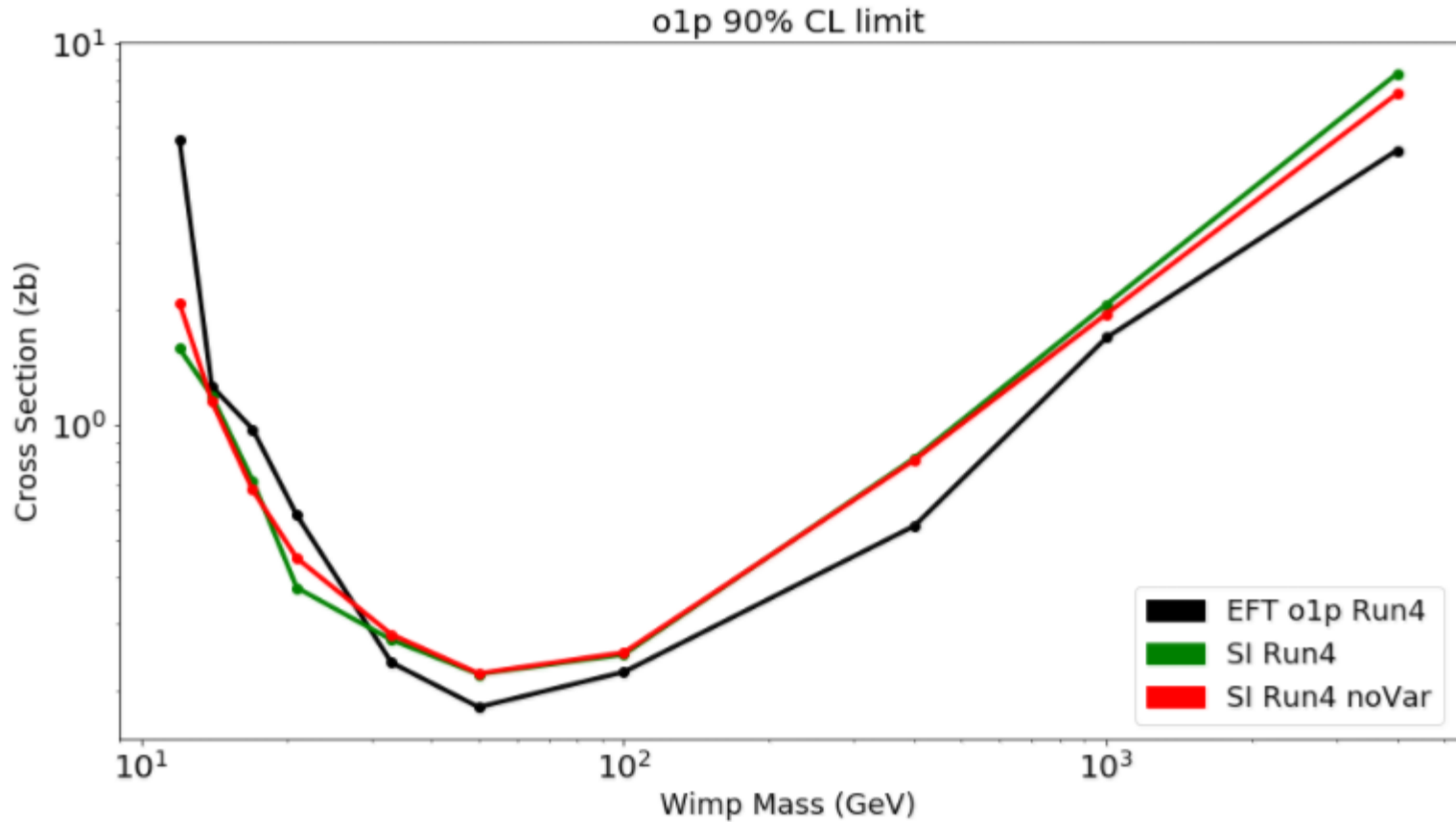


# Effective Field Theory in LUX Run4

Shaun Alsum

# Comparing limits



Running (I hope) proper SI comparison now.

Previously, in EFT...

