VBS WH Analysis

Towards a 2D κ_W, κ_Z exclusion

January 25th, 2023

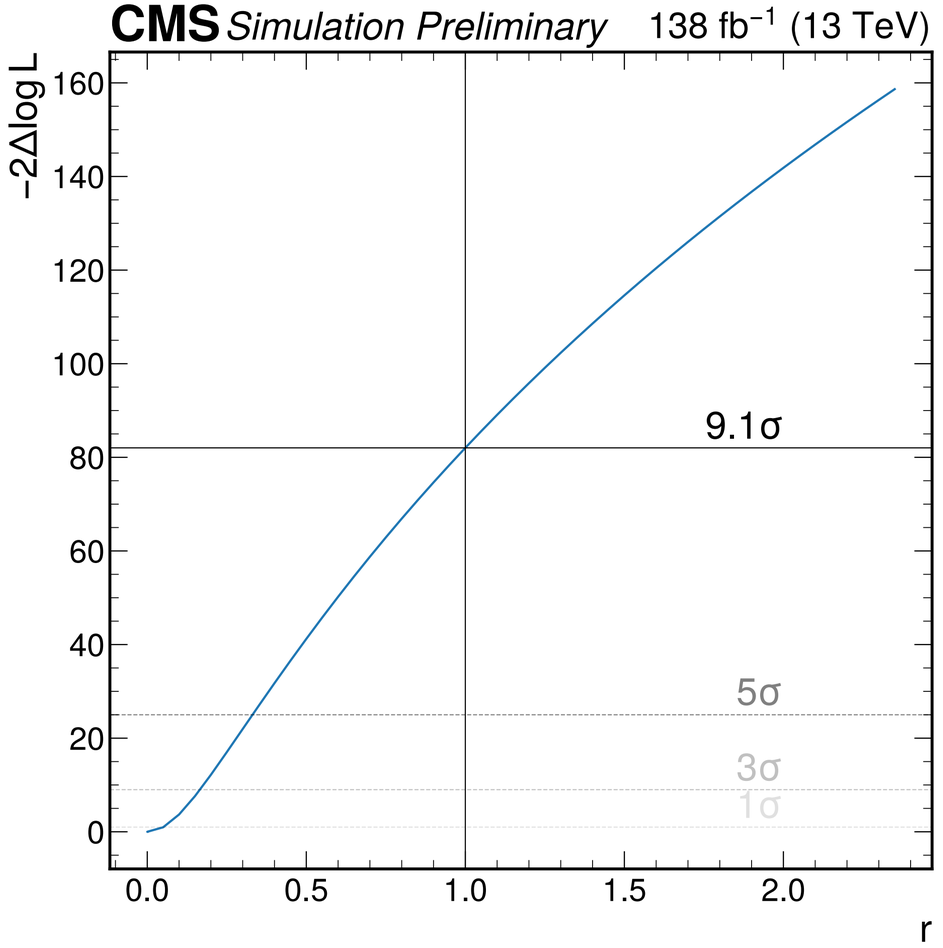
*P. Chang, L. Giannini, **J. Guiang**, F. Würthwein, Y. Xiang*

CMS

UC San Diego

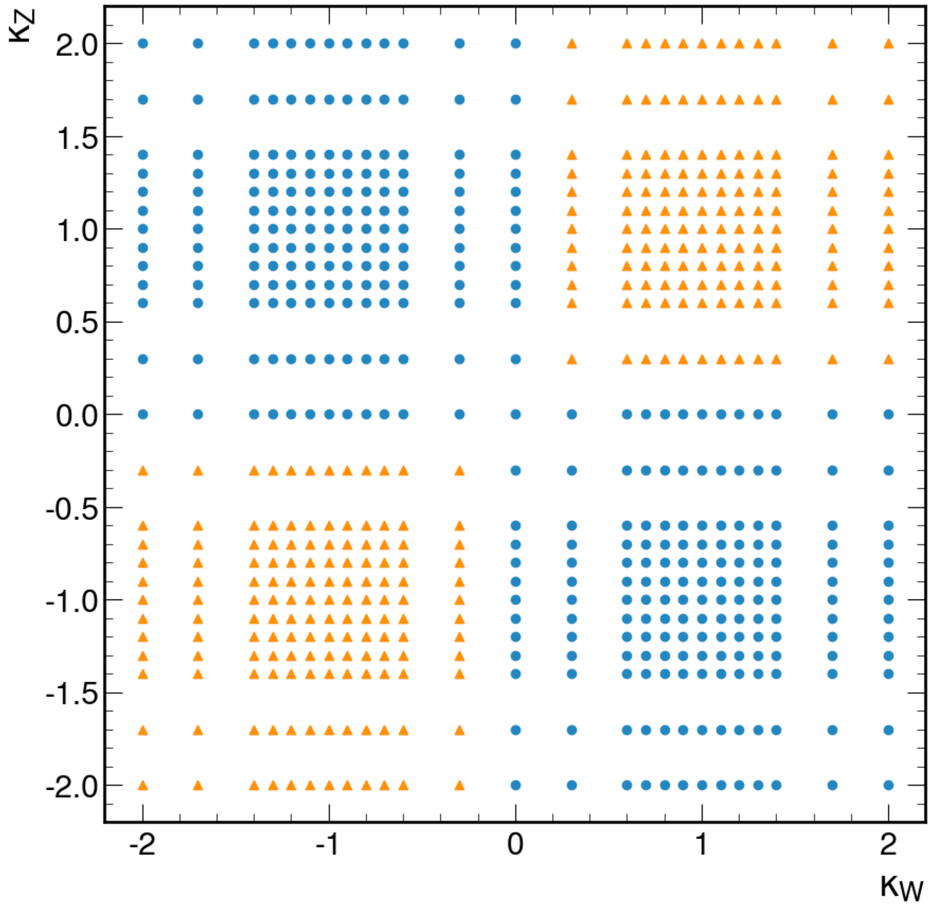
Problem Statement

- Comfortably exclude $\kappa_W = -1$, $\kappa_Z = +1$
- A “prettier” result would exclude some range of κ_W, κ_Z values
  - i.e. combined with current limits, would be able to *definitively* say “we exclude $\lambda_{WZ} = -1$”
- Goal:
  - Generate a signal sample with a κ_W and κ_Z (reweighted) scan
  - Produce a 2D (κ_W, κ_Z) exclusion

