Generator (MG) + Hands-on

Qianying Guo, Meng Lu

19 Dec 2021

Part1

Why we need generator

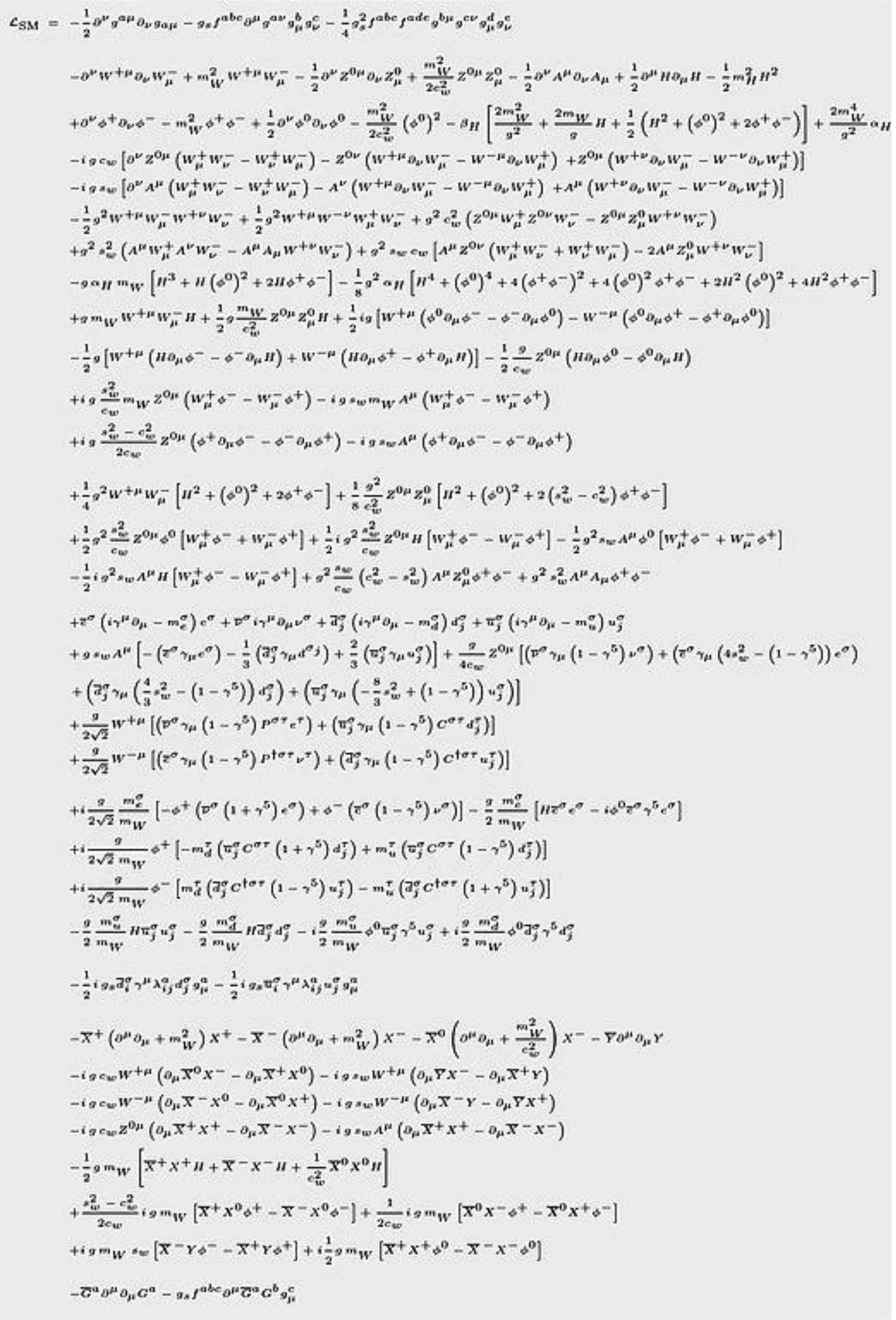
Why do we need generator?

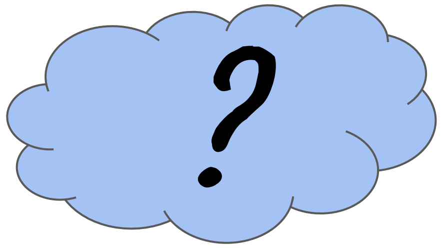
Physics : a subject based on experiment -> what we learned from middle school.

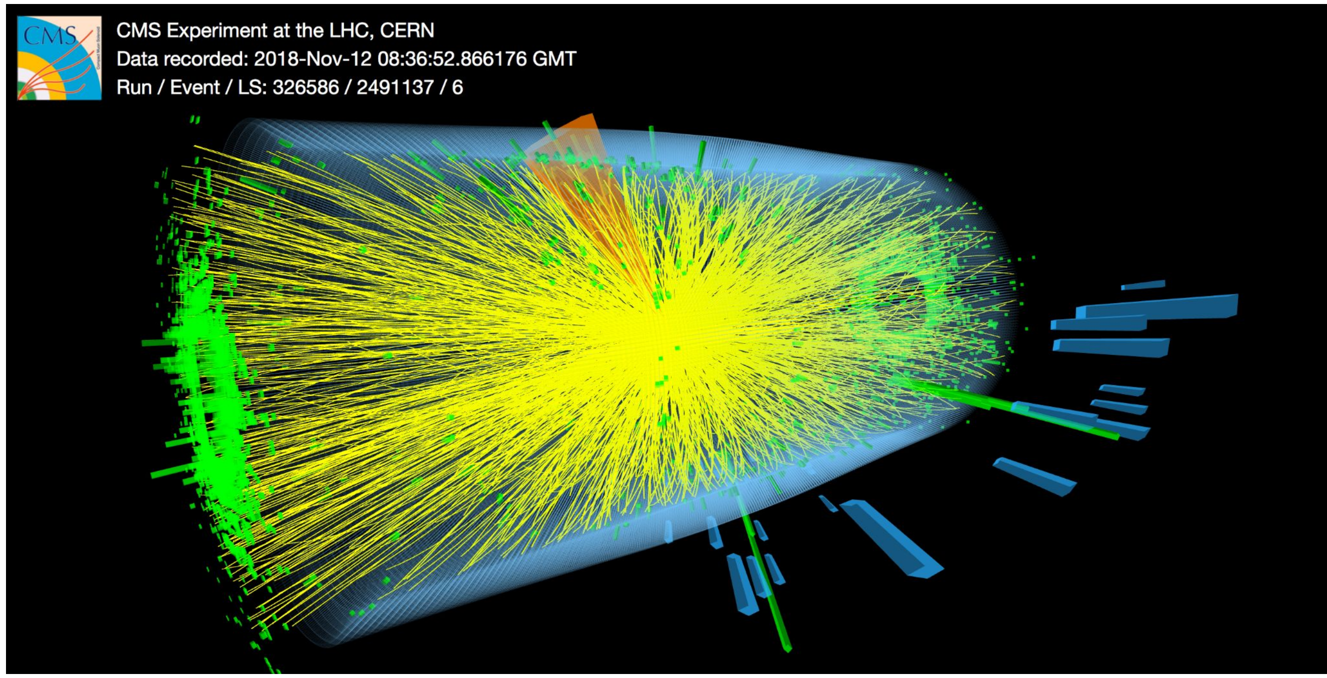
For experimental high energy physics, there are lots of machines providing different data for us, e.g., LHC, BEPC ...,
-> ok, we collider electron-positron, proton-proton, we have different experiments

For theoretical particle physics, we have SM (EW, QCD)

