Optimisation of fast likelihood functions for dark matter and rare event searches

Author: Joshua Robert Green
LUX-ZEPPLIN (LZ) @ University of Oxford

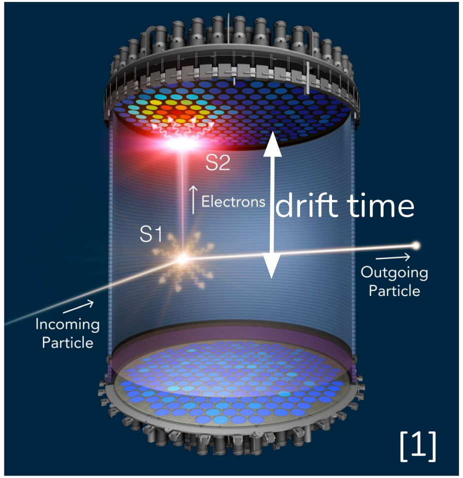
Particular thanks: Robert James
for introduction to this topic and continued help

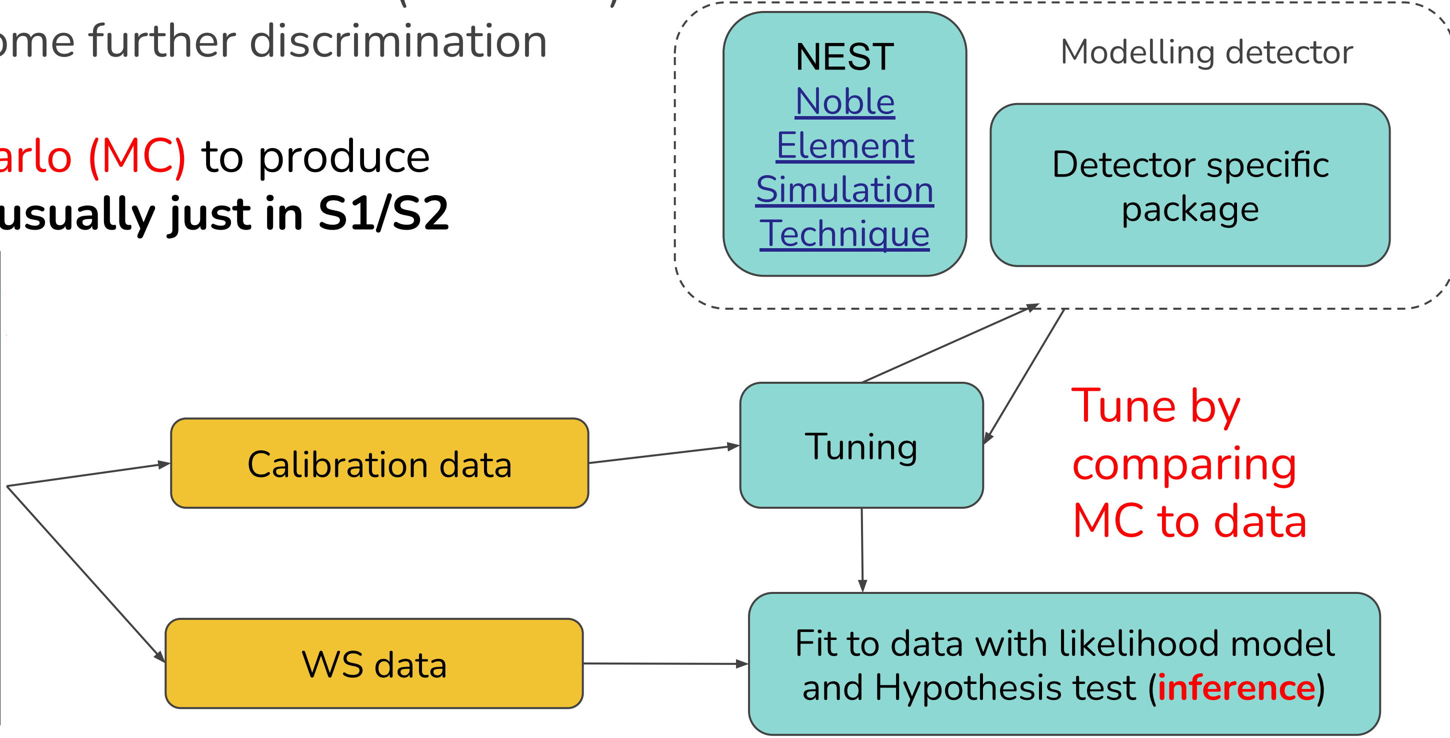
How does LUX ZEPLIN (LZ) search for WIMPs?

Observables:

S2/S1 sizes - Electron vs Nuclear Recoil(ER vs NR)
Radius/drift-time - some further discrimination

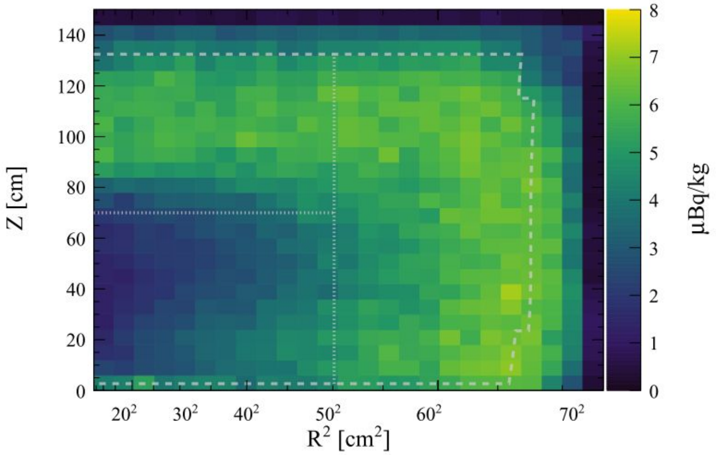
Use Models Monte Carlo (MC) to produce probability functions **usually just in S1/S2**



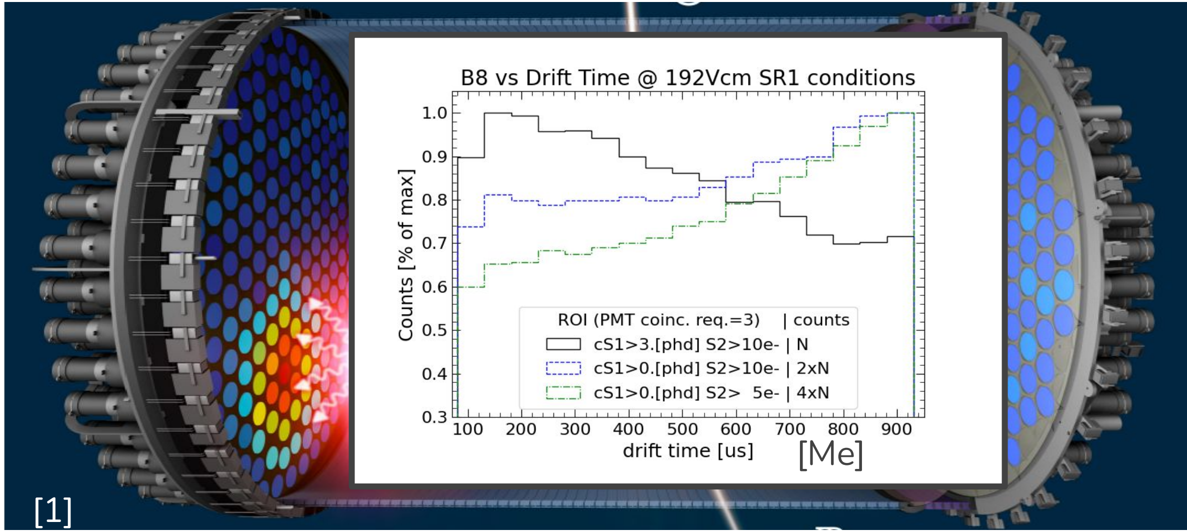


Why have a multidimensional model?

