# LED Timing Calibration Update

**Goal**: using LED data to determine per-pixel/channel timing offsets to be put into CCDB

**This study** (a closure test): in simulation, inject some offsets by hand and see if (and how well) we can recover the injected offsets

Yunjie Yang, 1/22/19

# PMT Hit Time in MC and data

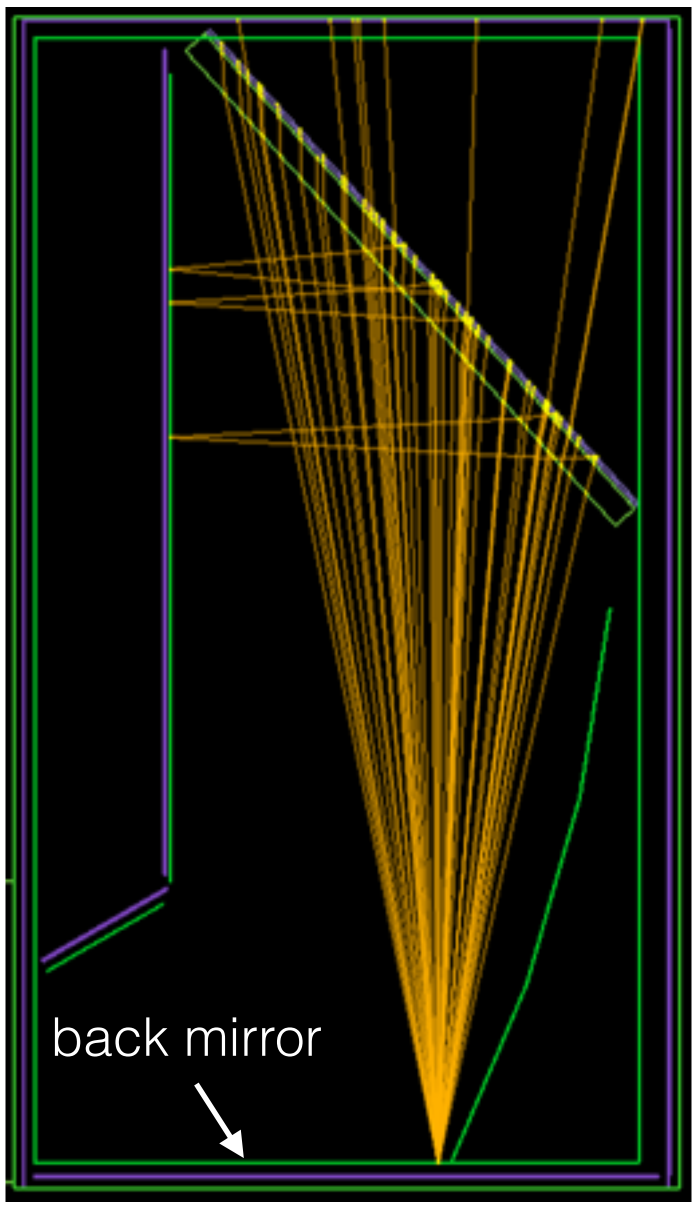
In MC:

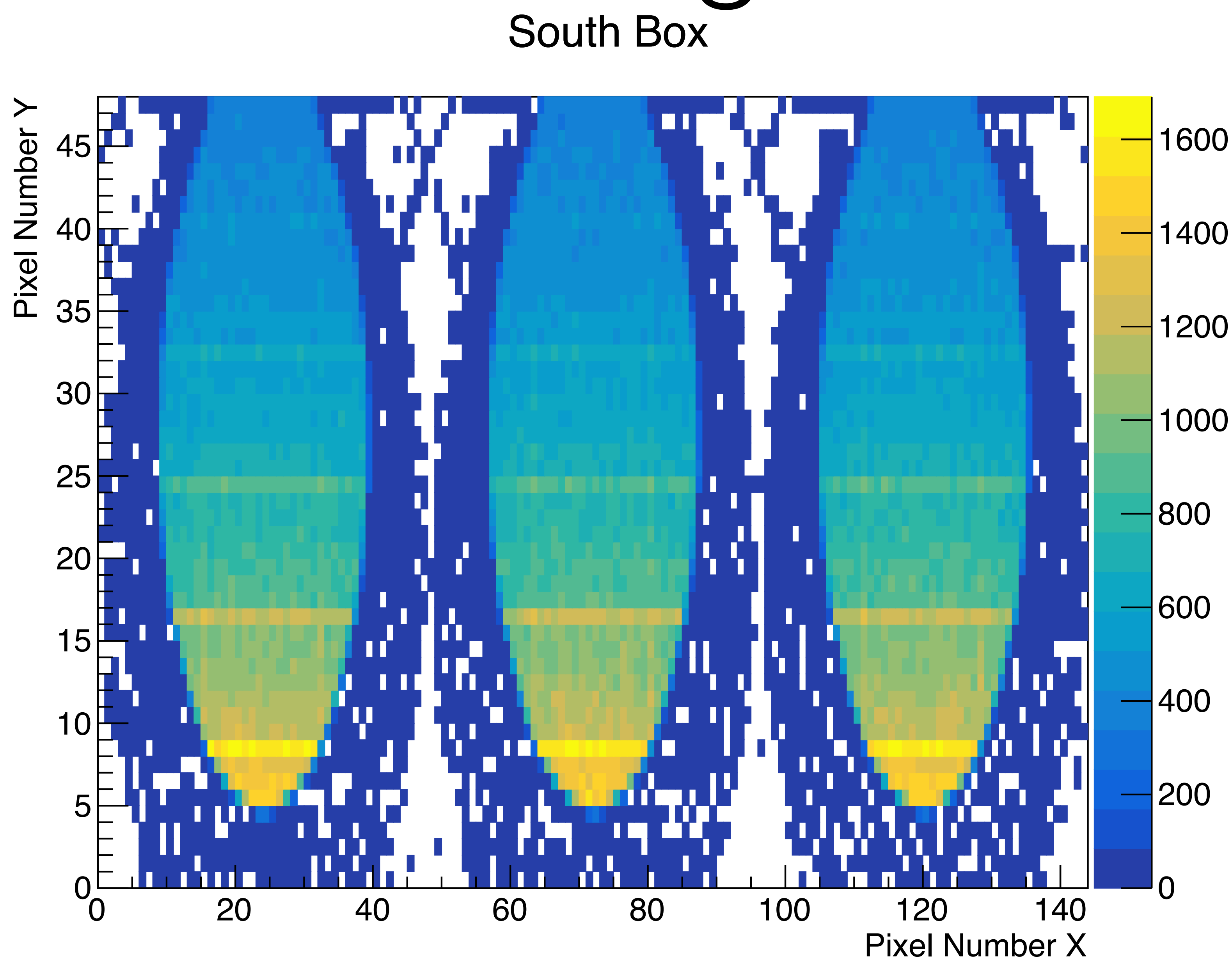
$$t = \underbrace{t_{\text{path}} + t_{\text{pulse shape}} + t_{\text{feedthrough delay}} + t_{\text{time reso.}}} + t_{\text{offset,i}}$$