Why we need generator

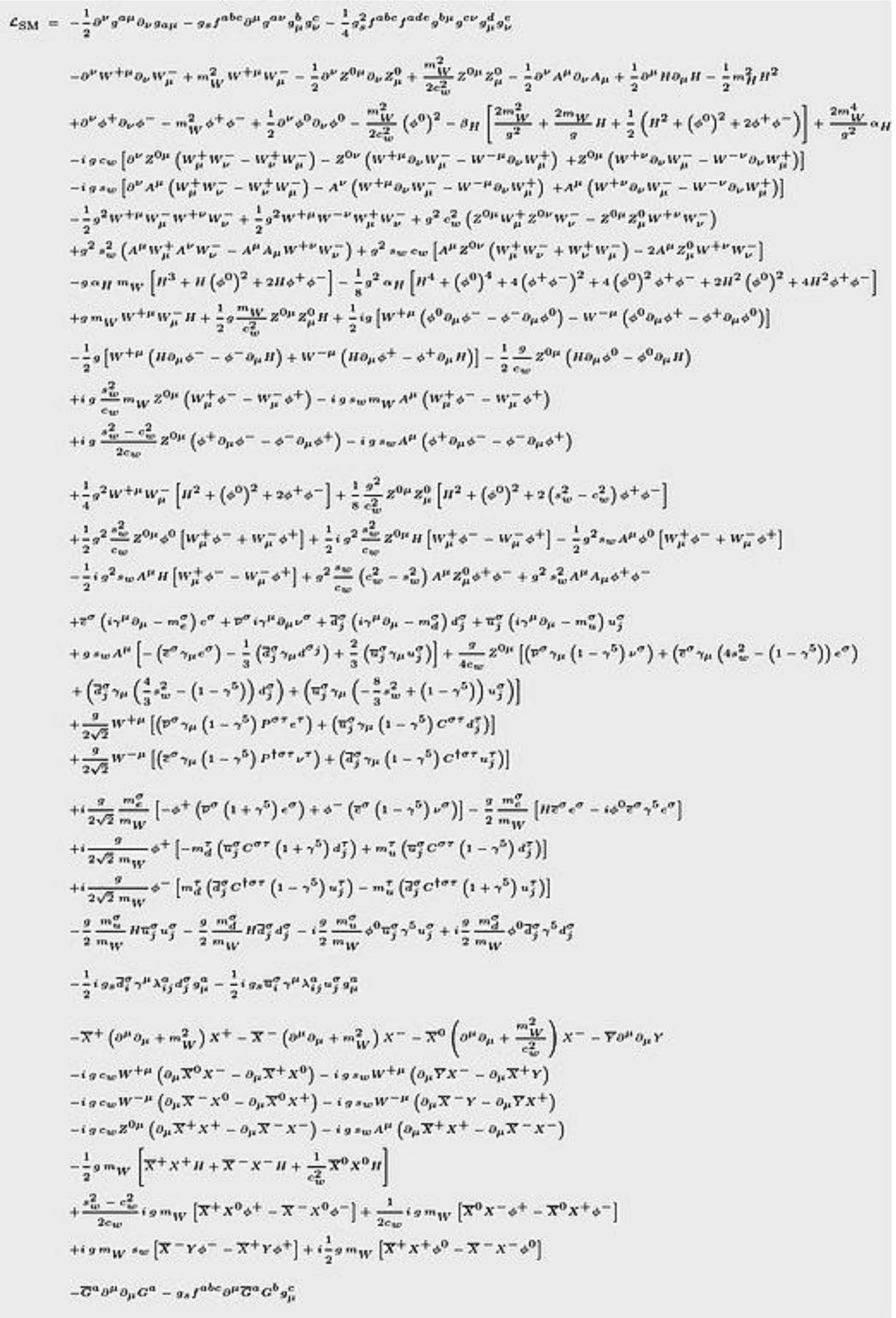


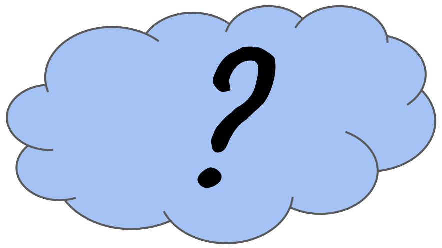


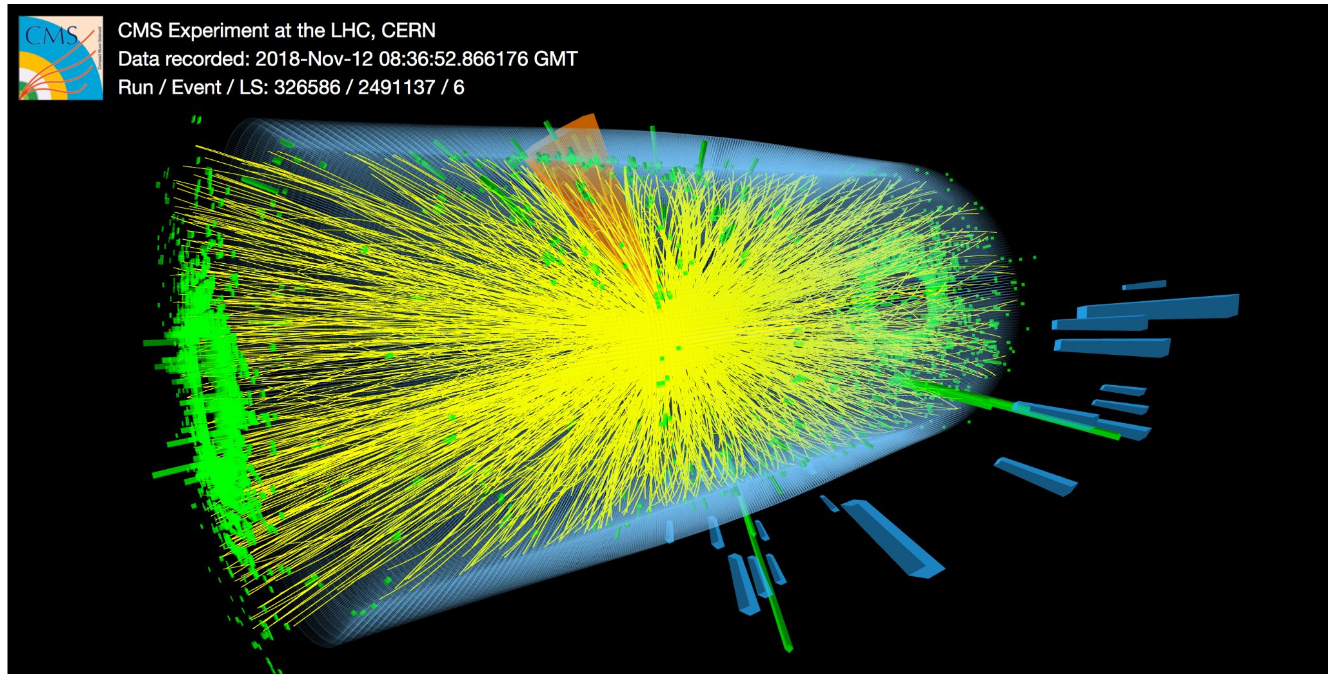


How to perform measurements, i.e., how do we compare the theory to experiment, or vice versa

Why we need generator

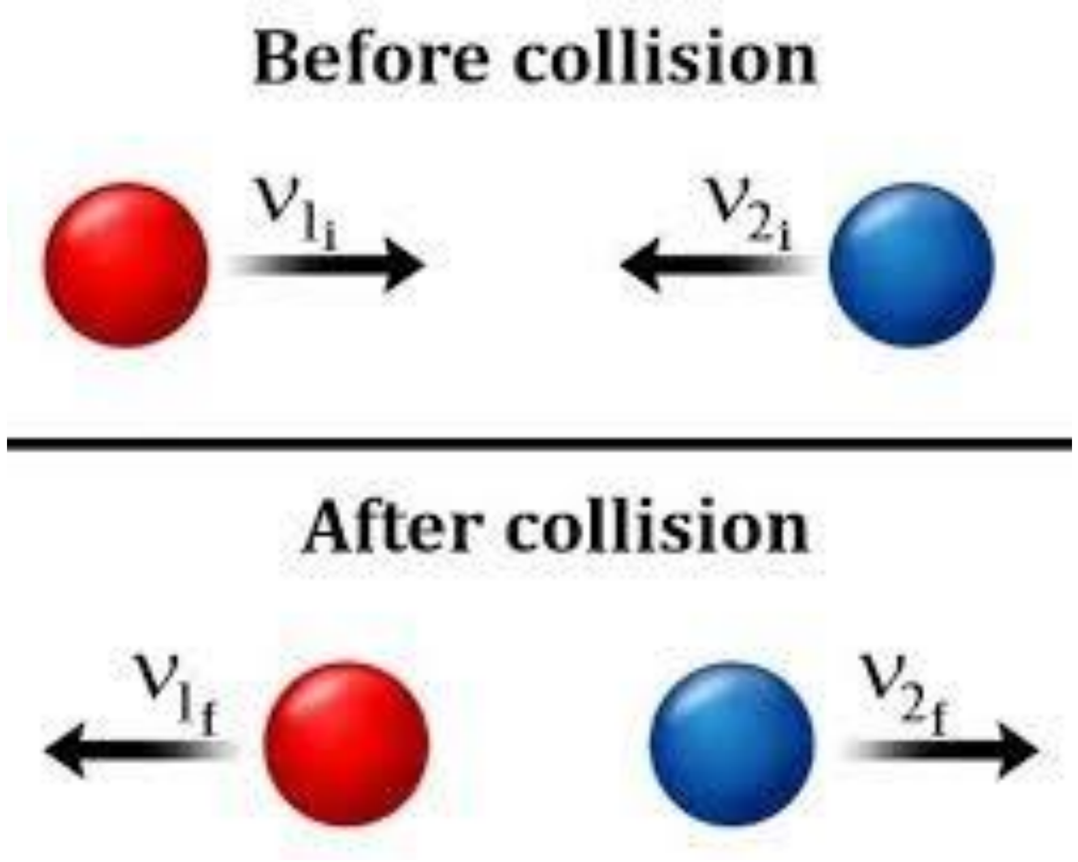






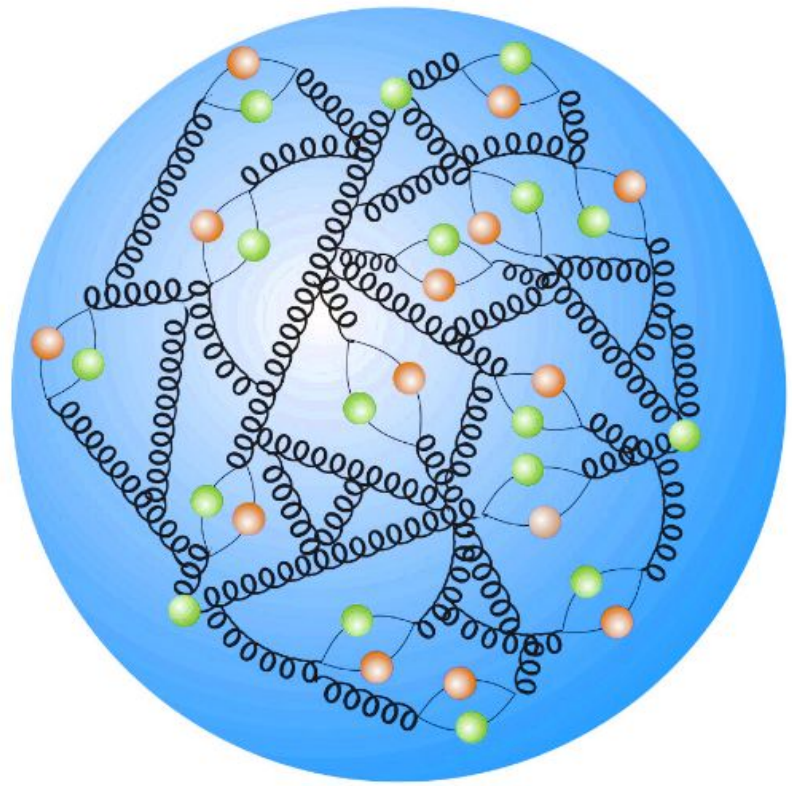
Generator is the bridge between theory and experiment, the generator is based on the theory, and generate events through the Monte Carlo (MC) method.
Then we can compare the MC events with data, i.e., the measurement.

Hadron collision



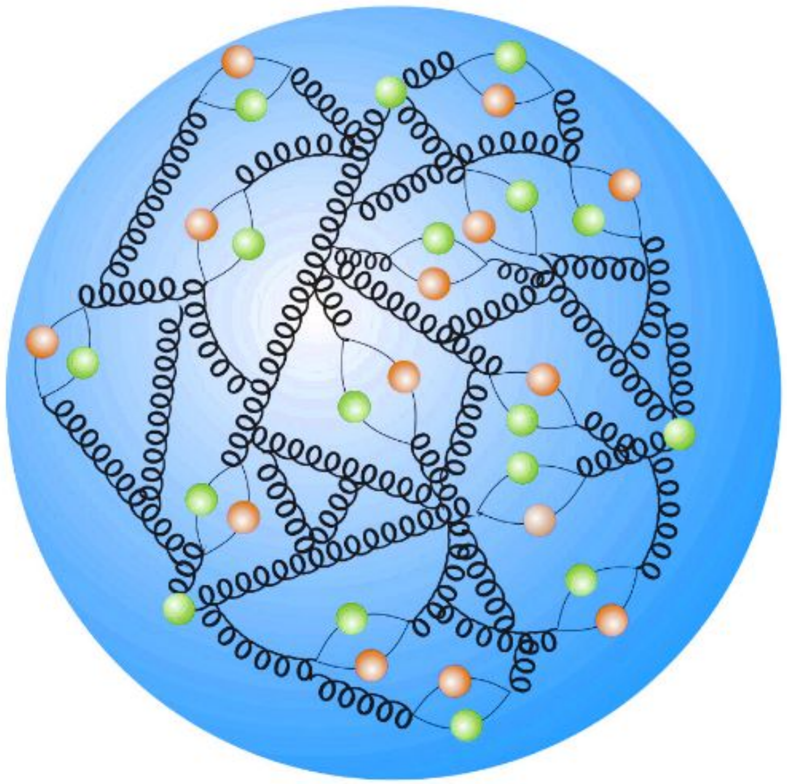
What we learned about the collision during high school. Very clear and beautiful idea, no object is destroyed or created. So far so good!

But if we step into the particle physics, we learn about the concept "elementary particle", and collision becomes complicated due to the disappear and creation of particles!