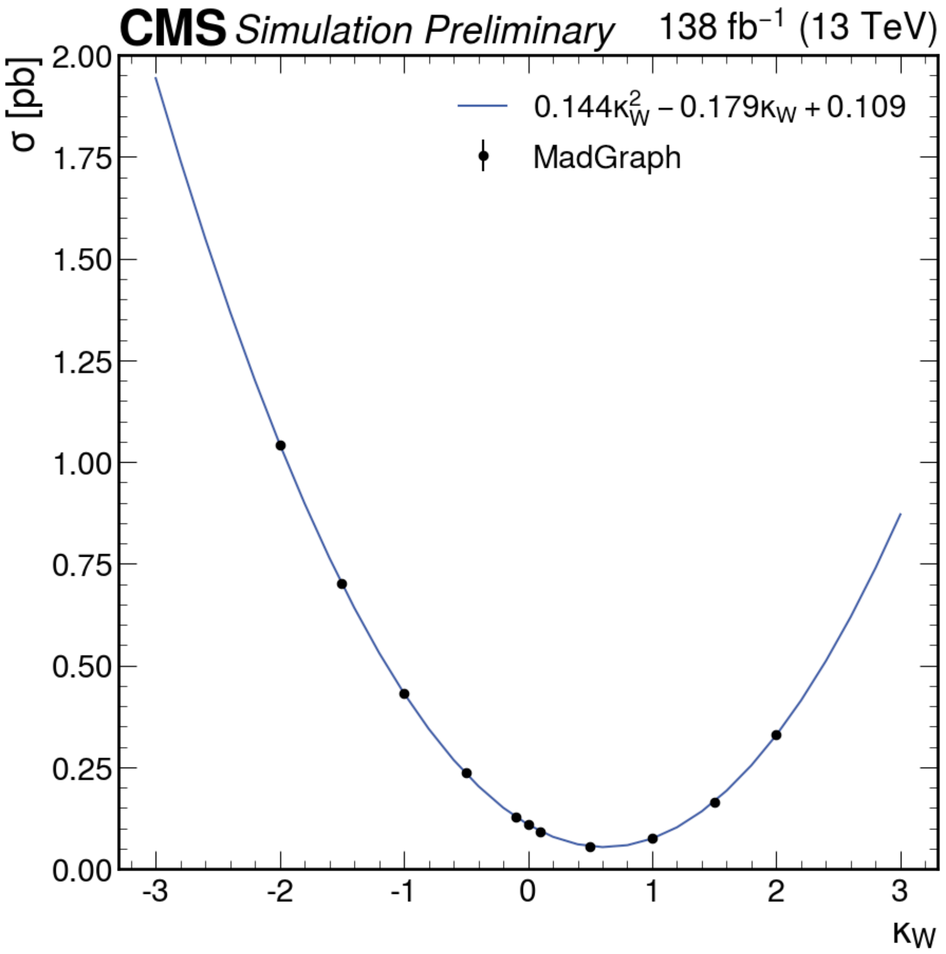


New Signal Samples

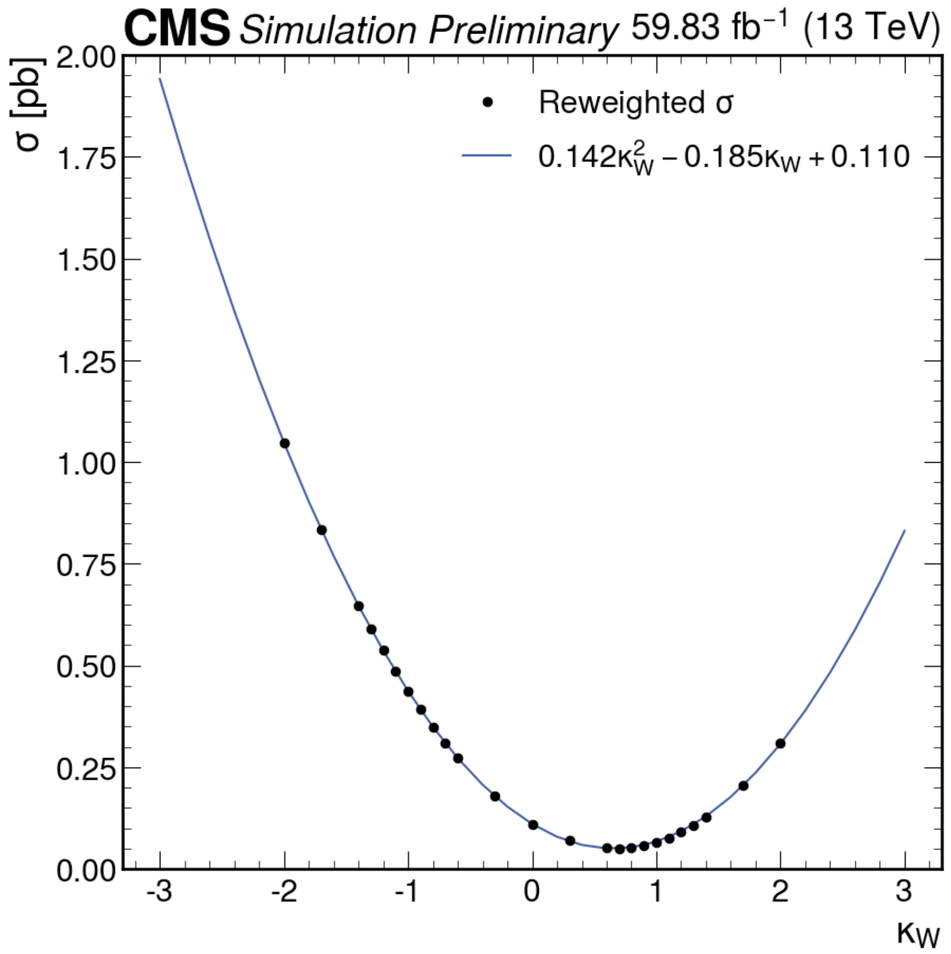
- Generated two signal samples:
  - $\lambda_{WZ} \leq 0$ sample
    - Reweighted around ($\kappa_W = -1$, $\kappa_Z = +1$)
  - $\lambda_{WZ} > 0$ sample
    - Reweighted around ($\kappa_W = +1$, $\kappa_Z = +1$)
  - Used PKU reweighting model
  - Full Run 2 samples
  - 100k events per NanoAODv9 “year”



