized calibrated energy
E_t	Energy of the maximum of $g(E_c, E_t)$
$R(E_t) = \frac{E_r^*}{E_t}$	Energy response
$v(E_c) = E_r$	Inverse calibration function
$u(E_r) = E_c$	Calibration function

III – ODE resolution using neural network



$n-1$ layers of l neurons

Data

x

$$u(x) = \sum_{j=1}^{l} p_{n,j} a_{n-1}^{j} + b_n$$

Equation to solve : $\psi(x, u, u', u'') = 0$

$$loss = \overbrace{\underbrace{(\psi(x, u, u', u''))^2}_{\text{eqLoss}}}^{\text{Differential equation}} + \underbrace{\sum (u(x_i) - u_i)^2}_{\text{bcLoss}}$$

For boundary conditions (x_i, u_i) on u or u'

$$a_1^i = act\left(p_1^i x + b_1^i\right)$$

Neuron i from layer 1

$$a_2^i = act\left(\sum_{j=1}^{l} p_{2,j}^i a_1^j + b_2^i\right)$$

Neuron i from layer 2

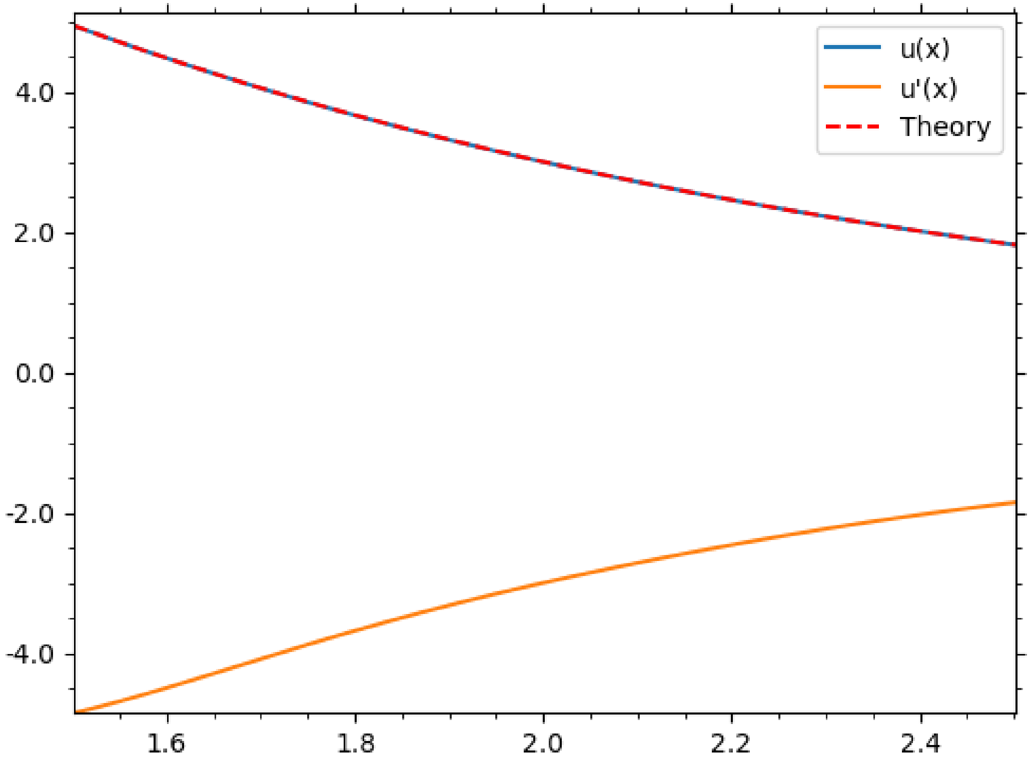
$$a_{n-1}^i = act\left(\sum_{j=1}^{l} p_{n-1,j}^i a_1^j + b_2^i\right)$$

Neuron i from layer $n-1$

Gradient descent on each $p_{m,j}^i$ and b_m^i

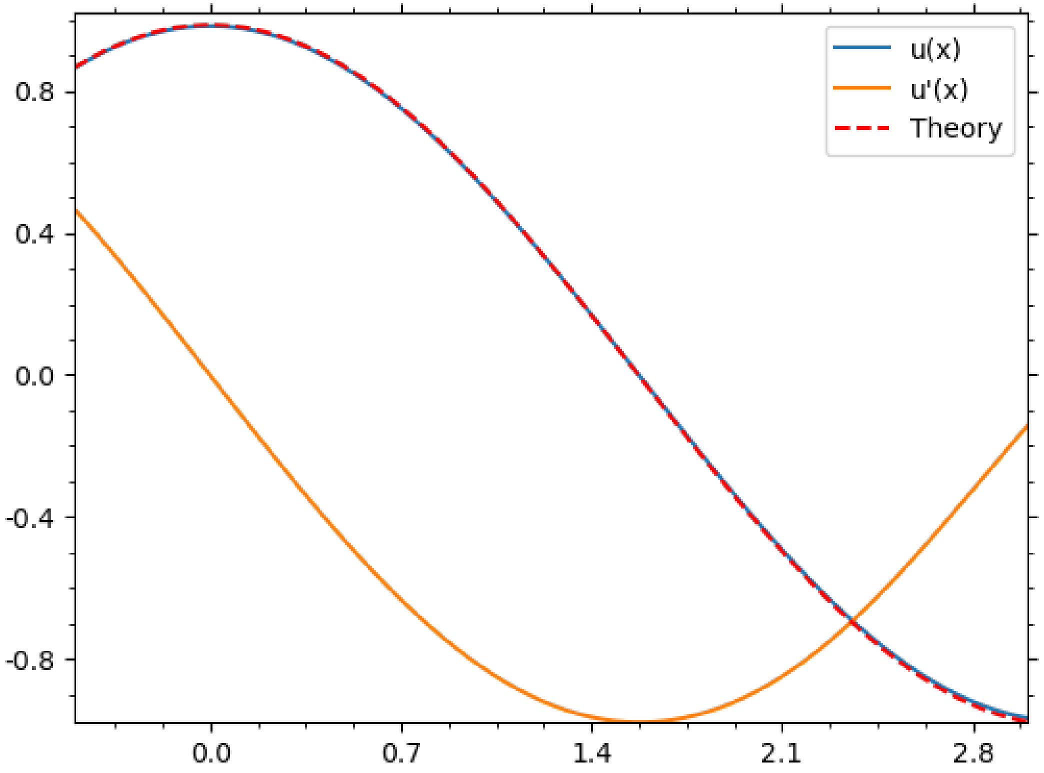
III – ODE resolution using neural network

