MadGraph5aMC@NLO Polarization Tutorial

#COMETA Polarization Workshop – Toulouse, FR

Richard Ruiz

Institute of Nuclear Physics – Polish Academy of Science (IFJ PAN)

24 September 2024

Outline

- **Pt1:** `MadGraph5_aMC@NLO` history (brief) and polarized, external parton scattering formalism (brief)
- **Pt2:** polarized, same-sign $W^{\pm}W^{\pm}$ scattering at LO in `mg5amc`
- **Pt3:** $Z_\lambda Z_\lambda$ pairs in $gg \to e^+e^-\mu^+\mu^-$ with `vpolar+mg5amc`

pt1. MadGraph5_aMC@NLO (mg5amc) in a nutshell

History (1 slide)

MG5aMC is the 5th (or 6th) iteration of the Monte Carlo (MC) event generator `MadisonGraph` (or `MadGraph`) by Stelzer and Long at Wisconsin

[hep-ph/9401258]

- For a given scattering process, generates Feynman graphs and helicity amplitudes (`HELAS` routines) for *fast* numerical evaluation

History (1 slide)

MG5aMC is the 5th (or 6th) iteration of the Monte Carlo (MC) event generator `MadisonGraph` (or `MadGraph`) by Stelzer and Long at Wisconsin

[hep-ph/9401258]

- For a given scattering process, generates Feynman graphs and helicity amplitudes (HELAS routines) for *fast* numerical evaluation

- Phase space integration via MC sampling (`MadEvent`)
  `MadEvent` writes phase space points (external momenta!) to file with integration weight (probability), i.e., `MG+ME` is a MC event generator

  Doing this efficienctly and robustly is difficult but doable. Maltoni, Stelzer [hep-ph/0208156]

History (1 slide)

MG5aMC is the 5th (or 6th) iteration of the Monte Carlo (MC) event generator `MadisonGraph` (or `MadGraph`) by Stelzer and Long at Wisconsin

[hep-ph/9401258]

- For a given scattering process, generates Feynman graphs and helicity amplitudes (HELAS routines) for *fast* numerical evaluation

- Phase space integration via MC sampling (`MadEvent`)
  `MadEvent` writes phase space points (external momenta!) to file with integration weight (probability), i.e., `MG+ME` is a MC event generator

  Doing this efficienctly and robustly is difficult but doable. Maltoni, Stelzer [hep-ph/0208156]

- + arbitrary color structures, +jet matching/merging
  + spin correlated decays of resonances (`MadSpin`), + amplitude support for arbitrary Feynman Rule (`ALOHA`), , + loop-induced processes (`MadLoop`)

History (1 slide)

MG5aMC is the 5th (or 6th) iteration of the Monte Carlo (MC) event generator `MadisonGraph` (or `MadGraph`) by Stelzer and Long at Wisconsin

[hep-ph/9401258]

- For a given scattering process, generates Feynman graphs and helicity amplitudes (`HELAS` routines) for *fast* numerical evaluation

- Phase space integration via MC sampling (`MadEvent`)
  `MadEvent` writes phase space points (external momenta!) to file with integration weight (probability), i.e., `MG+ME` is a MC event generator

  Doing this efficienctly and robustly is difficult but doable. Maltoni, Stelzer [hep-ph/0208156]

- + arbitrary color structures, +jet matching/merging
  + spin correlated decays of resonances (`MadSpin`), + amplitude support for arbitrary Feynman Rule (`ALOHA`), , + loop-induced processes (`MadLoop`)

- Merger with `MC@NLO` for NLO in QCD [1405.0301] and NLO in EW [1804.10017]

Then and Now (Publicity Plots)

(L) Early practioners of MadGraph

M.S. Chanowitz, W.B. Kilgore / Physics Letters B 347 (1995) 387–393

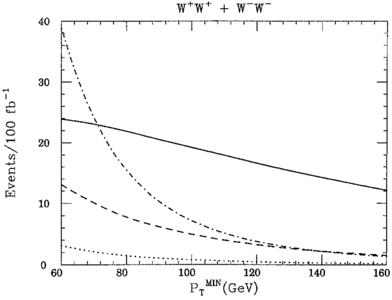
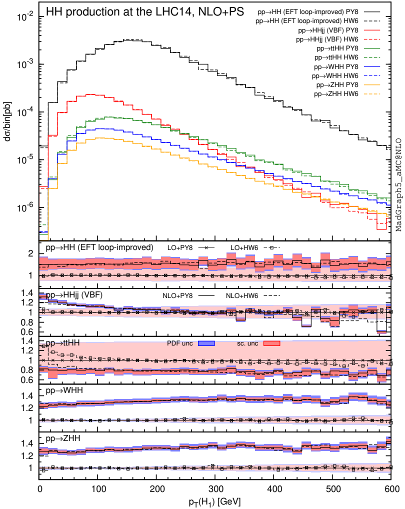


Fig. 2. The number of events per 100 fb^{-1} for which both like-sign leptons have transverse momentum greater than p_T^{MIN}. The rapidity and azimuthal angle cuts on the like-sign leptons are at the optimum values specified in Table 1 for $m_\rho = 2.52$ TeV. All events with the third lepton inside its acceptance region are rejected. The solid, dashed, dot-dashed, and dotted lines are, respectively, the signal and the backgrounds from $\bar{q}q \to l^\pm \nu_l \bar{l}l$ and from $qq \to qqW^+W^+/W^-W^-$ in orders α_W^2 and $\alpha_W\alpha_S$.

(R) MadGraph5_aMC@NLO today



Frederix, et al [1401.7340]

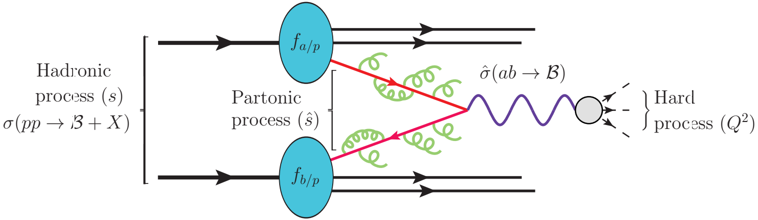
scattering with polarized external partons