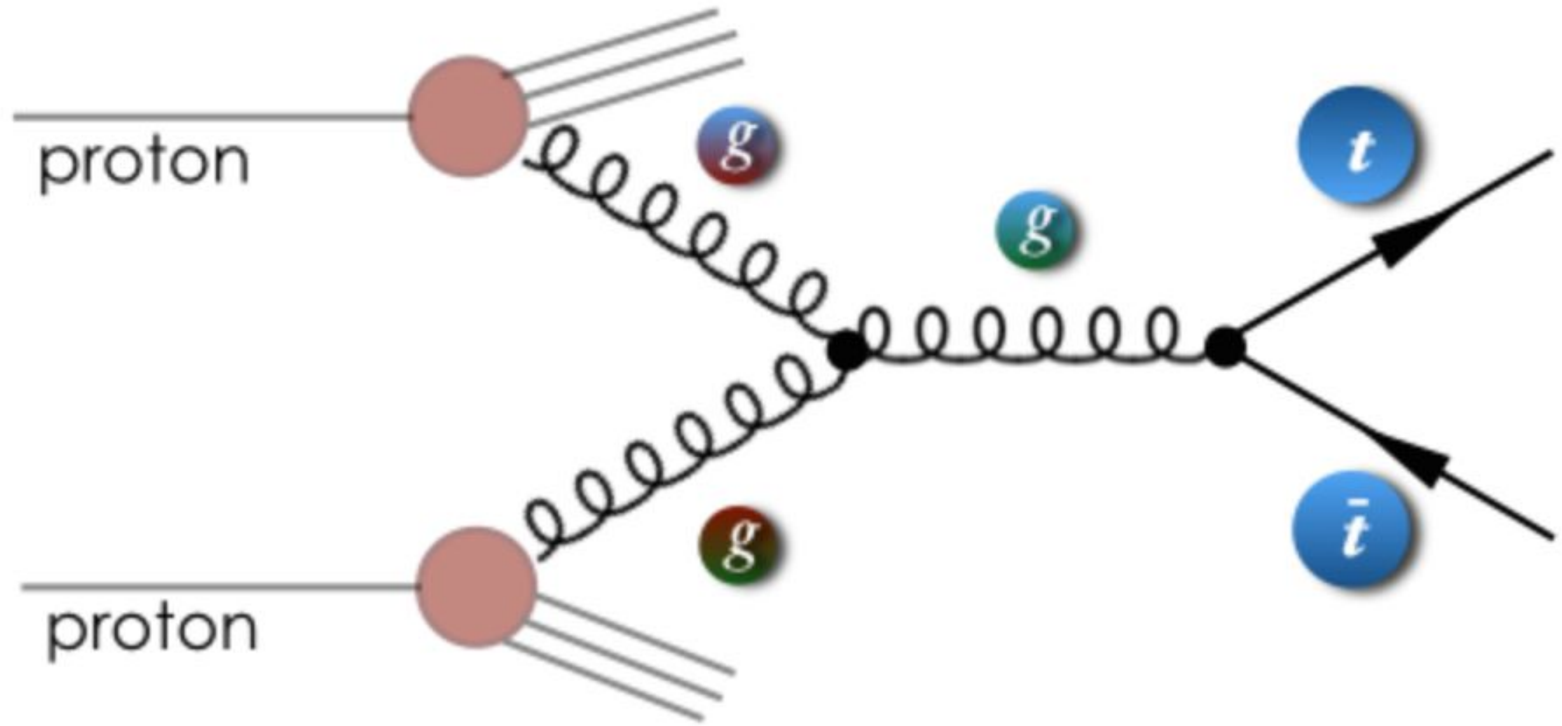


Proton has inner structure, there are all kinds of quarks and gluons inside!

Hadron collision



Proton rest mass: 0.938 GeV (1.67*10^-27 kg)
Due to the QCD Asymptotic Freedom (2004 Nobel Prize), the quarks and gluons are confined to be a composite proton. When the energy become larger, the confinement is weaker and the components of proton are able to "escape" from the proton and interact with each other.

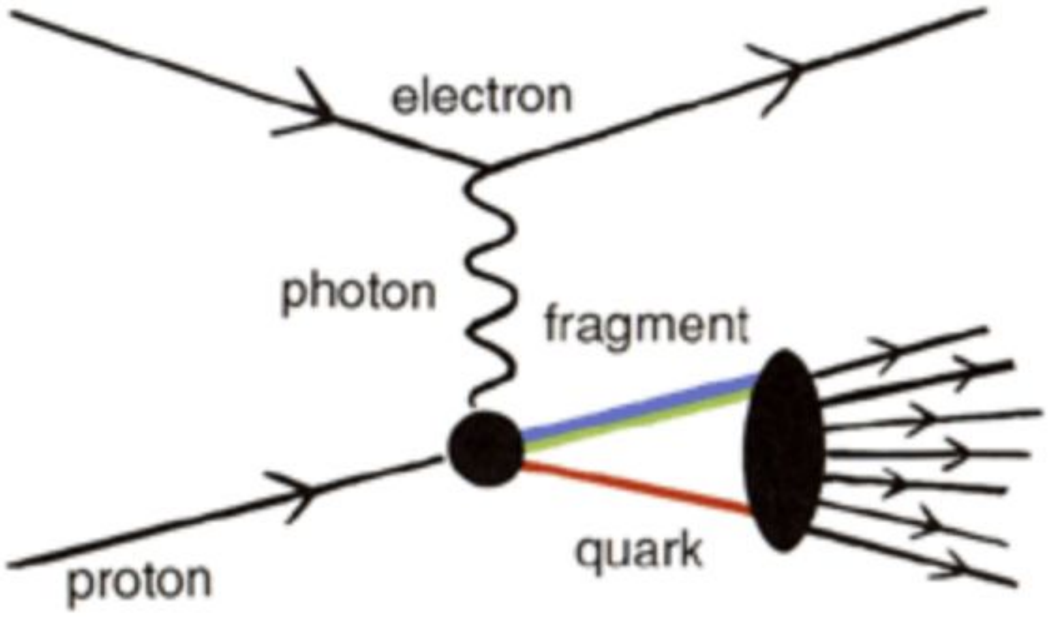


In the case of proton-proton collision (the LHC) at very high energy, the inner components (say gluon) escape from the incoming proton, carrying a fraction energy, and then interact with another gluon from the other proton.

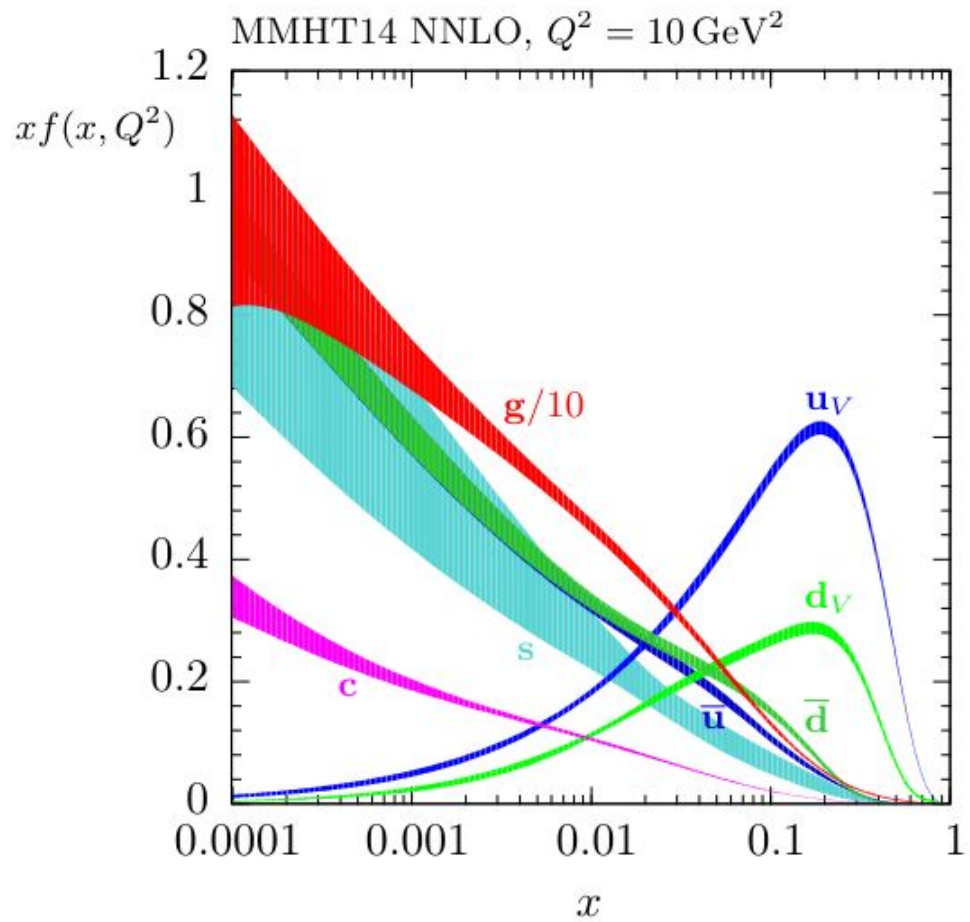
Q: how do we know how much the energy is carried by the escaping gluon?

N.B.: for lepton collider, e.g., the electron-positron and muon collider, no such scheme due to he leptons are elementary particles and they interact with the energy they have

Hadron collision



The deep inelastic collision ([1990 Nobel Prize]) is used to extract the inner structure of proton.



Parton density functions:

X-axis: the momentum carried by the escaped partons

Y-axis: distribution function

[LHAPDF]

$$\int \hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)\, dx_1 dx_2 d\Phi_{FS}$$

Parton level cross section | Parton density functions | Phase space integral

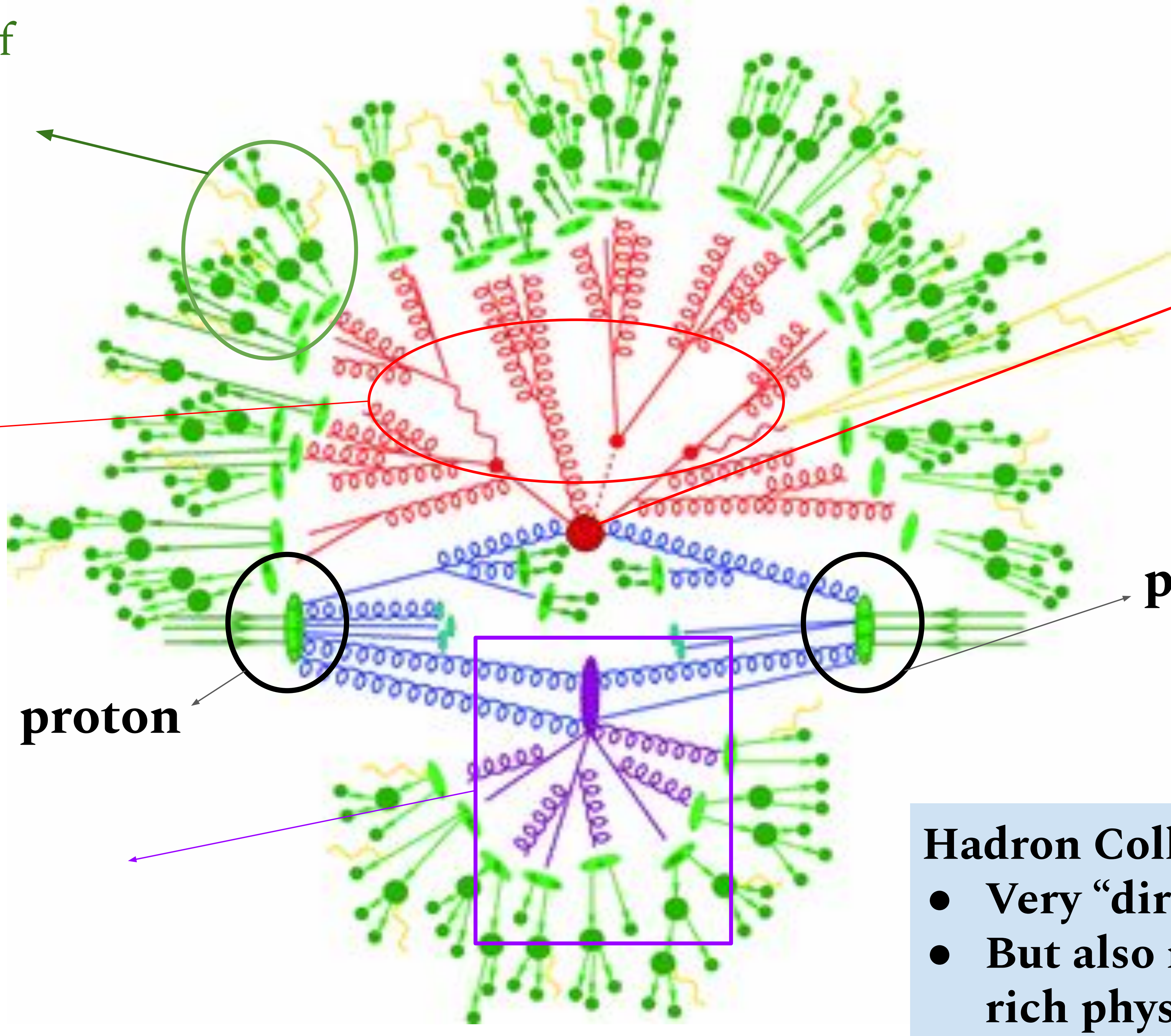
Now with the PDF, it's able to predict the cross section of individual process in proton-proton collision by calculation (Monte Carlo method).