Examples of ODE with analytical solutions



$$y' + y = 0$$

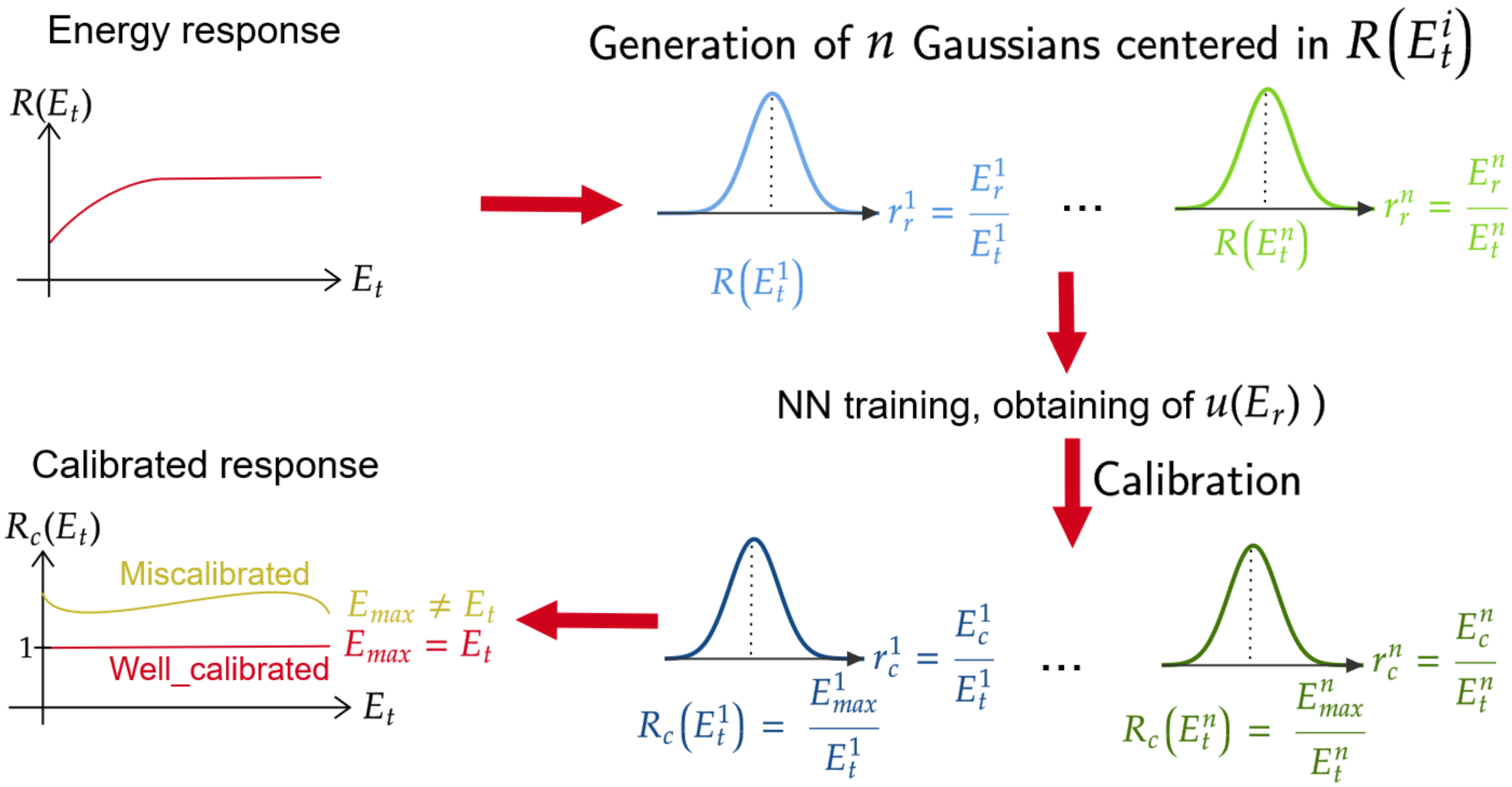
BC: $(2, u(2) = 3)$



$$y'' + y = 0$$

BC: $(0, u(0) = 1)(\pi/2, u(\pi/2) = 0)$

IV – Application to the calibration function



Energy response

Generation of n Gaussians centered in $R\left(E_t^i\right)$

NN training, obtaining of $u(E_r)$)

