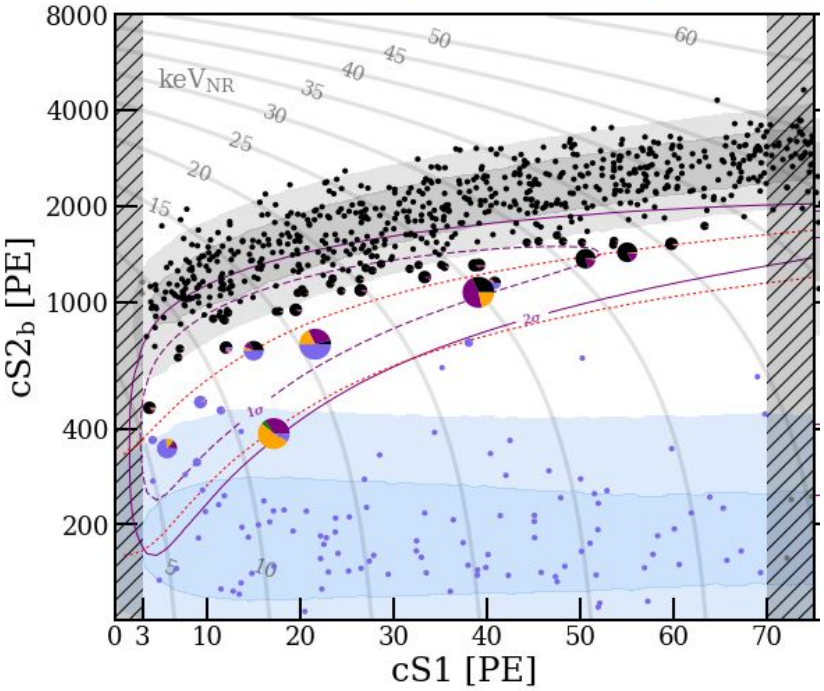
Fast likelihoods in more dimensions for LXe TPCs

Jelle Aalbers

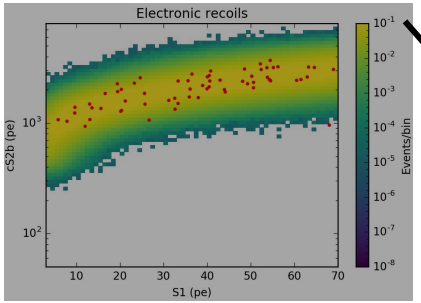
In collaboration with Bart Pelssers and Cristian Antochi