Hadron collision

Hadronization of parton shower particles and further decay

Hard process, this is what people usually interested in.

Parton shower: the evolution of the particles from Hard process



proton

proton

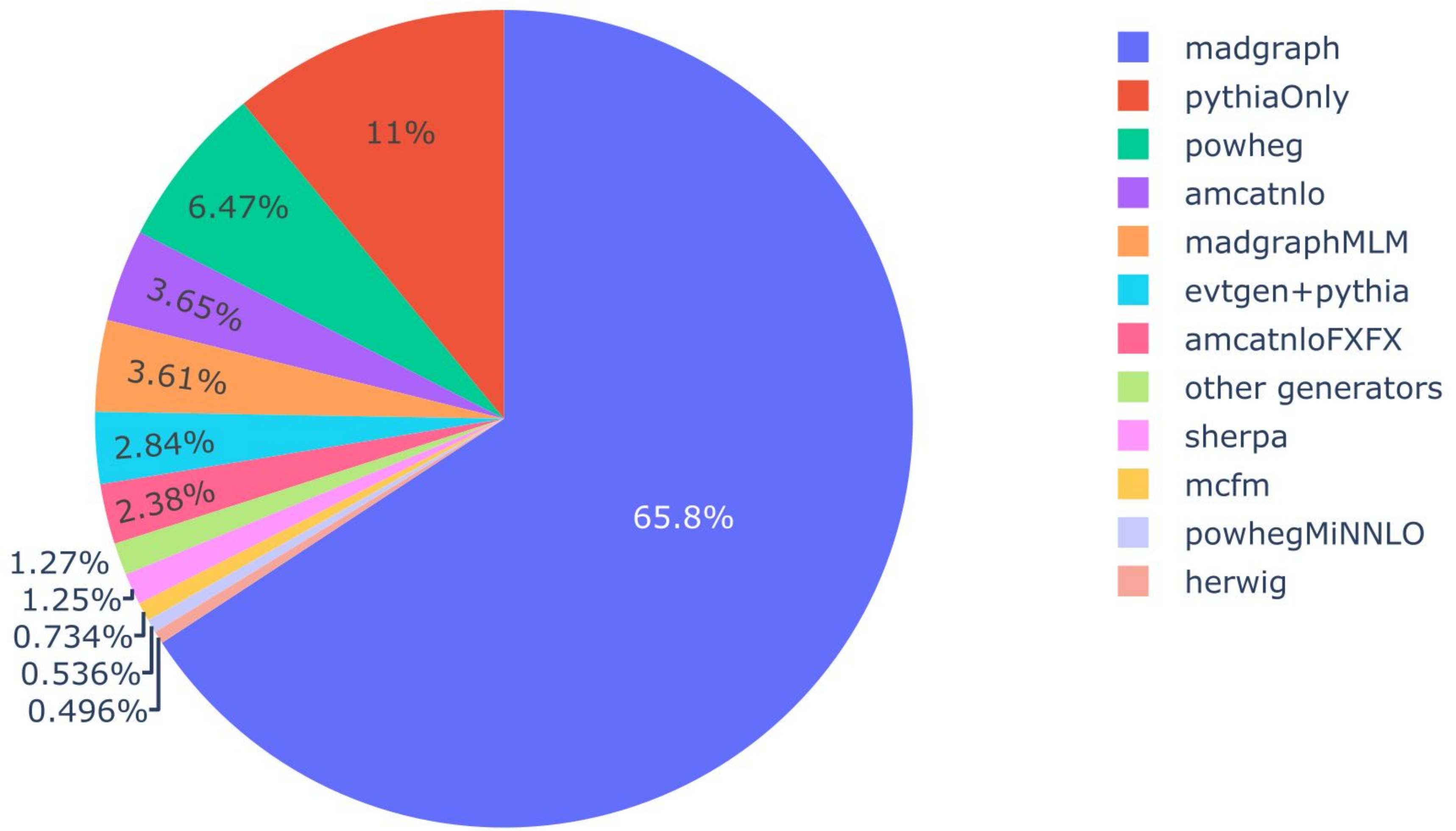
Double parton scattering

Hadron Collider:
- **Very “dirty” environment**
- **But also means extremely rich physics**

MadGraph5_aMC@NLO (MG) is the most commonly used generator within CMS.

Advantages:

- Able to handle both LO and NLO level accuracy
- Easy to use, hard scattering process
- Flexible to interface to other generator for parton showering
- Flexible to include new physics model
- Provide many useful things additionally, e.g., you can check the Feynman diagrams
- Update frequently with new features implemented
- ...



By #samples

MG install

The MG is kind of a “standalone” software, you download it and extract, and then it’s ready to use… (in case you haven’t downloaded it)

Log in / Register

MadGraph5_aMC@NLO

Overview | Code | Bugs | Blueprints | Translations | Answers

Click on the All downloads and it will list all the versions, here we use MG5_aMC_v2.7.3.tar.gz

Registered 2009-09-15 by Michel Herquet

MadGraph5_aMC@NLO is a framework that aims at providing all the elements necessary for SM and BSM phenomenology, such as the computations of cross sections, the generation of hard events and their matching with event generators, and the use of a variety of tools relevant to event manipulation and analysis. Processes can be simulated to LO accuracy for any user-defined Lagrangian, an the NLO accuracy in the case of models that support this kind of calculations -- prominent among these are QCD and EW corrections to SM processes. Matrix elements at the tree- and one-loop-level can also be obtained.

MadGraph5_aMC@NLO is the new version of both MadGraph5 and aMC@NLO that unifies the LO and NLO lines of development of automated tools within the MadGraph family. It therefore supersedes all the MadGraph5 1.5.x versions and all the beta versions of aMC@NLO. As such, the code allows one to simulate processes in virtually all configurations of interest, in particular for hadronic and e+e- colliders; starting from version 3.2.0, the latter include Initial State Radiation and beamstrahlung effects.

The standard reference for the use of the code is: J. Alwall et al, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations", arXiv:1405.0301 [hep-ph]. In addition to that, computations in mixed-coupling expansions and/or of NLO corrections in theories other than QCD (eg NLO EW) require the citation of: R. Frederix et al, "The automation of next-to-leading order electroweak calculations", arXiv:1804.10017 [hep-ph]. A more complete list of references can be found here: http://amcatnlo.web.cern.ch/amcatnlo/list_refs.htm

Get Involved

- Report a bug
- Ask a question
- Register a blueprint
- Help translate

Downloads

Latest version is 3.3.x

MG5_aMC_v2.9.6.tar.gz

MG5_aMC_v3.3.0.tar.gz

released on 2021-11-12

All downloads

MG install

In the cluster (or in your laptop if you are using linux):

⏭ wget https://launchpadlibrarian.net/485276105/MG5_aMC_v2.7.3.tar.gz

⏭ tar zxf MG5_aMC_v2.7.3.tar.gz

```
[lum@farm ~]$ ls MG5_aMC_v2_7_3/
aloha   doc      HELAS   INSTALL   madgraph   mg5decay   PLUGIN          README     tests             vendor
bin     doc.tgz  input   LICENSE   MadSpin    models     proc_card.dat   Template   UpdateNotes.txt   VERSION
```

```
[lum@farm ~]$ ls MG5_aMC_v2_7_3/bin/
mg5   mg5_aMC
```

We will use “mg5_aMC”

From this slide on, the input command will be start with bullets ⏭

MG install

- cd MG5_aMC_v2_7_3/bin/
- ./mg5_aMC
- “Ctrl + d”

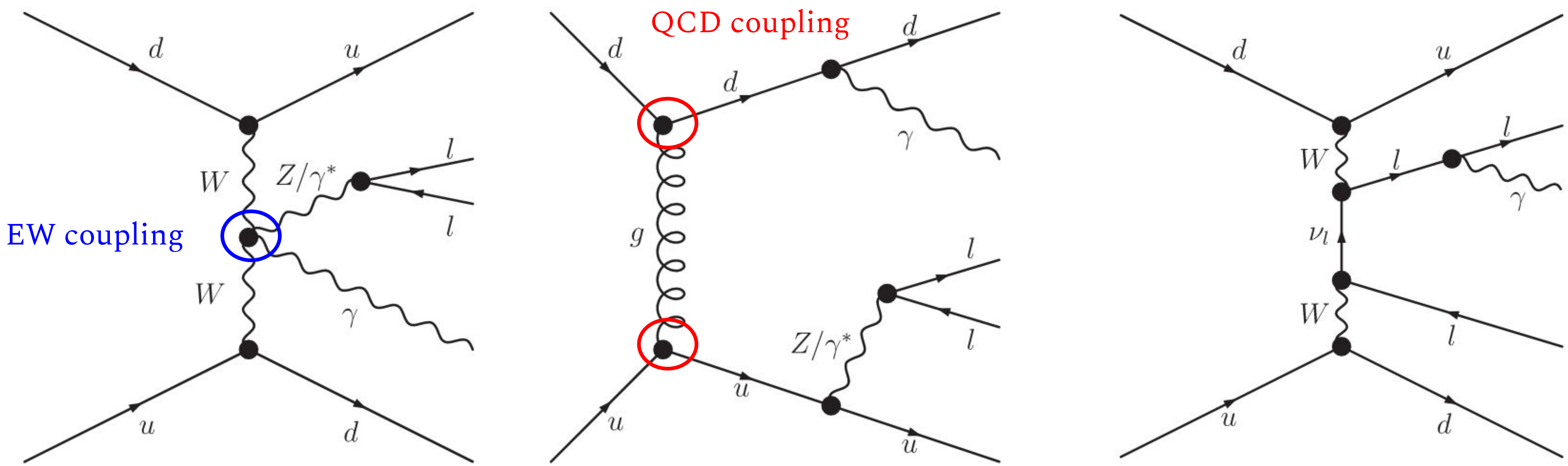
There will be loading messages, (you can ignore the updating warning)

```
Loading default model: sm
INFO: load particles
INFO: load vertices
INFO: Restrict model sm with file ../models/sm/restrict_default.dat .
INFO: Run "set stdout_level DEBUG" before import for more information.
INFO: Change particles name to pass to MG5 convention
Defined multiparticle p = g u c d s u~ c~ d~ s~
Defined multiparticle j = g u c d s u~ c~ d~ s~
Defined multiparticle l+ = e+ mu+
Defined multiparticle l- = e- mu-
Defined multiparticle vl = ve vm vt
Defined multiparticle vl~ = ve~ vm~ vt~
Defined multiparticle all = g u c d s u~ c~ d~ s~ a ve vm vt e- mu- ve~ vm~ vt~ e+ mu+ t b t~ b~ z w+ h w-
 ta- ta+
```

These lines tell you which model is loaded by default (sm), and in this model, how the multiparticles are defined.