Calibration

Calibrated response

Main quantitites

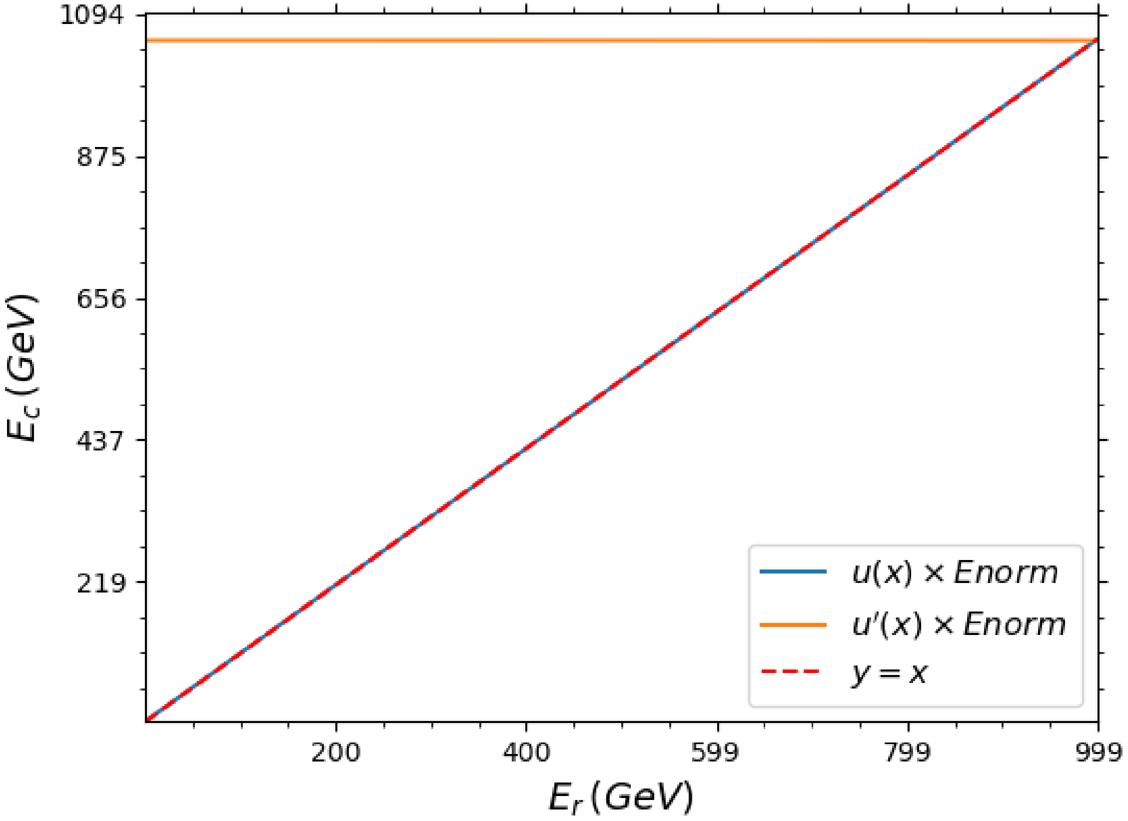
E_r	Measured energy
E_c	Calibrated energy
x	Normalized calibrated energy
E_t	Energy of the maximum of $g(E_c, E_t)$
$R(E_t) = \frac{E_r^*}{E_t}$	Energy response
$u(E_r) = E_c$	Calibration function
E_{max}	Energy of the maximum of the distribution at NN output

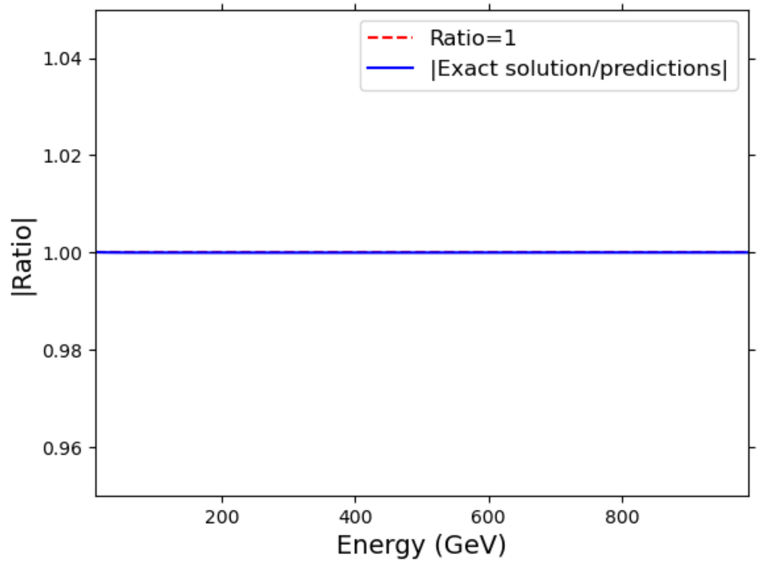
IV – Application to the calibration function