- Backgrounds: Inferred spatial distribution of dominant background of lead-214, tagged by its progenitor polonium.
- Detector effects: Low energy NRs like 8B solar neutrinos coherent nuclear scatters have drift time dependence from light collection efficiencies



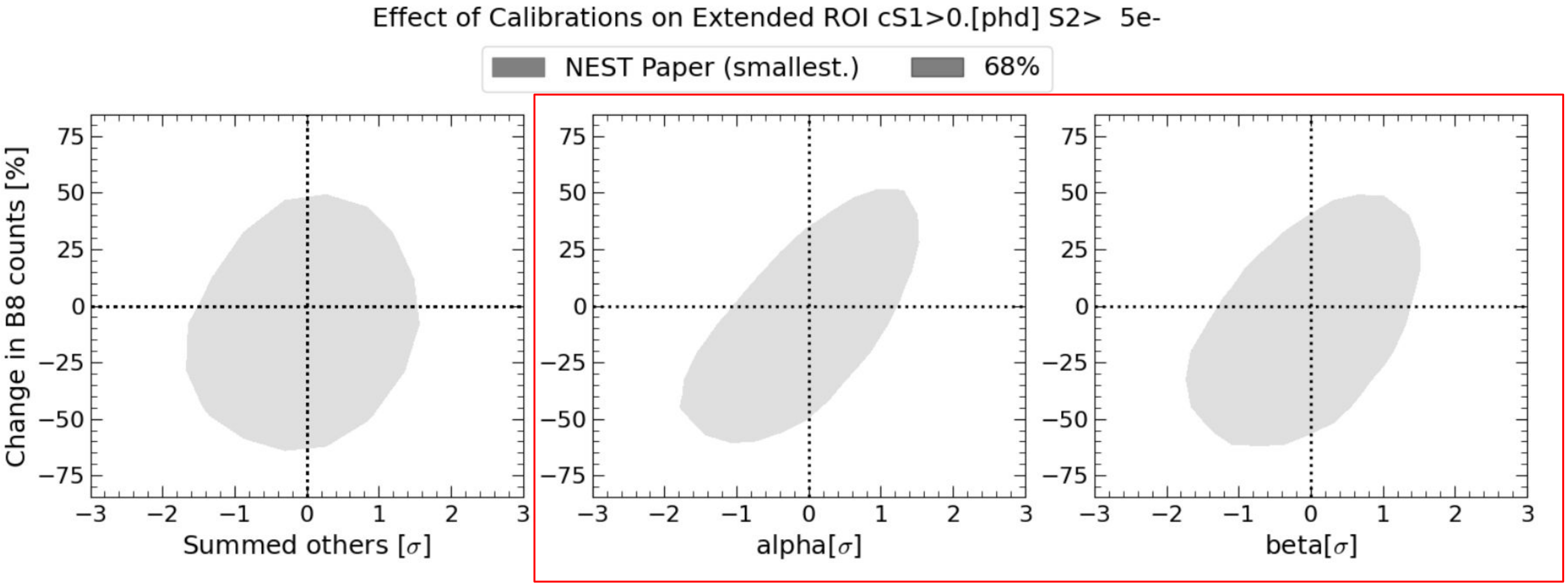
[2] (a) Observed ^{218}Po Distribution



[1]

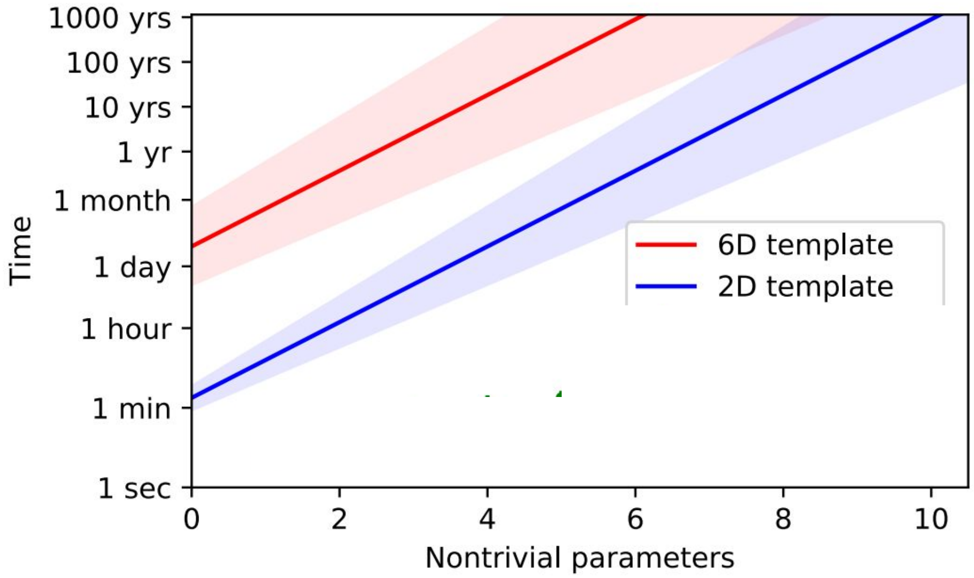
Why have a model with shape varying parameters?

- Acceptance driven by shape variation around boundary cuts
- Significant shape uncertainty gives rate uncertainty
- Default NEST parameters' uncertainty are significance
    - Calibrations tell us more than this!



Why only S1/S2?