To get pp scattering rates, `mg5amc` uses the Collinear Factorization Thm

Collins, Soper, Sterman ('85,'88,'89); Collins, Foundations of pQCD (2011)

$$d\sigma(pp \to W\gamma + X) = \sum_{i,j} f_i \otimes f_j \otimes \Delta_{ij} \otimes d\hat{\sigma}(ij \to W\gamma) + \mathcal{O}\left(\Lambda^p_{\rm NP}/Q^{p+2}\right)$$

hadron-level scattering probabilities are the product (convolution) of parton-dist. (PDFs), -emission (Sudakov), and -scattering probs. ($|\mathcal{M}|^2$)

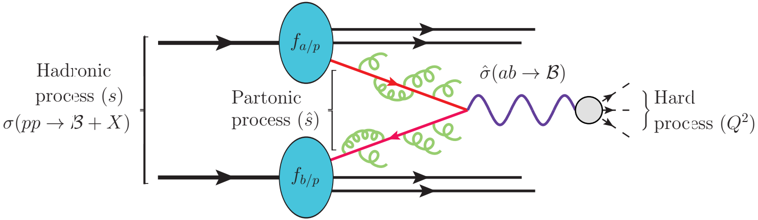


To get pp scattering rates, `mg5amc` uses the Collinear Factorization Thm

Collins, Soper, Sterman ('85,'88,'89); Collins, Foundations of pQCD (2011)

$$d\sigma(pp \to W\gamma + X) = \sum_{i,j} f_i \otimes f_j \otimes \Delta_{ij} \otimes d\hat{\sigma}(ij \to W\gamma) + \mathcal{O}\left(\Lambda_{\rm NP}^{p}/Q^{p+2}\right)$$

hadron-level scattering probabilities are the product (convolution) of parton-dist. (PDFs), -emission (Sudakov), and -scattering probs. ($|\mathcal{M}|^2$)



The partonic scattering rate is given by the usual (textbook) expression:

$$d\hat{\sigma}(ij \to W\gamma) = \underbrace{\frac{1}{2Q^2}}_{\text{hard scale}} \underbrace{\overline{|\mathcal{M}(ij \to W\gamma)|^2}}_{\text{dof avg./summed.}}$$

The *unpolarized* external parton scattering rate is given by the dof-averaged[1] (initial states) and dof-summed (final state) matrix element:

$$\overline{|\mathcal{M}(ij \to W\gamma)|^2} = \underbrace{\frac{1}{\mathcal{S}_i \mathcal{S}_j}}_{\text{spin dof}} \underbrace{\frac{1}{N_c^i N_c^j}}_{\text{color dof}} \sum_{\text{dof}} |\underbrace{\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}} \gamma_{\tilde{\lambda}'})}_{\text{ME in helicity basis}}|^2$$

[1] degrees of freedom = all discrete quantum numbers, e.g., color, spin, electric charge

The *unpolarized* external parton scattering rate is given by the dof-averaged[1] (initial states) and dof-summed (final state) matrix element:

$$\overline{|\mathcal{M}(ij \to W\gamma)|^2} = \underbrace{\frac{1}{\mathcal{S}_i \mathcal{S}_j}}_{\text{spin dof}} \underbrace{\frac{1}{N_c^i N_c^j}}_{\text{color dof}} \sum_{\text{dof}} |\underbrace{\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}} \gamma_{\tilde{\lambda}'})}_{\text{ME in helicity basis}}|^2$$

For **polarized** scattering, **truncate spin averaging/summing**

$$\overline{|\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}} \gamma_{\tilde{\lambda}'})|^2} = \underbrace{\frac{1}{N_c^i N_c^j}}_{\text{color dof}} \sum_{\text{dof}} |\underbrace{\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}} \gamma_{\tilde{\lambda}'})}_{\text{ME in helicity basis}}|^2$$

[1] degrees of freedom = all discrete quantum numbers, e.g., color, spin, electric charge

The *unpolarized* external parton scattering rate is given by the dof-averaged[1] (initial states) and dof-summed (final state) matrix element:

$$\overline{|\mathcal{M}(ij \to W\gamma)|^2} = \underbrace{\frac{1}{\mathcal{S}_i\mathcal{S}_j}}_{\text{spin dof}} \underbrace{\frac{1}{N_c^i N_c^j}}_{\text{color dof}} \sum_{\text{dof}} |\underbrace{\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'})}_{\text{ME in helicity basis}}|^2$$

For **polarized** scattering, **truncate spin averaging/summing**

$$\overline{|\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'})|^2} = \underbrace{\frac{1}{N_c^i N_c^j}}_{\text{color dof}} \sum_{\text{dof}} |\underbrace{\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'})}_{\text{ME in helicity basis}}|^2$$

The two are related by reintroducing spin averaging/summing

$$\overline{|\mathcal{M}(ij \to W\gamma)|^2} = \underbrace{\frac{1}{\mathcal{S}_i\mathcal{S}_j}}_{\text{spin dof}} \sum_{\lambda,\lambda',\tilde{\lambda}\tilde{\lambda}'} \overline{|\mathcal{M}(i_\lambda j_{\lambda'} \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'})|^2}$$

[1] degrees of freedom = all discrete quantum numbers, e.g., color, spin, electric charge

Polarized External Parton Scattering (3/3)

Polarized parton scattering in LHC collisions is given by

$$d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}} = f_{i_\lambda} \otimes f_{i_{\lambda'}} \otimes \Delta_{i_\lambda j_{\lambda'}} \otimes d\hat{\sigma}(i_\lambda j_{\lambda'} \to W_\lambda \gamma_{\tilde{\lambda}'})$$

- f_{i_λ} is the PDF for parton i with helicity λ in *unpolarized proton p*
- $\Delta_{i_\lambda j_{\lambda'}}$ is the parton shower / evolution for i, j with helicities λ, λ'

Polarized External Parton Scattering (3/3)

Polarized parton scattering in LHC collisions is given by

$$d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}} = f_{i_\lambda} \otimes f_{i_{\lambda'}} \otimes \Delta_{i_\lambda j_{\lambda'}} \otimes d\hat{\sigma}(i_\lambda j_{\lambda'} \to W_\lambda \gamma_{\tilde{\lambda}'})$$

- f_{i_λ} is the PDF for parton i with helicity λ in *unpolarized proton p*
- $\Delta_{i_\lambda j_{\lambda'}}$ is the parton shower / evolution for i, j with helicities λ, λ'

Again, unpolarized scattering is recovered by spin averaging/summing

$$d\sigma(pp \to W\gamma + X) = \underbrace{\sum_{i_\lambda, j_{\lambda'}}}_{\text{partons}} \underbrace{\frac{1}{\mathcal{S}_i \mathcal{S}_j}}_{\text{spin dof}} \underbrace{\sum_{\lambda, \lambda', \tilde{\lambda}\tilde{\lambda}'}}_{\text{helicities}} d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}}$$

