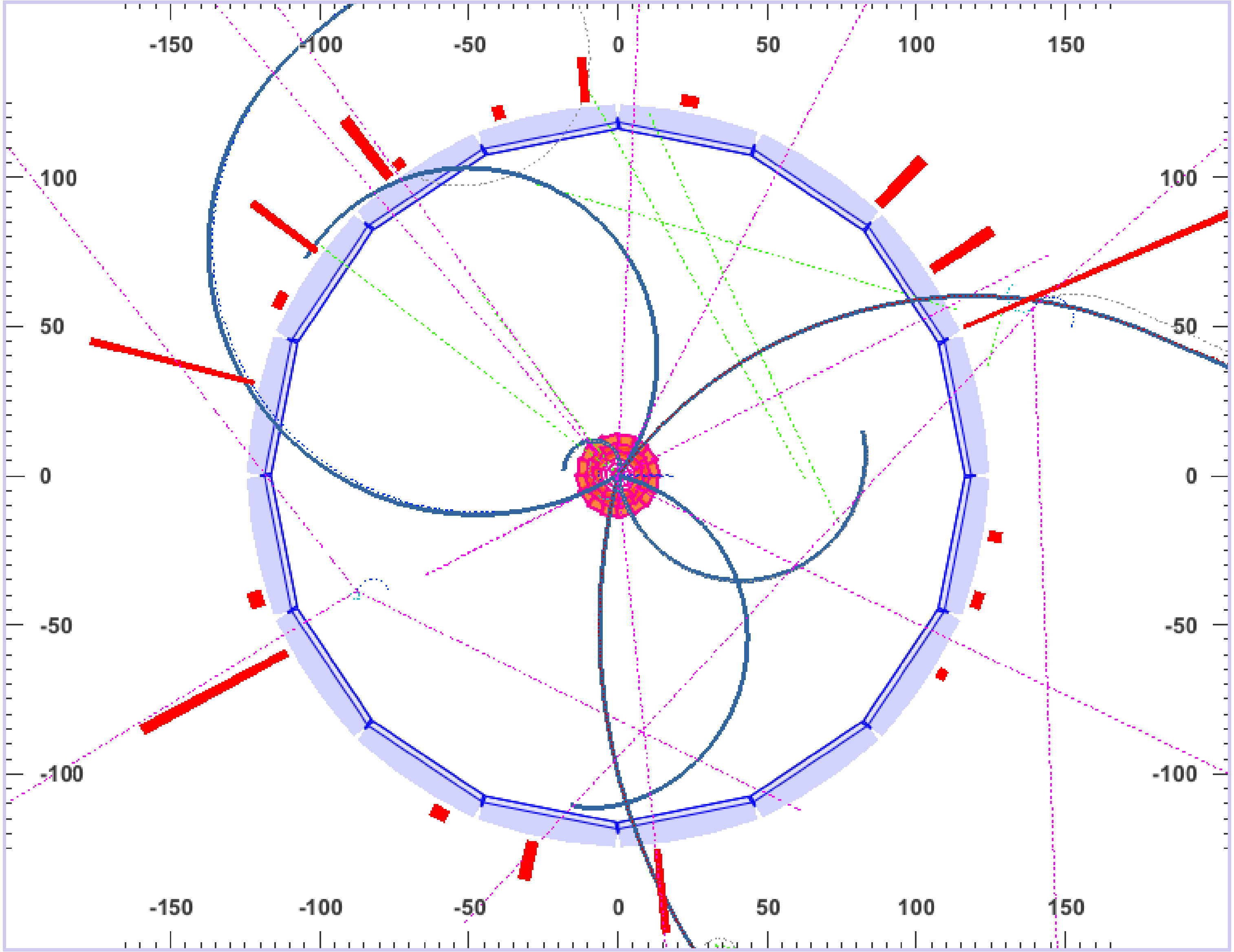
B Vertex Fitting with Graph Neural Networks

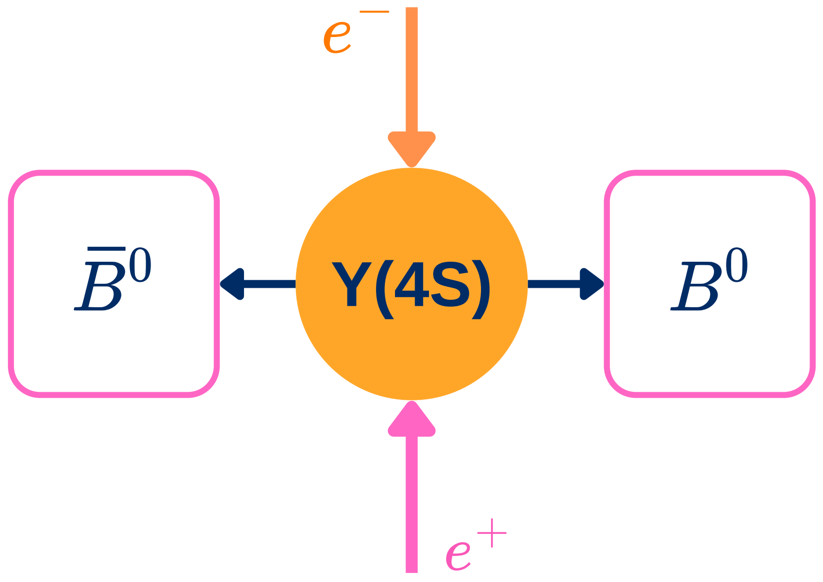


Daria Senina, M2 PSA
Supervisor: Jérôme Baudot

19/06/2025

Motivation

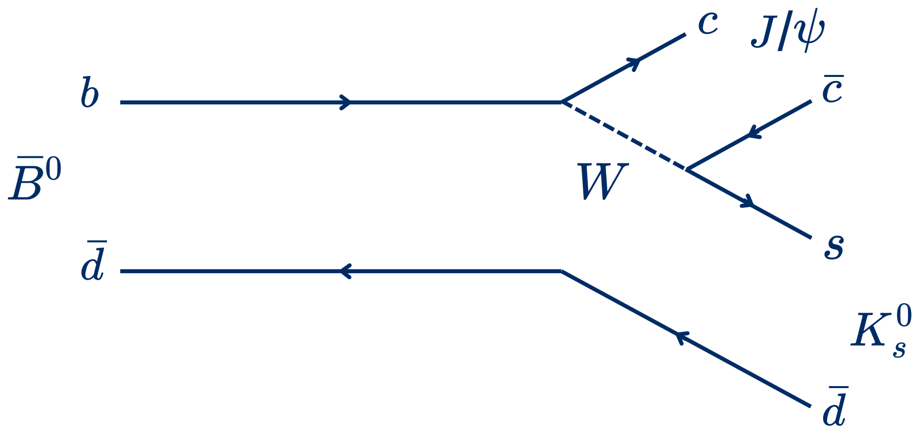
Time Dependent CP Violation in B meson systems



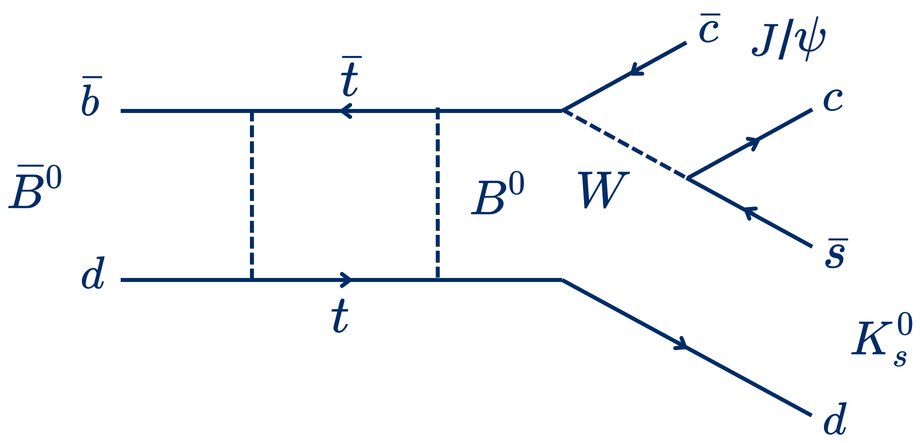
B meson system

- **Matter-antimatter asymmetry → CP violation**
- **Indirect CP violation: interference between neutral B meson mixing and decay**

$$A_{CP}(\Delta t) = \frac{\Gamma\left(\bar{B}^0 \to f_{CP}\right)(\Delta t) - \Gamma\left(B^0 \to f_{CP}\right)(\Delta t)}{\Gamma\left(\bar{B}^0 \to f_{CP}\right)(\Delta t) + \Gamma\left(B^0 \to f_{CP}\right)(\Delta t)} \propto S\sin\left(\Delta m_d \Delta t\right) - C\cos\left(\Delta m_d \Delta t\right)$$



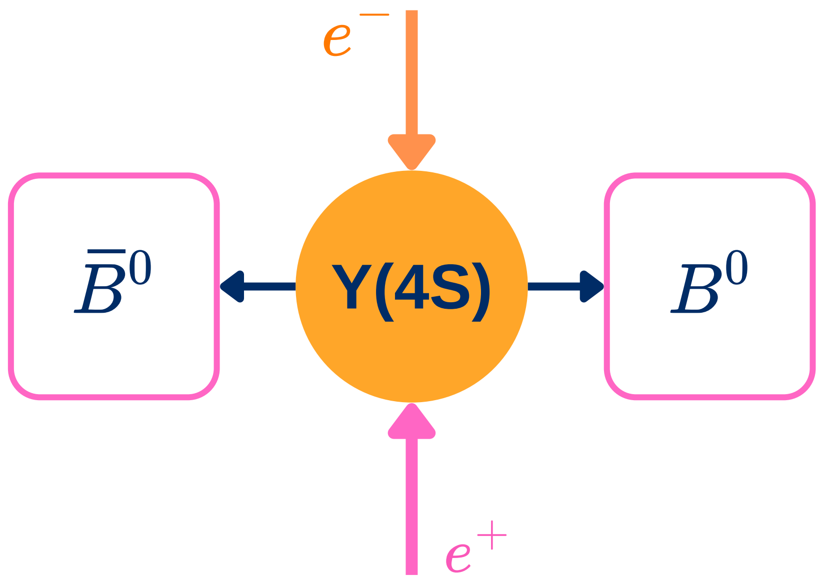
Direct decay



Mixing decay

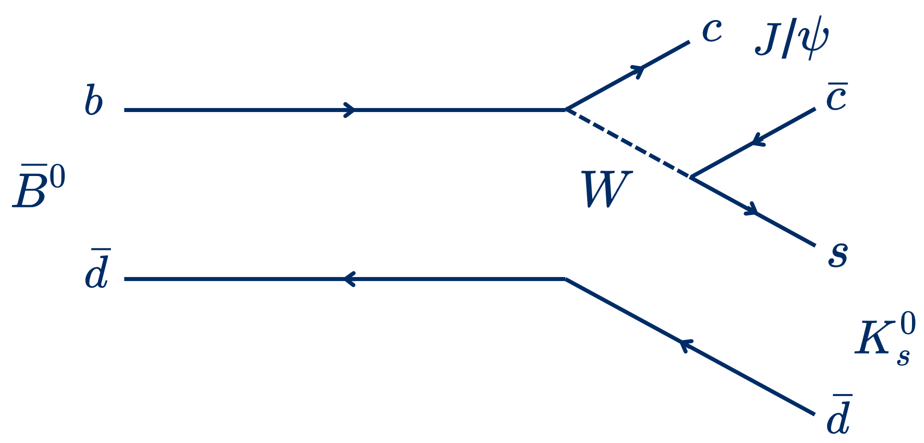
Time Dependent CP Violation in B meson systems

- **Matter-antimatter asymmetry → CP violation**
- **Indirect CP violation: interference between neutral B meson mixing and decay**

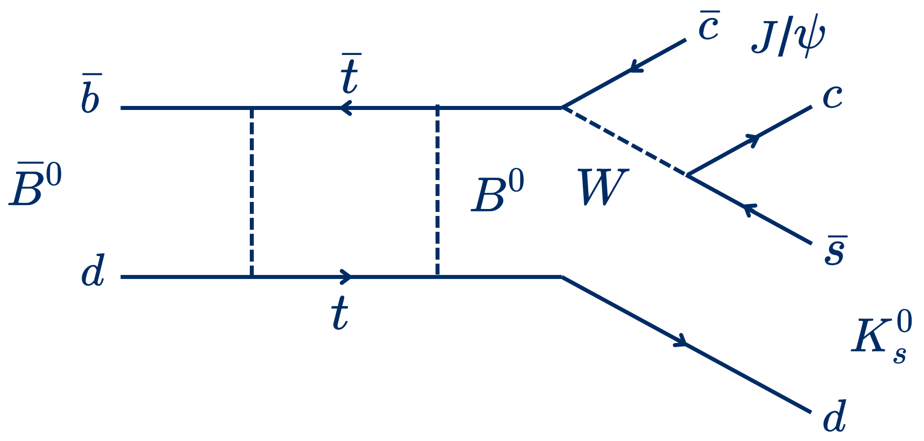


B meson system

$$A_{CP}(\Delta t) = \frac{\Gamma\left(\bar{B}^0 \to f_{CP}\right)(\Delta t) - \Gamma\left(B^0 \to f_{CP}\right)(\Delta t)}{\Gamma\left(\bar{B}^0 \to f_{CP}\right)(\Delta t) + \Gamma\left(B^0 \to f_{CP}\right)(\Delta t)} \propto S \sin\left(\Delta m_d \Delta t\right) - C \cos\left(\Delta m_d \Delta t\right)$$



