S1 detection with cross correlation

Giacomo Petrillo
2021-xx-xx
University of Pisa
info@giacomopetrillo.com
https://bitbucket.org/Gattocrucco/locs12/

Summary

- Simulate S1 hits + dark count hits (temporal only)
- Add gaussian smearing for temporal resolution + photon flight time
- Do a kernel density estimation of the hits temporal distribution
- Cross-correlate the temporal distribution with the S1 pdf (ER and NR)
- Set a threshold on the output and measure S1 efficiency and fake rate
- Compare with other filters
- Look at ER/NR discrimination (preliminary, no real results)

S1 distribution (1/2)

$$p_{\mathrm{S1}}(t; p_{\mathrm{fast}}, \tau_{\mathrm{fast}}, \tau_{\mathrm{slow}}) = p_{\mathrm{fast}} \,\mathrm{expon}(t; \tau_{\mathrm{fast}}) + (1 - p_{\mathrm{fast}}) \,\mathrm{expon}(t; \tau_{\mathrm{slow}})$$

$$\mathrm{expon}(t; \tau) \otimes \mathrm{norm}(t; \sigma) = \frac{1}{\tau} e^{-\frac{t}{\tau}} e^{\frac{1}{2}\left(\frac{\sigma}{\tau}\right)^2} \Phi\left(\frac{t}{\sigma} - \frac{\sigma}{\tau}\right) \equiv f(t; \tau, \sigma)$$

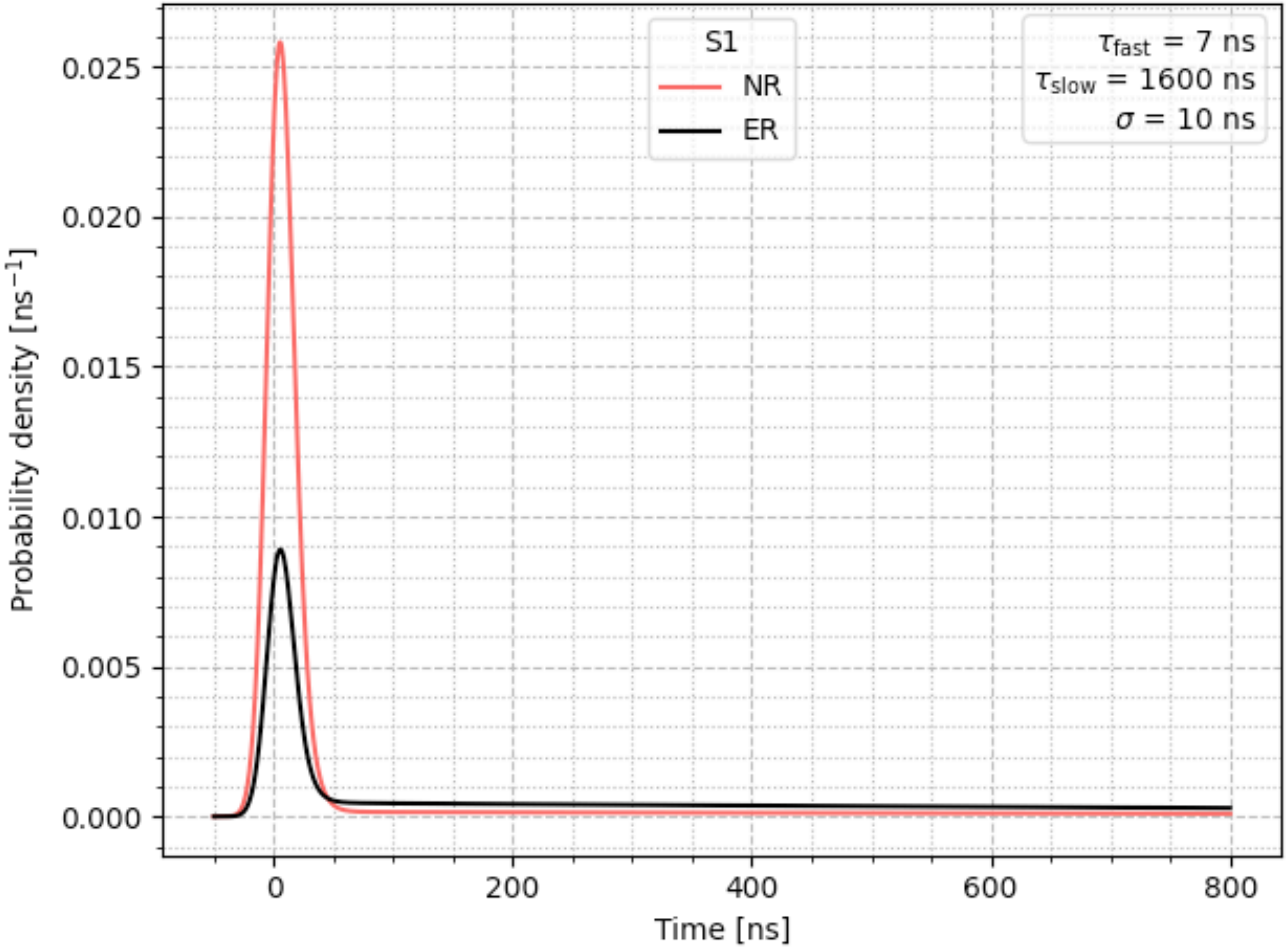
(Φ is the gaussian CDF)

$$p_{\mathrm{S1,gauss}}(t; p_{\mathrm{fast}}, \tau_{\mathrm{fast}}, \tau_{\mathrm{slow}}, \sigma) = p_{\mathrm{fast}} f(t; \tau_{\mathrm{fast}}, \sigma) + (1 - p_{\mathrm{fast}}) f(t; \tau_{\mathrm{slow}}, \sigma)$$

$$\begin{aligned} p_{\mathrm{fast}} &= 0.75 \text{ (NR)},\ 0.25 \text{ (ER)} \\ \tau_{\mathrm{fast}} &= 7\,\mathrm{ns} \\ \tau_{\mathrm{slow}} &= 1600\,\mathrm{ns} \\ \sigma &= 10\,\mathrm{ns} \end{aligned}$$

Sigma is an eyeball estimate from some ns for temporal resolution and 3 ns per meter light speed (the detector is some meters across)

S1 distribution (2/2)



Cross correlation (1/2)

We want to estimate the temporal distribution of photons from data only. The *empirical density* is putting a dirac delta on each hit:

$$f(t) = \sum_i \delta(t - t_i)$$

A *kernel density estimation* (KDE) is the same but putting a gaussian on each hit:

$$\begin{aligned} g(t) &= \sum_i \mathrm{norm}(t - t_i; \sigma_{\mathrm{KDE}}) \\ &= f(t) \otimes \mathrm{norm}(t; \sigma_{\mathrm{KDE}}) \end{aligned}$$

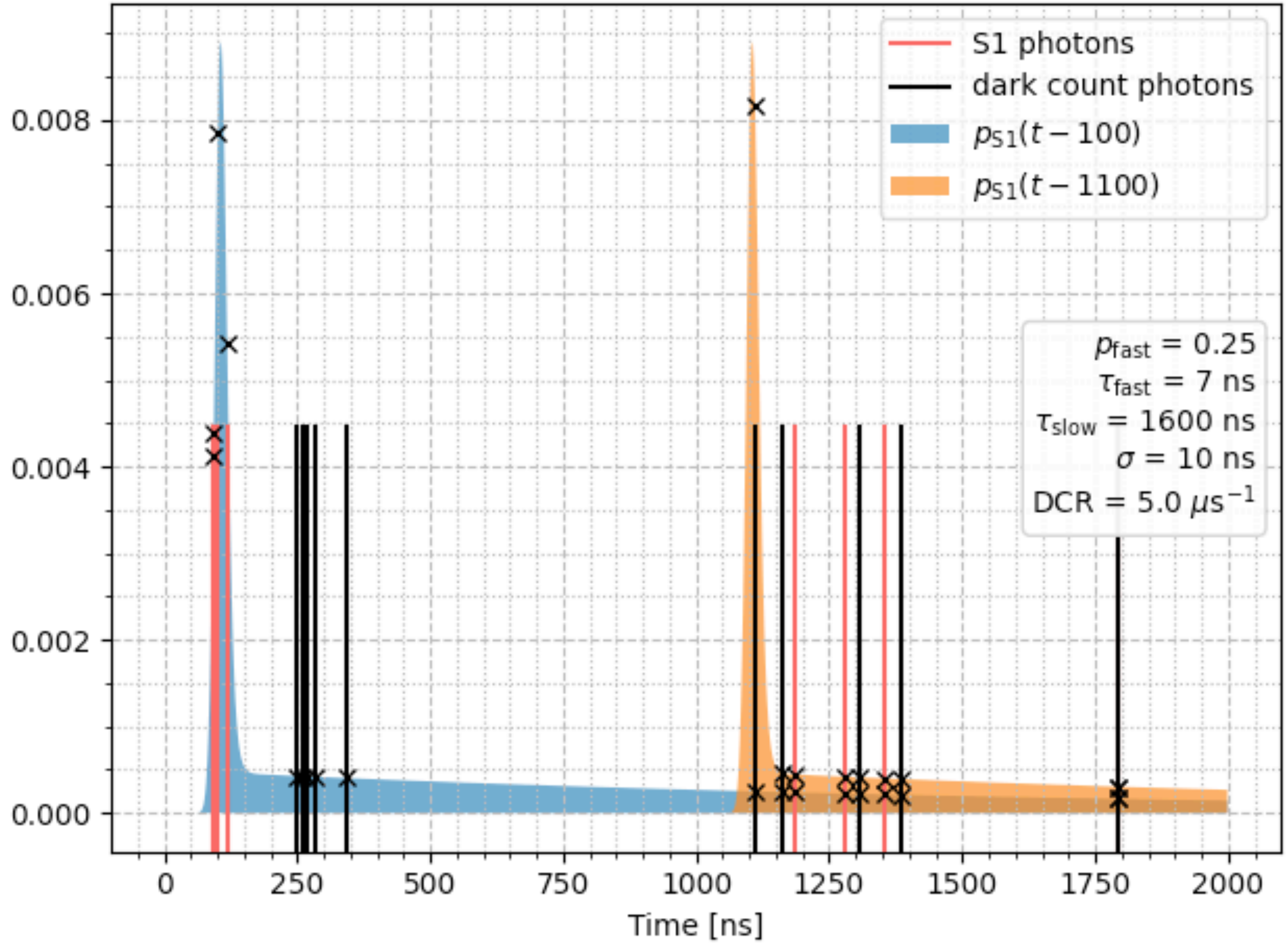
Now we do something like the matched filter, we cross correlate the KDE with the S1 distribution:

$$\begin{aligned} cc(t) &= g(t) \otimes p_{\mathrm{S1,gauss}}(-t; \ldots, \sigma) \\ &= \left[f(t) \otimes \mathrm{norm}(t; \sigma_{\mathrm{KDE}})\right] \otimes p_{\mathrm{S1,gauss}}(-t; \ldots, \sigma) \\ &= f(t) \otimes \left[\mathrm{norm}(t; \sigma_{\mathrm{KDE}}) \otimes p_{\mathrm{S1,gauss}}(-t; \ldots, \sigma)\right] \\ &= \sum_i p_{\mathrm{S1,gauss}}\left(t_i - t; \ldots, \sqrt{\sigma^2 + \sigma_{\mathrm{KDE}}^2}\right) \end{aligned}$$

Cross correlation (2/2)

$$cc(t) = \sum_i p_{\mathrm{S1,gauss}}\left(t_i - t; \ldots, \sqrt{\sigma^2 + \sigma^2_{\mathrm{KDE}}}\right)$$

As we translate the pS1 pdf we sum its value on all hits.