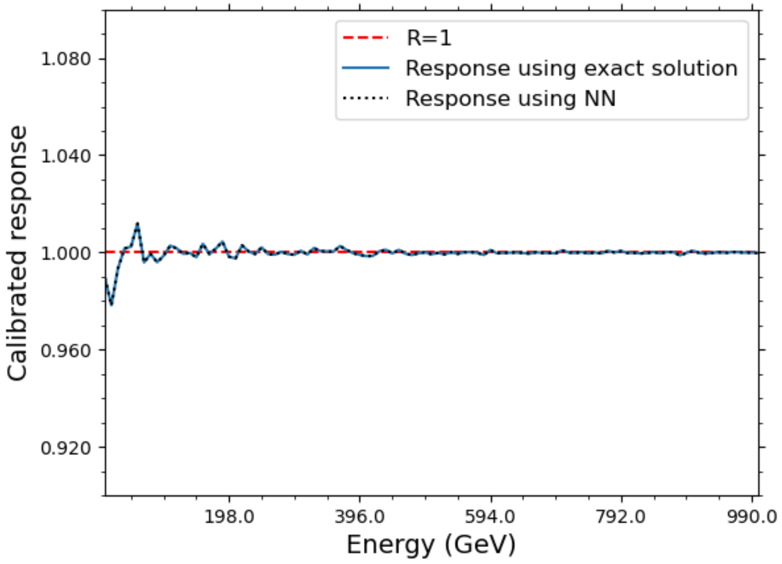
Analytic solution :

$\longrightarrow R(x) = \alpha + \frac{b}{x}, \quad \sigma(x) = 0.1$

$\longrightarrow u(x) = \frac{x-b}{\alpha}$ (Analytic solution)