Polarized External Parton Scattering (3/3)

Polarized parton scattering in LHC collisions is given by

$$d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}} = f_{i_\lambda} \otimes f_{i_{\lambda'}} \otimes \Delta_{i_\lambda j_{\lambda'}} \otimes d\hat{\sigma}(i_\lambda j_{\lambda'} \to W_\lambda \gamma_{\tilde{\lambda}'})$$

- f_{i_λ} is the PDF for parton i with helicity λ in *unpolarized proton p*
- $\Delta_{i_\lambda j_{\lambda'}}$ is the parton shower / evolution for i, j with helicities λ, λ'

Again, unpolarized scattering is recovered by spin averaging/summing

$$d\sigma(pp \to W\gamma + X) = \underbrace{\sum_{i_\lambda, j_{\lambda'}}}_{\text{partons}} \underbrace{\frac{1}{\mathcal{S}_i \mathcal{S}_j}}_{\text{spin dof}} \underbrace{\sum_{\lambda,\lambda',\tilde{\lambda}\tilde{\lambda}'}}_{\text{helicities}} d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}}$$

For polarized final states from unpolarized initial states:

$$d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X) = \sum_{i_\lambda, j_{\lambda'}} \underbrace{\frac{1}{\mathcal{S}_i \mathcal{S}_j}}_{\text{spin dof}} \sum_{\lambda,\lambda'} d\sigma(pp \to W_{\tilde{\lambda}}\gamma_{\tilde{\lambda}'} + X)|_{i_\lambda, j_{\lambda'}}$$

what about internal/intermediate states?

Propagator decomposition (weak boson)

Popular (and successful) paradigm: decompose numerator of propagator via completeness relationship

care is need at this step!

$$-g_{\mu\nu} + q_\mu q_\nu / M_V^2 = \sum_{\lambda=\pm,0,S} \varepsilon_\mu(q,\lambda)\varepsilon^*_\nu(q,\lambda)$$

vector boson propagator becomes sum over **truncated propagators**

$$\Pi^V_{\mu\nu}(q) = \frac{-i\left(g_{\mu\nu} - q_\mu q_\nu/M_V^2\right)}{q^2 - M_V^2 + iM_V\Gamma_V}$$

$$= \sum_{\lambda\in\{0,\pm1,A\}} \underbrace{\eta_\lambda}_{\pm1} \underbrace{\left(\frac{i\varepsilon_\mu(q,\lambda)\,\varepsilon^*_\nu(q,\lambda)}{q^2 - M_V^2 + iM_V\Gamma_V}\right)}_{\equiv\Pi^{V\lambda}_{\mu\nu}\ \text{truncated prop.}}$$

Propagator decomposition (weak boson)

Popular (and successful) paradigm: decompose numerator of propagator via completeness relationship

care is need at this step!

$$-g_{\mu\nu} + q_\mu q_\nu / M_V^2 = \sum_{\lambda=\pm,0,S} \varepsilon_\mu(q,\lambda)\varepsilon_\nu^*(q,\lambda)$$

vector boson propagator becomes sum over **truncated propagators**

$$\Pi_{\mu\nu}^V(q) = \frac{-i\left(g_{\mu\nu} - q_\mu q_\nu / M_V^2\right)}{q^2 - M_V^2 + iM_V\Gamma_V}$$

$$= \sum_{\lambda\in\{0,\pm1,A\}} \underbrace{\eta_\lambda}_{\pm1} \underbrace{\left(\frac{i\varepsilon_\mu(q,\lambda)\ \varepsilon_\nu^*(q,\lambda)}{q^2 - M_V^2 + iM_V\Gamma_V}\right)}_{\equiv\Pi_{\mu\nu}^{V\lambda}\ \text{truncated prop.}}$$

dPS integration in `mg5amc`: spin-correlated Narrow Width Approximation [1212.3460] with Breit-Wigner propagator (full finite-width effects)

Propagator decomposition (fermion)

Popular (and successful) paradigm: decompose numerator of propagator via completeness relationship

$$(\not{q} \ \pm \ m) \ = \ \sum_{\lambda=\pm} \ u(q,\lambda)\overline{u}(q,\lambda) \quad \text{or} \quad \sum_{\lambda=\pm} \ v(q,\lambda)\overline{v}(q,\lambda)$$

spin-1/2 fermion propagator becomes sum over **truncated propagators**

$$S_F(q) = \frac{i(\not{q}+m)}{q^2 - m_q^2 + i m_q \Gamma_q} = \underbrace{\frac{i\sum_{\lambda\in\{\pm1\}} u(q,\lambda)\overline{u}(q,\lambda)}{q^2 - m_q^2 + i m_q \Gamma_q}}_{\equiv S_F^\lambda \text{ truncated prop.}}$$

$$S_{\overline{F}}(q) = \frac{-i(\not{q}-m)}{q^2 - m_q^2 + i m_q \Gamma_q} = \underbrace{\frac{-i\sum_{\lambda\in\{\pm1\}} v(q,\lambda)\overline{v}(q,\lambda)}{q^2 - m_q^2 + i m_q \Gamma_q}}_{\equiv S_{\overline{F}}^\lambda(q) \text{ truncated prop.}}$$

dPS integration in `mg5amc`: spin-correlated Narrow Width Approximation [1212.3460] with Breit-Wigner propagator (full finite-width effects)

why all this detail?

Polarization in MGaMC

Polarization formalism in MGaMC has been used in several contexts:

- polarized final states from unpolarized initial states

  Buarque Franzosi, RR, et al (JHEP'20)[1912.01725]

- one-loop electroweak Sudakov logarithms

  Pagani & Zaro (JHEP'22) [2110.03714]

- Effective W Approximation (W and Z PDFs for high-energy leptons)

  RR, et al (JHEP'22) [2111.02442]

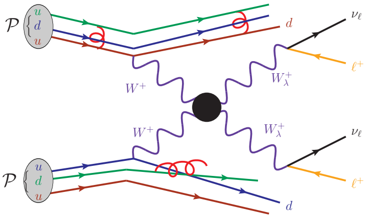
- polarization as a Feynman rule

  Javurkova, RR, et al (PLB'24) [2401.17365]

- Effective W Approximation at next-to-leading power

  Bigaran & RR [*to appear*]

pt2. polarized, same-sign $W^\pm W^\pm$



From Feynman rules to cross sections (and events!)

