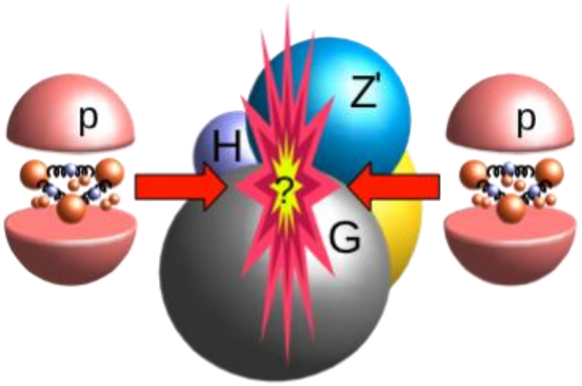
JHUGen Tutorial

Jeffrey Davis, and on behalf of all JHUGen authors

JOHNS HOPKINS UNIVERSITY

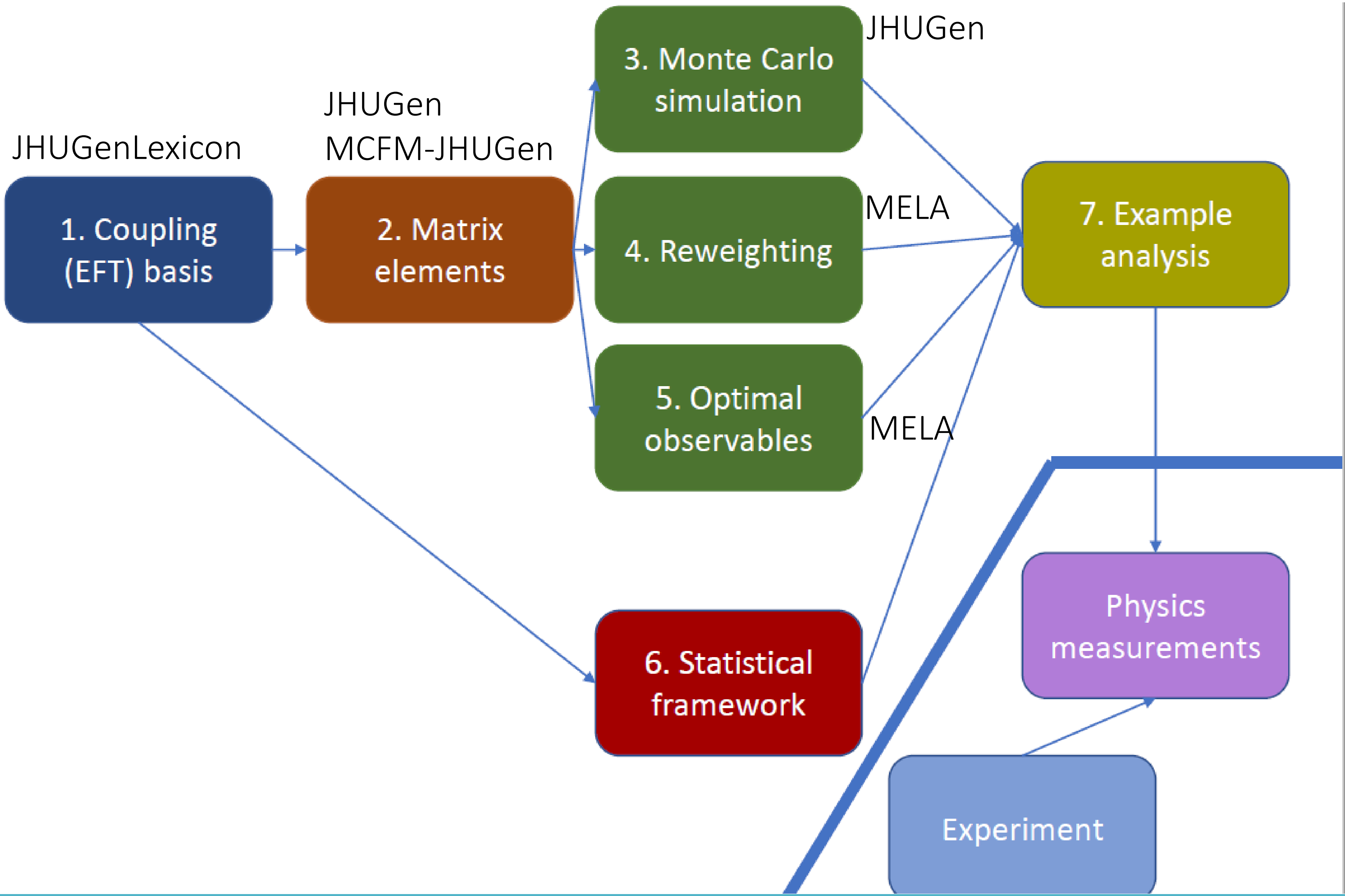
JHUGen Framework

- JHUGenerator + MCFM-JHUGen: https://spin.pha.jhu.edu/
- *Simulate wide range of processes involving spin 0,1,2 particles with a general coupling model*
- JHUGen MELA – Matrix Element Likelihood Approach
- *Calculate observables to optimally isolate processes or operators*
- *Reweight generated samples from one hypothesis to another*
- JHUGenLexicon
- *Tool for translation between different EFT bases and the JHUGen amplitude basis convention*

