# Results:

To begin with, spectra with two positron lifetimes, and were simulated using PALSSIM, using an instrument resolution function (IRF) made up of three gaussians (see Table 1). With kept fixed at 180ps, spectra were generated for = 220-280ps at 10ps intervals using PALSSIM. For each value of , three spectra were generated, with intensities of and (respectively) set to 20%-80%, 50%-50%, 80%-20%. The goal was to evaluate how well PALSFIT returns each lifetime , and the intensity of the two lifetimes for each simulated spectrum.

Table

|  |  |  |
| --- | --- | --- |
| FWHM (ps) | Intensity (%) | Shift (ps) |
| 213.3 | 80 | 0 |
| 150 | 10 | -5 |
| 267 | 10 | 17 |

Starting with , we can see in Figure 1 that for some values of the software performed well at fitting the shorter lifetime, while it struggled for others.

Zooming in specifically to the = 250-270ps range, we can see in Figure 2 that, for these values in particular, it struggled to determine in the 50-50 and 80-20 case.

To evaluate the software’s performance when determining , we do the same as before, but instead of plotting on the y axis, we can plot the difference between the result and the original simulated values of to essentially tells us how far off PALSFIT is from the simulated value. Doing so we can see in Figure 3 that, aside from the 20-80 spectrum for = 220, here PALSFIT performs reasonably well.

Observing the error bars in the figures lets us know how confident our fit for the two lifetimes is. From here we can notice two important factors that affect the precision of the fitting. By looking at Figure 1 and Figure 3, we can deduce that the relative intensities of the two lifetimes is one of them. In Figure 1, where we are analysing , the error bars are the smallest for the 80-20 data, where the shorter lifetime is more intense, and in Figure 3, the opposite is the case and we have the 20-80 data being the most precise. Looking at all the figures, we can also notice that the error bars progressively shrink as the time interval between the lifetimes, which is the second factor, increases. Both these factors make intuitive sense.