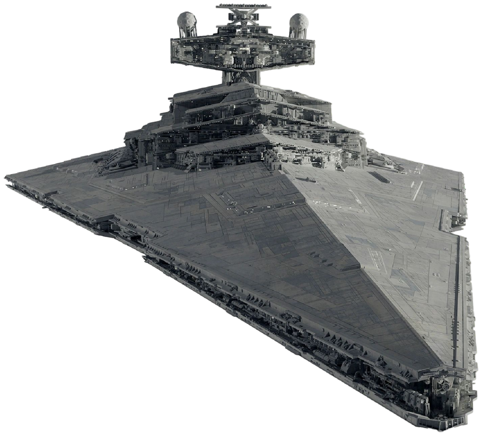
If we want full multidimensional fits w or w/out shape varying nuisance parameters templates won't cut it but flamedisx will



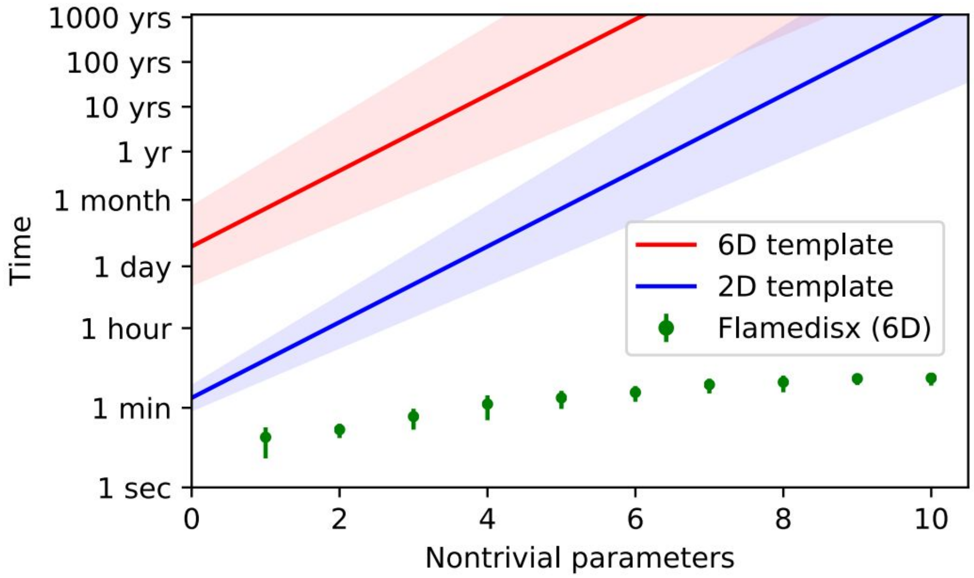
[3] Fitting to O(1000) ER data set.
7 templates for each parameter (anchor points)

Why only S1/S2?

If we want full multidimensional fits w or w/out shape varying nuisance parameters templates won't cut it but flamedisx will



Imperial I class start destroyer

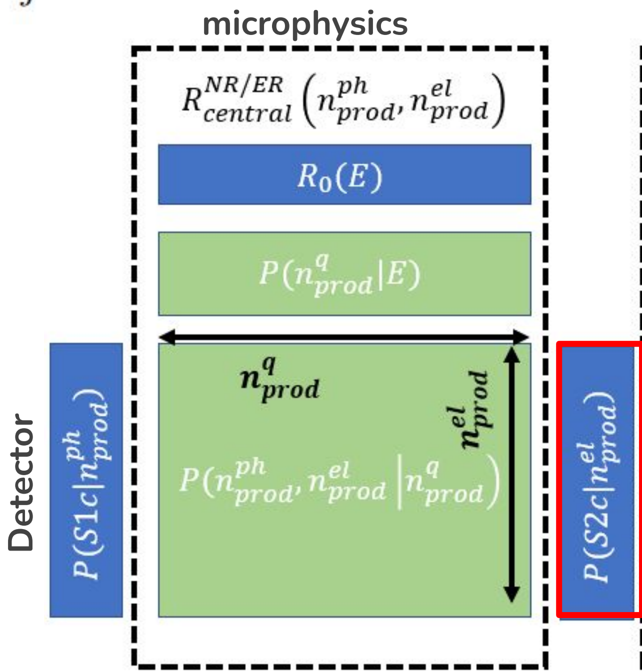


[3] Fitting to O(1000) ER data set.
7 templates for each parameter (anchor points)

Using flamedisx

$$ln(L(\vec{\theta}, \{\vec{s_i}\})) = -\mu(\vec{\theta}) + \sum_{i}^{events} ln(\sum_{j}^{sources} R^j(\vec{\theta}, \vec{s_i})) + const.$$

Guess all underlying parameters that significantly contribute to data

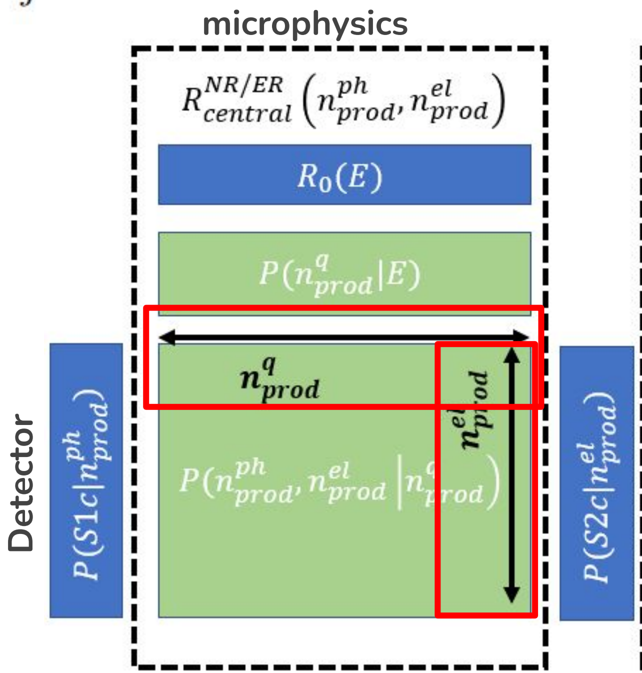


Using flamedisx

$$ln(L(\vec{\theta}, \{\vec{s_i}\})) = -\mu(\vec{\theta}) + \sum_{i}^{events} ln(\sum_{j}^{sources} R^j(\vec{\theta}, \vec{s_i})) + const.$$

Guess all underlying parameters that significantly contribute to data

Explicitly evaluate the **differential rate** on those parameters



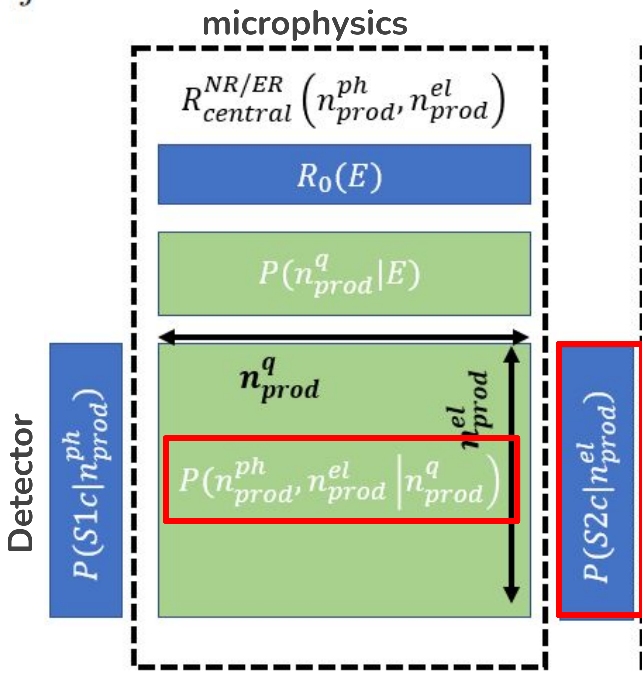
Using flamedisx

$$ln(L(\vec{\theta}, \{\vec{s_i}\})) = -\mu(\vec{\theta}) + \sum_{i}^{events} ln(\sum_{j}^{sources} R^j(\vec{\theta}, \vec{s_i})) + const.$$

