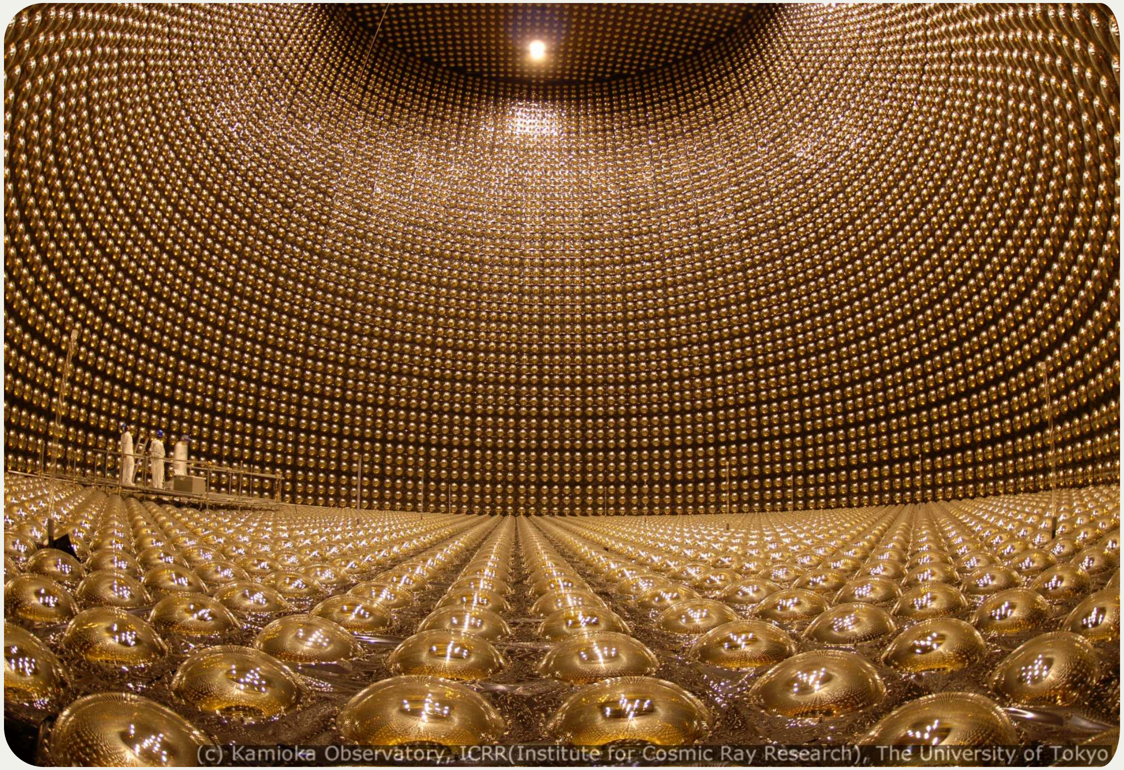
Event Reconstruction of the Hyper-Kamiokande Detector

Ben Carew - IMAPP - 29/09/25

Supervisors: Dr Vladimir Gligorov and Dr Mathieu Guigue

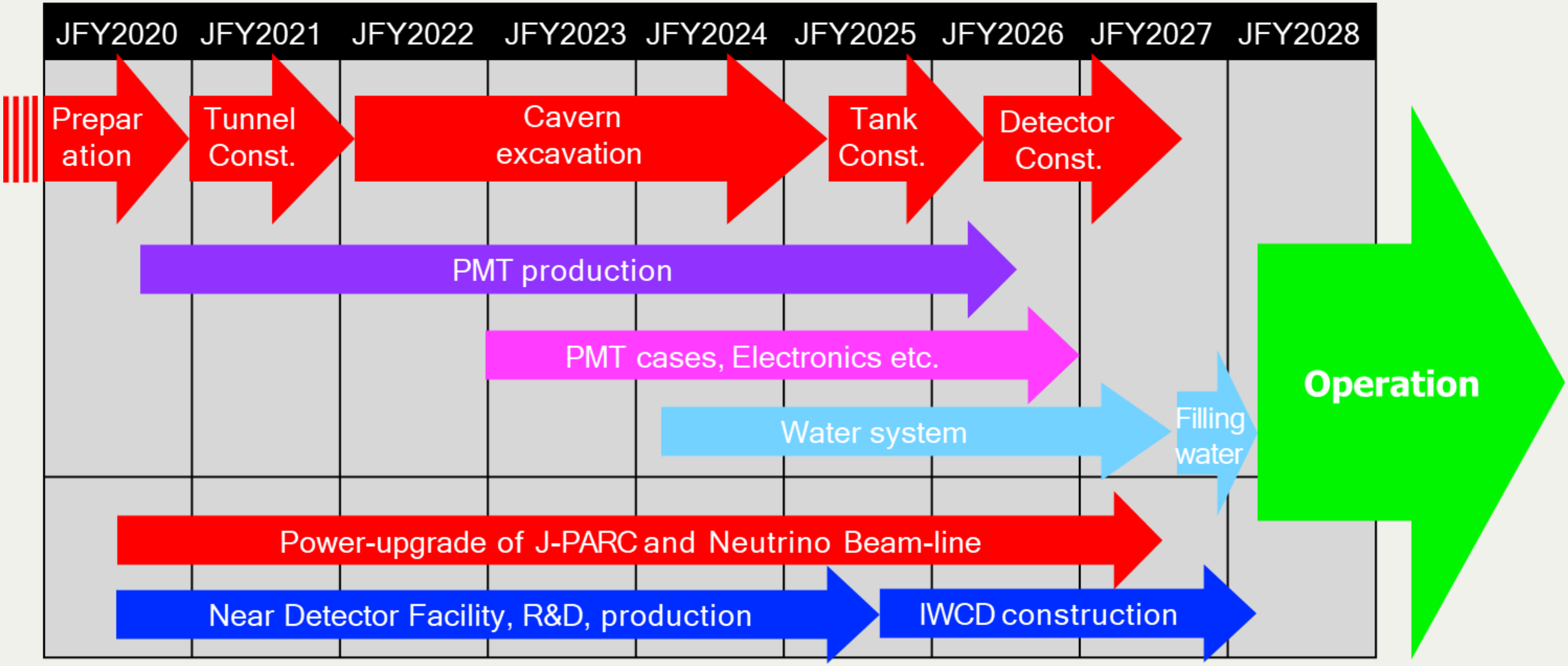
Hyper-Kamiokande



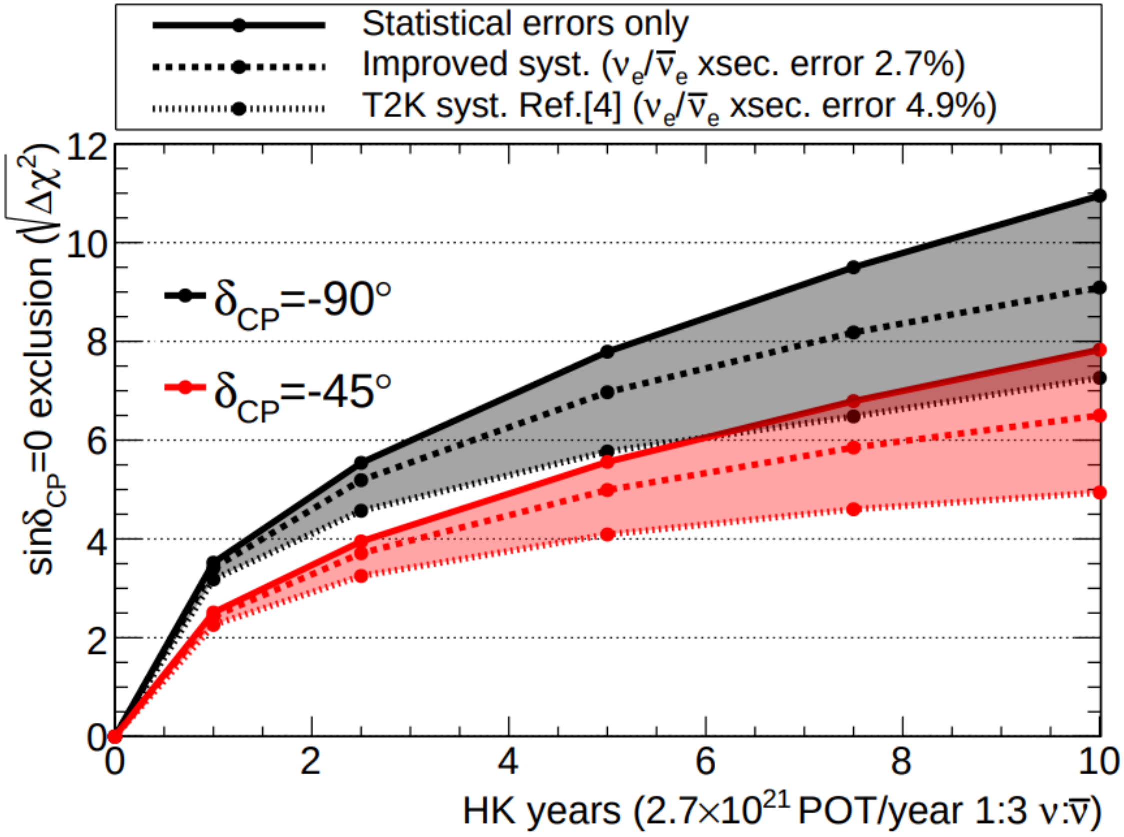
SuperK, the predecessor of HyperK

- Next-generation **neutrino** detector
- Water Cherenkov - collects radiation from super-luminal particles
- 20,000 photomultiplier tubes with **charge** and **time** data
- Need to **reconstruct** physical events from the PMT data

Hyper-Kamiokande



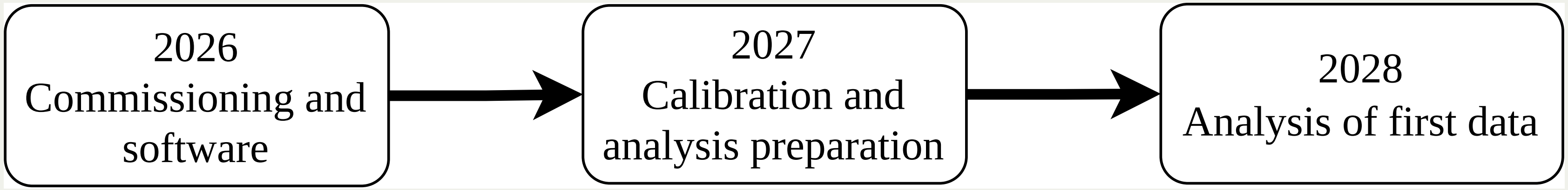
Motivation



- **Charge-Parity Violation** explains matter-antimatter asymmetry in the universe
- HyperK will have **world-leading sensitivity** to CPV in neutrinos
- Fast reconstruction directly **reduces systematic errors** for this analysis

Motivation

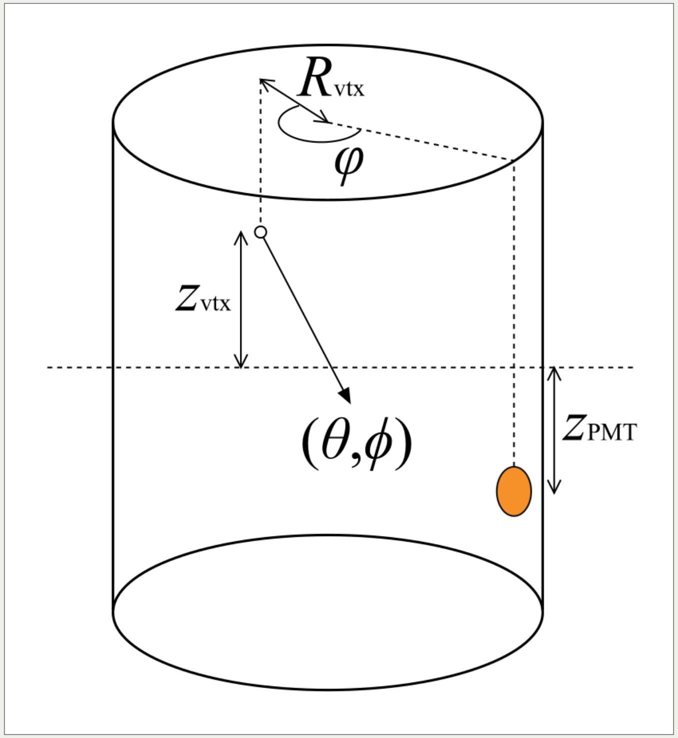
- Event reconstruction needs to be **efficient** and **robust**
  - Unexpected deviations from simulated data and backgrounds
  - Malfunctions and unpredictable detector behaviour
- **Faster reconstruction** $\rightarrow$ **larger MC sample** $\rightarrow$ **greater discovery potential**
  - Focus on CPV analysis, but will benefit HyperK's broad physics goals
- Crucial to optimise reconstruction before calibration in **July 2027**