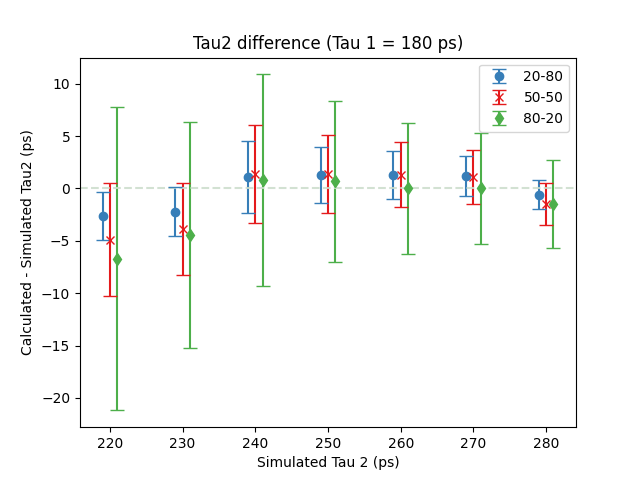
A graph of numbers and lines

Description automatically generated with medium confidence

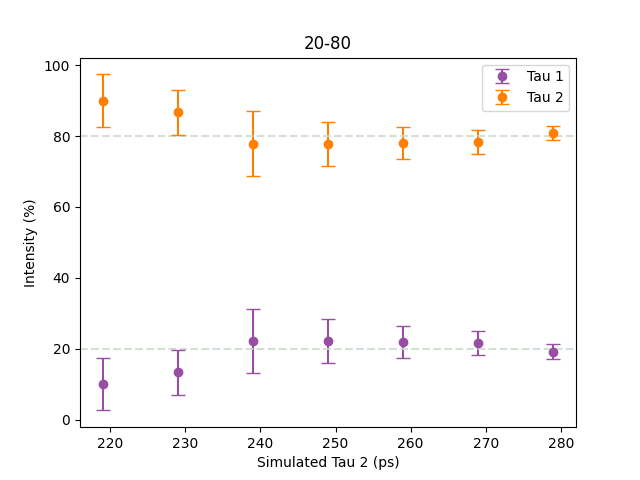
Figure

A graph of numbers and lines

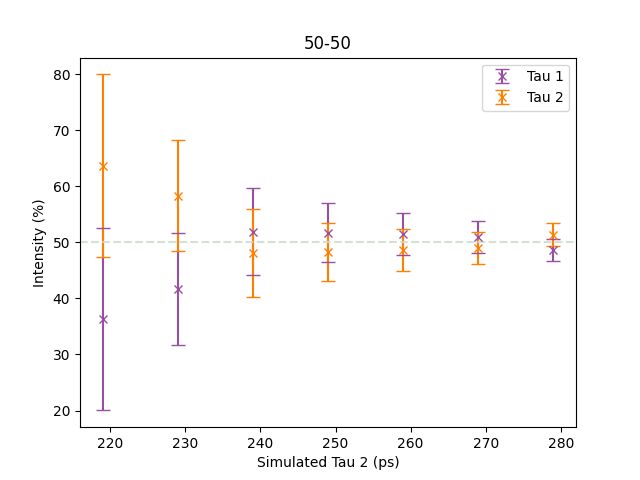
Description automatically generated with medium confidence

Figure 

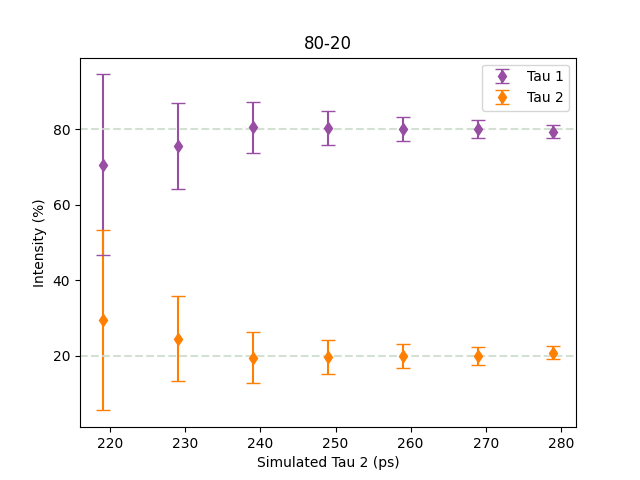
Figure



Figure



Figure

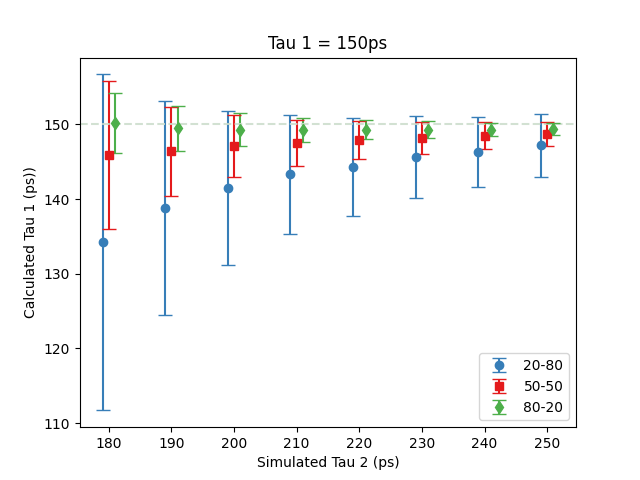


Figure

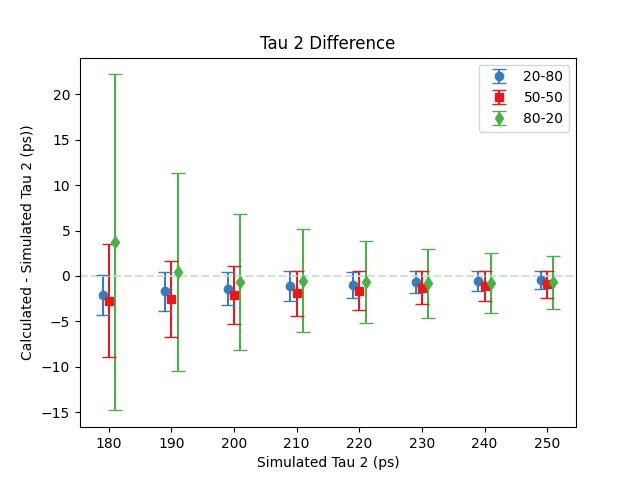
In Figure 4 - Figure 6, we can observe that when the relative intensity of was greater or equal to , then PALSFIT was able to calculate the appropriate lifetime intensities. However, when the intensity of was at 80%, as we can see in Figure 4, for the first two simulated values of , the software was unable to output the correct intensities. Looking at our error bars, we can see the same relationship between the size of the error bars and the time separation of the two lifetimes.