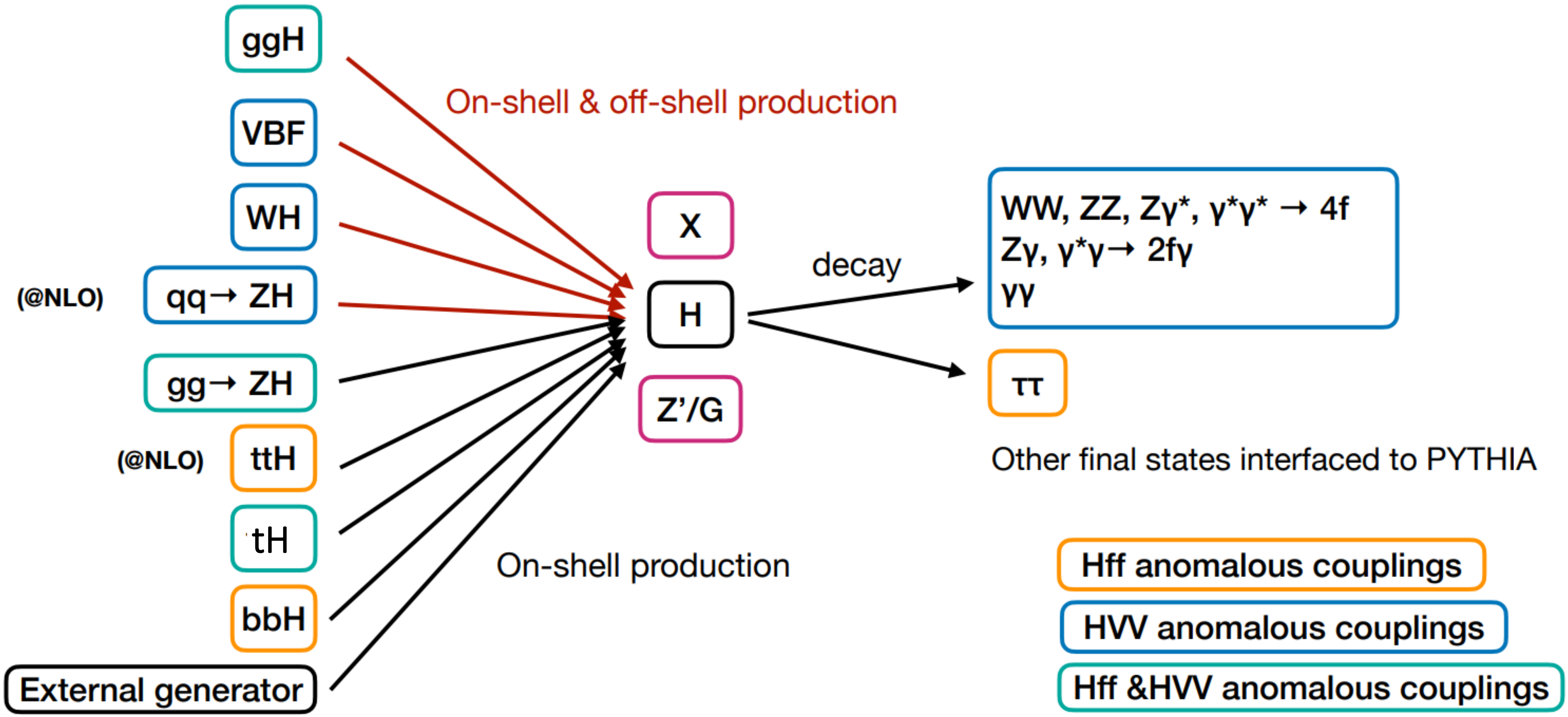



```
*********************************************************************************
*                              JHU Generator v7.5.5                             *
*********************************************************************************
*                                                                               *
*    Spin and parity determination of single-produced resonances at hadron colliders    *
*                                                                               *
*         I. Anderson, S. Bolognesi, F. Caola, J. Davis, Y. Gao, A. V. Gritsan,   *
*    L. S. Mandacaru Guerra, Z. Guo, L. Kang, S. Kyriacou, C. B. Martin, T. Martini,   *
*         K. Melnikov, R. Pan, M. Panagiotou, R. Rontsch, J. Roskes, U. Sarica,   *
*        M. Schulze, M. V. Srivastav, N. V. Tran, A. Whitbeck, M. Xiao, Y. Zhou   *
*              Phys.Rev. D81 (2010) 075022;  arXiv:1001.3396  [hep-ph],          *
*              Phys.Rev. D86 (2012) 095031;  arXiv:1208.4018  [hep-ph],          *
*              Phys.Rev. D89 (2014) 035007;  arXiv:1309.4819  [hep-ph],          *
*              Phys.Rev. D94 (2016) 055023;  arXiv:1606.03107 [hep-ph],          *
*              Phys.Rev. D102 (2020) 056022; arXiv:2002.09888 [hep-ph],          *
*              Phys.Rev. D102 (2021) 055045; arXiv:2104.04277 [hep-ph].          *
*              Phys.Rev. D105 (2022) 096027; arXiv:2109.13363 [hep-ph].          *
*                                                                               *
*********************************************************************************
```

JHUGen Framework



JHUGenerator

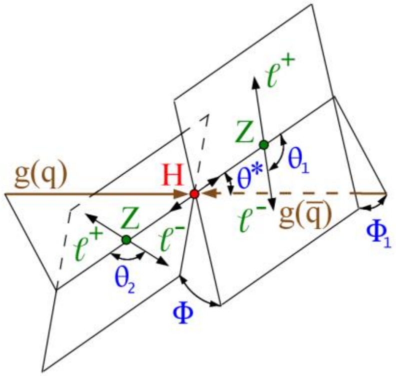


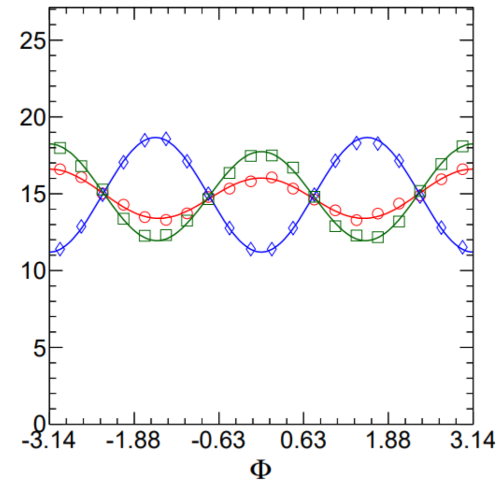
Anomalous Couplings and EFT

- HVV scattering amplitude parametrized in terms of 3 Lorentz tensor structures
- Couplings parametrized by g_i or a_i -> g_1(SM-Tree Level), g_2(CP-Even Dim-6) , g_4(CP-Odd Dim-6)

$$A(HV_1V_2) = \frac{1}{v}\left\{ M_{V_1}^2 \left(g_1^{VV} + \frac{\kappa_1^{VV} q_1^2 + \kappa_2^{VV} q_2^2}{\left(\Lambda_1^{VV}\right)^2} + \frac{\kappa_3^{VV} (q_1 + q_2)^2}{\left(\Lambda_Q^{VV}\right)^2} + \frac{2 q_1 \cdot q_2}{M_{V_1}^2} g_2^{VV} \right) (\varepsilon_1 \cdot \varepsilon_2) - 2 g_2^{VV} (\varepsilon_1 \cdot q_2)(\varepsilon_2 \cdot q_1) - 2 g_4^{VV} \varepsilon_{\varepsilon_1 \varepsilon_2 q_1 q_2} \right\},$$