Computing the cross correlation (1/2)

The filter output is continuous. We could evaluate it in steps of say 1 ns.

But instead we try to compute it on less points. We translate the p_S1 template to have the maximum at t = 0, such that for a single photon the peak would occur exactly on the photon hit time.

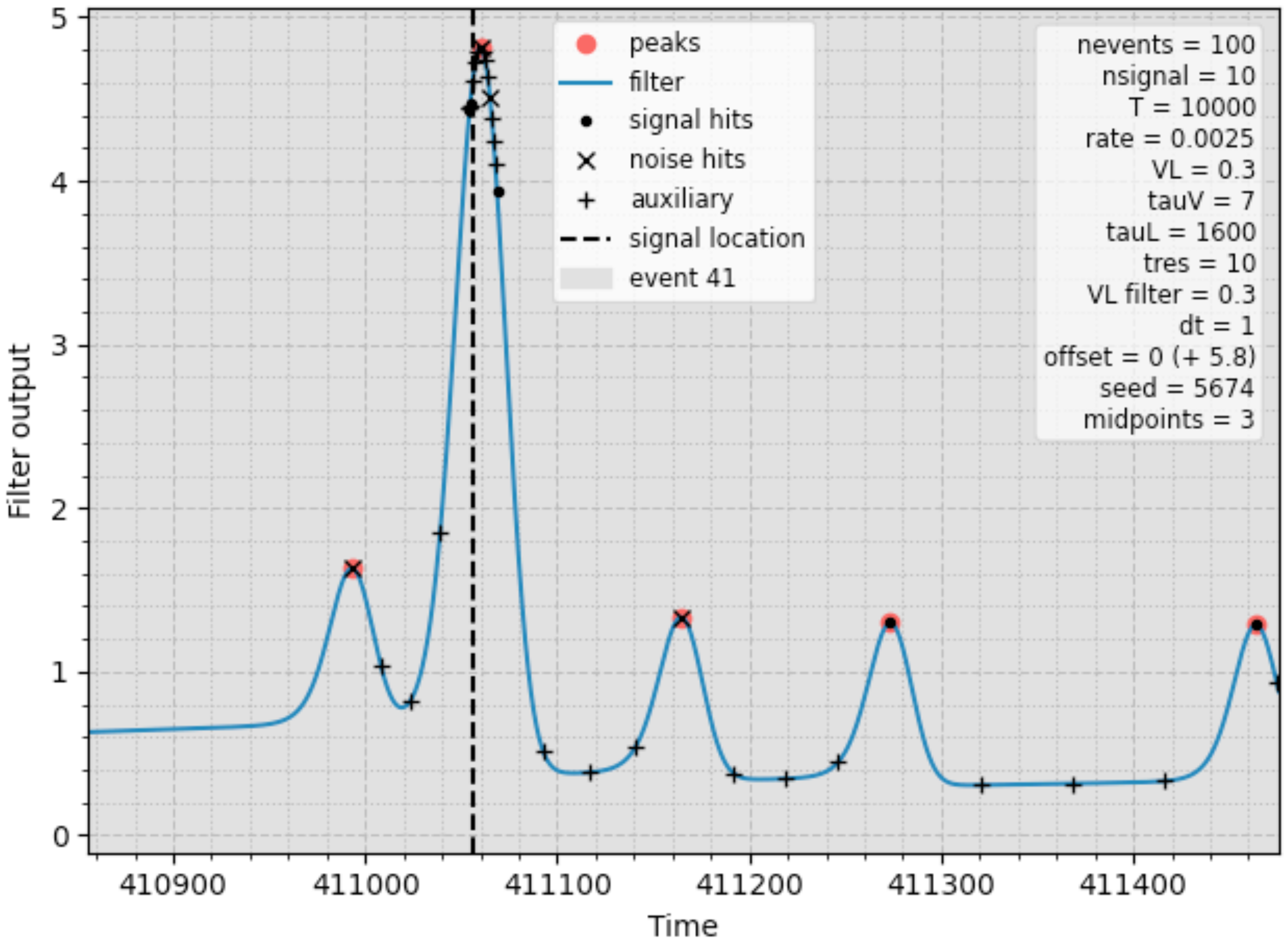
Then we evaluate the filter on photon hit times and on 3 midpoints between consecutive photons.

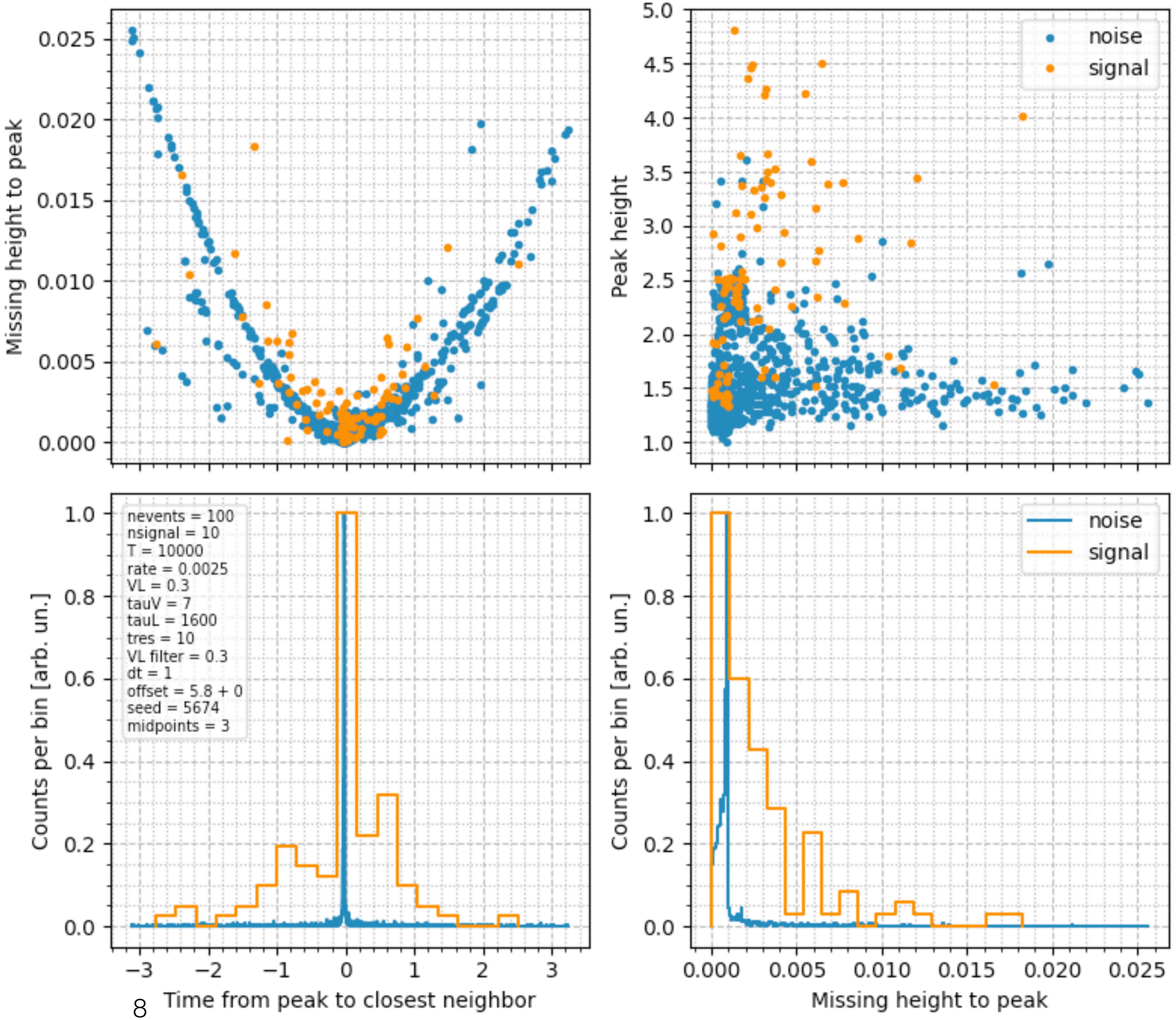
We also set the peak height to 1 such that the single photon response is 1.

Finally we compare this with 1 ns discretization + quadratic interpolation.

Computing the cross correlation (2/2)

The scatterplots on the right show by how much we underestimate the peaks height by not evaluating the filter continuously. At most 1 %. This simulation is with ER 10 photon S1 and DCR=300 cps/pdm.