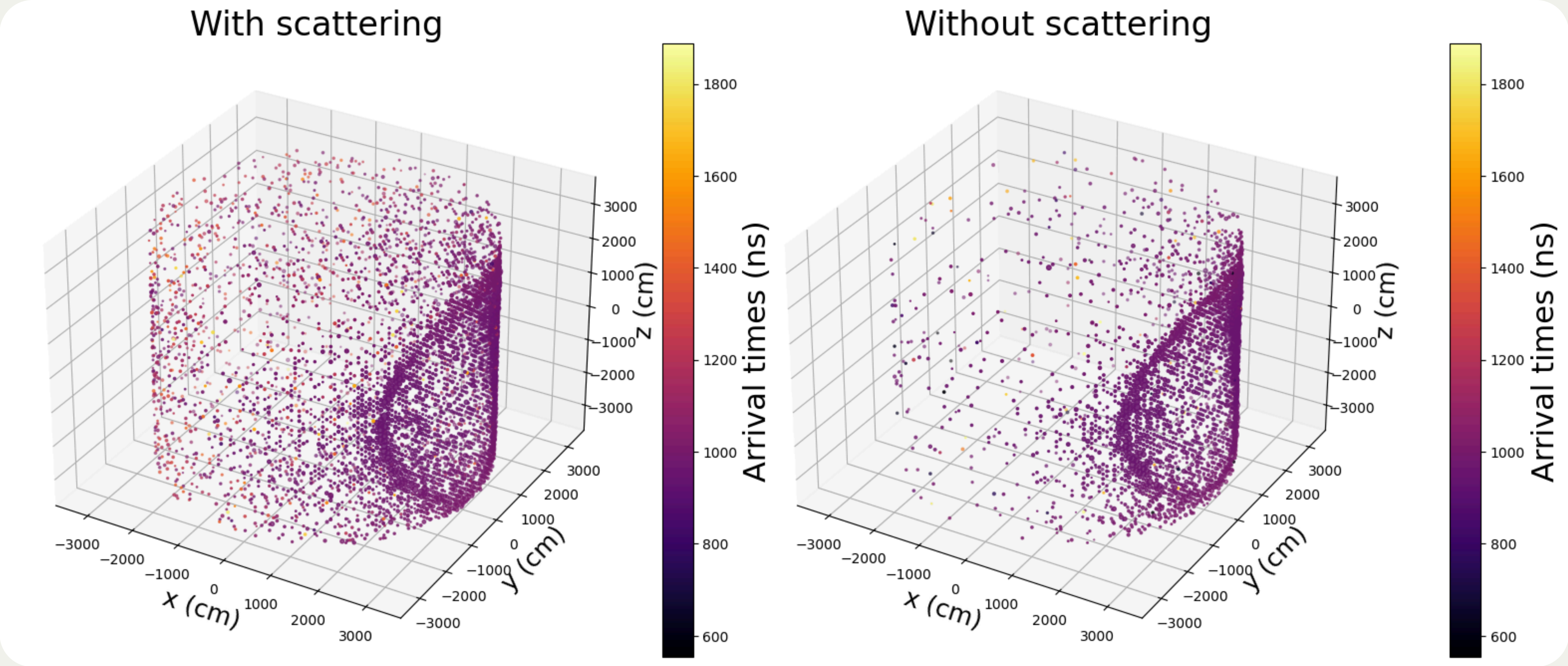
Scattered Light Table

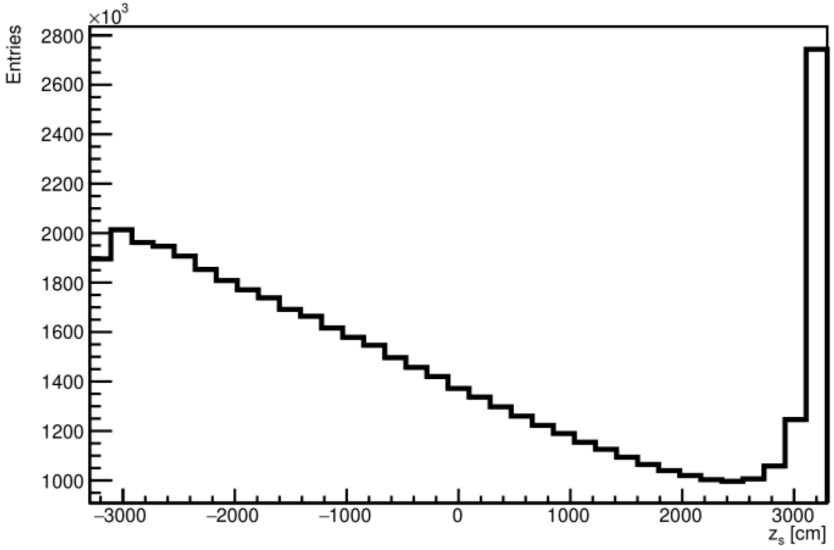
$$A(s) = A(x_{\mathrm{PMT}}, z_{\mathrm{vtx}}, R_{\mathrm{vtx}}, \varphi, \theta, \phi) = \frac{\mathrm{d}\mu^{\mathrm{sct}}}{\mathrm{d}\mu^{\mathrm{iso,sct}}}$$

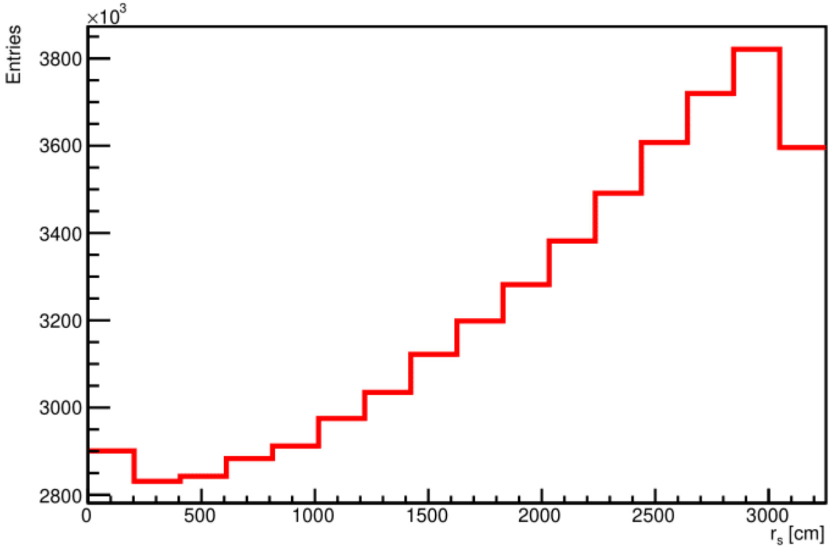
- Accounts for predicted charge from light scattered in water and reflected
- Table requires computationally taxing 6D interpolation at each point s along the track - ~35% of total runtime
- Optimisable by studying the physics behind the table's structure

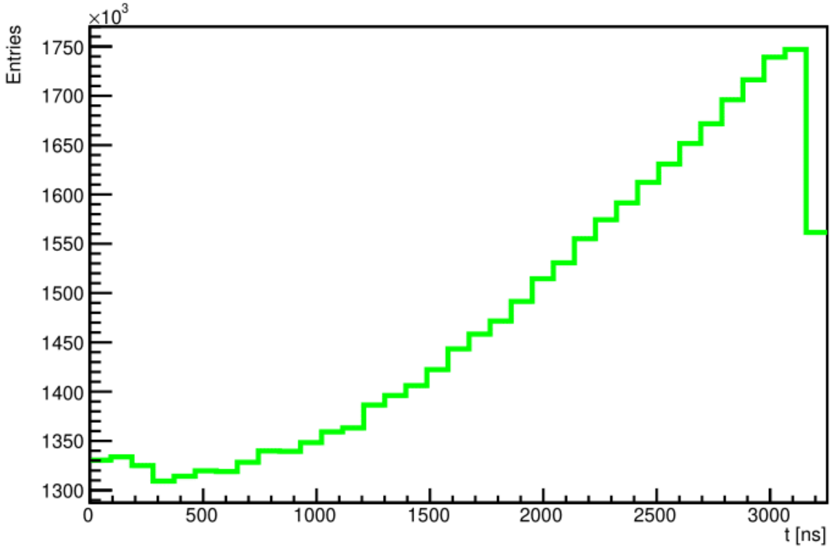


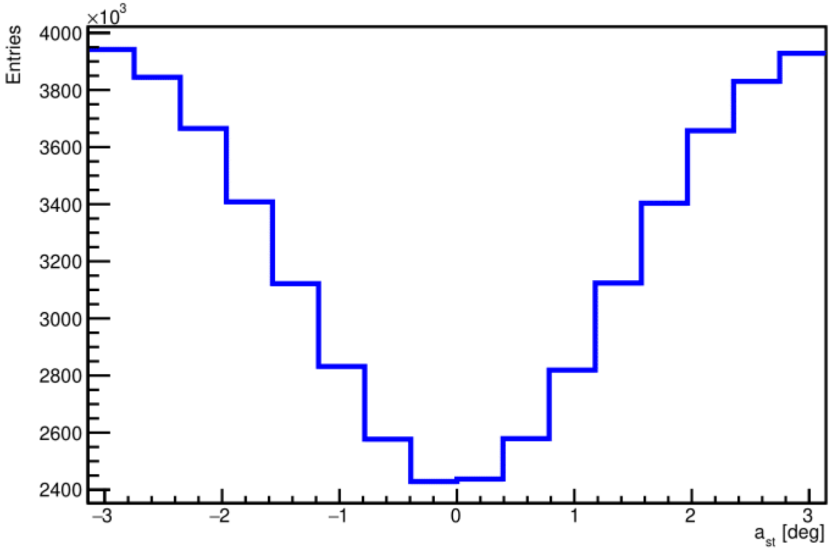
Scattered Light Table

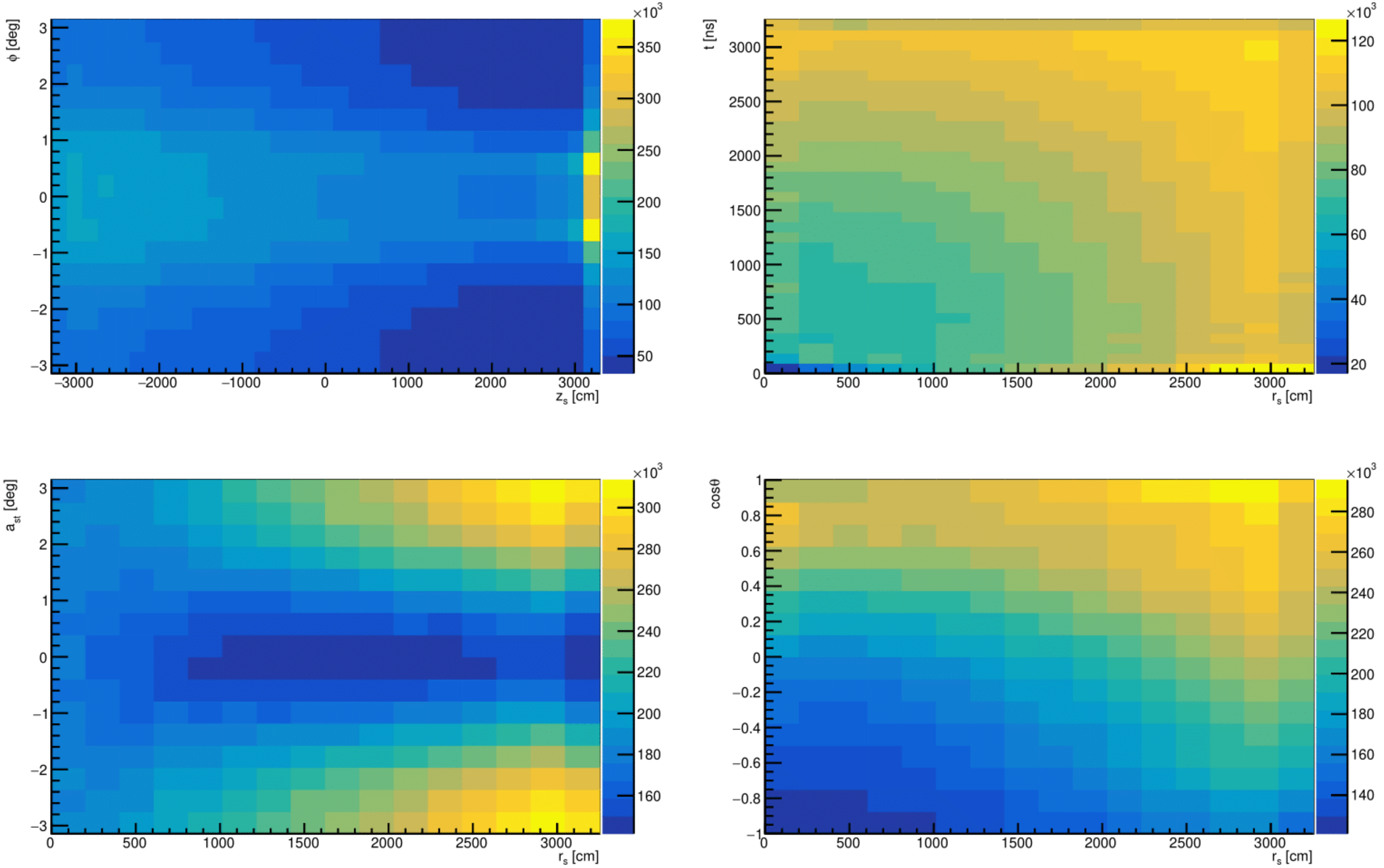



Entries
×10^3
2800
2600
2400
2200
2000
1800
1600
1400
1200
1000
−3000
−2000
−1000
0
1000
2000
3000
z_s [cm]