Direct decay



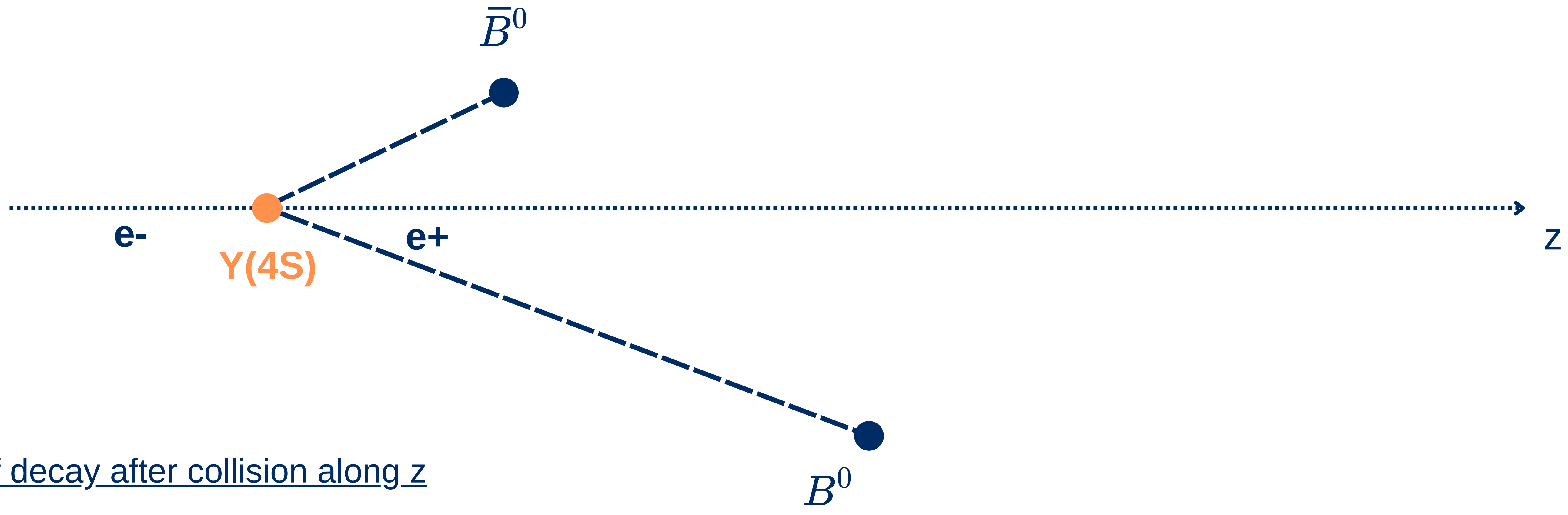
Mixing decay

Goal: measure Δt

Motivation

Time Dependent CP Violation $S\sin(\Delta m_d \Delta t) - C\cos(\Delta m_d \Delta t)$

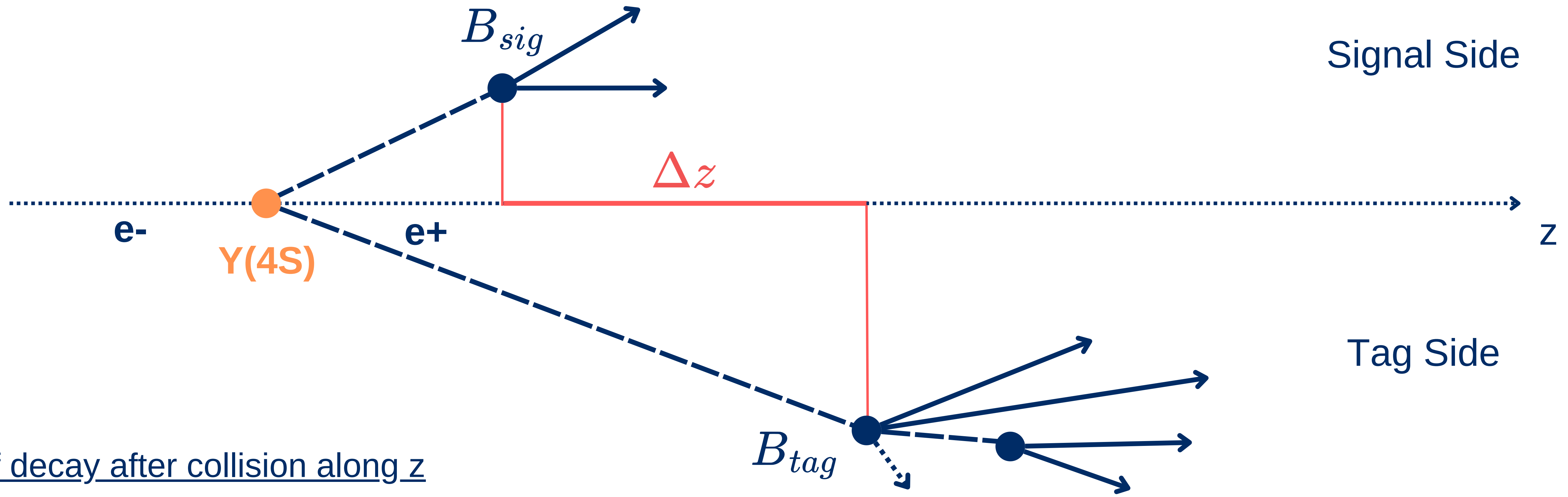
- **Boost along z axis**



Example of decay after collision along z

Time Dependent CP Violation $S\sin(\Delta m_d \Delta t) - C\cos(\Delta m_d \Delta t)$

- **Boost along z axis**
- **To get :** $\Delta t \leftarrow \Delta z$

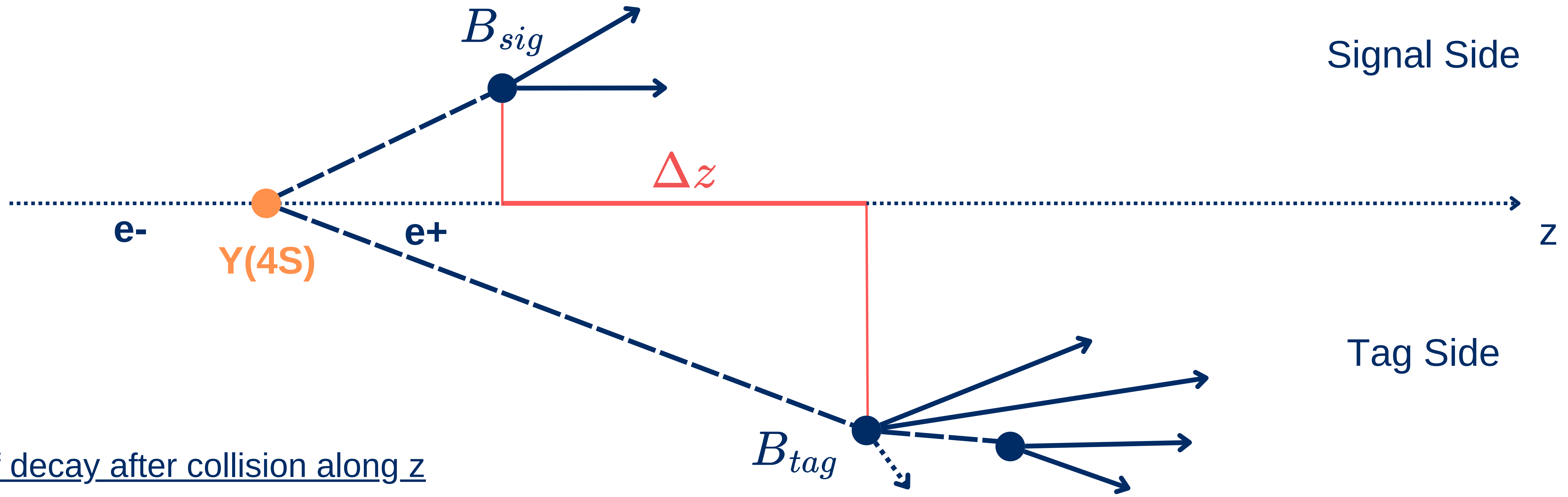


Example of decay after collision along z

Motivation

<u>**Time Dependent CP Violation**</u> $[S\sin(\Delta m_d \Delta t) - C\cos(\Delta m_d \Delta t)] \times res_{\Delta t}$

- **Boost along z axis**
- **To get :** $\Delta t \leftarrow \Delta z \leftarrow$ **precise Btag z vertex position since:** $res_{\Delta t} = \dfrac{res_{\Delta z}}{\gamma\beta}$

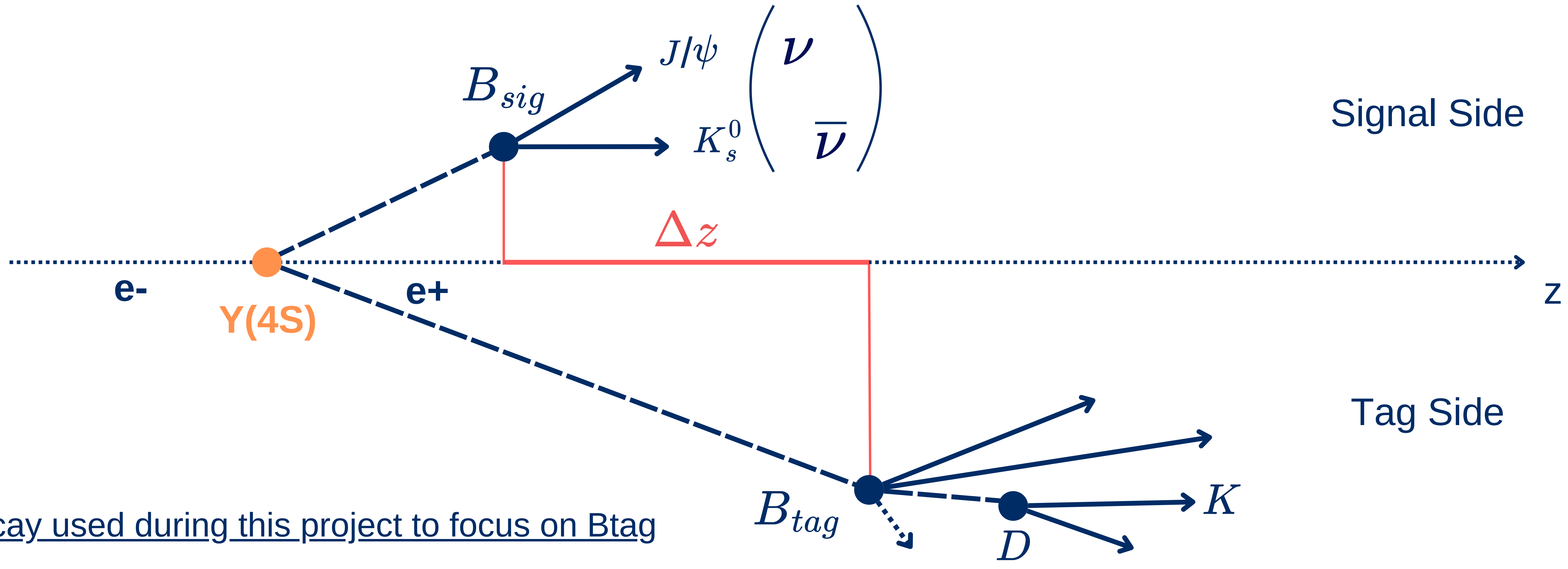


Example of decay after collision along z

Motivation

Time Dependent CP Violation $[S\sin(\Delta m_d \Delta t) - C\cos(\Delta m_d \Delta t)] \times res_{\Delta t}$

- **Bsig: exclusive reconstruction (low efficiency),** $J/\psi \to \mu^+\mu^-$
- **Btag: inclusive reconstruction (high efficiency), TagV using global vertex fit**



Type of decay used during this project to focus on Btag

The Experiment

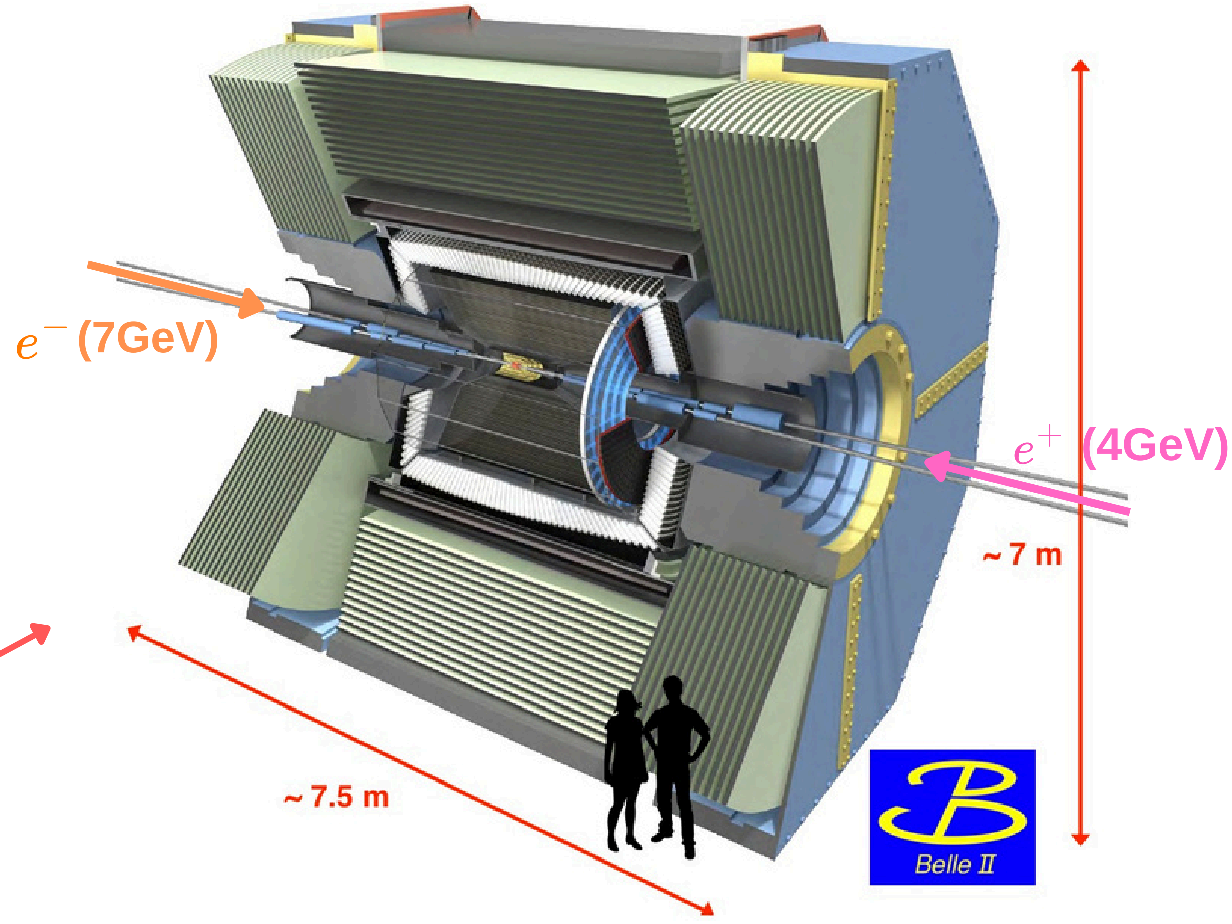
- **Belle II experiment, SuperKEKB Japan**