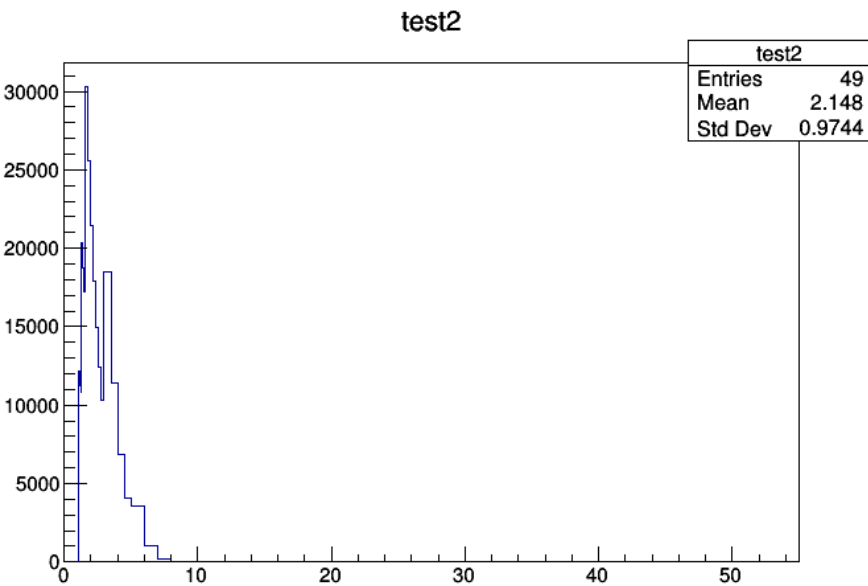
# Additional features!