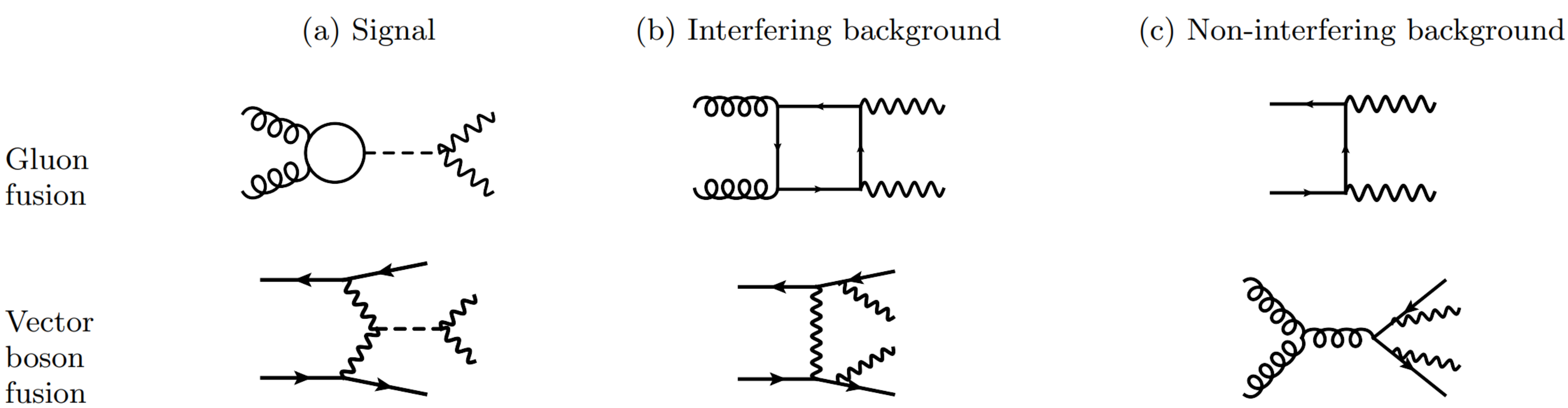
- Anomalous couplings at Higgs production or decay vertex, visible in kinematic observables
- Couplings can be mapped directly to couplings in the EFT Lagrangian





Off-Shell Simulation

- Off-shell processes can be simulated with the MCFM-JHUGen package or JHUGenerator (EW only)
- *Custom version of MCFM v.7*
- *Allows for anomalous couplings in off-shell production + decay*
- Supported Processes: Gluon Fusion Higgs, Electroweak(VBF+VH),ggZZ,qqZZ
- Supported Simulation: Signal, Background, Signal + Background (including interference)



MCFM Input Card for ggZZ

```
[Flags to specify the mode
-1        [nevtrequested]
.false.       [creatent]
.false.       [skipnt]
.false.       [dswhisto]
.false.       [creategrid]
```

nevtrequested: -1 for weighted distributions and cross-section, otherwise n = number of lhe events

creategrid: set to true to write grid for initial integration step (useful for making gridpacks)

```
81.ELMU       [nproc]
```

nproc: <ProcNumber>.<FinalState> (El,MU,TL)

- 128: gg -> H -> 4L
- 131: gg -> H -> 4L + gg -> ZZ -> 4L
- 132: gg -> ZZ -> 4L box diagrams

```
13000d0      [sqrts in GeV]
+1        [ih1 =1 for proton and -1 for antiproton]
+1        [ih2 =1 for proton and -1 for antiproton]
125.6d0      [hmass]
125.6d0      [scale:QCD scale choice]
125.6d0      [facscale:QCD fac_scale choice]
'm(3456)'        [dynamicscale]
```

sqrts: Collider energy

hmass: Higgs mass

scales: Various QCD scale parameters

Full MCFM manual can be found in MCFM-JHUGen/Doc/mcfm.pdf

MCFM Input Card for ggZZ

```
4         [itmx1, number of iterations for pre-conditioning]
100000        [ncall1]
10        [itmx2, number of iterations for final run]
100000        [ncall2]
1500          [ij]
```

itmx1,ncall1: initial integration step

itmx2,ncall2: final integration step

ij: random seed for the run

```
[Heavy quark masses]
173.2d0       [top mass]
4.75d0        [bottom mass]
1.275d0       [charm mass]
```

quark masses: pole mass for quarks, ggH loop is sensitive to these masses as well as kappa couplings

```
[Pdf selection]
'cteq6l1'    [pdlabel]
4       [NGROUP, see PDFLIB]
46      [NSET - see PDFLIB]
'NNPDF31_lo_as_0130'         [LHAPDF group]
0                [LHAPDF set]
```

LHAPDF group: name of pdf set in LHAPDF

LHAPDF set: set number from LHAPDF

MCFM Input Card for ggZZ

```
[Jet definition and event cuts]
2.5d0               [m34min]
13000d0             [m34max]
2.5d0               [m56min]
13000d0             [m56max]
70d0                [m3456min]
13000d0             [m3456max]
.true.              [inclusive]
'ankt'              [algorithm]
15d0                [ptjet_min]
0d0                 [|etajet|_min]
99d0                [|etajet|_max]
0.4d0               [Rcut_jet]
.true.              [makecuts]
3d0                 [ptlepton_min]
2.7d0               [|etalepton|_max]
0d0,0d0             [|etalepton|_veto]
0d0                 [ptmin_missing]
3d0                 [ptlepton(2nd+)_min]
2.7d0               [|etalepton(2nd+)|_max]
0d0,0d0             [|etalepton(2nd+)|_veto]
0d0                 [minimum (3,4) transverse mass]
0d0                 [R(jet,lept)_min]
0d0                 [R(lept,lept)_min]
0d0                 [Delta_eta(jet,jet)_min]
.false.             [jets_opphem]
0                   [lepbtwnjets_scheme]
0d0                 [ptmin_bjet]
99d0                [etamax_bjet]
```

- List of relevant cuts for gg->4l:
  - **m34min(max):** Z1Mass cut
  - **m56min(max):** Z2Mass cut
  - **m3456min(max) :** m4l cut
  - **makecuts:** Do lepton cuts
  - **ptlepton(2nd+)_min:** lepton pt cuts
    - new in JHUGen 7.5.5 cuts are stable to 0 pt
  - **|etalepton(2nd+)|** : eta cuts on leptons