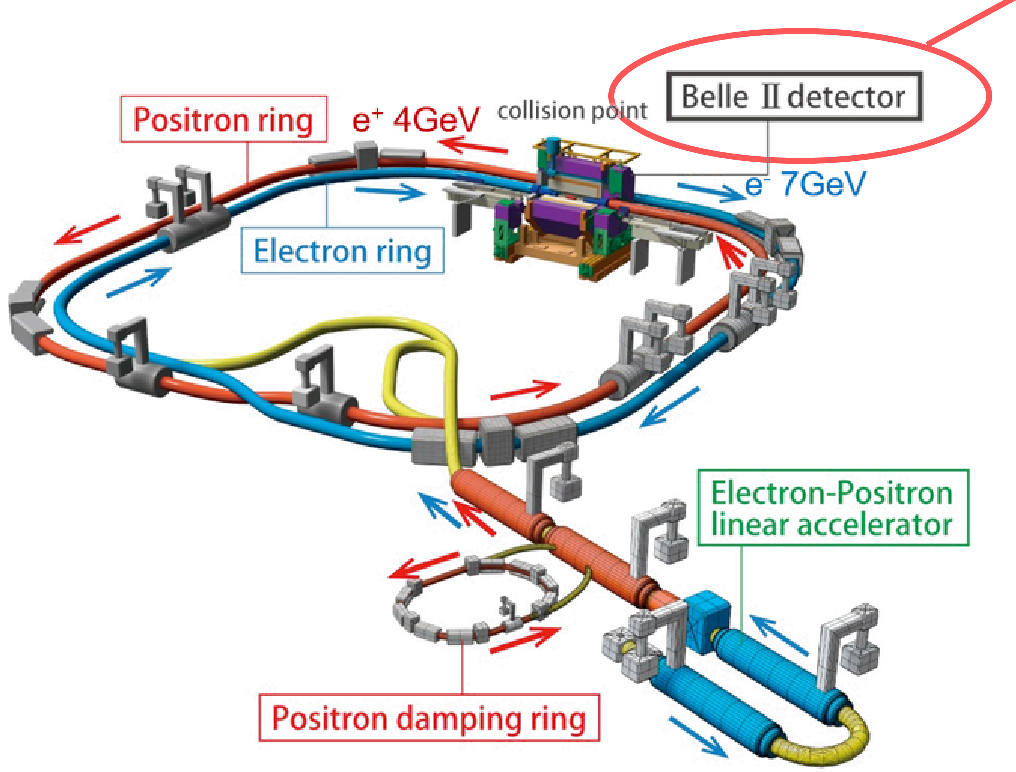
  → Flavor physics, dark matter...

- **Energy Asymmetric → boosted collision**

- **Creation of Υ(4S) → B physics**

  B mesons : $B^0\,(\bar{b}d)\ B^+\,(\bar{b}u)\ B_c^0\,(\bar{b}c)\ B_s^0\,(\bar{b}s)$





Here : Time Dependent CP Violation

The Belle II detector

Project Overview

- **Goal : Use Neural Networks to predict the B decay vertex position**

- **Building on existing Belle II GraFEI tool**

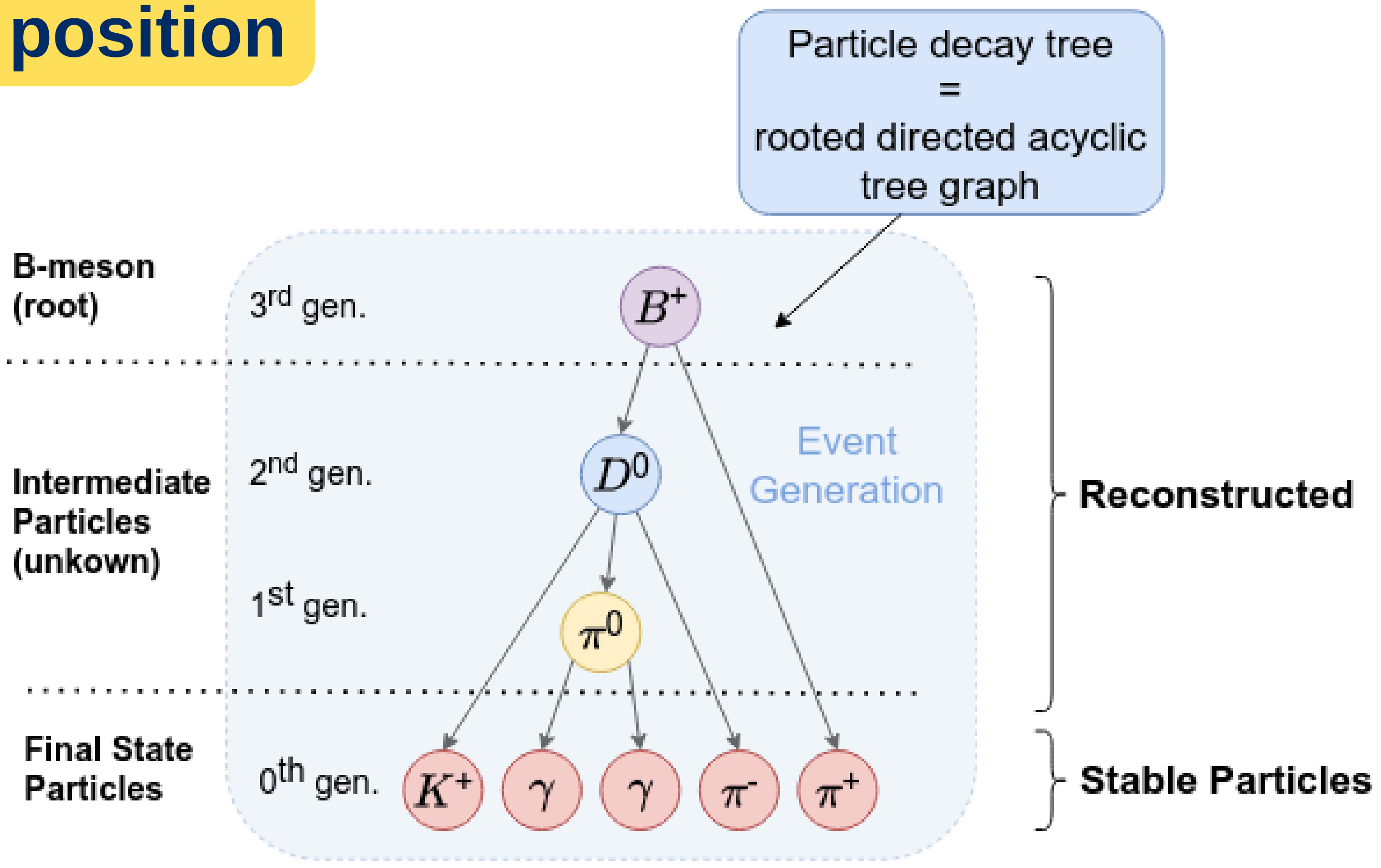
⟶ **Graph Full Event Interpretation**

⟶ **Decay tree = graph**

	B^+	D^0	π^0	K^+	γ	γ	π^-	π^+
B^+	0	1	0	0	0	0	0	1
D^0	1	0	1	1	0	0	1	0
π^0	0	1	0	0	1	1	0	0
K^+	0	1	0	0	0	0	0	0
γ	0	0	1	0	0	0	0	0
γ	0	0	1	0	0	0	0	0
π^-	0	1	0	0	0	0	0	0
π^+	1	0	0	0	0	0	0	0

Adjacency matrix corresponding to decay tree

⟶ **Need knowledge of intermediate particles**



Example of decay tree

Project Overview

- **Goal : Use Neural Networks to predict the B decay vertex position**

- **Building on existing Belle II GraFEI tool**

  ⟶ **Graph Full Event Interpretation**

  ⟶ **Decay tree = graph**

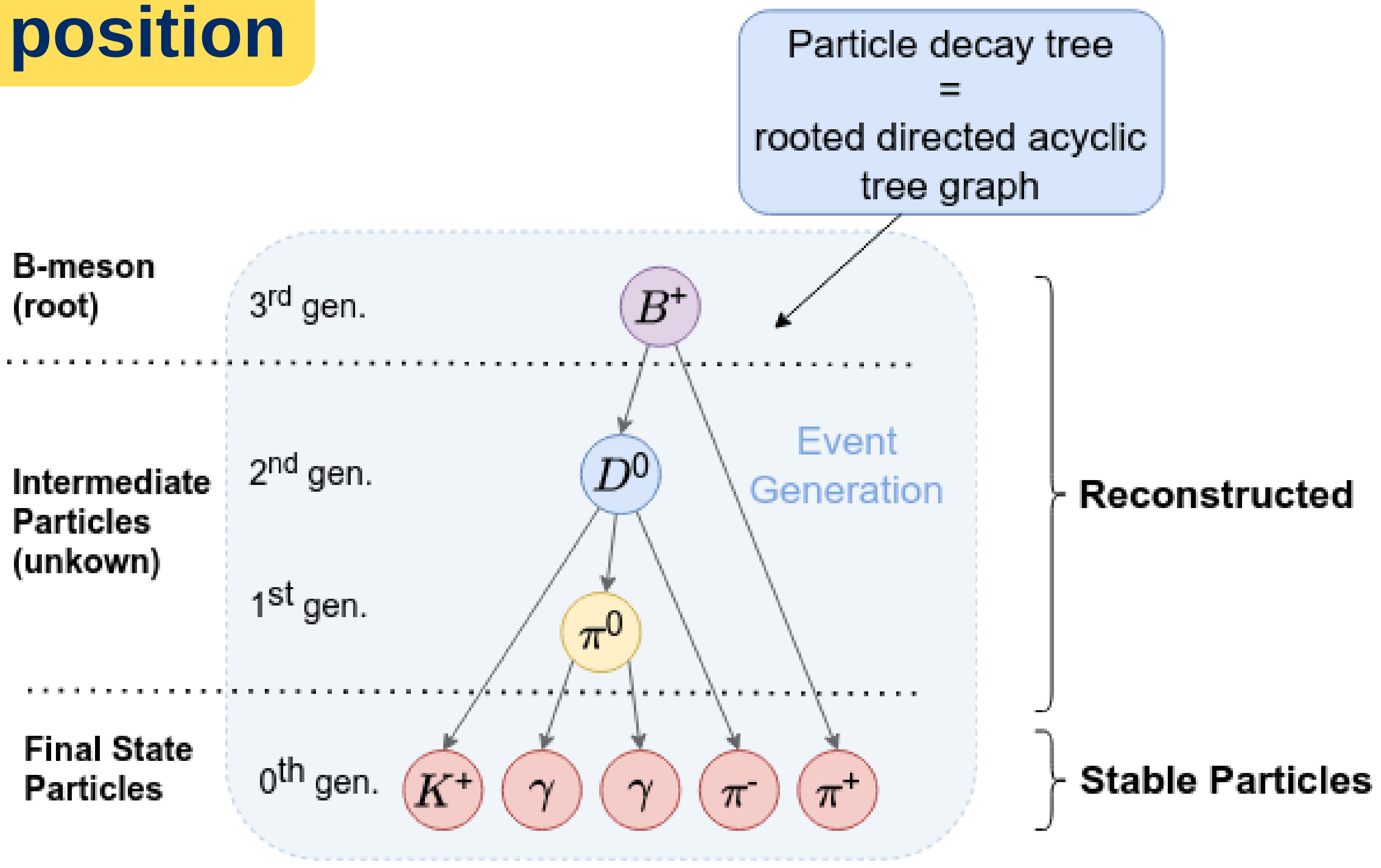
  ⟶ **Learning from leaves only**

	B^+	D^0	π^0	K^+	γ	γ	π^-	π^+
B^+	0	1	0	0	0	0	0	1
D^0	1	0	1	1	0	0	1	0
π^0	0	1	0	0	1	1	0	0
K^+	0	1	0	0	0	0	0	0
γ	0	0	1	0	0	0	0	0
γ	0	0	1	0	0	0	0	0
π^-	0	1	0	0	0	0	0	0
π^+	1	0	0	0	0	0	0	0

