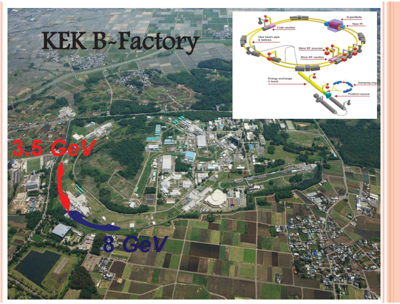
The neural z-Vertex Trigger for the Belle II Detector

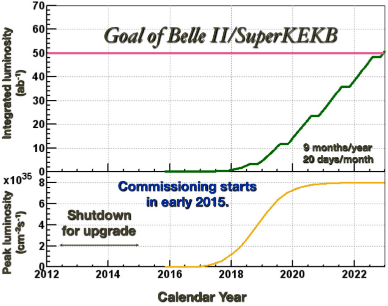
Fernando Abudinén

17. März 2014

$\Delta p \cdot \Delta q \geq \frac{1}{2}\hbar$

Upgrade of KEKB to SuperKEKB (Tsukuba)





$e^+ \rightarrow \Upsilon(4S) \leftarrow e^-$

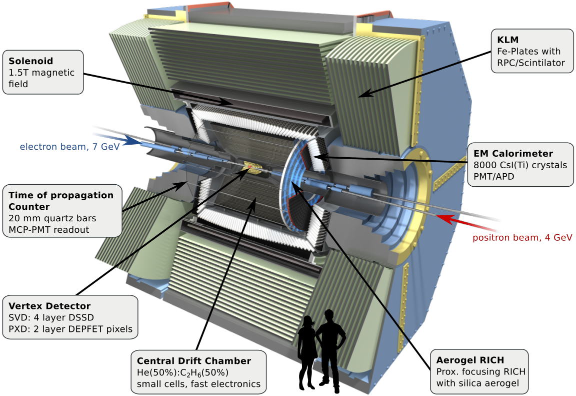
$B\bar{B} \;\Rightarrow\; \sqrt{s} = 10.58$ GeV

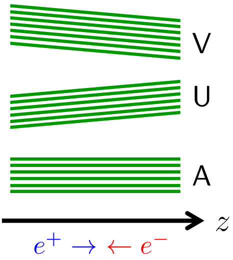
- Instantaneous luminosity of $L = 0.8 \cdot 10^{36}$ cm^{-2}s^{-1}

$\Rightarrow$ 40 times higher than the world record reached by KEKB.

$\Rightarrow$ 50 times larger data sample

The Belle II Detector





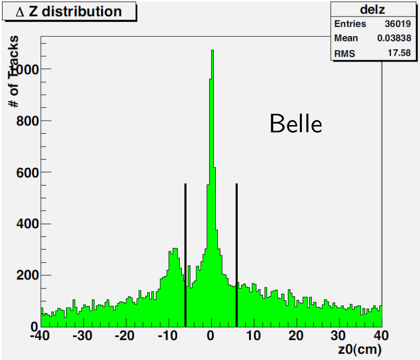
- CDC: Provides track information for the trigger.

$\Rightarrow$ 15000 sense wires in 9 superlayers

$\Rightarrow$ Config. AUAVAUAVA "A" $\hat{=}$ Axial, "U" and "V" $\hat{=}$ stereo

Why a z-Vertex Trigger?

