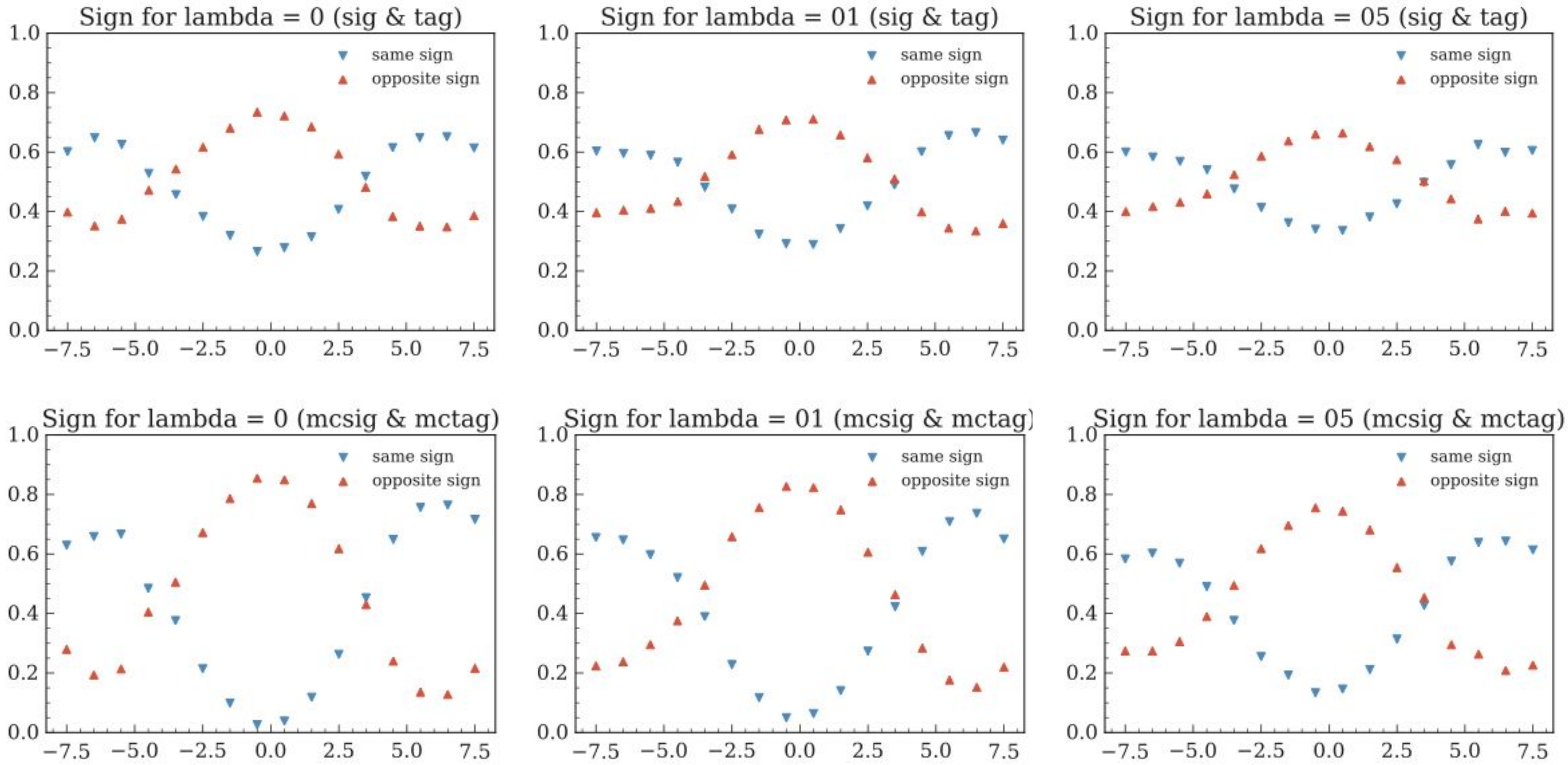
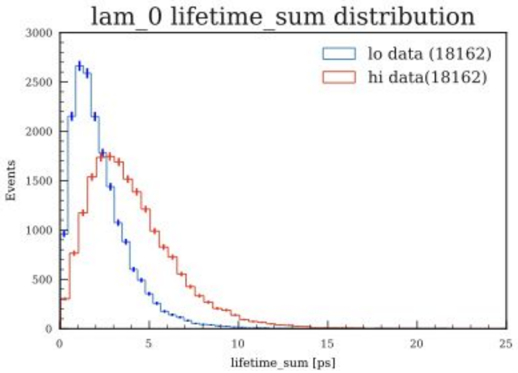
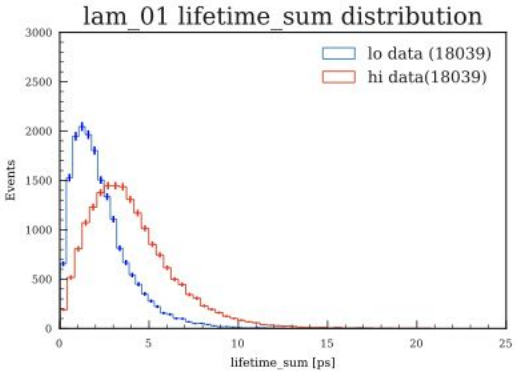
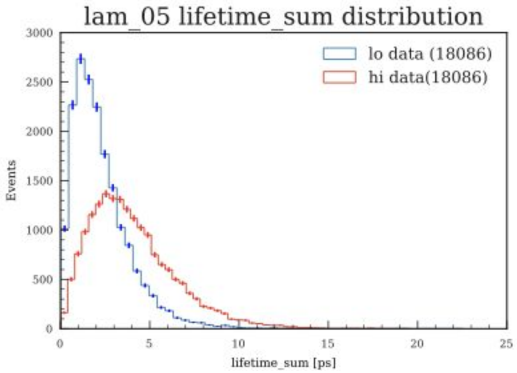
same/opposite sign plots using “sig/tagflav” and “**mc**sig/tagflav”:

λ=0 λ=0.1 λ=0.5

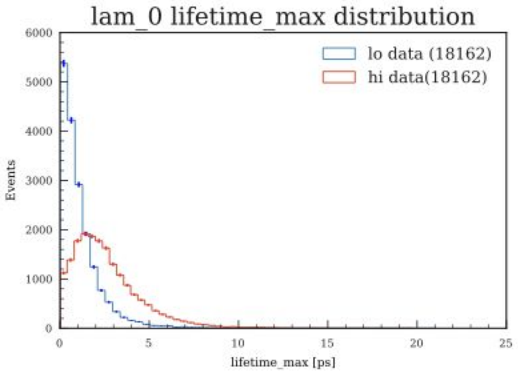


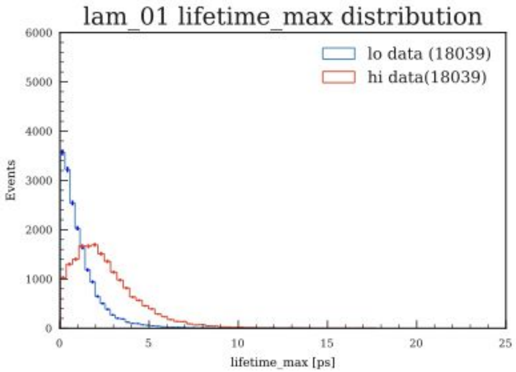


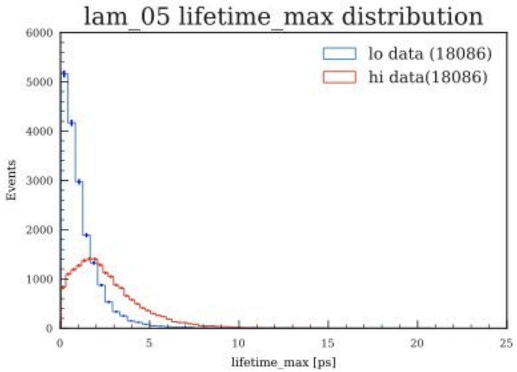




lifetime_max = lifetime of B meson with larger x_vertex from IP

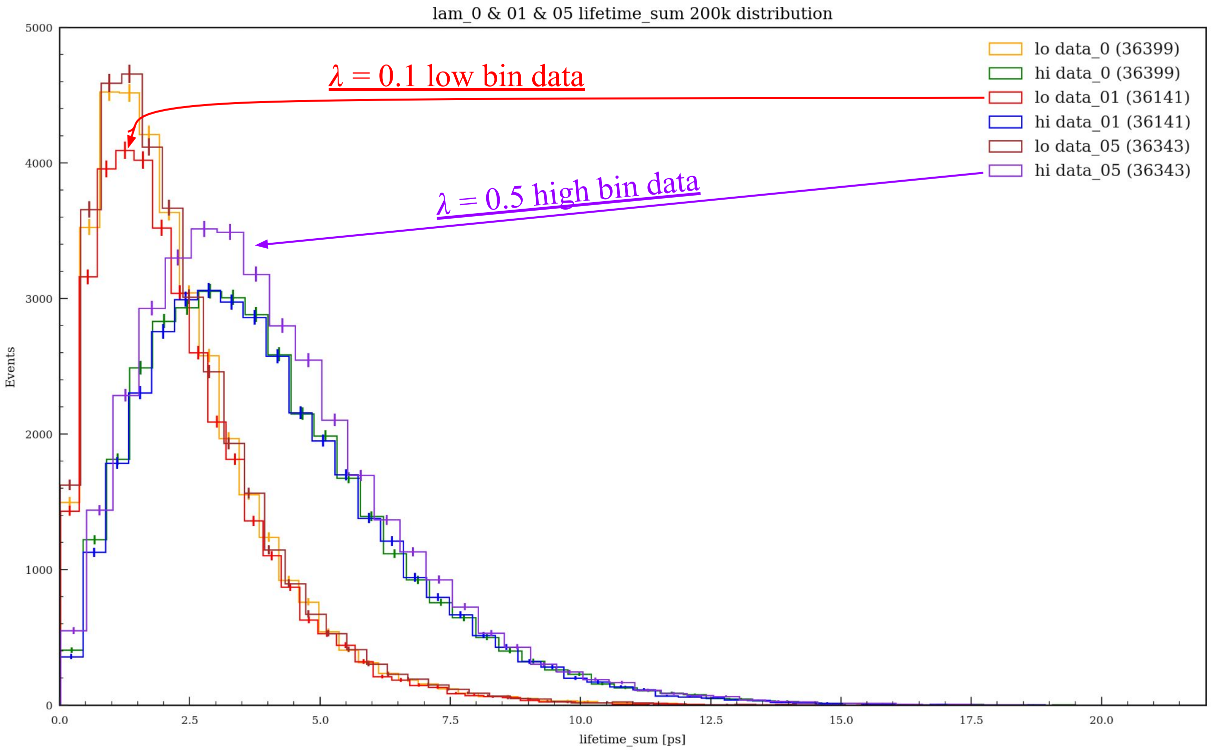






From this distribution, we see significant deviations by the $\lambda = 0.1$ low bin data, and the $\lambda = 0.5$ high bin data. Either:

- These are bugs in the generation/analysis ($\lambda = 0.5$ looks like it has too much data, $\lambda = 0.1$ looks like it has too little)
- ~~There is a non-trivial dependence on λ (=0 gives standard dist., turning on λ causes a discontinuity in dist., which then evolves continuously in λ)~~
    - ~~want to test this by plotting different λ values ($\lambda = 0.05$ & $\lambda = 1$)~~