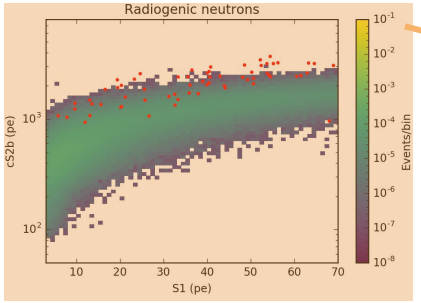
28 October 2019

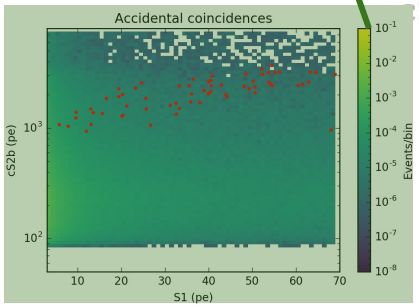
https://github.com/FlamTeam/flamedisx

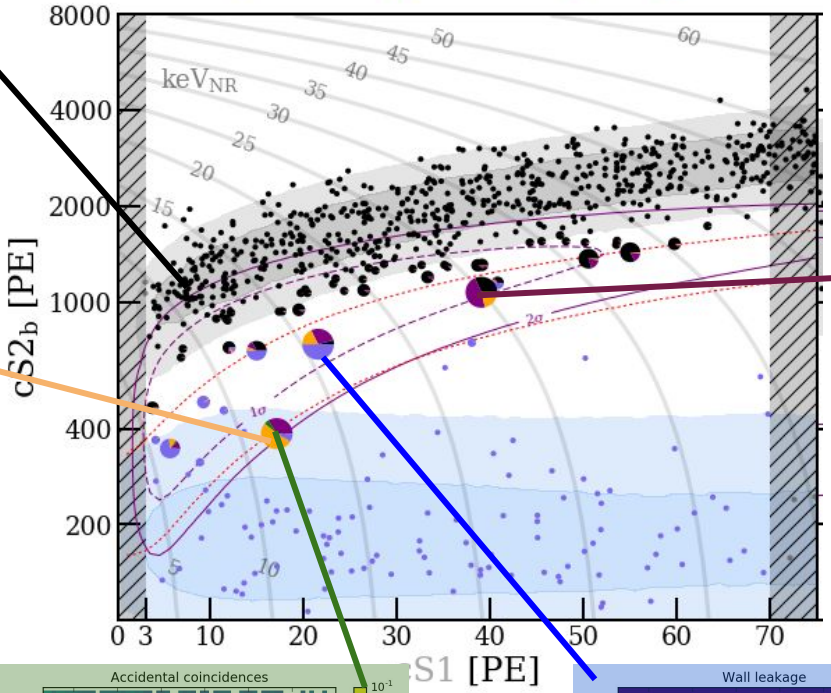


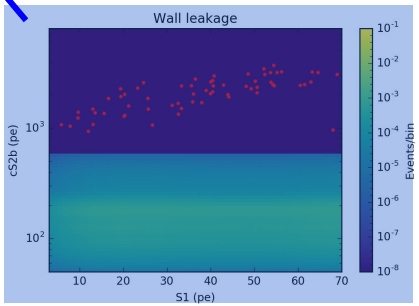
Likelihoods use densities / differential rates.
We histogram simulations to estimate these.

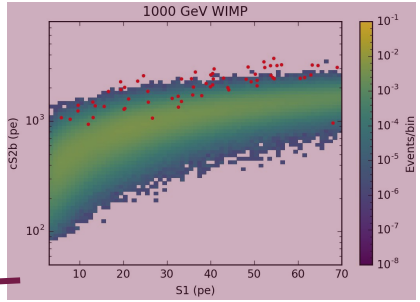


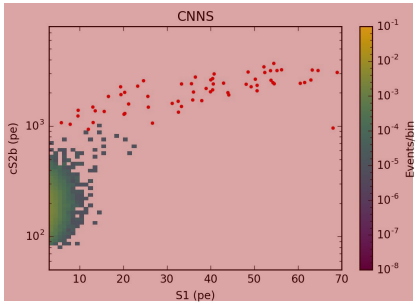








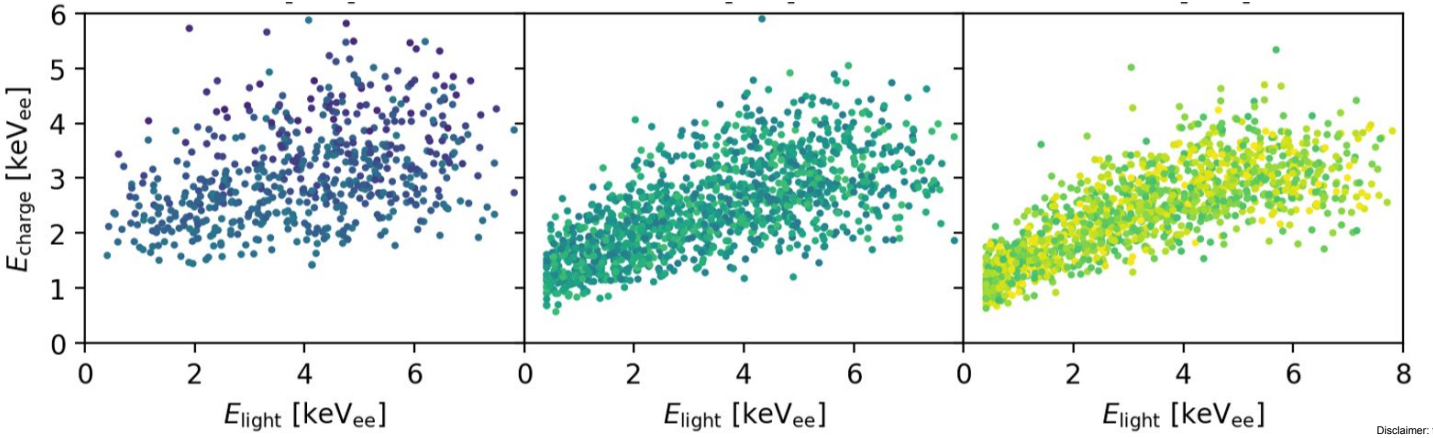




Disclaimer: this display is for illustration only. The per-component background plots are from our first results paper, not our main results paper (as you can see, they have far fewer events).