New Signal Samples: Validation

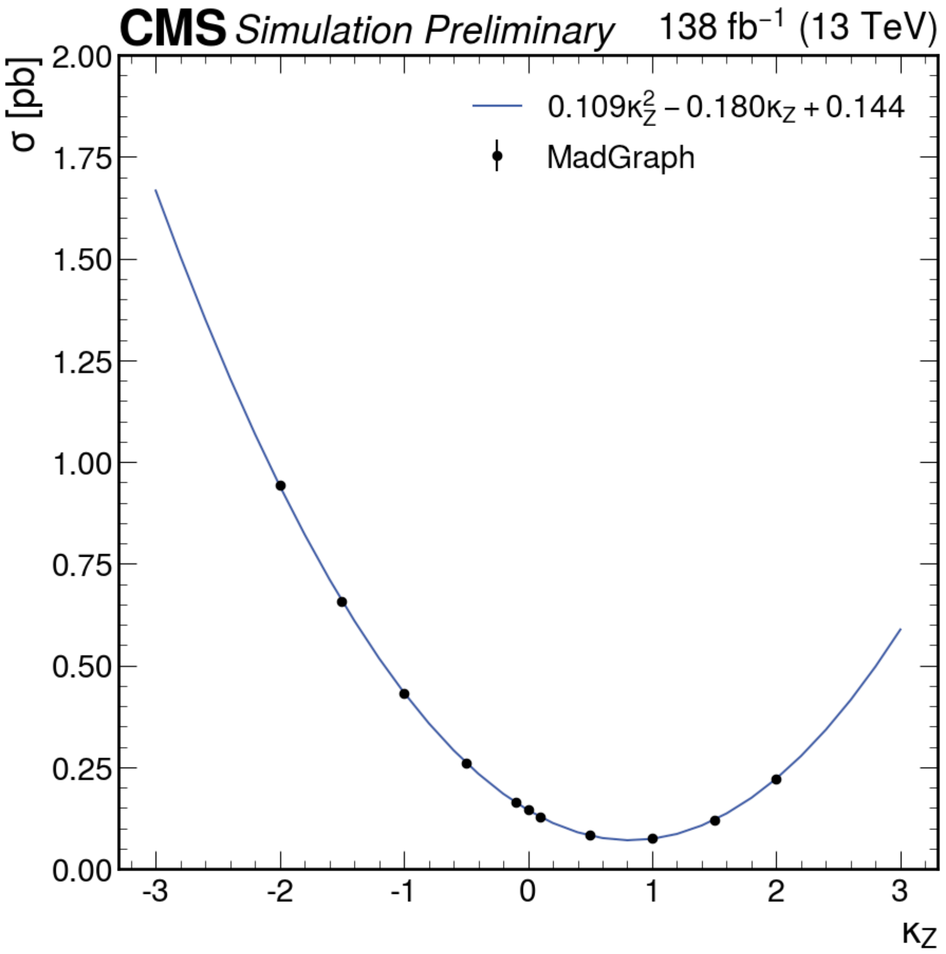


Single-points

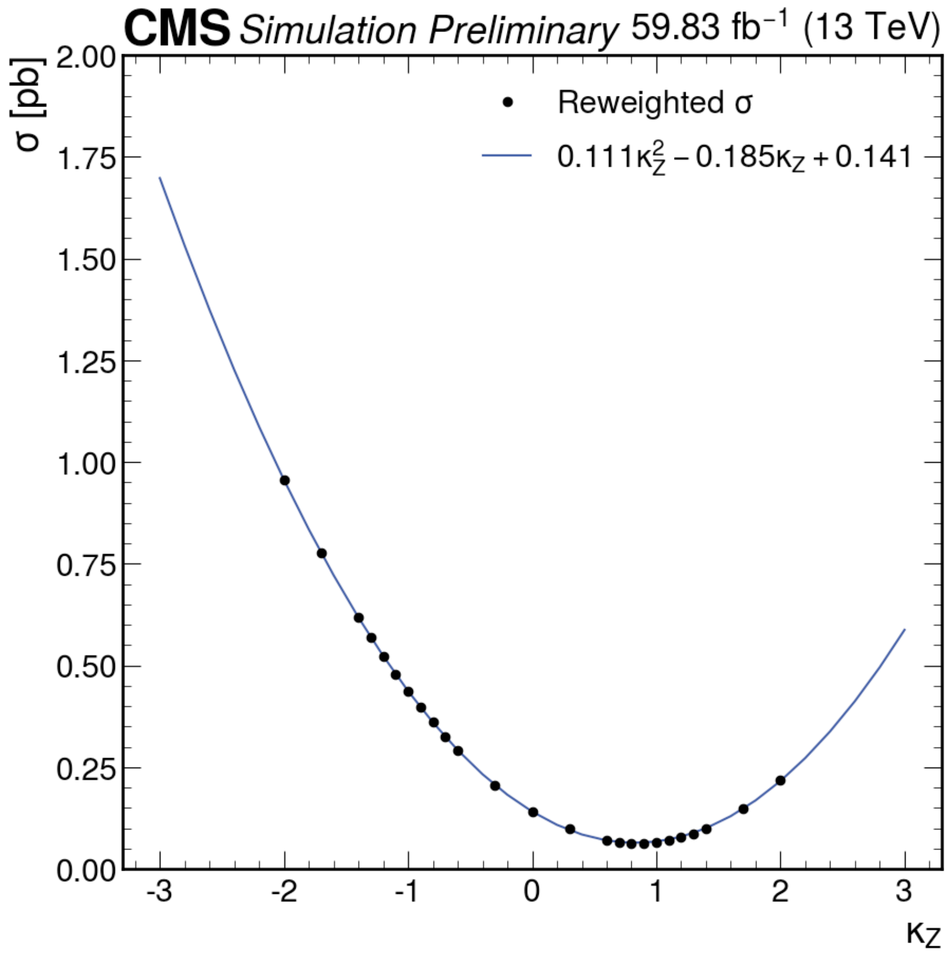


Reweighted points

New Signal Samples: Validation



Single-points

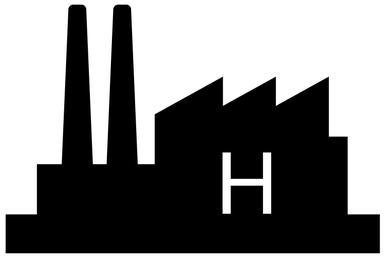


Reweighted points

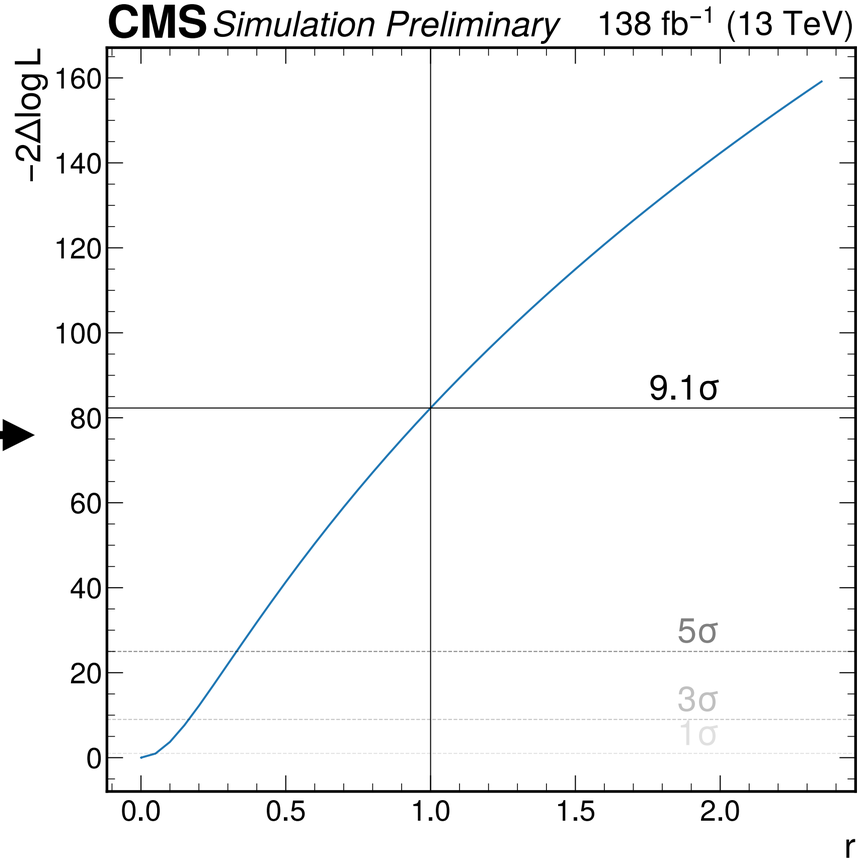
HiggsCombine

```
imax 1 number of channels
jmax 1 number of backgrounds
kmax 19 number of nuisance parameters
----------------------------------------
bin                         bin1
observation                  128
----------------------------------------
bin                         bin1        bin1
process                VBSWH_mkW     TotalBkg
process                        0            1
rate                      413.34       127.92
----------------------------------------
abcd_syst      lnN             -        1.084
abcd_stat      lnN             -        1.133
pdf_vars       lnN         1.022            -
muF_scale      lnN         1.178            -
isr_weights    lnN         1.001            -
fsr_weights    lnN         1.015            -
pu_rwgt        lnN         1.002            -
L1_prefire     lnN         1.010            -
hlt_sfs        lnN         1.008            -
mc_stat        lnN         1.022            -
lep_id         lnN         1.015            -
elec_reco      lnN         1.003            -
muon_iso       lnN         1.000            -
xbb_sfs        lnN         1.057            -
btag_sfs       lnN         1.003            -
met_unc        lnN         1.003            -
jes            lnN         1.066            -
jer            lnN         1.008            -
lumi           lnN         1.025            -
```