⟷

	K^+	γ	γ	π^-	π^+
K^+	K^+	D^0	D^0	D^0	B^+
γ	D^0	γ	π^0	D^0	B^+
γ	D^0	π^0	γ	D^0	B^+
π^-	D^0	D^0	D^0	π^-	B^+
π^+	B^+	B^+	B^+	B^+	π^+

Adjacency to LCA matrix corresponding to decay tree



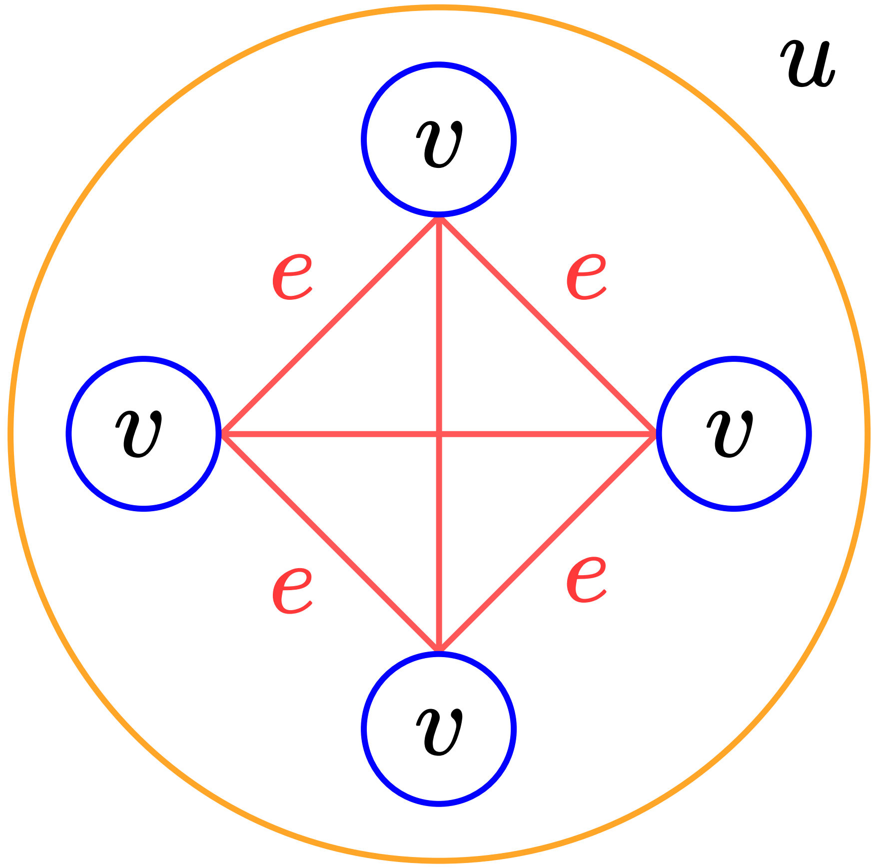
Example of decay tree

Final State Particles ⟶ **GraFEI** ⟶ **LCA matrix** (Lowest Common Ancestor)

Graph Object

- **New task: Not LCA, but estimate Btag vertex position**
- **Graph Object, the data structure :**

Goal : Have vertex position as global feature



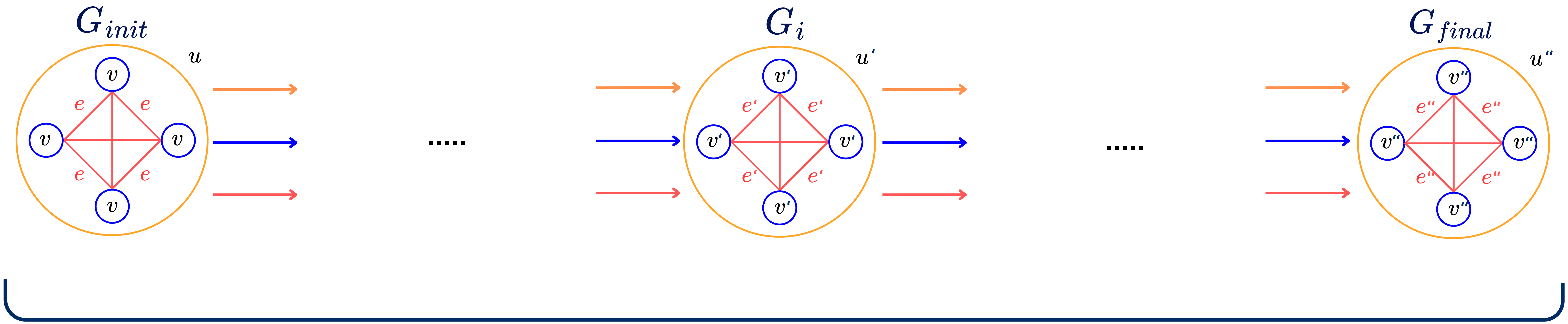
global (u) : vertex position

edges (e) : distance of closest approach between tracks, angle

nodes (v) : particle nature, PDG, pt, pz, dr, dz, charge..

Graph Network

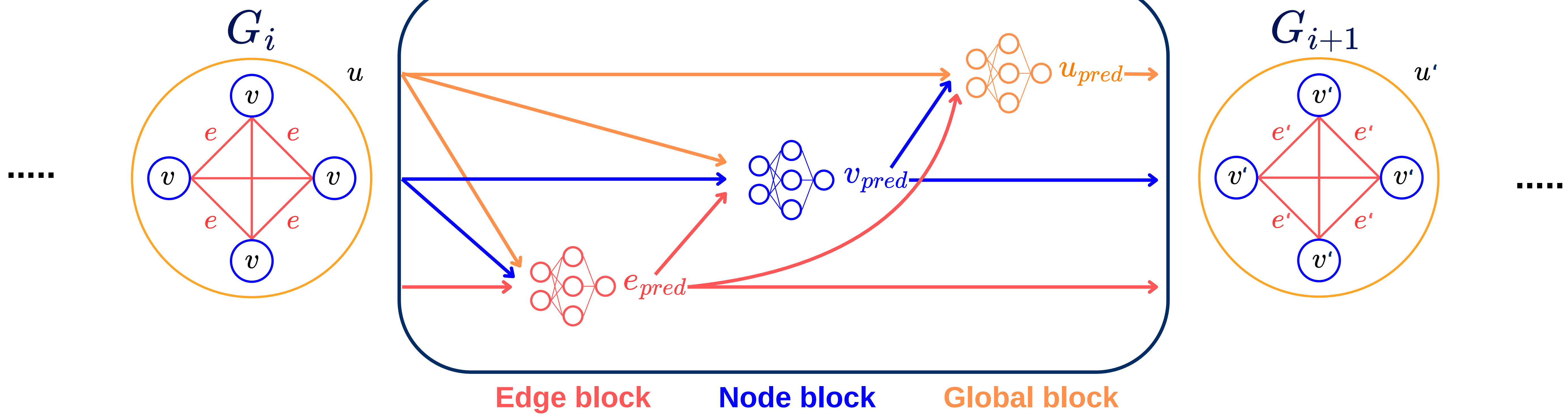
- **The network is composed of N connected graphs**
- **Similar to multilayer neural networks, with weights**



N layers

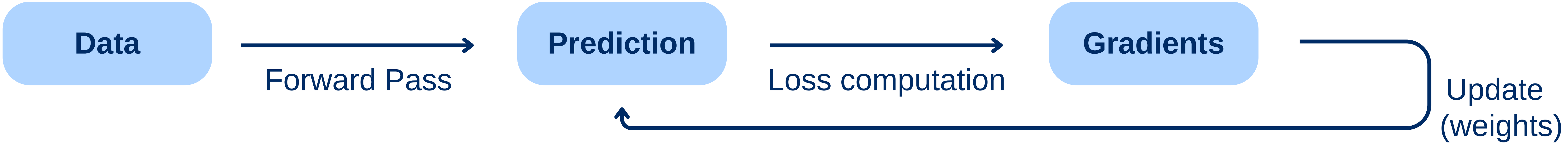
Graph Network Block

- The network is composed of N connected graphs
- Similar to multilayer neural networks, with weights
- With N-1 Network Blocks composed of three sub-blocks

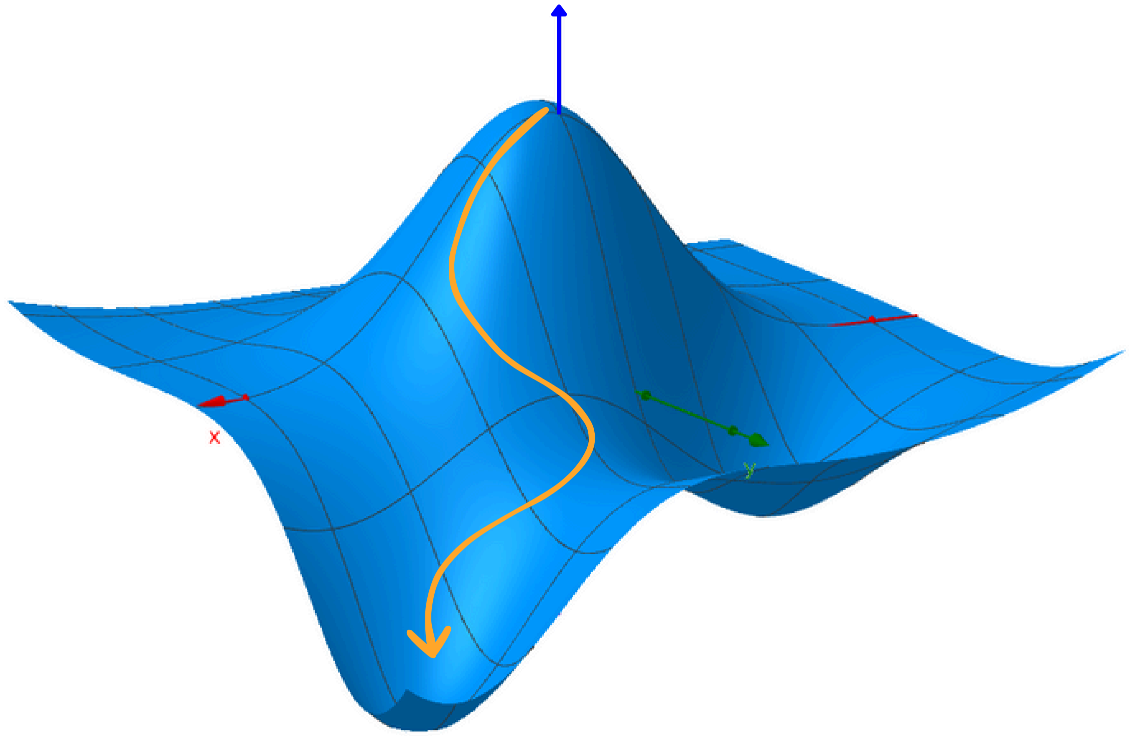


Machine Learning

- **Learns through loss function :** $L = f\,(prediction - truth)$



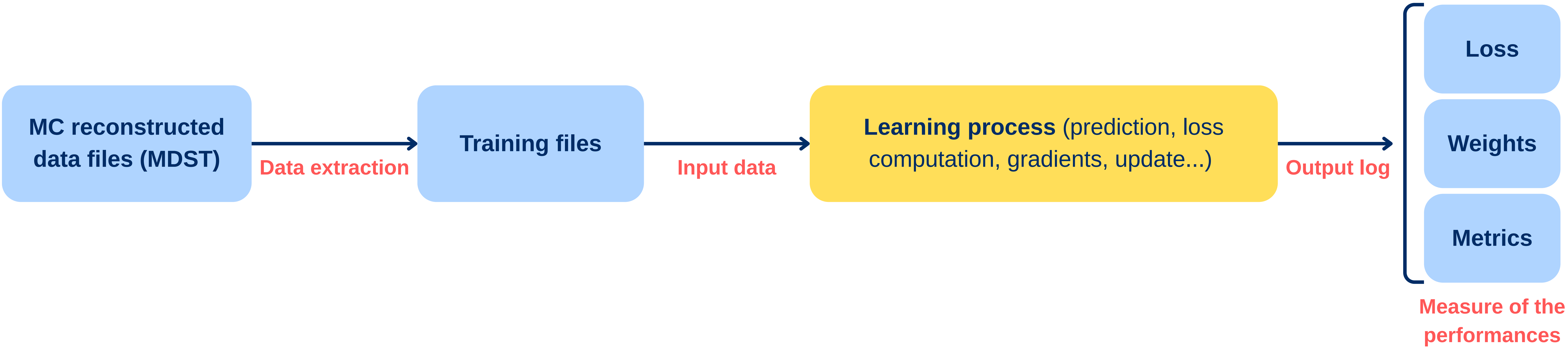
- **Minimization of gradients of the loss, one technique : gradient descent**
- **Minimization over *epochs* (iteration of learning process)**
- **Loss function different for data type**



Loss visualization in 3D

Implementation

- **Extended GraFEI to predict 3 global features : b_decay_x, b_decay_y, b_decay_z**
- **Simplified flow:**



Implementation

- **GraFEI's loss function :** $L = \alpha_{lca} L_{lca} + \alpha_{mass} L_{mass}$
- **New loss function :** $L = \underline{\alpha_{lca} L_{lca}} + \underline{\alpha_{vtx} L_{vtx}} \; (+\alpha_{mass} L_{mass})$

	K^+	γ	γ	π^-	π^+
K^+	K^+	D^0	D^0	D^0	B^+
γ	D^0	γ	π^0	D^0	B^+
γ	D^0	π^0	γ	D^0	B^+
π^-	D^0	D^0	D^0	π^-	B^+
π^+	B^+	B^+	B^+	B^+	π^+