Detector response varies. Cannot "correct" fluctuations.

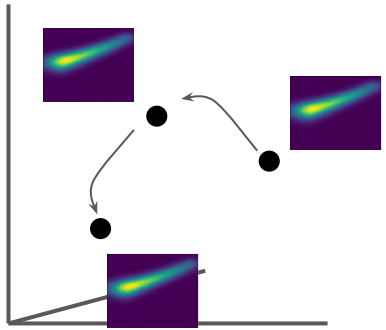
High-dimensional densities (S1, S2, x y z t) give best sensitivity.



Disclaimer: for illustration, I chose a dreadful electron lifetime (200 us).

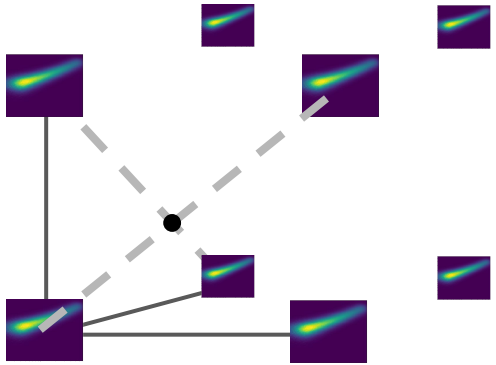
Simulation-based densities allow few nuisance parameters.

On-demand simulation



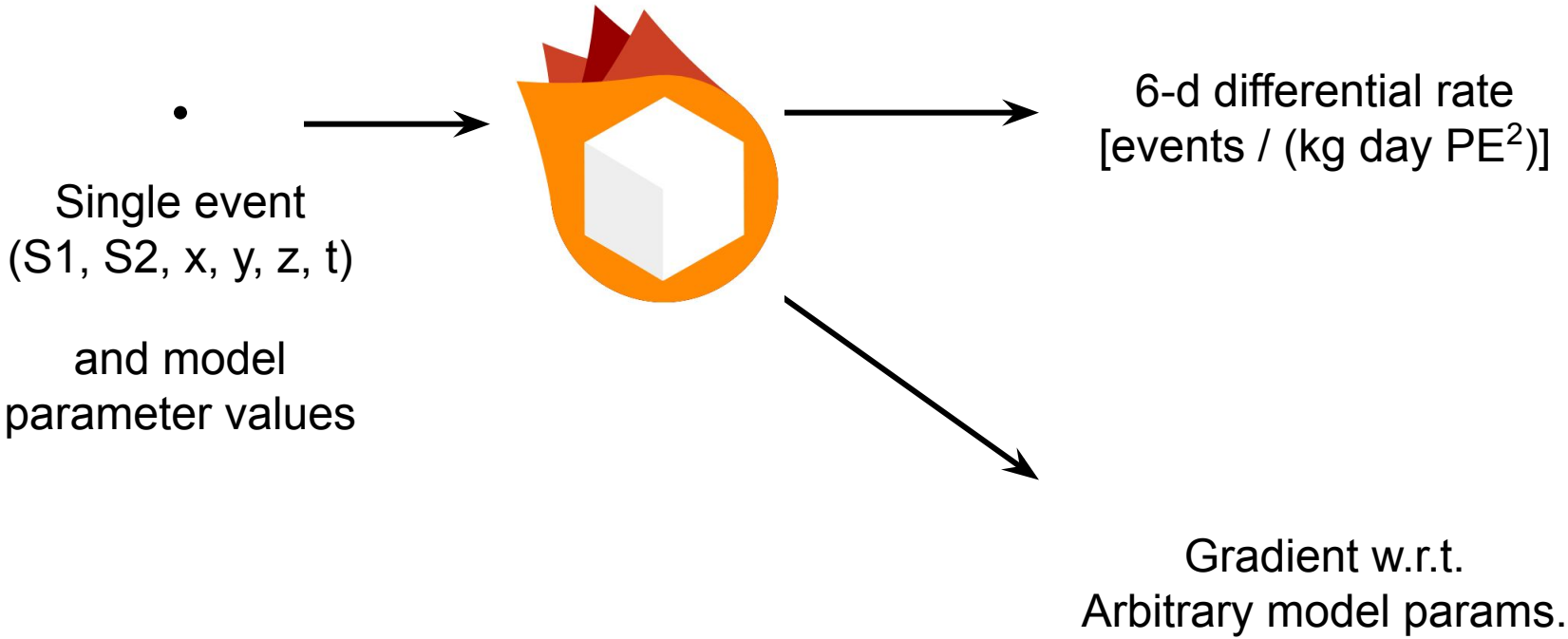
Many simulator calls, no gradient
Difficult and expensive to find minimum