Guess all underlying parameters that significantly contribute to data

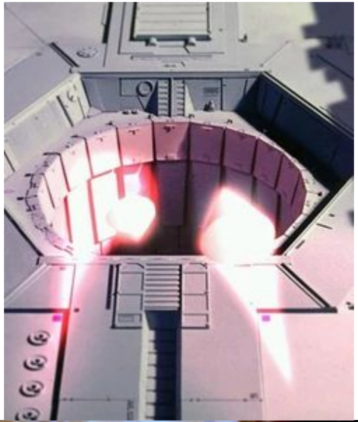
Explicitly evaluate the **differential rate** on those parameters

Treat as tensor operation and utilise differentiable programming

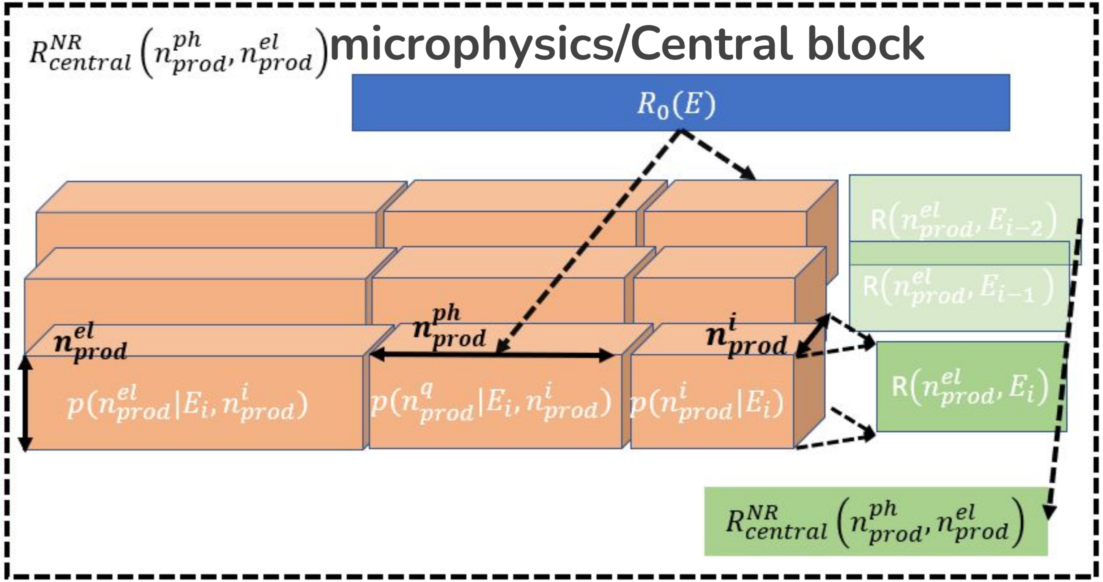


Implementing NEST -> FlameNEST [4]

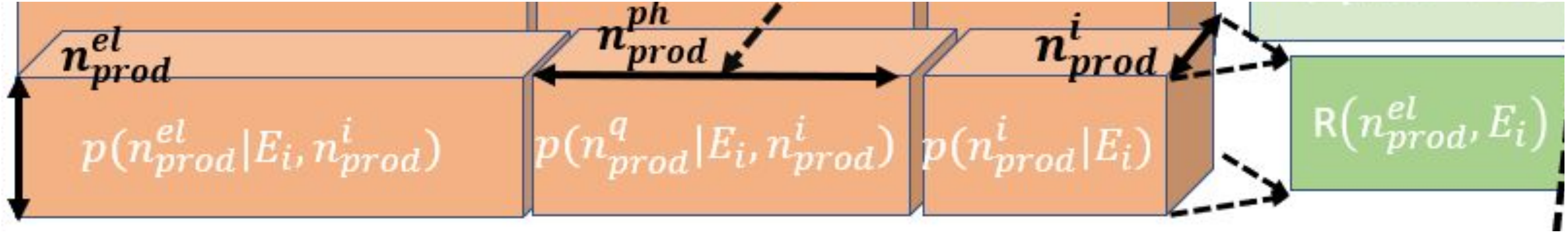
Convoluted yield models



Developers implemented the NEST models and caused performance issues

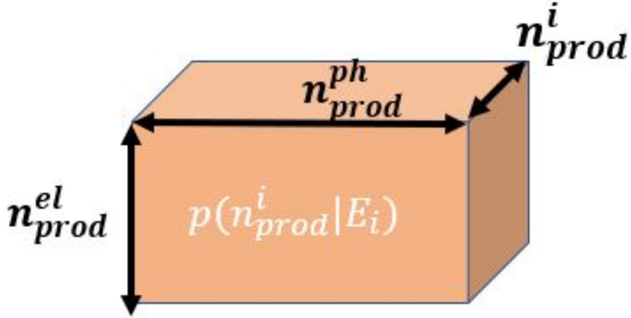


Fixing the problem



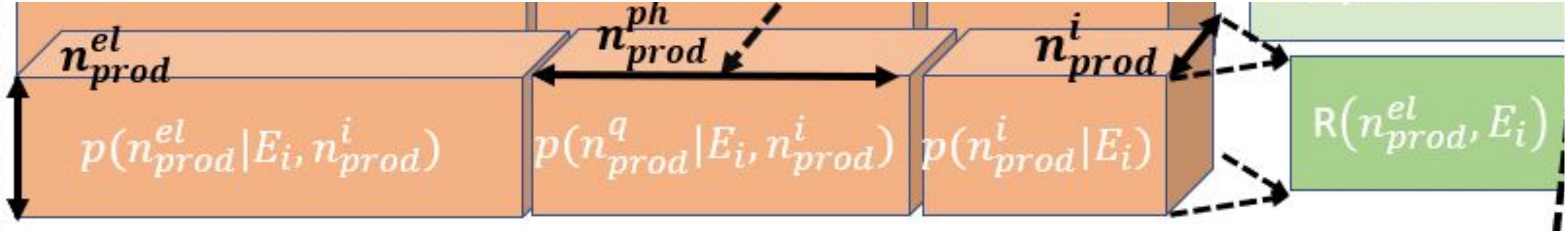
- Each block represents a tensor
- Each dimension of the block is the range of underlying parameters
- Each function is evaluated for every element of that block