Classes : Cross Entropy

$$e_{pred} = p = [\overset{0}{0.3}, \overset{1}{0.8}, \overset{2}{0.1}, \overset{3}{-0.2}, \overset{4}{-0.7}, \underbrace{\overset{5}{1.2}}_{p_j}]$$

$$e_{truth} = j = 5$$

$$L_{lca} = \sum - \log\left(\frac{e^{p_j}}{\sum_{k=0}^{5} e^{p_k}}\right)$$

Floats : Mean Squared Error

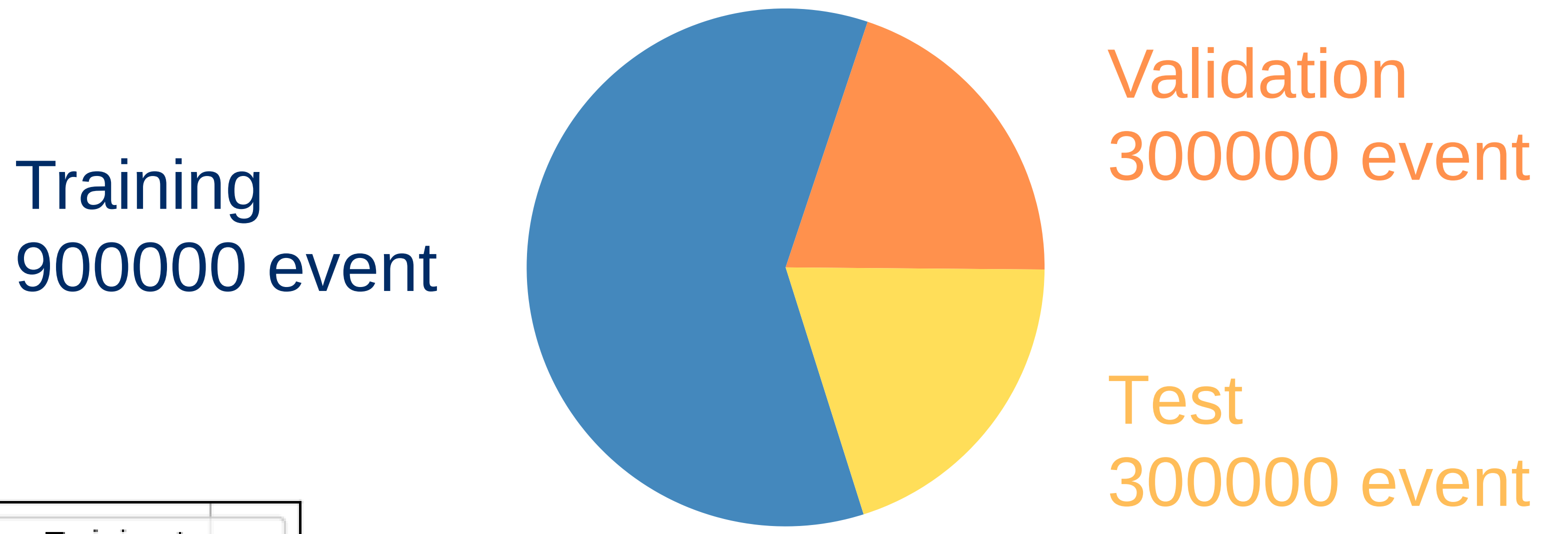
$$L_{vtx} = \frac{1}{N_{event}} \sum (u_{pred} - u_{truth})^2$$

$\longrightarrow$ **Coefficients tuning**

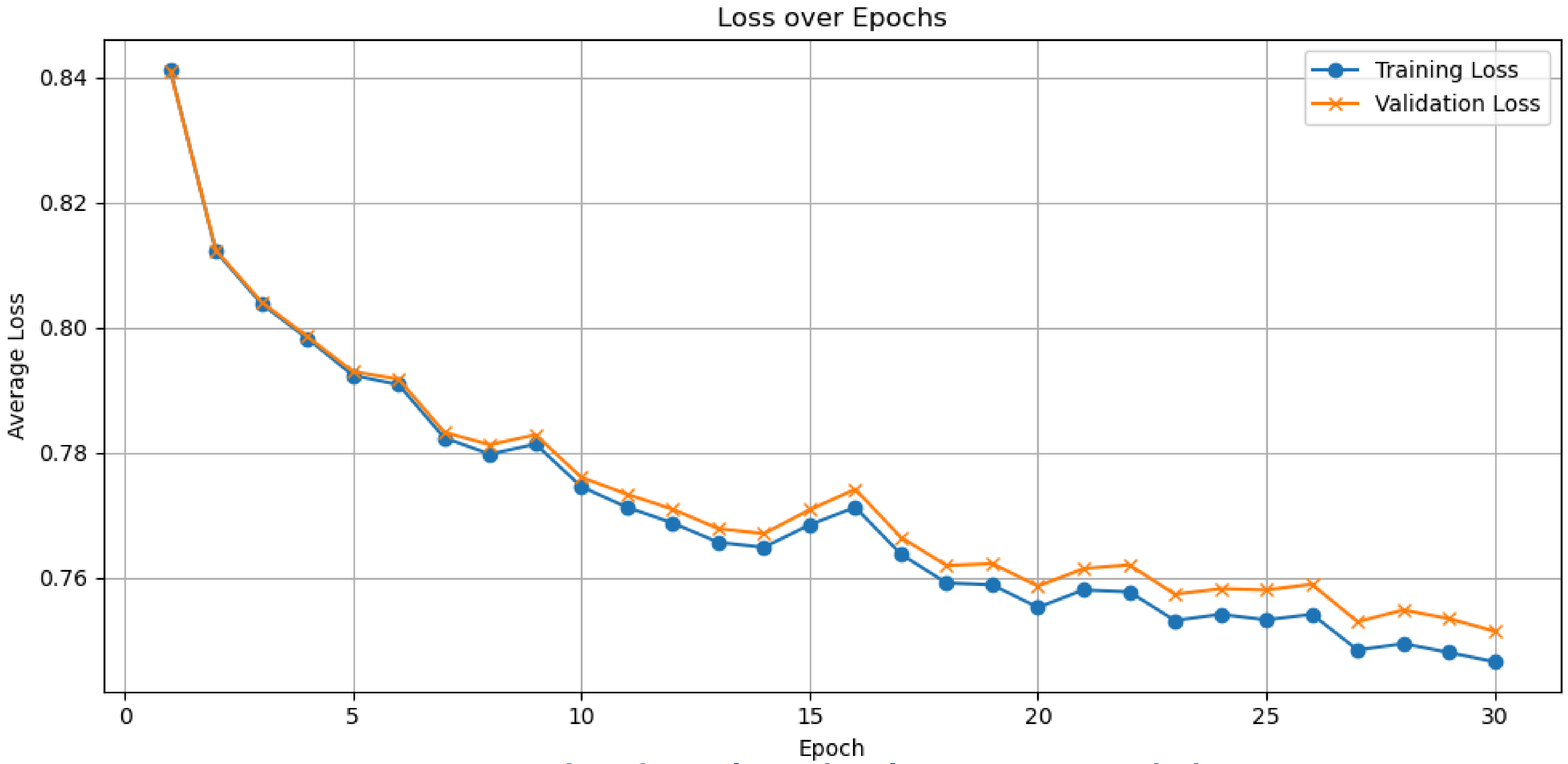
$(\alpha_{lca} = 1000, \; \alpha_{vtx} = 1)$

Setup

- **MC data of $B\bar{B}$ with $B_{sig} \to \nu\bar{\nu}$**
- **Partition for training, validation and test steps :**



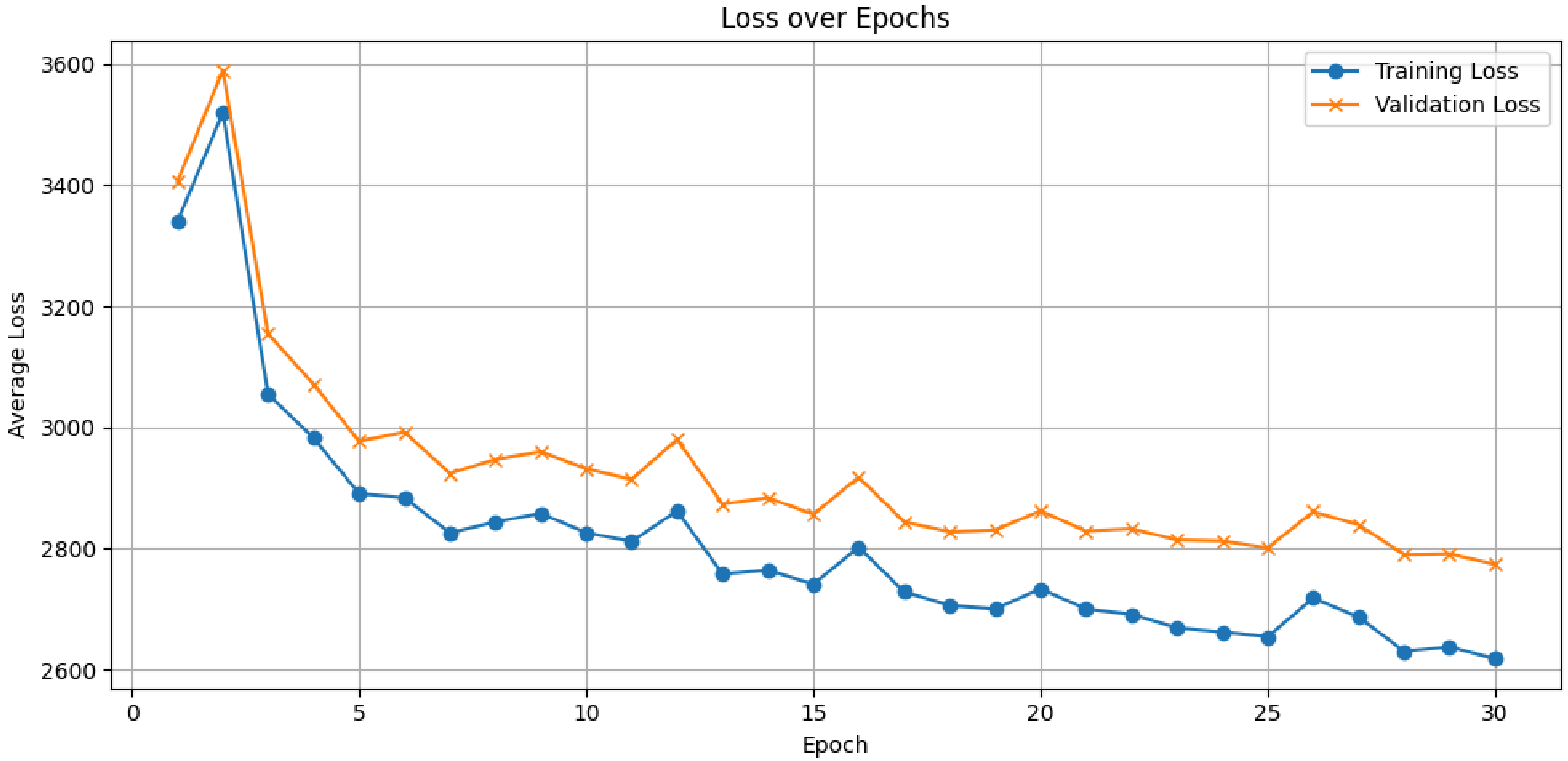
- **Loss function per epochs :**

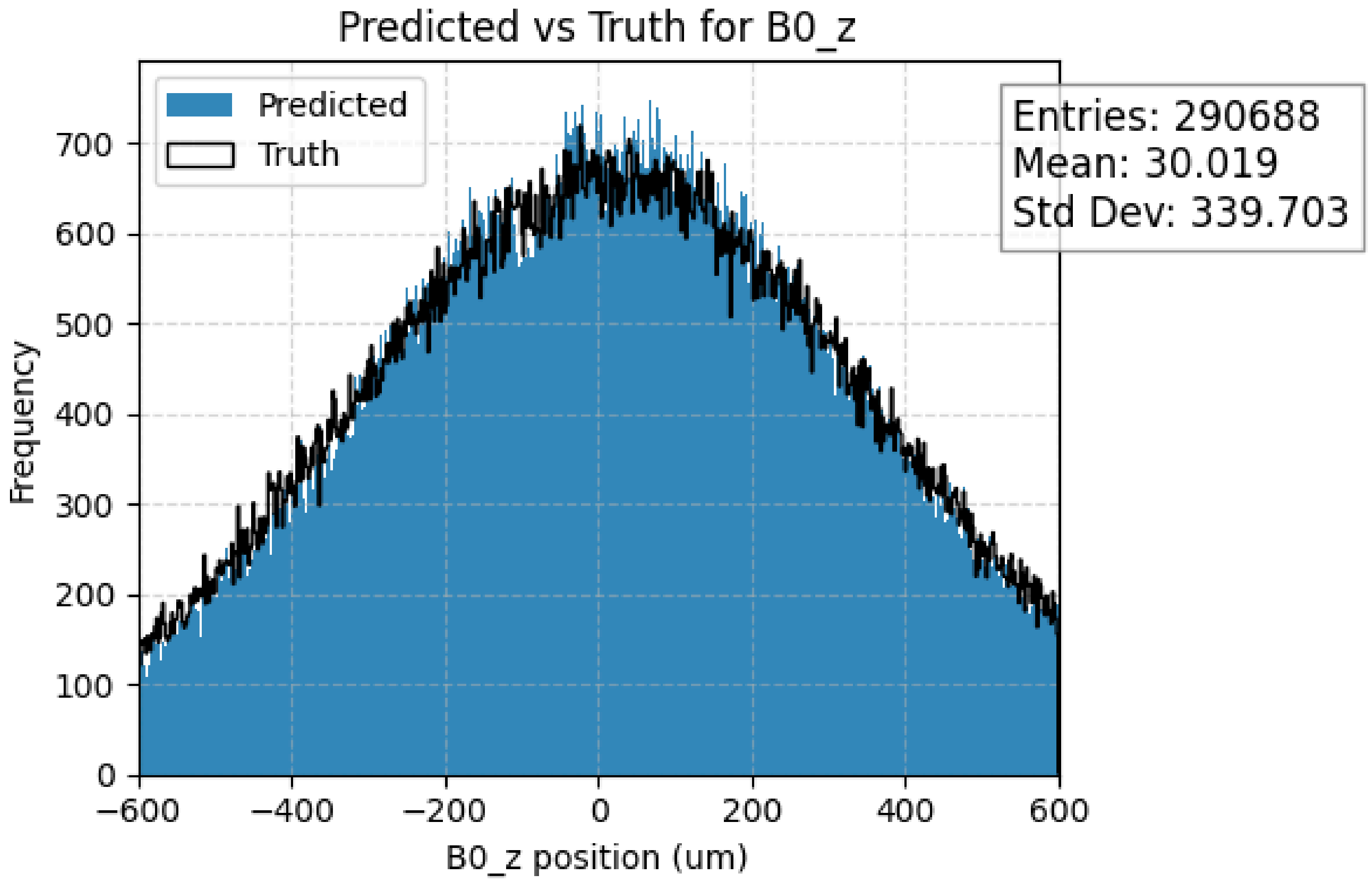


Loss per epochs for classical GraFEI training on LCA

- **Two Trainings :**
  - VTX Only $(\alpha_{lca} = 0,\ \alpha_{vtx} = 1)$
  - LCA and VTX $(\alpha_{lca} \neq 0,\ \alpha_{vtx} = 1)$
- **Objective :**
  - Get position of $B^0_{tag,z}$