- Undesired scattering processes (Touschek, Beam-Gas scattering, etc) $\Rightarrow$ **background**
- Background events not from collision point ($z = 0$)
- Higher luminosity $\Rightarrow$ higher background (factor $\sim$ **30**)
- Data prod. rate $\gg$ transfer + record capacity $\Rightarrow$ L1 Trigger



- Filter out events with vertex ($z_0 \neq 0$)

$\Rightarrow$ Goal: High resolution z-vertex trigger ($\sigma \leq 2$ cm)

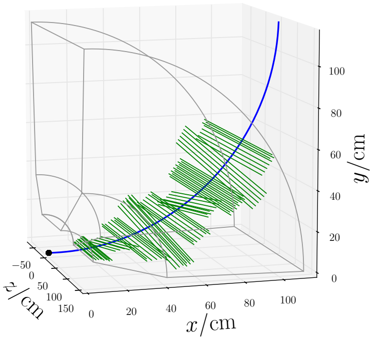
$\Rightarrow$ Cut at $\pm 3\sigma$

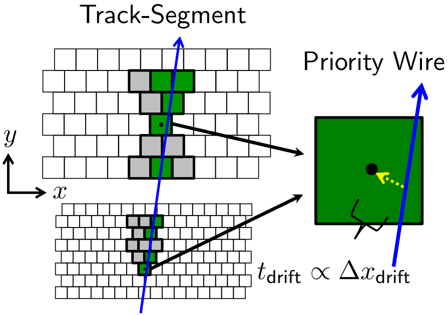
- Trigger latency ($\sim 5\ \mu s$)

Why a z-Vertex Trigger?

CDC Tracking:

- 15000 wires $\Rightarrow$ 2336 track-segments (TS)





Which information is available for triggering?

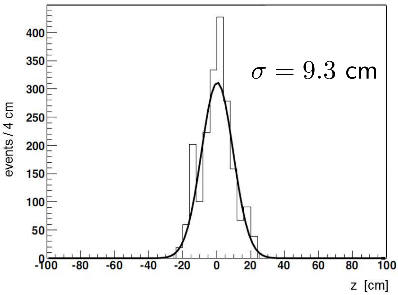
a) Identification numbers of active track-segments (TS-IDs)
$\Rightarrow$ Position of priority wire

b) Drift times of priority wires

Why a z-Vertex Trigger?

- Belle II: Present z-vertex trigger uses **only TS-IDs** (Hough transformation)
- No time for full track reconstruction !!!

Belle II 3D CDC Trigger



- Significantly better results expected from neural networks (MLP) using **drift times**

$\Rightarrow$ Parallelism of MLP computations suitable for L1 trigger

Multi Layer Perceptron

a) TS $\hat{=}$ Input-Neurons

b) Drift times $\hat{=}$ t_k Input values

$\Rightarrow$ Hidden layer: $n_{\text{hidden}} = 3 \cdot n_{\text{input}}$

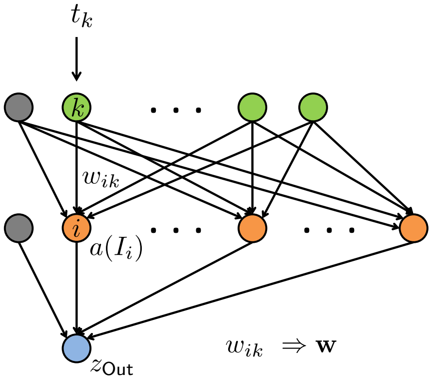
$\Rightarrow$ Input: $I_i = \sum_{k=0}^{n_{\text{input}}} w_{ik} t_k$

$\Rightarrow$ Output: $a_i = \tanh(I_i)$

$\Rightarrow$ Output neuron:

$z_{\text{Out}} = a(\sum_{i=0}^{n_{\text{hidden}}} w_{\text{Out},i} \cdot a(I_i))$

- Real vertex from simulation: z_{True}



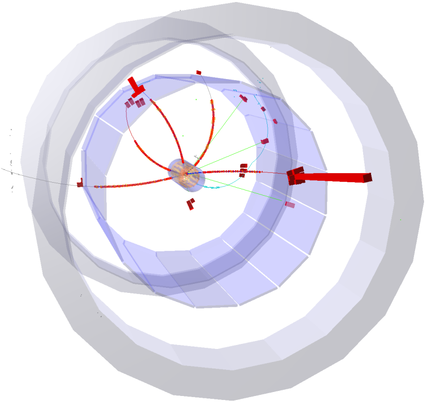
$w_{ik} \Rightarrow \mathbf{w}$

Training:

- Modify $\mathbf{w} \Rightarrow z_{\text{Out}} \rightarrow z_{\text{True}}$
- Cost function (MSE):

$$E(\mathbf{w}) \equiv \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left(z^j_{\text{True}} - z^j_{\text{Out}} \right)^2$$

Method



- Decompose events in single tracks $\Rightarrow$

CDC Phase Space:

- $\phi \in [0°, 360°]$
- $\theta \in [17°, 150°]$
- $p_\mathsf{T} \in [0.2, 5.2]$ GeV$/c$
  $p_\mathsf{T} \propto \kappa^{-1}$

- Whole CDC phase space too much input for a single MLP

$\Rightarrow$ Divide it in sectors:
$\Delta\phi \sim 1°$, $\Delta\theta \sim 6°$
$\Delta p_\mathsf{T} \sim 0.05\text{-}0.6$ GeV$/c$

$\Rightarrow$ $\sim 2 \cdot 10^6$ sectors

- Find the sectors which are firing!

Method

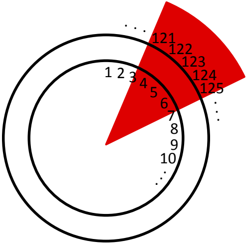
Train a MLP for each specific sector!

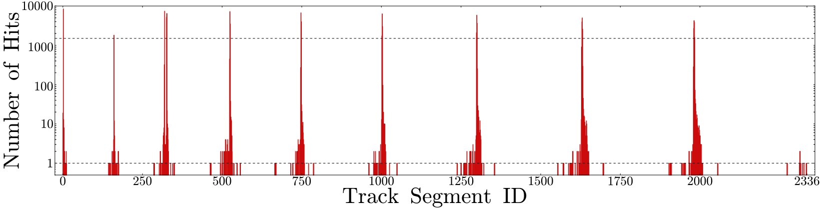
- Illuminate uniformly each sector

$\Rightarrow$ Look for active TS

- Select TS which are active in more than 15% of the events

$\Rightarrow$ 16-28 Input neurons





$\Rightarrow$ Drift times of selected TS $\Rightarrow$ Input values for MLP.

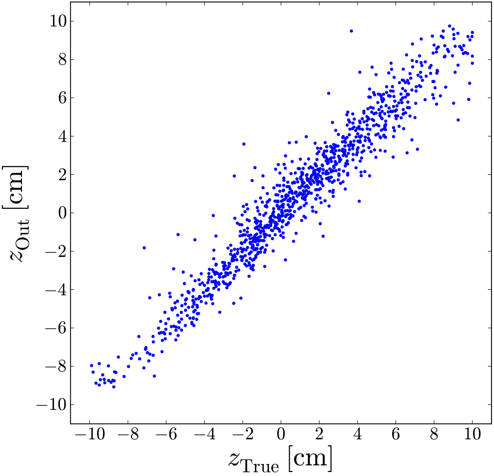
MLP Test

Test of the MLP after training:

⇒ Compare MLP output with true value! (simulated tracks)

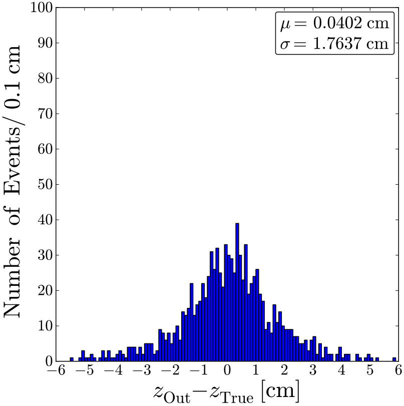
- To evaluate the performance consider:

$$\mu = \text{mean}(z_{\text{Out}} - z_{\text{True}})$$
$$\sigma = \text{std}(z_{\text{Out}} - z_{\text{True}})$$