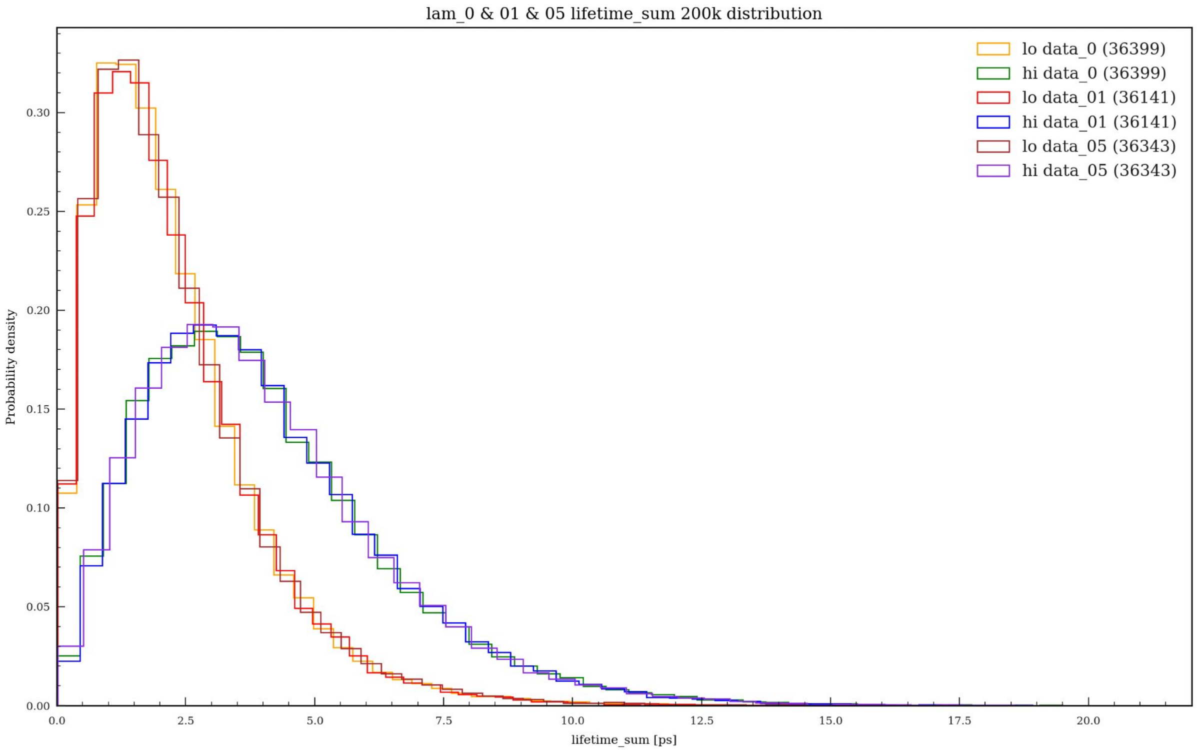
total events:
plt.hist(..., density=False, …)

A normalized plot of these distributions show a much more equal relationship for each λ

→ Implies issue with my analysis (analysus…) rather than decoherence dependence

<u>NOT SURE THAT THIS CAN BE TRUSTED</u>!!

- normalizes each of the 6 curves individually → lo/hi bins know nothing about each other

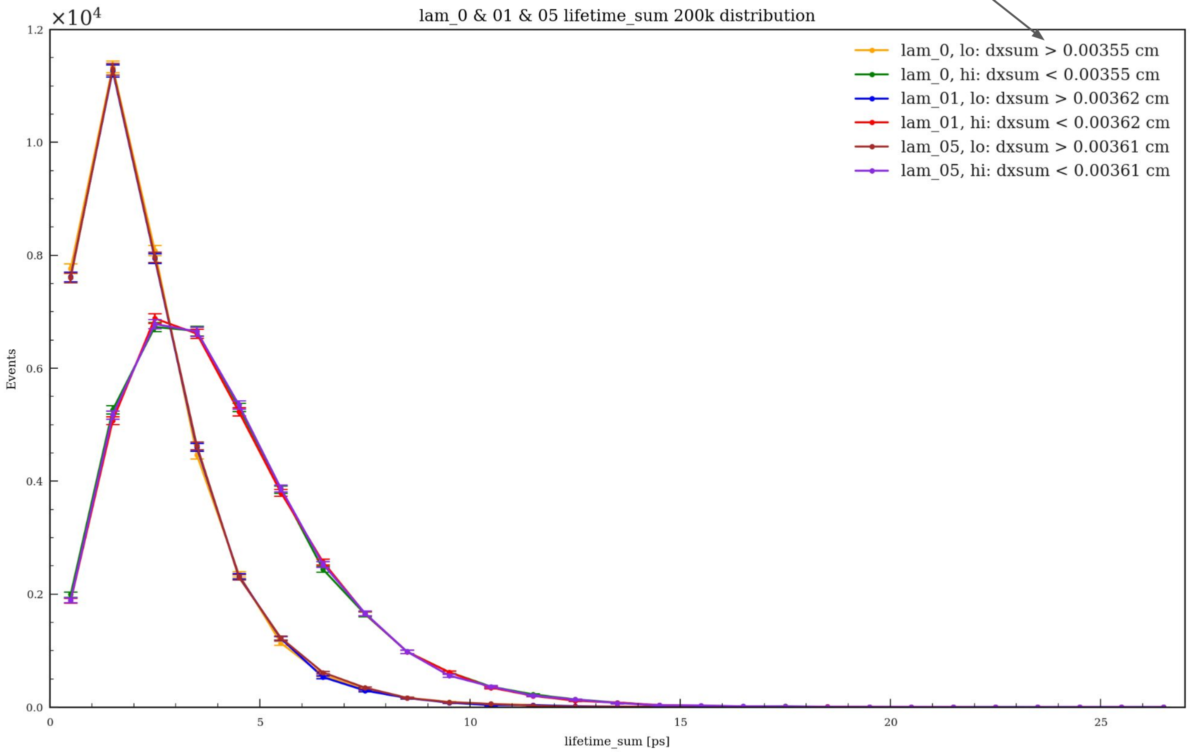


relative:
plt.hist(..., density=True, …)

Standardize the binwidths/centers…

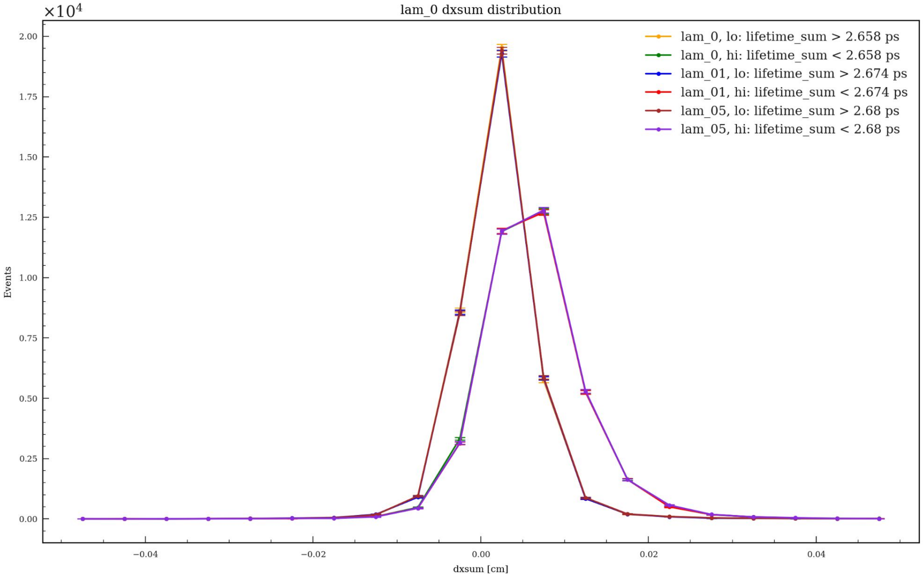
Making the binwidths the same for each dataset seems to fix things → <u>no λ dependence</u>

legend is wrong

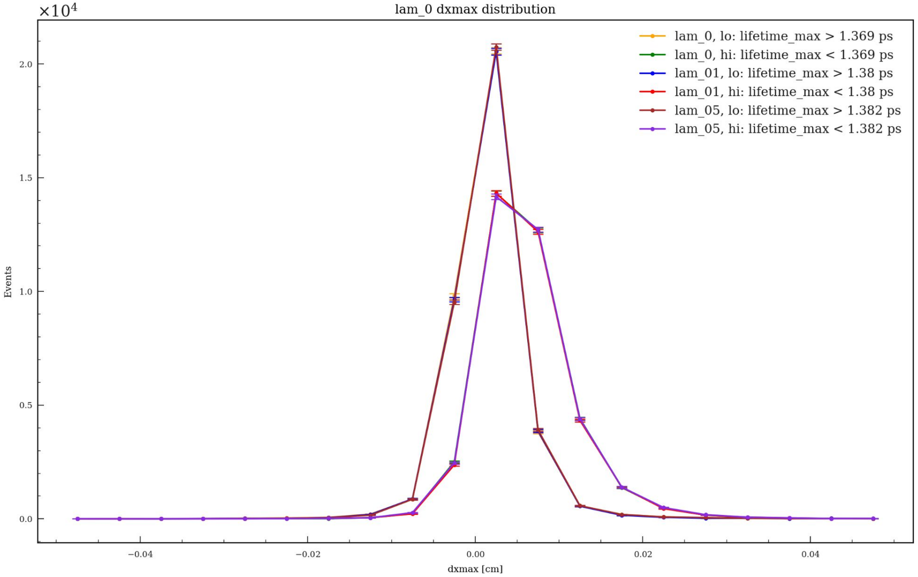


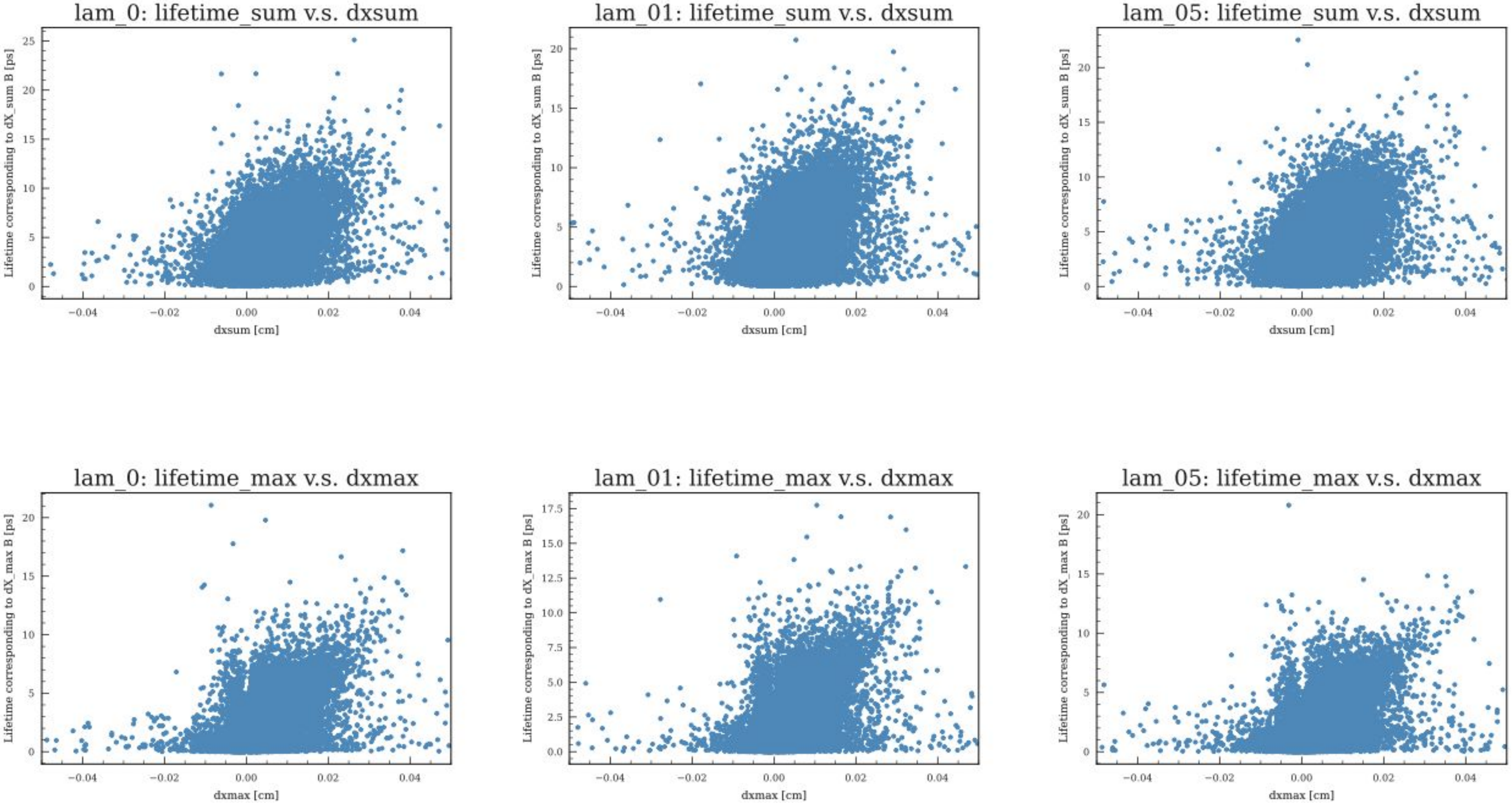
Looking at other relationships:

$\sum dx$ values for hi/lo $\sum t$ bins

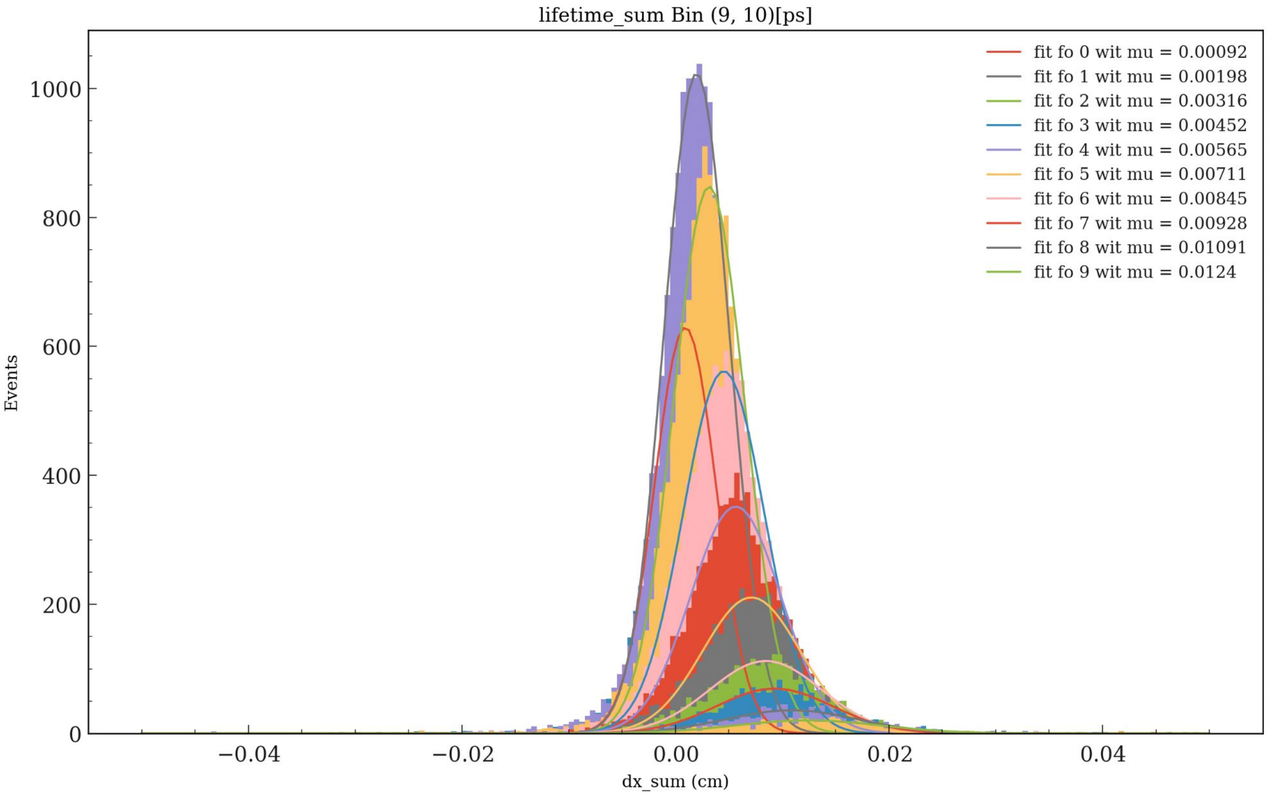


dx_{max} values for hi/lo t_{max} bins




lam_0: lifetime_sum v.s. dxsum
Lifetime corresponding to dX_sum B [ps]
dxsum [cm]
lam_01: lifetime_sum v.s. dxsum
Lifetime corresponding to dX_sum B [ps]
dxsum [cm]
lam_05: lifetime_sum v.s. dxsum
Lifetime corresponding to dX_sum B [ps]
dxsum [cm]
lam_0: lifetime_max v.s. dxmax
Lifetime corresponding to dX_max B [ps]
dxmax [cm]
lam_01: lifetime_max v.s. dxmax
Lifetime corresponding to dX_max B [ps]
dxmax [cm]
lam_05: lifetime_max v.s. dxmax
Lifetime corresponding to dX_max B [ps]
dxmax [cm]

Linear binning of the ∑Lifetime variable. Fitting the corresponding ∑dx distribution with a Gaussian → doesn't seem to describe tails well enough…



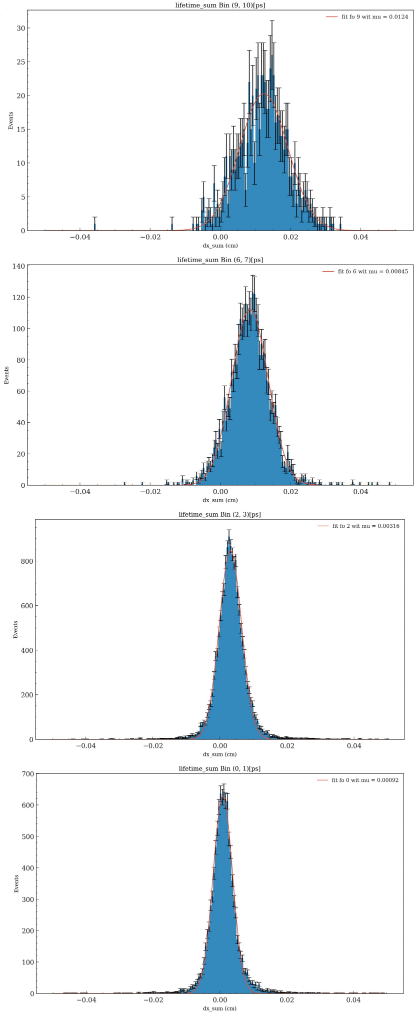
∑Lifetime bin:

(9, 10) [ps]

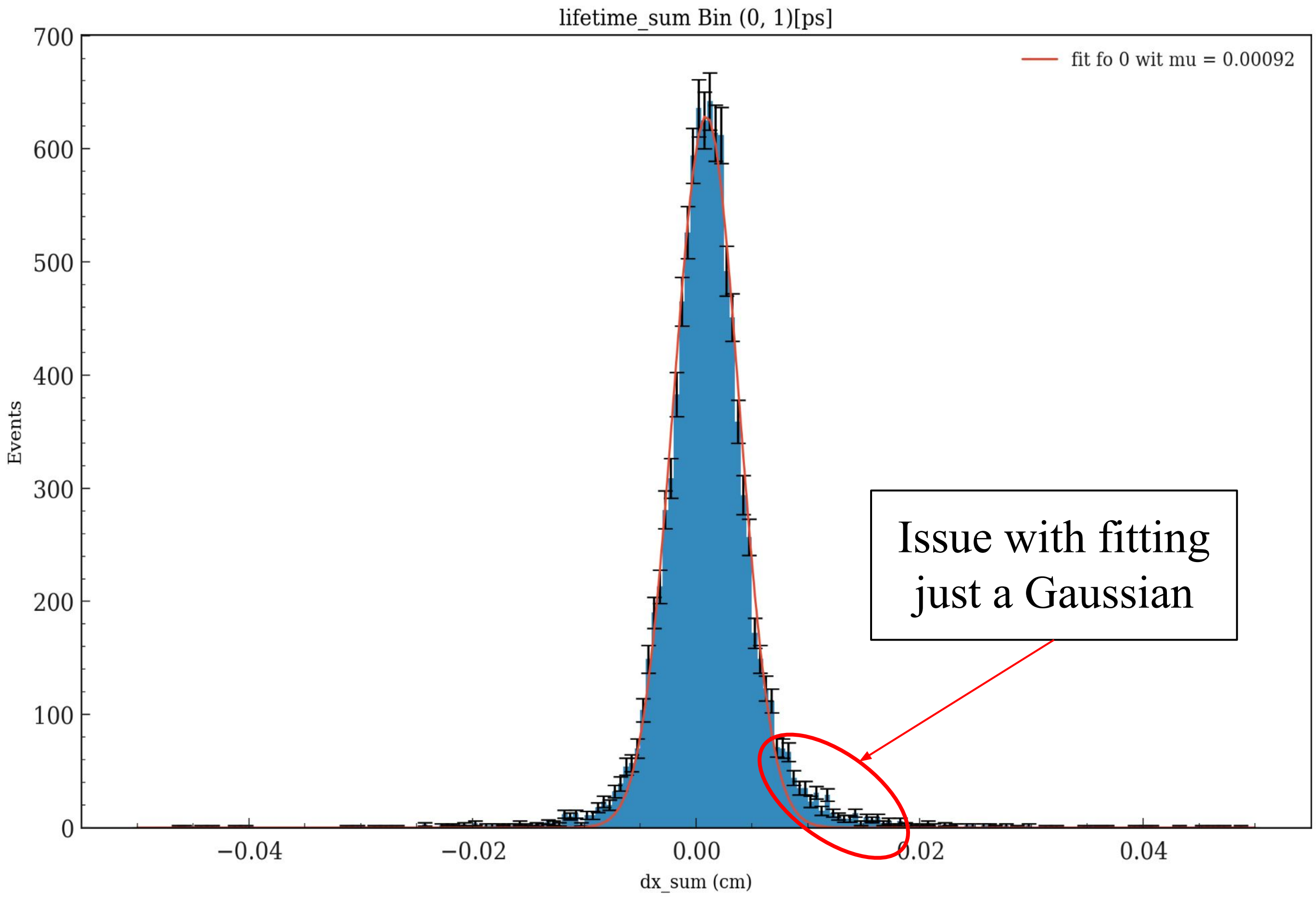
(6, 7) [ps]

(2, 3) [ps]

(0, 1) [ps]

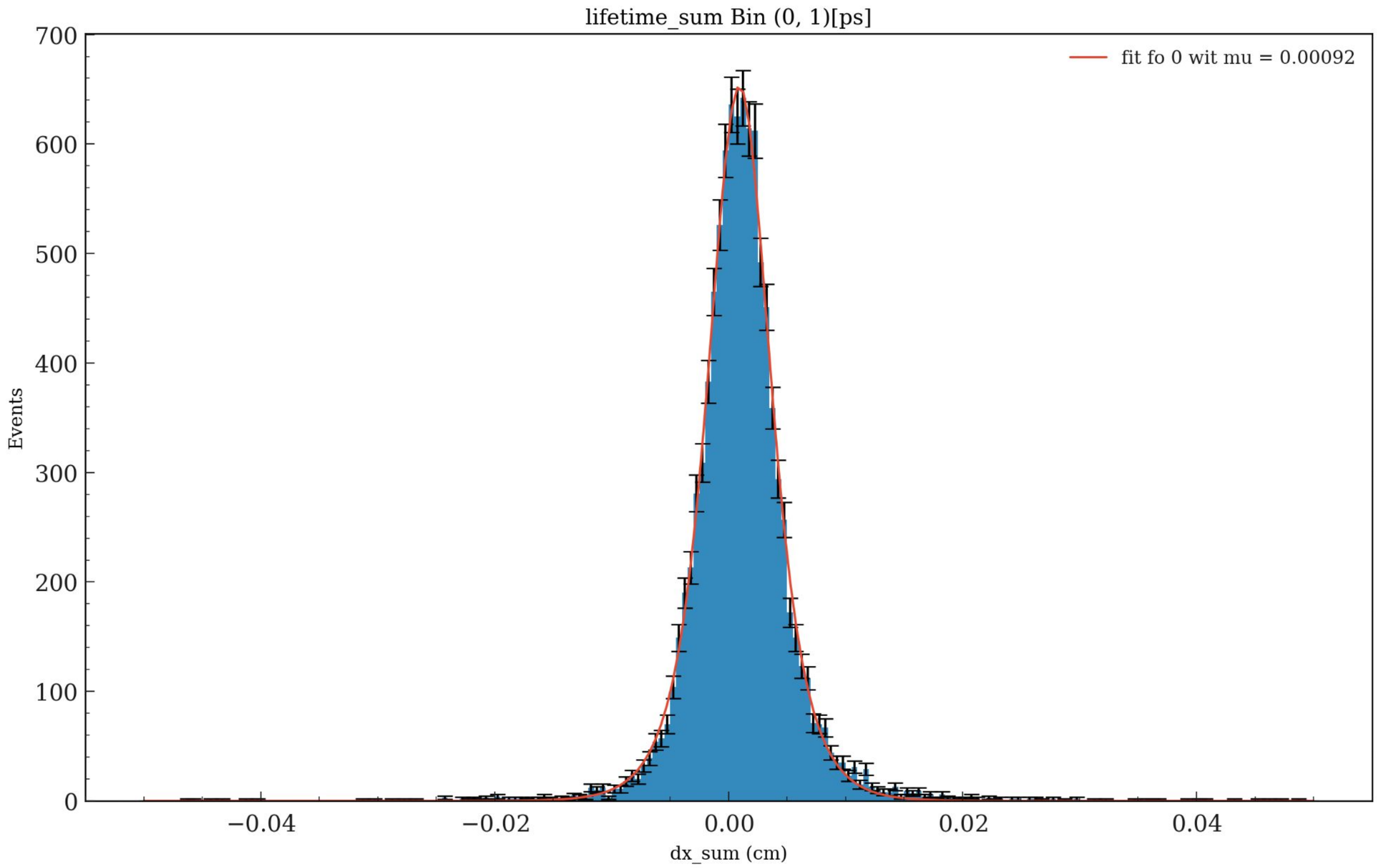


∑dx [cm]