VTX training





Loss per epochs for training on VTX position

$(\alpha_{lca} = 0,\ \alpha_{vtx} = 1)$

→ **Some hyperparameter to fix, but prediction ok**

VTX training

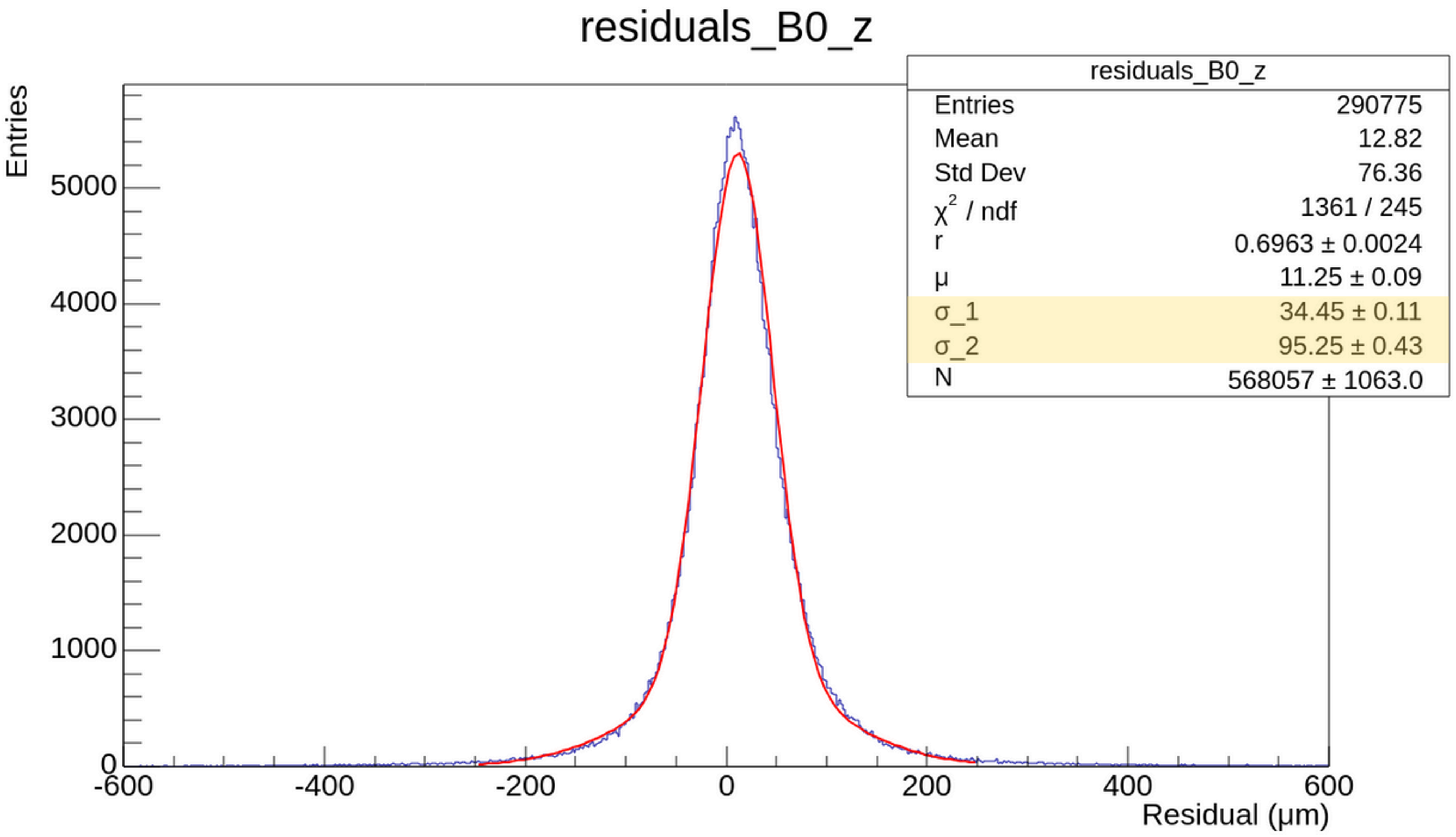
- **Residue:** $z_{pred} - z_{truth}$
- **Mixed Gaussian Fit:**

$$f\left(x\right) = rN\left(x\left|\mu_1, \sigma_1\right.\right) + (1-r)N\left(x\left|\mu_2, \sigma_2\right.\right)$$

$$\sigma = r\sigma_1 + (1-r)\sigma_2$$

$$\boxed{\sigma = 52.91 \pm 0.31\ \mu m}$$

although $\frac{\chi^2}{ndf} = 5.56\ldots$



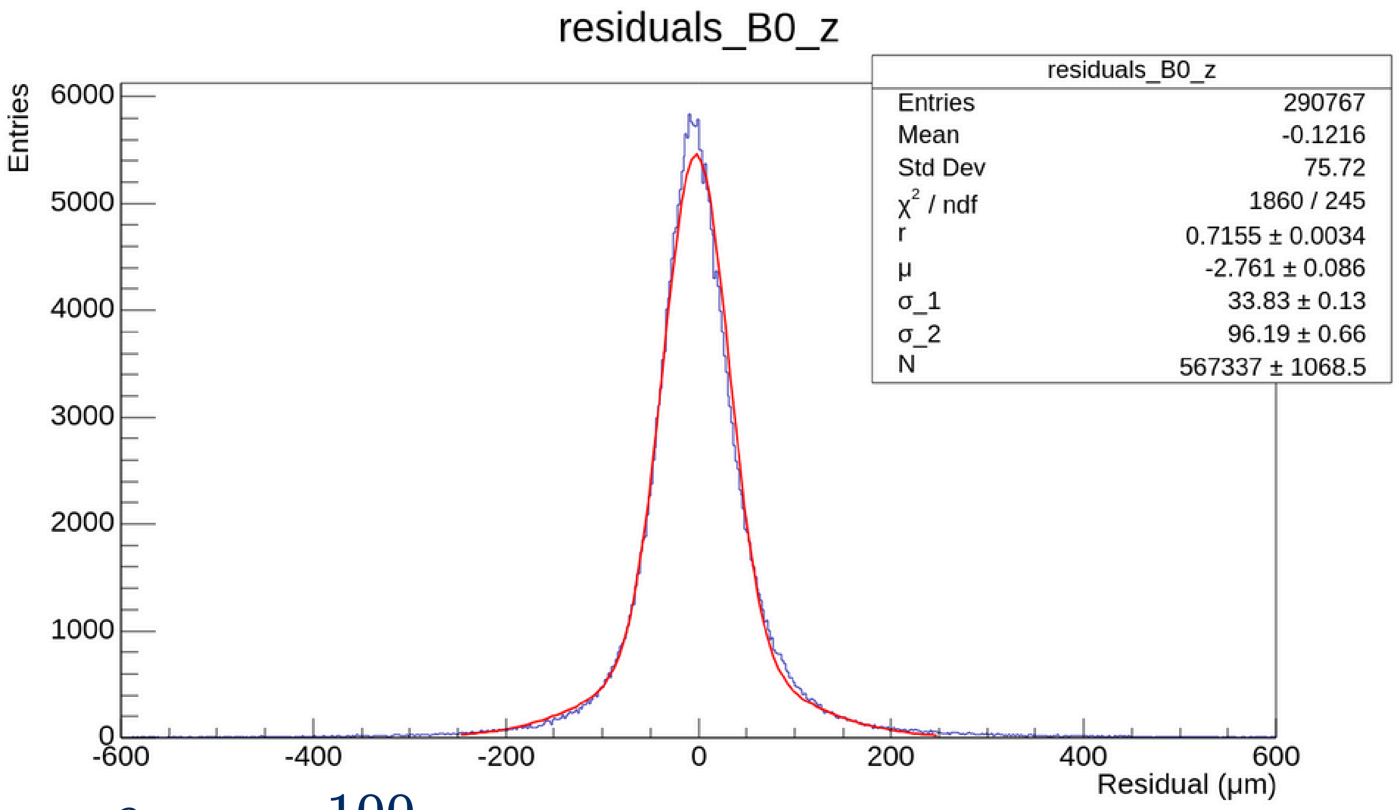
Fit with more than 95% of the distribution between [-250;250]

VTX + LCA Training

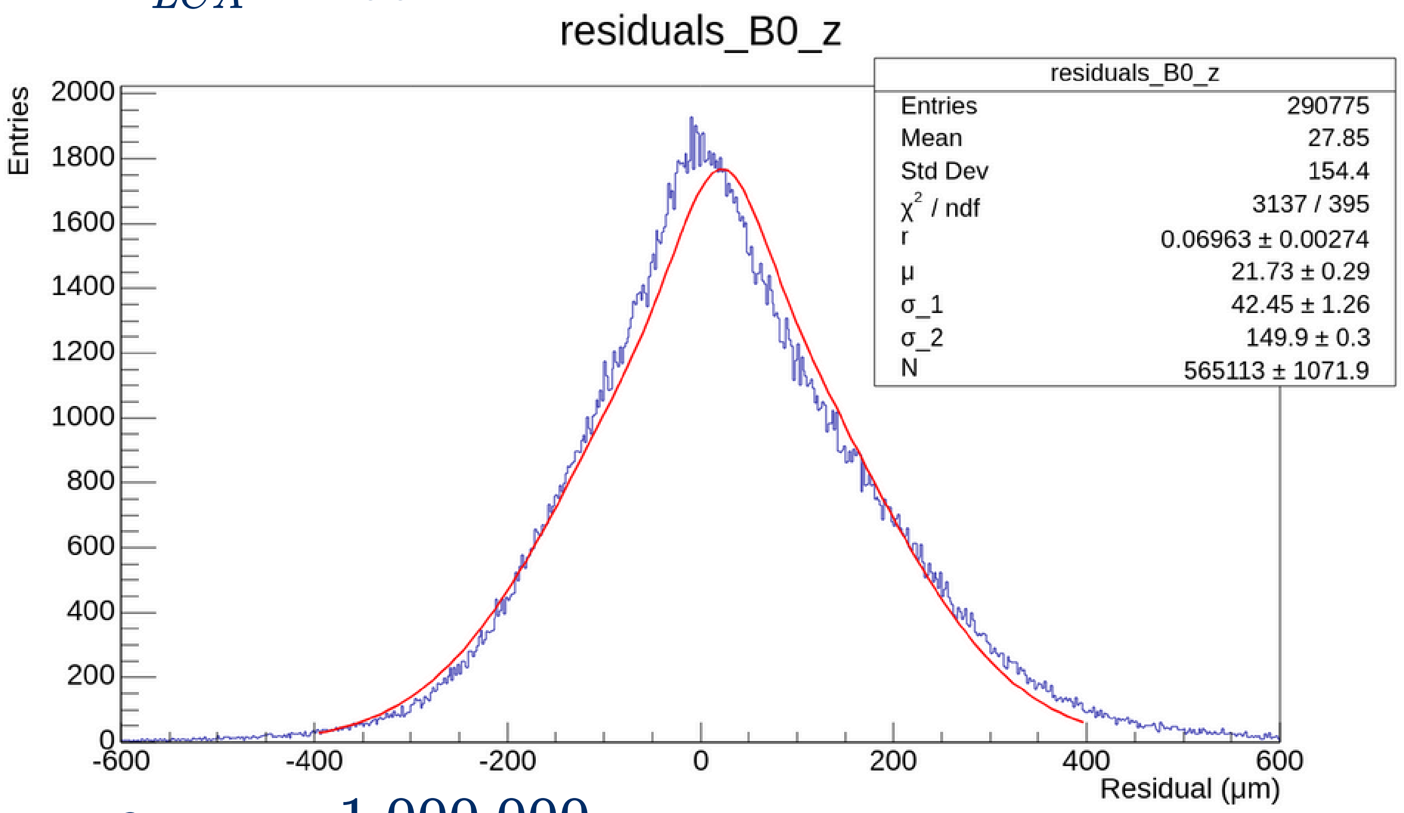
- **Coefficient tuning, now training on LCA too**
- **For different weight of LCA training:**