In data (LED):

$$t = t_{\text{DDIRCTDCDigiHit}} + t_{\text{offset,i}} + t_{\text{base}}$$

# LED source setting

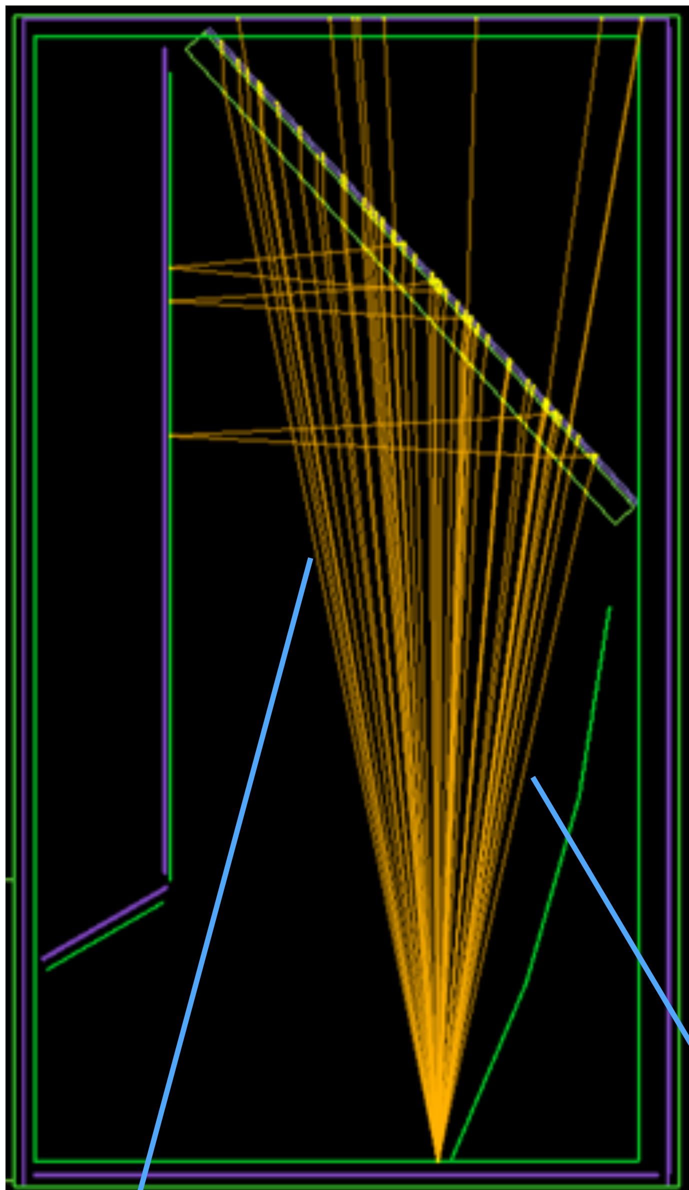


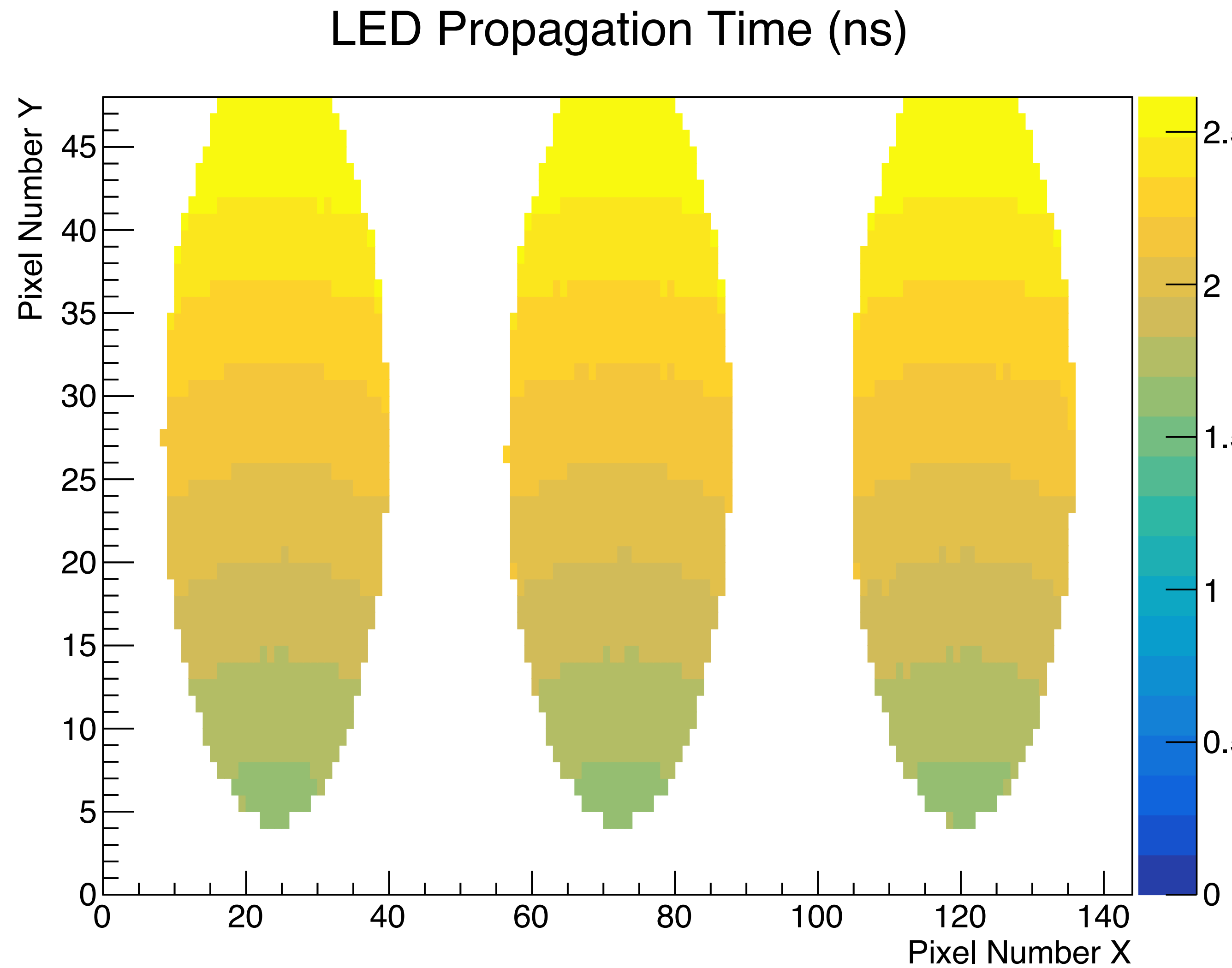




opening angle: 25 degree
inclination angle: 0 degree
z-position: at the reflective surface of the back mirror

# $t_{\text{path}}$

$$t = \boxed{t_{\text{path}}} + t_{\text{pulse shape}} + t_{\text{feedthrough delay}} + t_{\text{time reso.}} + t_{\text{offset,i}}$$