Precompute and interpolate

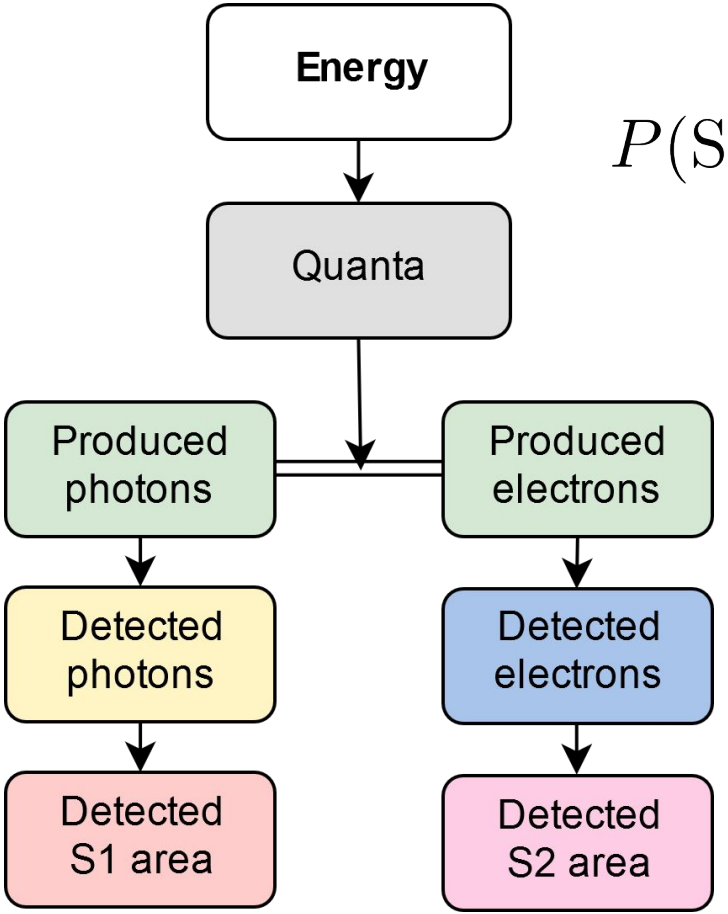


Need ~ $(2 - 7)^{\#\text{params}}$ templates
Interpolation is inaccurate

Flamedisx: replace the simulator with an *equivalent* integral.



The LXe emission model can be compactly expressed.



$$P(\mathrm{S1},\mathrm{S2}) = \int dE \sum_{n_\mathrm{q}} \sum_{n_\mathrm{ph}^\mathrm{det}} \sum_{n_\mathrm{ph}^\mathrm{prod}} \sum_{n_\mathrm{el}^\mathrm{det}} \sum_{n_\mathrm{el}^\mathrm{prod}}$$