α_{LCA}	σ	$\frac{\chi^2}{ndf}$
0	52.91 ± 0.31	5.56
100	51.53 ± 0.29	7.59
1000	53.74 ± 0.33	9.44
10000	52.91 ± 0.31	11.56
1000000	142.42 ± 0.42	7.94

→ **No significant improvement**



$\alpha_{LCA} = 100$



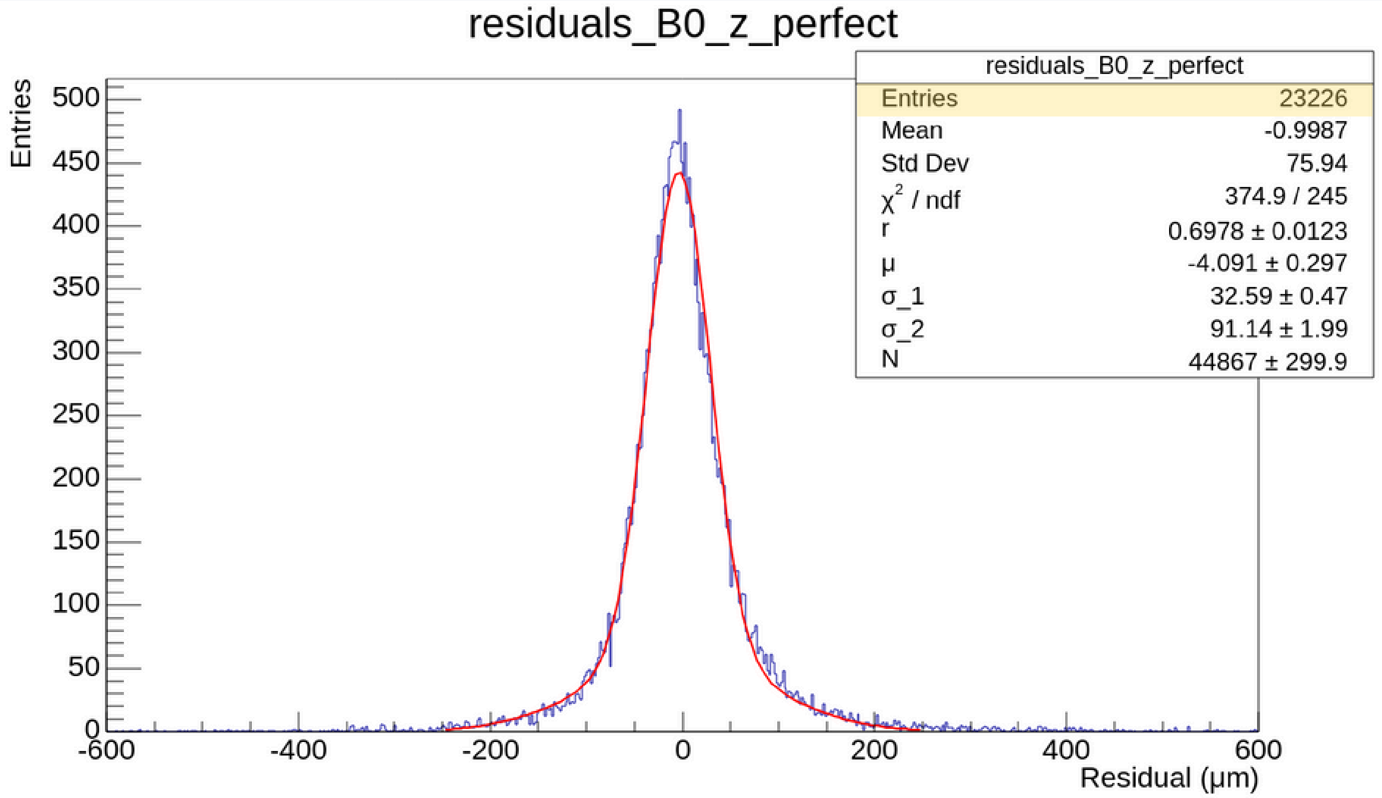
$\alpha_{LCA} = 1\,000\,000$

VTX + Perfect LCA Training

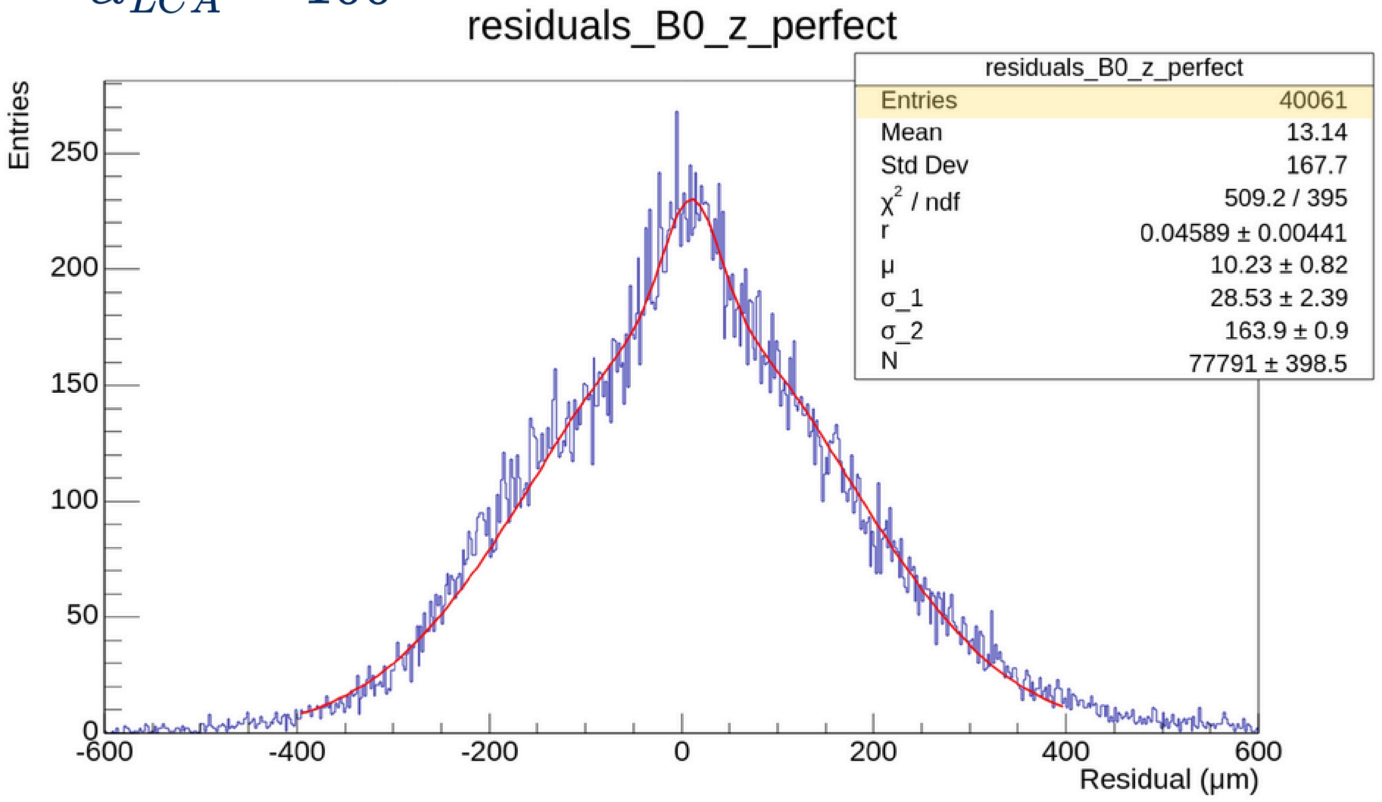
- **Extraction of perfectly reconstructed LCA events**
- **For different weight of LCA training:**

α_{LCA}	σ	$\frac{\chi^2}{ndf}$
100	50.05 ± 0.97	1.58
1000	51.13 ± 0.89	2.73
10000	51.66 ± 0.82	2.32
1000000	157.68 ± 1.05	1.29

→ **Increasing "perfect" events, better fit, but small dataset (<10%)**



$\alpha_{LCA} = 100$

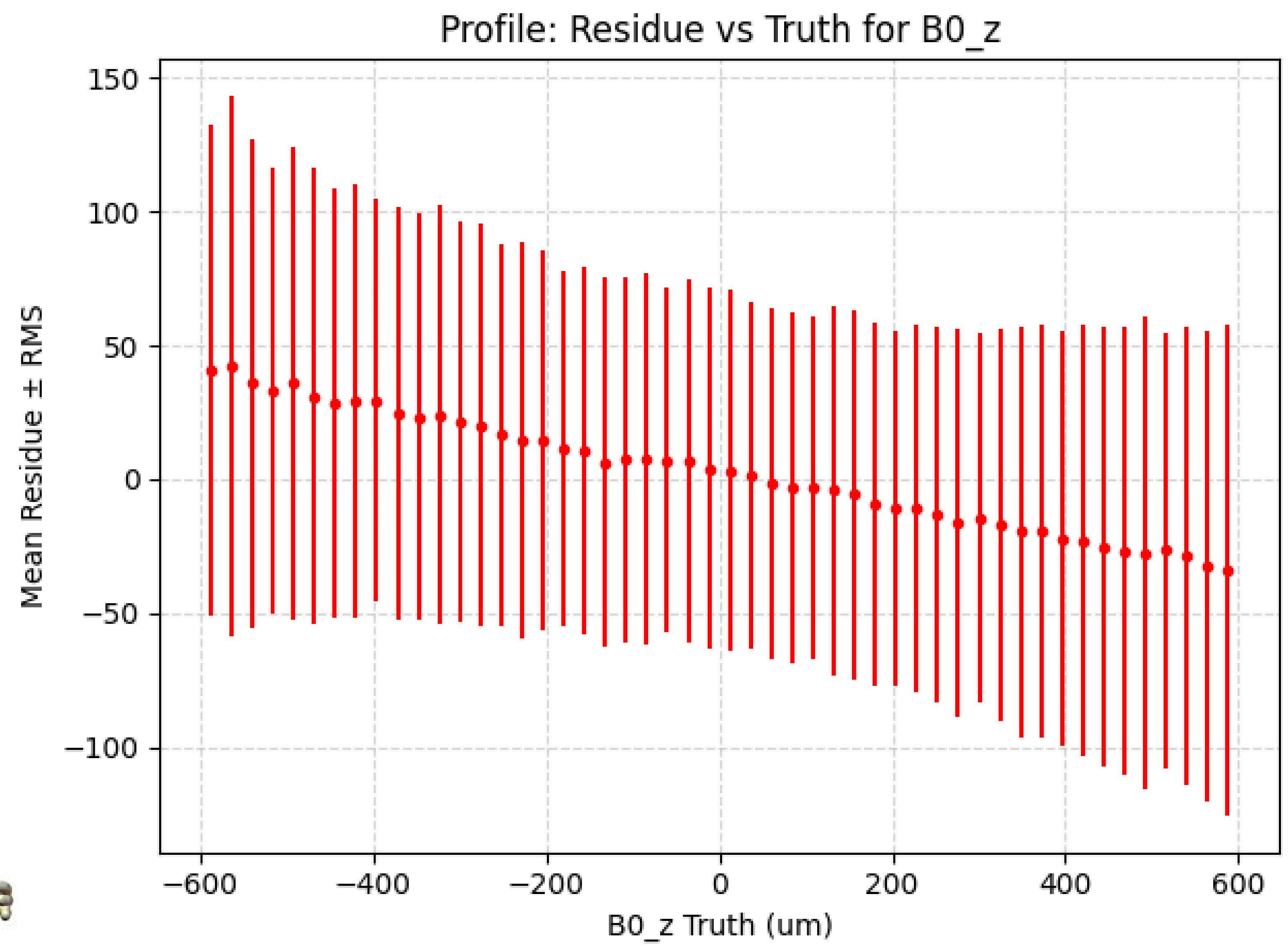


$\alpha_{LCA} = 1\,000\,000$

Biased model

- **Correlation observed between residues and truth !**
- **Deviation of predictions with z ?**

→ Need to investigate ! Maybe better resolution if fixed



Summary of results

For B tag z position:

TagV, Belle II vertex tagger

- **From literature:** $\sigma \approx 100\ \mu m$
- Evaluation requires reconstructible signal side decay

GraFEI's results

- **resolution of** $\sigma = 52.91 \pm 0.31\ \mu m$

→ **Improvement of almost 50% !**

→ Useful for precise **TDCPV measurement**

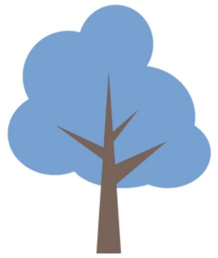
Time Dependent CP violation is modulated by resolution

$$[S \sin(\Delta m_d \Delta t) - C \cos(\Delta m_d \Delta t)] \times \sigma_{\Delta t}$$

Direct impact with z vertex resolution

$$\sigma_{\Delta t} = \frac{\sigma_{\Delta z}}{\gamma \beta}$$

Next steps

1. **Compare to TagV on new data** $B_{sig} \to \mu^+\mu^-$
2. **Hyperparameter optimization for VTX and LCA+VTX trainings**
3. **Extract new data for further analysis (nb particle, D mesons...)**
5. **Compare : current / ideal / upgraded geometry**
6. **Towards a Tree Fitter** 