longest: ~57 cm

shortest: ~35 cm

n ~ 1.34

v in water: ~22.4 cm/ns

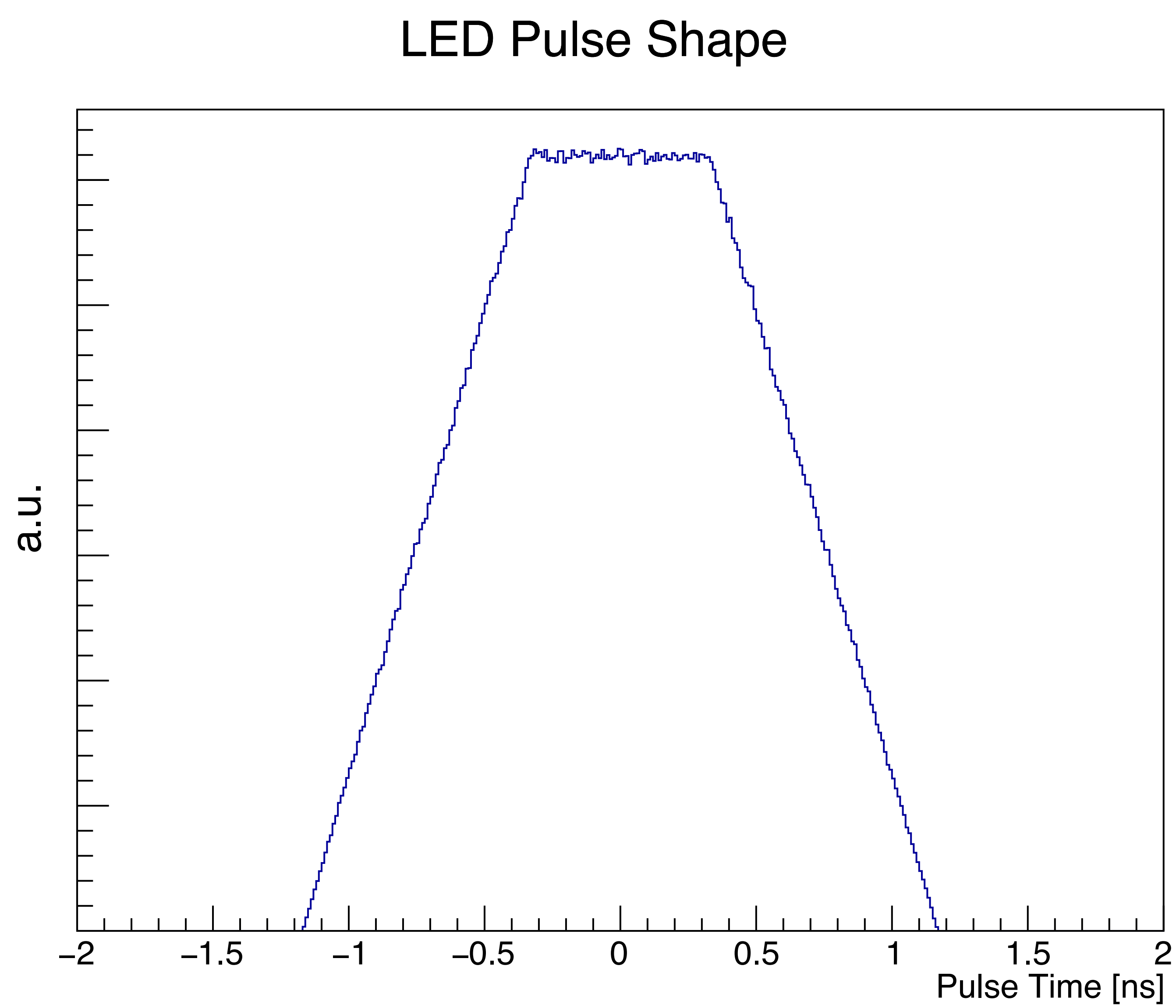
=> Δt ~ 1 ns

=> bonus: agreement between simulation output and pen-and-paper calculation

# $t_{\text{pulse shape}}$

$$t = t_{\text{path}} + \boxed{t_{\text{pulse shape}}} + t_{\text{feedthrough delay}} + t_{\text{time reso.}} + t_{\text{offset,i}}$$

$t_{\text{pulse shape}}$ is a number drawn from this pulse shape PDF:

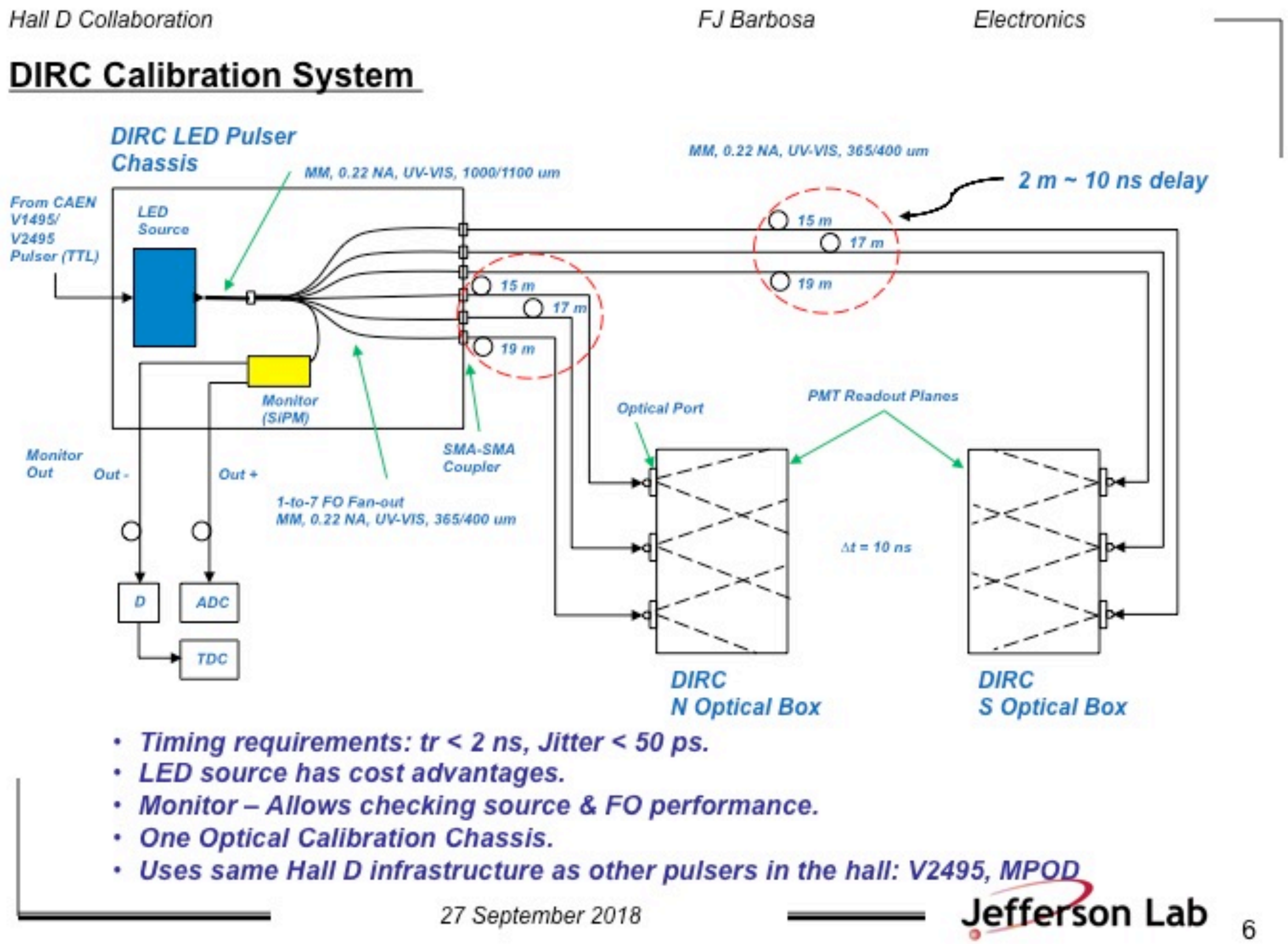



$t_{\text{rise}}$ ~ 0.84 ns

FWHM ~ 1.5 ns

# $t_{\mathrm{feedthrough\ delay}}$

$$t = t_{\mathrm{path}} + t_{\mathrm{pulse\ shape}} + t_{\mathrm{feedthrough\ delay}} + t_{\mathrm{time\ reso.}} + t_{\mathrm{offset,i}}$$