Entries
×10^3
3800
3600
3400
3200
3000
2800
0
500
1000
1500
2000
2500
3000
r_s [cm]

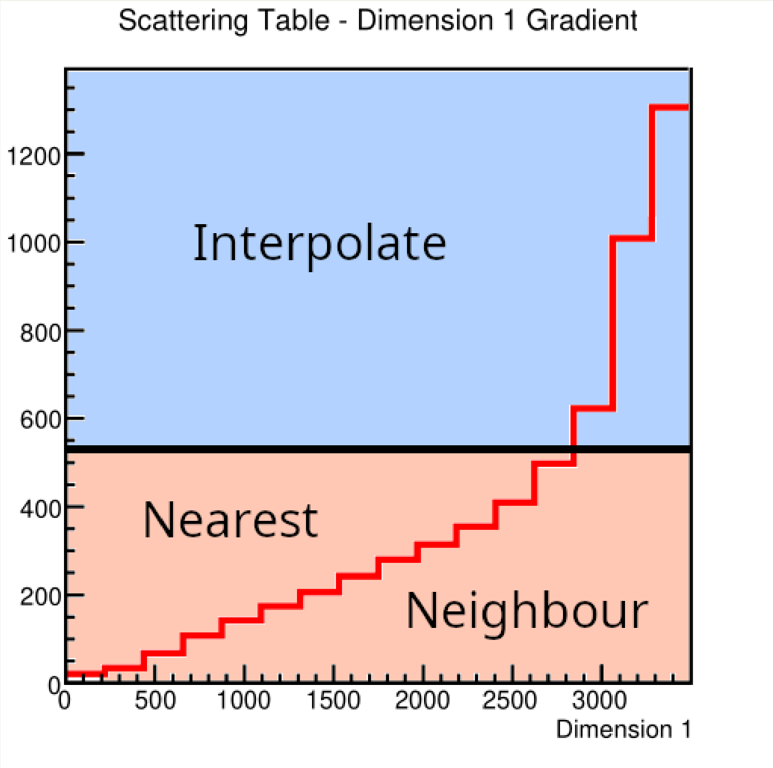

Entries
×10^3
1750
1700
1650
1600
1550
1500
1450
1400
1350
1300
0
500
1000
1500
2000
2500
3000
t [ns]


Entries
×10^3
4000
3800
3600
3400
3200
3000
2800
2600
2400
−3
−2
−1
0
1
2
3
a_st [deg]



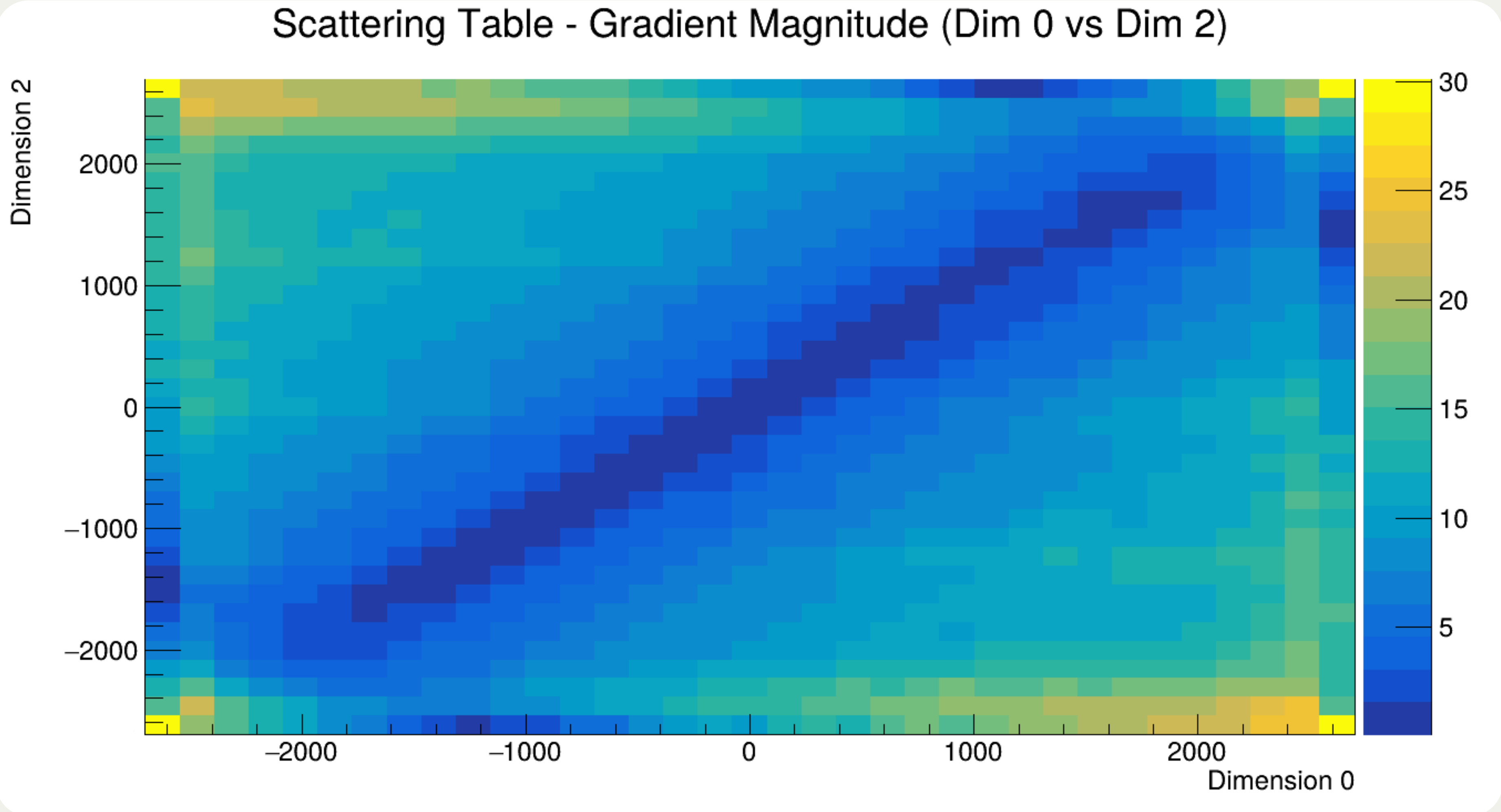
φ [deg]
z_s [cm]
×10^3
t [ns]
r_s [cm]
×10^3
a_st [deg]
r_s [cm]
×10^3
cosθ
r_s [cm]
×10^3

Gradient Threshold Optimisation

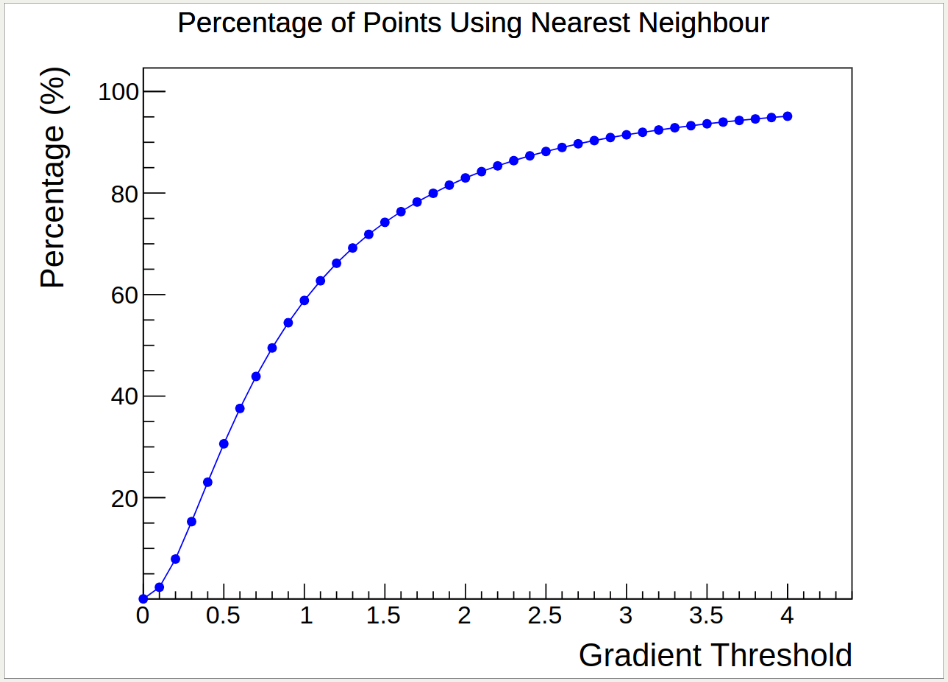


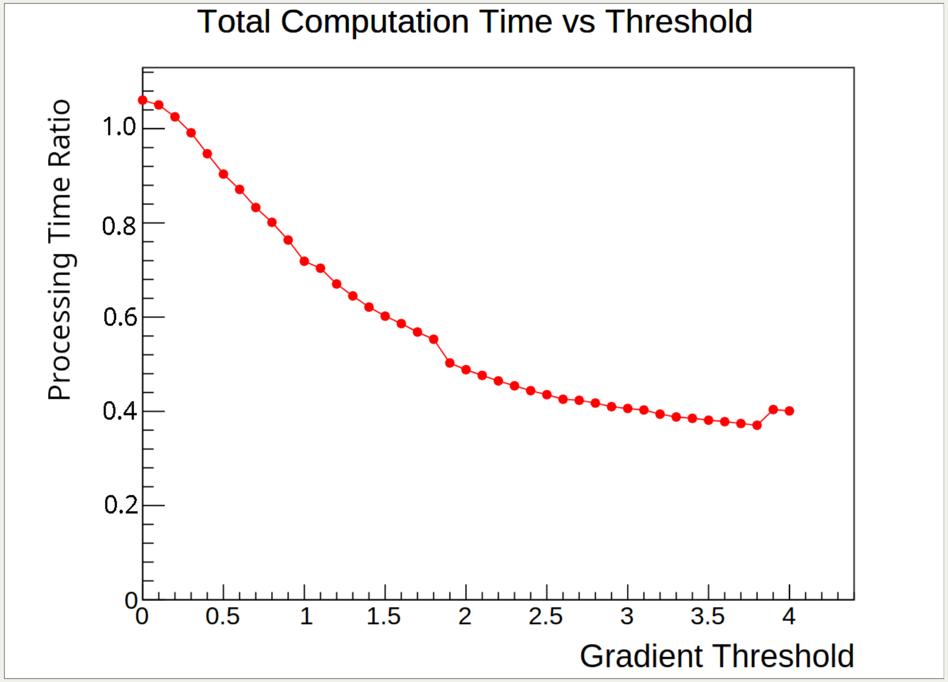
- Calculate 6D gradient between each table value
- Apply a threshold:
  - Above -> interpolate
  - Below -> nearest neighbour
- Find highest threshold value before error becomes large