Full MCFM manual can be found in MCFM-JHUGen/Doc/mcfm.pdf

MCFM input card (for gridpacks)

- This information is helpful for the following use cases:
- *Submitting multiple jobs at once with high initial integration steps*
- Useful for batch production of precise samples
- *Creating/ debugging gridpacks for central production*
- Required for central CMS production to reduce time out issues on crab
- Usually handled by MC contacts

```
[How to resume/save a run]
.false.      [readin]
.false.      [writeout]
''       [ingridfile]
''       [outgridfile]
```

- **readin:** set to false on intial run (always true after)
- **writeout:** set to true on initial run (always false after)
- **ingridfile:** name of grid file to read in (after initial run)
- **outgridfile:** name of grid file to output

- Example: Simulate process 128 with precompiled grid
  - Step 1: **writeout**=true **outgridfile**=''
    - Expected grid file = <some_name>_grid
  - Step 2: **readin**=true **ingridfile**='<some_name>_grid'

Anomalous couplings with MCFM

- Anomalous couplings can be implemented in ggZZ processes
- Steps to generate ggZZ with anomalous couplings
- *Step 1: Open src/User/mdata.f*

```
c       Begin anomalous couplings
        include 'spinzerohiggs_anomcoupl.f'
        include 'AnomZffCouplings.f'

        data AllowAnomalousCouplings / 0 / ! Dis
        data AllowAnomalousZffCouplings / 0 / !
        data distinguish_HWWcouplings / 0 /
        data AnomalCouplPR,AnomalCouplDK / 1, 1/
        data channeltoggle_stu / 2 /
        data vvhvvtoggle_vbfvh / 2 /
```

- *Step 2: Set AllowAnomalousCouplings = 1*
- *Optional: Set AllowAnomalousZffCouplings = 1*
- *Step 3: Set the anomalous couplings of interest*
- *Step 4: run make again*

Anomalous Couplings with MCFM

- Anomalous couplings in gg -> H -> ZZ on production side:

```
4th generation quark masses
data mt_4gen / 100000d0 /
data mb_4gen / 100000d0 /

MARKUS: anomalous couplings for the first resonance
Hgg and Hff anomalous couplings
data kappa_top / (0d0,0d0) / ! SM=1,0
data kappa_tilde_top / (0d0,0d0) / ! SM=0,0
data kappa_bot / (0d0,0d0) / ! SM=1,0
data kappa_tilde_bot / (0d0,0d0) / ! SM=0,0
data ghg2 / (0d0,0d0) / ! SM=0,0
data ghg3 / (0d0,0d0) / ! SM=0,0
data ghg4 / (0d0,0d0) / ! SM=0,0

data kappa_4gen_top / (0d0,0d0) /
data kappa_tilde_4gen_top / (0d0,0d0) /
data kappa_4gen_bot / (0d0,0d0) /
data kappa_tilde_4gen_bot / (0d0,0d0) /
data ghg2_4gen / (0d0,0d0) /
data ghg3_4gen / (0d0,0d0) /
data ghg4_4gen / (0d0,0d0) /
```

m_<>_4gen = fourth generation fermion mass

CP-Even CP-Odd

$$\mathcal{L}(Hf\bar{f}) = -\frac{m_f}{v}\bar{\psi}_f\left(\kappa_f + \mathrm{i}\,\tilde{\kappa}_f\gamma_5\right)\psi_f H$$

kappa_(top/bot) kappa_tilde_(top/bot)

ghg(2/4): effective point like coupling ggH (CP-Even/CP-odd)

Same as about, but for fourth generation fermions

Anomalous Couplings with MCFM

- Anomalous couplings in gg -> H -> ZZ on decay side:

$$A(HV_1V_2) = \frac{1}{v}\left\{ M_{V_1}^2 \left(g_1^{VV} + \frac{\kappa_1^{VV} q_1^2 + \kappa_2^{VV} q_2^2}{\left(\Lambda_1^{VV}\right)^2} + \frac{\kappa_3^{VV}(q_1+q_2)^2}{\left(\Lambda_Q^{VV}\right)^2} + \frac{2 q_1 \cdot q_2}{M_{V_1}^2} g_2^{VV} \right)(\varepsilon_1 \cdot \varepsilon_2) - 2 g_2^{VV} (\varepsilon_1 \cdot q_2)(\varepsilon_2 \cdot q_1) - 2 g_4^{VV} \varepsilon_{\varepsilon_1 \varepsilon_2 q_1 q_2} \right\},$$

- How would I know what coupling to modify in the fortran code???

Name in JHUGen/MCFM ↔ **Name in Amplitude**

gh<VV><i> ↔ g_i^{VV}

VV String	VV
z	ZZ
zgs	Zγ
gsgs	γγ
zgs1_prime	K Zγ
z1_prime	K ZZ

Anomalous Zff couplings in MCFM

- New feature added in JHUGen v7.5.5!
  - *Anomalous Zff coupling in ggZZ loop*

```
Anomalous Couplings for the Z to up type quarks in production
Note, these are called as shifts to SM values
data clanou / 0d0 / ! SM = 0
data cranou / 0d0 / ! SM = 0

Anomalous Couplings for the Z to down type quarks in production
Note, these are called as shifts to SM values
data clanod / 0d0 / ! SM = 0
data cranod / 0d0 / ! SM = 0
```

  - *Anomalous Zff couplings in Z decay*

```
right handed Z couplings to charged leptons (set separately for each Z)
data reZ / 0d0 / ! SM = (-2*-1*xw_inp)/(2*sqrt(xw_inp*(1-xw_inp)))

left handed Z couplings to charged leptons (set separately for each Z)
data leZ / 0d0 / ! SM = (-1-2*-1*xw_inp)/(2*sqrt(xw_inp*(1-xw_inp)))

End anomalous couplings
left handed Z couplings to neutrinos
data lnZ / 0d0 / ! SM = (1)/(2*sqrt(xw_inp*(1-xw_inp)))

right handed Z couplings to neutrinos
data rnZ / 0d0 / ! SM = 0
```

Note: Must flag AllowAnomalousZffCouplings in mdata.f

MCFM Tutorial

- Goal: Produce an LHE file for signal + background interference with anomalous couplings

- Note: This tutorial is designed to be run on lxplus9
  - *Could just as easily be run on any cluster with cvmfs /cmsrel*

- Link to git repo where I store the scripts in case you get lost/fall behind/want to use the examples later:
  - *https://github.com/Offshell-Workshop-LPC/JHUGenTutorial*
  - Note that command line information is mostly stored in this PowerPoint (will update later if there is interest)