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set group_subprocesses False
```

From Feynman rules to cross sections (and events!)

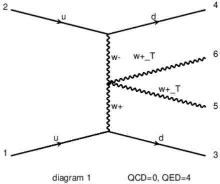
```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set group_subprocesses False
```

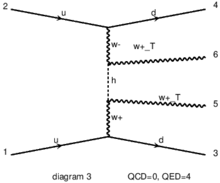
\# transverse-transverse $VV \to W^{\pm}_T W^{\pm}_T$

```
> generate p p > j j w+{T} w+{T} QCD=0 QED=4
> add process p p > j j w-{T} w-{T} QCD=0 QED=4
```

\# equivalent syntax $(\lambda = \pm)$

```
> generate p p > j j w+{+} w+{+} QCD=0 QED=4
> add process p p > j j w+{-} w+{-} QCD=0 QED=4
> add process p p > j j w-{+} w-{+} QCD=0 QED=4
> add process p p > j j w-{-} w-{-} QCD=0 QED=4
```





From Feynman rules to cross sections (and events!)

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set group_subprocesses False
```

transverse-transverse $VV \to W_T^{\pm} W_T^{\pm}$

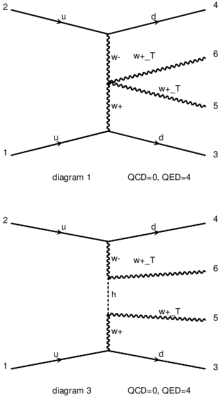
```
> generate p p > j j w+{T} w+{T} QCD=0 QED=4
> add process p p > j j w-{T} w-{T} QCD=0 QED=4
```

equivalent syntax $(\lambda = \pm)$

```
> generate p p > j j w+{+} w+{+} QCD=0 QED=4
> add process p p > j j w+{-} w+{-} QCD=0 QED=4
> add process p p > j j w-{+} w-{+} QCD=0 QED=4
> add process p p > j j w-{-} w-{-} QCD=0 QED=4
```

longitudinal $VV \to W_0^{\pm} W_T^{\mp}$

```
> generate p p > j j w+{0} w+{T} QCD=0 QED=4
> add process p p > j j w-{0} w-{T} QCD=0 QED=4
```



From Feynman rules to cross sections (and events!)

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set group_subprocesses False
```

\# transverse-transverse $VV \to W^\pm_T W^\pm_T$

```
> generate p p > j j w+{T} w+{T} QCD=0 QED=4
> add process p p > j j w-{T} w-{T} QCD=0 QED=4
```

\# equivalent syntax $(\lambda = \pm)$

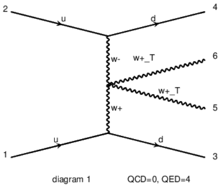
```
> generate p p > j j w+{+} w+{+} QCD=0 QED=4
> add process p p > j j w+{-} w+{-} QCD=0 QED=4
> add process p p > j j w-{+} w-{+} QCD=0 QED=4
> add process p p > j j w-{-} w-{-} QCD=0 QED=4
```

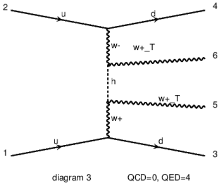
\# longitudinal $VV \to W^\pm_0 W^\mp_T$

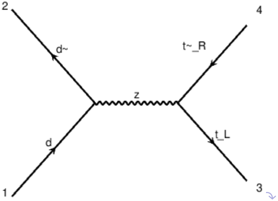
```
> generate p p > j j w+{0} w+{T} QCD=0 QED=4
> add process p p > j j w-{0} w-{T} QCD=0 QED=4
```

\# fermions $q\overline{q} \to t_L \overline{t_R}$ $(\lambda = L, R)$

```
> generate p p > t{L} t~{R} QCD=0 QED=2
```







Decay Syntax

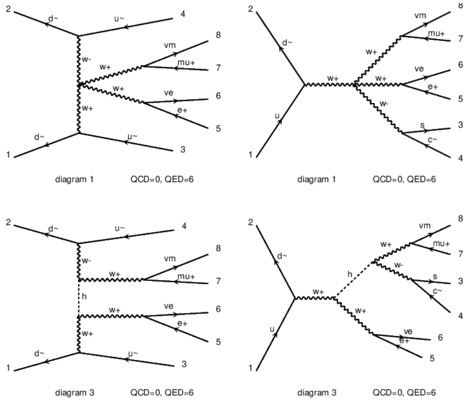
mg5 features spin-correlated Narrow Width Approximation [1212.3460]

$VV \to W_T^{\pm} W_T^{\pm} \to 2\ell 2\nu$

```
> generate p p > j j w+{T} w+{T}
                              QCD=0 QED=4,
          (w+ > e+ ve), (w+ > mu+ vm)

> add process p p > j j w-{T} w-{T}
                              QCD=0 QED=4,
          (w+ > e+ ve), (w+ > mu+ vm)

> output <MY_FAV_NAME>
```



Decay Syntax + Diagram Removal

mg5amc features two types of diagram removal:

– custom, diagram-by-diagram

coding req.:

tree: `<MGdir>/PLUGIN/user_filter.py`

1-loop: `<MGdir>/madgraph/loop/loop_diagram_generation.py`

– basic, broad-stroke [no coding]

Decay Syntax + Diagram Removal

`mg5amc` features two types of diagram removal:

– custom, diagram-by-diagram

– basic, broad-stroke [no coding]

coding req.:

tree: `<MGdir>/PLUGIN/user_filter.py`

1-loop: `<MGdir>/madgraph/loop/loop_diagram_generation.py`

Syntax for basic removal:

`$$ X` = no s-channel `X`

`/ X` = no internal `X`

`p` = `g u c d s u~ c~ d~ s~`

Decay Syntax + Diagram Removal

mg5amc features two types of diagram removal:
– custom, diagram-by-diagram
– basic, broad-stroke [no coding]

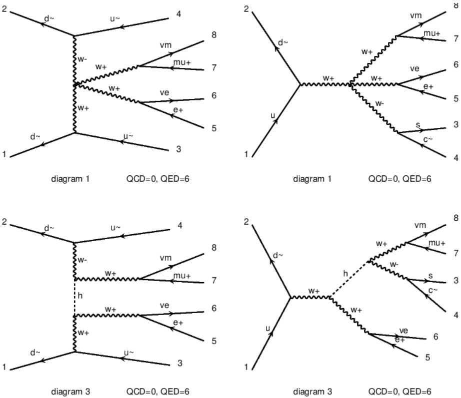
coding req.:

tree: <MGdir>/PLUGIN/user_filter.py

1-loop: <MGdir>/madgraph/loop/loop_diagram_generation.py

Syntax for basic removal:

`$$ X` = no s-channel `X`
`/ X` = no internal `X`
`p` = `g u c d s u~ c~ d~ s~`



`#` $VV \to W_T^\pm W_T^\pm \to 2\ell 2\nu$ `w/o s-chanel`