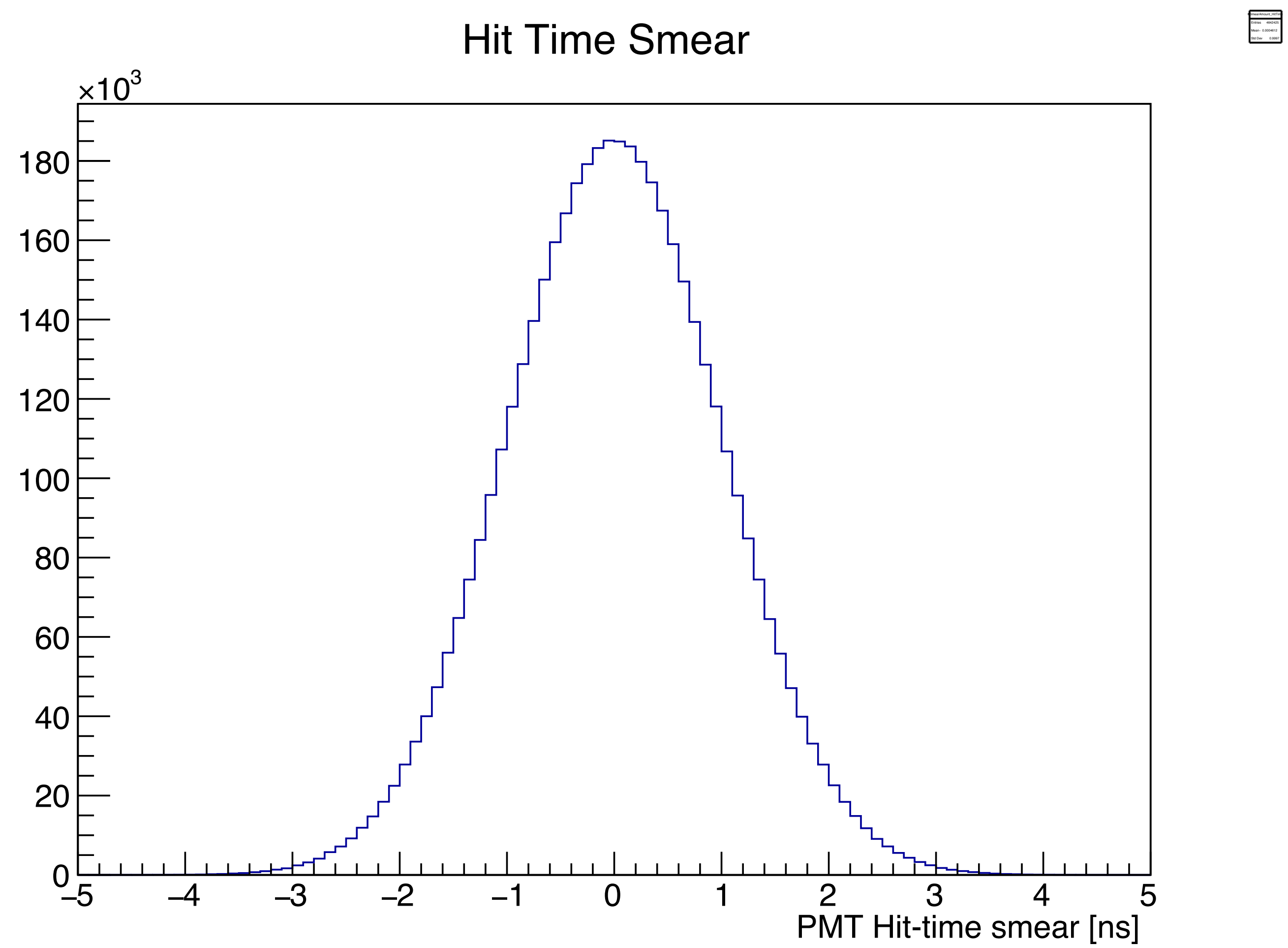

$t_{\mathrm{feedthrough\ delay}}$ = 0 /10 / 20 ns

# $t_{\text{time reso.}}$

$$t = t_{\text{path}} + t_{\text{pulse shape}} + t_{\text{feedthrough delay}} + \boxed{t_{\text{time reso.}}} + t_{\text{offset,i}}$$

$t_{\text{time reso.}}$

- encodes resolution in time measurement, including PMT and electronics resolution;
- is a number drawn from this pulse shape PDF:

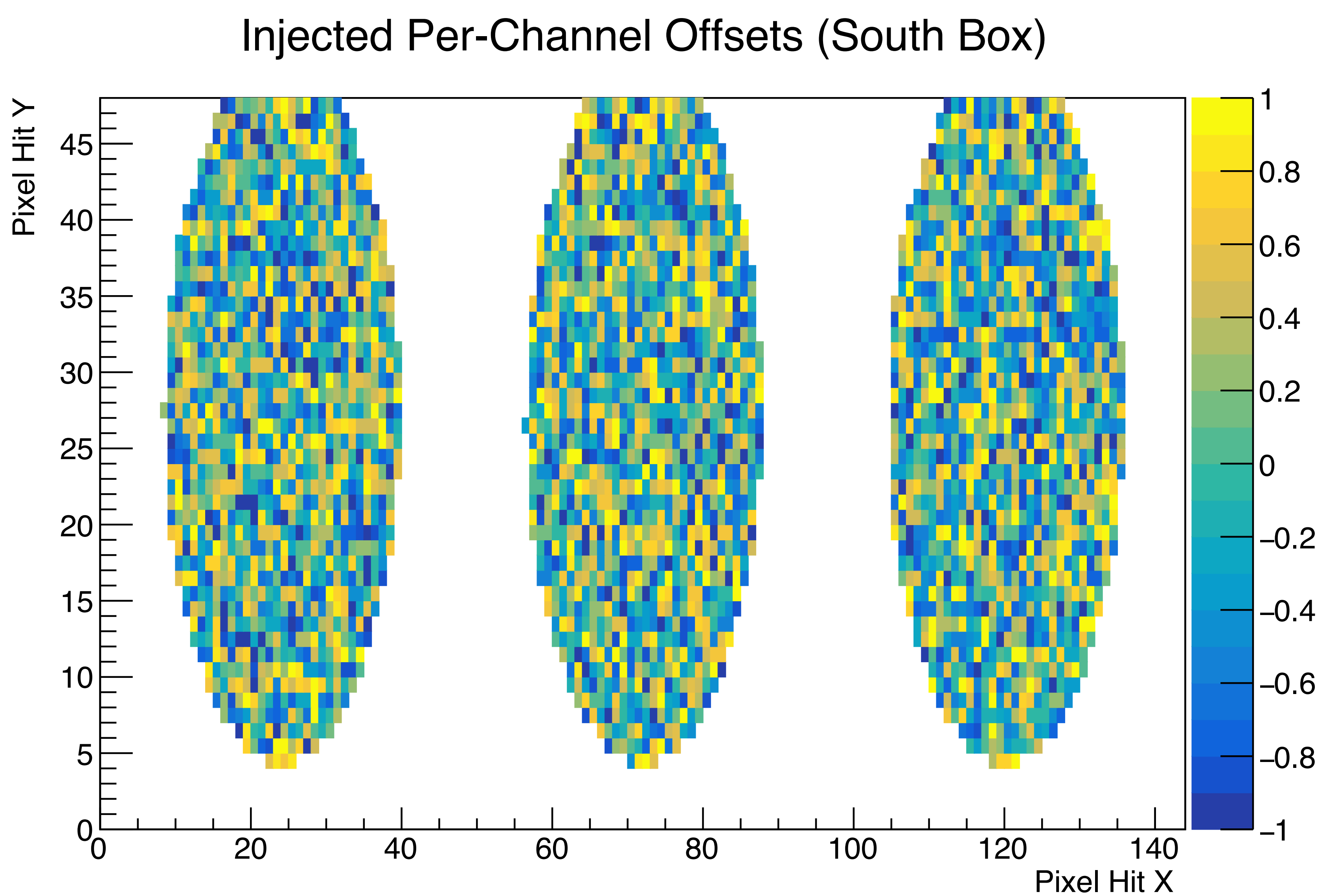



$\sigma = 1$ ns

# $t_{\mathrm{offset,i}}$

$$t = t_{\mathrm{path}} + t_{\mathrm{pulse\ shape}} + t_{\mathrm{feedthrough\ delay}} + t_{\mathrm{time\ reso.}} + \boxed{t_{\mathrm{offset,i}}}$$

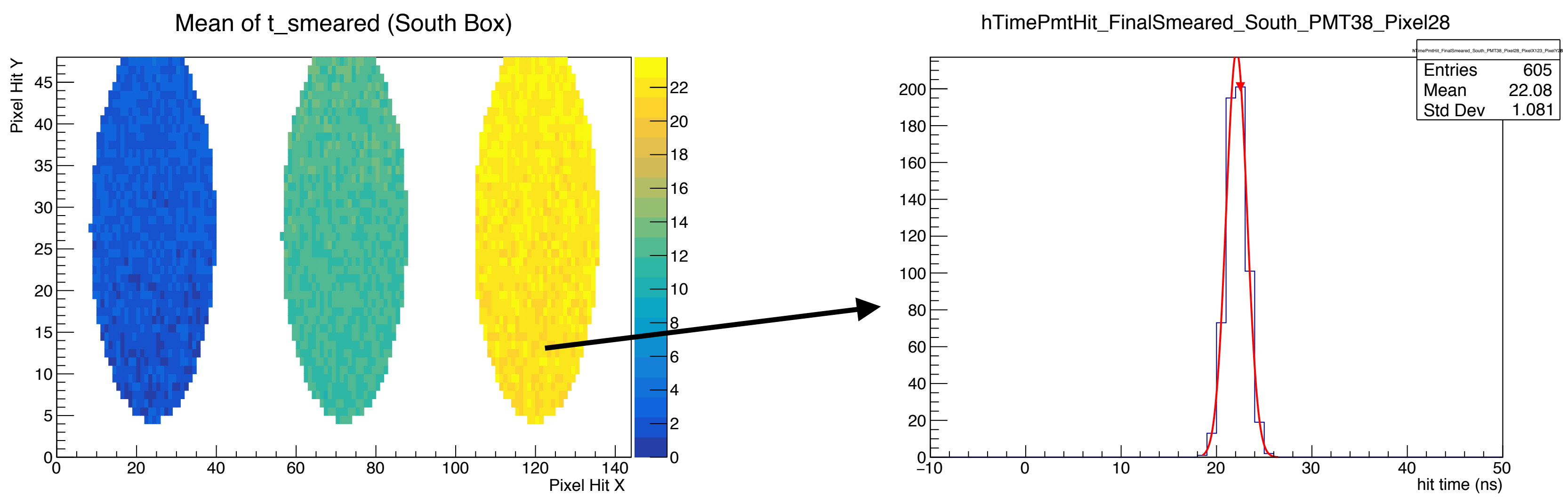
For each channel, choose a random number between [-1,1]:




Note: The map is generated for all channels, but only showing the ones that will be populated with LEDs this time.