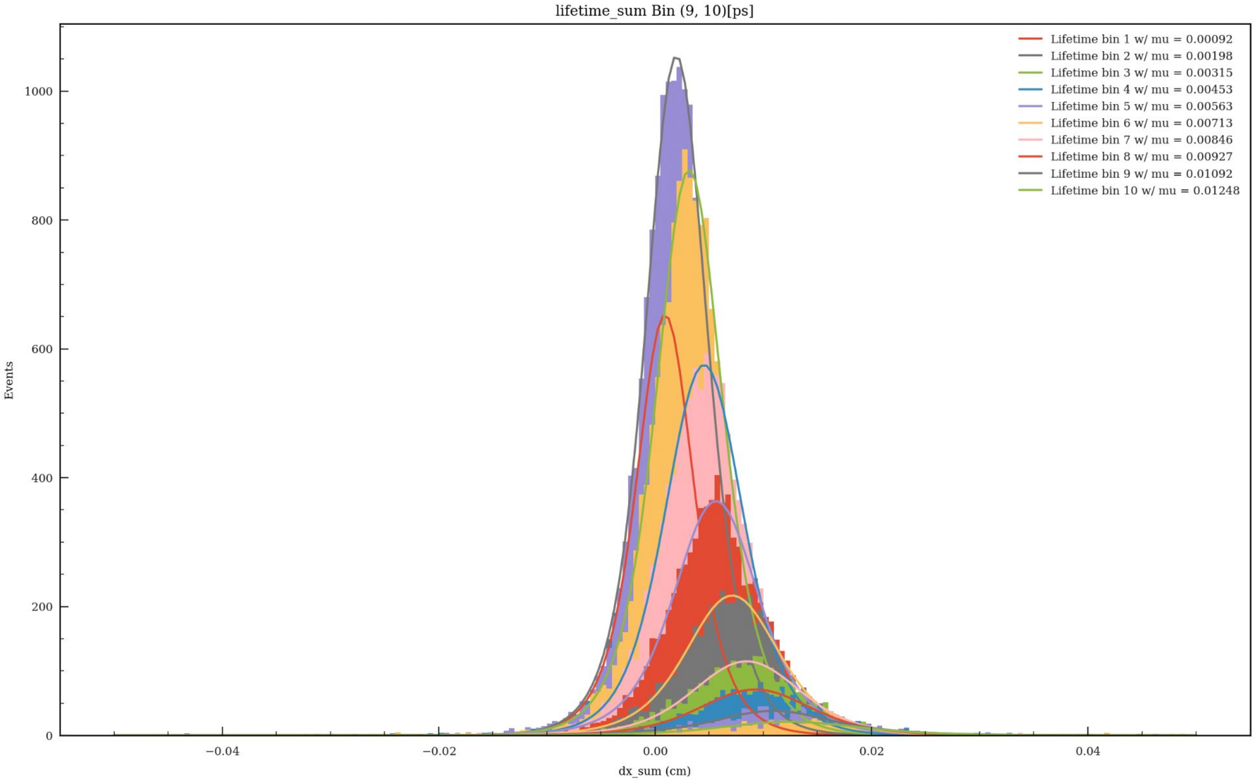


lifetime_sum Bin (0, 1)[ps]
fit fo 0 wit mu = 0.00092
Events
dx_sum (cm)
700
600
500
400
300
200
100
0
−0.04
−0.02
0.00
0.02
0.04
Issue with fitting just a Gaussian

Initial results from a Gaussian convolved with Exponential tails to either side: (using curve_fit, χ^2/NDF needs to be calculated manually)



Updated using a Gaussian with exponential tails on either side. Rudimentary fitting using curve_fit, χ^2/NDF needs to be calculated manually.



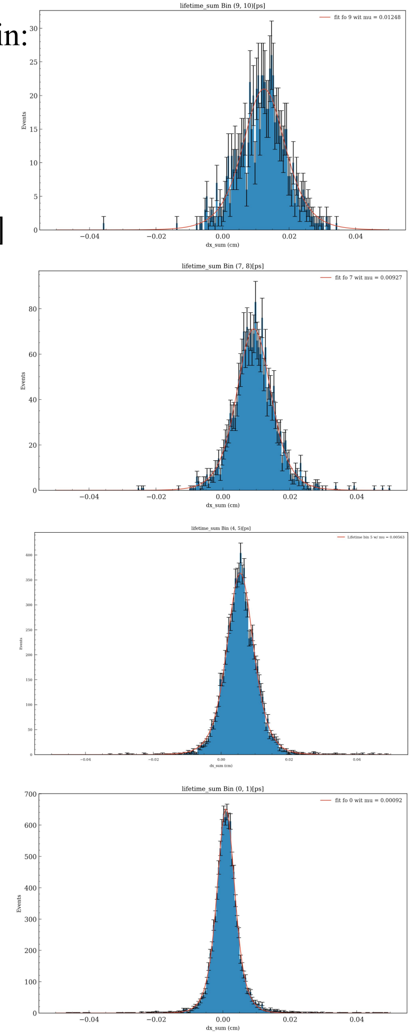
∑Lifetime bin:

(9, 10) [ps]

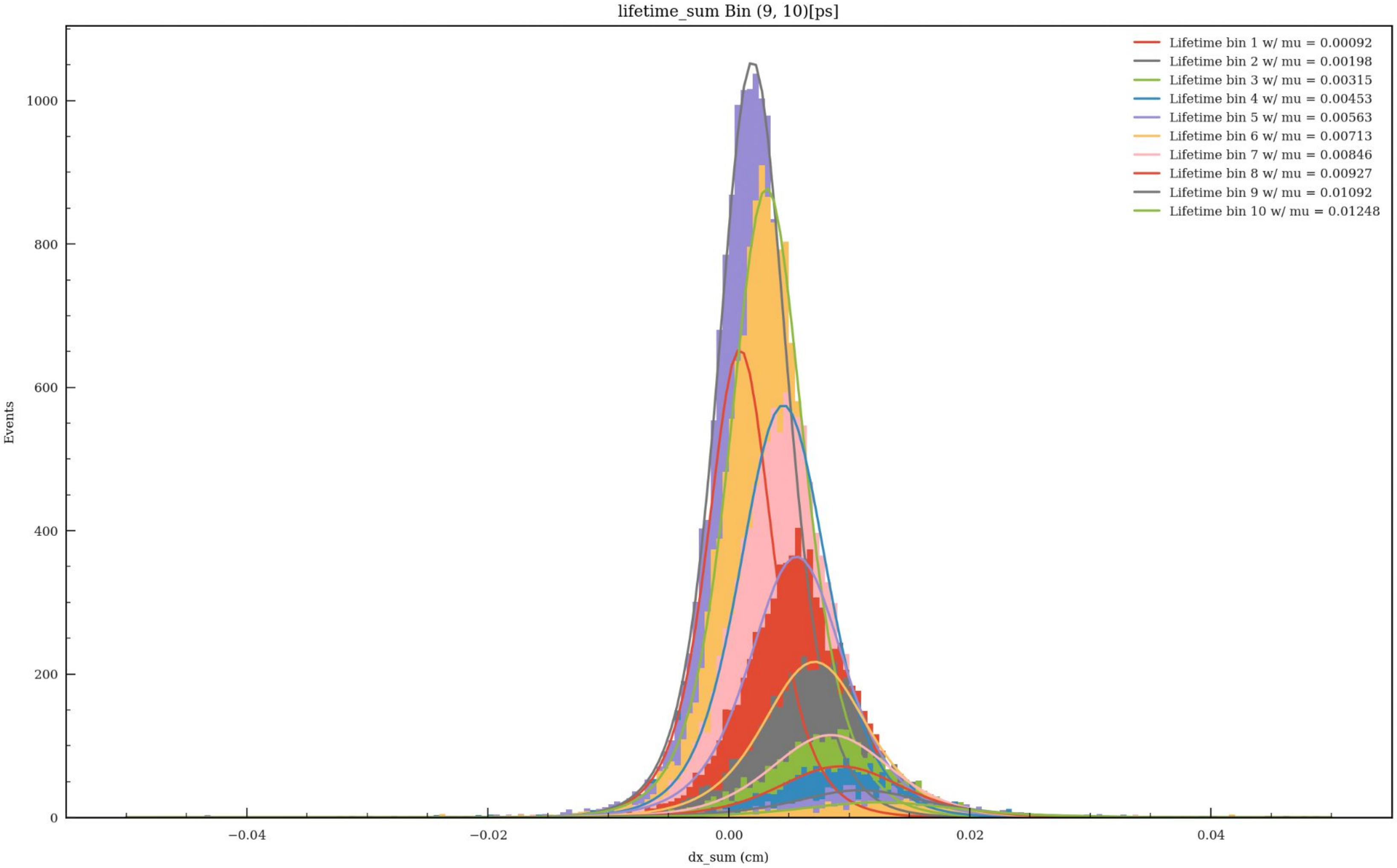
(7, 8) [ps]

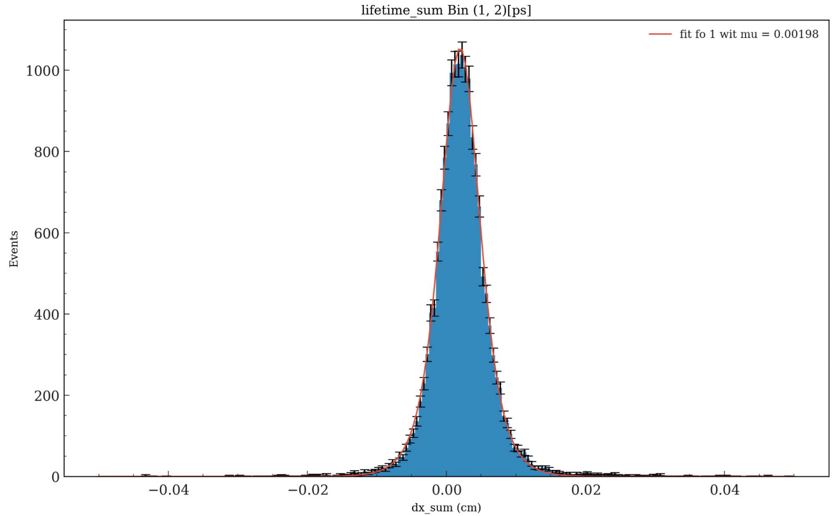
(4, 5) [ps]