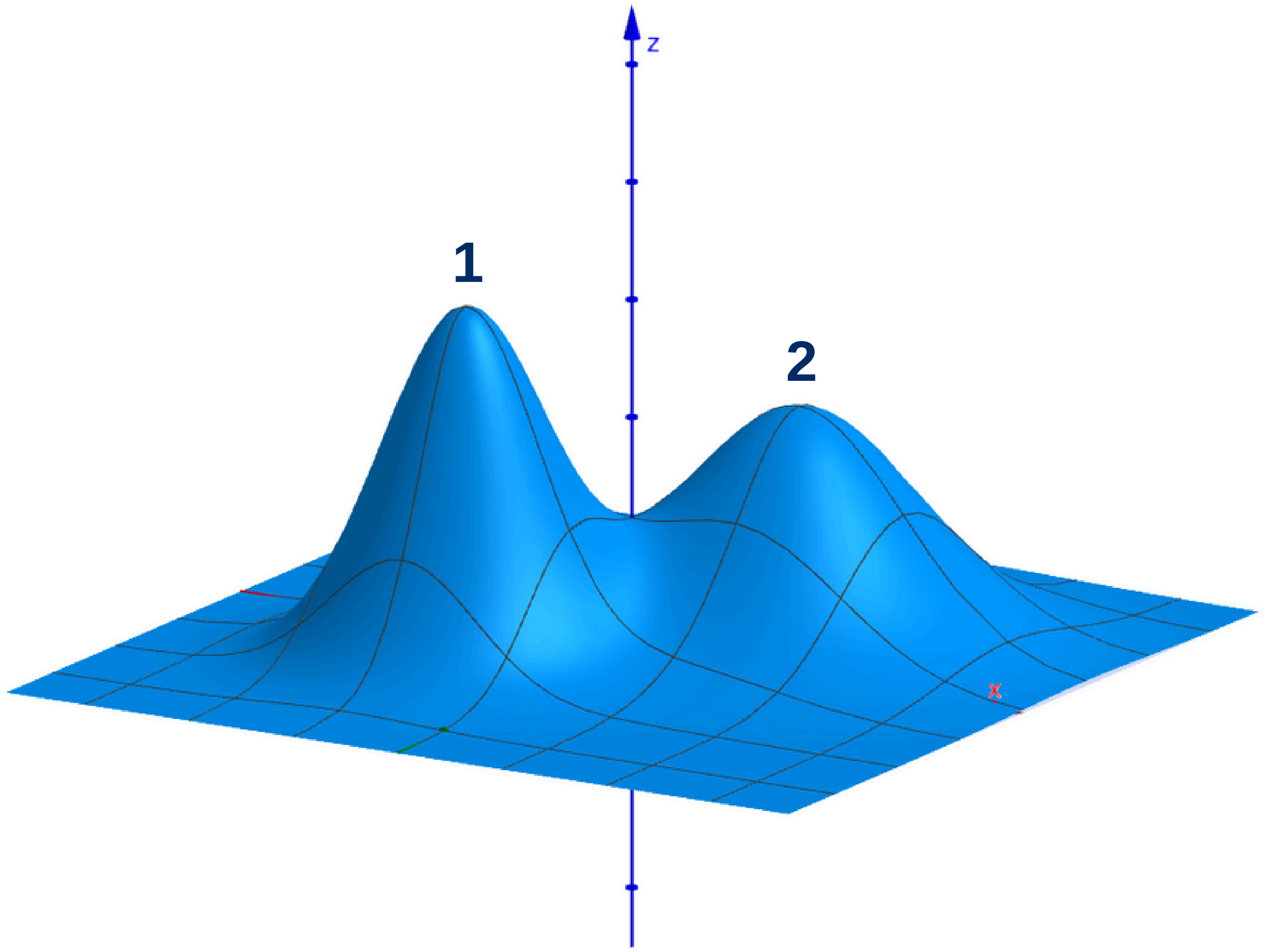
BACKUP

Mixed Gaussian Fit

Fit of multi-modal distributions or distributions with larger tails but single mode

$$f\left(x\right)=rN\left(x\left|\mu_1,\sigma_1\right.\right)+(1-r)N\left(x\left|\mu_2,\sigma_2\right.\right),\ with\ N\left(x\left|\mu,\sigma\right.\right)=\frac{1}{\sqrt{2\pi}\sigma}exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

- r **mixture weight**
- μ **mean**
- σ **standard deviation**



Propagating uncertainties for single mode:

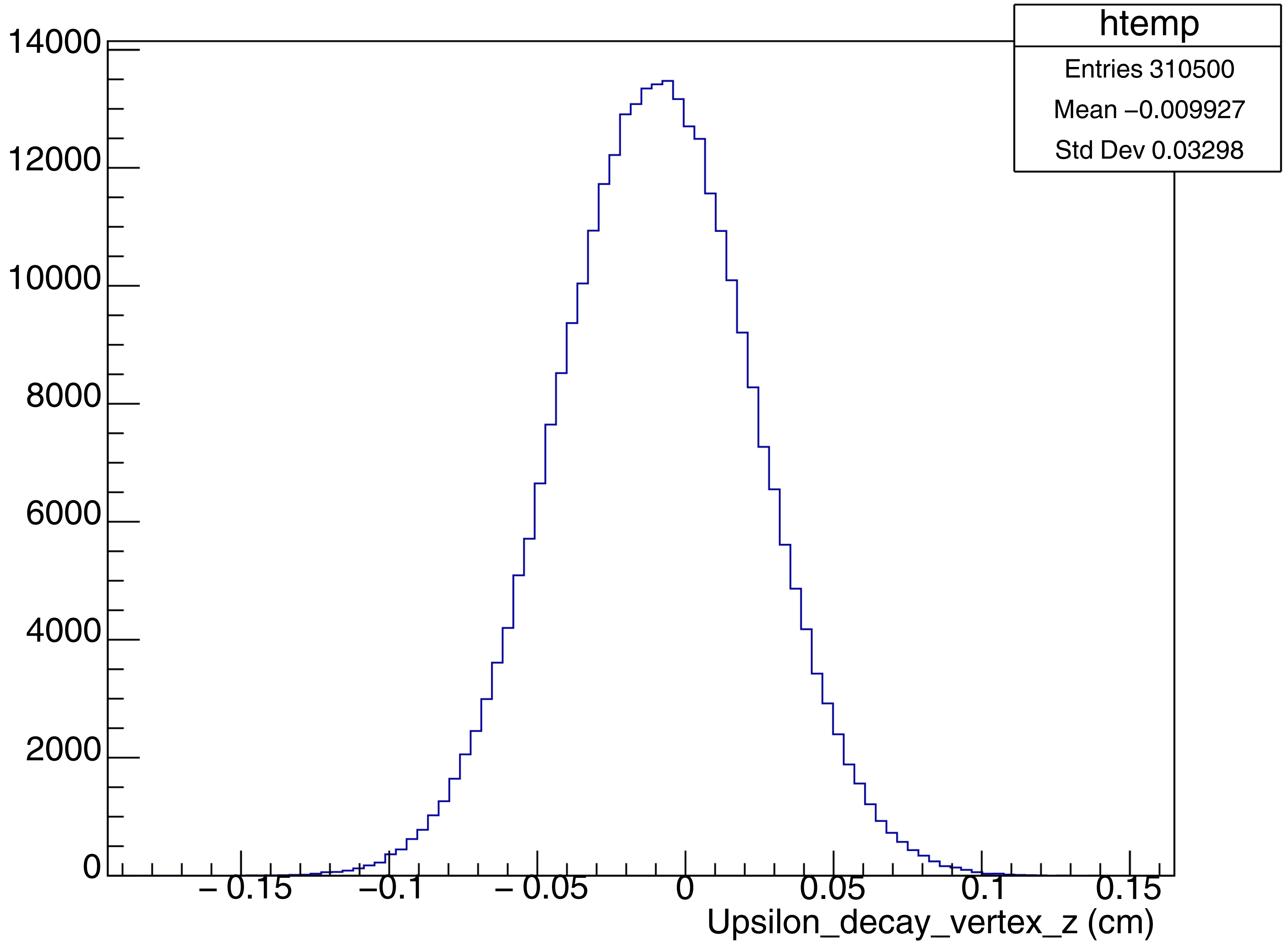
$$\sigma=r\sigma_1+(1-r)\sigma_2$$

$$\delta\sigma=\sqrt{\left(\frac{d\sigma}{dr}\delta r\right)^2+\left(\frac{d\sigma}{d\sigma_1}\delta\sigma_1\right)^2+\left(\frac{d\sigma}{d\sigma_2}\delta\sigma_2\right)^2}$$

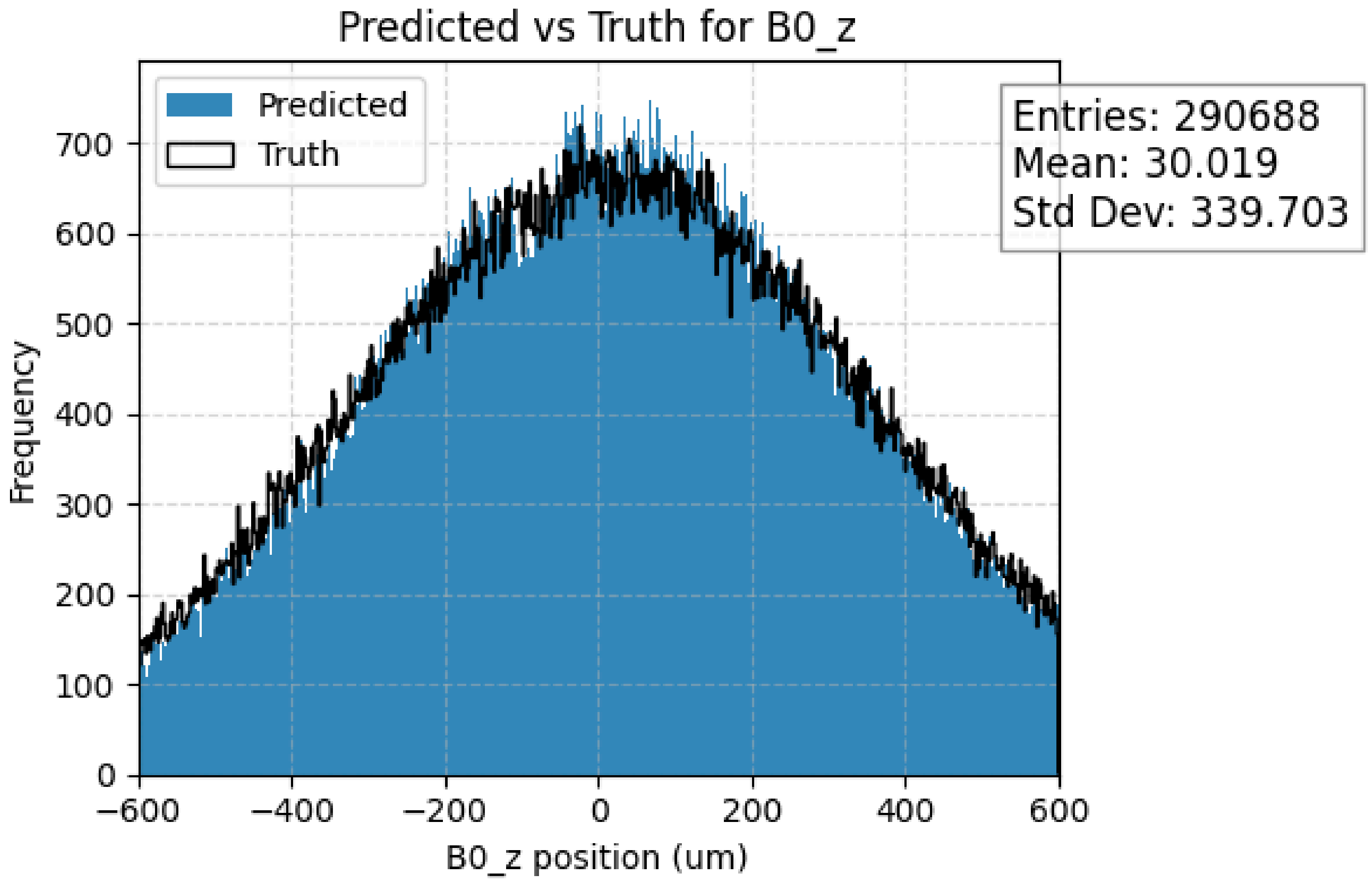
$d:$ **partial derivative**

$$\frac{d\sigma}{dr}=\sigma_1-\sigma_2,\ \frac{d\sigma}{d\sigma_1}=r,\ \frac{d\sigma}{d\sigma_2}=(1-r)$$

VTX position



Upsilon decay vertex along z



→ **PDG:** $c\tau = 455.4\ \mu m$, **after projection:** $z \approx 135\ \mu m$ **and (0,0,0) point = detector's (0,0,0)**

What did I do 

- **Implementation of the global feature** (understand code, setup environment, write scripts, find feature, implement feature, test everything) → see Gantt Chart
- **Change the loss function** (tuning with coefficients, addition of MSE loss, addition of metrics)
- **Change the hyperparameters** (Learning rate)
- **Do the training for different values of coefficients** (get figures, debug... a lot of debug...)
- **Found limits to the existing code!** (Open merge request on Basf2)

Internship timeline

Gantt chart

Task \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	07/03	10/03	17/03	24/03	31/03	07/04	14/04	21/04	28/04	05/05	12/05	19/05	26/05	02/06	09/06	16/06
Bibliography																
Discovering framework																
Implementing new feature																
Use of large datasets																
Optimal training conditions																
Working code																
Trying TagV																
Results analysis																
Presentation & new mdst																

Belle II France Presentation

M2 defense !

Pipeline

Simplified model

create training files script

- Creates Ntuples with all requested data (Tree)
- Creation of new variable/branch "b_decay_x/y/z"

→ Fetches vertex if PDG == 511 AND NOT Bnunu

training script

- Loads model configuration
- Loads root ntuple and creates graph objects
- Splits datasets
- Updates features / computes loss & metrics

→ Global layer has new feature b_decay_x/y/z to learn/predict

Test script

- Loads weights and config of last epoch
- Predicts features on test dataset
- Returns analysis data for figures