- All Isotopes added
- all masses added
- method generalized to all operators

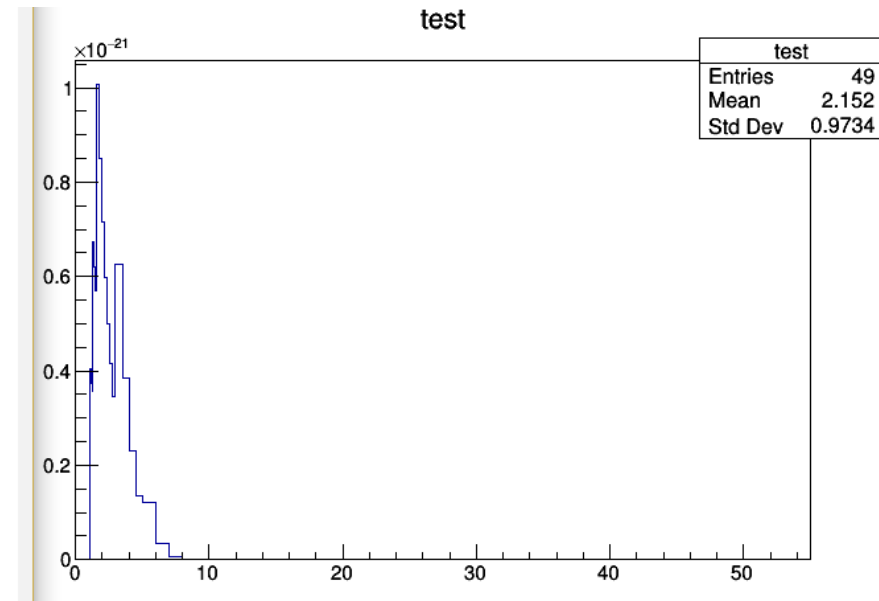
# Compare PLR and EFT Recoil Spectra

Note: bin sizes vary so shape is funky

From PLR code



From EFT Mathematica code  
with 1.1 keV cut



# Generate S1 vs $\log_{10}(S2)$ PDF

- Make generation file use my recoil spectrum builder (discussed previously) instead of PLR default

```
TString histname = TString::Format("WimpSpectrumHist_%d", timeBin);
TH1D *WimpSpectrumHist = new TH1D(histname.Data(), histname.Data(), NRECOILE, ws_xb
FillWimpSpectrumHist(WimpSpectrumHist, mWimp);
histname = TString::Format("WimpSpectrumHistCheck_%d", timeBin);
TH1D *WimpSpectrumHistCheck = new TH1D(histname.Data(), histname.Data(), NRECOILE,
// Fill S1 vs. log10S2 hisograms (conditional on fields)
```

```
- TString histname = TString::Format("WimpSpectrumHist_%d", timeBin);
TH1D *WimpSpectrumHist = new TH1D(histname.Data(), histname.Data(), NRECOILE, ws_xbins);
// FillWimpSpectrumHist(WimpSpectrumHist, mWimp); changed for EFT
FillWimpSpectrumEFT(WimpSpectrumHist, mWimp, 1, 'p'); //op1, proton for now
histname = TString::Format("WimpSpectrumHistCheck_%d", timeBin);
TH1D *WimpSpectrumHistCheck = new TH1D(histname.Data(), histname.Data(), NRECOILE, ws_xbins);
// Fill S1 vs. log10S2 hisograms (conditional on fields)
```

# The Idea

- An interaction can have the arbitrary form:

$$\mathcal{L}_{\text{int}} = \chi \mathcal{O}_x \chi^\dagger \mathcal{O}_N N.$$

- We generally consider the case where  $\mathcal{O}_x = \mathcal{O}_N = [\text{identity}]$  (spin independent) because this naively should be dominant for low momentum-transfer interactions
- In general, however, we can consider any operator that satisfies all known symmetries.

# The Idea continued

- There are 5 quantities that are Galilean invariant.

$$- \mathbf{I}, i\mathbf{q}, \mathbf{v}^\perp, \mathbf{S}_X, \mathbf{S}_N$$

- These can be combined into 16 operators

$$\mathcal{O}_1 = 1_X 1_N$$

$$\mathcal{O}_2 = (v^\perp)^2$$

$$\mathcal{O}_3 = i\vec{S}_N \cdot \left( \frac{\vec{q}}{m_N} \times \vec{v}^\perp \right)$$

$$\mathcal{O}_4 = \vec{S}_X \cdot \vec{S}_N$$

$$\mathcal{O}_5 = i\vec{S}_X \cdot \left( \frac{\vec{q}}{m_N} \times \vec{v}^\perp \right)$$

$$\mathcal{O}_6 = \left( \vec{S}_X \cdot \frac{\vec{q}}{m_N} \right) \left( \vec{S}_N \cdot \frac{\vec{q}}{m_N} \right)$$

$$\mathcal{O}_7 = \vec{S}_N \cdot \vec{v}^\perp$$

$$\mathcal{O}_8 = \vec{S}_X \cdot \vec{v}^\perp$$

$$\mathcal{O}_9 = i\vec{S}_X \cdot \left( \vec{S}_N \times \frac{\vec{q}}{m_N} \right)$$

$$\mathcal{O}_{10} = i\vec{S}_N \cdot \frac{\vec{q}}{m_N}$$

$$\mathcal{O}_{11} = i\vec{S}_X \cdot \frac{\vec{q}}{m_N}$$

$$\mathcal{O}_{12} = \vec{S}_X \cdot (\vec{S}_N \times \vec{v}^\perp)$$

$$\mathcal{O}_{13} = i(\vec{S}_X \cdot \vec{v}^\perp) \left( \vec{S}_N \cdot \frac{\vec{q}}{m_N} \right)$$

$$\mathcal{O}_{14} = i\left( \vec{S}_X \cdot \frac{\vec{q}}{m_N} \right) (\vec{S}_N \cdot \vec{v}^\perp)$$

$$\mathcal{O}_{15} = -\left( \vec{S}_X \cdot \frac{\vec{q}}{m_N} \right) \left( (\vec{S}_N \times \vec{v}^\perp) \cdot \frac{\vec{q}}{m_N} \right)$$

$$\mathcal{O}_{16} = -\left( (\vec{S}_X \times \vec{v}^\perp) \cdot \frac{\vec{q}}{m_N} \right) \left( \vec{S}_N \cdot \frac{\vec{q}}{m_N} \right).$$



# The Idea continued more

- These different operators can give rise to different recoil energy spectra according to

$$\frac{dR}{dE_R} = \frac{\rho_0}{32\pi m_\chi^3 m_p^2} \int_{v > v_{min}} \frac{f(\vec{v})}{v} (c_i^{(N)})^2 \sum_{k=M, \Sigma'', \Sigma', \Delta, \Phi'', \tilde{\Phi}'} a_{iik} F_k^{(N,N)}$$

Where the quantity represented by the sum is  $O_i$  broken into calculable nuclear form factors.