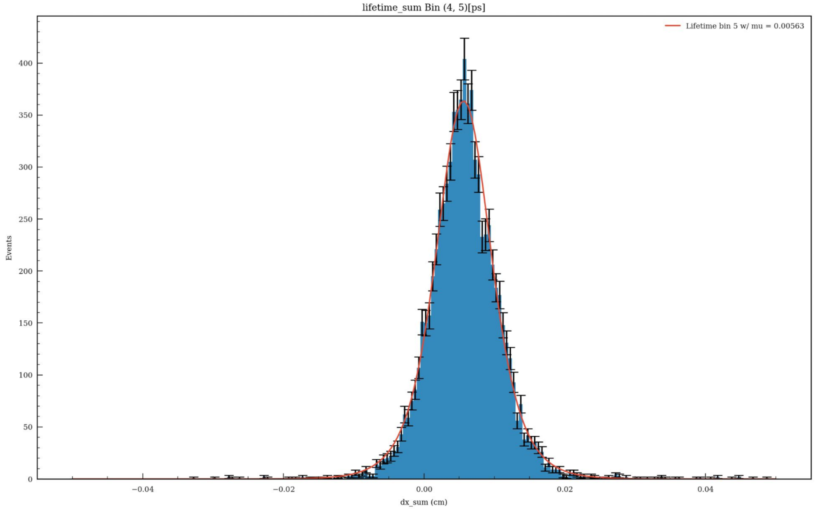
(0, 1) [ps]

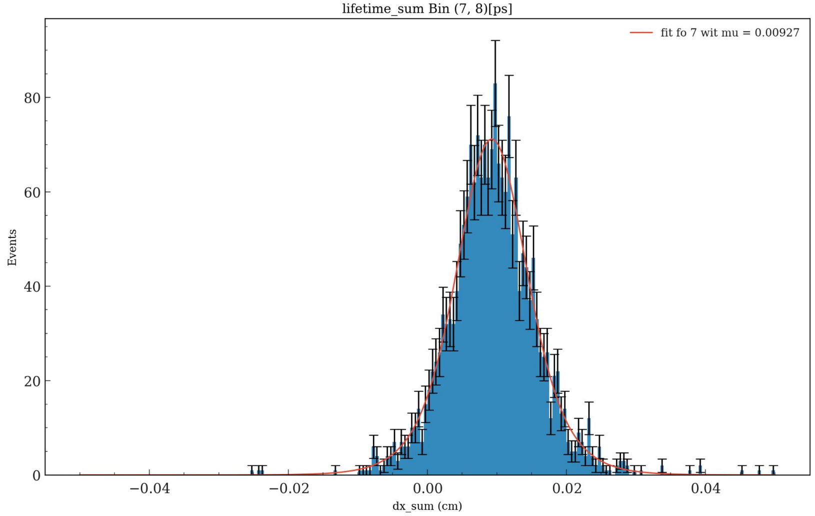


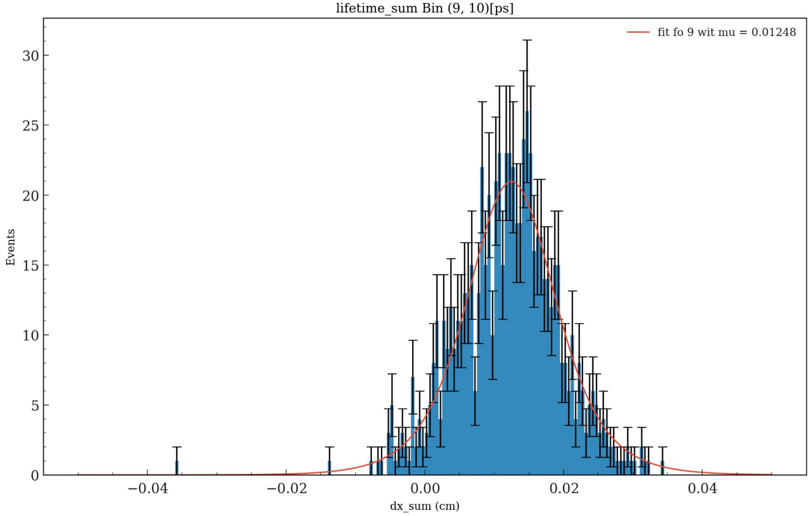
∑dx [cm]


lifetime_sum Bin (9, 10)[ps]
Lifetime bin 1 w/ mu = 0.00092
Lifetime bin 2 w/ mu = 0.00198
Lifetime bin 3 w/ mu = 0.00315
Lifetime bin 4 w/ mu = 0.00453
Lifetime bin 5 w/ mu = 0.00563
Lifetime bin 6 w/ mu = 0.00713
Lifetime bin 7 w/ mu = 0.00846
Lifetime bin 8 w/ mu = 0.00927
Lifetime bin 9 w/ mu = 0.01092
Lifetime bin 10 w/ mu = 0.01248
Events
1000
800
600
400
200
0
−0.04
−0.02
0.00
0.02
0.04
dx_sum (cm)

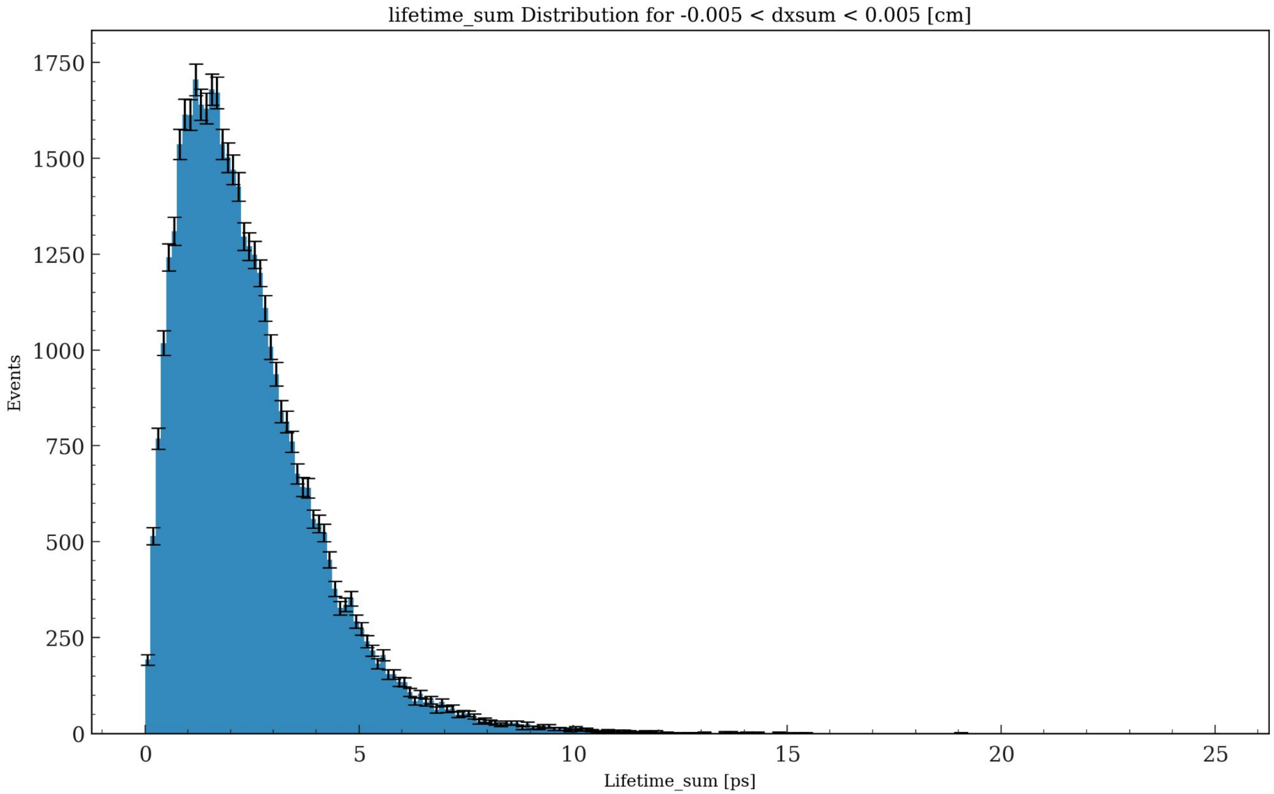

lifetime_sum Bin (1, 2)[ps]
fit fo 1 wit mu = 0.00198
Events
dx_sum (cm)
0
200
400
600
800
1000
−0.04
−0.02
0.00
0.02
0.04


lifetime_sum Bin (4, 5)[ps]
Lifetime bin 5 w/ mu = 0.00563
Events
dx_sum (cm)
0
50
100
150
200
250
300
350
400
−0.04
−0.02
0.00
0.02
0.04


lifetime_sum Bin (7, 8)[ps]
fit fo 7 wit mu = 0.00927
Events
dx_sum (cm)
0
20
40
60
80
−0.04
−0.02
0.00
0.02
0.04

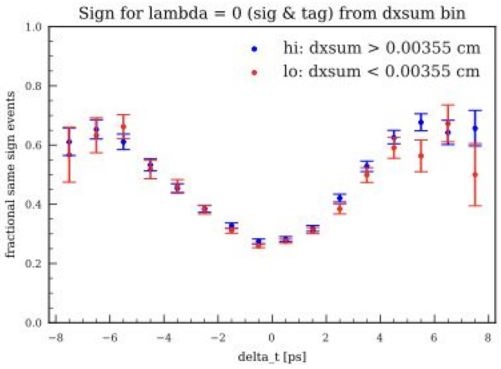

lifetime_sum Bin (9, 10)[ps]
fit fo 9 wit mu = 0.01248
Events
dx_sum (cm)
0
5
10
15
20
25
30
−0.04
−0.02
0.00
0.02
0.04

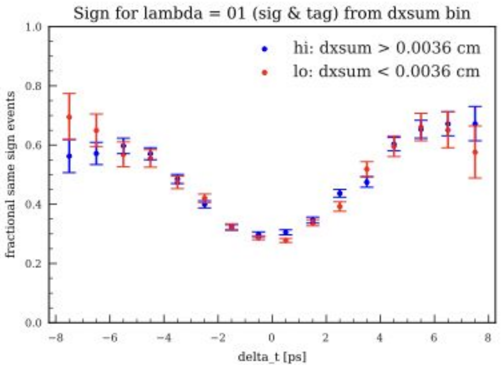
Binning in lifetime_sum shape: not exponential

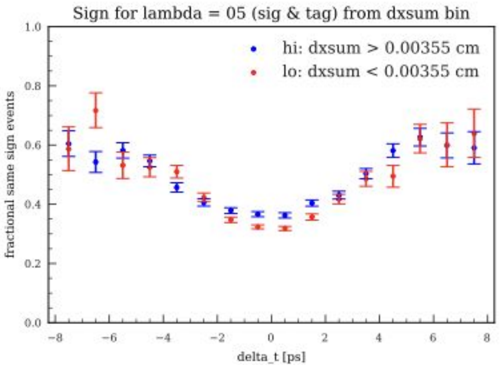


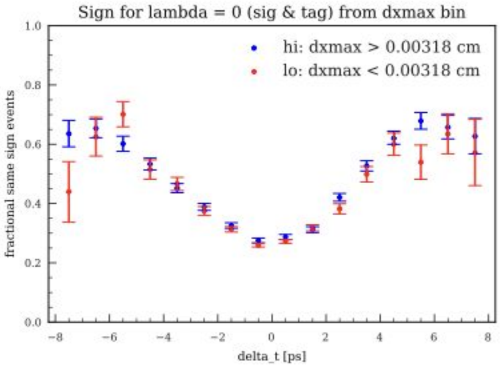
Can we do a binning that equalizes the events in each bin? How much does the dx distribution change within one of these high lifetime bins?