Degenerate dimensions



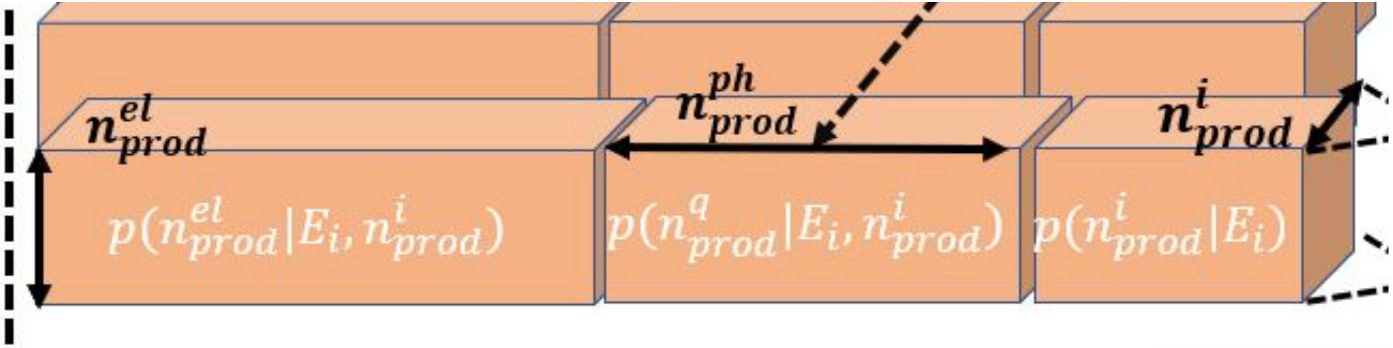
- The model function that represents recombination only depends on ions produced
- It is being evaluation on a tensor of ions, photons, and electrons
- Many degenerate evaluations of the model

Degenerate dimensions



- Every model here has degenerate dimensions in this way
- Each evaluation of a function in differentiable programming represents a graph of primitive functions
- Consumes a lot more memory than just the value of the function

Fixing this problem



- To fix this problem I **carefully** implemented **unique and gather** to calculate the model functions.
  - Careful as these functions can cause performance issues
  - **Only use when significantly reduces degeneracy**
- Photons not explicitly in the model but quanta=photons+electrons

Explicit profiling results

Differential Rates	Before: Peak Memory (GiBs)	Before: Trace time (MM:SS)	Before: ex.time (SS)	Before: Top Operations	After: Peak Memory (GiBs)	After: Trace time (MM:SS)	After: ex.time (SS)	After: Top Operations
det.param g1 batch size 5	12.0	02:05	02	yield tfp functions tensor ops	2.0	08:16	02	gather/tensordot tensor ops
yield.param α batch size 1	28.0	04:13	02	yield tfp functions and gradients of them	1.0	04:25	01	gather/tensordot tensor ops

Reduction of 6/28x of memory usage for detector/yield parameters.

- =6/28x **speed up** as can processes more events simultaneously

Memory dominated by tensor manipulation instead of model functions

- Weaker scaling of memory with parameters= can float many more parameters

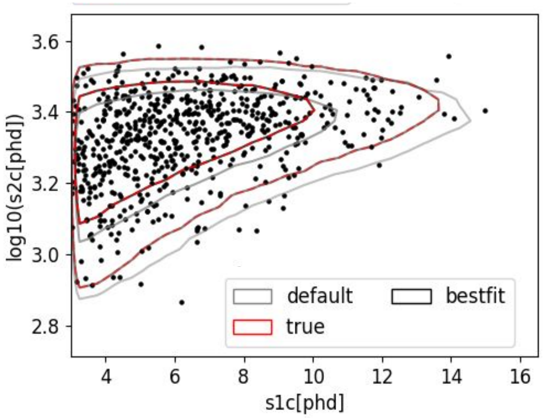
Tracing time does increase but execution time same/smaller

- Negligible as long as batch size<< data size.

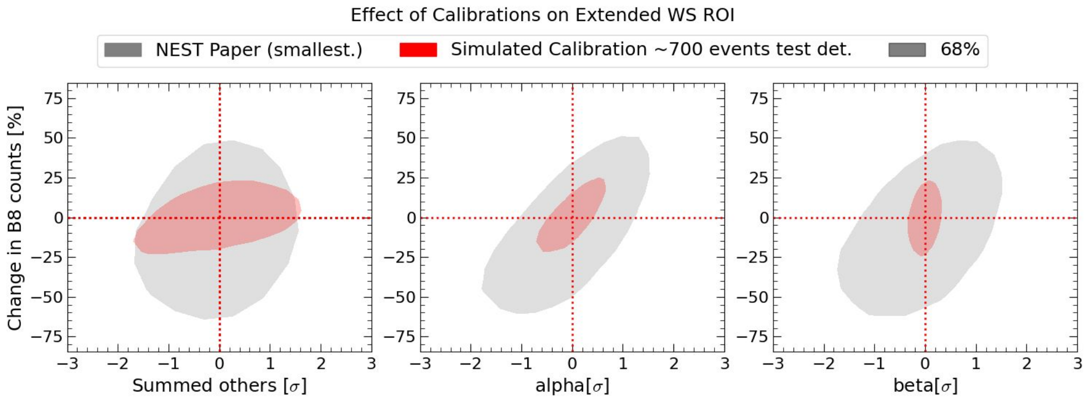
Testing with simulated detector (public LZ information)

- Using a test low energy flat nuclear recoil source:
- Time: 11-14mins to fit
  - 30mins to generate total rate estimator.
- **Accurately finds the distribution!**

- Auto-differentiation gives covariance and uncertainties at bestfit
- Significant constraints with just few number of points

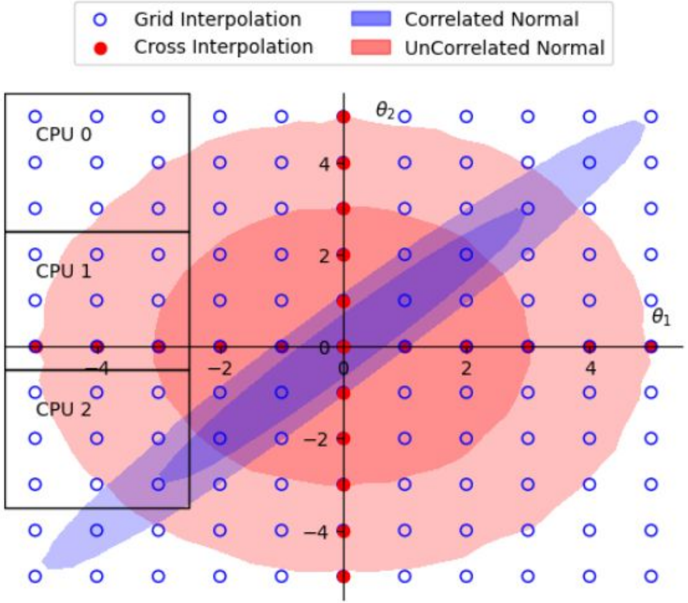


	predicted	True	bestfit	scale
$\mu_{r.m}$	n/a	2.00	2.00 ± 0.09	10^0
α	1.10 ± 0.05	1.11	1.11 ± 0.02	10^1
β	1.10 ± 0.05	1.08	1.08 ± 0.01	10^0
ε	1.26 ± 0.29	1.07	1.04 ± 0.03	10^1



Why is it incomplete?

$$ln(L(\vec{\theta}, \{\vec{s}_i\})) = -\mu(\vec{\theta}) + \sum_{i}^{events} ln(\sum_{j}^{sources} R^j(\vec{\theta}, \vec{s}_i))$$