Gradient Threshold Optimisation

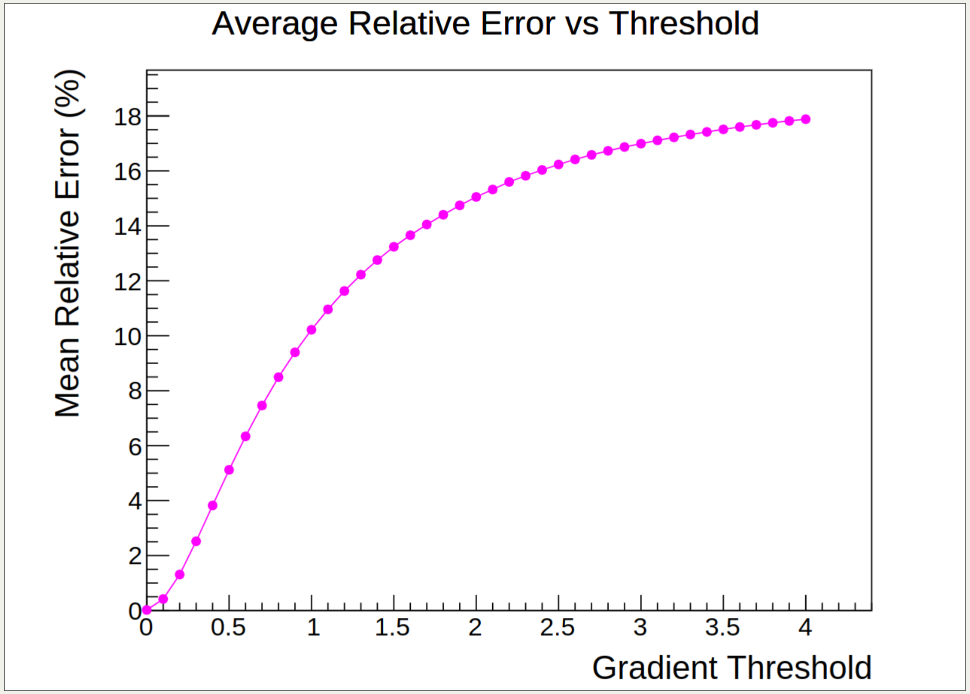


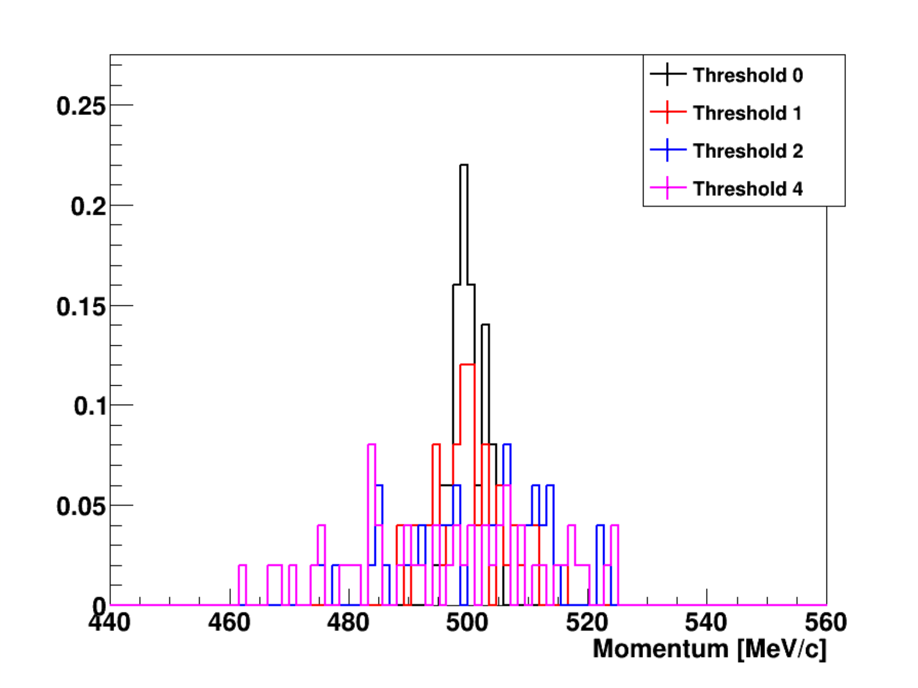
Gradient Threshold Results



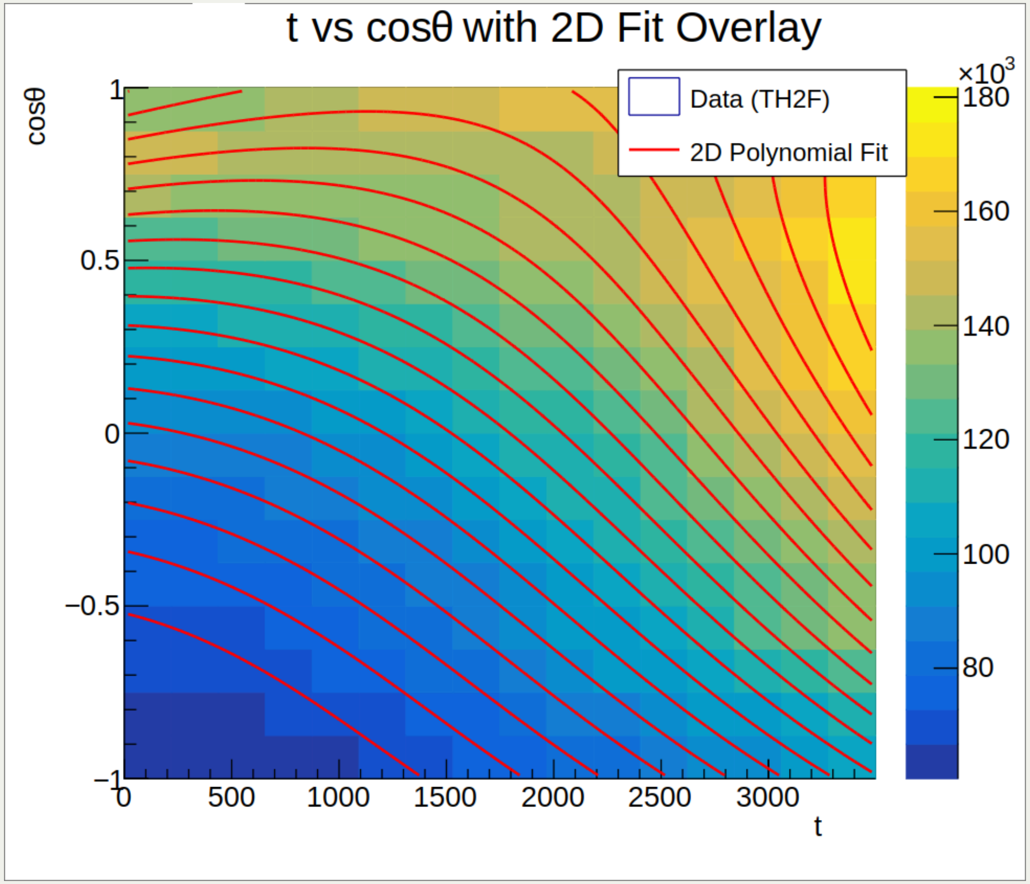


Gradient Threshold Results

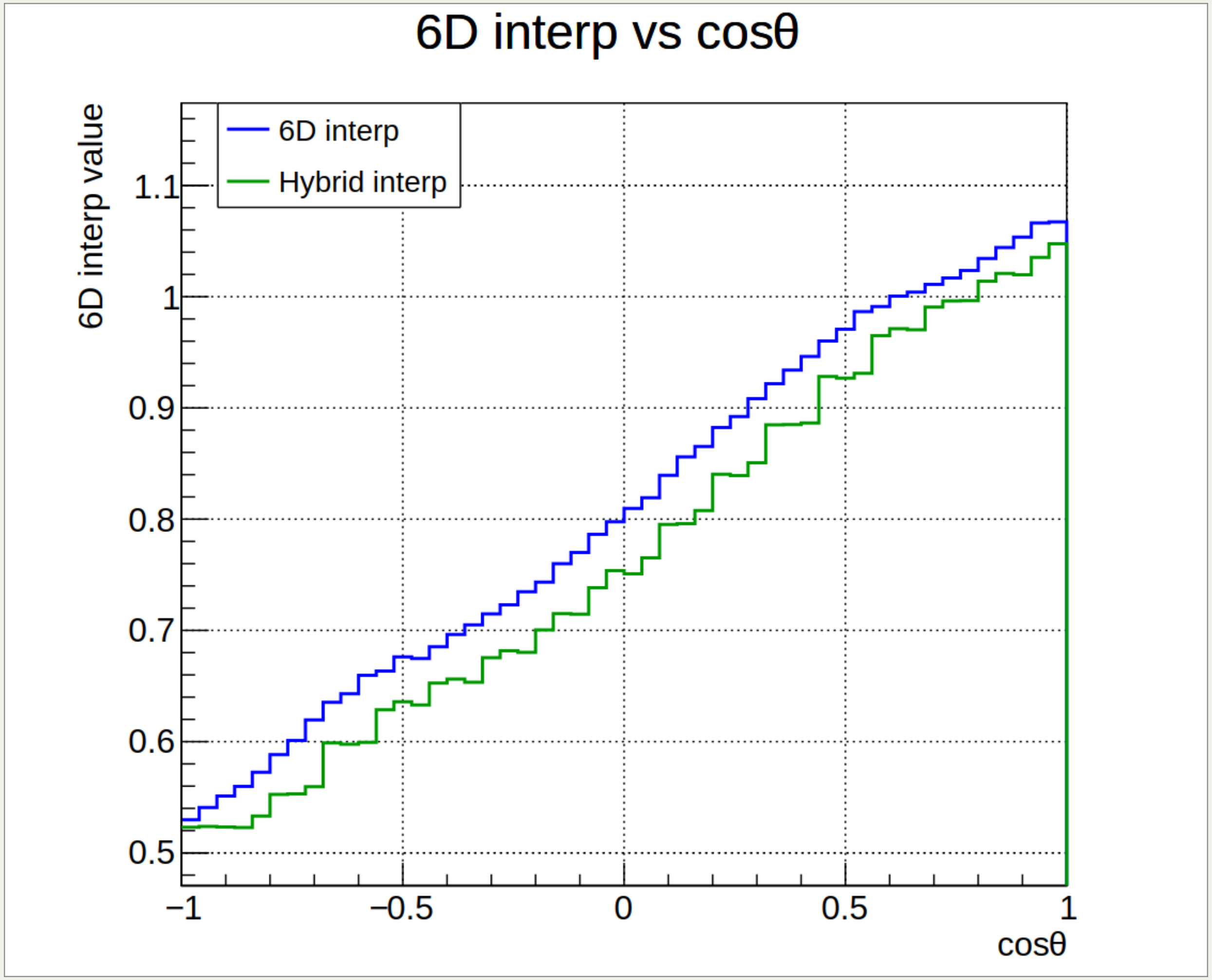


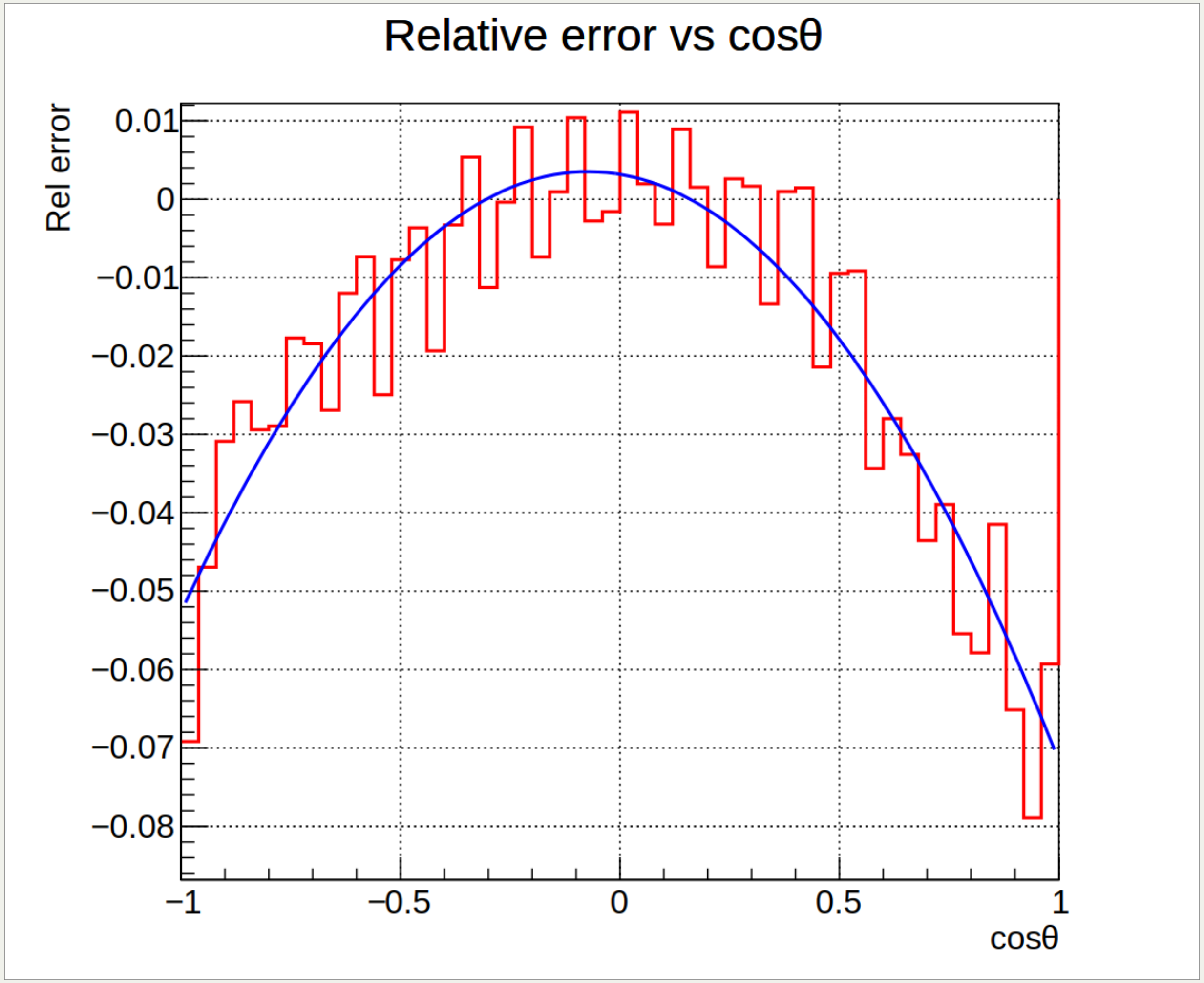