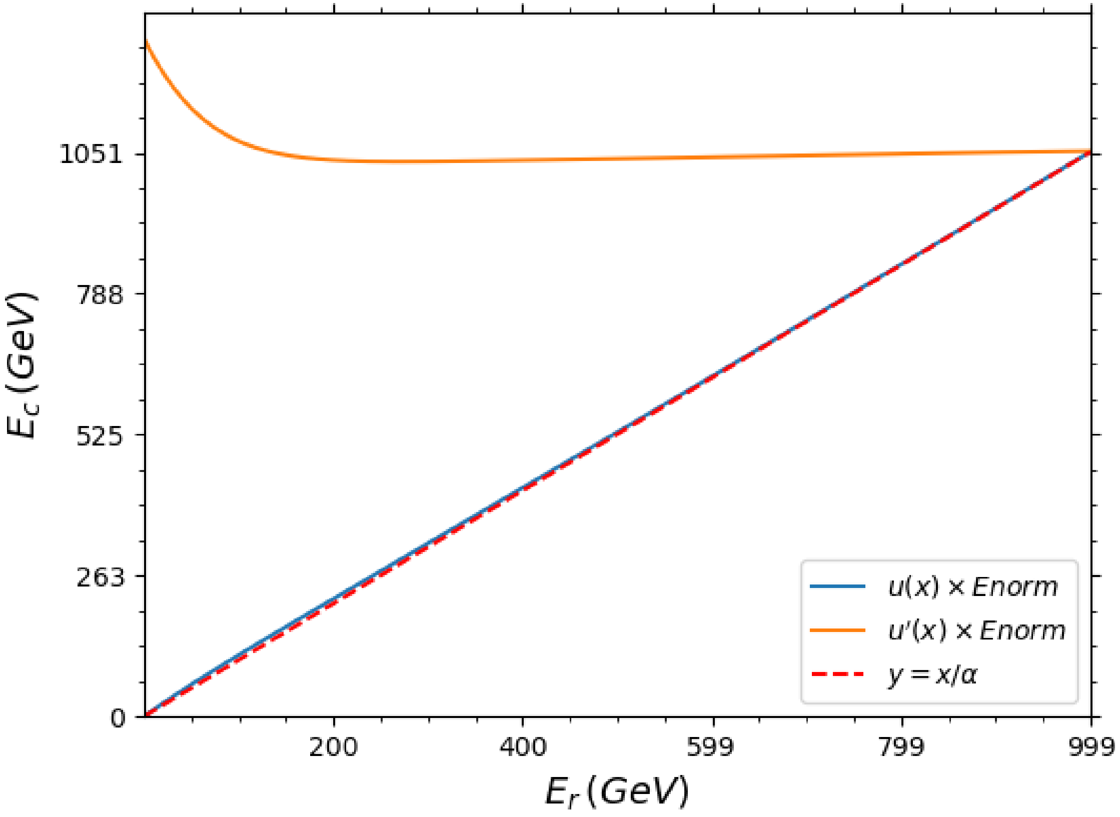


IV – Application à la fonction de calibration

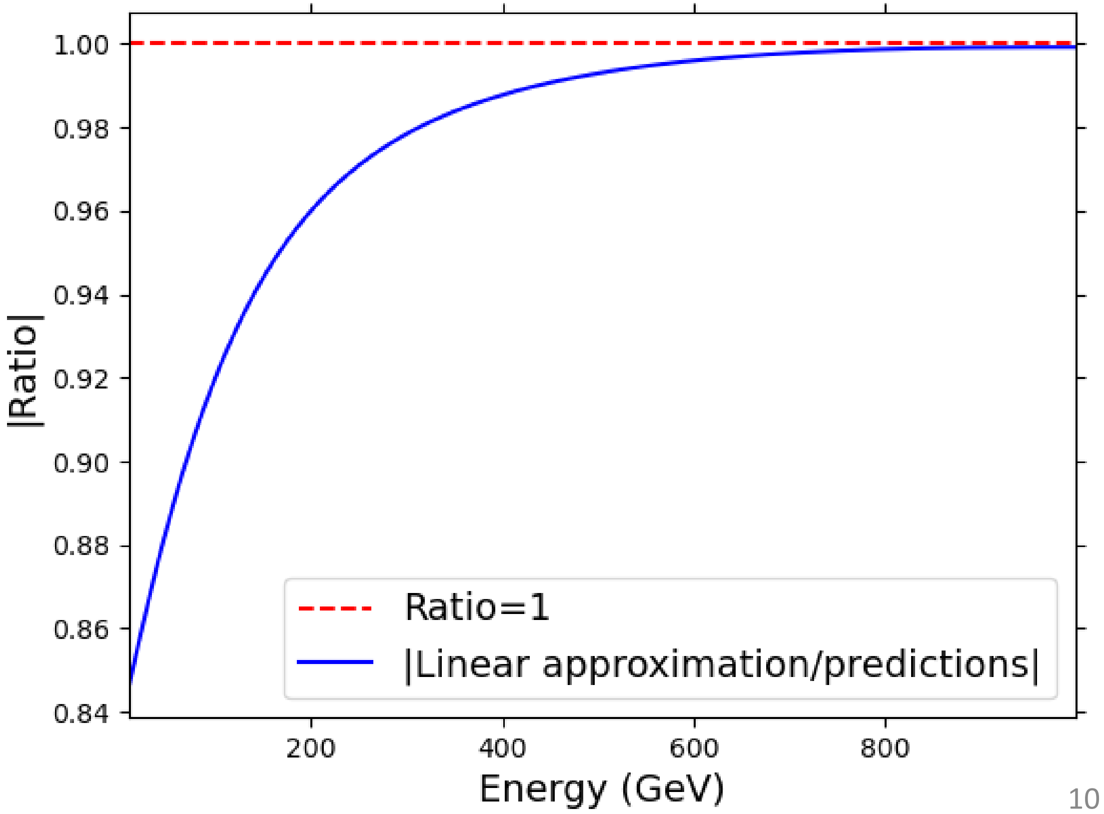
Realistic energy response :

$$\longrightarrow R(x) = \alpha\left(1 + \frac{1}{(b+x)^9}\right) \ , \quad \sigma(x) = Ax + b$$