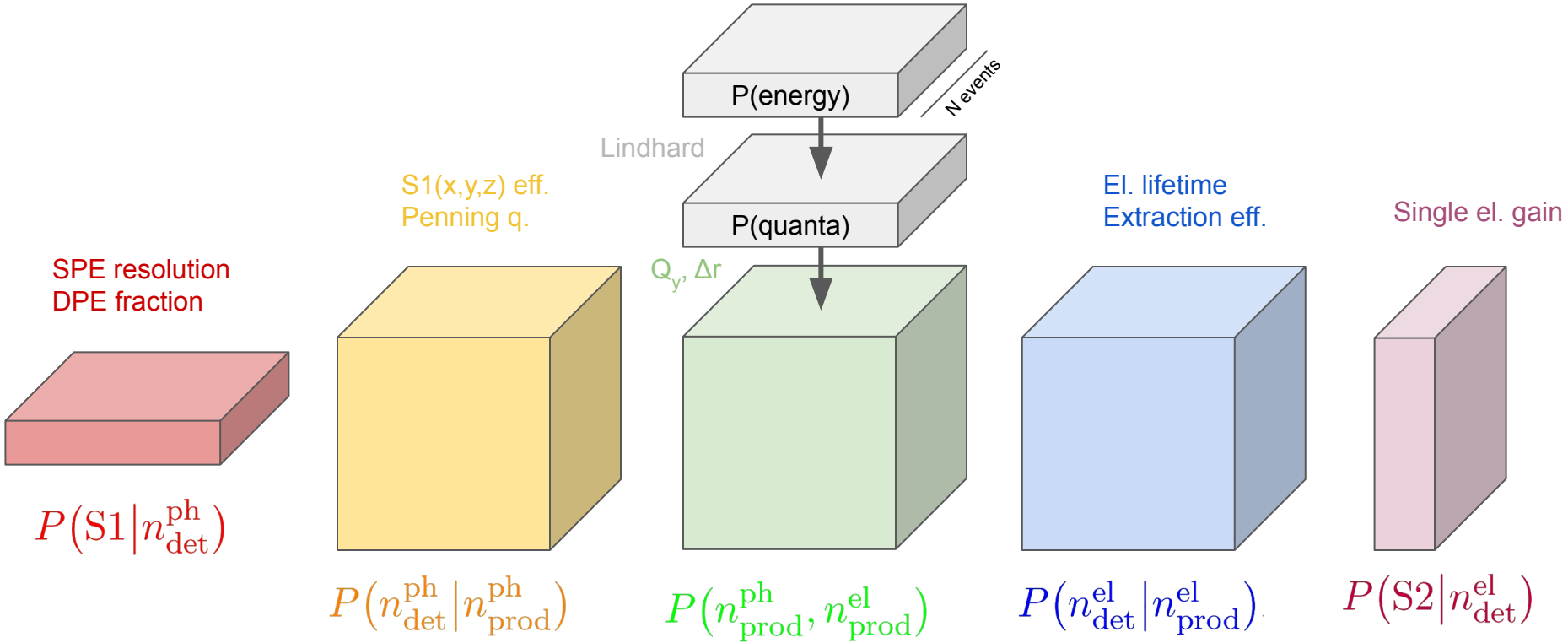
$$P(E)P(n_q|E)$$

$$P(\mathrm{S1}|n_\mathrm{ph}^\mathrm{det})P(n_\mathrm{ph}^\mathrm{det}|n_\mathrm{ph}^\mathrm{prod})$$

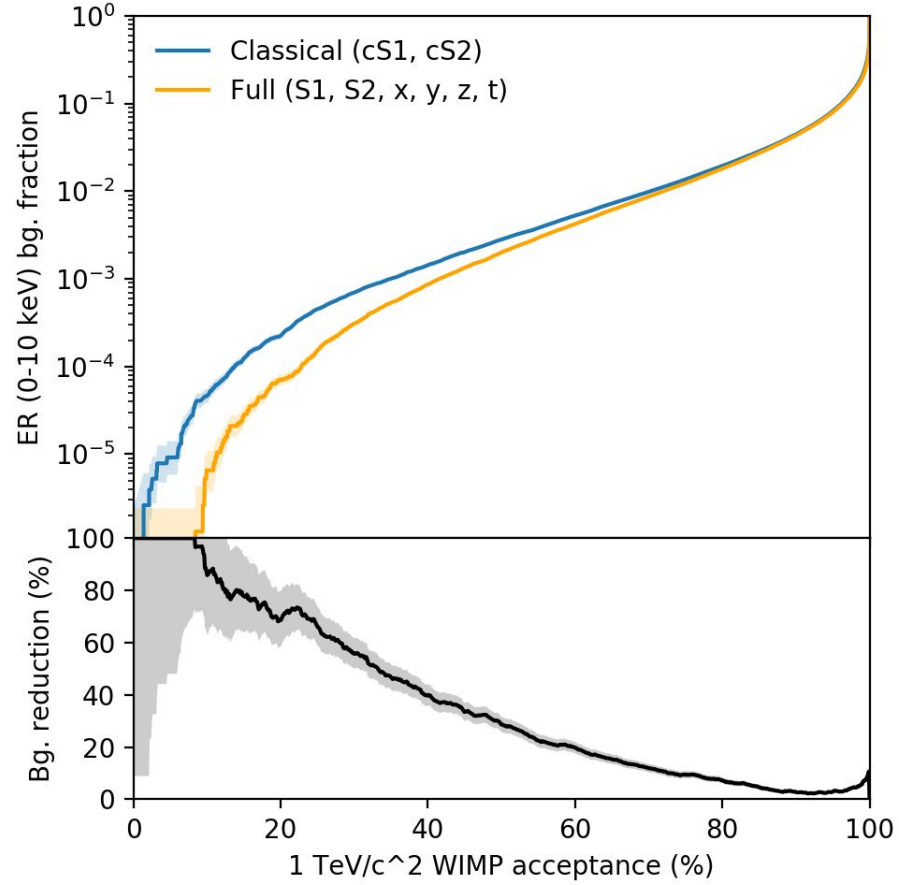
$$P(n_\mathrm{ph}^\mathrm{prod}, n_\mathrm{el}^\mathrm{prod}|n_q)$$