EW and QCD $Z\gamma$

Before moving to the production, let's have a brief feeling on the process we are going to produce. The following analysis is kind of reproducing this work, the VBS $Z\gamma$.



Final state: two leptons, two quarks (will be reconstructed as jets), and a photon.

Signal: include only Electroweak coupling, left plot and right plot, but the left plot is what we're really interested in, it's a Vector Boson Scattering (VBS) process

QCD Background: those processes with the same final states, but include QCD coupling.

EW $Z\gamma$

EW $Z\gamma$ process official production setup:

import model sm-ckm_no_b_mass → Define the model we use and the restriction, e.g., the b quark mass is set to zero here

define lep+ = e+ mu+ ta+
define lep- = e- mu- ta- → Define the lepton container lep to include the tau lepton

generate p p > lep+ lep- a j j QCD=0 → Generate the process, here "QCD=0" means the max QCD vertex is 0. So "QCD=0" is equal to "QCD==0", and "QCD=2" is euqal to "QCD<=2".
If no specific appendix is defined, it will automatically calculate the process with maximum QCD vertex (i.e., always calculate the process with max cross section)

```
MG5_aMC>import model sm-ckm_no_b_mass
INFO: Restrict model sm with file ../models/sm/restrict_ckm_no_b_mass.dat .
INFO: Run "set stdout_level DEBUG" before import for more information.
Error detected in "import model sm-ckm_no_b_mass"
write debug file MG5_debug
If you need help with this issue please contact us on https://answers.launchpad.net/mg5amcnlo
MadGraph5Error : No such file /home/Documents/MG5_aMC_v2_7_3/models/sm/restrict_ckm_no_b_mass.dat
```

If you directly "import model sm-ckm_no_b_mass", you will get exception above. This is due to the file define the restriction "sm-ckm_no_b_mass" is missing in the default MadGraph model.

- cd ../models/sm/
- cp restrict_ckm.dat restrict_ckm_no_b_mass.dat
- sed -i 's/4.7/0.0/g' restrict_ckm_no_b_mass.dat
- cd -
- ./mg5_aMC
- import model sm-ckm_no_b_mass

```
MG5_aMC>import model sm-ckm_no_b_mass
INFO: Restrict model sm with file ../models/sm/restrict_ckm_no_b_mass.dat .
INFO: Run "set stdout_level DEBUG" before import for more information.
INFO: Change particles name to pass to MG5 convention
Pass the definition of 'j' and 'p' to 5 flavour scheme.
Kept definitions of multiparticles l- / vl / l+ / vl~ unchanged
Defined multiparticle all = g u c d s b u~ c~ d~ s~ b~ a ve vm vt e- mu- ve~ vm~ vt~ e+ mu+ t t~ z w+ h w-
 ta- ta+
```

- display multiparticles

```
MG5_aMC>display multiparticles
Multiparticle labels:
all = g u c d s b u~ c~ d~ s~ b~ a ve vm vt e- mu- ve~ vm~ vt~ e+ mu+ t t~ z w+ h w- ta- ta+
l- = e- mu-
j = g u c d s u~ c~ d~ s~ b b~
vl = ve vm vt
l+ = e+ mu+
p = g u c d s u~ c~ d~ s~ b b~
vl~ = ve~ vm~ vt~
```

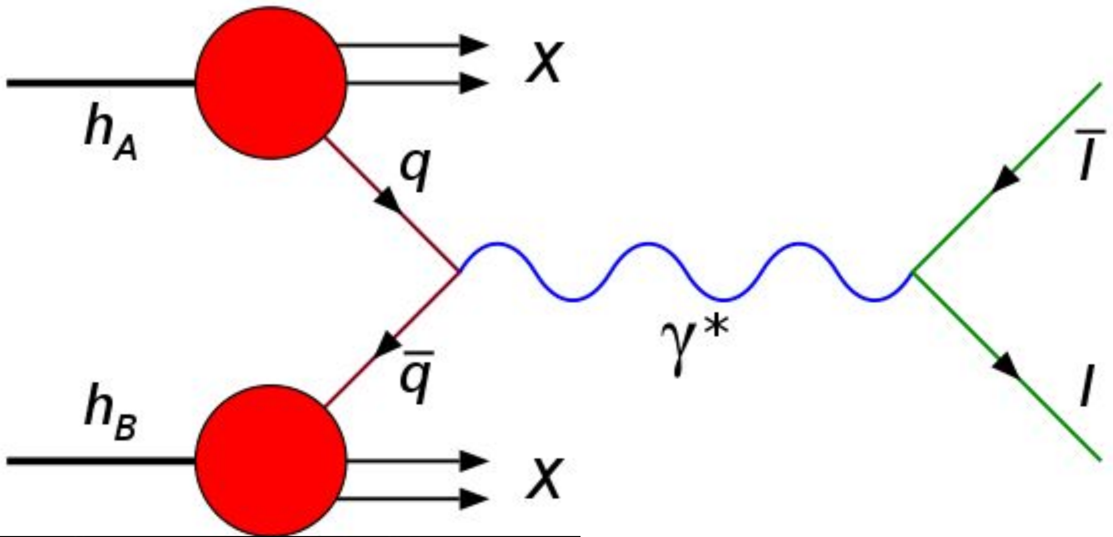
- define lep+ = e+ mu+ ta+
- define lep- = e- mu- ta-
- display multiparticles

```
MG5_aMC>display multiparticles
Multiparticle labels:
all = g u c d s b u~ c~ d~ s~ b~ a ve vm vt e- mu- ve~ vm~ vt~ e+ mu+ t t~ z w+ h w- ta- ta+
l- = e- mu-
j = g u c d s u~ c~ d~ s~ b b~
vl = ve vm vt
l+ = e+ mu+
lep- = e- mu- ta-
p = g u c d s u~ c~ d~ s~ b b~
lep+ = e+ mu+ ta+
vl~ = ve~ vm~ vt~
```

Drell-Yan process

```
INFO: Process b~ b > ta+ ta- a b b~ added to mirror process b b~ > ta+ ta- a b b~
INFO: Crossed process found for b~ b~ > e+ e- a b~ b~, reuse diagrams.
INFO: Crossed process found for b~ b~ > mu+ mu- a b~ b~, reuse diagrams.
INFO: Crossed process found for b~ b~ > ta+ ta- a b~ b~, reuse diagrams.
765 processes with 105948 diagrams generated in 120.187 s
Total: 765 processes with 105948 diagrams
```

The EW Zγ process is complicated and we don't have enough time to run it, thus we switch to Drell-Yan process out of pedagogical implication.



⏭ generate p p > lep+ lep-

```
15 processes with 30 diagrams generated in 1.309 s
Total: 15 processes with 30 diagrams
```

⏭ output DY

```
Output to directory /data/pku/home/lum/MG5_aMC_v2_7_3/bin/DY done.
Type "launch" to generate events from this process, or see
/data/pku/home/lum/MG5_aMC_v2_7_3/bin/DY/README
Run "open index.html" to see more information about this process.
```

Drell-Yan process

- "Ctrl + d"

Now in the "bin" path, you will have repository "DY"

- ls DY

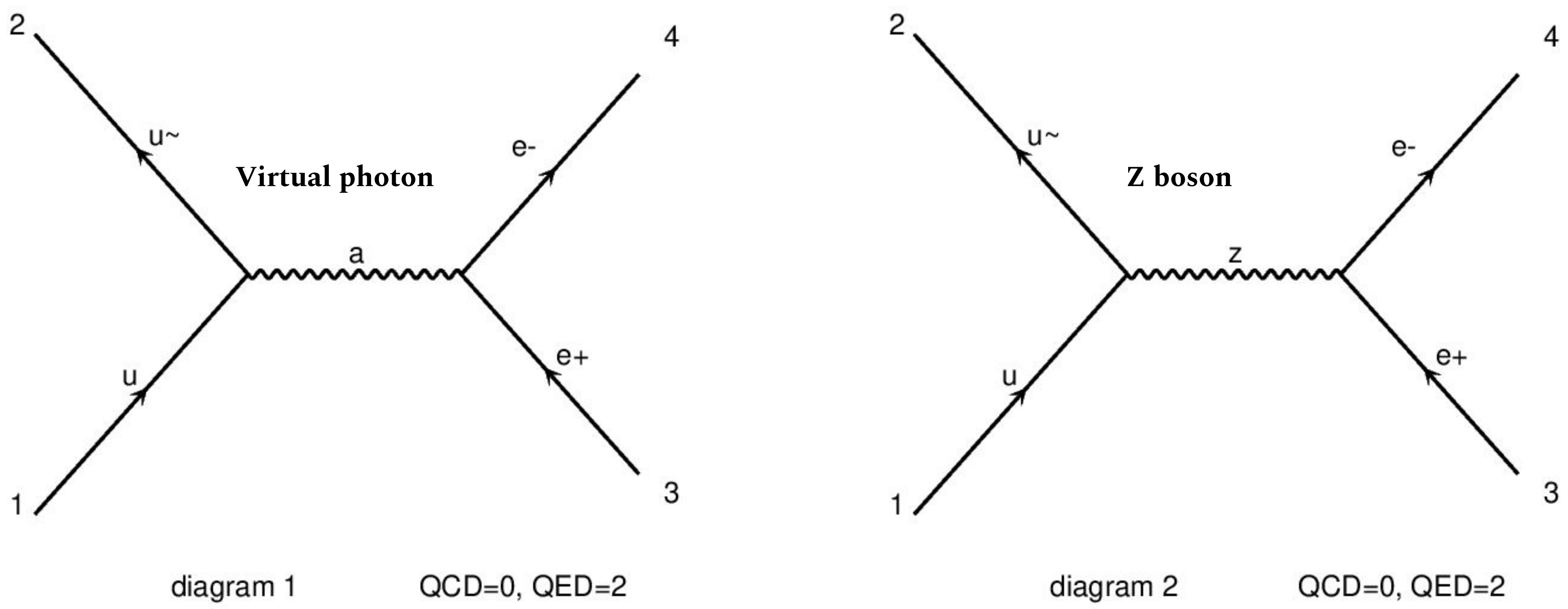
```
bin    Events  index.html  madevent.tar.gz   README                Source        TemplateVersion.txt
Cards  HTML    lib         MGMEVersion.txt   README.systematics    SubProcesses
```

- ls DY/SubProcesses/

```
addmothers.f  done            lhe_event_infos.inc  P1_qq_taptam          run_config.inc  survey.sh
cluster.f     dummy_fct.f     makefile             proc_characteristics  run.inc         symmetry.f
cluster.inc   epsterms.f      maxconfigs.inc       procdef_mg5.dat       setcuts.f       transform.f
coupl.inc     finiteterms.f   maxparticles.inc     projection.f          setscales.f     transformint.f
cuts.f        genps.f         message.inc          randinit              shrinktops.f    unwgt.f
cuts.inc      genps.inc       MGVersion.txt        refine.sh             subproc.mg
dipole.inc    idenparts.f     myamp.f              refine_splitted.sh    subproc.txt
dipolesub.f   initcluster.f   P1_qq_ll             reweight.f            sudakov.inc
```

In the repository "P1_qq_ll" and "P1_qq_taptam", there are plots (jpg or ps) which are the process Feynman diagrams, you can download them to your local laptop to play around.

Drell-Yan process



We have used 5-flavor pdf, for each type of parton, there are two intermediator, “virtual photon” and “Z boson”, and for each intermediator, the final state could be electron pair, muon pair and tau pair. We have 5*2*3=30 diagrams.