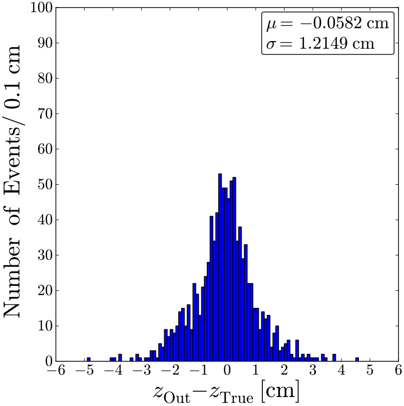


Results without Background

$p_{\mathrm{T}} \in [5.0, 5.2]$ GeV/$c \Rightarrow$ vary θ



(a) $\theta \in [30°, 36°]$

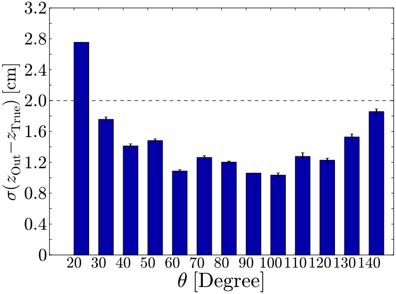


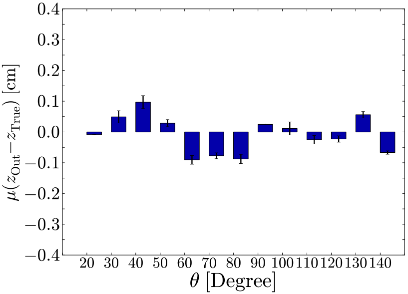
(b) $\theta \in [80°, 86°]$

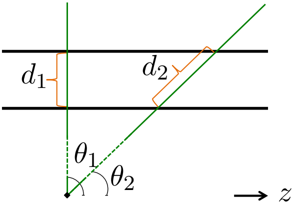
Results without Background

Network resolution as a function of polar angle θ:

$\Delta\theta = 6^\circ$

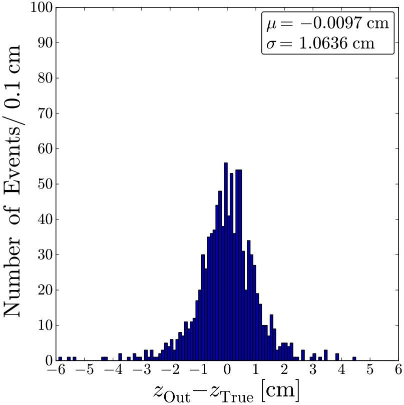




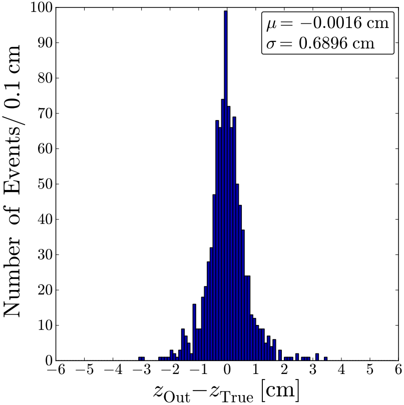


Results without Background

$\theta \in [56°, 62°] \;\Rightarrow\;$ vary p_T



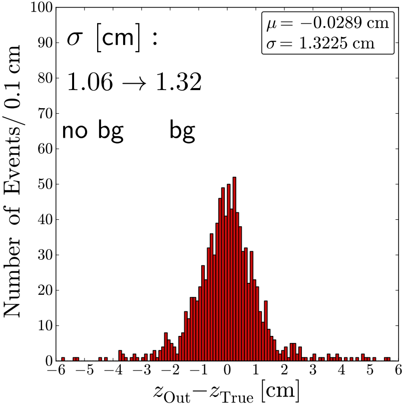
(a) $p_T \in [0.50, 0.55]$ GeV/c



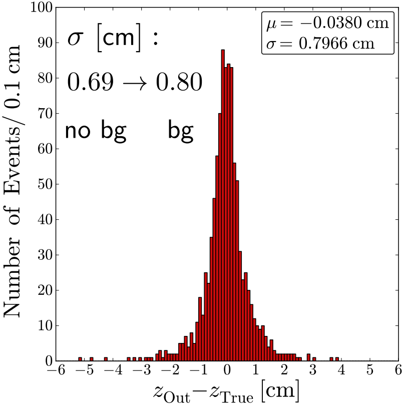
(b) $p_T \in [3.20, 3.80]$ GeV/c

Results with Background

- Mix background hit patterns into the single tracks events (Touschek, Coulomb and Radiative Bhabha).

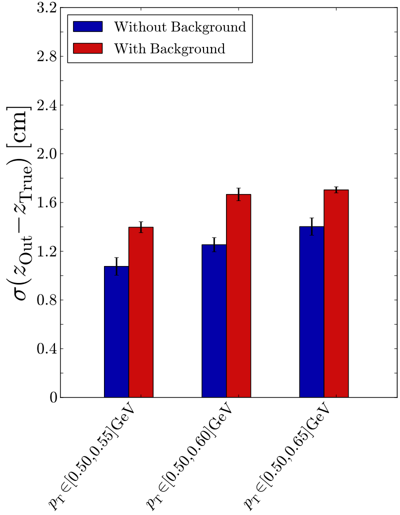


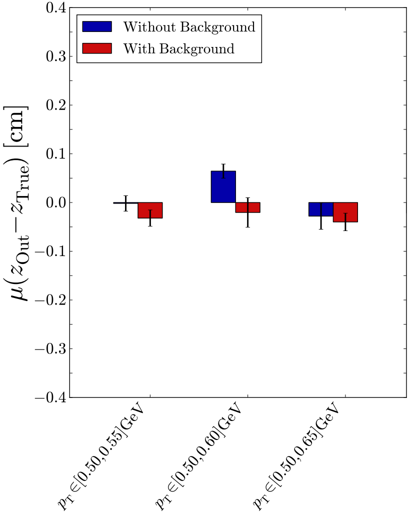
(a) $p_{\mathrm{T}} \in [0.50, 0.55]$ GeV/c



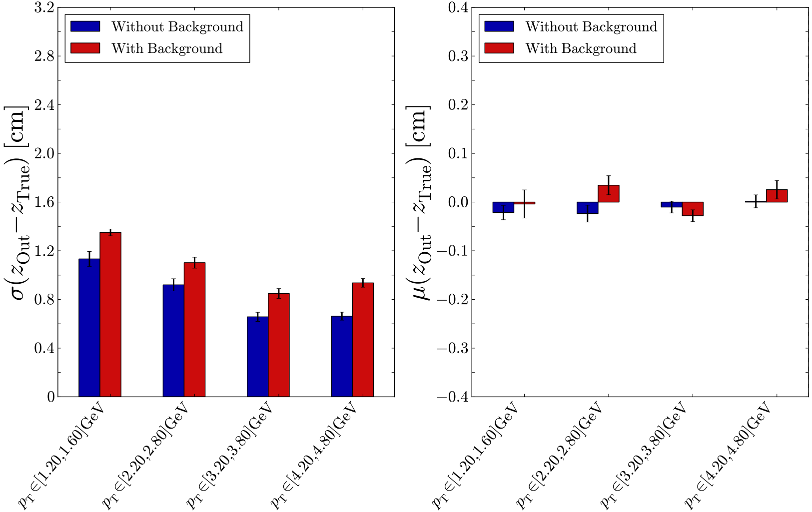
(b) $p_{\mathrm{T}} \in [3.20, 3.80]$ GeV/c