scan_kW_X_kZ_Y.dat

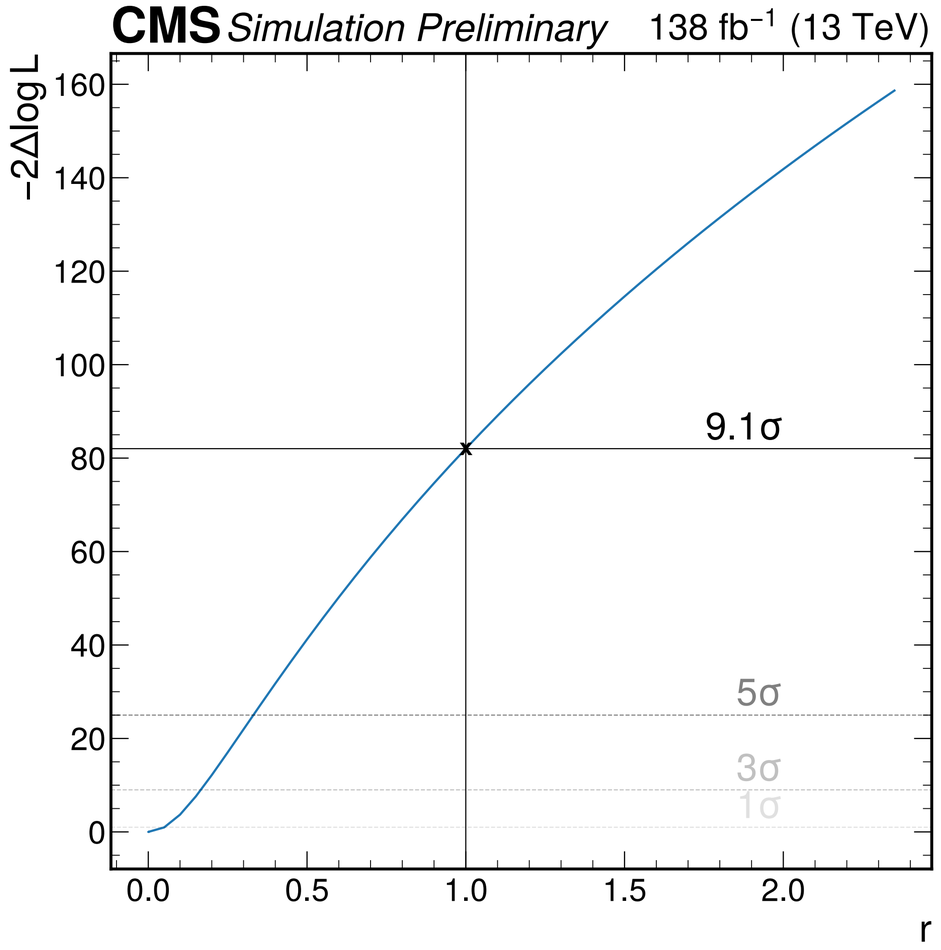



```
combine -M MultiDimFit -d scan_kW_X_kZ_Y.root
    -m 125 -t -1
    --expectSignal=0
    --setParameters r_VBSWH_mKW=0
    --setParameterRanges r_VBSWH_mKW=0.0,2.0
    --saveNLL
    --algo grid
    --points 101
    --rMin 0 --rMax 5
    --alignEdges 1
```