```
> generate p p > j j w+{T} w+{T}
        QCD=0 QED=4 $$ w+ w- z a / p,
          (w+ > e+ ve), (w+ > mu+ vm)

> add process p p > j j w-{T} w-{T}
        QCD=0 QED=4 $$ w+ w- z a / p,
          (w+ > e+ ve), (w+ > mu+ vm)
```

some numbers

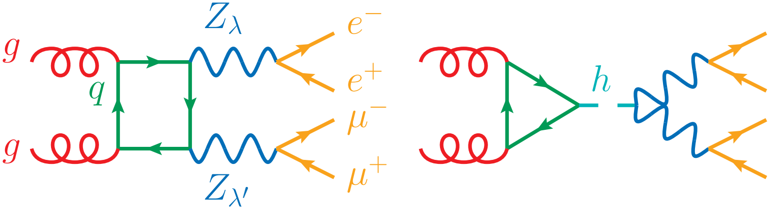
```
launch COMETA_osWW_WTWT_2el2mu
analysis=off
set pdlabel lhapdf
# NNPDF31_nlo_as_0118_luxqed
set lhaid 324900
set lhc 13
set nevents 4k
set dynamical_scale_choice 3
set me_frame [3,4,5,6]
set no_parton_cut
set ptj 20
set etaj 5
set mmjj 250
set deltaeta 2.5
set mxx_min_pdg {24:250}
set pt_min_pdg {24:  30}
set eta_max_pdg {24:  2.5}
set use_syst true
done
```

Buarque Franzosi, et al (JHEP'20) [1912.01725]

Process	p-CM SM ($a = 1$)	
	σ [fb]	$f_{\lambda\lambda'}$
jjW^+W^-	171	...
$jjW_T^+W_T^-$	119	70%
$jjW_0^+W_T^-$	20.6	12%
$jjW_T^+W_0^-$	23.8	14%
$jjW_0^+W_0^-$	5.45	3%

Process	Decay Scheme	Generator-Level Cuts		Analysis-Level Cuts	
		σ [fb]	f_λ	σ [fb]	f_λ
jjW^+W^-	MadSpin	3.818	...	3.243	...
$jjW^+W_T^-$	MadSpin	3.043	79.7%	2.567	79.2%
$jjW^+W_T^-$	OSP	3.041	79.6%	2.568	79.2%
$jjW^+W_0^-$	MadSpin	0.7824	20.5%	0.6527	20.1%
$jjW^+W_0^-$	OSP	0.7797	20.4%	0.6514	20.1%

pt3. polarized $Z_\lambda Z_\lambda$ in $gg \to e^+e^-\mu^+\mu^-$ at $\mathcal{O}(\alpha_s^2\alpha^4)$



Polarization@1-loop not (yet) supported in standalone `mg5amc`

– reweighting for loop+NWA is computationally expensive

– polarization in tree subgraphs vs polarization in loop subgraphs

– reference frame choice ✓

Polarization@1-loop not (yet) supported in standalone `mg5amc`

– reweighting for loop+NWA is computationally expensive

– polarization in tree subgraphs vs polarization in loop subgraphs

– reference frame choice ✓

Use support for BSM@1-loop:

– `SM_Loop_ZPolar`

– `SM_Loop_WPolar`

– `SM_Loop_VPolar`

feynrules.irmp.ucl.ac.be/wiki/VPolarization

Split $Z, W^\pm$ in into four states:

```
...
Definitions ->
    {Z[mu_]->Z0[mu]+ZT[mu]+ZA[mu]+ZX[mu]}
...
Definitions ->
    {W[mu_]->W0[mu]+WT[mu]+WA[mu]+WX[mu]}
```

FeynRules

Rules

A Mathematica package to calculate Feynman rules

`VPolar`: The Standard Model at NLO in QCD with helicity-polarized W and Z bosons

Contact Author

Richard Ruiz

- Institute of Nuclear Physics Polish Academy of Science (IFJ PAN)
- rruiz AT ifj.edu.pl

In collaboration with:

M. Javurkova, R.C.L. de Sá, and J. Sandesara arXiv:2401.17365 [1]
N. Buarque Franzosi, O. Mattelaer, and Sujay Shil arXiv:1912.01725 [2]

Usage resources

- For instructions and examples on using the VPolar UFO libraries, see M. Javurkova, et al, arXiv:2401.17365 [1]
- For additional background, see also D. Buarque Franzosi, et al, arXiv:1912.01725 [2]
- See **Validation** section below for additional information

- **Special note:** this UFO was developed using MG5aMC and calls the **1L**, **1T**, and **1A** propagators defined in ALOHA (see **aloha_object.py** and *create_aloha.py*). These may be defined differently in other generators. If they are not defined in your favorite generators, they must be added to the *propagators.py* file in the `VPolar` UFO. The file **particles.py** must then be updated to reflect the propagator names. R. Ruiz is happy to assist with this.

Citation requests

- If using the UFO, please cite , see M. Javurkova, et al, arXiv:2401.17365 [1]

Polarization@1-loop not (yet) supported in standalone `mg5amc`
– reweighting for loop+NWA is computationally expensive
– polarization in tree subgraphs vs polarization in loop subgraphs
– reference frame choice ✓

Use support for BSM@1-loop:
– `SM_Loop_ZPolar`
– `SM_Loop_WPolar`
– `SM_Loop_VPolar`

feynrules.irmp.ucl.ac.be/wiki/VPolarization

Split $Z, W^{\pm}$ in into four states:
– V_0 is $\lambda = 0$
– V_T is sum of $\lambda = \pm$
– V_A is $\lambda = A$ (S) ← **needed for gauge inv.**
– V_X is V_{SM} ← **redundancy for checks**

Polarization@1-loop not (yet) supported in standalone `mg5amc`

– reweighting for loop+NWA is computationally expensive

– polarization in tree subgraphs vs polarization in loop subgraphs

– reference frame choice ✓

Use support for BSM@1-loop:

– `SM_Loop_ZPolar`

– `SM_Loop_WPolar`

– `SM_Loop_VPolar`

feynrules.irmp.ucl.ac.be/wiki/VPolarization

Split $Z, W^{\pm}$ in into four states:

– V_0 is $\lambda = 0$

– V_T is sum of $\lambda = \pm$

– V_A is $\lambda = A$ (S) ← **needed for gauge inv.**

– V_X is V_{SM} ← **redundancy for checks**

`mg5amc` installation:

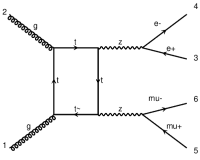
```
$ cd <MGPATH>/models
$ wget <URL>/SM_Loop_ZPolar.tgz
$ tar -zxvf SM_Loop_ZPolar.tgz
$ cd ..
$ ./bin/mg5_aMC
```

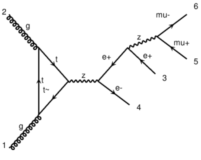
From Feynman rules to cross sections (and events!)

Given a UFO library, VPolar+mg5amc runs out of the box

see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set auto_convert_model True
> set group_subprocesses False
> import model SM_Loop_ZPolar
```