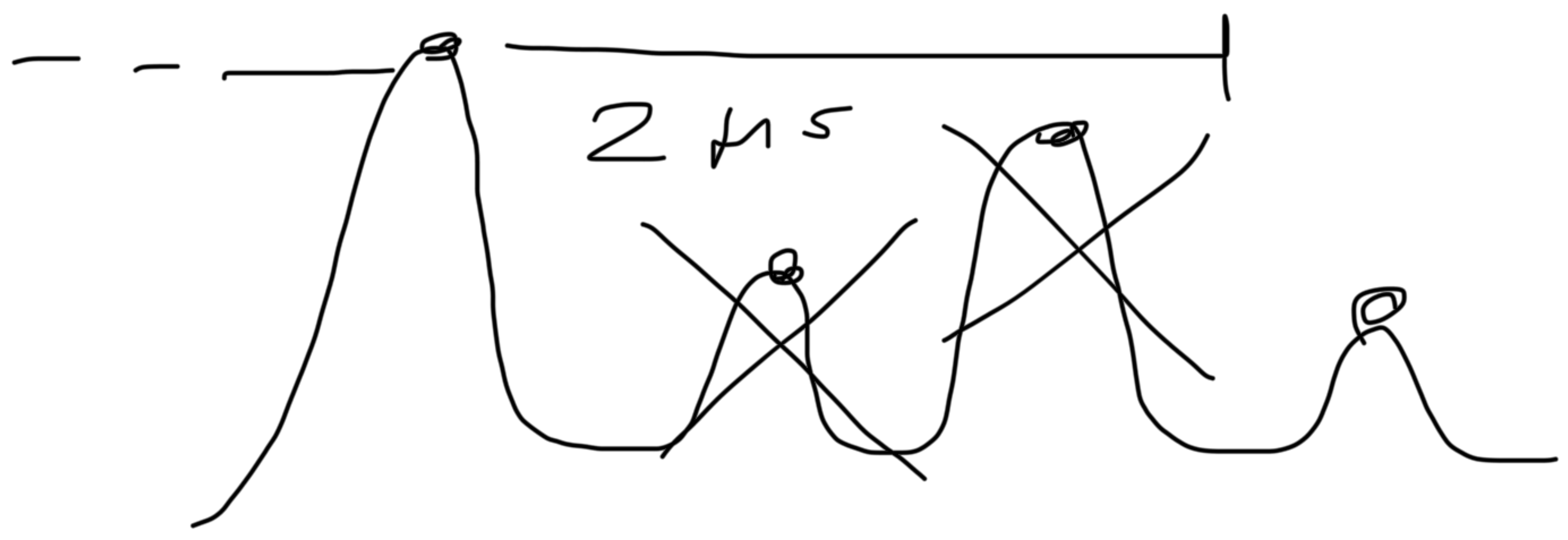
S1 candidate definition

- We put a "dead radius" of 2 μs around evaluated points, giving precedence to higher values, i.e. a sort of peak finding with a minimum distance of 2 μs.

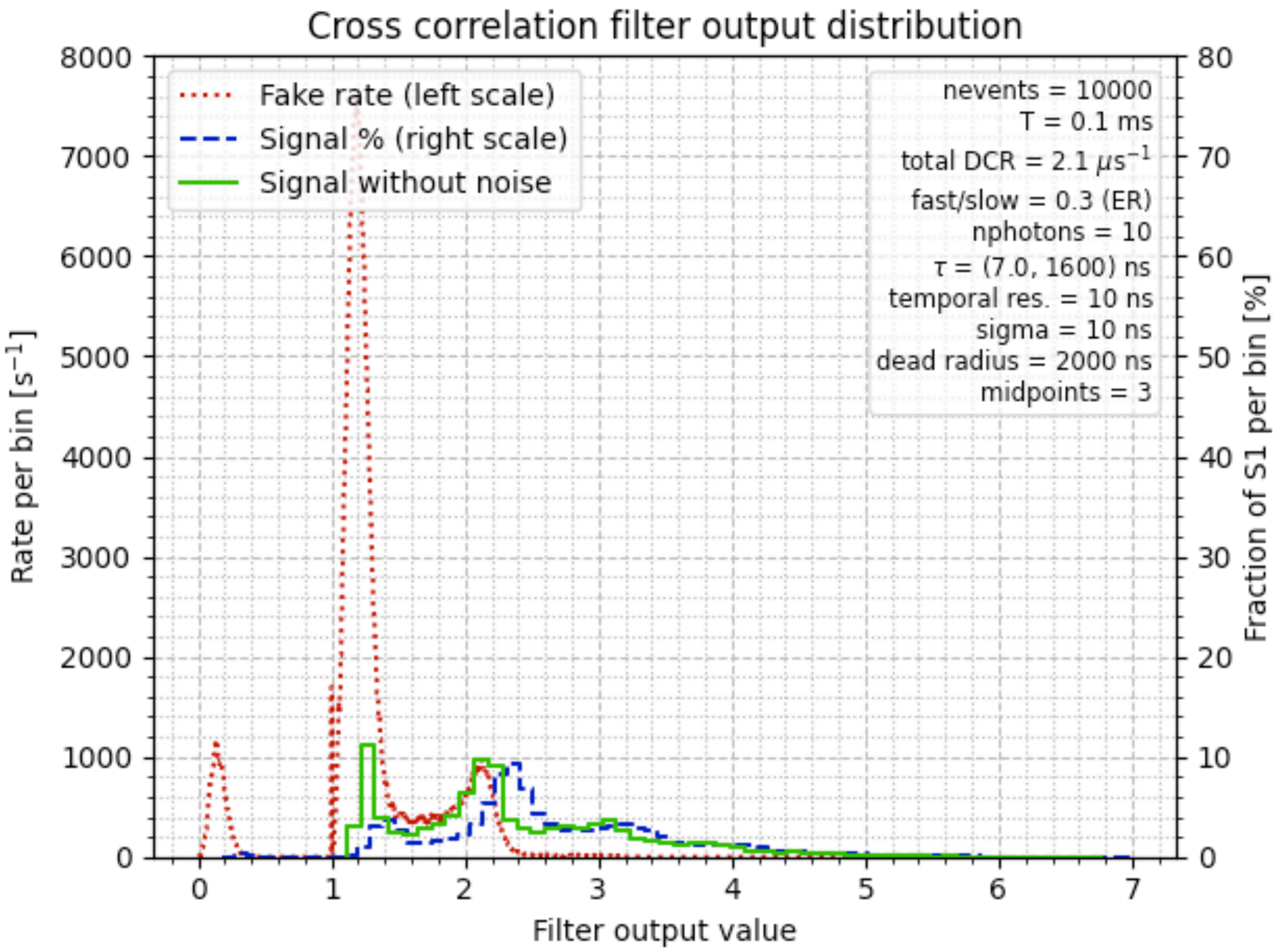
- All surviving points are considered S1 candidates (thus the rate is at most 1 every 2 μs).

- In the case that a non-S1 peak obscures a S1 peak, the signal will be within 2 μs of the candidate anyway, so we don't lose it if we imagine to save e.g. 10 μs of photons (so we are neglecting temporal localization).

- I need this instead of actual threshold-crossing + deadtime to do the computation at once for all thresholds.

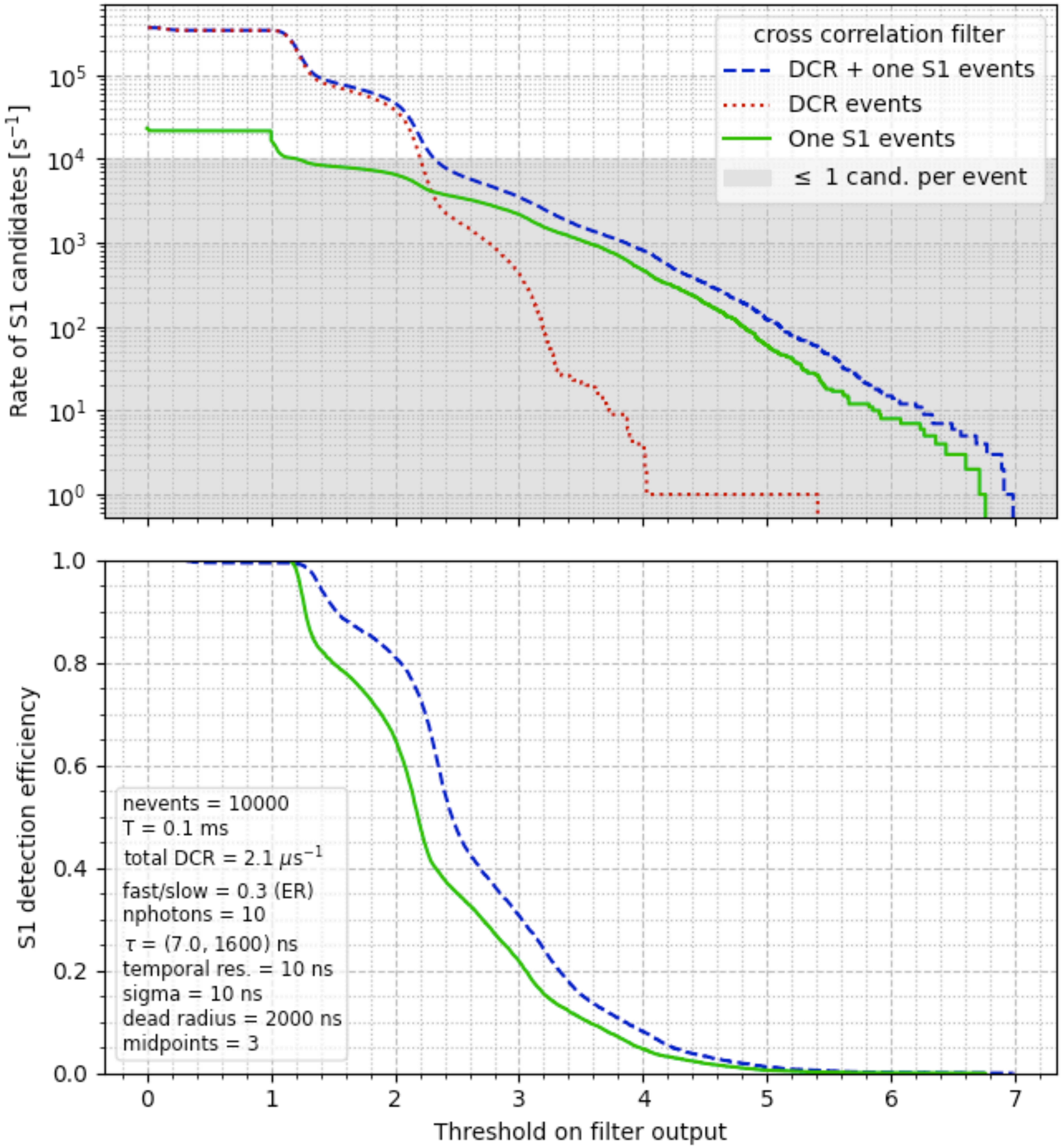


Efficiency and fake rate (1/3)