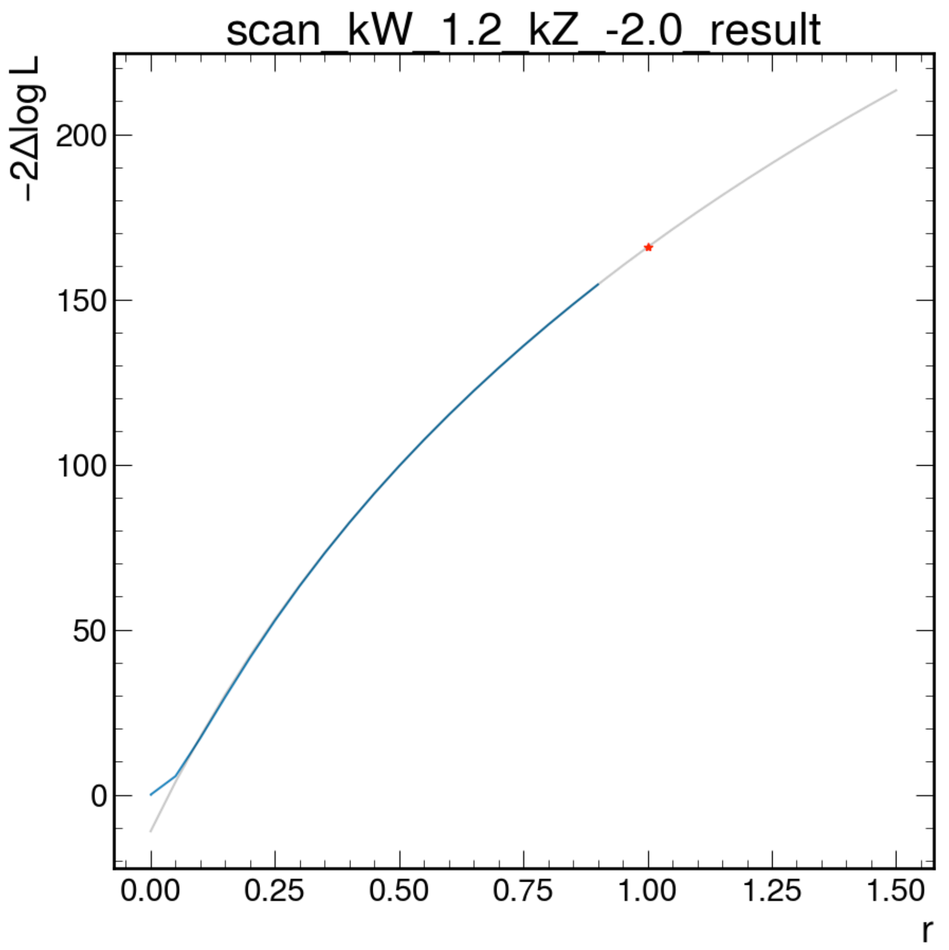


Repeat for each point κ_W = X, κ_Z = Y

HiggsCombine



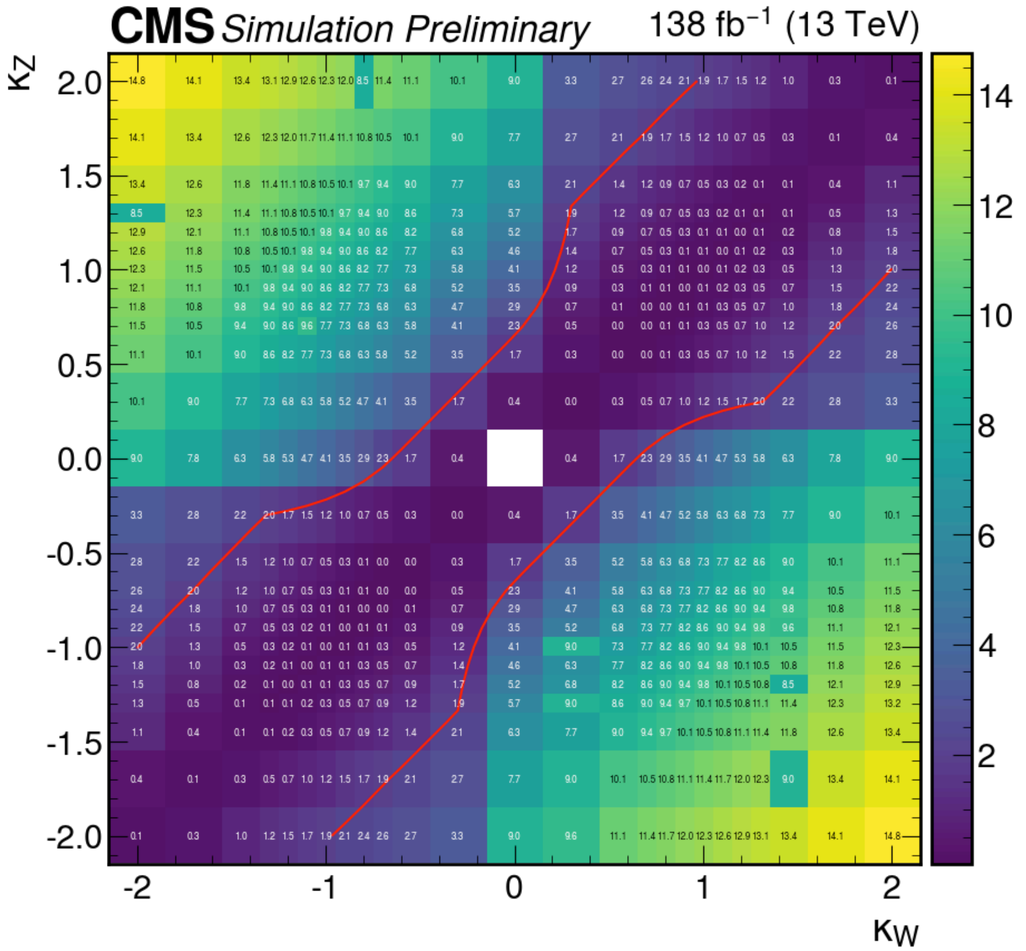
Take σ exclusion of r = 1



Infer σ exclusion of r = 1

Collected Results

- Bins centered on scanned κ_W, κ_Z points
- Exclusion limit plotted on z-axis
- Red contour roughly shows σ = 2 boundary
  - Simplistically derived by Matplotlib
- Discontinuities not large enough to be a concern
  - More details in backup
  - Maybe we can smooth it out?

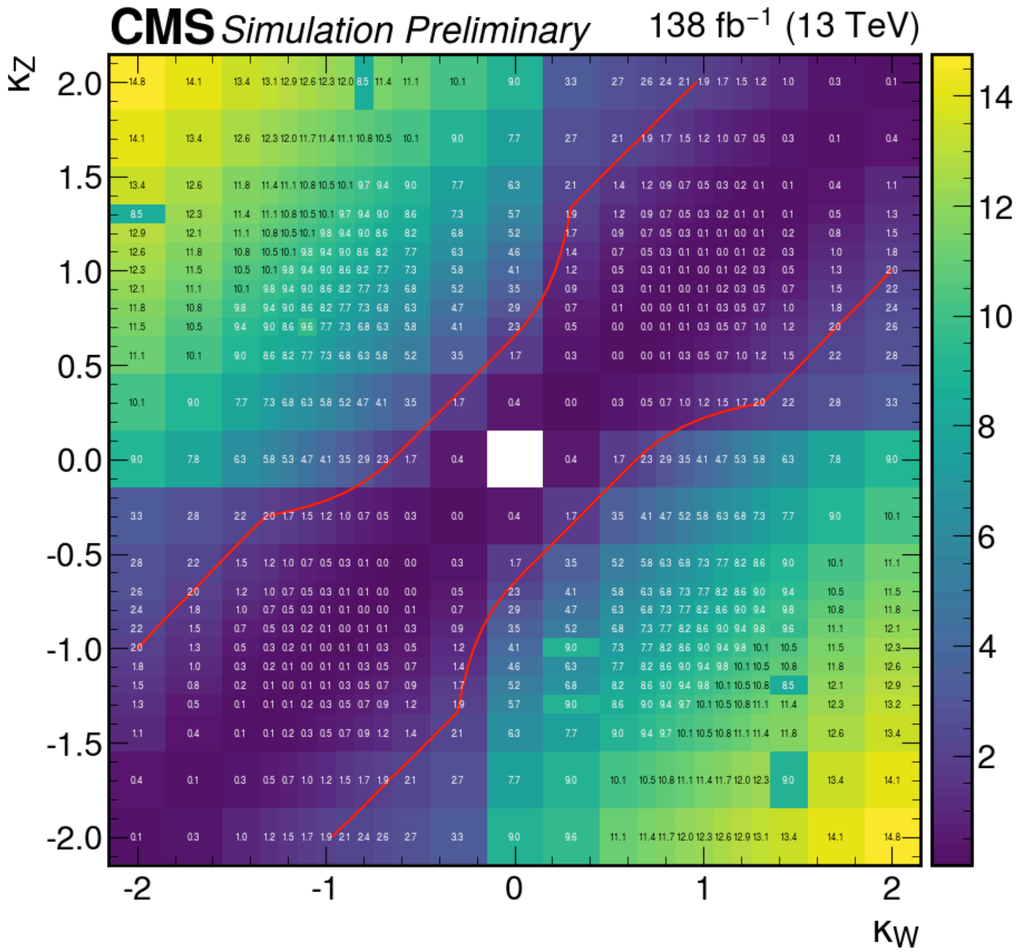


Collected Results: Fit

- Try fitting $\lambda_{WZ} \leq 0$ and $\lambda_{WZ} > 0$ quadrants separately
  - Fitting them together yields an inaccurate result
- Settled on the following functions to fit:

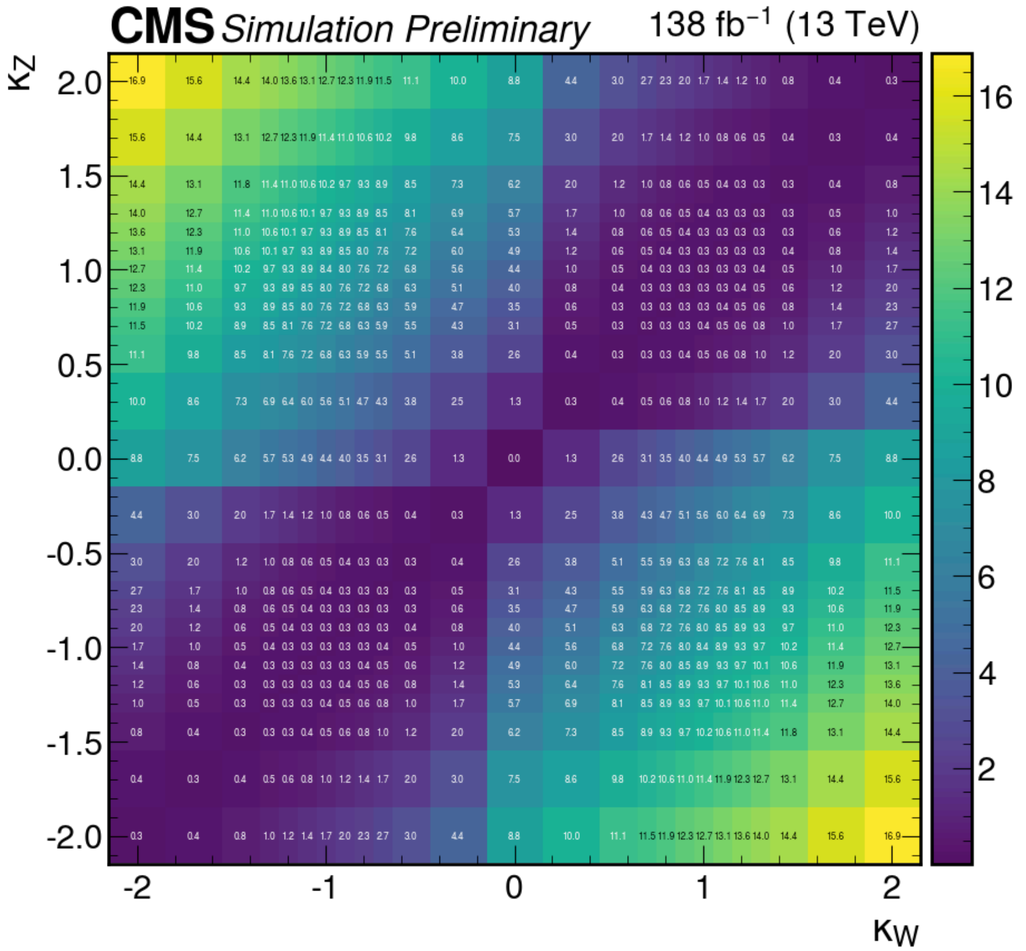
$$f_{+}(x,y) = \sqrt{|\,a(x-y)^2 + b(x+y)^2\,|} + c$$