# Strategy to extract $t_{\mathrm{offset,i}}$

1. For each channel, fit the final time distribution to extract $t_{\mathrm{fitted}}$



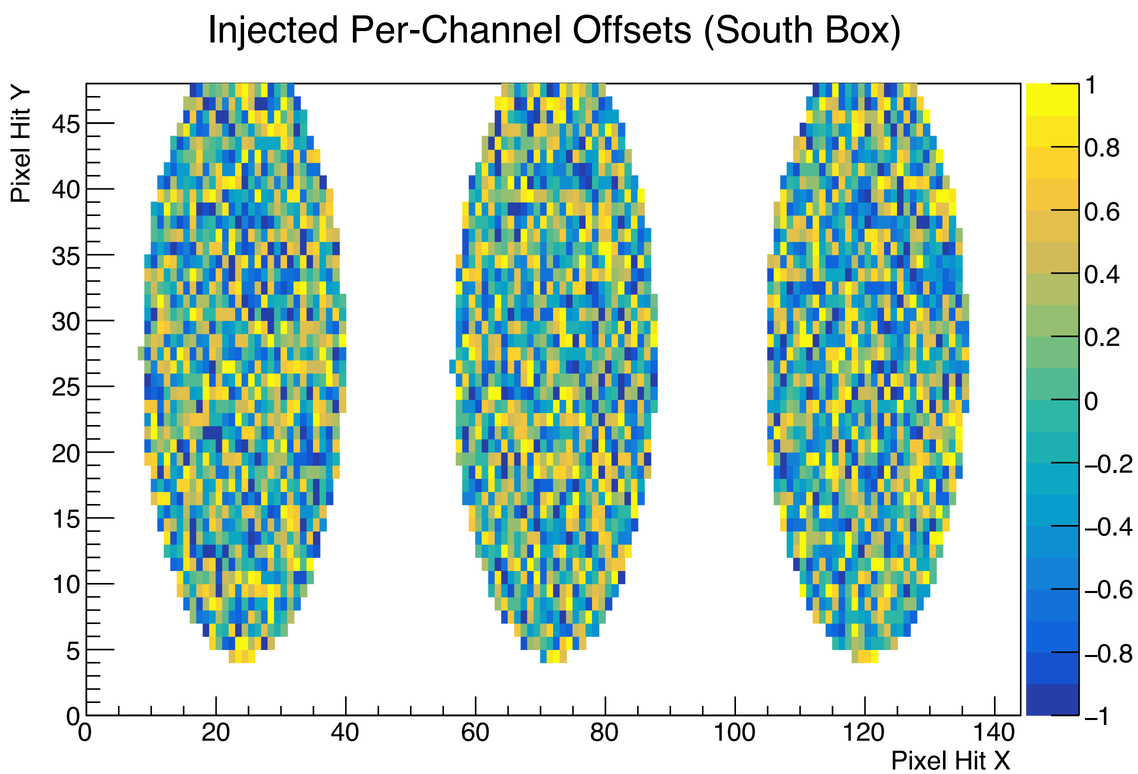

2. Use the following to get the extracted offset for that channel

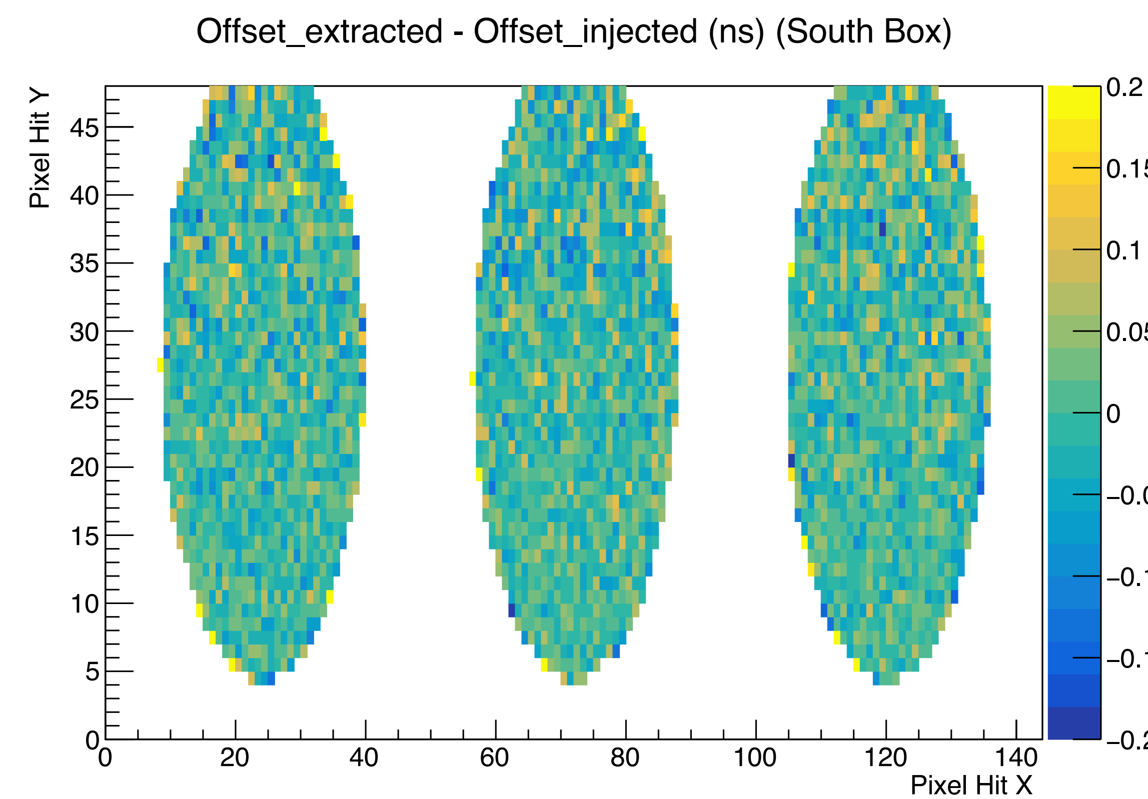
$$t_{\mathrm{offset,i,extracted}} = t_{\mathrm{fitted}} - t_{\mathrm{path}} - t_{\mathrm{feedthrough\ delay}}$$