Drell-Yan process

ls DY/Cards/

```
delphes_card_ATLAS.dat          madanalysis5_hadron_card_default.dat  pgs_card_TEV.dat
delphes_card_CMS.dat            madanalysis5_parton_card.dat          plot_card.dat
delphes_card_default.dat        madanalysis5_parton_card_default.dat  plot_card_default.dat
delphes_trigger_ATLAS.dat       madspin_card_default.dat              proc_card_mg5.dat
delphes_trigger_CMS.dat         me5_configuration.txt                 pythia8_card_default.dat
delphes_trigger.dat             param_card.dat                        pythia_card_default.dat
delphes_trigger_default.dat     param_card_default.dat                README
grid_card.dat                   pgs_card_ATLAS.dat                    replace_card1.dat
grid_card_default.dat           pgs_card_CMS.dat                      reweight_card_default.dat
ident_card.dat                  pgs_card_default.dat                  run_card.dat
madanalysis5_hadron_card.dat    pgs_card_LHC.dat                      run_card_default.dat
```

Many cards in this repository, currently we only care about the "proc_card_mg5.dat" and "run_card.dat".
proc_card_mg5.dat: the particle definition and process we specified in previous steps
run_card.dat: kinematics we can further define for the objects

cat DY/Cards/run_card.dat

Print out the content, information are formatted in blocks, and with banner at the beginning of each banner.

Drell-Yan process

```
#*********************************************************************
# Number of events and rnd seed                                      *
# Warning: Do not generate more than 1M events in a single run       *
#*********************************************************************
  10000 = nevents ! Number of unweighted events requested
  0    = iseed    ! rnd seed (0=assigned automatically=default))
```

Be careful of the iseed, if "0" is set, the system will assign seed automatically.
Say student A and student B, they both run the DY process, and use both "iseed=0", the iseed is the same (could also depend on the system) and they will have identical outputs.
If student A "launch" (we will discuss later) the DY several times, use "iseed=0" every time, the seed assignment will be handled by MG and the seed will be different.
Change "10000" to "1000".

Drell-Yan process

The random seed will affect the event production, we need different random seeds.

PKU: start with 100, e.g., 100, 101, 102…
THU: start with 200, e.g., 200, 201, 202…
BUAA: start with 300
IHEP: start with 400
FDU: start with 500
ZJU: start with 600
NNU: start with 700
SYSU: start with 800
Online: choose the number you like :-)

```
#**********************************************************************
# Collider type and energy                                           *
# lpp: 0=No PDF, 1=proton, -1=antiproton, 2=photon from proton,      *
#                                          3=photon from electron    *
#**********************************************************************
     1         = lpp1    ! beam 1 type
     1         = lpp2    ! beam 2 type
     6500.0      = ebeam1  ! beam 1 total energy in GeV
     6500.0      = ebeam2  ! beam 2 total energy in GeV
# To see polarised beam options: type "update beam_pol"

#**********************************************************************
# PDF CHOICE: this automatically fixes also alpha_s and its evol.    *
#**********************************************************************
     nn23lo1     = pdlabel     ! PDF set
     230000     = lhaid     ! if pdlabel=lhapdf, this is the lhapdf number
```

Collision energy and parton pdf, we will use the defaults.

Drell-Yan process

```
#*********************************************************************
# Minimum and maximum pt's (for max, -1 means no cut)               *
#*********************************************************************
 10.0  = ptl          ! minimum pt for the charged leptons
 -1.0  = ptlmax       ! maximum pt for the charged leptons
```

Since the energy carried in the beam-direction is unknown in the hadron collider, the transverse momentum (pt) is commonly used in CMS.
Other blocks are similar, the information is written in the banner.

⏭ ./mg5_aMC
⏭ launch DY

You will be asked to modify some files or not, just "Enter" to use the default.
After production complete, quit MG.

⏭ "Ctrl + d"

Drell-Yan process

```
Generating 10000 events with run name run_01
survey  run_01
INFO: compile directory
```

How many events will be generated

```
compile Source Directory
Using random number seed offset = 21
INFO: Running Survey
```

Random seed used

```
INFO: Running Systematics computation
INFO:  Idle: 0,  Running: 4,  Completed: 0 [ current time: 10h42 ]
INFO:  Idle: 0,  Running: 3,  Completed: 1 [  58.7s  ]
INFO:  Idle: 0,  Running: 0,  Completed: 4 [  1m 2s  ]
INFO: # events generated with PDF: NNPDF23_lo_as_0130_qed (247000)
INFO: #Will Compute 145 weights per event.
INFO: #***********************************************************************
#
# original cross-section: 2634.36976809
#     scale variation: +17.1% -17.1%
#     central scheme variation: + 0% -21.7%
# PDF variation: +3.04% -3.04%
#
# dynamical scheme # 1 : 2514.23 +18% -18% # \sum ET
# dynamical scheme # 2 : 2514.78 +18% -18% # \sum\sqrt{m^2+pt^2}
# dynamical scheme # 3 : 2061.91 +22% -21.2% # 0.5 \sum\sqrt{m^2+pt^2}
# dynamical scheme # 4 : 2634.37 +17.1% -17.1% # \sqrt{\hat s}
#***********************************************************************
```

Cross section and related uncertainty information

Drell-Yan process

⏭ cd DY/Events/run_01

```
[lum@farm run_01]$ ls
log_sys_0.txt  log_sys_2.txt  parton_systematics.log   tag_1_MA5_analysis1.log
log_sys_1.txt  log_sys_3.txt  run_01_tag_1_banner.txt  unweighted_events.lhe.gz
```

The “run_01_tag_1_banner.txt” contains the process definition and run_card information, and also the final cross section. The “unweighted_events.lhe.gz” contains information of 1000 events we just generated.

⏭ gunzip unweighted_events.lhe.gz

```
log_sys_0.txt  log_sys_2.txt  parton_systematics.log   tag_1_MA5_analysis1.log
log_sys_1.txt  log_sys_3.txt  run_01_tag_1_banner.txt  unweighted_events.lhe
```

LHE format

LHE: The Les Houches Event file format (LHE) is an agreement between Monte Carlo event generators and theorists to define Matrix Element level event listings in a common language.

⏭️ vi unweighted_events.lhe

1st line: the LHE version

The header block (content between <header> and </header>) includes the process_card, run_card, and other cards defining other parameters in the model we used.

The initial block (right after header block, <init> and </init>) includes energy, pdf and cross section information, this block will appear only one time.

Contents between sequential <event> and </event> corresponding to one event.

LHE format

PDGID status: -1 means incoming, 2 means intermediate, 1 means final state particle 4-vector (px, py, pz, energy) and mass

```
692 <event>
693  5      1 +2.6351600e+03 9.07160600e+01 7.54677100e-03 1.30112700e-01
694        -2 -1    0    0    0  501 -0.0000000000e+00 +0.0000000000e+00 +2.3607368840e+02 2.360736884
    0e+02 0.0000000000e+00 0.0000e+00 1.0000e+00
695         2 -1    0    0  501    0 +0.0000000000e+00 -0.0000000000e+00 -8.7148676183e+00 8.714867618
    3e+00 0.0000000000e+00 0.0000e+00 -1.0000e+00
696        23  2    1    2    0    0 +0.0000000000e+00 +0.0000000000e+00 +2.2735882078e+02 2.447885560
    2e+02 9.0716061259e+01 0.0000e+00 0.0000e+00
697       -15  1    3    3    0    0 +1.1166069998e+01 -3.6540230108e+01 +1.7946313231e+02 1.834939866
    1e+02 1.7770000000e+00 0.0000e+00 1.0000e+00
698        15  1    3    3    0    0 -1.1166069998e+01 +3.6540230108e+01 +4.7895688476e+01 6.129456941
    0e+01 1.7770000000e+00 0.0000e+00 -1.0000e+00
699 <mgrwt>
700 <rscale>  0 0.90716061E+02</rscale>
701 <asrwt>0</asrwt>
702 <pdfrwt beam="2">  1       2 0.13407490E-02 0.90716061E+02</pdfrwt>
703 <pdfrwt beam="1">  1      -2 0.36319024E-01 0.90716061E+02</pdfrwt>
704 <totfact> 0.58531905E+04</totfact>
705 </mgrwt>
```

Line 693: “5” means there are five particles in this event (including incoming and outgoing)
Line 694-698: information of five particles

Block <rwgt> and </rwgt> include weights of event from other pdf.
Some events may just have 4 particles, this is due to virtual photon is not in the LHE

LHE format -> Root format

- cd /data/pubfs/pku_visitor/public_write/generator_resource/CMSSW_10_2_5/src
- source /cvmfs/cms.cern.ch/cmsset_default.sh
- cmsenv
- cd /YOURPATH/MG5_aMC_v2_7_3/bin
- ./mg5_aMC
- install ExRootAnalysis

```
MG5_aMC>install ExRootAnalysis
Downloading http://madgraph.phys.ucl.ac.be/Downloads/ExRootAnalysis/ExRootAnalysis_V1.1.5.tar.gz
--2021-12-06 12:15:46--  http://madgraph.phys.ucl.ac.be/Downloads/ExRootAnalysis/ExRootAnalysis_V1.1.5.tar
.gz
Resolving madgraph.phys.ucl.ac.be (madgraph.phys.ucl.ac.be)... 130.104.133.249
Connecting to madgraph.phys.ucl.ac.be (madgraph.phys.ucl.ac.be)|130.104.133.249|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49641 (48K) [application/x-gzip]
Saving to: 'ExRootAnalysis.tgz'

100%[=================================================================>] 49,641      150KB/s   in 0.3s

2021-12-06 12:15:48 (150 KB/s) - 'ExRootAnalysis.tgz' saved [49641/49641]

compile ExRootAnalysis. This might take a while.
```

After installation complete, exit MG

- "Ctrl + d"

LHE format -> Root format

- cd DY/Events/run_01/
- YOURPATH/MG5_aMC_v2_7_3/ExRootAnalysis/ExRootLHEFConverter unweighted_events.lhe dy.root

```
Warning in <TTree::Bronch>: Using split mode on a class: TRootWeight with a custom Streamer
** Reading unweighted_events.lhe
** [################################################################] (100.00%)
** Exiting...
```

You should have root file "dy.root" now.