$$f_{-}(x,y) = a(x-y)^2 + b$$



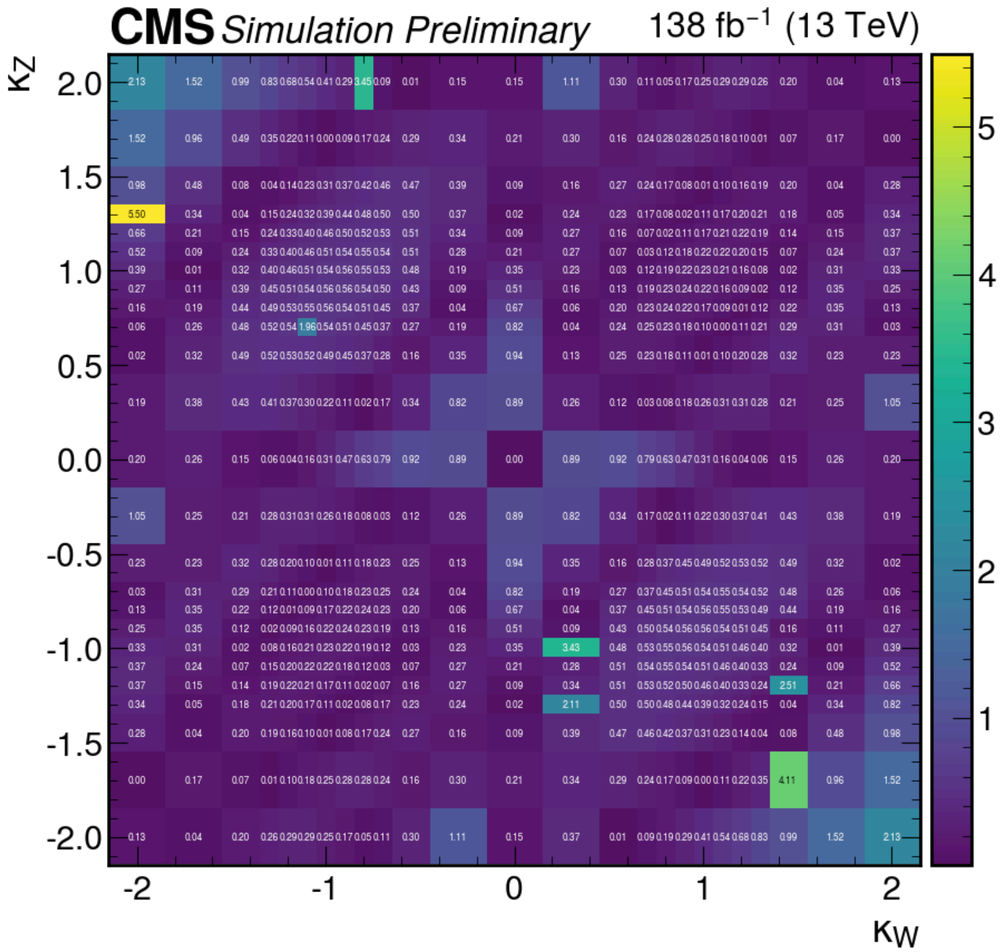
Collected Results: Fit

- Fit seems fairly accurate
- Tends to over-predict exclusion around $\kappa_W = \kappa_Z$
- Also over-predicts exclusion around large κ_W, κ_Z for $\lambda_{WZ} \leq 0$
- Could be used to smooth out the discontinuities in the original plot
  - Mostly superficial, as they don't affect exclusion boundary



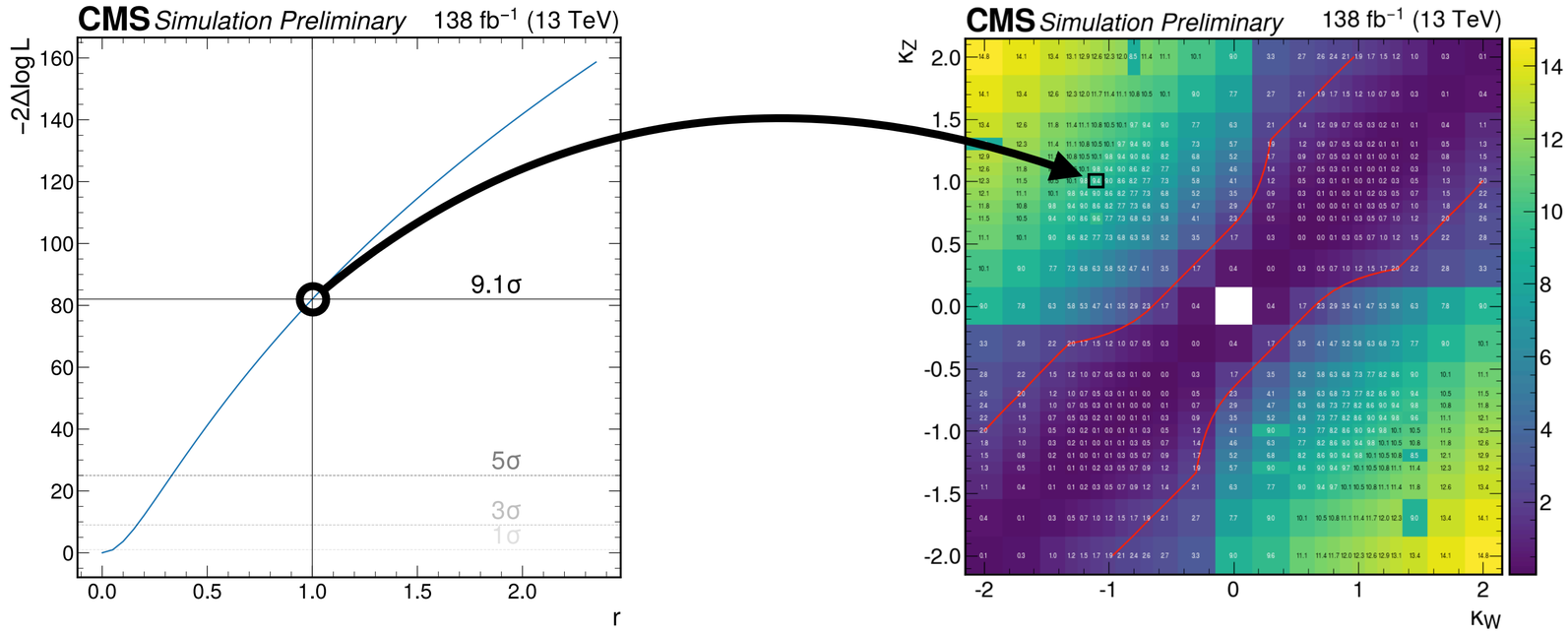
Collected Results: Fit

- Fit seems fairly accurate
- Tends to over-predict exclusion around $\kappa_W = \kappa_Z$
- Also over-predicts exclusion around large κ_W, κ_Z for $\lambda_{WZ} \leq 0$
- Plotted absolute difference between original and post-fit plot
  - i.e. plot reads as “the fit is off by Xσ for a given value of κ_W, κ_Z”