On the whole then, PALSFIT seemed to perform reasonably well, but not perfectly. As the first lifetime, , was kept constant at 180ps, the next step would be to change and see how that might alter our results. We can bring down to 150 and perform the same analysis. To keep things consistent, we’ll modify our range to maintain the spacing between the lifetimes consistent between runs. As such, we’ll set = 180-250ps. The results are in the figures below.

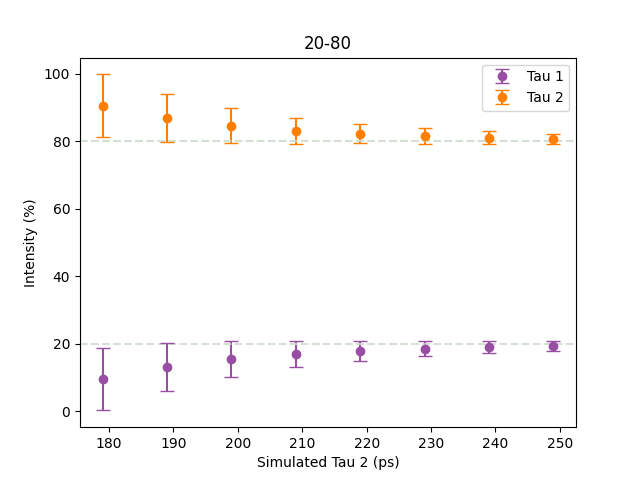
Interestingly, here the values calculated by PALSFIT are all, with a single exception, accurate to the simulated values. The single exception is in Figure 9, where the calculated intensity values are inaccurate for = 180.



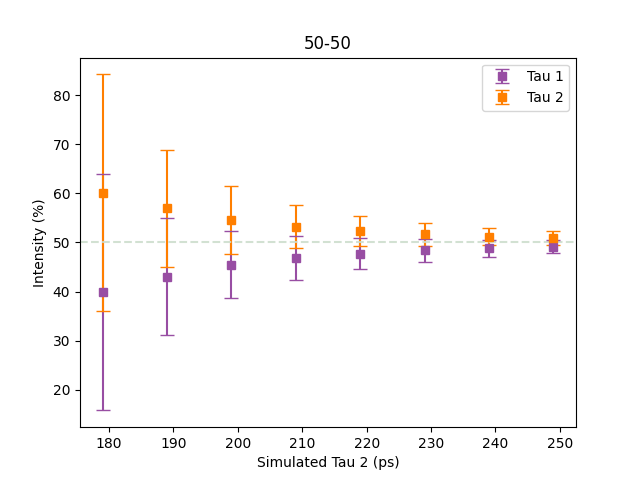
Figure



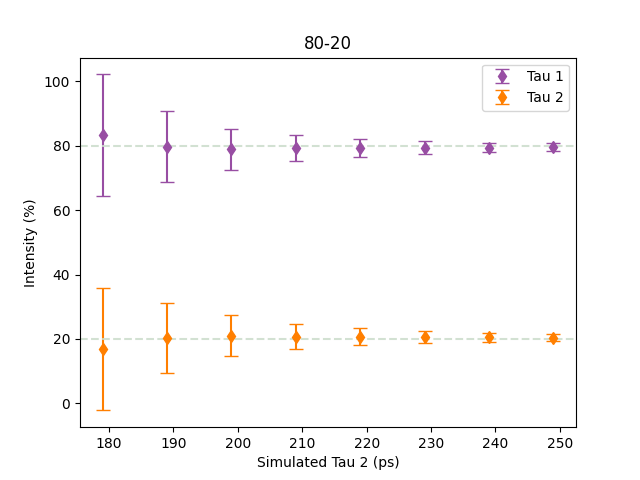
Figure



Figure



Figure



Figure

* Possible 220ps data? -> Summary: looks worse than 180 data

The next step was examining how the instrument resolution function (IRF) affected the results. Instead of a three gaussian IRF, a single gaussian was used to make things simpler. The values for and were picked from the most successful batch, and so was fixed to 150ps and ranged from 180-250ps.

Rationale: 1.4 rule of thumb- test