My work focused on the **differential rate** term

Evaluate the **total rate** using simulations of fixed points and interpolate

$$\mathcal{O}(n_{parameters} \times n_{anchors})$$

Still only need total counts so better than full templates

$$\mathcal{O}(n_{anchors}{}^{n_{parameters}})$$

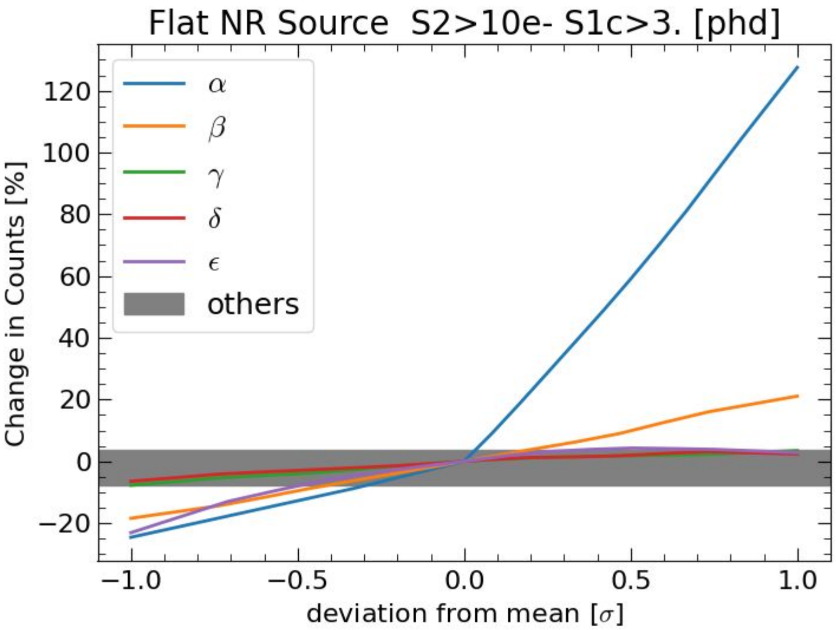
Solution: for now

Pick the three biggest impacts

Lots of solutions to explore:

Yield functions are easy to evaluate so:

- Multi-level simulations
- Multi-fidelity simulations
- Creating a better grid
- Reparameterize the model



Or explicit integration with better tools

$$\mu(\vec{\theta}) = \sum_j \mu^j(\vec{\theta}) \quad \mu^j(\vec{\theta}) \propto \int R^j(\vec{\theta}, \vec{s}) d\vec{s}$$

Conclusion

1. Hopefully my plot gore at the start convinced you that
    a. Position distributions are important
    b. Shape varying parameters are important
2. Flamedisx explicitly evaluates differential rate and allows for:
    a. Multidimensionality
    b. Shape varying parameters
3. Implementing NEST caused performance issues:
    a. **My work fixed those performance issues**
    b. Shown Multi-dimensional and parameter inside central block possible
4. Total rate estimators are the next challenge

LZ (LUX-ZEPLIN) Collaboration, 38 Institutions

@lzdarkmatter

https://lz.lbl.gov/

250 scientists, engineers, and technical staff

- Black Hills State University
- Brookhaven National Laboratory
- Brown University
- Center for Underground Physics
- Edinburgh University
- Fermi National Accelerator Lab.
- Imperial College London
- King's College London
- Lawrence Berkeley National Lab.
- Lawrence Livermore National Lab.
- LIP Coimbra
- Northwestern University
- Pennsylvania State University
- Royal Holloway University of London
- SLAC National Accelerator Lab.
- South Dakota School of Mines & Tech
- South Dakota Science & Technology Authority
- STFC Rutherford Appleton Lab.
- Texas A&M University
- University of Albany, SUNY
- University of Alabama
- University of Bristol
- University College London
- University of California Berkeley
- University of California Davis
- University of California Los Angeles
- University of California Santa Barbara
- University of Liverpool
- University of Maryland
- University of Massachusetts, Amherst
- University of Michigan
- University of Oxford
- University of Rochester
- University of Sheffield
- University of Sydney
- University of Texas at Austin
- University of Wisconsin, Madison
- University of Zürich

US Europe Asia Oceania

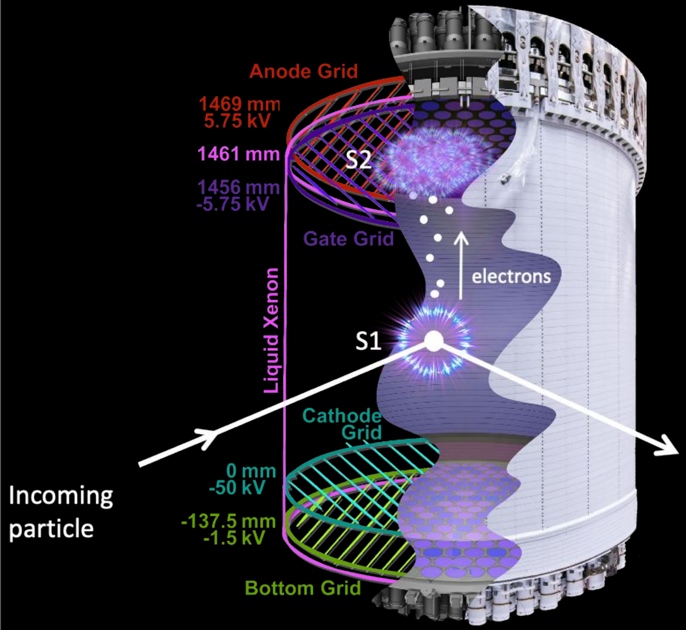
Science and Technology Facilities Council

FCT Fundação para a Ciência e a Tecnologia

Sanford Underground Research Facility
South Dakota Science and Technology Authority

Institute for Basic Science

Thanks to our sponsors and participating institutions!



Find more graphics here or directly contact Nicolas (Imperial)

Thank you!

Thanks to our sponsors and 38 participating institutions!

U.S. Department of Energy
Office of Science

Bibliography:

[1] LUX-ZEPLIN Technical Design report Arxiv:1703.09144

[2]- Background Determination for the LUX-ZEPLIN (LZ) Dark Matter Experiment : 10.1103/PhysRevD.108.012010