Summary

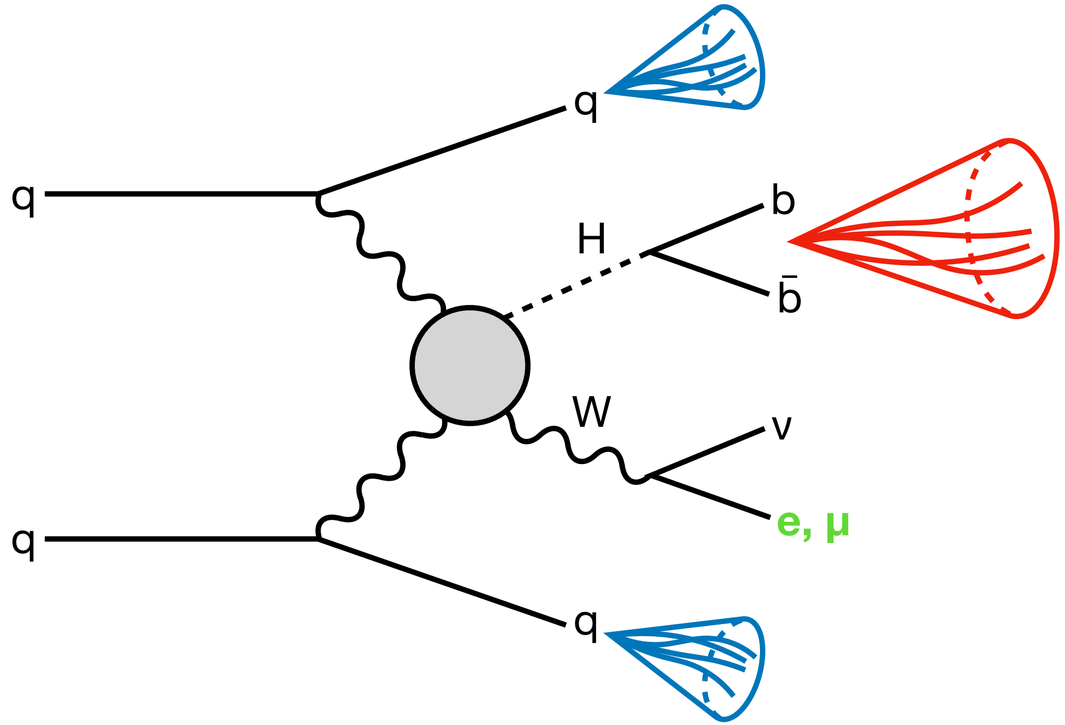
- Produced a 2D exclusion limit to strengthen final result
  - Some discontinuities, but not very important (could be smoothed out)
- Should we use this?

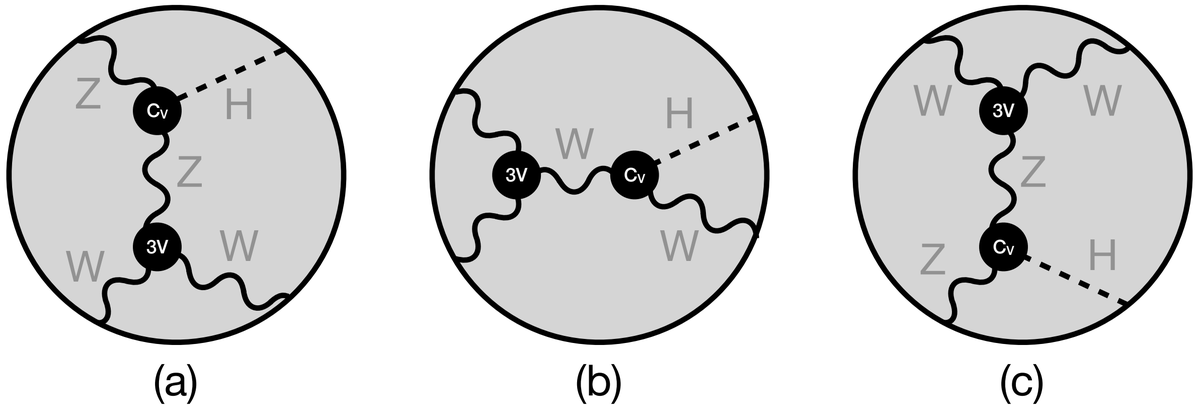


Backup

Target Final State

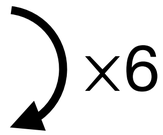
- Targeting **VBS** WH→**ℓνbb̄**
- Sensitive to $\kappa_V \rightarrow \lambda_{WZ} = \kappa_Z/\kappa_W$
  - Handle for ruling out $\lambda_{WZ} = -1$ (BSM)
- VBS WH BSM kinematics:
  - High-p_T H and W (high S_T)
  - VBS jets with large $\Delta\eta_{jj}$, M_{jj}





VBS WH Cross Sections

Model	σ [pb]
$\kappa_W = \kappa_Z = +1$ (SM)	0.075
$\kappa_W = -1$, $\kappa_Z = +1$	0.433
$\kappa_W = +1$, $\kappa_Z = -1$	0.433



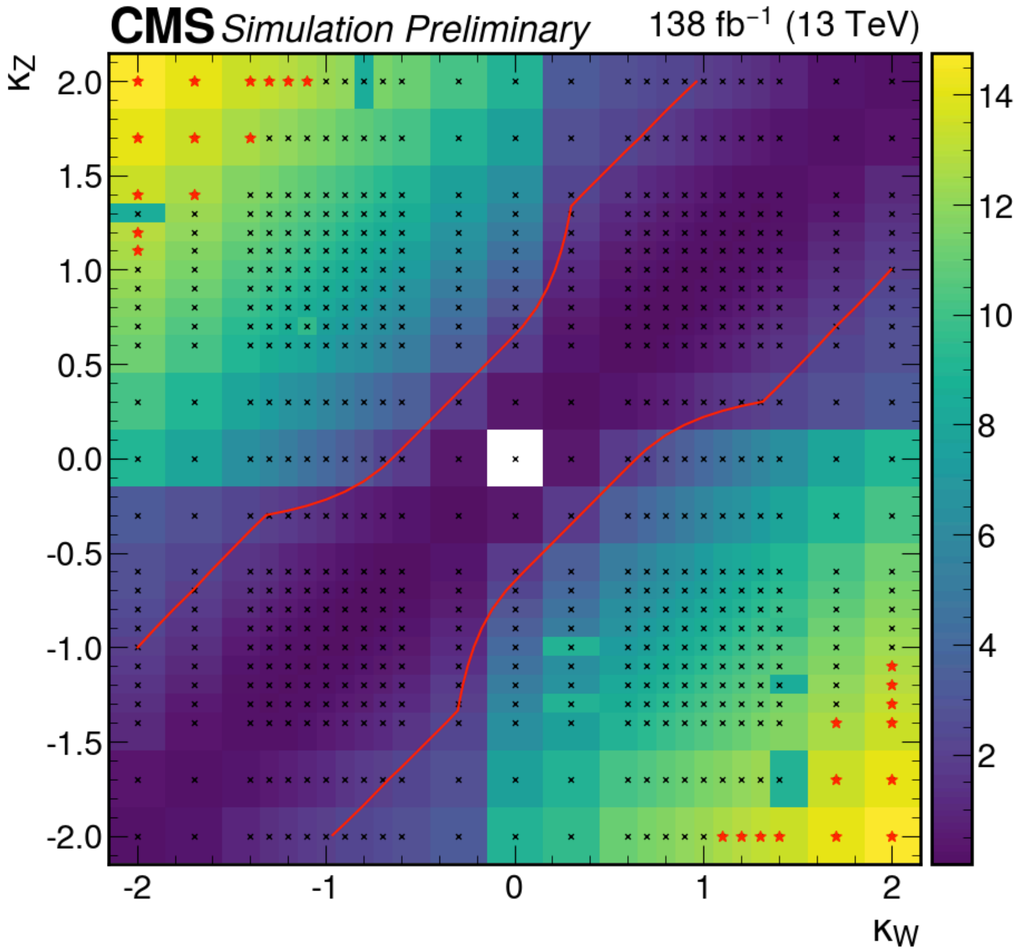
- Setting $\kappa_W = -1$ or $\kappa_Z = -1$ equivalently enhances cross section by a factor of 6
- These numbers are taken from MadGraph*: `generate p p > w h j j QCD=0`
  - Includes gen-level filters (e.g. jet $p_T > 10$ GeV)
  - Generated 10,000 events for each to obtain xsec value
- **Optimizing for $\kappa_W = -1$** (kinematics are equivalent to $\kappa_Z = -1$)
  - Generated 100k UL NanoAOD events for 2016 pre-VFP, 2016 post-VFP, 2017, and 2018