Results with Background





Results with Background



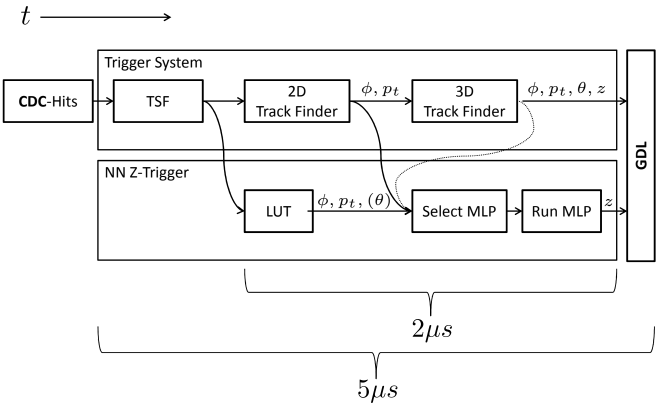
Conclusions and Outlook

- The MLP reaches a significantly better resolution than the present Belle II z-vertex trigger. ☺
- Background leads to an average resolution loss of $\sim 25\%$

$\Rightarrow$ With and without background: Resolution significantly better than required (< 2 cm). ☺

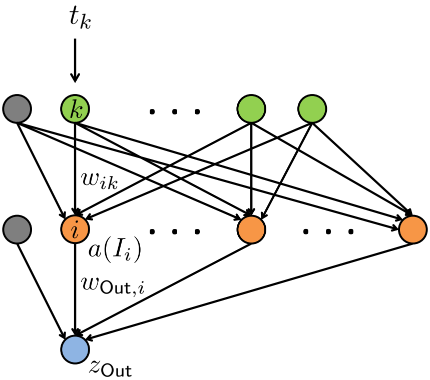
- Parallelism inherent to the computations makes the MLP suitable for L1 trigger $\Rightarrow$ realizable in FPGA. ☺
- The number of required MLPs ($\sim 2 \cdot 10^6 \hat{=} 10$ Gb) is a challenge for the hardware implementation.
- Next step: Decomposition of a specific event in separate tracks? (Look-Up-Table, 2D Trigger of Belle II?)

The Whole Trigger



Multi Layer Perceptron

How does it work?



- Input I_{Out} for output neuron:
  $I_{\text{Out}} = \sum_{i=0}^{n_{\text{hidden}}} w_{\text{Out},i} \cdot a(I_i)$

- CDC-Track Segments (TS) $\hat{=}$ Input-Neurons (Input Layer)
- Input values t_k $\hat{=}$ Drift times
- Neurons in the middle layer $n_{\text{hidden}} = 3 * n_{\text{input}}$ (Hidden Layer)
- Connection weights w_{ik}
- Activation function $a_i = \tanh(I_i) \in [-1, 1]$
- Input I_i for a neuron i:
  $I_i = \sum_{k=0}^{n_{\text{input}}} w_{ik} t_k$
- Output of the output neuron:
  $z_{\text{Out}} = a(I_{\text{out}}) \in [-1, 1]$

Multi Layer Perceptron

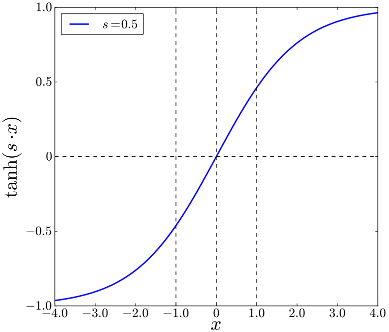
- From each simulated event: real vertex position on the z axis: z_{True} $\Rightarrow$ needs only to be scaled to $[-1, 1]$.
- Training means to iteratively modify all weights $\mathbf{w}$, in order for z_{Out} to converge to z_{True} using a training algorithm (iRPROP^{-}).
- The iRPROP^{-} algorithm evaluates at each iteration (training) step the Mean Squared Error function:

$$E(\mathbf{w}) \equiv \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left(z^{j}_{\text{True}} - z^{j}_{\text{Out}} \right)^2$$