MCFM Tutorial (Installation)

mkdir OffShell_Tutorial
cd OffShell_Tutorial
cmsrel CMSSW_13_3_1
cd CMSSW_13_3_1/src
cmsenv
cd ../../

wget https://spin.pha.jhu.edu/Generator/JHUGenerator.v7.5.5.tar.gz
tar -xf JHUGenerator.v7.5.5.tar.gz

cd JHUGenerator.v7.5.5/MCFM-JHUGen
./Install
If linking LHAPDF you would edit the make file

```
# Replace this with the location of LHAPDF on your system (if desired)
LHAPDFLIB   =/cvmfs/cms.cern.ch/el9_amd64_gcc12/external/lhapdf/6.4.0-52852f9a177b8e8b5b72e2ae6b1327b6/lib/
```

```
PDFROUTINES = LHAPDF
```

make

MCFM Tutorial (anomalous couplings)

First use vim or some other text editor to open src/User/mdata.f

Before

After

```
data AllowAnomalousCouplings / 0 / !
data AllowAnomalousZffCouplings / 0 /
```

```
data AllowAnomalousCouplings / 1 / !
data AllowAnomalousZffCouplings / 0 /
```

Set SM Gluon Fusion Loop

```
data kappa_top / (0d0,0d0) / ! SM=1,0
data kappa_tilde_top / (0d0,0d0) / ! SM=0,0
data kappa_bot / (0d0,0d0) / ! SM=1,0
data kappa_tilde_bot / (0d0,0d0) / ! SM=0,0
```

```
data kappa_top / (1d0,0d0) / ! SM=1,0
data kappa_tilde_top / (0d0,0d0) / ! SM=0,0
data kappa_bot / (1d0,0d0) / ! SM=1,0
data kappa_tilde_bot / (0d0,0d0) / ! SM=0,0
```

Remove SM HZZ coupling

```
data ghz1 / (1d0,0d0) /
```

```
data ghz1 / (0d0,0d0) /
```

Turn on anomalous coupling

```
data ghz4 / (0d0,0d0) /
```

```
data ghz4 / (1d0,0d0) /
```

Now run make!

MCFM Tutorial (edit input card)

Now use vim or other text editor to open **Bin/input.Dat**

Before

```
-1                      [nevtrequested]
```

After

```
100                     [nevtrequested]
```

Switch Process

Before:

```
[General options to specify the process and execution]
81                [nproc]
'lord'            [part 'lord','real' or 'virt','tota']
'test'            ['runstring']
8000d0            [sqrts in GeV]
```

After:

```
[General options to specify the process and execution]
131.ELMU                    [nproc]
'lord'            [part 'lord','real' or 'virt','tota']
'test'            ['runstring']
13000d0           [sqrts in GeV]
```

For simplicity we will keep the default invariant mass and lepton cuts

All that is left to do:

cd Bin

./mcfm

Now we have an lhe file named: ggZZ4l_lord_NNPDF31_125_125_125_ELMU_test.lhe

Electroweak off-shell JHUGen

- Remember:
- Couplings parametrized by g_i or a_i -> g_1(SM-Tree Level),g_2(CP-Even Dim-6) , g_4(CP-Odd Dim-6)

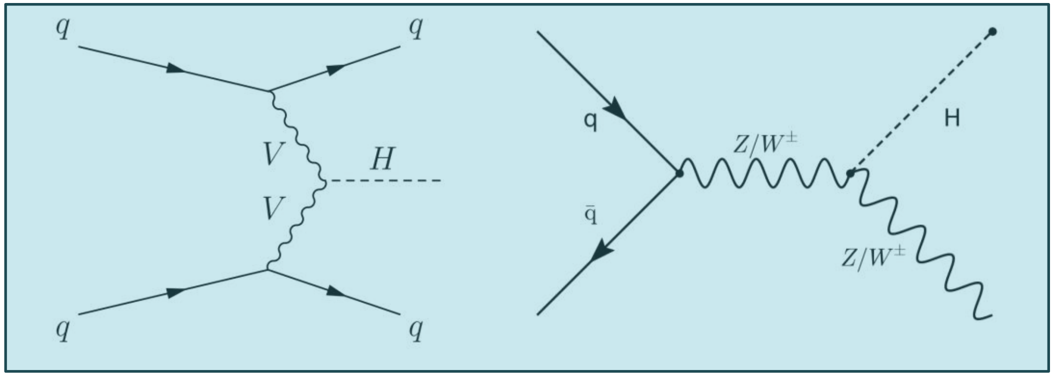
$$A(HV_1V_2)=\frac{1}{v}\left\{ M_{V_1}^2\left(g_1^{VV}+\frac{\kappa_1^{VV}q_1^2+\kappa_2^{VV}q_2^2}{\left(\Lambda_1^{VV}\right)^2}+\frac{\kappa_3^{VV}(q_1+q_2)^2}{\left(\Lambda_Q^{VV}\right)^2}+\frac{2q_1\cdot q_2}{M_{V_1}^2}g_2^{VV}\right)(\varepsilon_1\cdot\varepsilon_2) -2g_2^{VV}(\varepsilon_1\cdot q_2)(\varepsilon_2\cdot q_1)-2g_4^{VV}\varepsilon_{\varepsilon_1\,\varepsilon_2\,q_1\,q_2}\right\},$$