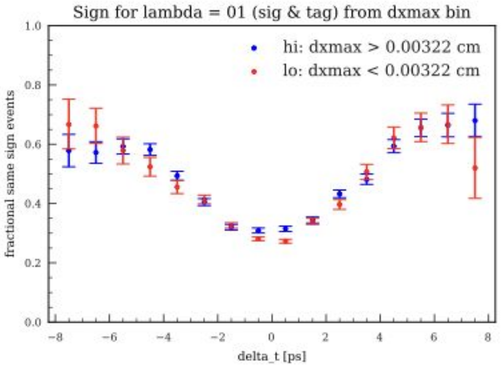
reconstructed variables (100k events)

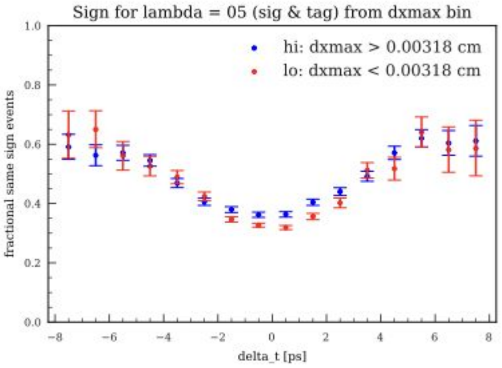




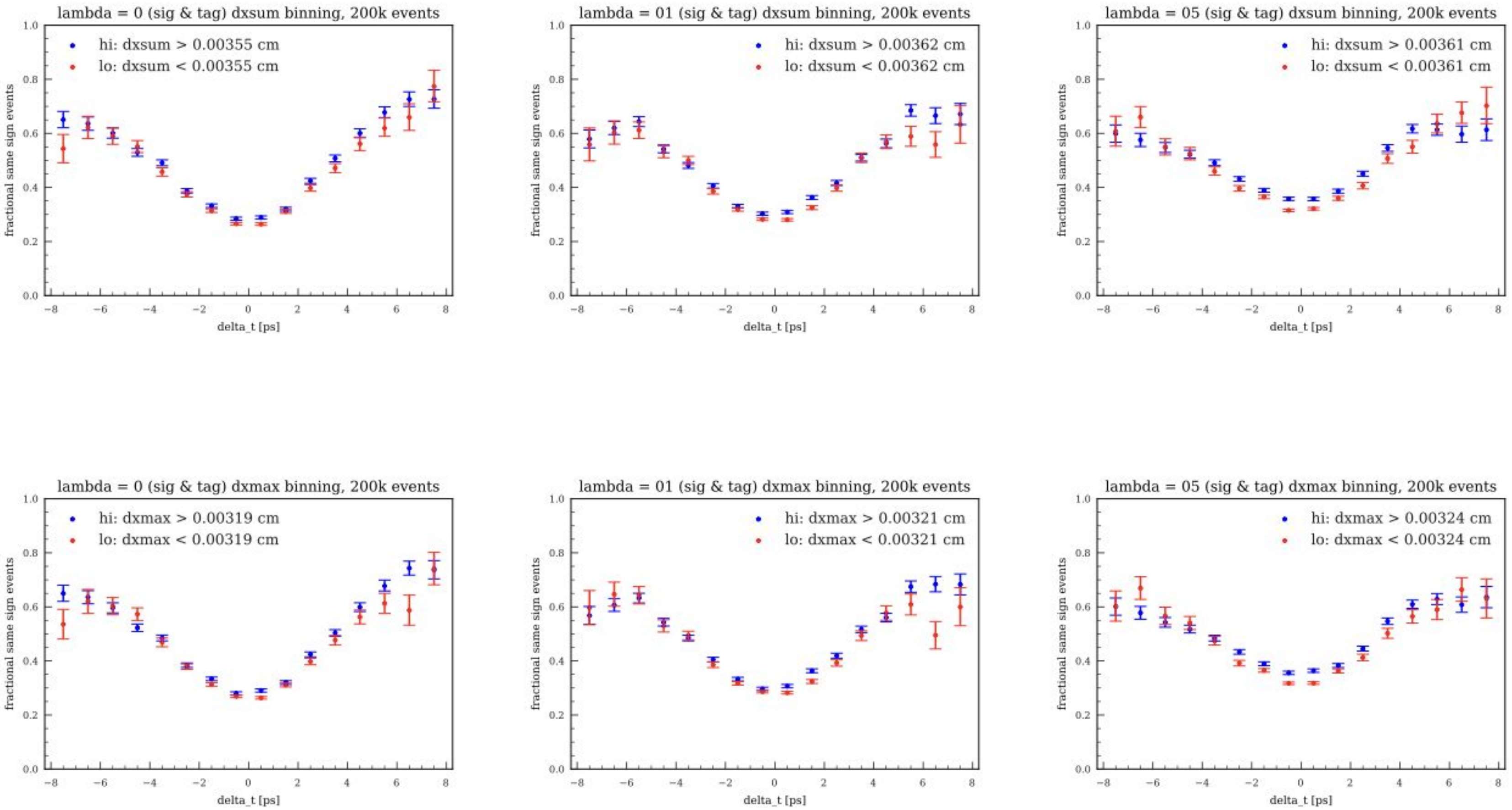




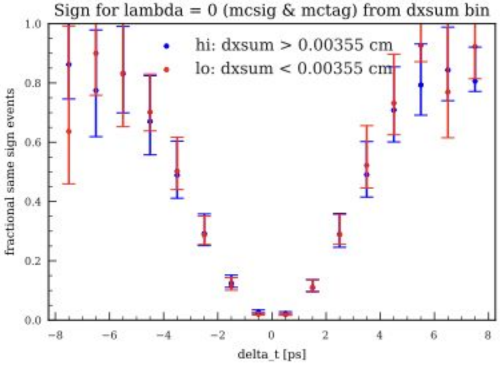


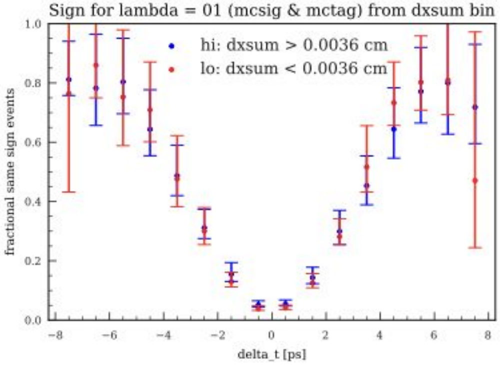


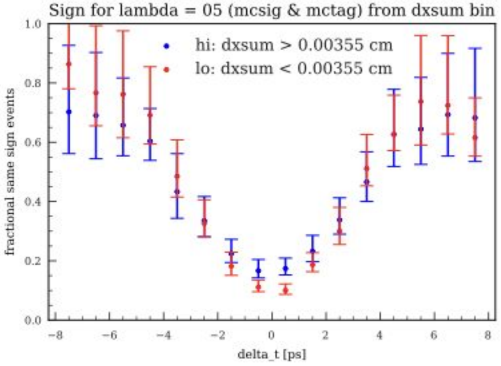
reconstructed variables (200k events) → all r_bins (~72k events)

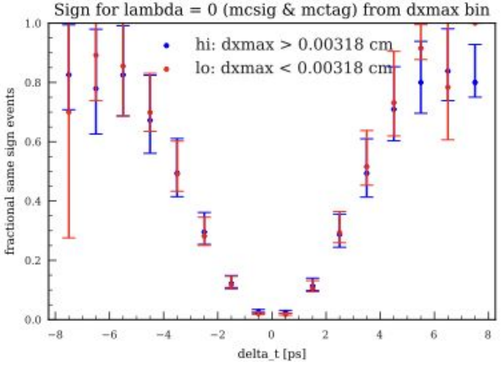


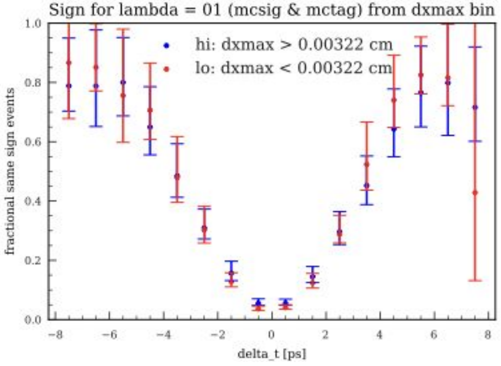
MC variables (errorbars are unfinished) (100k events)

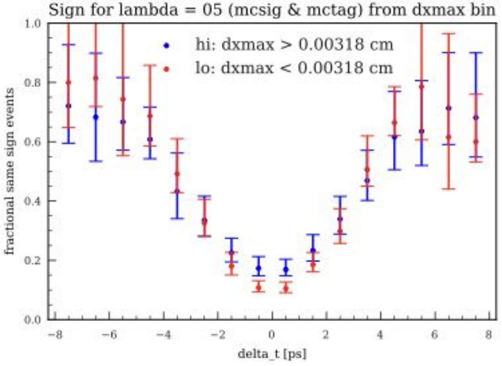




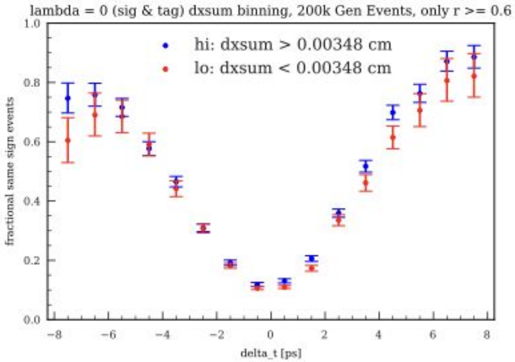


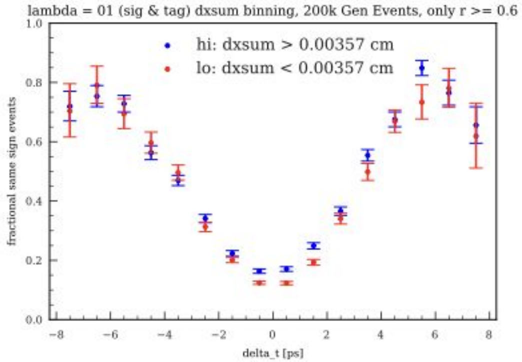


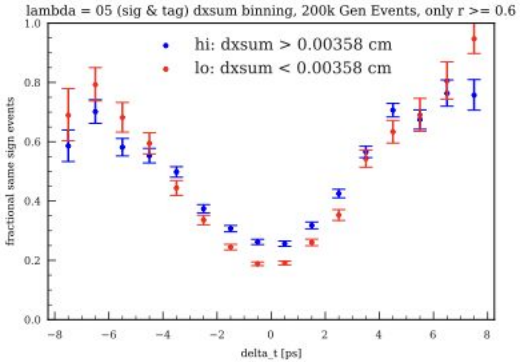




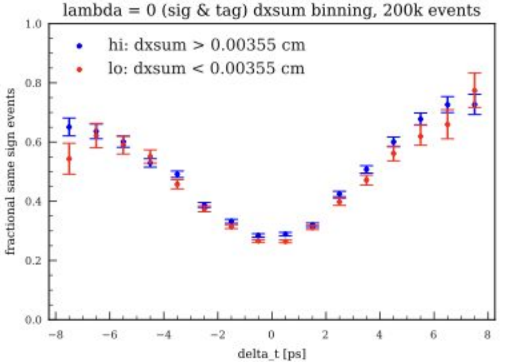
reconstructed variables (200k events) → only r ≥ 0.6 (top 3 out of 7 bins) (~28k events)

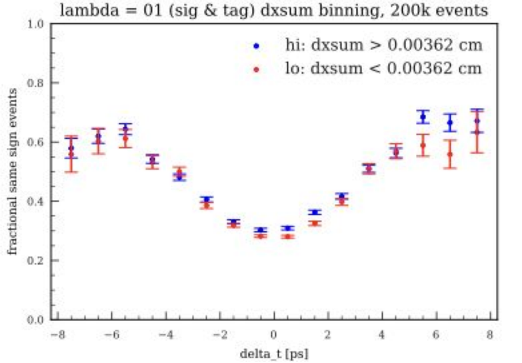


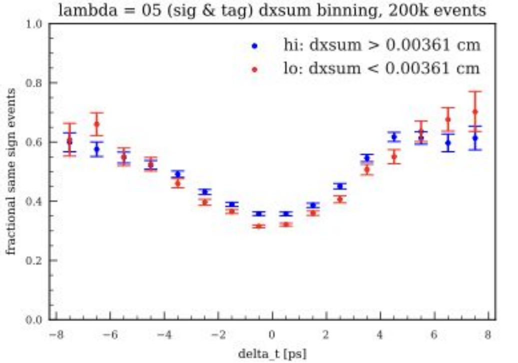




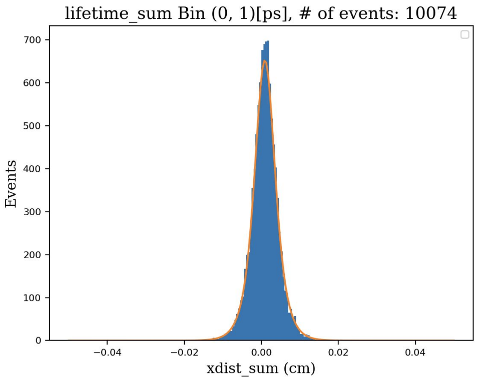
versus all r-bins (~72k events)