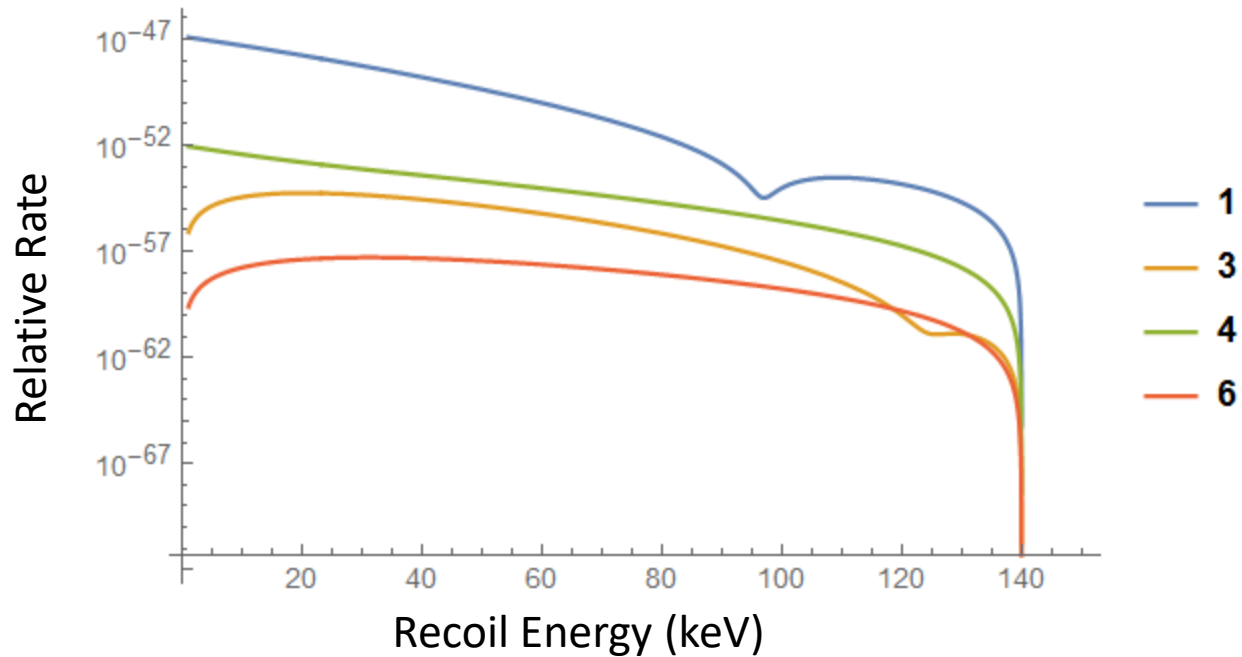
- Want to put limits on  $c_i^{(N)}$

# The process

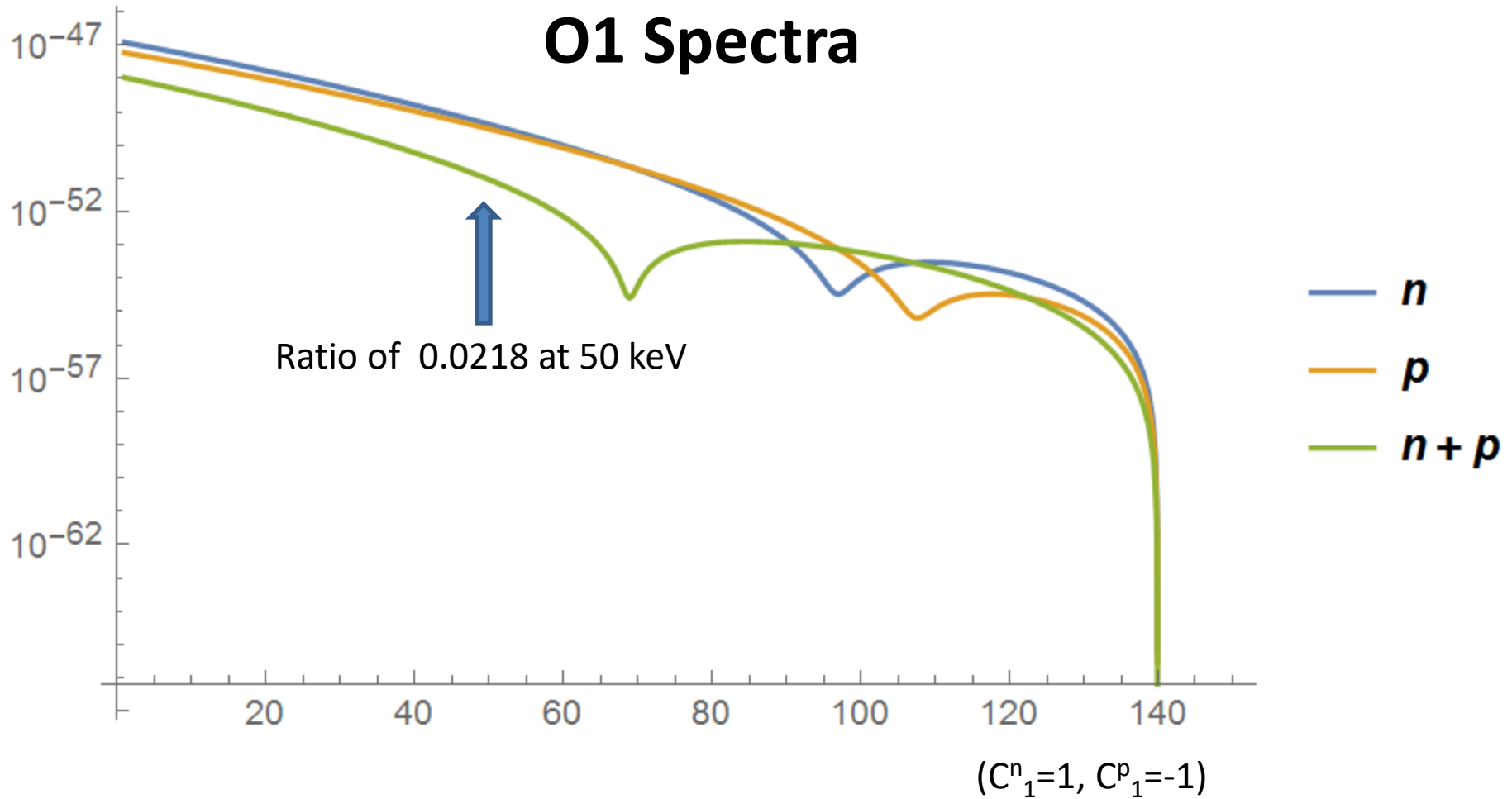
- Compare calculated spectra with observed.
  - Interference between operators is forbidden in most cases, so we can test 1 at a time.
  - Inteference between the proton and neutron operators is not, however, forbidden.  
Nevertheless, we still begin by doing these 1 at a time for simplicity.

# Some sample neutron spectra

Generated using a mathematica package created for this purpose.



# n-p Inteference Demonstration



# Integration into the PLR limit code

- The limits code main file is “SIRun4.cxx”
  - This calls “ImportSignalModel\_5D”
- ImportSignalModel\_5D is in “ImportSignalModel.h” and creates a “RooSignalPDF” object
  - ws-  
>factory(TString::Format("SignalPDF::nrPop%d(mWimp,S1,log10S2,r,phi,drift,G2Var,NoNuisParam,%d,%d)",tt,tt,(int)useAnalyticIntegration));
  - Confusing, right?
- RooSignalPDF creates a 1D histogram and fills it using a function called “FillWimpHist”
- I have replaced FillWimpHist with “FillWimpHistEFT”

# FillWimpHistEFT

- Opens a file called o#c.dat, where # is the operator number, and c is either 'p' for proton, or 'n' for neutron.
- These files are tables of integrated rates in predefined bins and WIMP masses generated using mathematica.
- Parses the tables using the input WIMP mass, operator number, etc. to grab the correct data
- Fills the histogram with the appropriate values

# Rate tables.

- Rates are calculated using a variety of WIMP masses, for only spin  $\frac{1}{2}$  WIMPS.
- Each operator (28 of them, 14x2) is its own table.

# Full Plot!

o1p 90% CL limit