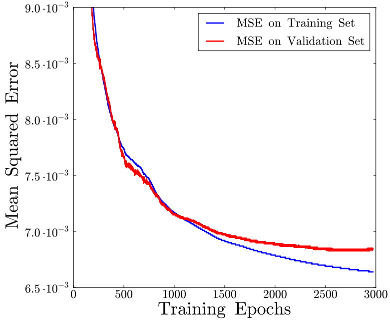
- Training Sample $N_{\text{train}} = 20000$ events

$$\Delta \mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}} \Rightarrow \mathbf{w}_{n+1} = \mathbf{w}_n + \Delta \mathbf{w}$$

- $n^{\text{weights}}_{\text{total}} = f_{\text{hidden}} \left(n^2_{\text{input}} + 2n_{\text{input}} \right) + 1 \;\sim 900 - 3000$

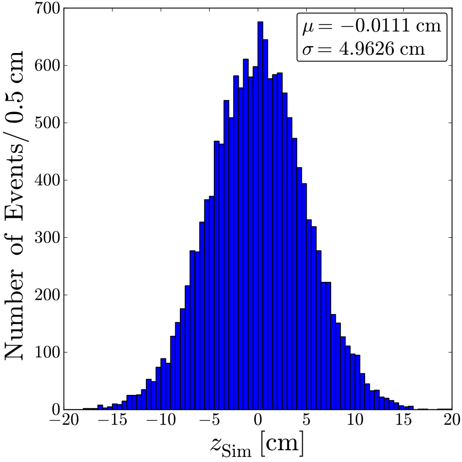


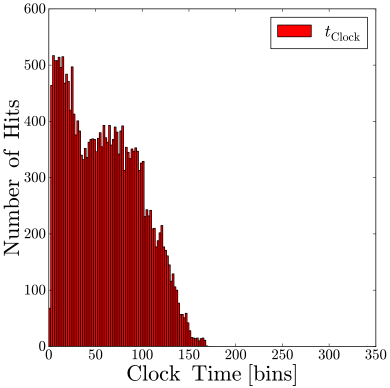
(a) Tangens Hyperbolicus



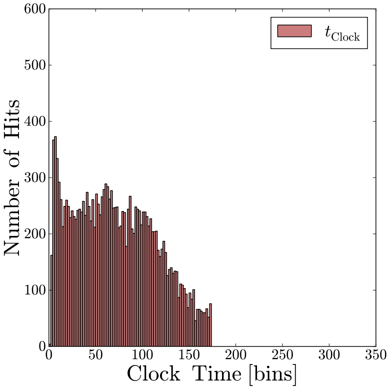
(b) Training Process

z-Vertex Distribution for all Experiments



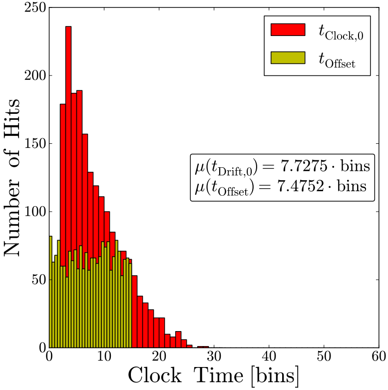


(a) No Background

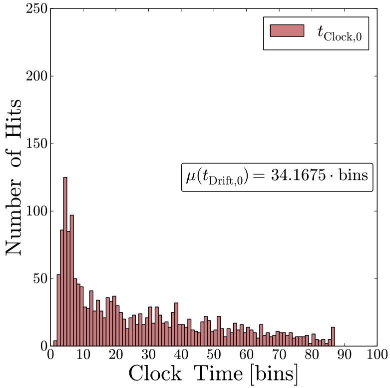


(b) Pure Background

First Hit Time Distr. with TSF Resolution



(a) No Background



(b) Pure Background

- Main Achievement: Measurement of time dependent violation of CP symmetry in the B-Meson system

$\Rightarrow$ Proof of CKM-theory (SM) $\Rightarrow$ Nobel Prize 2008