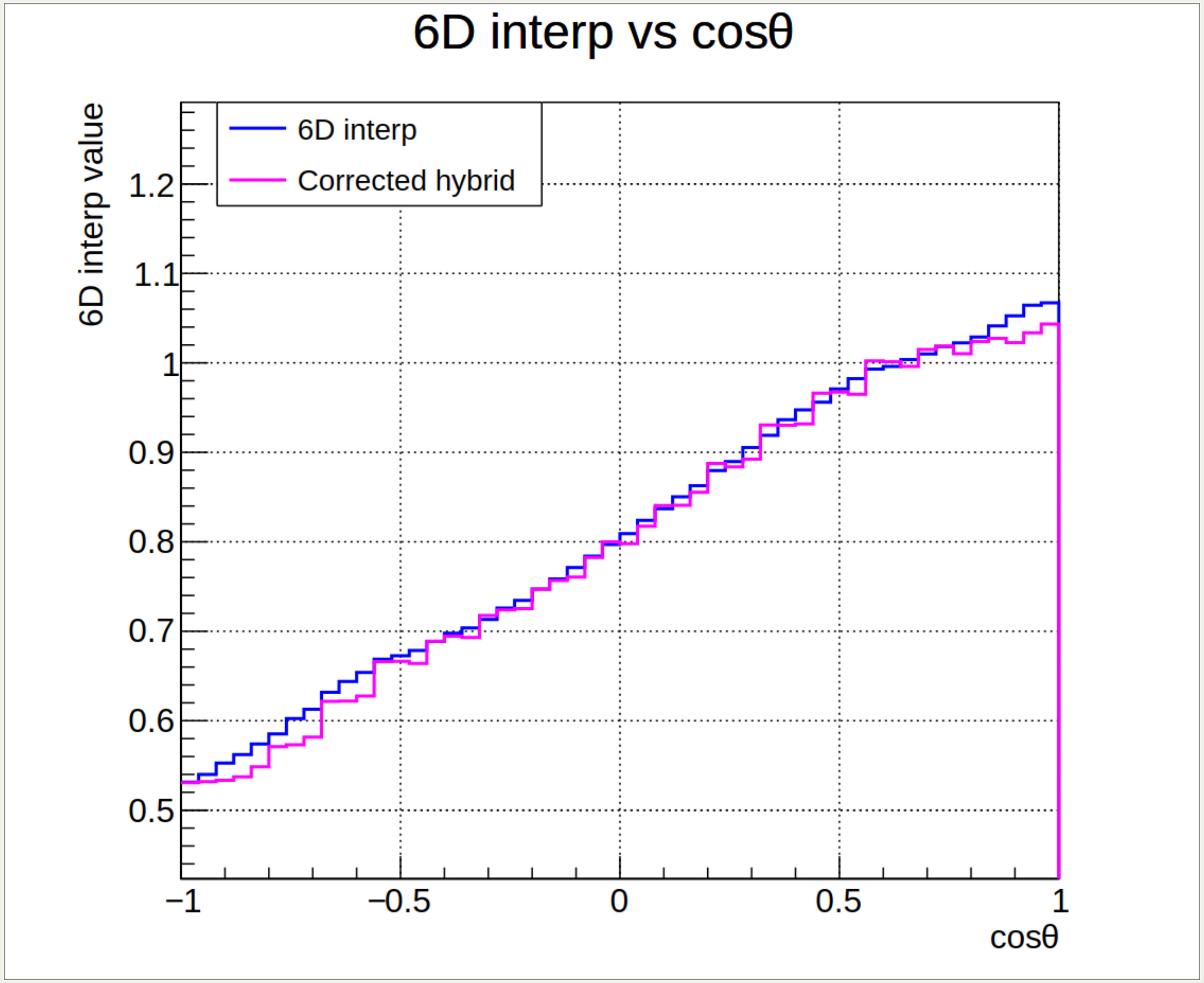
Parametrisation Optimisation

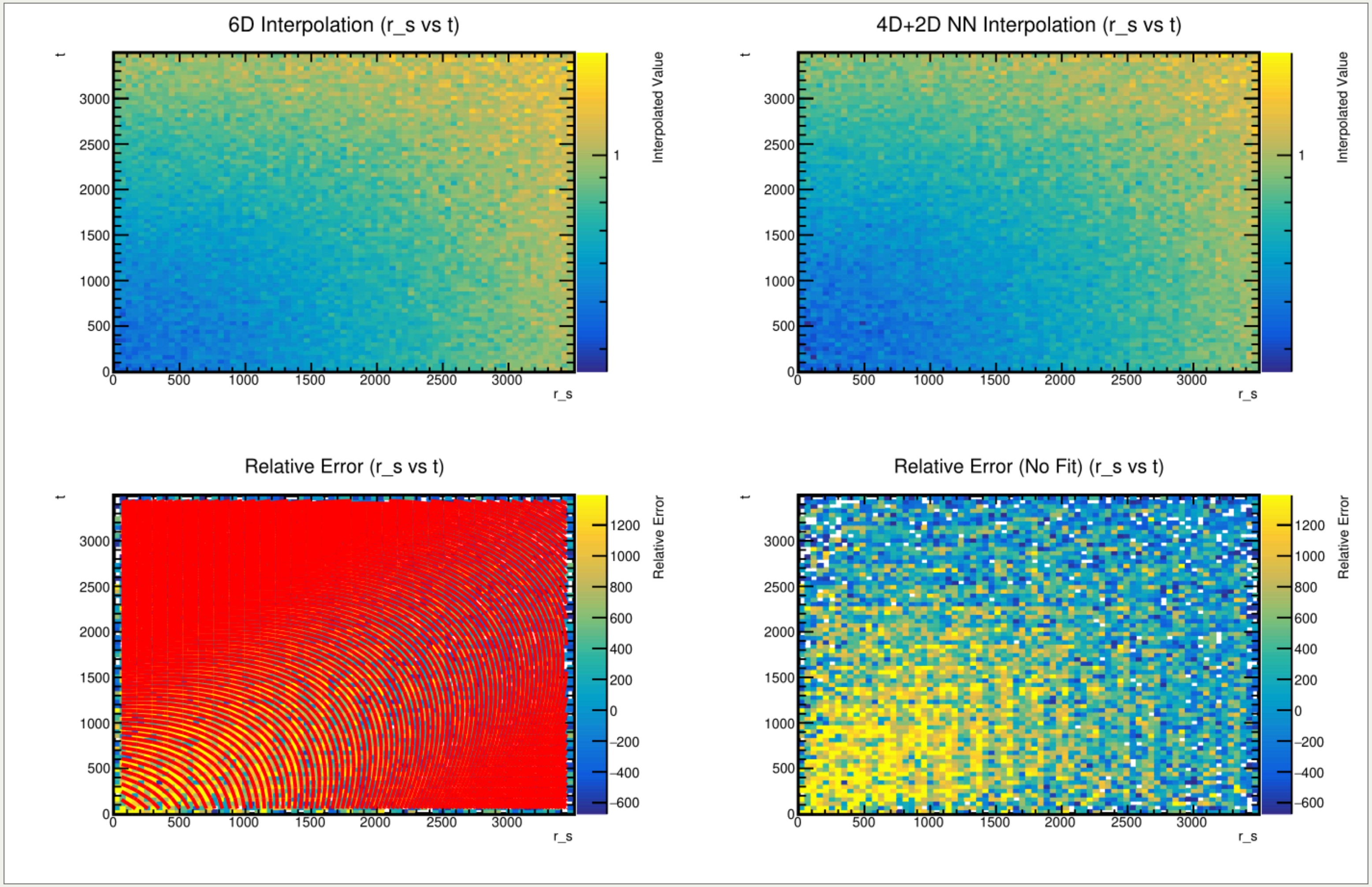


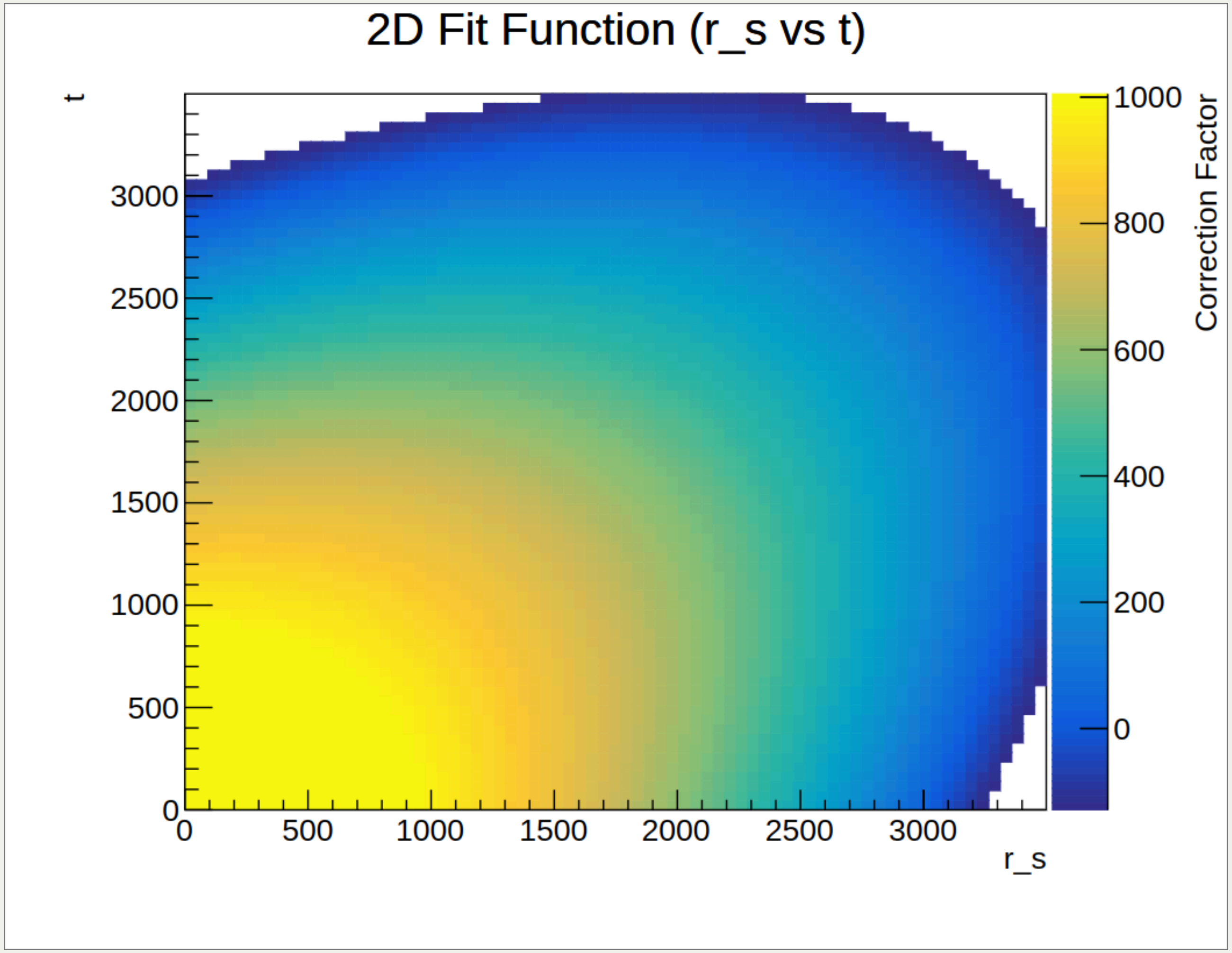
- Regions of the scattering table can be well-fit by analytical functions
- Use physically-motivated relationships between dimensions to reduce interpolation complexity