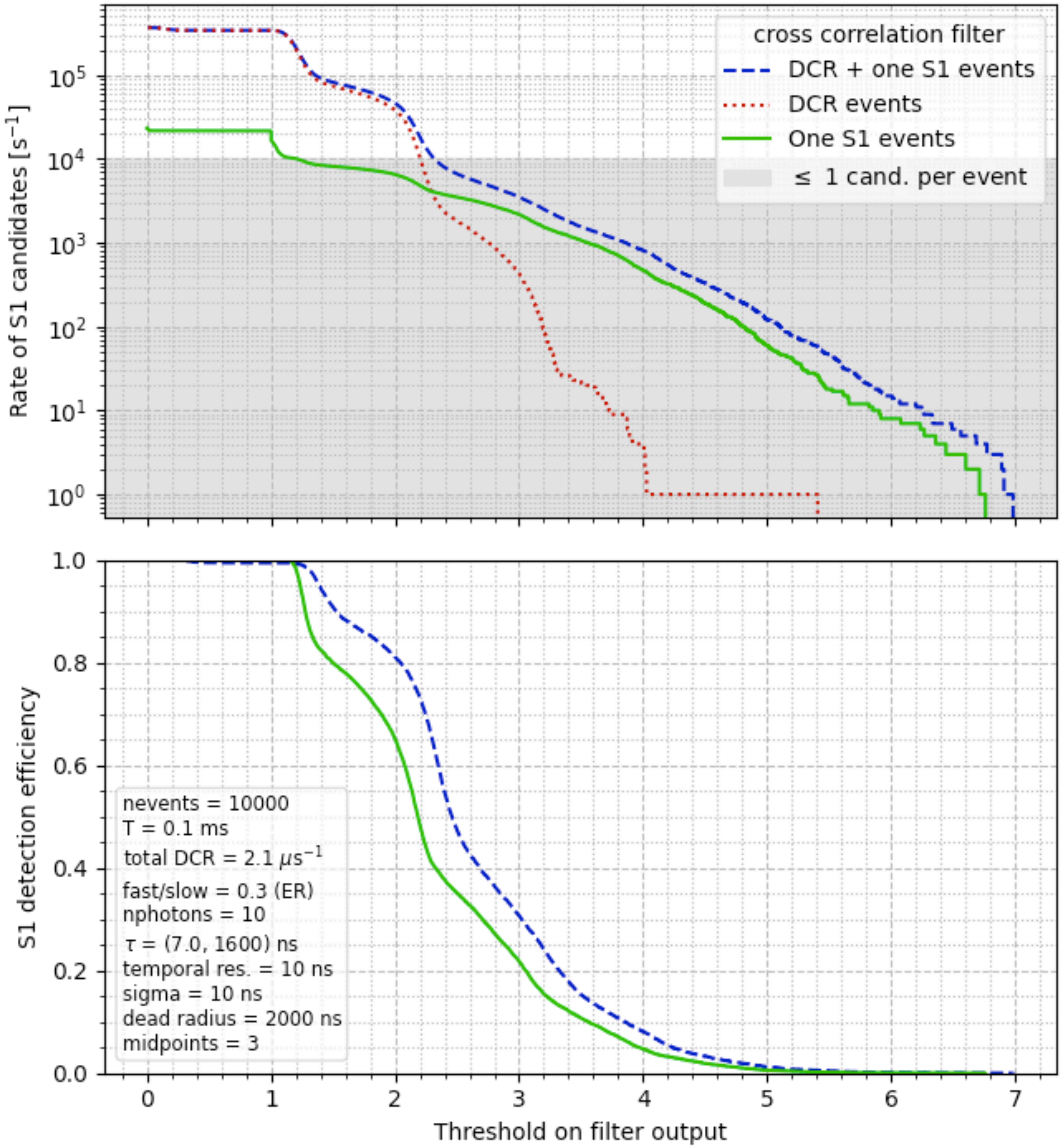
- 10000 "events" with 0.1 ms of dark count and an S1 in the middle
- In each event the candidate closer to the S1 location is considered the "true" S1
- We also show the distribution of peak height for S1 filtered without dark count photons (green line), just to show it must not be used since the noise is not zero mean for this filter

Efficiency and fake rate (2/3)

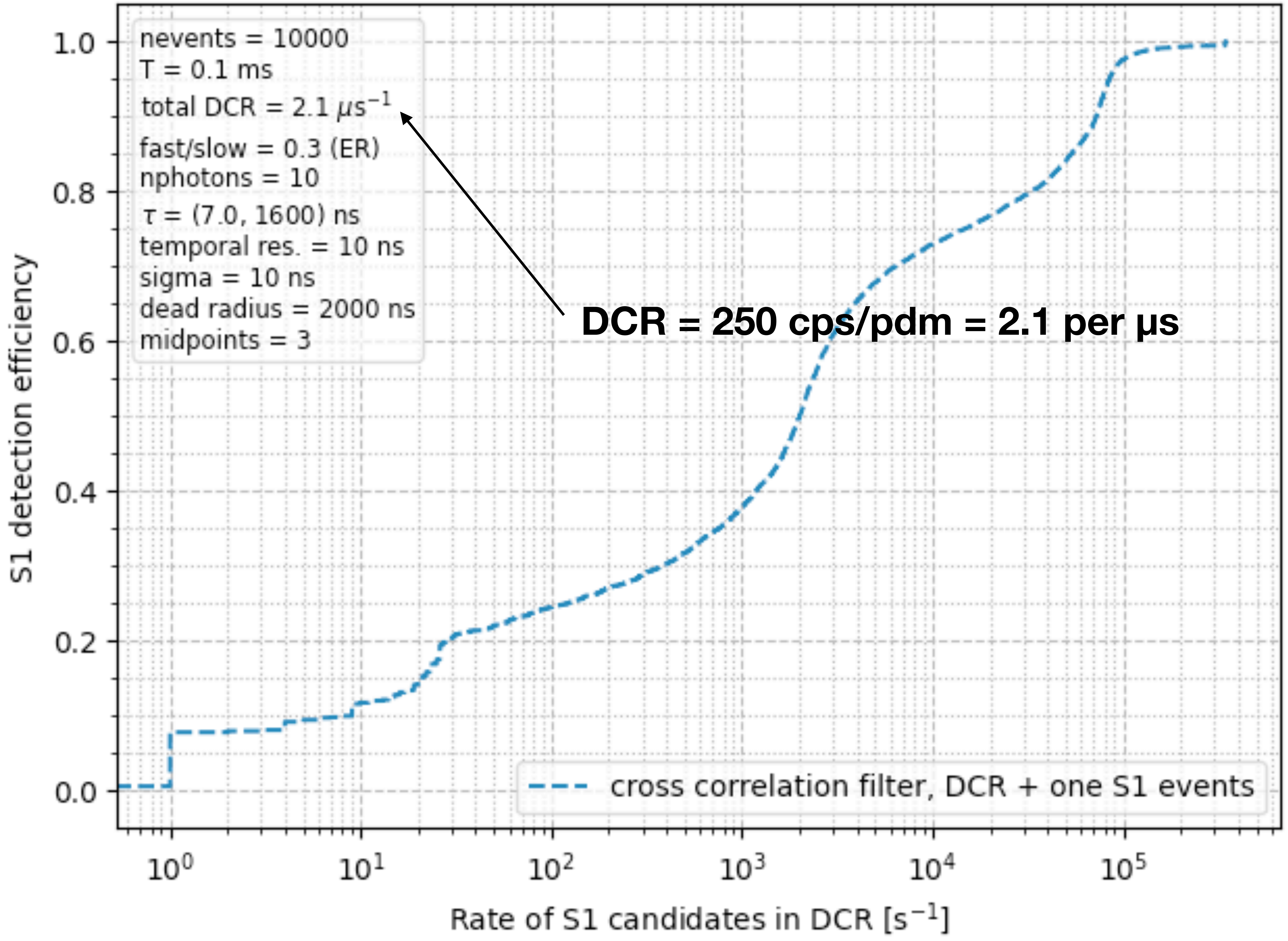


- The histogram is not very legible, here's the survival function (amount of candidates above threshold)
- The red dotted line top plot gives the fake rate, the blue dashed line bottom plot the efficiency, threshold on x-axis
- Since this simulation is 10000 x 0.1 ms the lowest fake rate detectable is 1 candidate = 1 per second

Efficiency and fake rate (3/3)