Plot



Comparison to linear approximation $u(x) = x/\alpha$

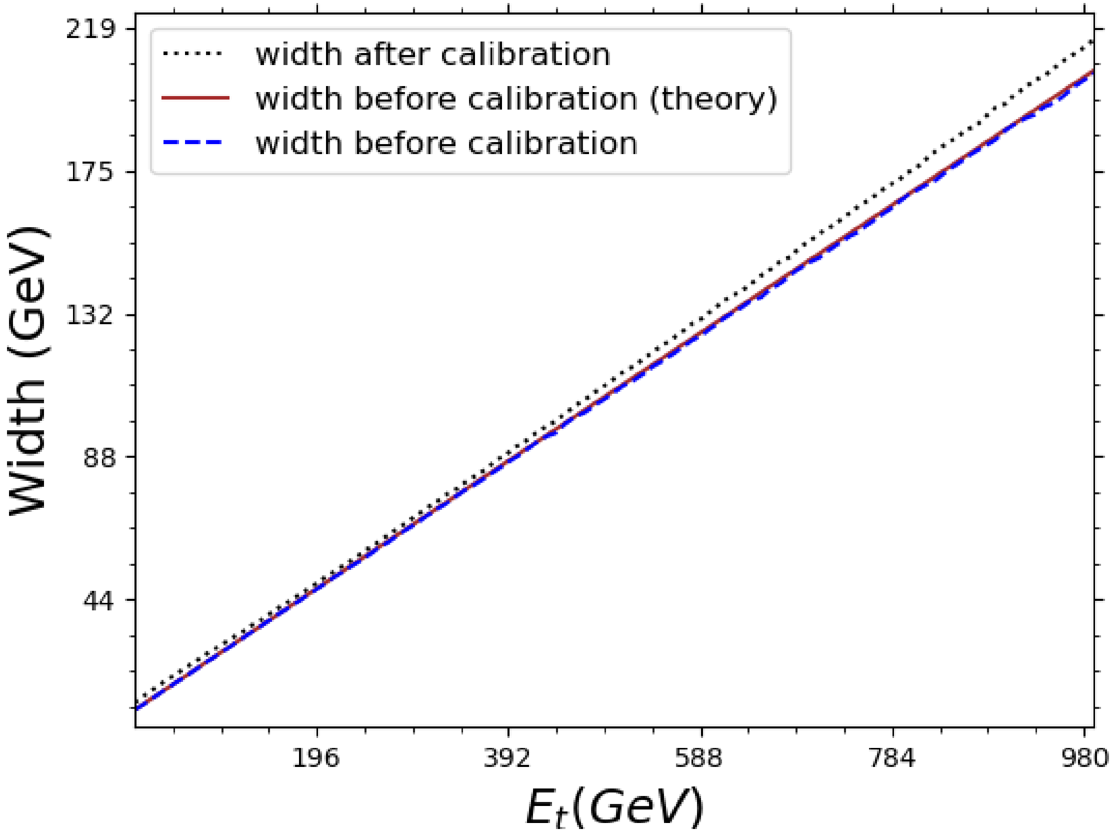


IV – Application à la fonction de calibration

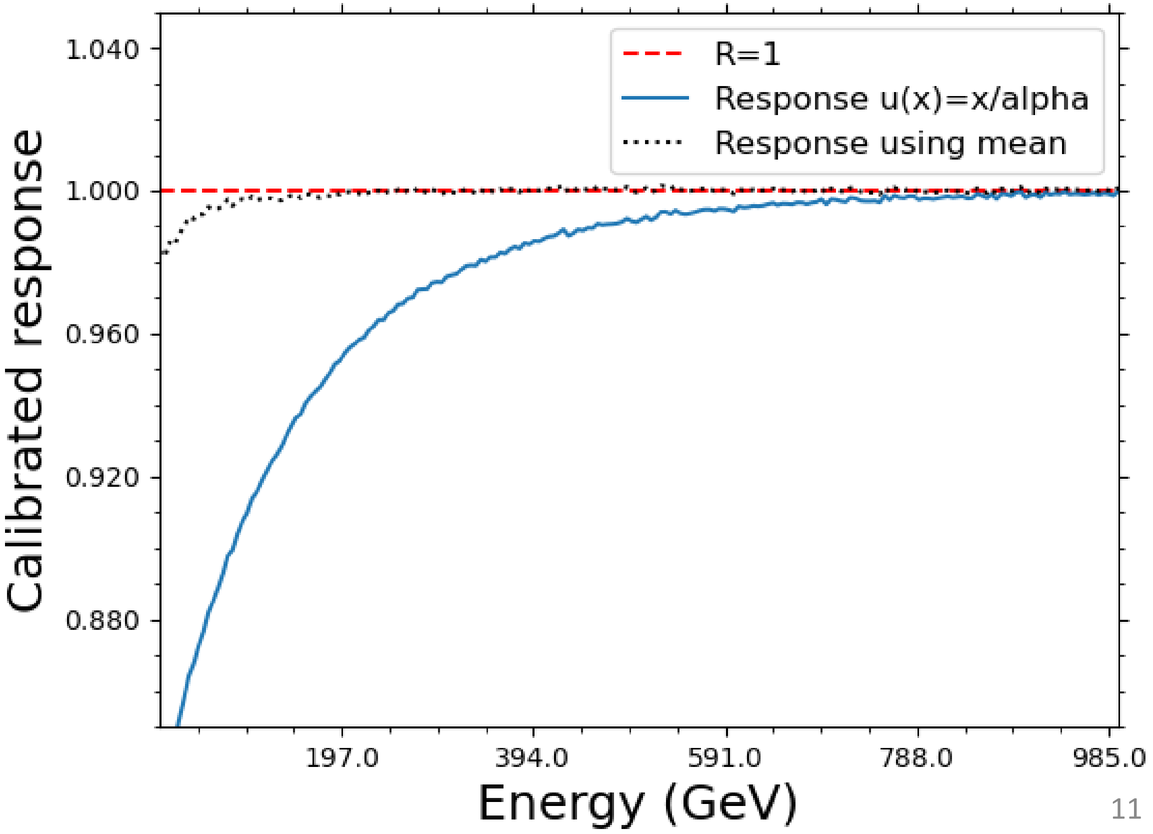
Realistic energy response :

$$\longrightarrow R(x) = \alpha\left(1 + \frac{1}{(b+x)^9}\right), \quad \sigma(x) = Ax + b$$

Larger distributions after calibration



Well calibrated with NN while linear approximation does not work.