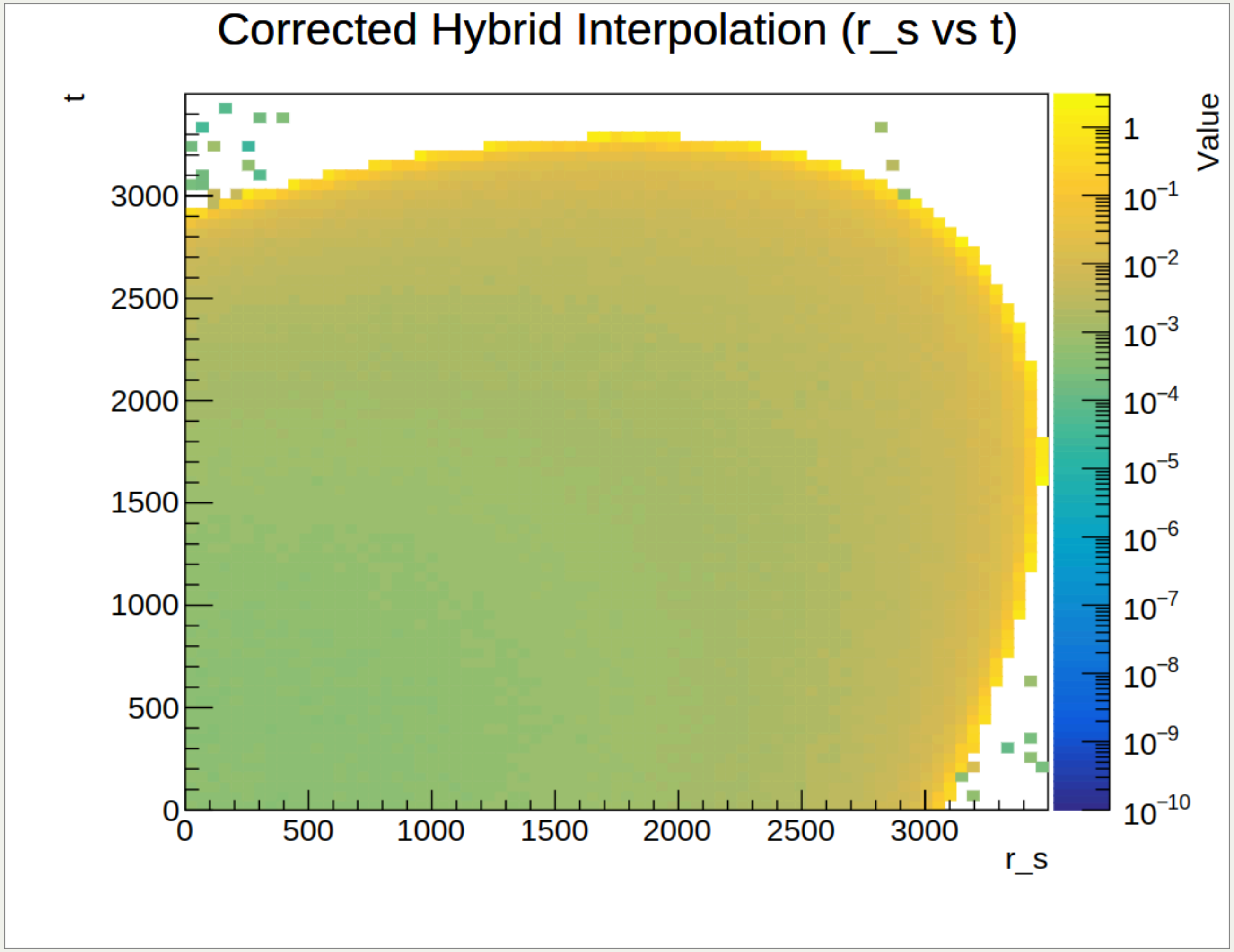

6D interp vs cosθ
6D interp
Hybrid interp
6D interp value
1.1
1
0.9
0.8
0.7
0.6
0.5
−1
−0.5
0
0.5
1
cosθ


Relative error vs cosθ
Rel error
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.05
−0.06
−0.07
−0.08
−1
−0.5
0
0.5
1
cosθ

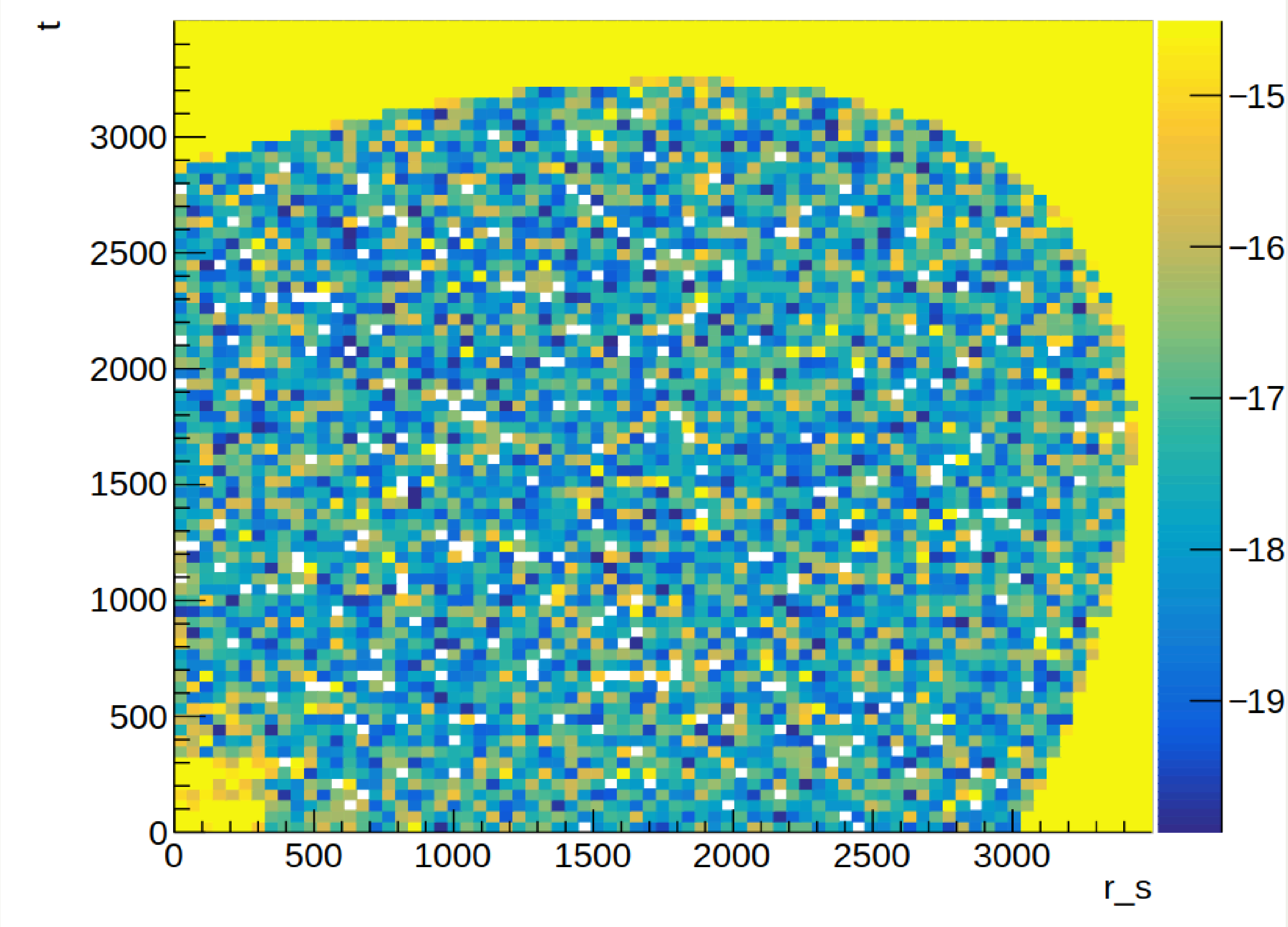

6D interp vs cosθ
6D interp
Corrected hybrid
6D interp value
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
−1
−0.5
0
0.5
1
cosθ


6D Interpolation (r_s vs t)
4D+2D NN Interpolation (r_s vs t)
Relative Error (r_s vs t)
Relative Error (No Fit) (r_s vs t)
Interpolated Value
Relative Error
r_s
t
0
500
1000
1500
2000
2500
3000
1
1200
800
600
400
200
-200
-400
-600

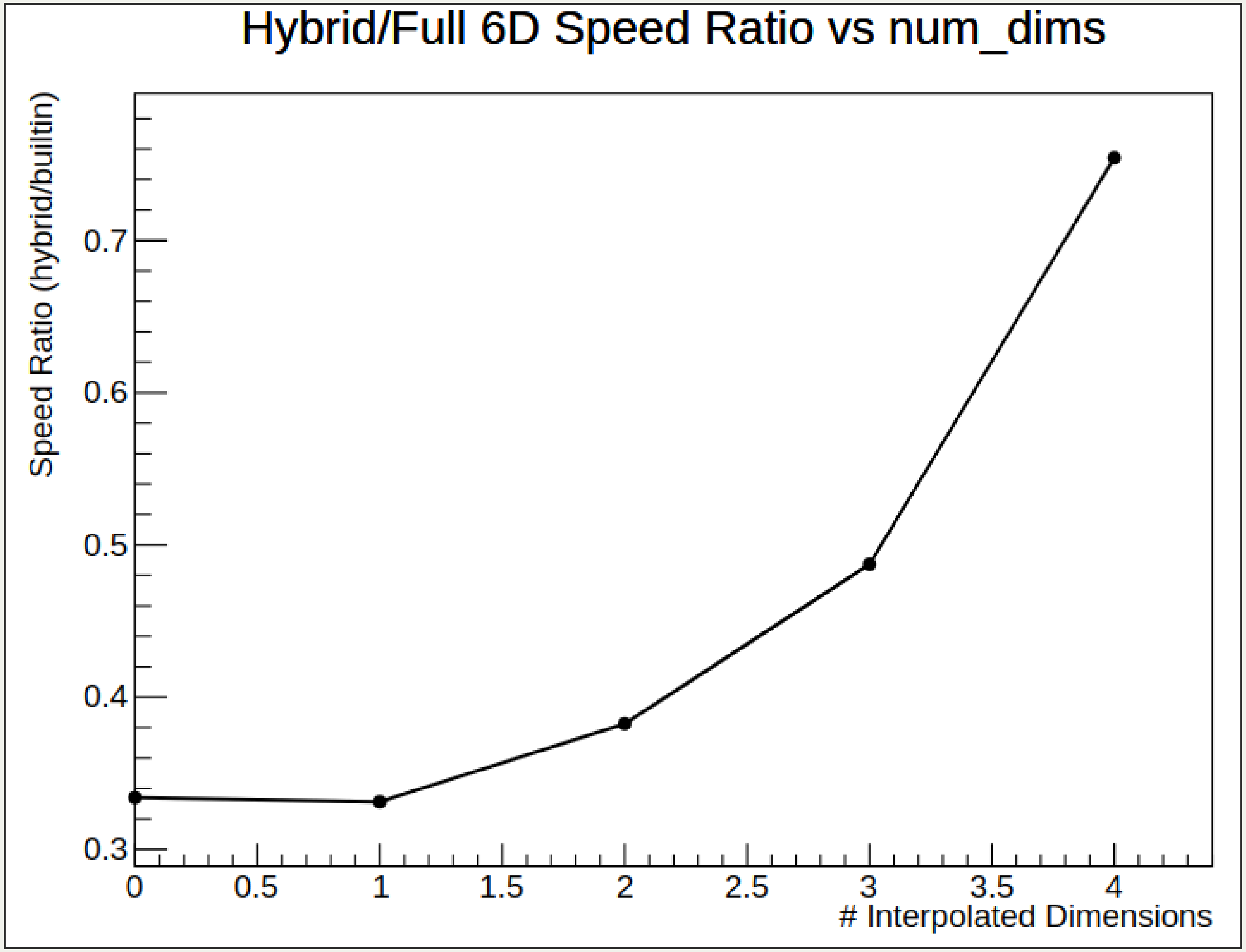

2D Fit Function (r_s vs t)
t
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
r_s
1000
800
600
400
200
0
Correction Factor



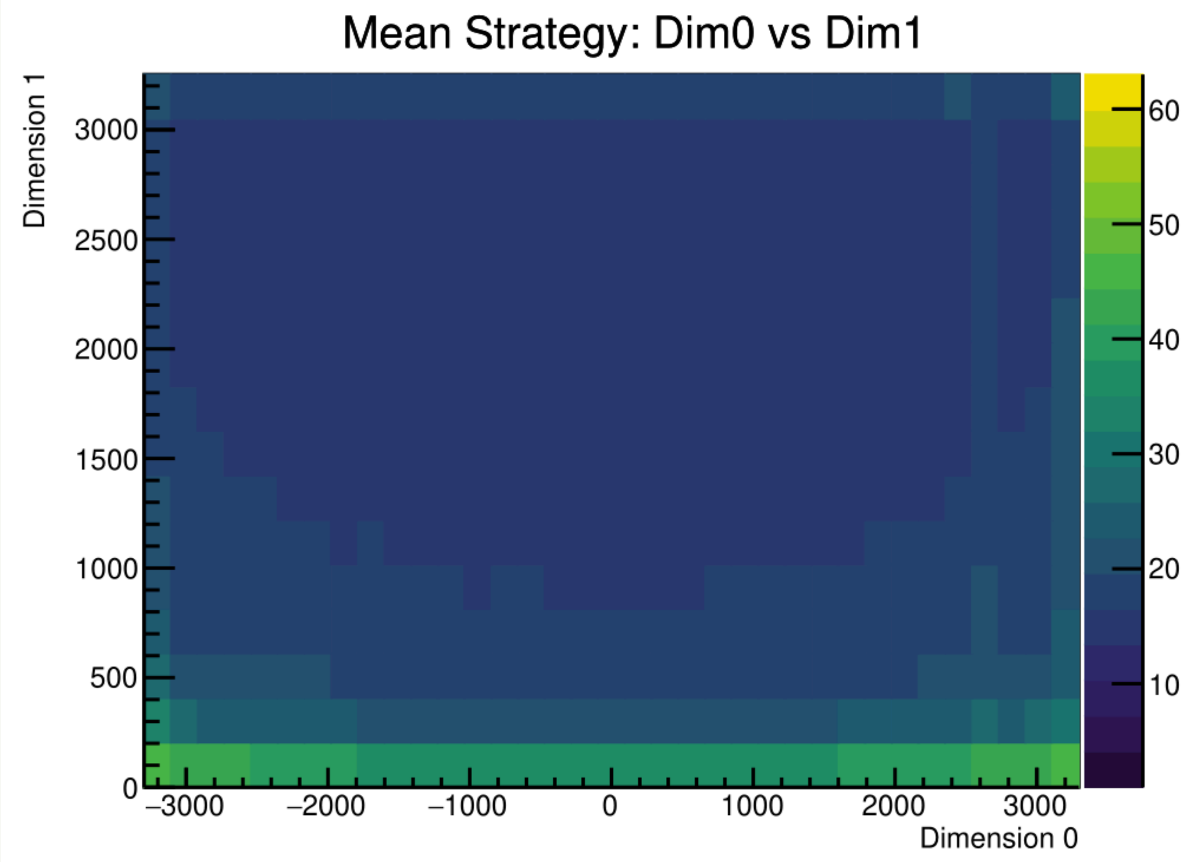
Corrected Hybrid Interpolation (r_s vs t)
t
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
r_s
Value
1
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
10^-10