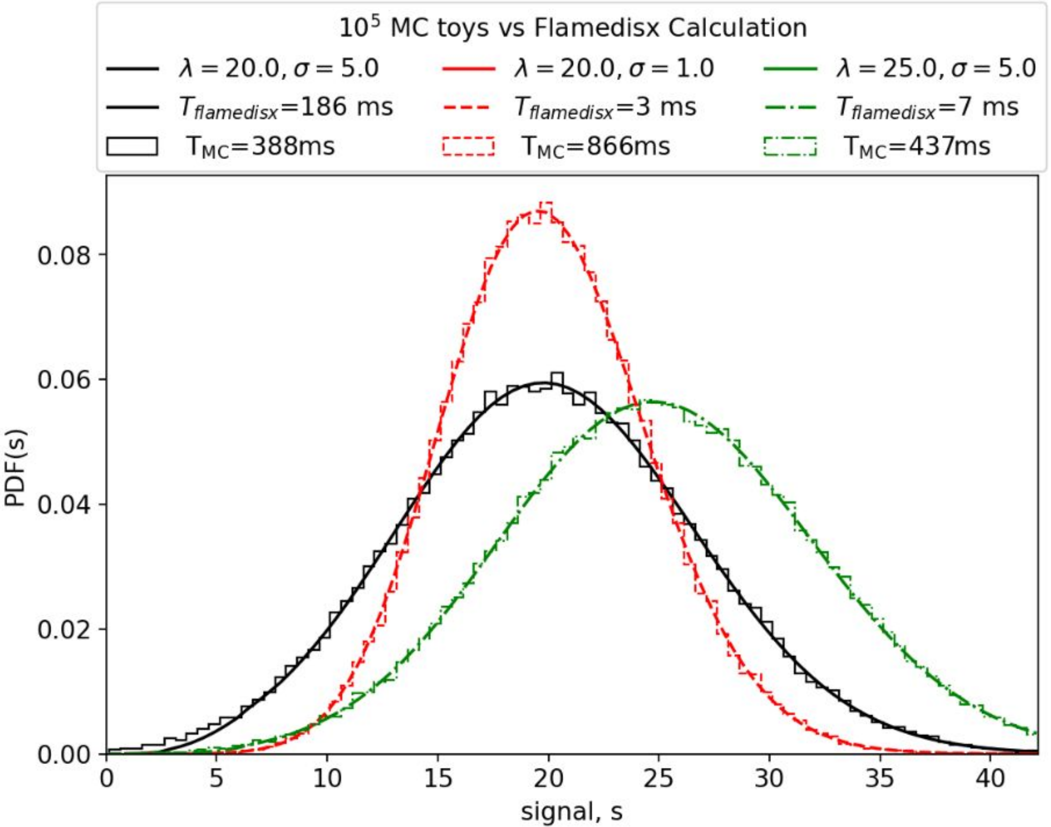
[3] Finding Dark Matter Faster with Explicit Profile Likelihoods 10.1103/PhysRevD.102.072010

[4] FlameNEST: explicit profile likelihoods with the Noble Element Simulation Technique 10.1088/1748-0221/17/08/P08012

Miscellania

Performance metric

$$P(s) = \sum_n Gaus(s|n,\sigma) \times Pois(n|\lambda)$$

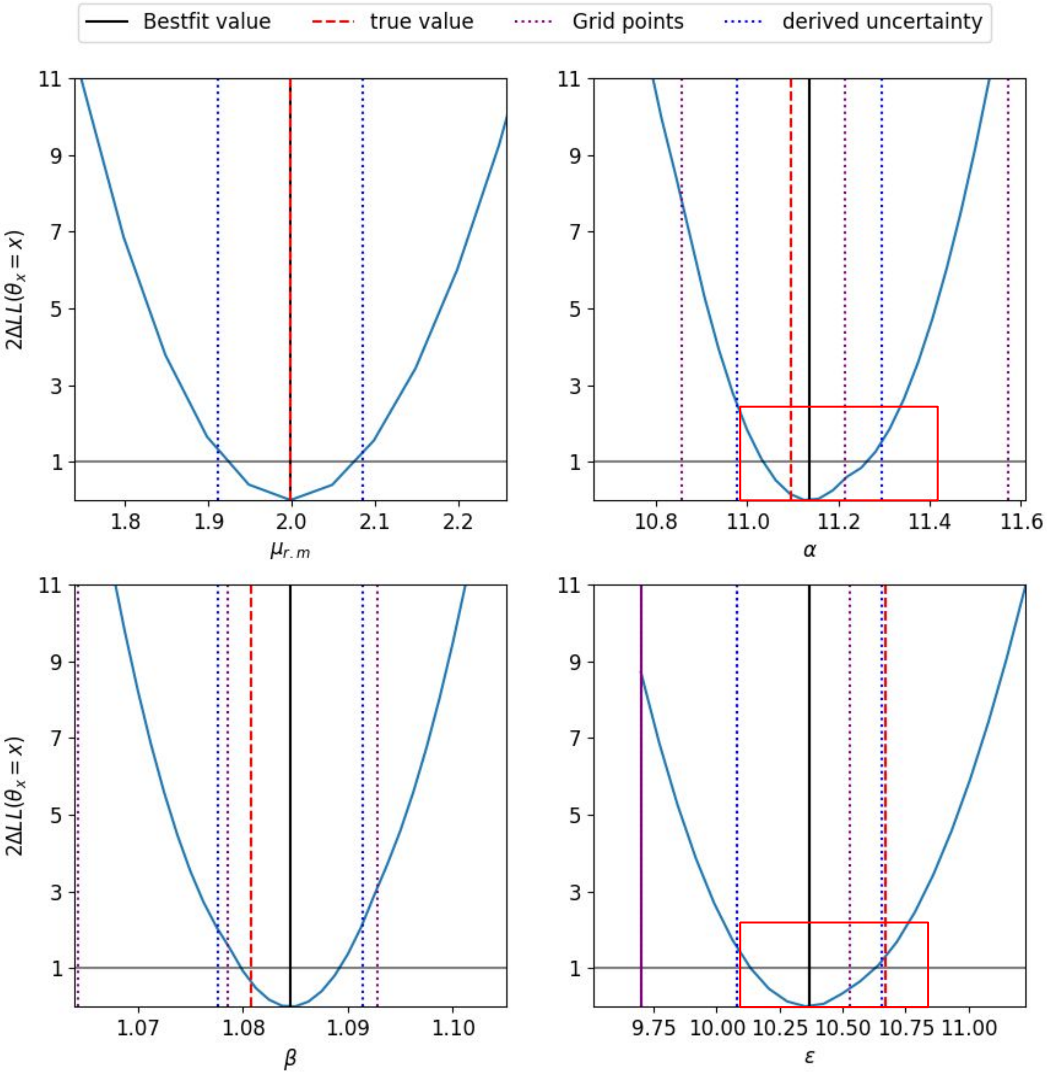


Parameters I'm talking about

Parameter	Desciption	Trace time
α	Linearly scale mean n^{q}_{prod} with energy	$11^{+2.0}_{-0.5}\mathrm{keV}^{-\beta}$
β	Power law of mean n^{q}_{prod} with energy	1.1 ± 0.5
γ	Linear dependence of mean n^{el}_{prod} with density and electric field	$(4.8 \pm 0.2) \times 10^{-2}$
δ	power law of mean n^{el}_{prod} with electric field	$(4.8 \pm 0.2) \times 10^{-2}$
ϵ	Changes energy scale of mean n^{el}_{prod} energy dependence changes	$12.6^{+3.4}_{-2.9}$
ζ	Translates sigmoid of mean n^{el}_{prod} in energy	0.3 ± 0.1
η	Changes sigmoid shape of mean n^{el}_{prod} in energy	2 ± 1
θ	Translates sigmoid of mean n^{ph}_{prod} in energy	0.30 ± 0.05
l	Changes sigmoid shape of mean n^{ph}_{prod} in energy	2.0 ± 0.5

Some issues

Kinks in the likelihood between anchor points indicate that the the differential rate term is showing correlation between parameters not captured in rate estimator.