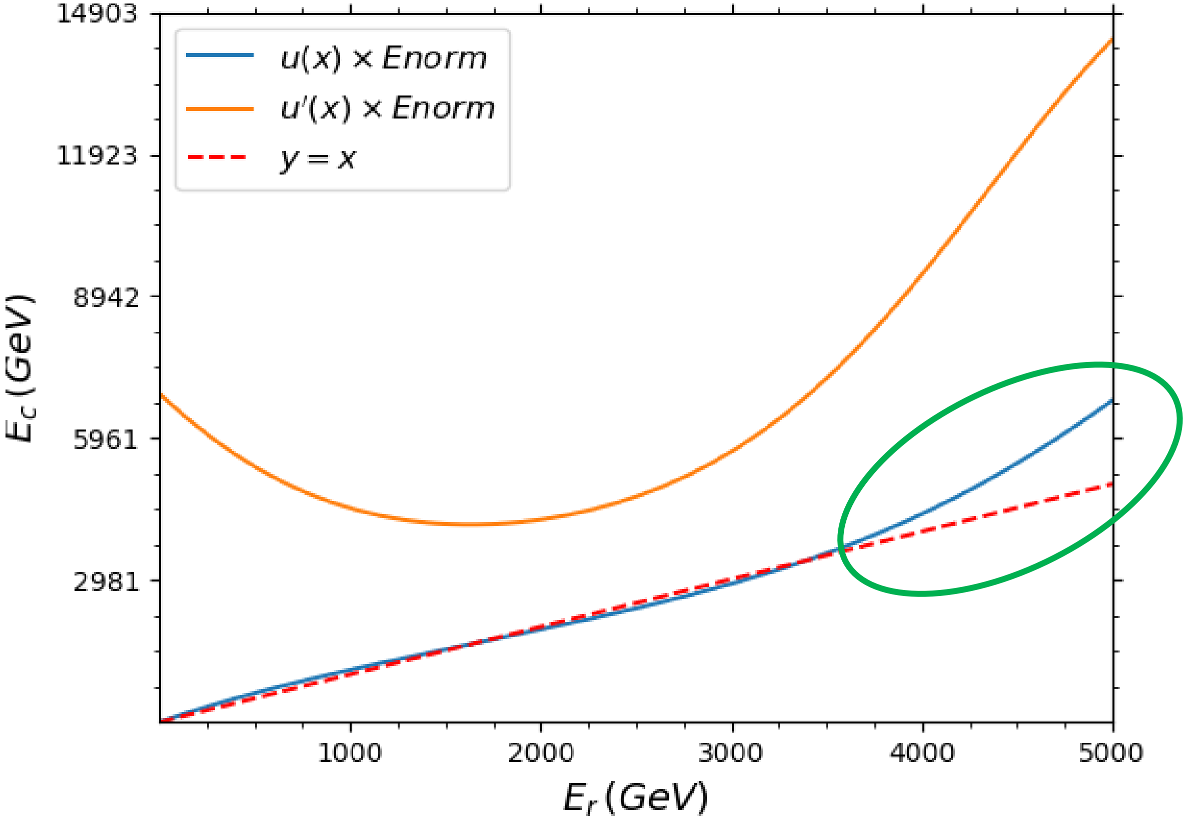


IV – Application to the calibration function

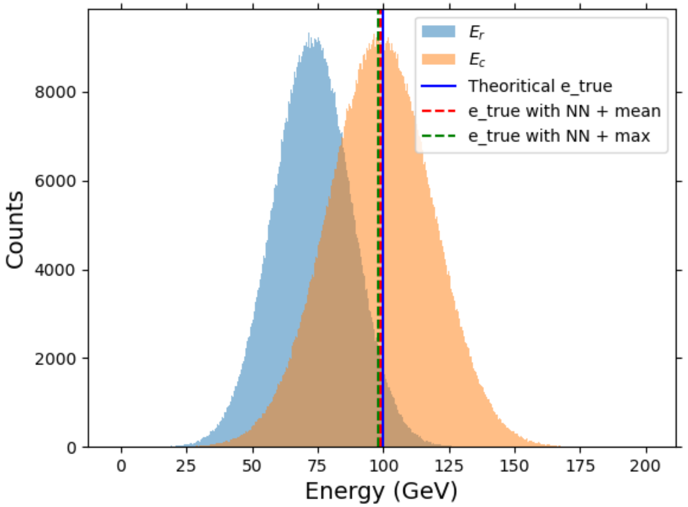
Realistic mass response :

$$\longrightarrow R(x) = \alpha \left(1 + \frac{1}{(b+x)^9}\right), \quad \sigma_m(x) = \sigma_a x + \sigma_b + \sigma_c x^2$$