*More details in backup

UC San Diego

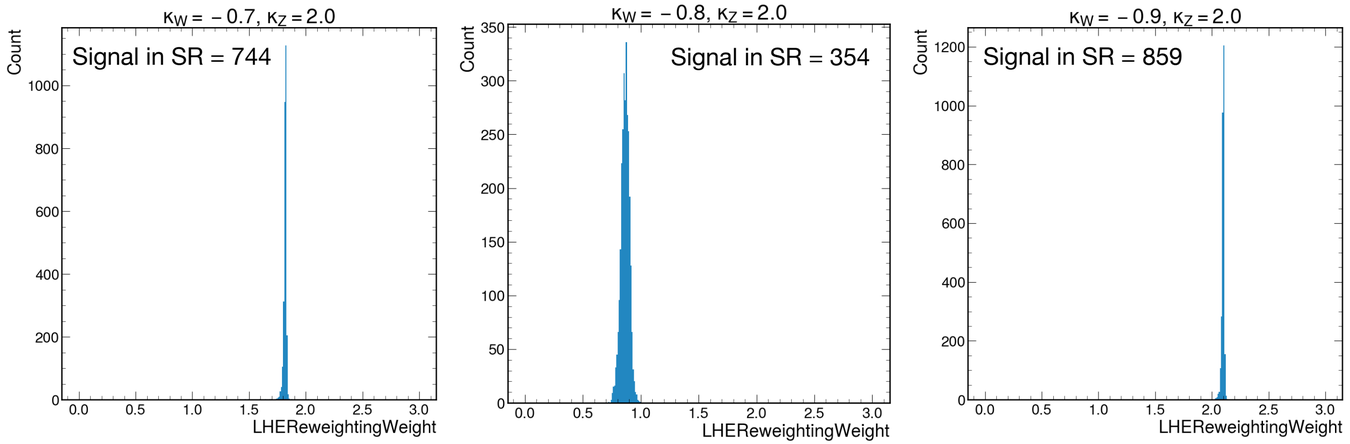
Collected Results

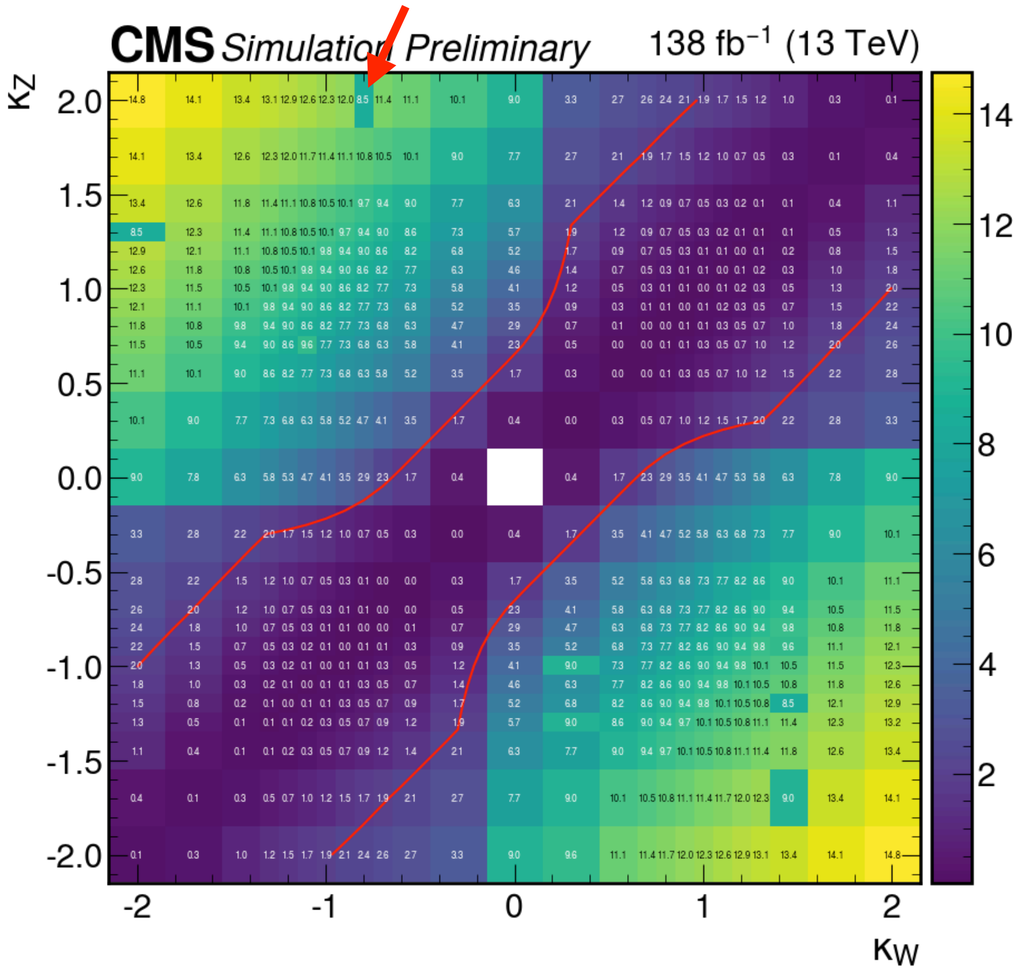
- Bins centered on scanned κ_W, κ_Z points
- Exclusion limit plotted on z-axis
- Red contour roughly shows σ = 2 boundary
  - Simplistically derived by Matplotlib
- Points taken directly from HiggsCombine plotted as a black **x**
- Points inferred from HiggsCombine result plotted as a **red** star



Collected Results

- Discontinuities come from places where signal yield does not smoothly vary
  - Based on xsec comparison, reweighting seems trustworthy
- Generating κ_W = -0.8, κ_Z = 2.0 sample to compare, so WIP…





Collected Results: Fit

- Functions used for the fit are a bit difficult to invert for plotting the exact contour
- Can instead make an arbitrarily finely binned plot and use Matplotlib contour function again
- Probably not useful…