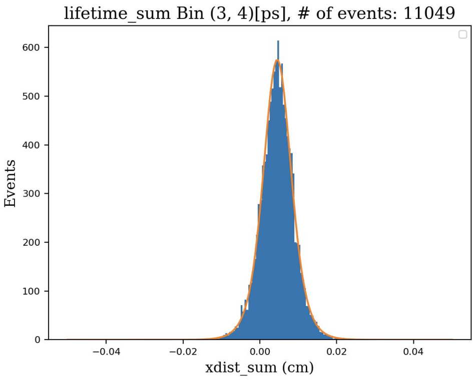


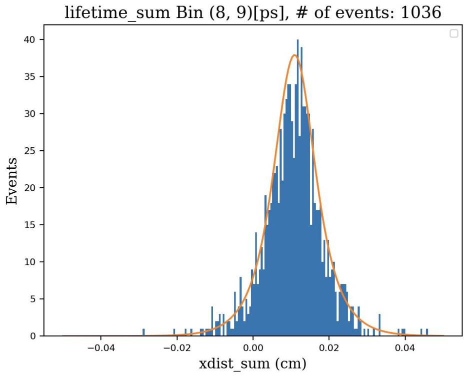


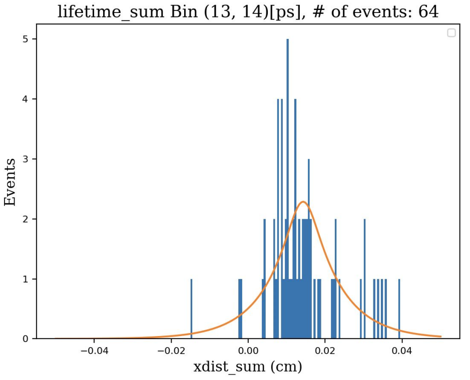


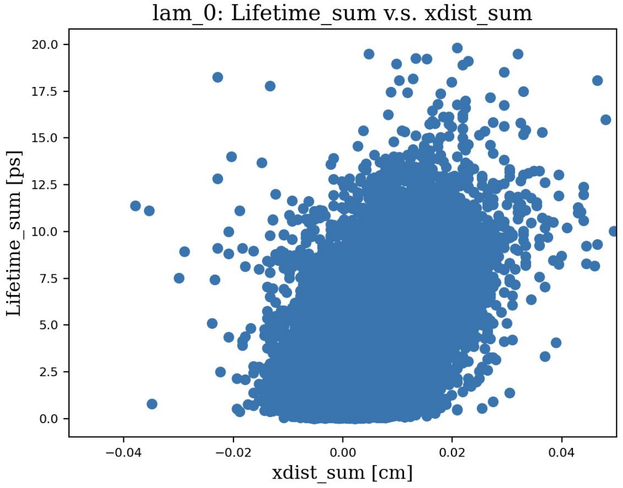
Simulations: 72,798 B pairs simulated

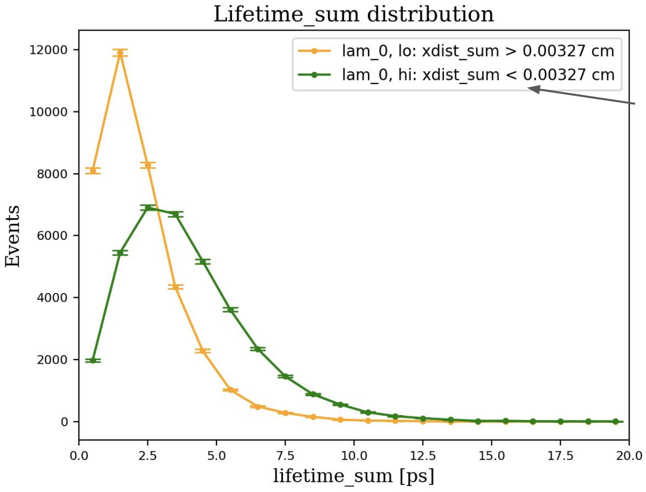






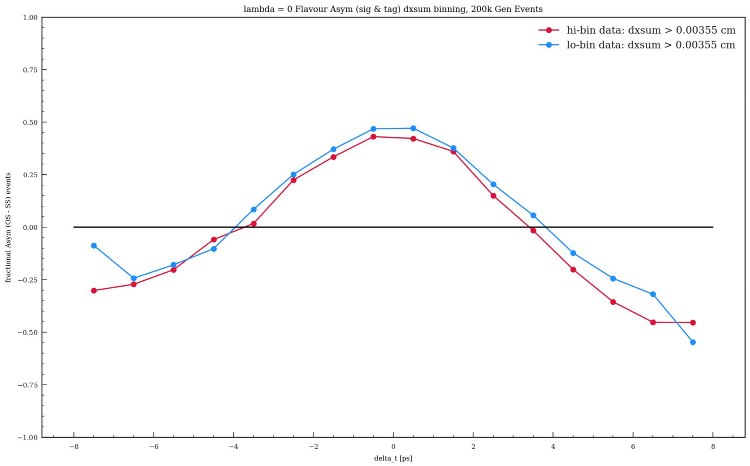


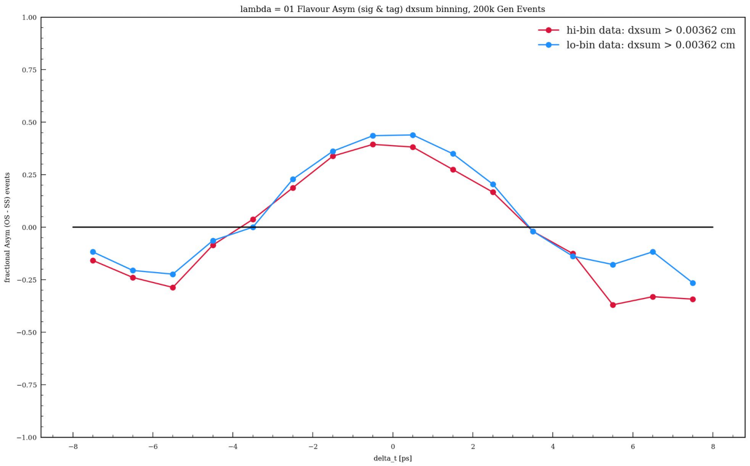




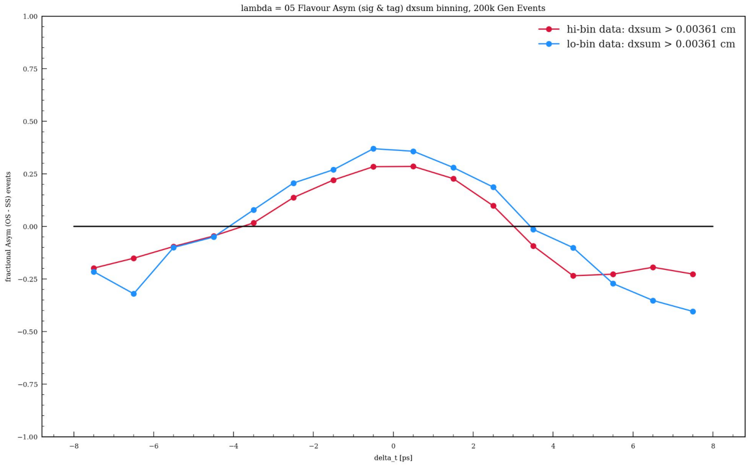
legend is wrong

(Opp - Same) Asymmetry, full data (after recon & vertex cuts)