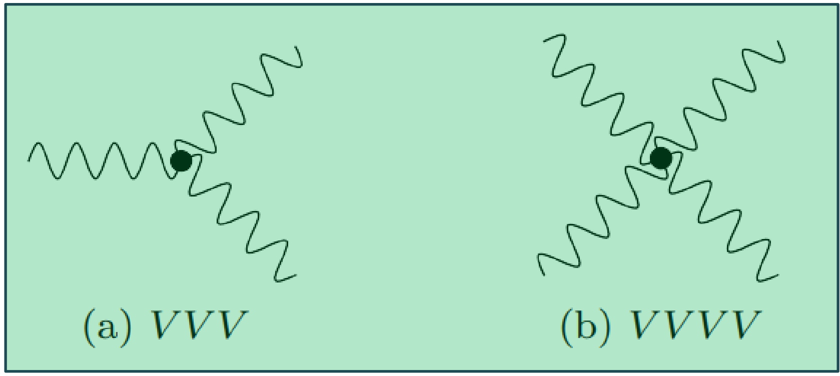
All can be mapped to EFT Lagrangian

$$\mathcal{L}_{\rm hvv} = \frac{h}{v}\Big[(1+\delta c_z)\,\frac{(g^2+g'^2)v^2}{4}Z_\mu Z_\mu + c_{zz}\,\frac{g^2+g'^2}{4}Z_{\mu\nu}Z_{\mu\nu} + c_{z\Box}g^2 Z_\mu\partial_\nu Z_{\mu\nu} + \tilde{c}_{zz}\,\frac{g^2+g'^2}{4}Z_{\mu\nu}\tilde{Z}_{\mu\nu}$$
$$+(1+\delta c_w)\,\frac{g^2v^2}{2}W_\mu^+W_\mu^- + c_{ww}\,\frac{g^2}{2}W_{\mu\nu}^+W_{\mu\nu}^- + c_{w\Box}g^2\left(W_\mu^-\partial_\nu W_{\mu\nu}^+ + \text{h.c.}\right) + \tilde{c}_{ww}\,\frac{g^2}{2}W_{\mu\nu}^+\tilde{W}_{\mu\nu}^-$$
$$+c_{z\gamma}\,\frac{e\sqrt{g^2+g'^2}}{2}Z_{\mu\nu}A_{\mu\nu} + \tilde{c}_{z\gamma}\,\frac{e\sqrt{g^2+g'^2}}{2}Z_{\mu\nu}\tilde{A}_{\mu\nu} + c_{\gamma\Box}gg'Z_\mu\partial_\nu A_{\mu\nu}$$
$$+c_{\gamma\gamma}\,\frac{e^2}{4}A_{\mu\nu}A_{\mu\nu} + \tilde{c}_{\gamma\gamma}\,\frac{e^2}{4}A_{\mu\nu}\tilde{A}_{\mu\nu} + c_{gg}\,\frac{g_s^2}{4}G_{\mu\nu}^aG_{\mu\nu}^a + \tilde{c}_{gg}\,\frac{g_s^2}{4}G_{\mu\nu}^a\tilde{G}_{\mu\nu}^a\Big]\,,$$

Electroweak off-shell JHUGen

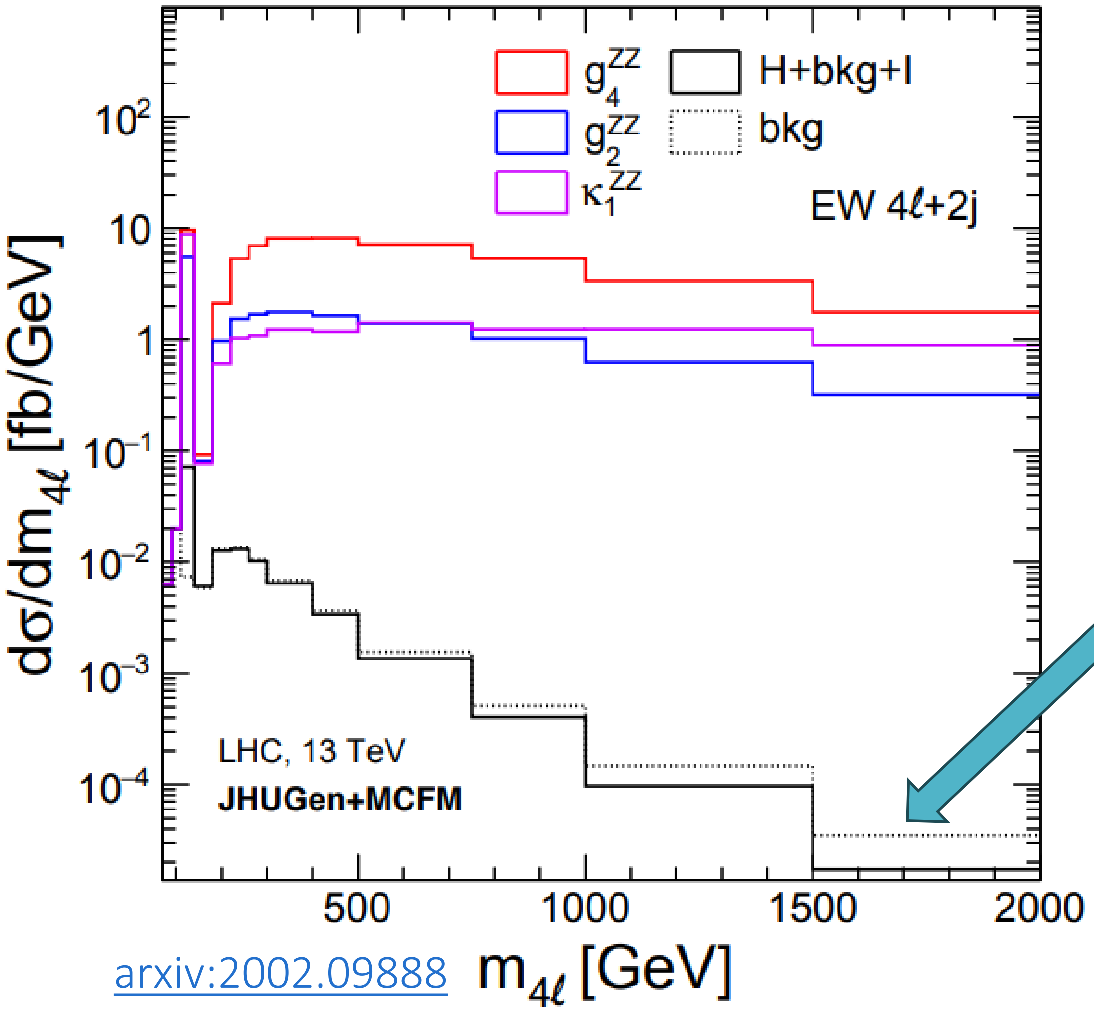
- Electroweak off-shell production is fully supported in JHUGen
- *JHUGen interfaces with MCFM matrix elements*
- *Electroweak production is defined as any qq->ZZ process*
- Simulation of Pure higgs signal, qqZZ background, signal + bkg +interference
- This includes VBF and VH production and VVV/VVVV backgrounds





arxiv:2002.09888

Interference is important!