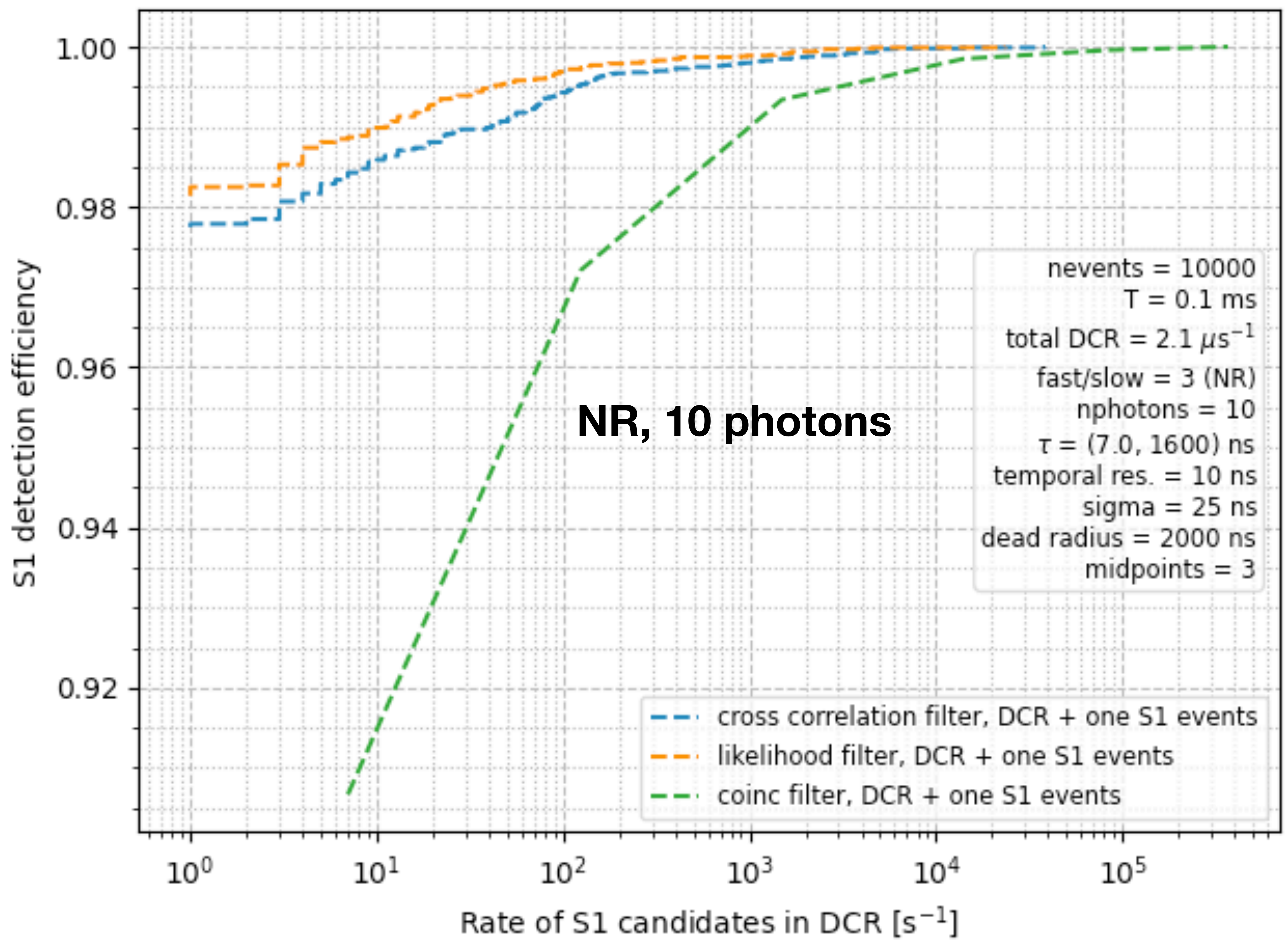
And below efficiency vs. fake rate

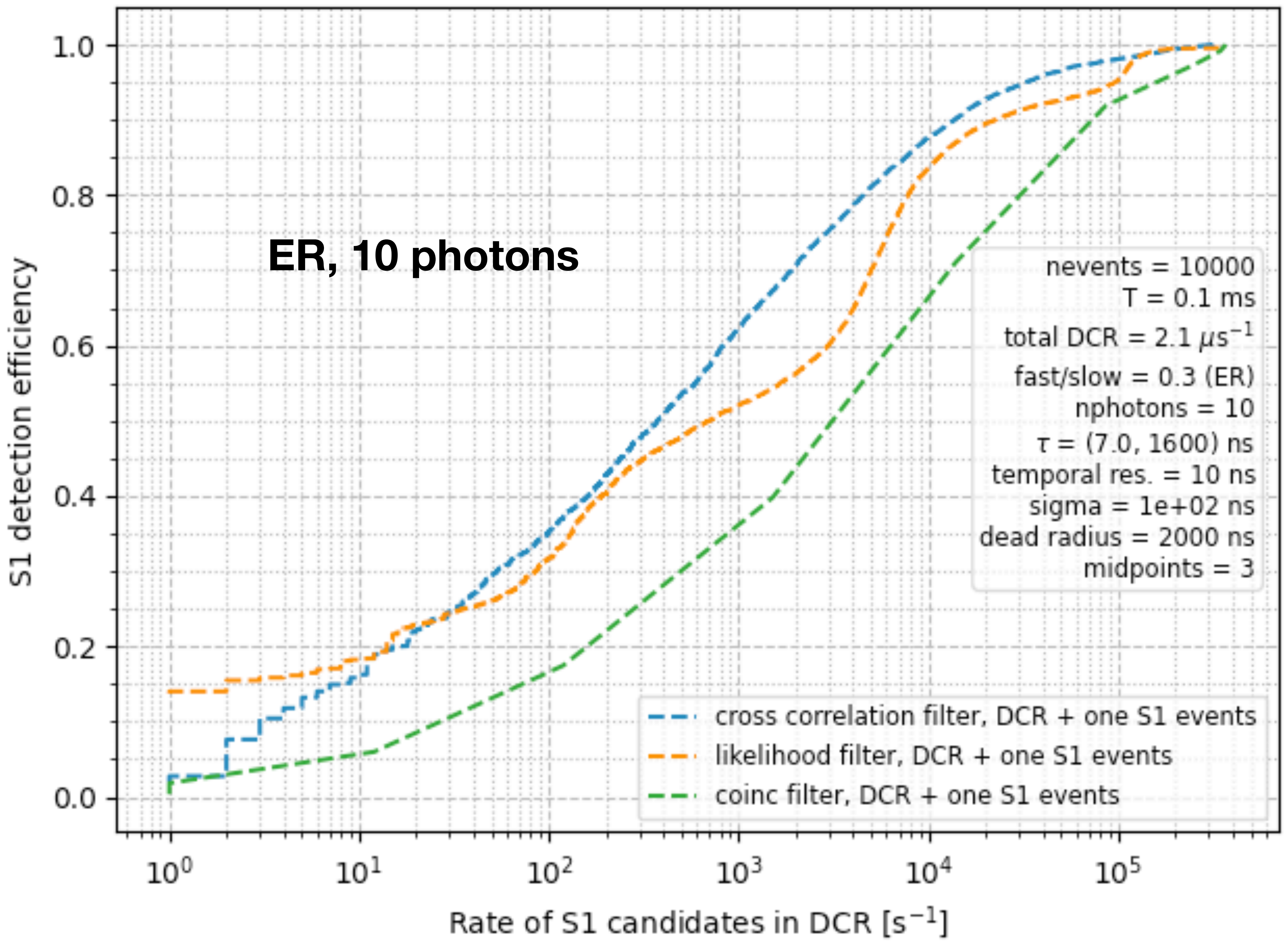


Comparison with other filters (1/2)

In the legends: "coinc filter" = count the photons in a 200 ns window, "likelihood filter" = use the likelihood of S1 + DCR.

The cross correlation is better that the simple coincidence. The likelihood is better for NR, maybe for ER too at low fake rate. However: 1) I've not optimized the coincidence time; 2) the likelihood is done using the true number of photons instead of fitting it.





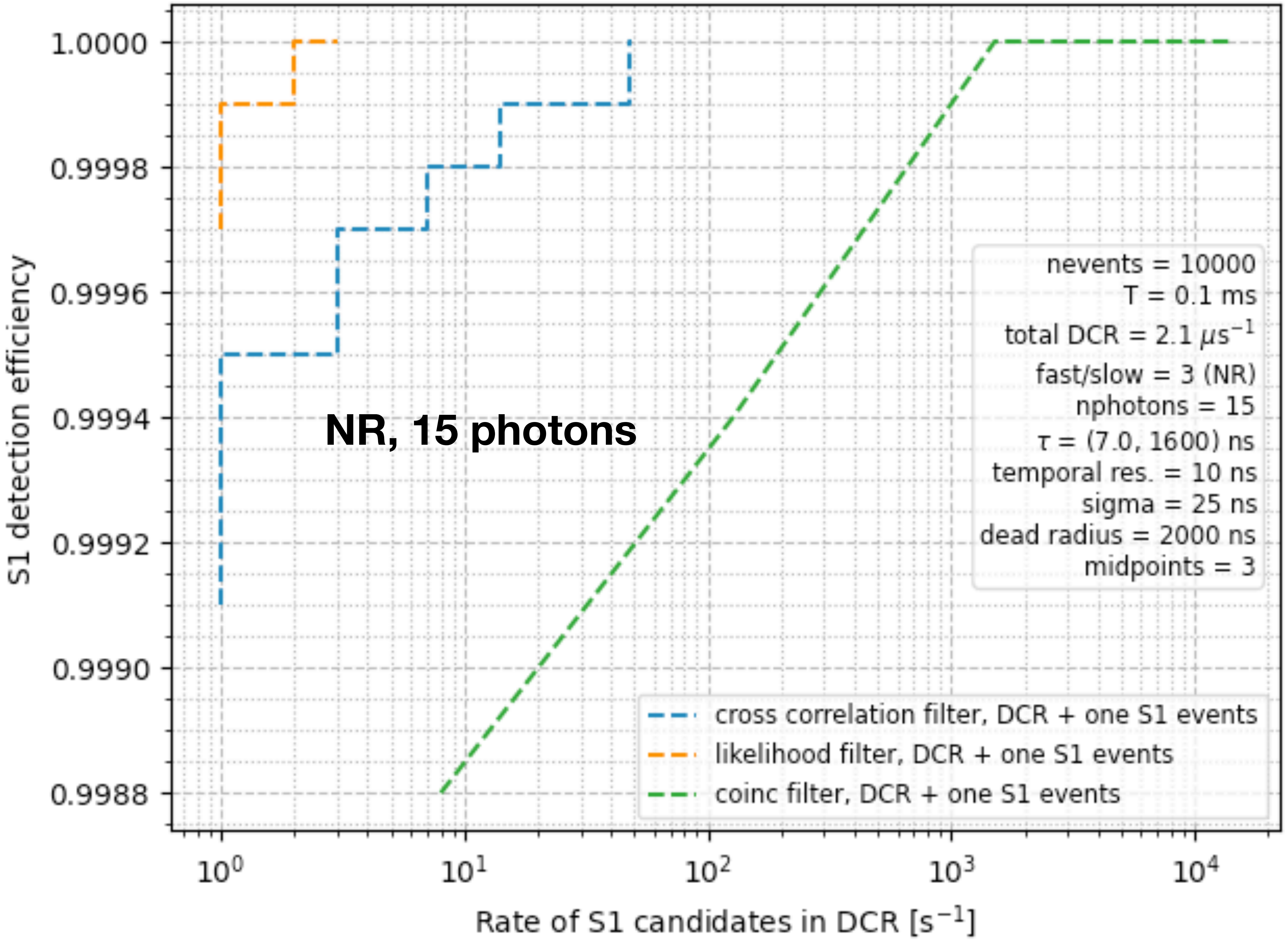
Comparison with other filters (2/2)