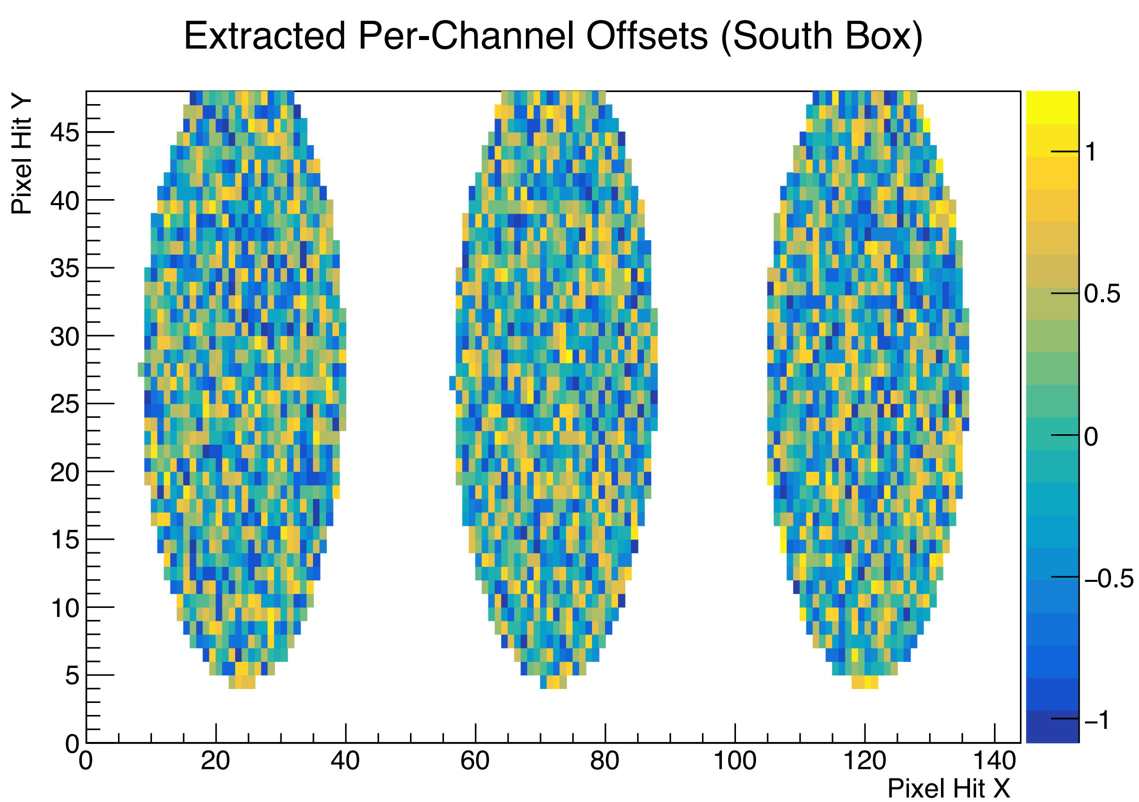
where $t_{\mathrm{path}}$ and $t_{\mathrm{feedthrough\ delay}}$ are the corresponding values for that channel

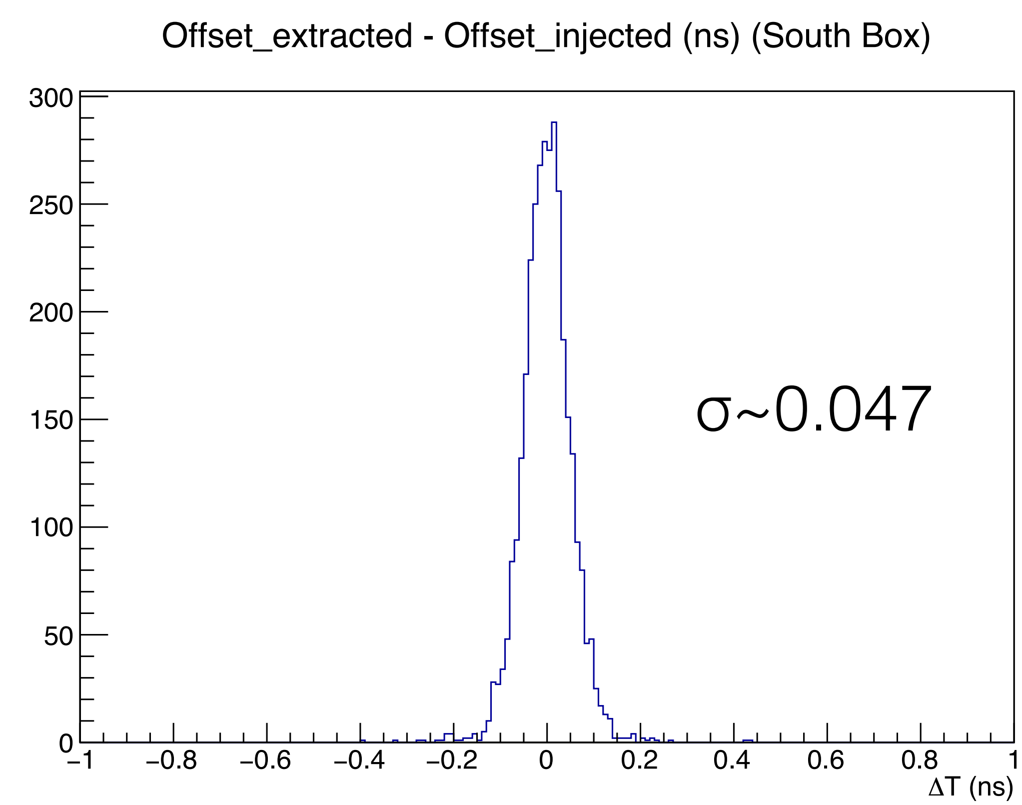
3. Compare with the injected offset for that channel