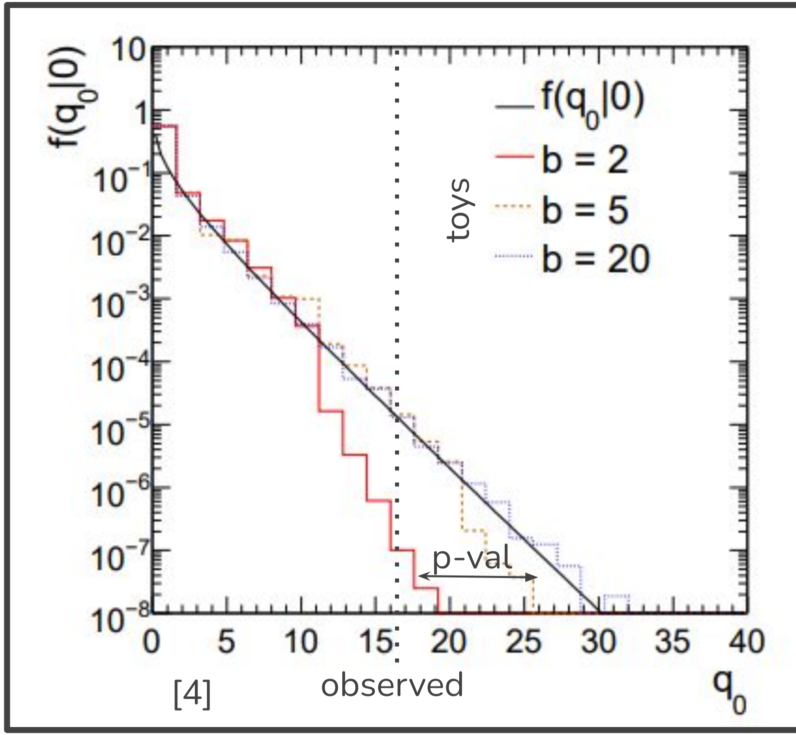


Why use *interpolated* rate estimators?

Markov Chain Monte Carlo could efficiently find the best fit with many parameters - simulate rate at every step.

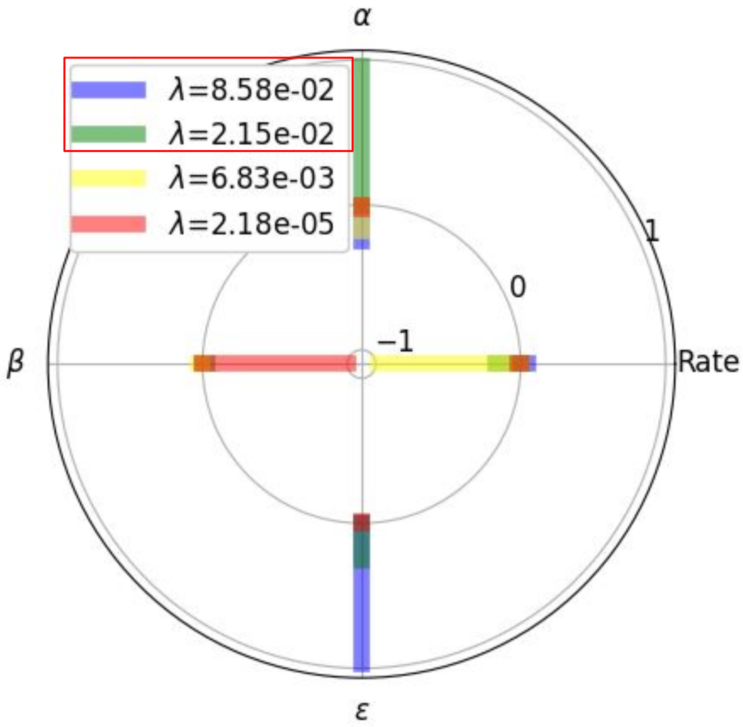
Issues:

a) Throw out all our diff programming benefits- too slow to evaluate rate estimator gradients+hessians
b) Non-asymptotic inference requires many many best fits O(1000)
c) Asymptotic limit setting still requires O(40).



Fun Possible solution

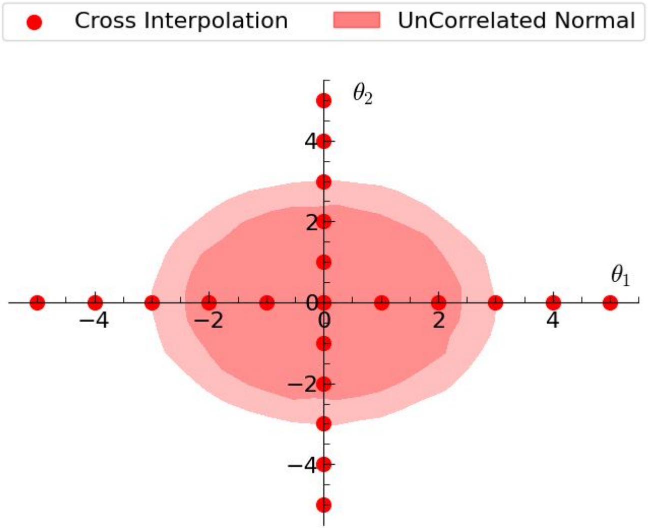
1. Use MCMC or some other approximation to find the best fit to calibration data
2. Perform a **principal component analysis**
   a. Largest eigenvalue eigenvectors of covariance matrix **“most information”**
   b. Covariance ~inverse hessian of likelihood
   c. Tells us “**in which direction the likelihood/constraint is most flat**”
3. Use this to inform a reduced dimensionality



Here we would use epsilon-alpha and alpha-rate-beta.

Why is it incomplete?

$$ln(L(\vec{\theta}, \{\vec{s_i}\})) = -\mu(\vec{\theta}) + \sum_i^{events} ln(\sum_j^{sources} R^j(\vec{\theta}, \vec{s_i}))$$



My work focused on the **differential rate** term

Have to evaluate the **total rate** using simulations of fixed points and interpolate

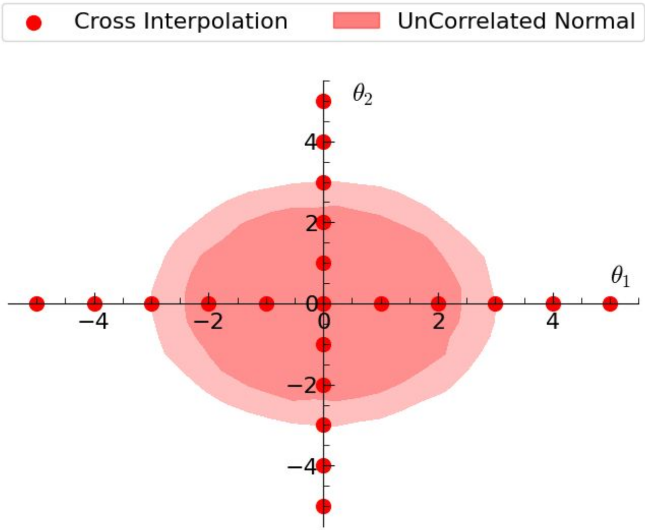
Still only need total counts so better than full templates

$$\mathcal{O}(n_{parameters} \times n_{anchors})$$

Rate estimator kerfuffle

Once you’re correlated you need a grid to capture correlations and it gets out of hand

$$\mathcal{O}(n_{parameters} \times n_{anchors})$$



$$\mathcal{O}({n_{anchors}}^{n_{parameters}})$$