arxiv:2002.09888

Interference from Higgs + bkg suppresses events at high m4l

Coupling naming convention

- HVV coupling naming scheme is the same as described in ggZZ section
- Triple gauge and quartic gauge couplings naming conventions:

$$\begin{aligned}\mathcal{L}_{\rm tgc} = &\quad {\rm ie}\left(W^+_{\mu\nu}W^-_\mu - W^-_{\mu\nu}W^+_\mu\right)A_\nu + {\rm ie}\left[(1+\delta\kappa_\gamma)A_{\mu\nu}\,W^+_\mu W^-_\nu + \tilde{\kappa}_\gamma \tilde{A}_{\mu\nu}\,W^+_\mu W^-_\nu\right] \\ &+ \ {\rm ie}\frac{c_w}{s_w}\left[(1+\delta g_{1,z})\left(W^+_{\mu\nu}W^-_\mu - W^-_{\mu\nu}W^+_\mu\right)Z_\nu + (1+\delta\kappa_z)\,Z_{\mu\nu}\,W^+_\mu W^-_\nu + \tilde{\kappa}_z\,\tilde{Z}_{\mu\nu}\,W^+_\mu W^-_\nu\right]\end{aligned}$$

```
dFour_A / (0d0,0d0) /
dFour_Z / (0d0,0d0) /
```

$$\begin{aligned}\mathcal{L}_{\rm qgc} = &\quad e^2(W^+_\mu A_\mu W^-_\nu A_\nu - W^+_\mu W^-_\mu A_\nu A_\nu) + \frac{e^2}{2s_w^2}(1+2c_w^2\delta g_{1,z})(W^+_\mu W^+_\mu W^-_\nu W^-_\nu - W^+_\mu W^-_\mu W^+_\nu W^-_\nu) \\ &+ \ e^2\frac{c_w^2}{s_w^2}(1+2\delta g_{1,z})(W^+_\mu Z_\mu W^-_\nu Z_\nu - W^+_\mu W^-_\mu Z_\nu Z_\nu) \\ &+ \ e^2\frac{c_w}{s_w}(1+\delta g_{1,z})(W^+_\mu Z_\mu W^-_\nu A_\nu + W^+_\mu A_\mu W^-_\nu Z_\nu - 2W^+_\mu W^-_\mu Z_\nu A_\nu)\,.\end{aligned}$$

```
dV_A / (1d0,0d0) /
dP_A / (1d0,0d0) /
dM_A / (1d0,0d0) /
dV_Z / (1d0,0d0) /
dP_Z / (1d0,0d0) /
dM_Z / (1d0,0d0) /
dZZWpWm / (1d0,0d0) /
dZAWpWm / (1d0,0d0) /
dAAWpWm / (1d0,0d0) /
```

JHUGen Command Line Inputs

- JHUGen is run from the command line so inputs are of the form:
- *./JHUGen <command line arg 1>= <value> <command line arg 2> = <value>*
- Process:
- *66,67,68: Electroweak qq -> VV+JJ (Signal,Background,Signal+Background)*
- *70,71,72: Electroweak qq -> VV+ll (Signal,Background,Signal+Background)*
- Still in Beta
- DecayMode(1/2) (for each Vector Boson):

```
0=Z->2l,  1=Z->2q, 2=Z->2tau, 3=Z->2nu,
4=W->lnu, 5=W->2q, 6=W->taunu,
7=gamma, 8=Z->2l+2tau,
9=Z->anything, 10=W->lnu+taunu, 11=W->anything
```

For a more detailed list of options, see the manual: JHUGen_Manual.pdf

Off-Shell command line settings

```
Resonance parameters:
  MReso:              resonance mass in GeV (default=125.00)
  GaReso:             resonance width in GeV (default=0.00407)

WidthScheme:      Higgs width scheme: 1 for running width, 2 for fixed width (default),
                  3 for the CPS, 4 for alternate running width (narrow width decay products)

   pTjetcut:          Minimum pT for jets in GeV (default: 15)
   deltaRcut:         Minimum deltaR for jets (default: 0.3)
   mJJcut:            Minimum dijet mass in GeV (default: 0)
   MPhotonCutoff:     Minimum mass for off-shell photons in GeV, when included (default: 4)
   etajetcut:         Maximum |eta| for jets in off-shell EW (default: 4)
   detajetcut:        Minimum deltaeta between jets in off-shell EW (default: 2)
   JetsOppositeEta:   Require sgn(eta) to be opposite for the two jets in off-shell EW
                      (default: true)
   pTlepcut:          Minimum pT for leptons in off-shell EW, in GeV (default: 3)
   etalepcut:         Maximum |eta| for leptons in off-shell EW (default: 2.7)
   m4l_min, m4l_max:  Minimum and maximum four-lepton mass
```

For a more detailed list of options, see the manual: JHUGen_Manual.pdf

Off-Shell command line settings

```
Statistics options:
  VegasNc0:          number of evaluations for integrand scan
  VegasNc1:          number of evaluations for accept-reject sampling
  VegasNc2:          number of events for accept-reject sampling
  ReadCSmax:         Read the results of the grid generation step from a file
  CSmaxFile:         File to use for reading (if ReadCSmax is set) or writing (otherwise)
                     the results of the grid generation step.  Depending on the process,
                     suffixes are appended to this base name. (default: DataFile without .lhe)
```

```
  Interf:            0=neglect interference for 4f final states,
                     1=include interference
```

For a more detailed list of options, see the manual: JHUGen_Manual.pdf

Tutorial JHUGen EW

The goal of this tutorial will be to simulate EW off-shell higgs production including the interference with background

- This includes VBF -> H +JJ and VH -> H + JJ

General steps required to produce these samples:

1. Compile MELA
2. Compile JHUGen and link MELA
3. Generate the grids for all 164 partonic channels
4. Generate events for all 164 partonic channels and merge output lhe

Tutorial JHUGen EW (compile MELA)

- First we need to install MELA!

cd JHUGen

cd JHUGenerator.v7.5.5/JHUGenMELA/

./setup.sh

eval $(./setup.sh env)

cd ../JHUGenerator/

This could take a few minutes, but it is required to run the electroweak production since we need to interface with MCFM

Compiling JHUGen

Using a text editor such as vim, open the makefile

```
################
## USER FLAGS ##
################
## link pdfs via LHA library ('Yes' or 'No')
UseLHAPDF=No
# remember to export
#           LD_LIBRARY_PATH=/.../LHAPDF-x.y.z/lib/:${LD_LIBRARY_PATH}
#           LHAPDF_DATA_PATH=/.../LHAPDF-x.y.z/share/LHAPDF/:${LHAPDF_DATA_PATH}

## link MCFM libraries from MELA
linkMELA = No
# remember to export
#           LD_LIBRARY_PATH=/.../MELA/data/$SCRAM_ARCH/:${LD_LIBRARY_PATH}

## include Collier Loop Integral Library
linkCollierLib = No
```

We will only set linkMELA = Yes for this tutorial

Then: The change below is only needed for newer gcc versions

```
MELALibDir = $(MELADataDir)/$(SCRAM_ARCH)
```

→

```
MELALibDir = $(MELADataDir)/$(MELA_ARCH)
```

Now run make!