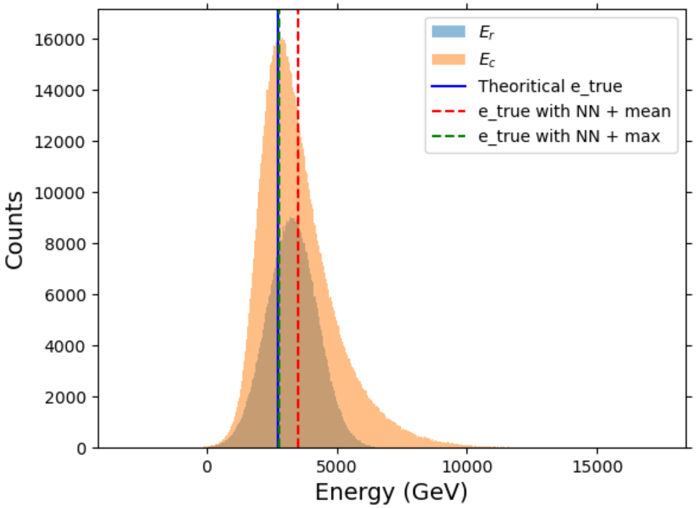
Plot : unexpected result at high energy



For $E_t = 100GeV$ → Symmetrical distribution



For $E_t = 2750GeV$ → Asymmetrical distribution

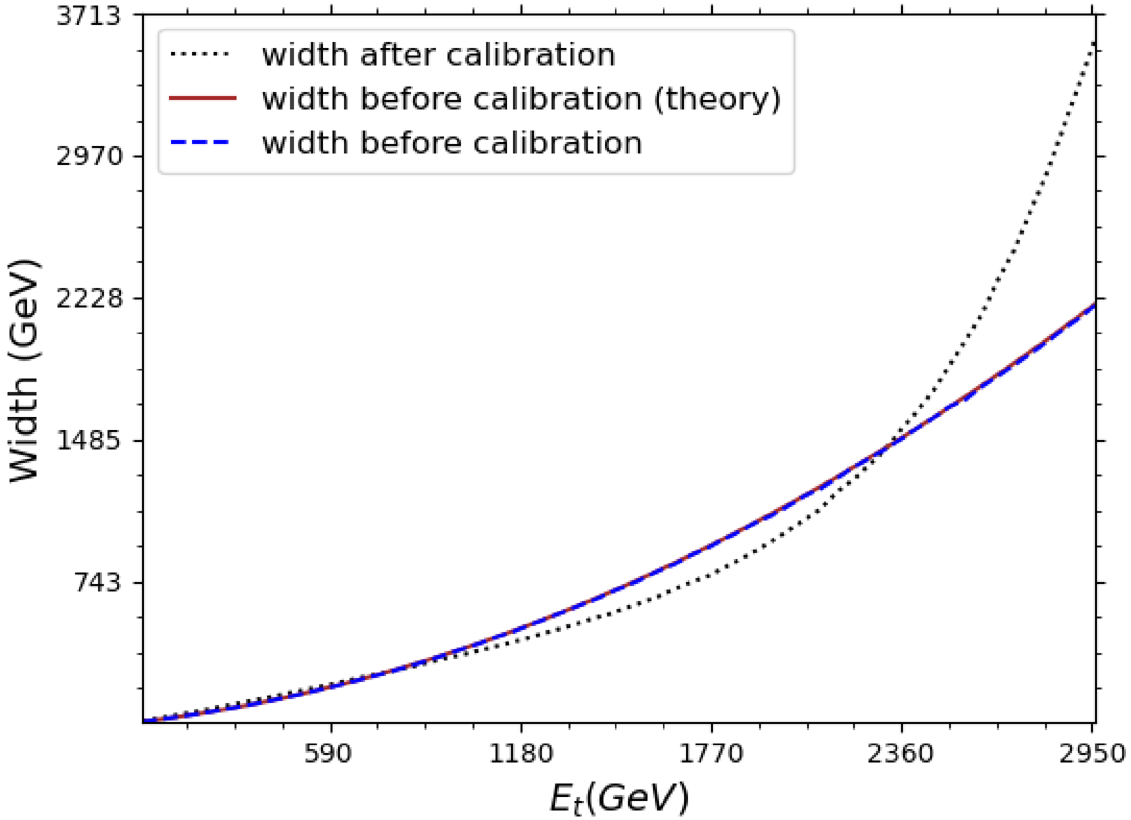


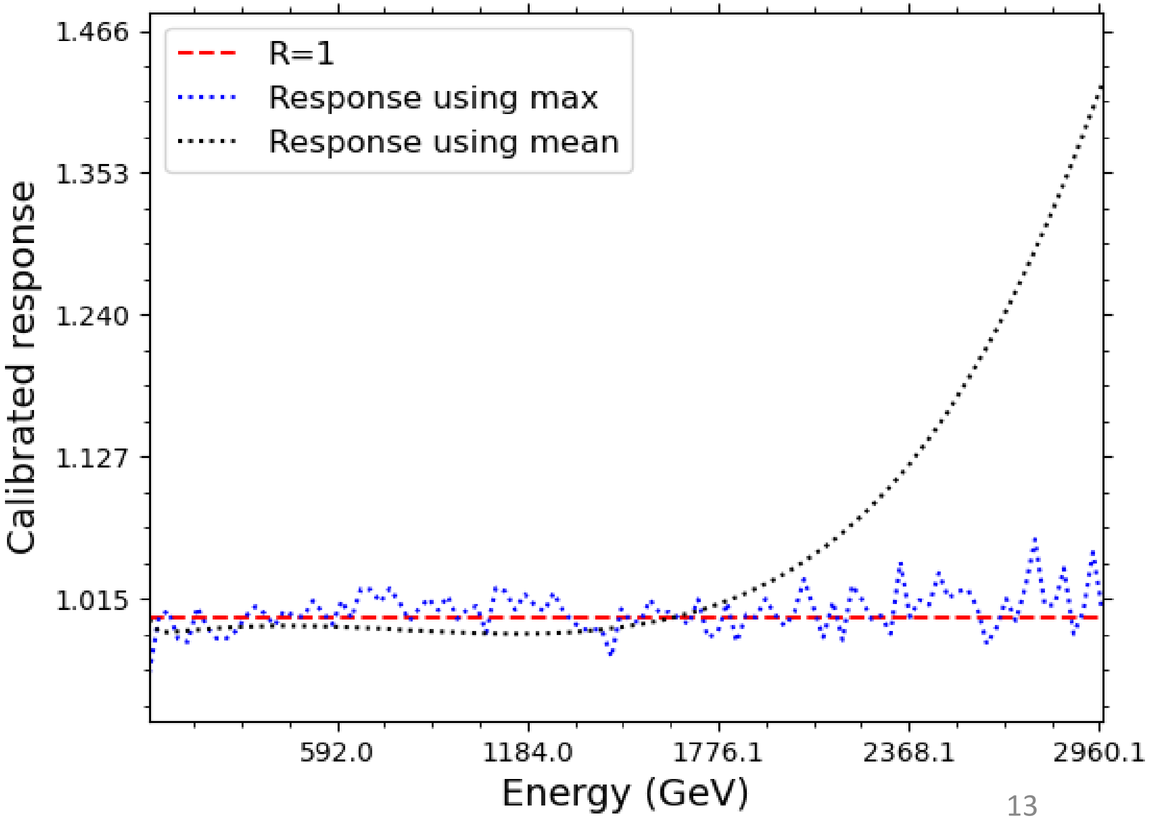
IV – Application to the calibration function

Realistic mass response :

$$\longrightarrow R(x) = \alpha\left(1+\frac{1}{(b+x)^9}\right), \quad \sigma_m(x) = \sigma_a x + \sigma_b + \sigma_c x^2$$

Well calibrated using maximum of distribution instead of the approximation of mean, to avoid problem with asymetrical distribution.





V - Conclusion

Neural network capable of solving **non-linear 2nd-order differential equations.**

Solve the differential equation on the **calibration function** for the response:

- With an analytical solution,
- In **energy**
- In **mass**.

Next :

Qualitative **comparison** with current methods.

Integration of the method into the analysis chain using NN.

ANNEXES- Back-up

$$R(x) = \alpha\left(1 + \frac{1}{(b+x)^p}\right)$$