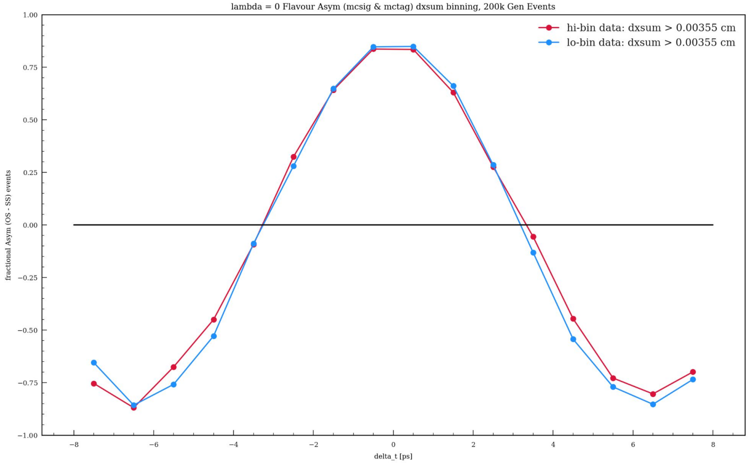


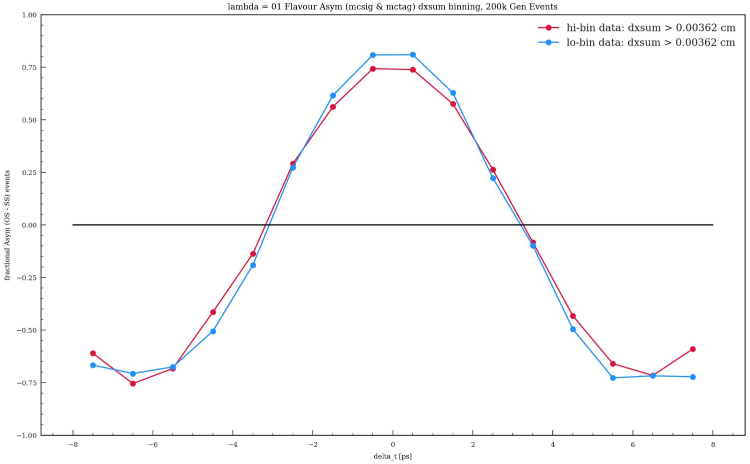


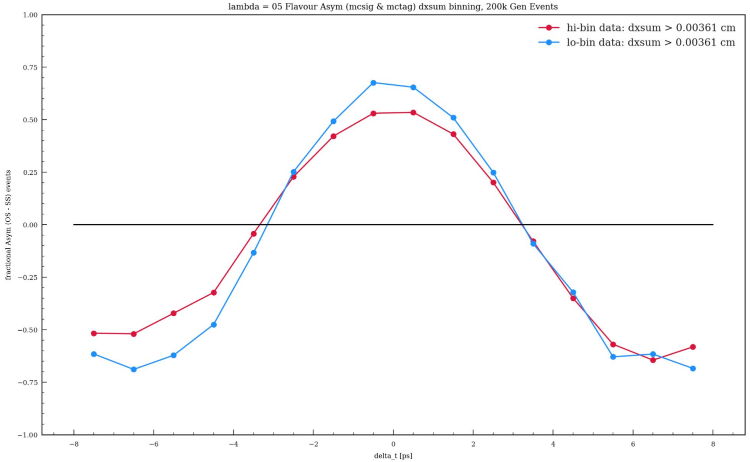
look into what may be causing this…



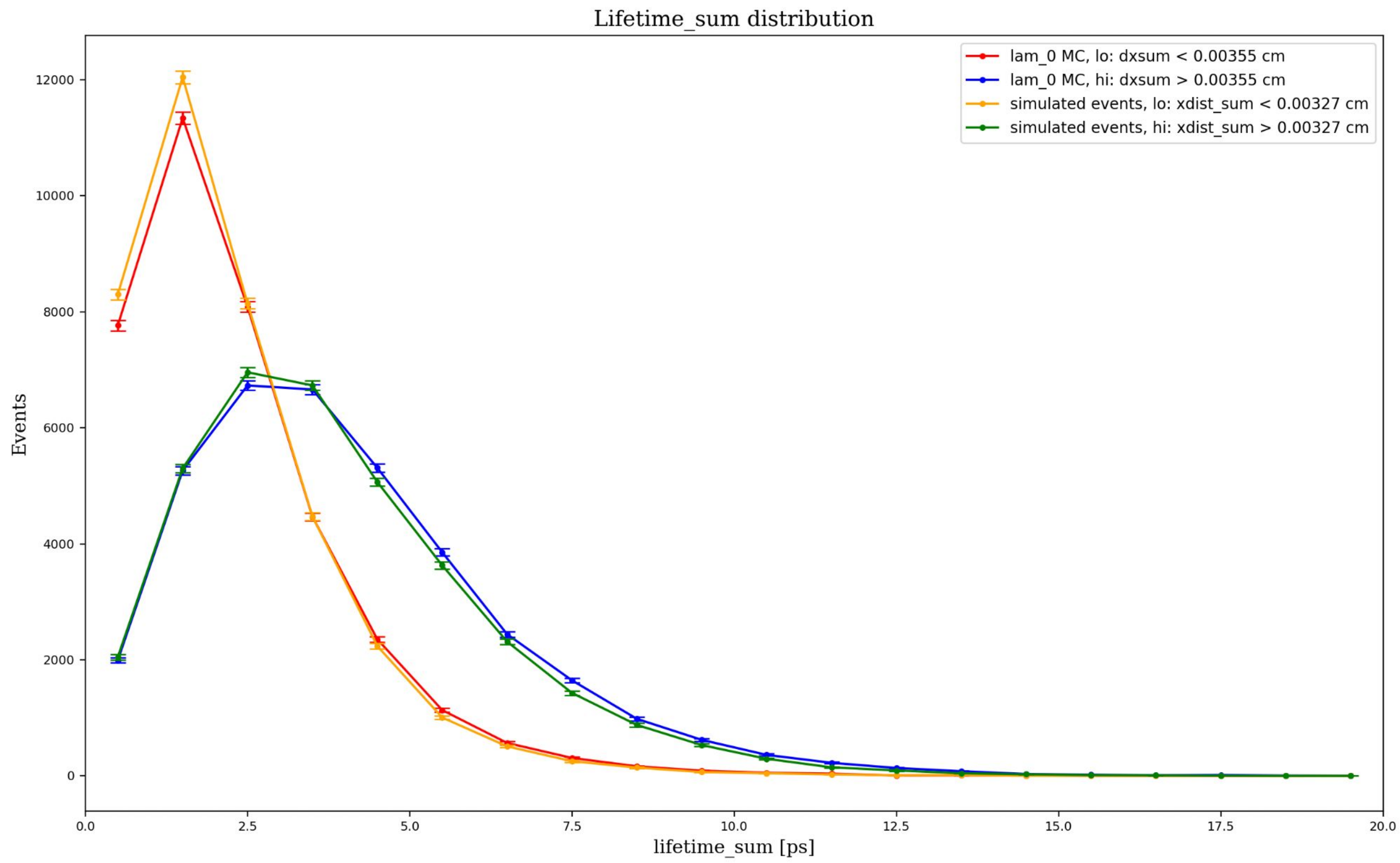
(Opp - Same) Asymmetry, full data (after recon & vertex cuts) using MC variables





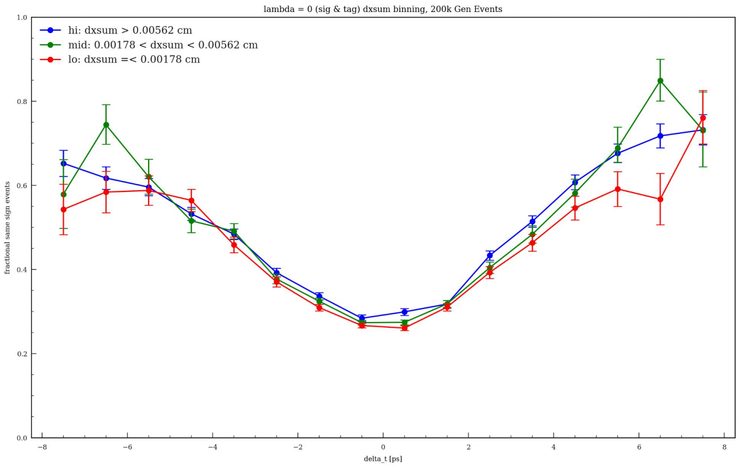


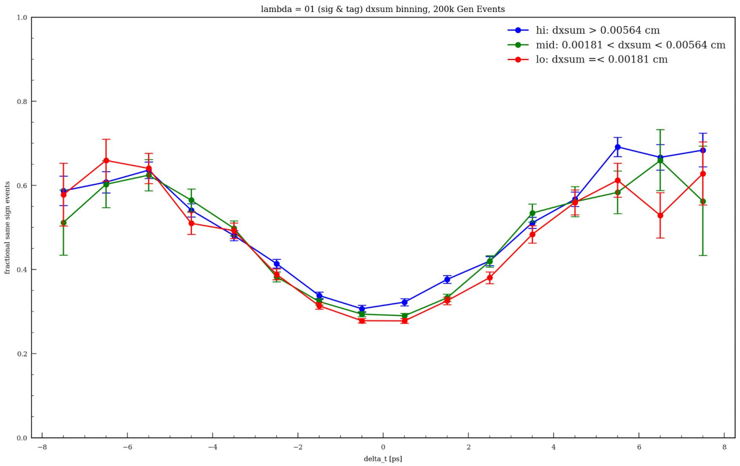
MC data versus simulated data: λ=0

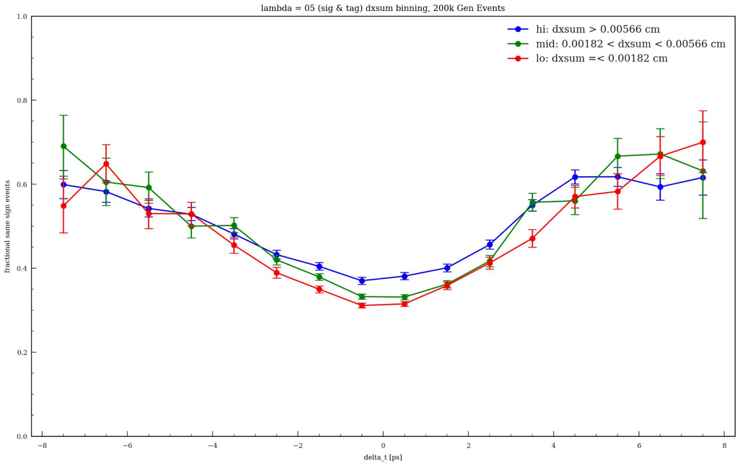


Distributions are similar, but not identical: shape and x_median values

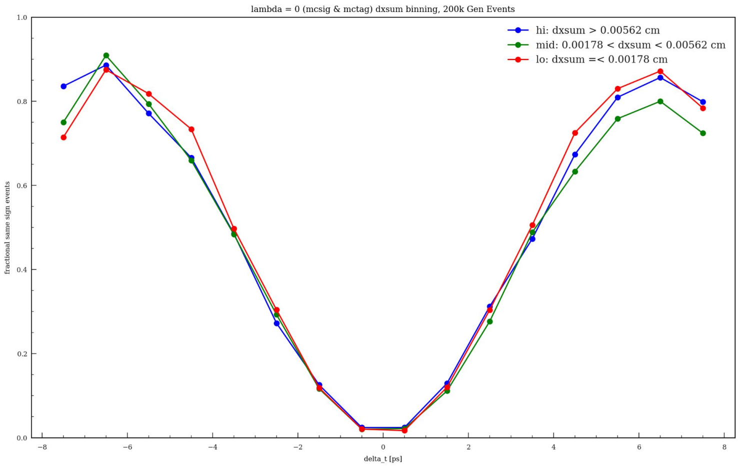
Fractional same flavour (3 dx_sum bins)

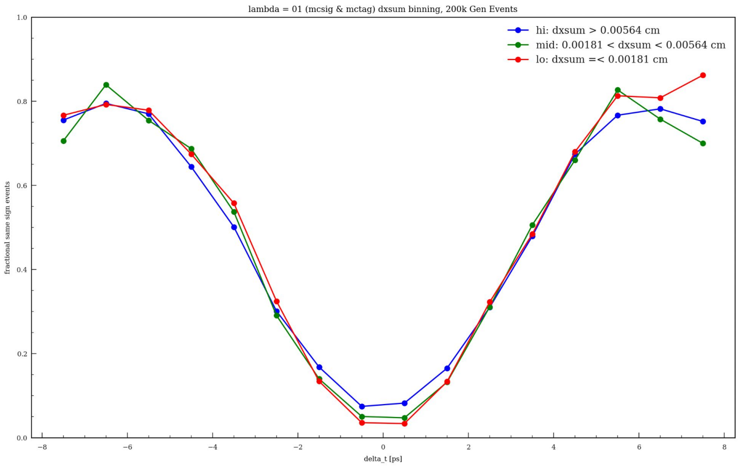


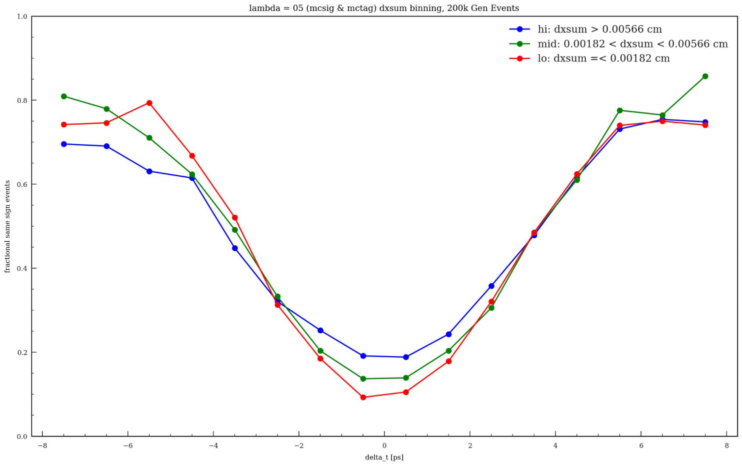




Fractional same flavour (3 dx_sum bins) MC variables







Hand doing χ^2 calculation for same flavour oscillation plots

full data/r >= 0.6 data

want to recheck the calculation??

lam_0

the total chi_sq for deco data lam_0 is: 41.74/27.71

the number of degrees of freedom (# of bins) is: 16

the reduced chi squared for deco data lam_0 is: 2.61/1.73

this is the result from scipy for lam_0 and size 200k using flavour values: 0.644/0.647

lam_01

the total chi_sq for deco data lam_01 is: 52.02/72.58

the number of degrees of freedom (# of bins) is: 16

the reduced chi squared for deco data lam_01 is: 3.25/4.54

this is the result from scipy for lam_01 and size 200k using flavour values: 0.562/0.78

lam_05

the total chi_sq for deco data lam_05 is: 98.98/145.89

the number of degrees of freedom (# of bins) is: 16

the reduced chi squared for deco data lam_05 is: 6.19/9.12

this is the result from scipy for lam_05 and size 200k using flavour values: 0.931/0.95

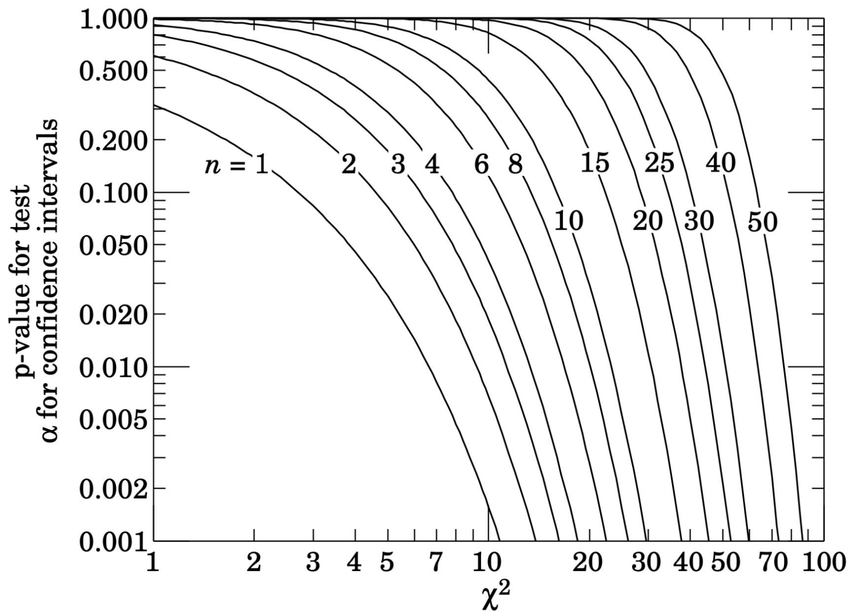


Figure 40.1: One minus the χ^2 cumulative distribution, $1 - F(\chi^2; n)$, for n degrees of freedom. This gives the p-value for the χ^2 goodness-of-fit test as well as one minus the coverage probability for confidence regions (see Sec. 40.4.2.2).