t
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
r_s
−15
−16
−17
−18
−19

Parametrisation Results

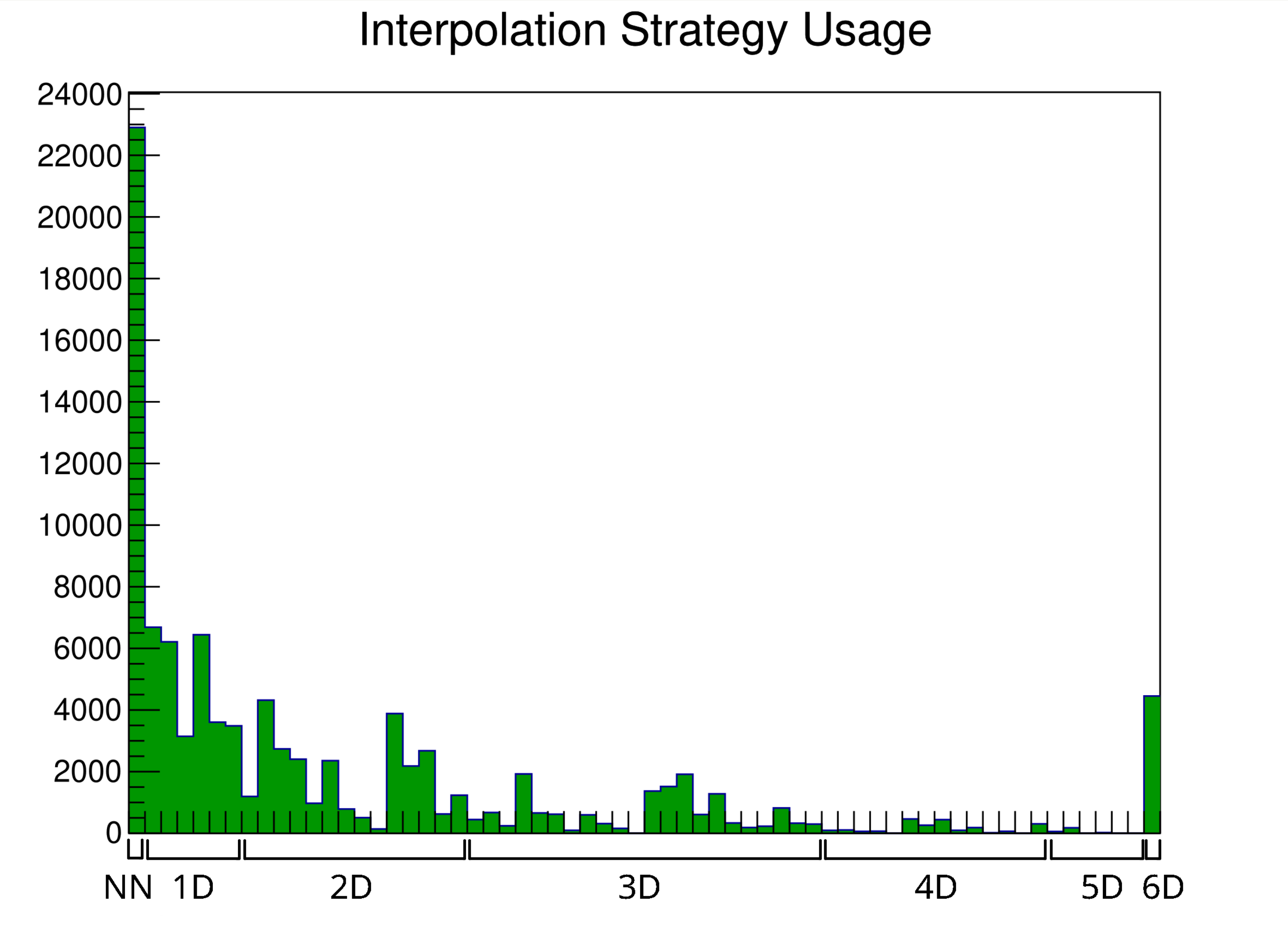


Selective Interpolation Optimisation

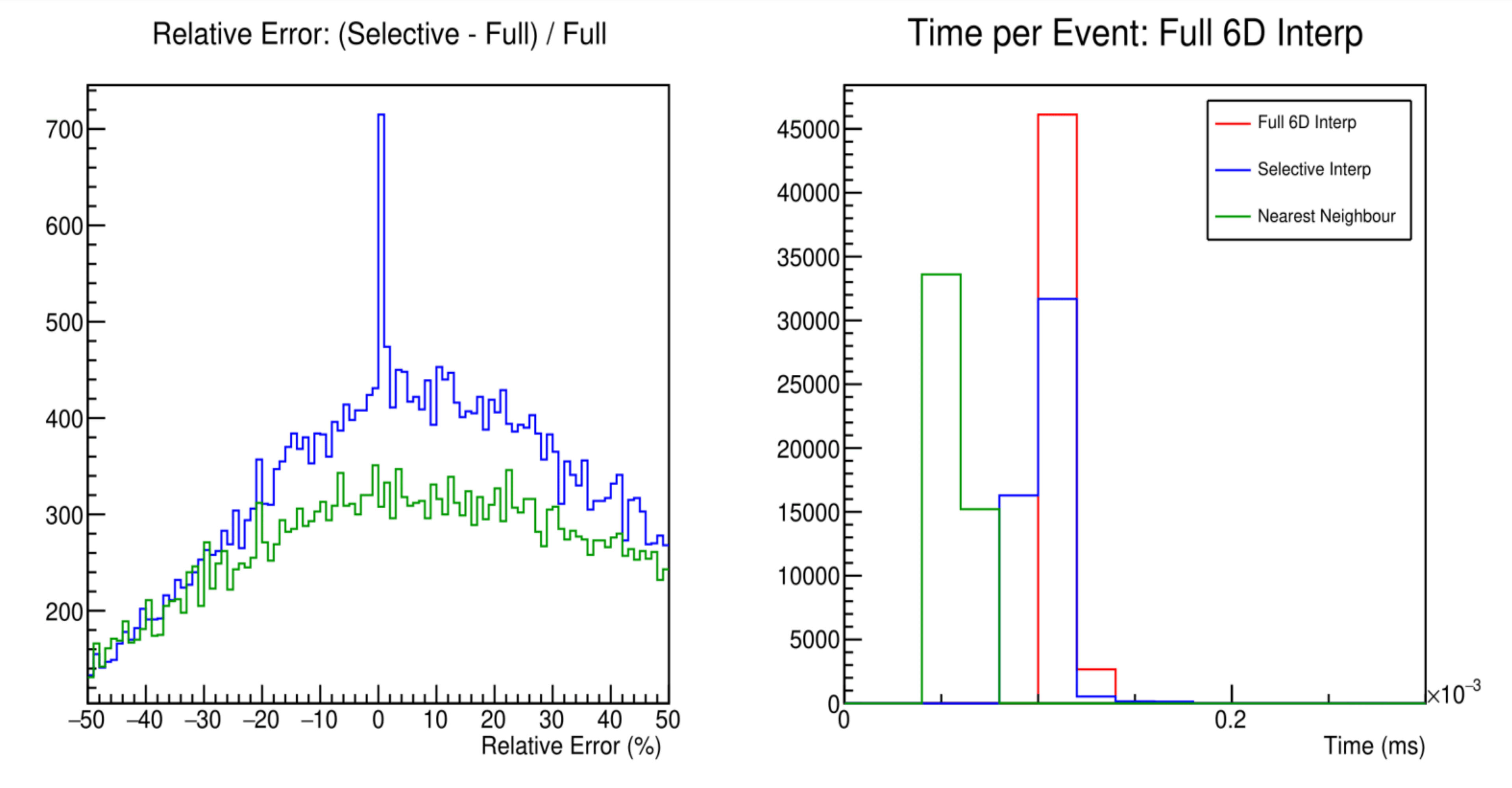


- Determine which dimensions must be interpolated for each bin
- Select most efficient interpolation strategy for a given point
- Generate bespoke interpolation function at runtime