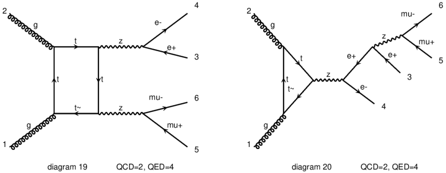


From Feynman rules to cross sections (and events!)

Given a UFO library, `VPolar+mg5amc` runs out of the box

see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set auto_convert_model True
> set group_subprocesses False
> import model SM_Loop_ZPolar
```



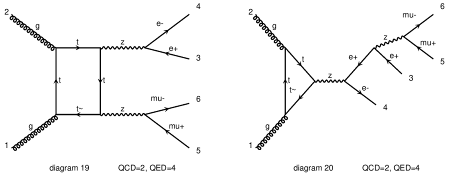

```
## unpolarized w/ everything
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD] / a z0 za zt
> output COMETA_gg_2el2mu_noPol_allInt
```

From Feynman rules to cross sections (and events!)

Given a UFO library, `VPolar+mg5amc` runs out of the box

see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ

```
$ ./bin/mg5_aMC
> set acknowledged_v3.1_syntax True
> set auto_convert_model True
> set group_subprocesses False
> import model SM_Loop_ZPolar
```



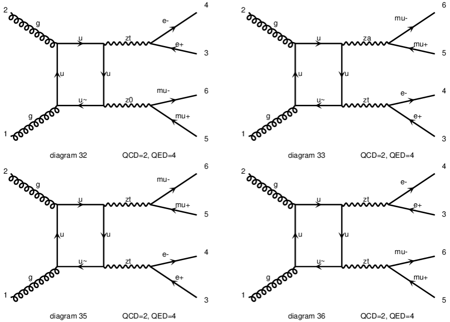

```
## unpolarized w/ everything
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD] / a z0 za zt
> output COMETA_gg_2el2mu_noPol_allInt

## unpolarized w/o s-channel mu/e
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD]
                                        / a z0 za zt e+ mu+
> output COMETA_gg_2el2mu_noPol_noElMu
```

Given a UFO library, VPolar+mg5amc runs out of the box

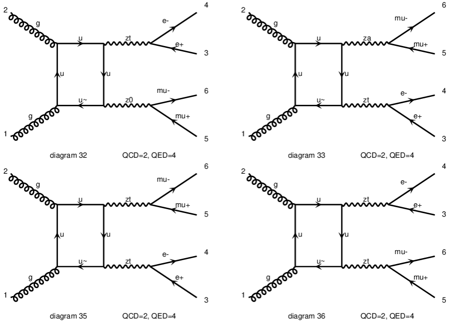
see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ

```
## unpolarized w/ everything
> generate g g > e+ e- mu+ mu-
    QED=4 QCD=2 [noborn = QCD] / a z
> output COMETA_gg_2el2mu_everything
```



Given a UFO library, VPolar+mg5amc runs out of the box

see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ




```
## unpolarized w/ everything
> generate g g > e+ e- mu+ mu-
    QED=4 QCD=2 [noborn = QCD] / a z
> output COMETA_gg_2el2mu_everything
```

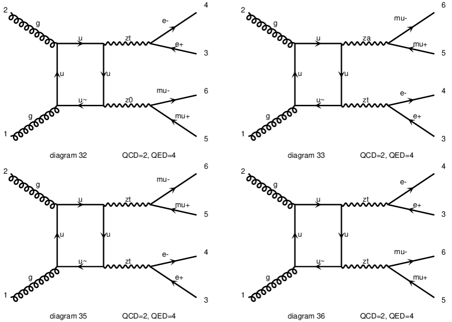
$Z_0 Z_0$ and s-channel mu/e

```
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD] / a z za zt
> output COMETA_gg_2el2mu_Z0Z0_allInt
```

Given a UFO library, VPolar+mg5amc runs out of the box

see Javurkova, et al [2401.17365] or gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ

```
## unpolarized w/ everything
> generate g g > e+ e- mu+ mu-
    QED=4 QCD=2 [noborn = QCD] / a z
> output COMETA_gg_2el2mu_everything
```



$Z_0 Z_0$ and s-channel mu/e

```
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD] / a z za zt
> output COMETA_gg_2el2mu_Z0Z0_allInt
```

$Z_T Z_T$ and s-channel mu/e

```
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD] / a z za z0
> output COMETA_gg_2el2mu_ZTZT_allInt
```

mixed polarization case:

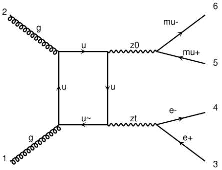


diagram 7 QCD=2, QED=4

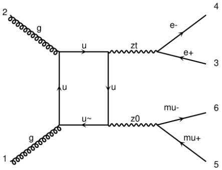


diagram 8 QCD=2, QED=4

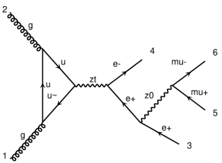


diagram 9 QCD=2, QED=4

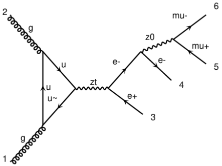
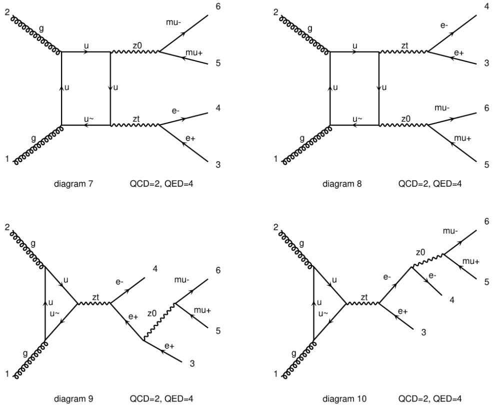


diagram 10 QCD=2, QED=4

mixed polarization case:




```
# 1. find ID of tree containing a vertex with a "bad" leg
#   Note: a "tree in structs" loop runs over all tree subgraphs
#   -- a "tree" here is any tree-level subgraph attached to a loop
#   -- structure contains all "tree" diagrams (non-loops)
#   -- a tree contains one or more vertices
#   -- a vertex contains one or more "legs"
#   -- a leg has a PID
#
#   case 1. single tree with two ZX
#   case 2. two trees, each with one ZX


for tree in structs:
    isTreeAlreadyVetoed=False

    # does the tree have only one vertex? (box/pentagon/etc diagram!)
    #   -- check if vertex has a ``bad'' leg (veto)
    #   -- check if vertex has a ``good'' leg (tag)
    if len(tree.get('vertices')) == 1:
        vertex = tree.get('vertices')[0]
        for leg in vertex.get('legs'):
            if doFilterDebug: print("new leg")
            if (leg.get('id') in redList) or \
               (leg.get('id') in antiTagList):
                # automatically reject trees with red flags
                vetoTreeList.append(tree.get('id'))
                # no need to explore vertex any further
                break # out of loop

            elif (leg.get('id') in pol1TagList):
                if(doSplit0TPol):
                    for leg2 in vertex.get('legs'):
                        if doFilterDebug: print(f"{leg2.get('id')=}")
                        if abs(leg2.get('id')) in lep2TagList:
                            if doFilterDebug: print('vetoing tree')
                            vetoTreeList.append(tree.get('id'))
                            break
                # tag the simple tree as ``good''
                pol1TreeList.append(tree.get('id'))
```

$Z_0 Z_T + Z_T Z_0$ and s-channel mu/e

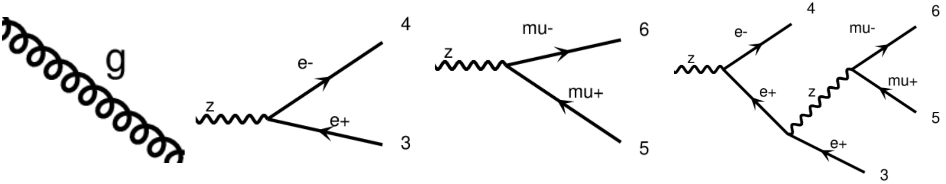
```
> generate g g > e+ e- mu+ mu- QED=4 QCD=2 [noborn = QCD]
                                          / a - -loop_filter=True

> output COMETA_gg_2el2mu_ZOZT_allInt
```

gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ/MadGraph/LoopFilter

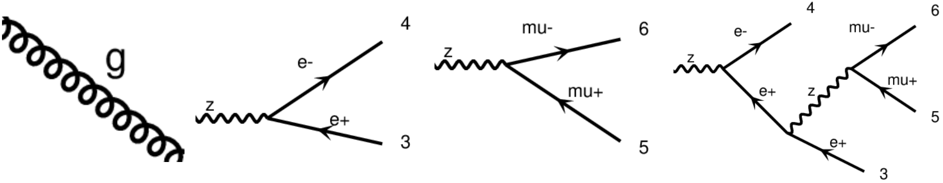
logic of MadLoop (1 slide)

step 1: up to given order, build **tree subgraphs**

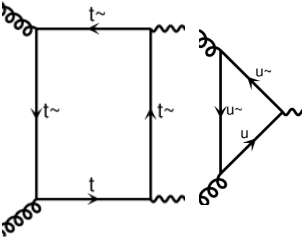


logic of MadLoop (1 slide)

step 1: up to given order, build **tree subgraphs**

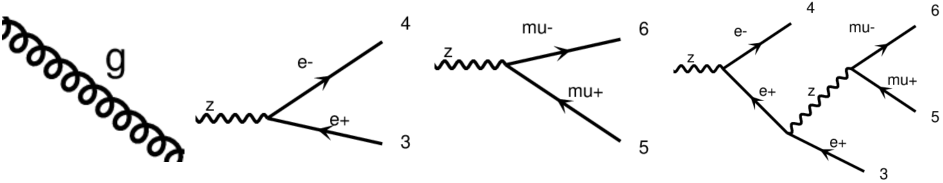


step 2: up to same order, build **loop subgraphs**

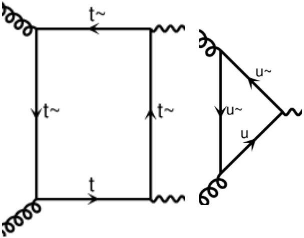


logic of MadLoop (1 slide)

step 1: up to given order, build **tree subgraphs**



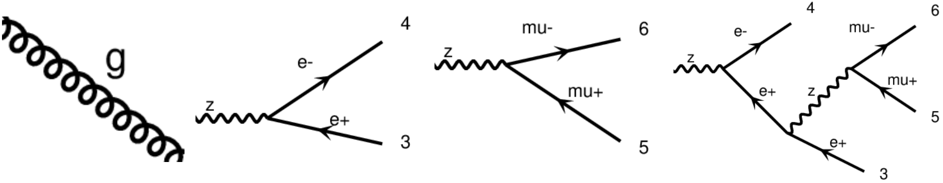
step 2: up to same order, build **loop subgraphs**



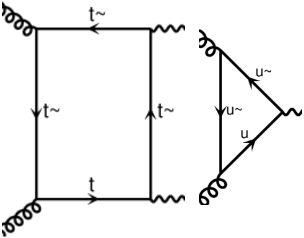
step 3: for $i \in \{\text{tree}\}$ and $j \in \{\text{loop}\}$, **build the diagram** $(i^k j)$

logic of MadLoop (1 slide)

step 1: up to given order, build **tree subgraphs**



step 2: up to same order, build **loop subgraphs**



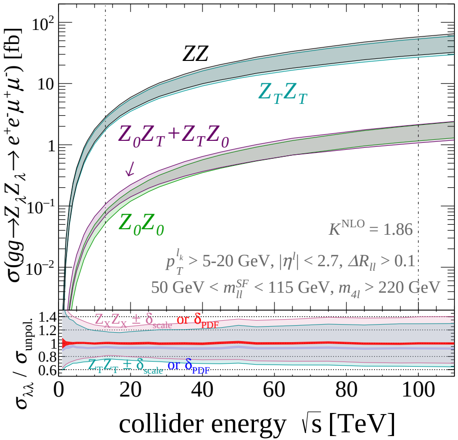
step 3: for $i \in \{\text{tree}\}$ and $j \in \{\text{loop}\}$, **build the diagram** $(i^k j)$

step 4: **filter out / reject** subset of $(i^k j)$ combinations

some numbers