Grid Generation

For time constraints, we will only produce one grid file

First make a directory to output the gridfiles:

mkdir Grids_Output

Then as an example run:

./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000 ReweightInterf=0 ghz1=1,0 VBFoffsh_run=1 DataFile=Grids_Output/Out

Sorry about the long command line, but all of these inputs are required

You should see a file named: **Out_001_step2.grid**

Event Generation

Again, because of time constraints we will generate LHE events for a single partonic channel

I included my own pre-compiled grids in a folder on the git repo named **Grids**

./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000 ReweightInterf=0 ghz1=1,0 VBFoffsh_run=1 VegasNc2=10000 ReadCSmax DataFile=Grids/Out

Interpreting output LHE

- The meaning of VegasNc2 is different for EW off-shell production
  - *Usually in JHUGen it is the number of events requested*
  - *In EW production we precompute how many expected events should be in each channel based on what VegasNc2 is set to*
- Example from what you all just ran:

```
Total xsec with weights (use for physics):
  1   1   2        0.00102629       33
  2   2   1        0.00272673       86
  3   2  -2        0.00700763      213
  5   3   2        0.00590854      185
  6   2   3        0.00289779       92
  7   1  -1        0.00086304       23
  8  -1   1        0.00011469        2
 10   2  -1        0.00016118        3
 11  -1   2        0.00009916        2
 12   2  -1        0.00070123       23
 13  -2   1        0.00036367       11
 14  -1   2        0.20391092     6294
 15  -2   1        0.00001310        2
 16   2  -4        0.01194773      378
 17  -4   2        0.00008633        3
 18   1  -3        0.01029161      311
 19   1  -2        0.00166683       41
 20   1  -2        0.00216202       60
 21   1   4        0.02747963      862
```

Channel 1 expects 33/10000 requested events so the lhe file will only contain 33 events

When lhe files are combined the relative cross-sections from each channel are maintained in the unweighted events!

JHUGen-Lexicon

- JHUGen Lexicon is a tool for mapping various EFT basis operators from one to the other
- This allows users to translate JHUGen from one EFT basis to another during either reweighting (i.e MELA) or event generation (JHUGenerator)
- Current Features:
- *Supported Basis: JHUGen Amplitude, Higgs basis, Warsaw basis*
- *Basis Constraints: Fix W mass to SM value, SU(2) x U(1) symmetry*
- *Triple and Quartic Gauge Couplings*
- *Convention on Covariant Derivative (for matching samples across generators)*

$$D_\mu = \partial_\mu - \mathrm{i}\frac{e}{2s_w}\sigma^i W_\mu^i - \mathrm{i}\frac{e}{2c_w}B_\mu$$ ← MadGraph convention

$$D_\mu = \partial_\mu - \mathrm{i}\frac{e}{2s_w}\sigma^i W_\mu^i + \mathrm{i}\frac{e}{2c_w}B_\mu$$ ← JHUGen convention

JHUGen Lexicon Tutorial

- Compilation is extremely easy!
- *Simply go in the JHUGenLexicon Directory and do make*
- Example usage: Translate amplitude basis to warsaw basis
- ***./JHUGenLexicon input_basis=eft_jhu output_basis=warsaw ghz1=2,0 ghz4=1,0 vev_lam=0.060624***
- ./JHUGenLexicon –help for more info about options

$$
\begin{aligned}
\delta g_1^{ZZ} &= \frac{v^2}{\Lambda^2}\left(2C_{H\Box} + \frac{6e^2}{s_w^2}C_{HWB} + \left(\frac{3c_w^2}{2s_w^2} - \frac{1}{2}\right)C_{HD}\right), \\
\kappa_1^{ZZ} &= \frac{v^2}{\Lambda^2}\left(-\frac{2e^2}{s_w^2}C_{HWB} + \left(1 - \frac{1}{2s_w^2}\right)C_{HD}\right), \\
g_2^{ZZ} &= -2\frac{v^2}{\Lambda^2}\left(s_w^2 C_{HB} + c_w^2 C_{HW} + s_w c_w C_{HWB}\right), \\
g_2^{Z\gamma} &= -2\frac{v^2}{\Lambda^2}\left(s_w c_w\left(C_{HW} - C_{HB}\right) + \frac{1}{2}\left(s_w^2 - c_w^2\right)C_{HWB}\right) \\
g_2^{\gamma\gamma} &= -2\frac{v^2}{\Lambda^2}\left(c_w^2 C_{HB} + s_w^2 C_{HW} - s_w c_w C_{HWB}\right),
\end{aligned}
$$

$$
\begin{aligned}
g_2^{gg} &= -2\frac{v^2}{\Lambda^2}C_{HG}, \\
g_4^{ZZ} &= -2\frac{v^2}{\Lambda^2}\left(s_w^2 C_{H\widetilde{B}} + c_w^2 C_{H\widetilde{W}} + s_w c_w C_{H\widetilde{W}B}\right), \\
g_4^{Z\gamma} &= -2\frac{v^2}{\Lambda^2}\left(s_w c_w\left(C_{H\widetilde{W}} - C_{H\widetilde{B}}\right) + \frac{1}{2}\left(s_w^2 - c_w^2\right)C_{H\widetilde{W}B}\right) \\
g_4^{\gamma\gamma} &= -2\frac{v^2}{\Lambda^2}\left(c_w^2 C_{H\widetilde{B}} + s_w^2 C_{H\widetilde{W}} - s_w c_w C_{H\widetilde{W}B}\right), \\
g_4^{gg} &= -2\frac{v^2}{\Lambda^2}C_{H\widetilde{G}},
\end{aligned}
$$

Translations utilized behind the scenes

arxiv:2109.13363

End of Tutorial

- Thank you all for participating in this tutorial
- There are many more rich features of JHUGen that have not been explored in this tutorial (mostly due to time constraints)
  - It is a great tool for off-shell Higgs physics (and on-shell!)

- Please feel free to email authors with questions about any of the JHUGen packages in the future!