# Results (I)



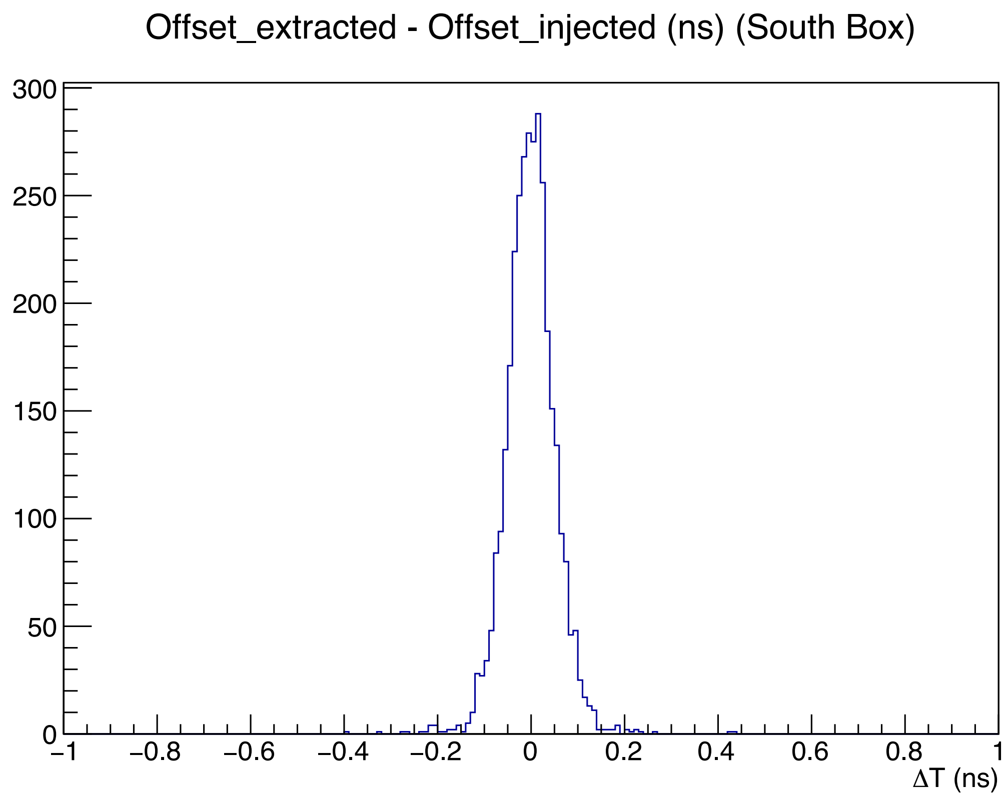



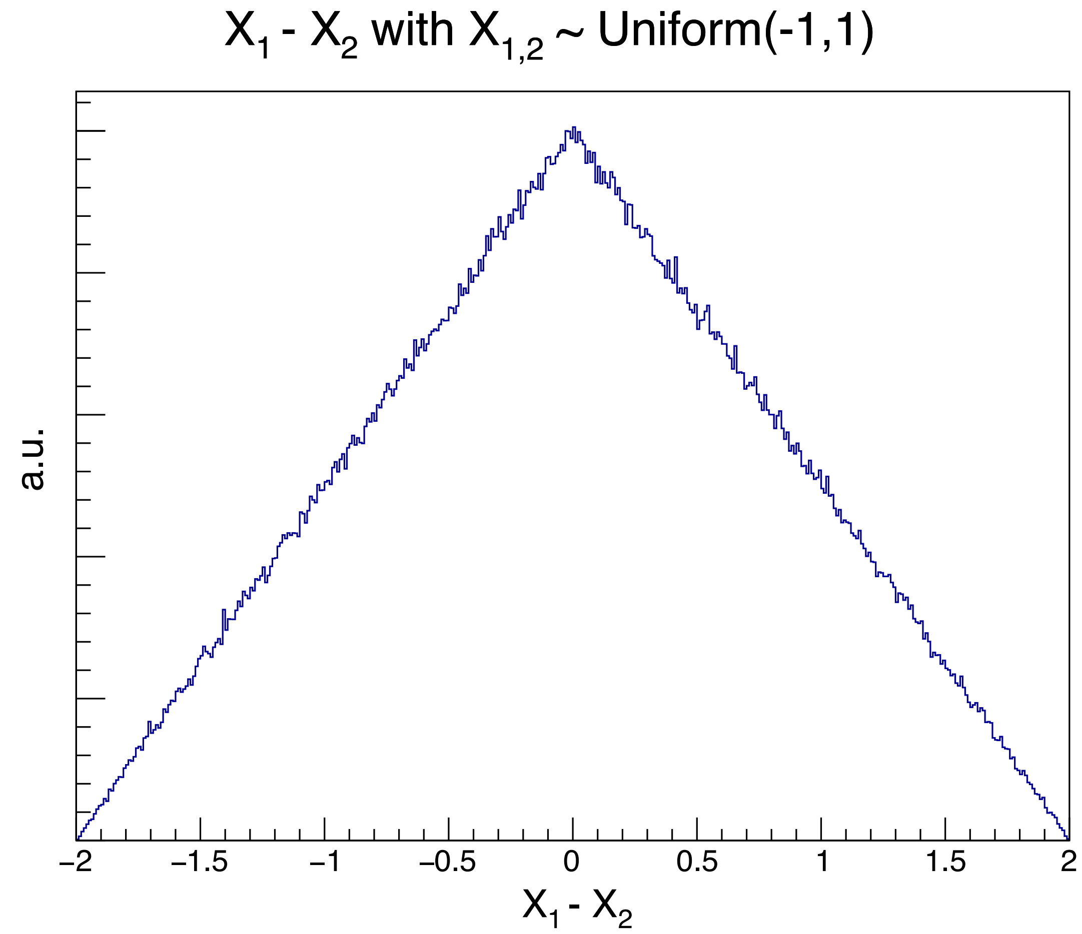

=> able to recover injected offsets within ~0.2 ns for almost all channels

# Results (I)

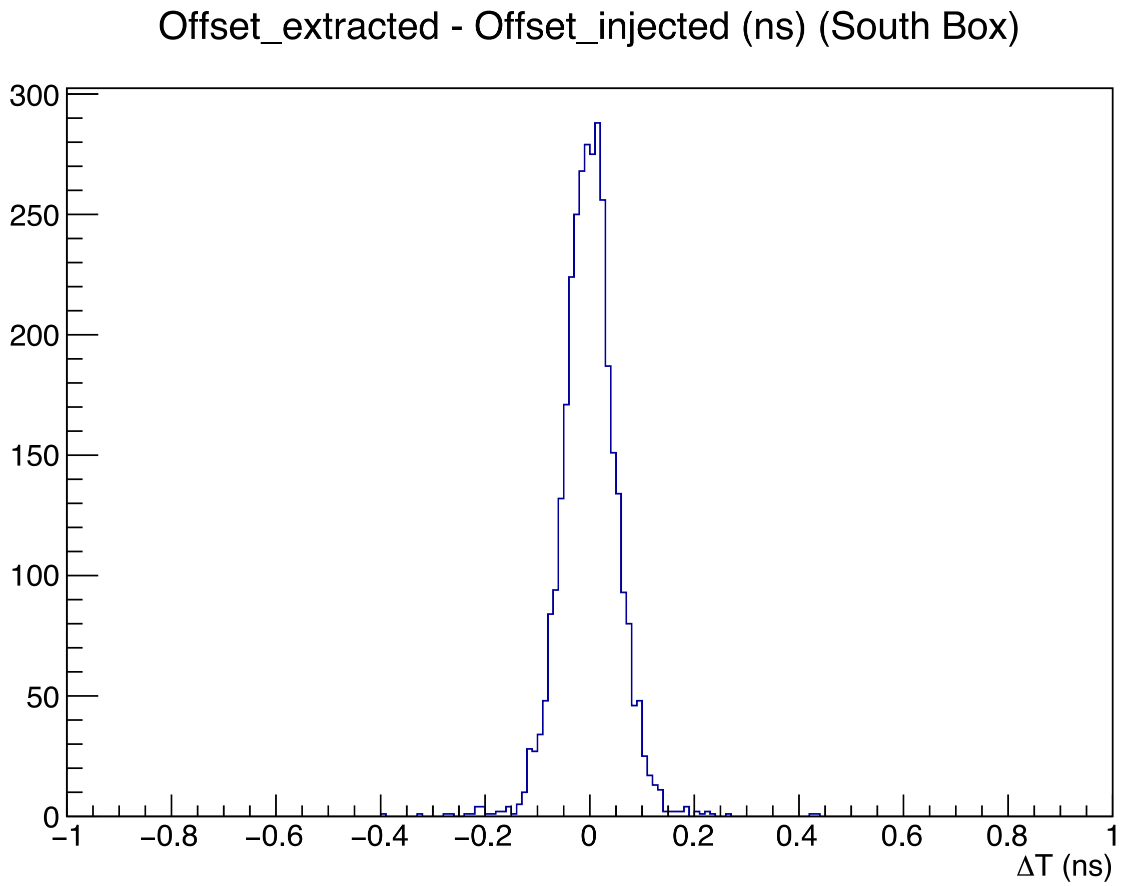
Much better than random guess…

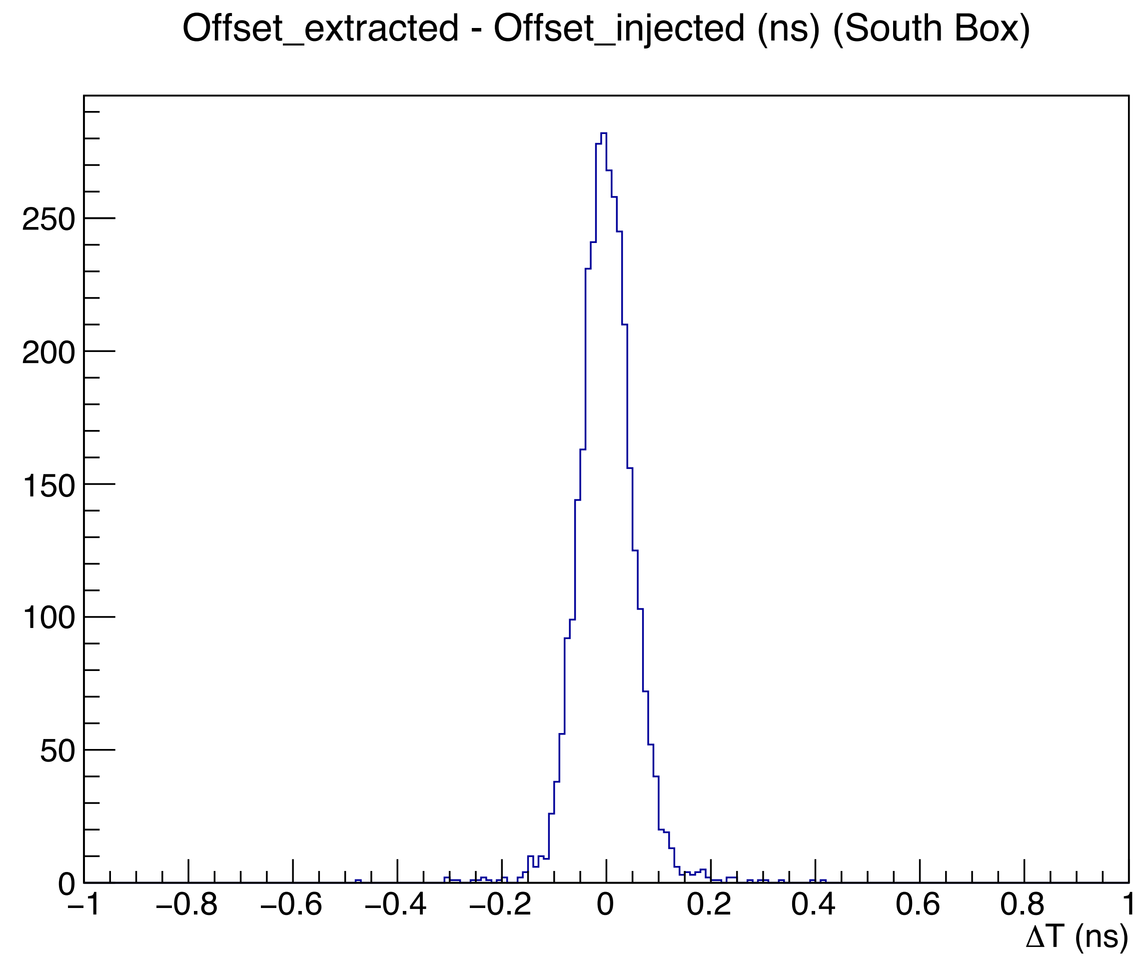



vs.



# Results (II): change range of offsets



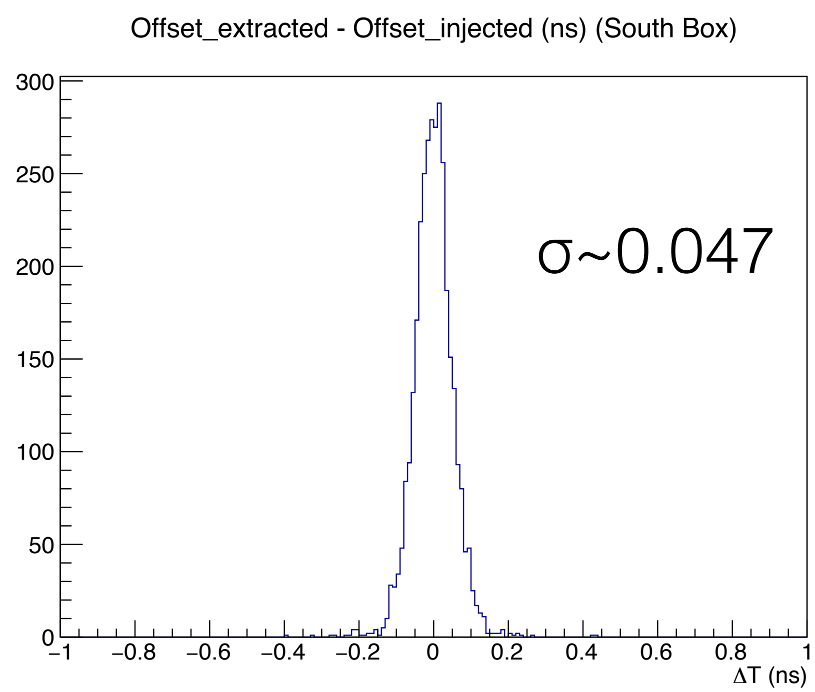

offset: random number from [-1,1]
PMT resolution: 1ns

offset: random number from [-2,2]
PMT resolution: 1ns

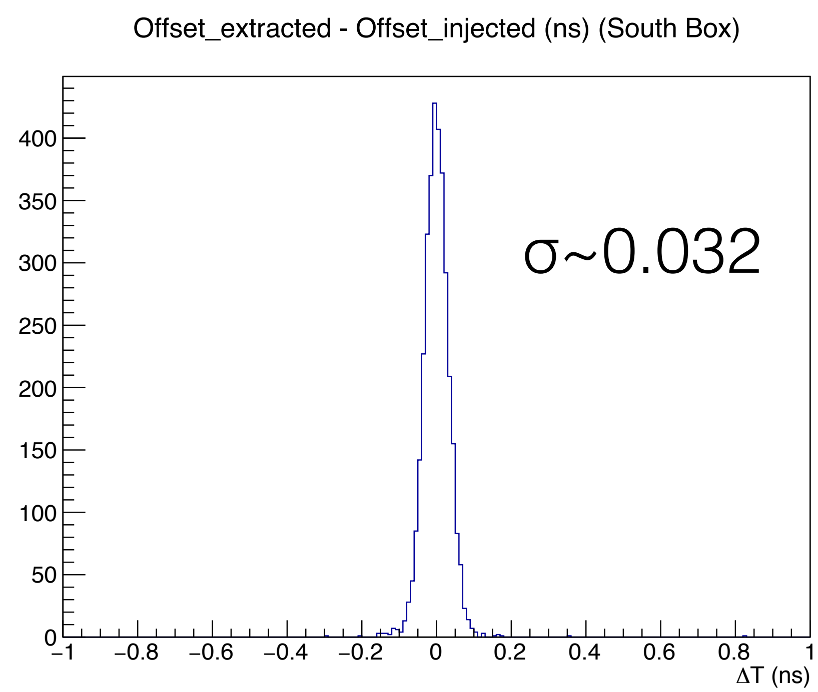
=> no difference in performance

# Results (III): change timing resolution

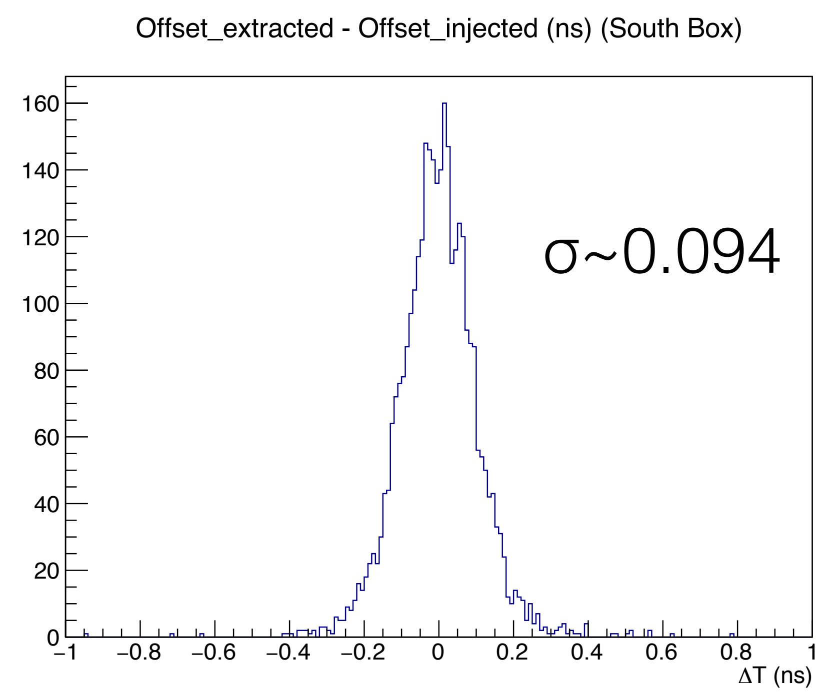
offset: random number from [-1,1]



timing resolution: 1 ns



timing resolution: 0.5 ns



timing resolution: 2 ns

=> gets worse as timing resolution gets bad, as expected

# Application to LED data

$$t_{\text{offset,i,extracted}} = t_{\text{fitted}} - t_{\text{path}} - t_{\text{feedthrough delay}}$$

✓ ($t_{\text{fitted}}$)

($t_{\text{path}}$) → Create a map from simulation ✓

($t_{\text{feedthrough delay}}$) → How well do we know them? If well enough, ✓

If not, one can plot

$$t_{\text{fitted}} - t_{\text{path}}$$

$$(= t_{\text{feedthrough delay}} + t_{\text{offset,i}})$$

and estimate from it?