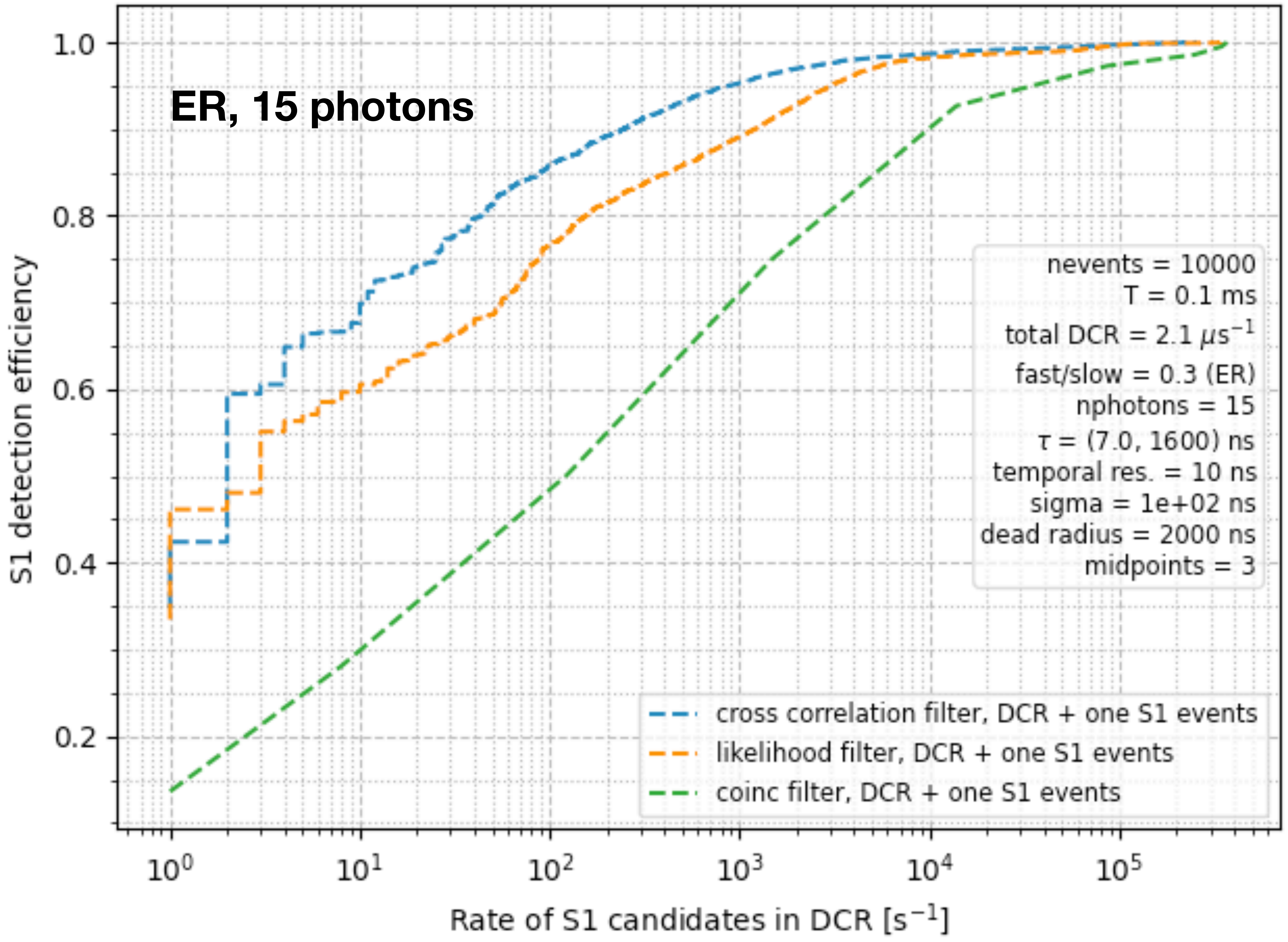
With 15 photons the comparison likelihood/cross correlation is still not decided.

Note: the likelihood "filter" is

$$lk(t) = \sum_i \log\left(1 + \frac{N}{R} p_{\mathrm{S1,gauss}}(t_i - t; \ldots, \sigma)\right)$$

where N = number of S1 photons, R = dark count rate



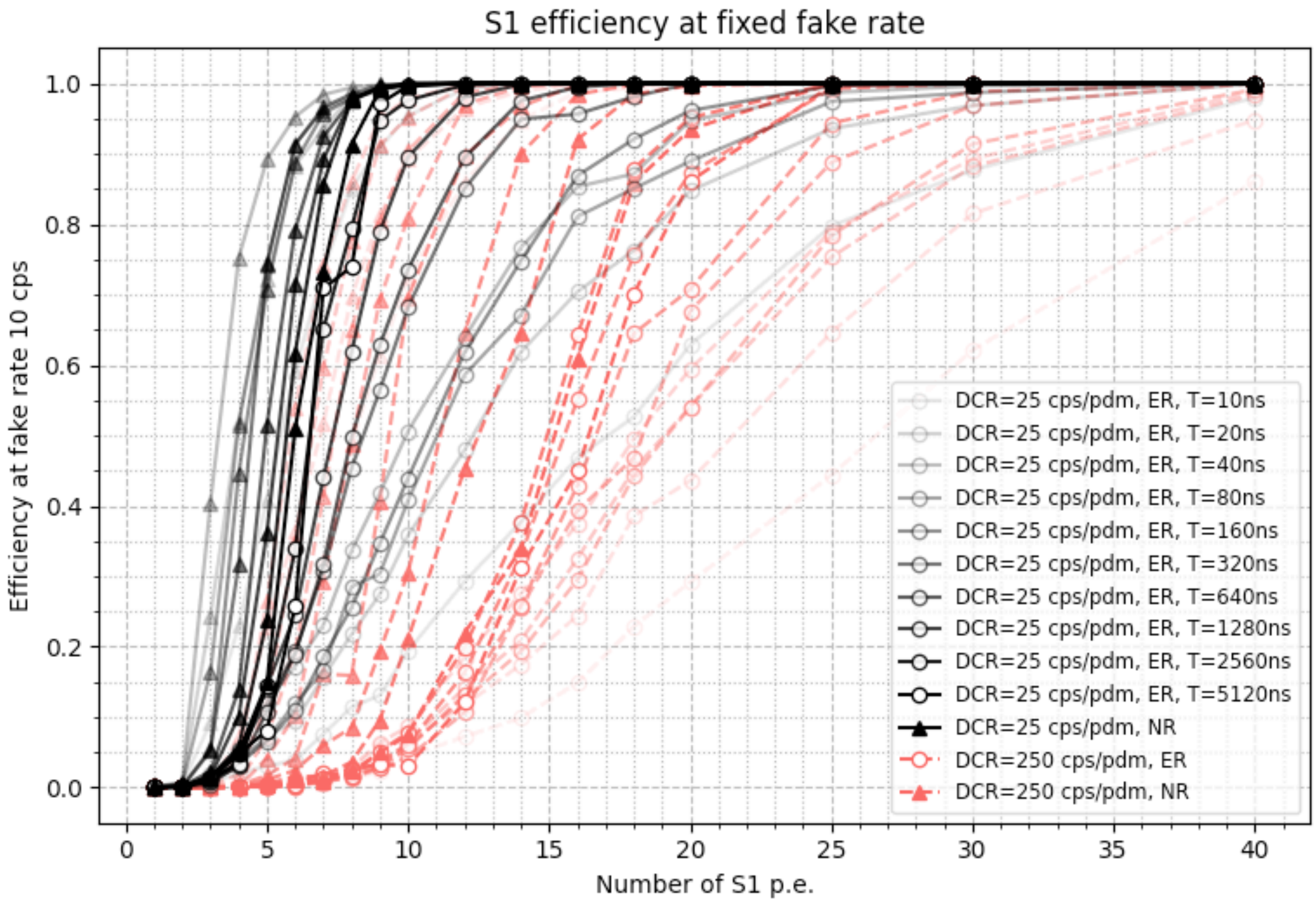


Coincidence optimization (1/2)

The 200 ns time for the coincidence was from the TDR. However we expect to need different times to optimize the filter, a short one for NR and a long one for ER.

So we repeat the simulation for a range of coincidence times shown in the legend, plotting efficiency vs. detected photons at fixed fake rate 10 cps.

This figure is a mess, so we take the point were the efficiency reaches 50 % for each curve (see next slide).

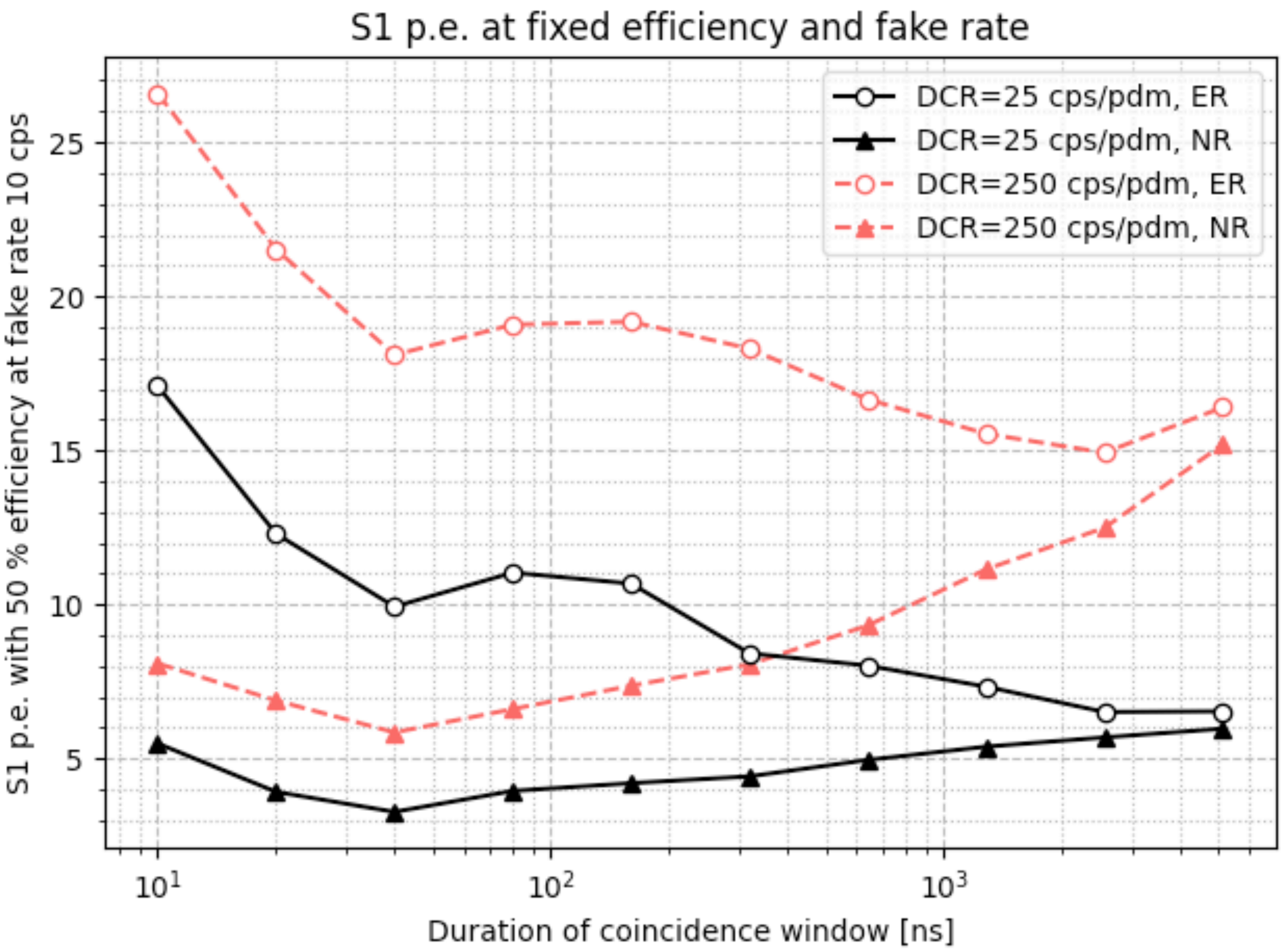


Coincidence optimization (2/2)

This plot is: x = length of coincidence time window, y = number of p.e. where the efficiency crosses 1/2.