- root -l dy.root
- _file0->ls()
- LHEF->GetEntries()

```
root [1] _file0->ls()
TFile**         dy.root
 TFile*         dy.root
  KEY: TTree    LHEF;1  Analysis tree
```

```
root [2] LHEF->GetEntries()
(long long) 10000
```

LHE format -> Root format

- LHEF->Show(1) -> show the second event

```
======> EVENT:1
 Event             = 1
 Event.fUniqueID = 0
 Event.fBits       = 50331648
 Event.Number      = 2
 Event.Nparticles = 5
 Event.ProcessID = 1
 Event.Weight      = 2635.16
 Event.ScalePDF    = 90.7161
 Event.CouplingQED = 0.00754677
 Event.CouplingQCD = 0.130113
 Event_size        = 1
 Rwgt              = 145
 Rwgt.fUniqueID  = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 Rwgt.fBits        = 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 5033164
8, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 50331648, 503
31648
 Rwgt.Weight       = 2328.5, 2249.01, 2249.4, 1914.45, 2328.5, 2635.16, 2561.67, 2562.04, 2249.4, 2635.16, 2
916.83, 2849.55, 2849.88, 2562.04, 2916.83, 2328.5, 2249.01, 2249.4, 1914.45, 2328.5
 Rwgt_size         = 145
 Particle          = 5
 Particle.fUniqueID = 0, 0, 0, 0, 0
 Particle.fBits    = 50331648, 50331648, 50331648, 50331648, 50331648
```

LHE format -> Root format

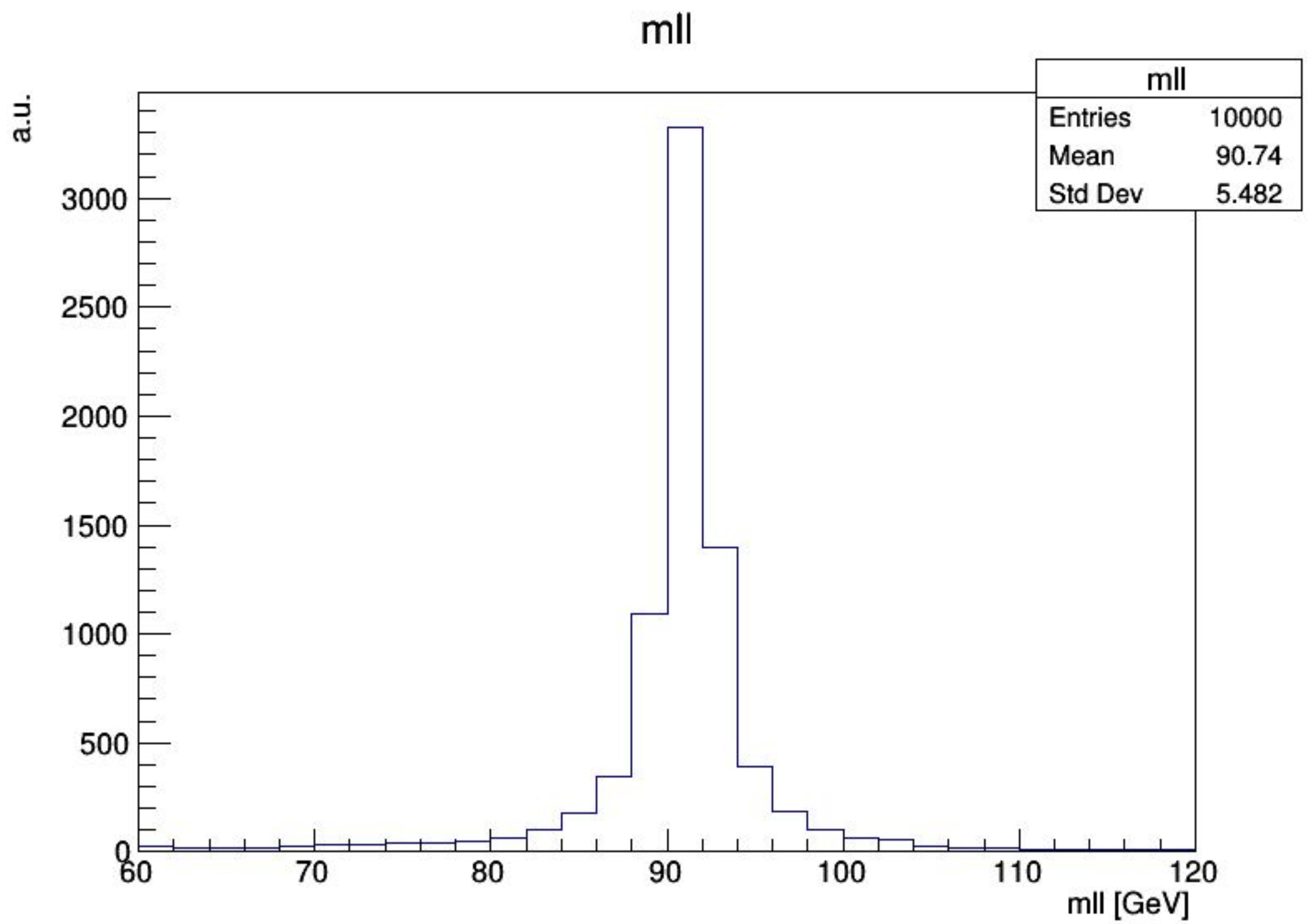
```
Particle.PID      = -2, 2, 23, -15, 15
Particle.Status = -1, -1, 2, 1, 1
Particle.Mother1 = -1, -1, 0, 2, 2
Particle.Mother2 = -1, -1, 1, 2, 2
Particle.ColorLine1 = 0, 501, 0, 0, 0
Particle.ColorLine2 = 501, 0, 0, 0, 0
Particle.Px       = -0, 0, 0, 11.1661, -11.1661
Particle.Py       = 0, -0, 0, -36.5402, 36.5402
Particle.Pz       = 236.074, -8.71487, 227.359, 179.463, 47.8957
Particle.E        = 236.074, 8.71487, 244.789, 183.494, 61.2946
Particle.M        = 0, 0, 90.7161, 1.777, 1.777
Particle.PT       = 0, 0, 0, 38.2082, 38.2082
Particle.Eta      = 999.9, -999.9, 999.9, 2.25121, 1.0498
Particle.Phi      = 0, 0, 0, -1.27423, 1.86737
Particle.Rapidity = 999.9, -999.9, 999.9, 2.25015, 1.04896
Particle.LifeTime = 0, 0, 0, 0, 0
Particle.Spin     = 1, -1, 0, 1, -1
Particle size     = 5
```

The particle related arrays are with length 5, corresponding to 5 particles. (compare these information with those in slide 25)

Making plot

- cp /data/pubfs/pku_visitor/public_write/generator_resource/mll.py .
  Modify the "libExRootPath" in mll.py to your path

- python mll.py



Invariant mass of two leptons, this plot include on-shell/off-shell Z contribution, and contribution from virtual photon.

Home work

In code mll.py, the invariant mass of two leptons are constructed using

```
for i in range(tree_all.GetEntries()):
  l1=TLorentzVector(0,0,0,0)
  l2=TLorentzVector(0,0,0,0)
  tree_all.GetEntry(i)
  for j in range(tree_all.Particle_size):
    if (abs(tree_all.Particle[j].PID)==11 or abs(tree_all.Particle[j].PID)==13 or abs(tree_all.Particle[j]
.PID)==15) and tree_all.Particle[j].Status==1:
      if l1.Pt()==0:
        l1.SetPtEtaPhiM(tree_all.Particle[j].PT,tree_all.Particle[j].Eta,tree_all.Particle[j].Phi,tree_all
.Particle[j].M)
      else:
        l2.SetPtEtaPhiM(tree_all.Particle[j].PT,tree_all.Particle[j].Eta,tree_all.Particle[j].Phi,tree_all
.Particle[j].M)
  h_all.Fill((l1+l2).M())
```

PID: particle ID, 11 is electron, 13 is muon, 15 is tau

Try to make plots for leptons pt/eta/energy……

CMS sample production

CMS production:
LHE -> Generator & Simulation -> Digitalization & reconstruction -> AOD -> MINIAOD -> NANOAOD

Keep the LHE file, it will be used in sample production chain.

Part2

Models

MG5 loads the SM by default

- Other sm versions (sm_loop, sm-no_bmass,...)

BSM models can be imported (SMEFT, MSSM, 2HDM,...)

- New particles, coupling orders
- Universal FeynRules Output (UFO) model (just a python module!)

```
MG5_aMC> import model SMEFTsim_topU3l_MwScheme_UFO
```

```
MG5_aMC> import model SMEFTatNLO
```

```
MG5_aMC> convert model …(convert model to python3)
```

Automatically downloaded from FeynRules database https://feynrules.irmp.ucl.ac.be

Restriction cards inside model folder: restrict_XYZ.dat

- param_card with e.g. parameters fixed or =0 to simplify model

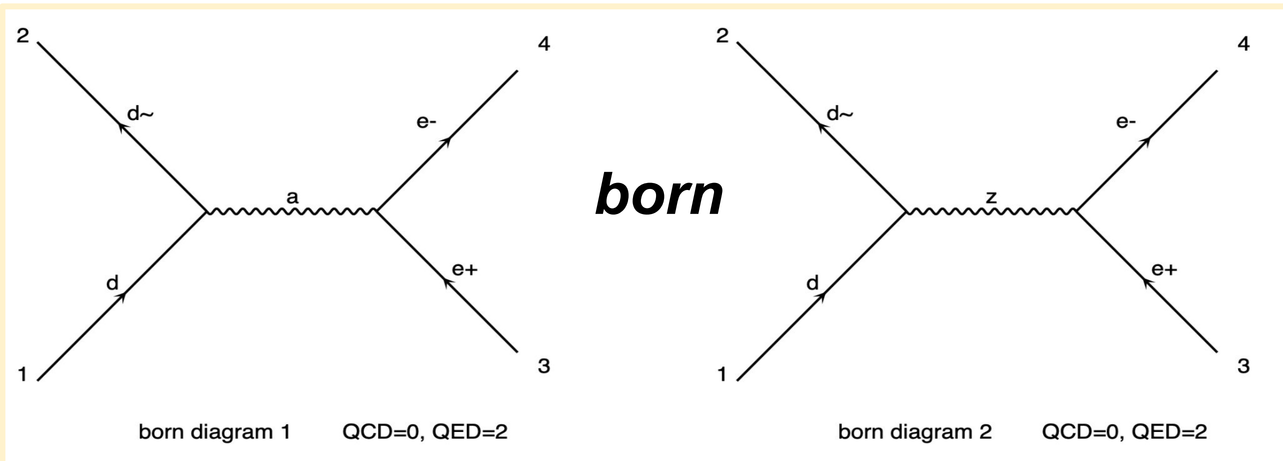
```
MG5_aMC> import model SMEFTatNLO-XYZ
```