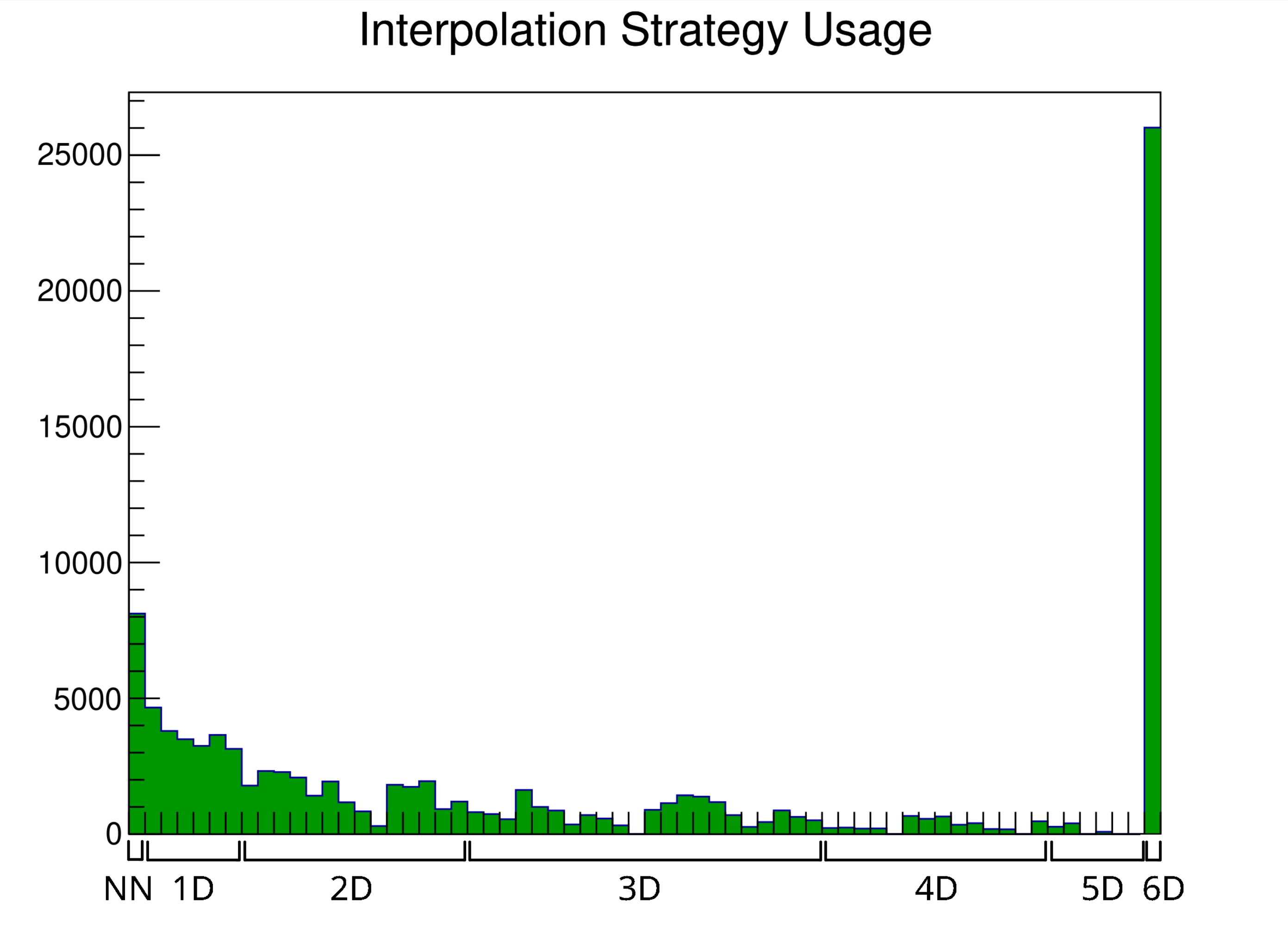
25% Error Threshold



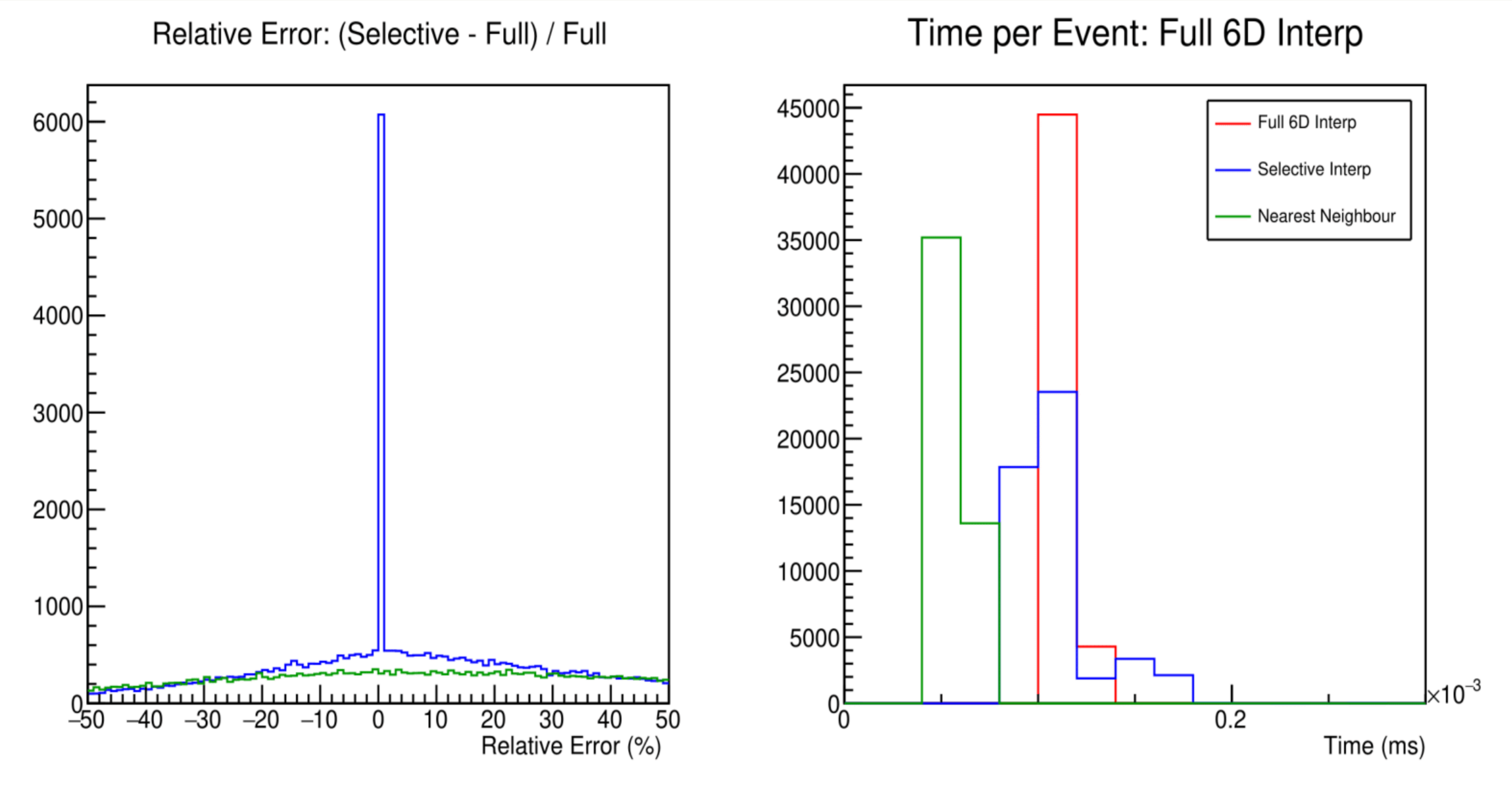
25% Error Threshold



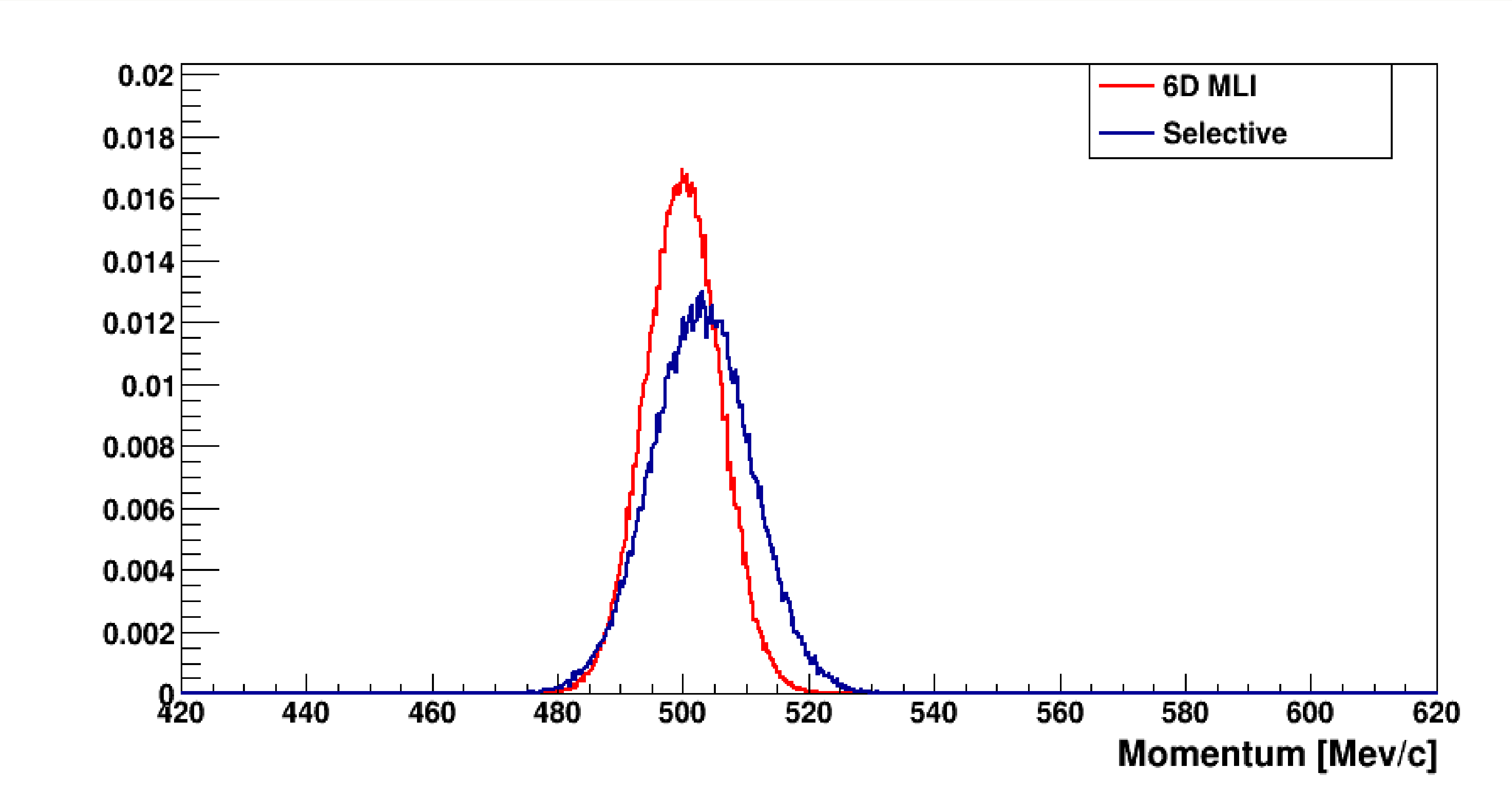
5% Error Threshold



5% Error Threshold



Momentum Reconstruction

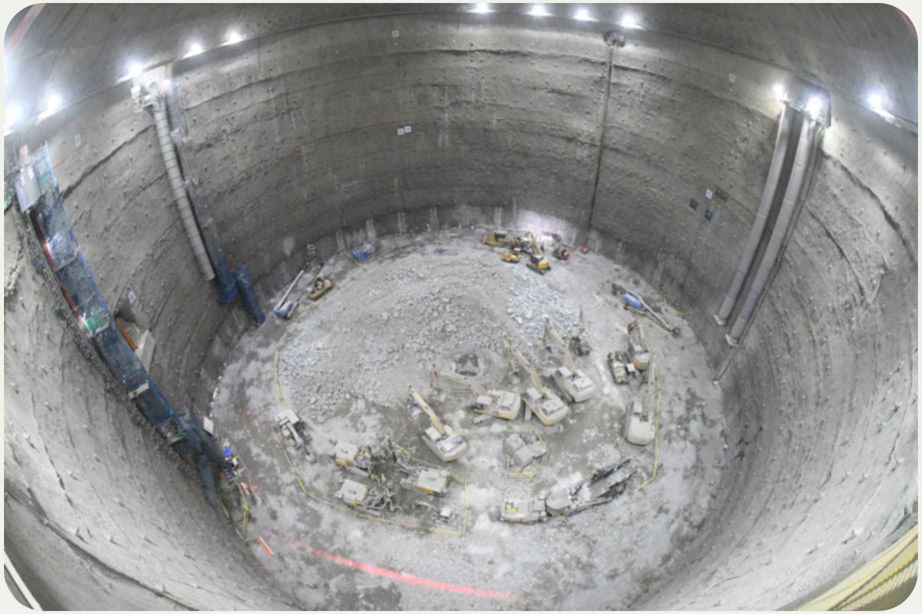


Final Results

Method	Runtime (s)	Time Ratio	p Mean (MeV)	p Variance
Nearest Neighbour	21.5	0.70	466.4	23.58
Selective Interpolation 25%	27.3	0.89	482.2	16.16
Selective Interpolation 5%	29.9	0.97	487.4	15.92
Full 6D Interpolation	30.7	1.00	491.3	14.39

10% speed-up with 2% additional error

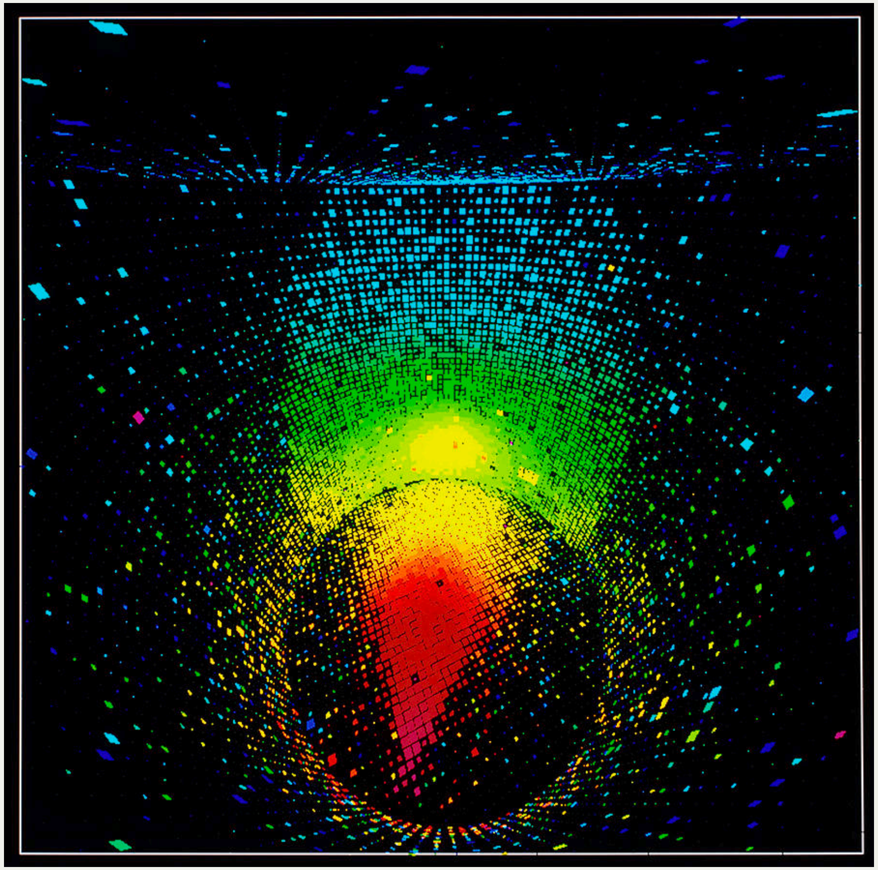
Next Steps



Recently completed HyperK cavern

- **Optimise reconstruction** with machine learning before calibration phase
- **Construction** shifts, installation and qualification of **timing system**
- Generate simulation samples and procedure for **data validation**
- Phenomenological studies preparing for **CPV** data analysis

Summary



- Developed three optimisation methods for the scattering table interpolation
- Selective interpolation provides good efficiency-accuracy trade-off
- Machine learning techniques may produce greater efficiency improvements
- Even 10% faster reconstruction = 10% more calibration data = improved systematics for HyperK