```
launch COMETA_gg_2el2mu_Z0Z0_allInt
analysis=off
set pdlabel lhapdf
# NNPDF31_nlo_as_0118_luxqed
set lhaid 324900
set lhc 13
set nevents 4k
set dynamical_scale_choice 3
set me_frame [3,4,5,6]
set no_parton_cut
set mmll 50
set mmllmax 115
set mmnl 220
set etal 2.7
set drll 0.1
set ptl1min 20
set ptl2min 15
set ptl3min 10
set ptl4min 5
set use_syst true
done
```

Javurkova, RR et al (PLB'24) [2401.17365]



scripts available here: gitlab.cern.ch/riruiz/public-projects/-/tree/master/VPolar_ggZZ/MadGraph/RunScripts

final words

senior postdoc vacancy in Krakow

3-year Adv/Senior Postdoctoral Researcher in Theoretical Particle Physics

Cracow, INP • Europe

hep-ph hep-th nucl-th PostDoc

Deadline on Nov 15, 2024

Job description:
Job Title: Adv/Senior Postdoctoral Researcher

The Department of Theoretical Particle Physics (NZ42) at the **Institute of Nuclear Physics – Polish Academy of Sciences** (IFJ PAN) in Krakow, Pola
postdoctoral appointment ("adjunct" in Polish) in the group of Prof. Richard Ruiz.

inspirehep.net/jobs/2829053

thank you!

backup

Decomposing Propagators

Completeness relationships between **propagators** & **polarization vectors** in gauge theories are subtle. Example: **QED** in Feynman gauge

$\implies \xi = 1$ so $(1-\xi)q_\mu q_\nu/q^2 \to 0$:

$$-g_{\mu\nu} = \begin{pmatrix} -1 & & & \\ & +1 & & \\ & & +1 & \\ & & & +1 \end{pmatrix} = \sum_{\lambda=\pm,0,S} \varepsilon_\mu(q,\lambda)\varepsilon^*_\nu(q,\lambda)$$

For $q = (q^0, 0, 0, q^3)$ and **transverse** pols $\varepsilon_\mu(\lambda = \pm) = (0, \mp 1, -i, 0)/\sqrt{2}$

$$\sum_{\lambda=\pm} \varepsilon_\mu(q,\lambda)\varepsilon^*_\nu(q,\lambda) = \begin{pmatrix} 0 & & & \\ & +1 & 0 & \\ & 0 & +1 & \\ & & & 0 \end{pmatrix}$$

Decomposing Propagators

For $q = (q^0, 0, 0, q^3)$ and **longitudinal** $\varepsilon_\mu(\lambda = 0) = (q^3, 0, 0, q^0)/\sqrt{q^2}$

$$\sum_{\lambda=0} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = \frac{q^2}{q^2}\begin{pmatrix} -1 & & & \\ & 0 & & \\ & & 0 & \\ & & & +1 \end{pmatrix} + \frac{q_\mu q_\nu}{q^2}$$

For "**auxiliary**" (A) or "**scalar**" (S) polarization $\varepsilon_\mu(\lambda = S) = q_\mu/\sqrt{-q^2}$

$$\sum_{\lambda=S} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = -\frac{q_\mu q_\nu}{q^2}$$

Precise form for $\lambda = 0, S$ depends on several factors:

– broken (massive) or unbroken (massless) gauge symmetry

– gauge (Feynman vs Landau vs Unitary vs Axial)

– gauge fixing ($\xi = 1$ or $n^2 = -1$)

Decomposing Propagators

For $q = (q^0, 0, 0, q^3)$ and **longitudinal** $\varepsilon_\mu(\lambda = 0) = (q^3, 0, 0, q^0)/\sqrt{q^2}$

$$\sum_{\lambda=0} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = \frac{q^2}{q^2}\begin{pmatrix} -1 & & & \\ & 0 & & \\ & & 0 & \\ & & & +1 \end{pmatrix} + \frac{q_\mu q_\nu}{q^2}$$

For "**auxiliary**" (A) or "**scalar**" (S) polarization $\varepsilon_\mu(\lambda = S) = q_\mu/\sqrt{-q^2}$

$$\sum_{\lambda=S} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = -\frac{q_\mu q_\nu}{q^2}$$

Example: for W/Z in Unitary gauge, $\varepsilon_\mu^{W/Z}(\lambda = S) = q_\mu\sqrt{\frac{1}{M_V^2} - \frac{1}{q^2}}$

$$\sum_{\lambda=S} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = -\frac{q_\mu q_\nu}{q^2} + \frac{q_\mu q_\nu}{M_V^2}$$

Decomposing Propagators

For $q = (q^0, 0, 0, q^3)$ and **longitudinal** $\varepsilon_\mu(\lambda = 0) = (q^3, 0, 0, q^0)/\sqrt{q^2}$

$$\sum_{\lambda=0} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = \frac{q^2}{q^2}\begin{pmatrix} -1 & & & \\ & 0 & & \\ & & 0 & \\ & & & +1 \end{pmatrix} + \frac{q_\mu q_\nu}{q^2}$$

For "**auxiliary**" (A) or "**scalar**" (S) polarization $\varepsilon_\mu(\lambda = S) = q_\mu/\sqrt{-q^2}$

$$\sum_{\lambda=S} \varepsilon_\mu(q,\lambda)\varepsilon_\nu(q,\lambda) = -\frac{q_\mu q_\nu}{q^2}$$

Bonus, longitudinal polarization vectors can be written as Dawson ('85)

$$\varepsilon_\mu(\lambda = 0) = \frac{q_\mu}{\sqrt{q^2}} + \mathcal{O}\left(\frac{\sqrt{q^2}}{q^0}\right) \leftarrow \text{not an approximation}$$