So we want to look at the minima: it's T = 40 ns for NR and T = 2560 ns for ER.

(Technical note: all the simulations I'm showing which include a coincidence filter have dead radius at least two times the maximum coincidence duration used, instead of the default 2 microseconds.)



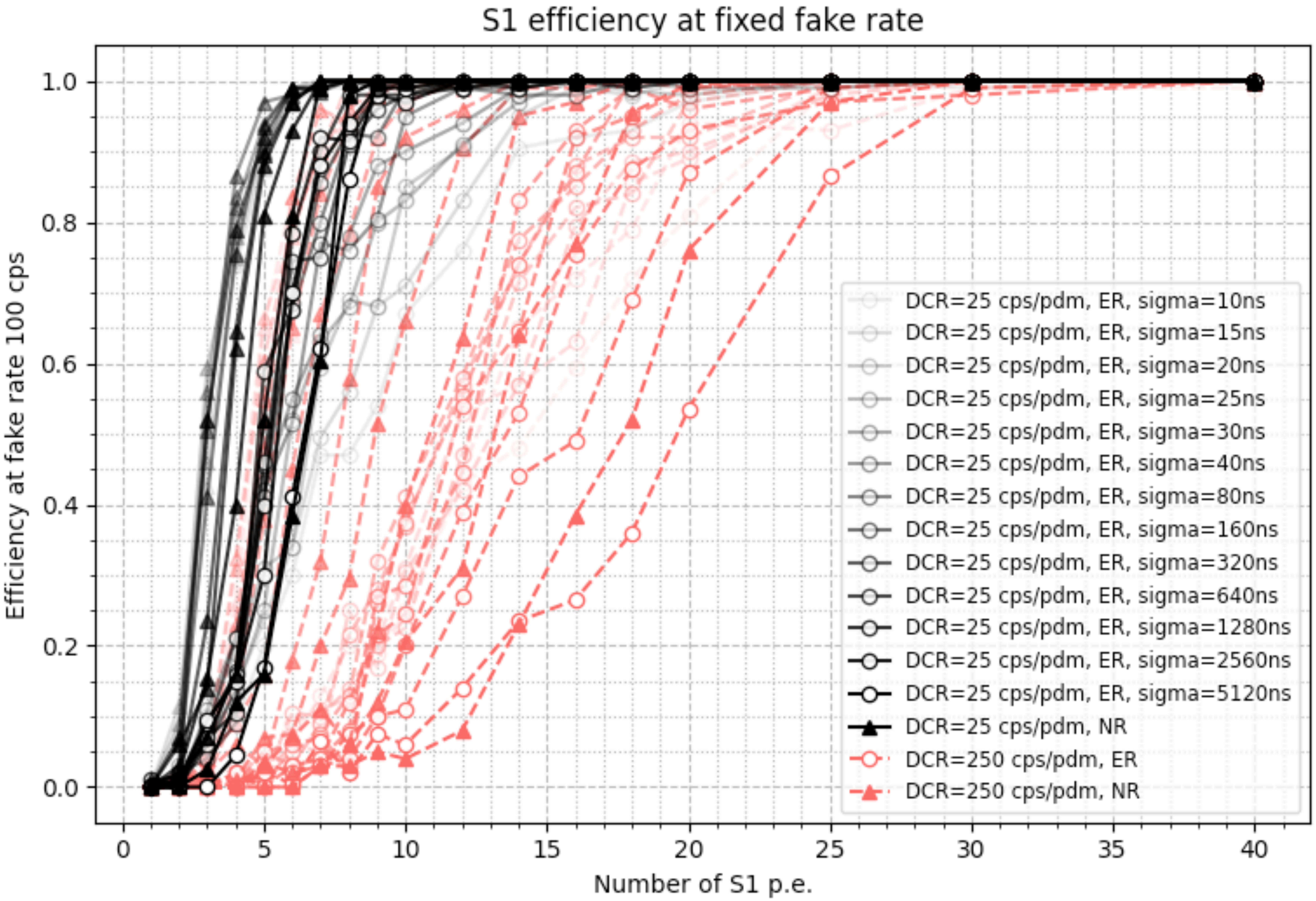
KDE bandwidth optimization (1/2)

We repeat the procedure we did to optimize the coincidence time to optimize the gaussian term scale of the cross correlation template.

(So the sigma in the legend is sqrt(sigma^2 + sigma_KDE^2).)

(Recall the sigma of the generated hits is 10 ns.)

See plot next slide.

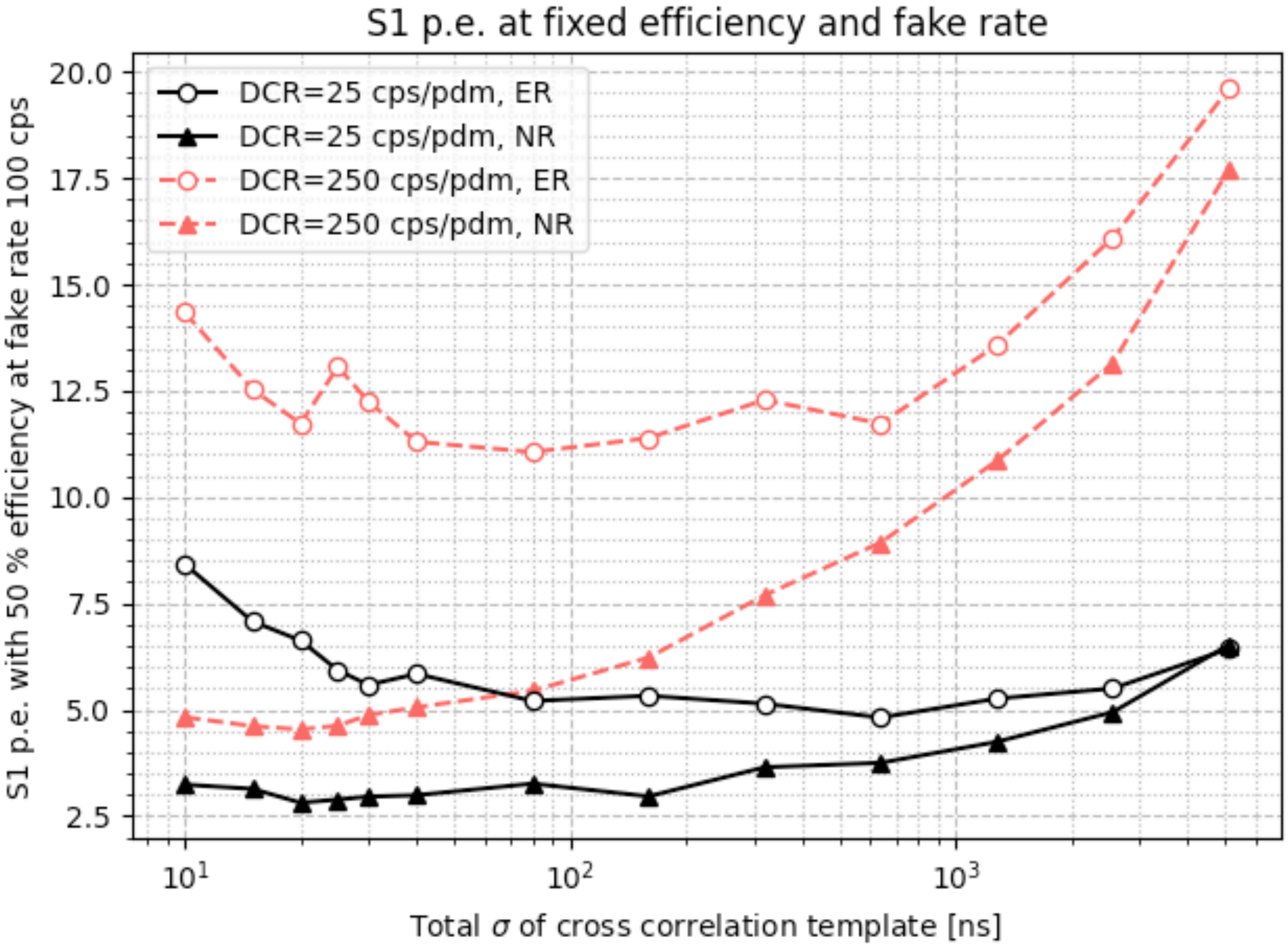


KDE bandwidth optimization (2/2)

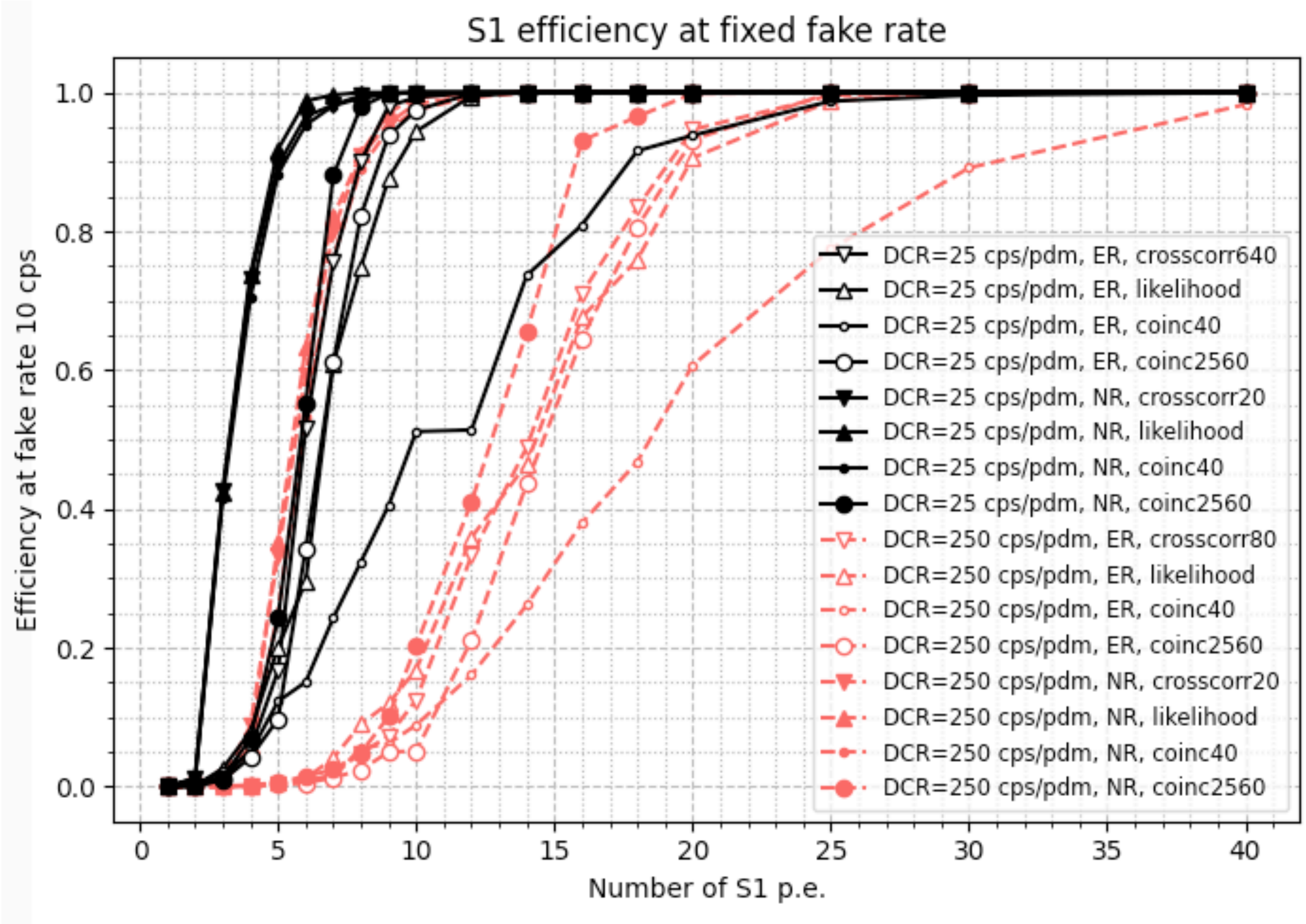
Result: for NR use sigma=20 ns, for ER use sigma=80 ns at high DCR, 640 ns at low DCR.