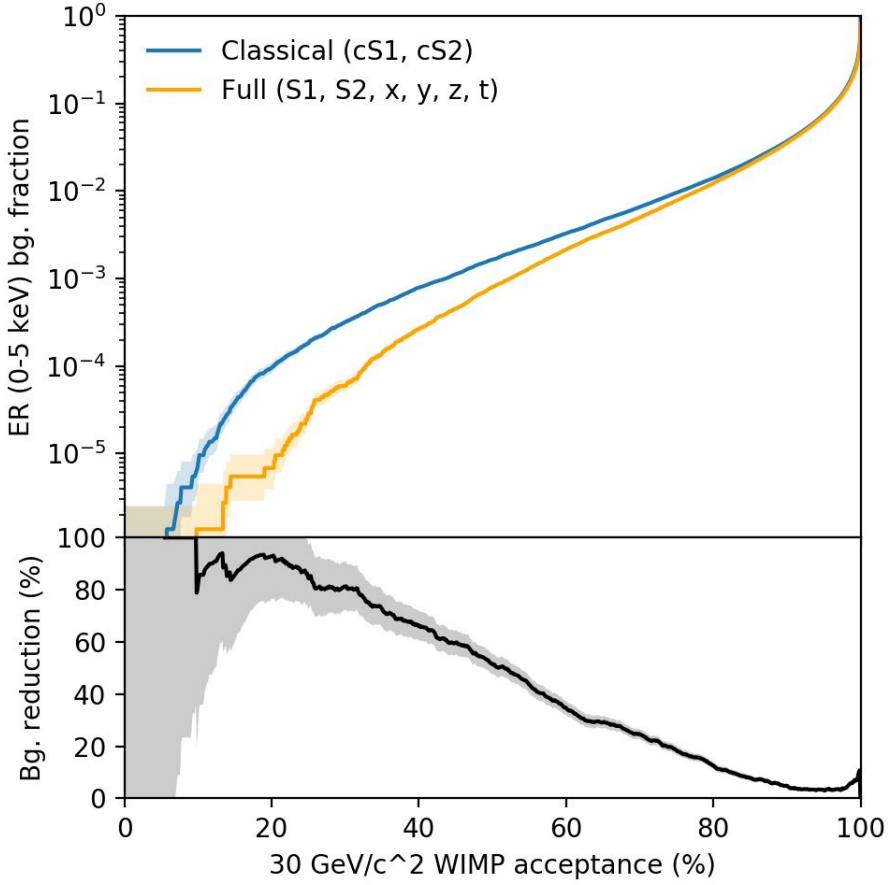
$$P(n_\mathrm{el}^\mathrm{det}|n_\mathrm{el}^\mathrm{prod})P(\mathrm{S2}|n_\mathrm{el}^\mathrm{det})$$

TensorFlow implementation for gradients and GPU support.



2x reduction of internal background compared to 2D model.





Flamedisx

Replace LXe emission simulation with an integral

TensorFlow computes it differentiably on a GPU

More dimensions / less discretization: better sensitivity

More nuisance parameters: more robust result

Open-source, paper in preparation

https://github.com/FlamTeam/flamedisx

Challenges

Convincing people high-dimensional likelihoods are OK

- High-dimensional goodness-of-fit tests

Much CPU compute allocated, little GPU

- Using Google’s CoLaboratory for development

TensorFlow 2 only just released

- No conda packages or stackoverflow documentation