(I've redone this plot varying crossing efficiency and fake rate, the result is stable.)

(I've used rate=100 cps instead of 10 cps just in this case to save on simulation time.)



Efficiency vs. number of photons



The DCR makes a large difference of course.

For NR all is good (apart from the long coincidence).

For ER cross correlation is the best, then long coincidence, then likelihood.

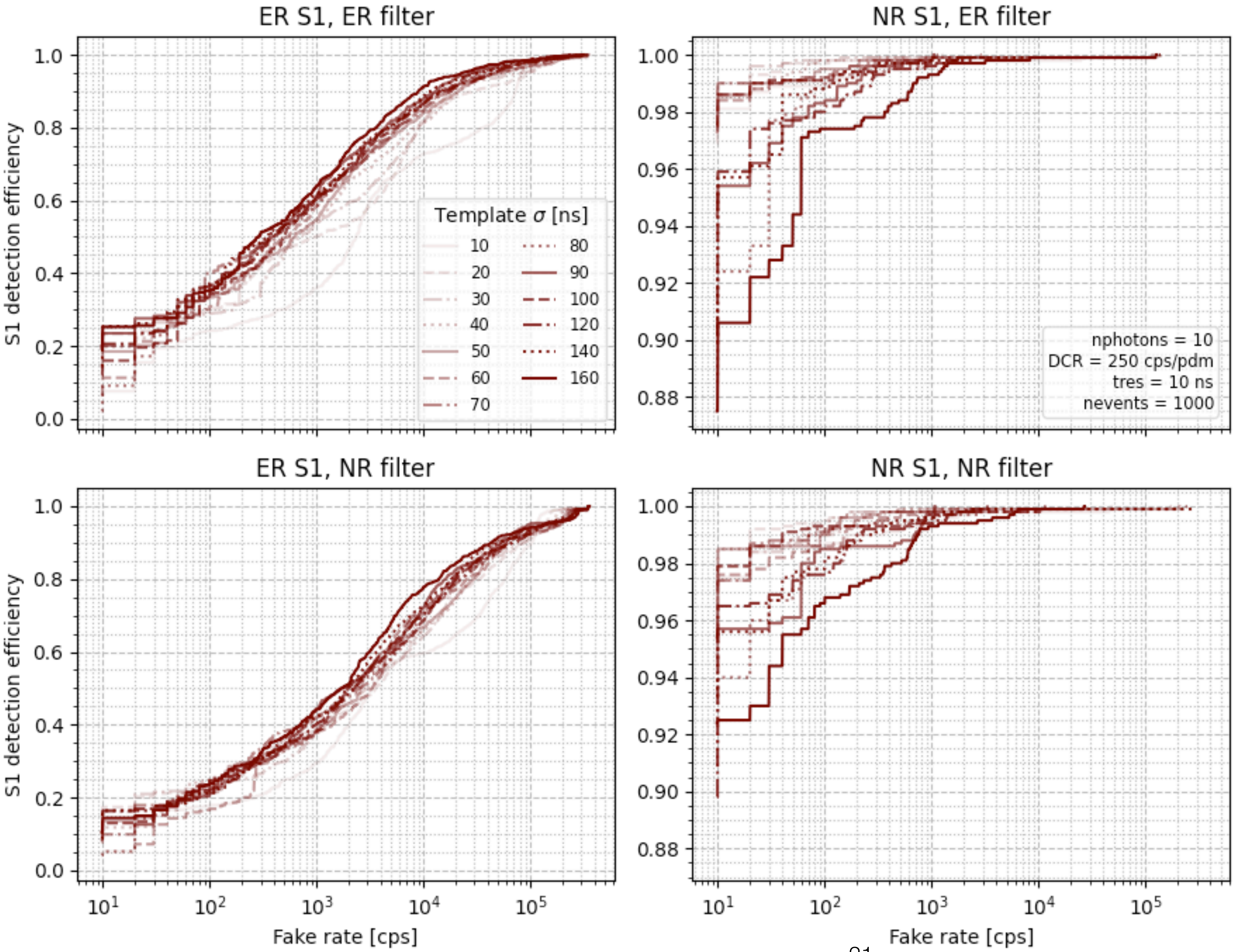
Conclusions

- Cross correlation is better, but only slightly, than simple coincidence, probably not worth it.
- Use a combination of short and long coincidence times.

Problems/limitations of this study:

- Not fitting the number of p.e. in the likelihood. I guess it would make it worse.
- Fake rate is 10 cps. I don't think the comparison would change with lower fake rate, but I'm not sure.
- Maybe the cross correlation would make a difference on ER/NR discrimination. I think probably not, but I'd have to check.
- The weird way I'm evaluating the cross correlation filter.

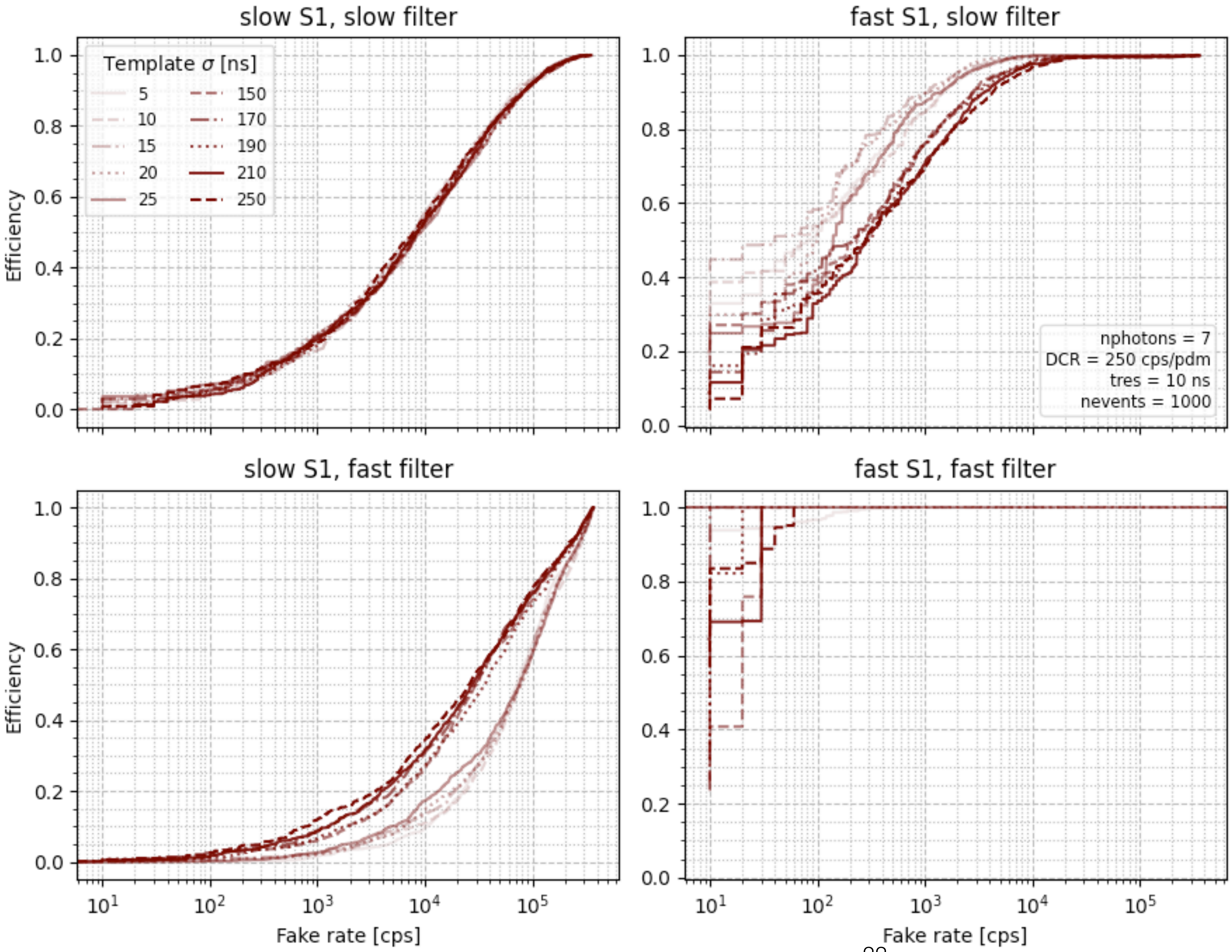
ER/NR (1/2)



Simulate ER and NR and filter with both ER and NR templates.

We don't try to define a discrimination criterion, just look at the efficiency curves.

We also try to change the template sigma.



ER/NR (2/2)

We simulate only fast/ only slow photons and filter with only fast/only slow pdf

May be better for optimization of sigma parameter

Maybe it's just a mess