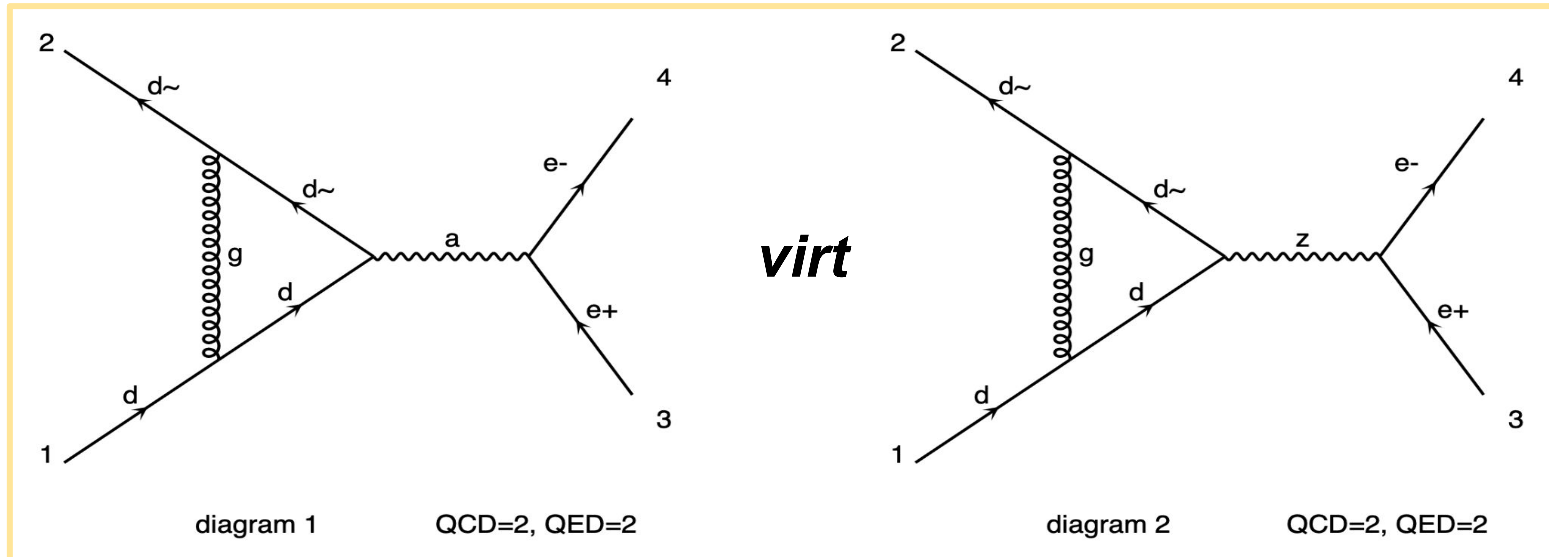
Models

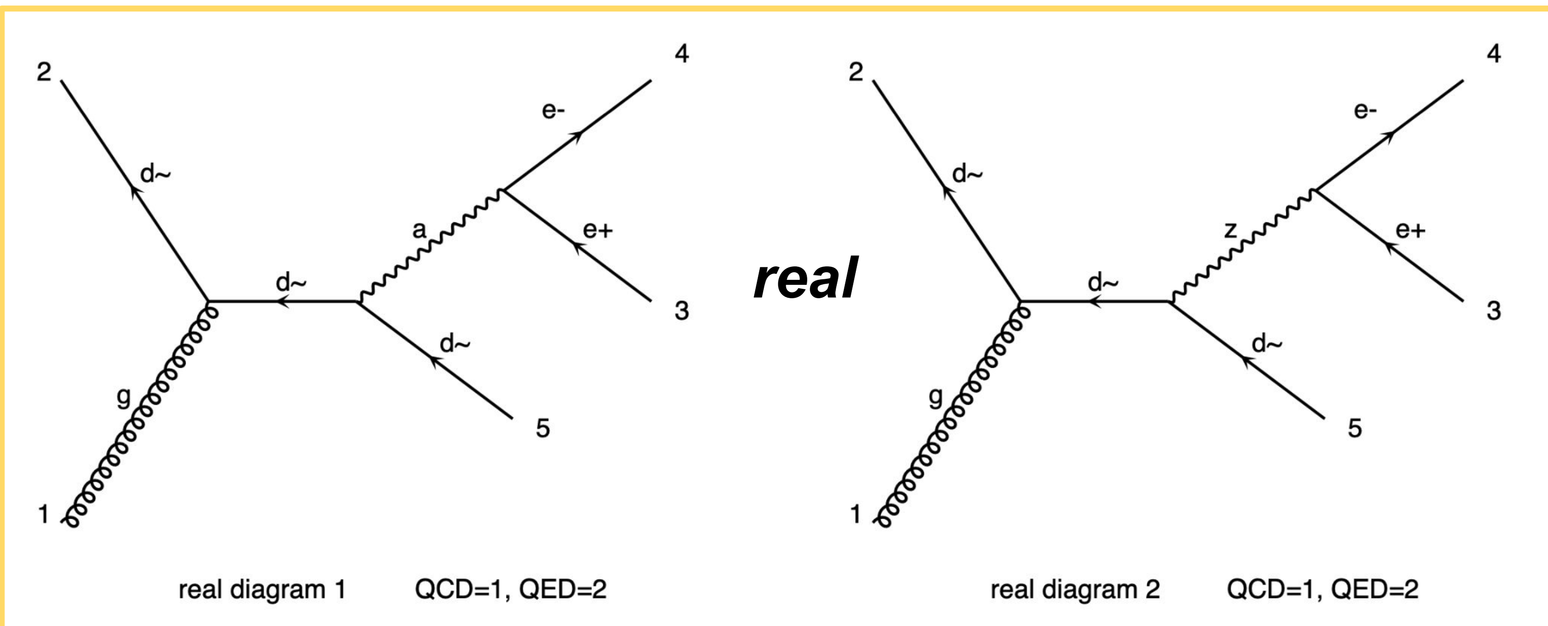
Previously we do with the LO model. Let's try the DY process with NLO model.

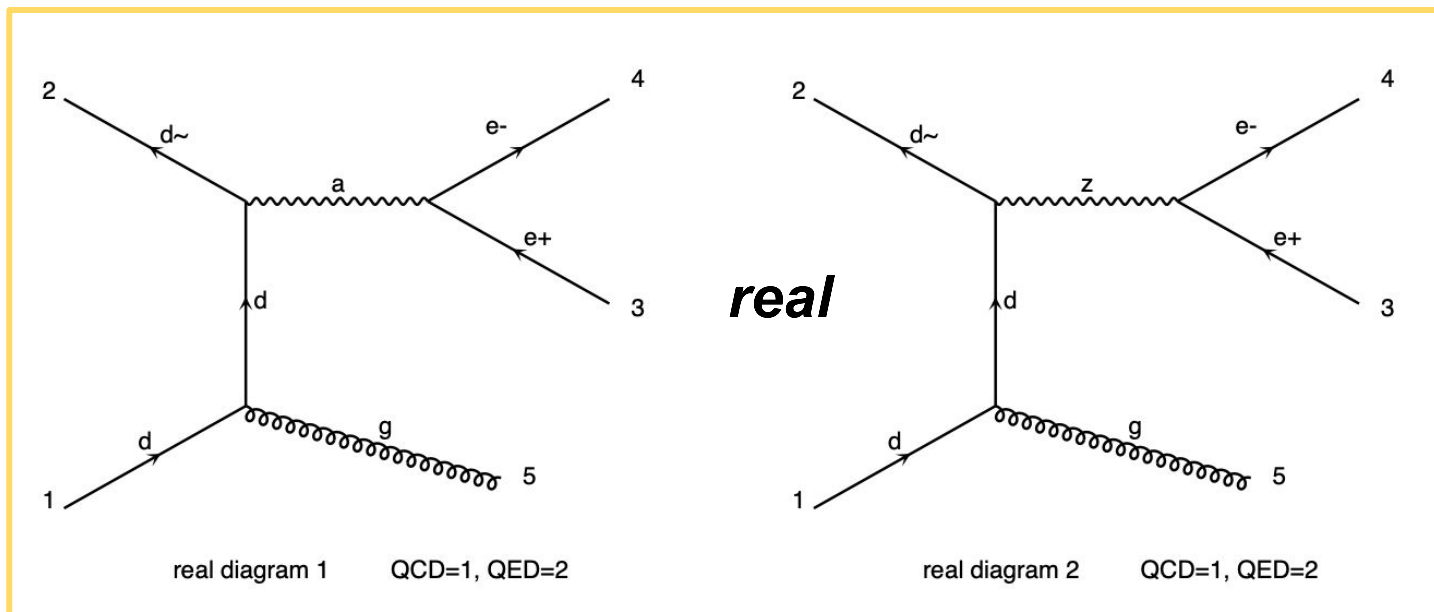
```
MG5_aMC> import model  loop_sm-no_b_mass
MG5_aMC>  define lep+ = e+ mu+ ta+
MG5_aMC>  define lep- = e- mu- ta-
MG5_aMC> generate p p > lep+ lep- QCD=0 QED=2 [QCD]
```

Feynman diagrams: https://qguo.web.cern.ch/qguo/MG5/MG5_aMC_v2_7_3/ppllm_NLO/









Syntax

Adding another process

- another “generate” will overlap the first generation

```
MG5_aMC> generate p p > t t~
MG5_aMC> add process p p > b b~
```

Jet matching

- Pythia does the parton shower and hadronization. Actually there is double-counting events. Need to "match" the two samples, such as MLM, FxFx…

```
MG5_aMC> generate g g > h
MG5_aMC> add process g g > h j
MG5_aMC> add process g g > h j j
```

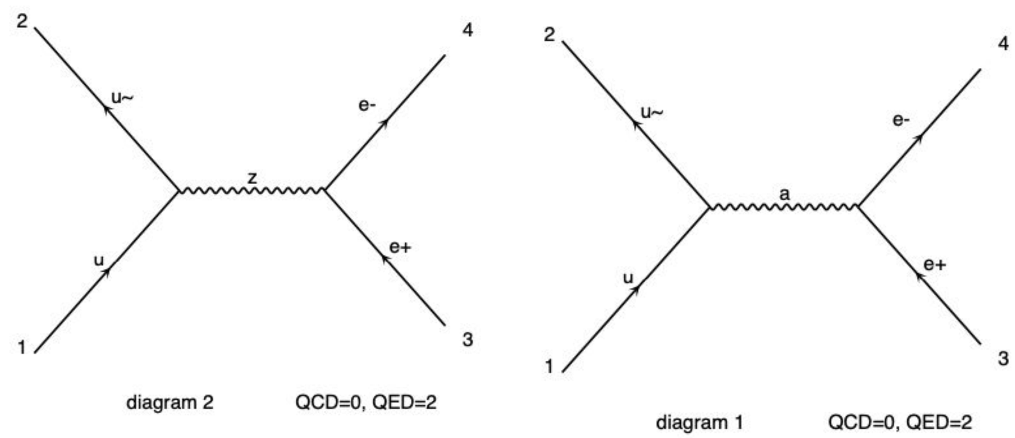
Decay

- Interested in only a subset of events with special properties. ***LO*** could do as such way.

```
MG5_aMC> generate p p >  W+, W+ > l+ vl
MG5_aMC> generate p p > t t~, (t > W+ b, W+ > j j), (t~ > W- b~, W- > l- vl~)
```

Syntax

Requirement of the presence or absence of particles in Feynman diagrams



- Exclude Feynman diagrams that contain a particular particle.
  - MG5_aMC> generate p p > e+ e- / Z Links: EZ
  - Xsec: 207.8 ± 0.49 pb
- Exclude a particle from appearing in the s-channel
  - MG5_aMC> generate p p > e+ e- $ Z Links: ESZ Xsec: 213.1 ± 0.58 pb
- Include only those Feynman diagrams in which a certain particle is present in the s-channel
  - MG5_aMC> generate p p > Z > e+ e- Links: SZ Xsec: 638.9 ± 1.3 pb
- Include only those Feynman diagrams with the certain particle to be the intermediate particle
  - MG5_aMC> generate p p > Z, Z > e+ e- Links: Z Xsec: 1421 ± 1.4 pb

Be careful of using each of these because they can result in violating gauge invariance or unitarity.

https://www.niu.edu/spmartin/madgraph/madsyntax.html

MG5_aMC> generate p p > e+ e-
Links: ppeem Xsec: 843 ± 2.1pb

Exercise1

git clone -b all https://github.com/qyguo/lhe_related.git cards
mv cards YOURPATH/MG5_aMC_v2_7_3/../

- cd YOURPATH/MG5_aMC_v2_7_3
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/ YOURPATH/MG5_aMC_v2_7_3/../

LO:

- ./bin/mg5_aMC ../cards/ppeem_EZ/proc_card.dat
- ./bin/mg5_aMC ../cards/ppeem_ESZ/proc_card.dat
- ./bin/mg5_aMC ../cards/ppeem_SZ/proc_card.dat
- ./bin/mg5_aMC ../cards/ppeem_Z/proc_card.dat
- ./bin/mg5_aMC ../cards/ppeem/proc_card.dat
- ### sed -i "s/^.* = ptl / 10.0 = ptl /" ppeem*/Cards/run_card.dat
- sed -i "s/^.* = use_syst/   False = use_syst/" ppeem*/Cards/run_card.dat
- ./ppeem_EZ/bin/generate_events
- ./ppeem_ESZ/bin/generate_events
- ./ppeem_SZ/bin/generate_events
- ./ppeem_Z/bin/generate_events
- ./ppeem/bin/generate_events
- mkdir plots
- cd plots
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/*.py ./
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/*.cc ./

Exercise1

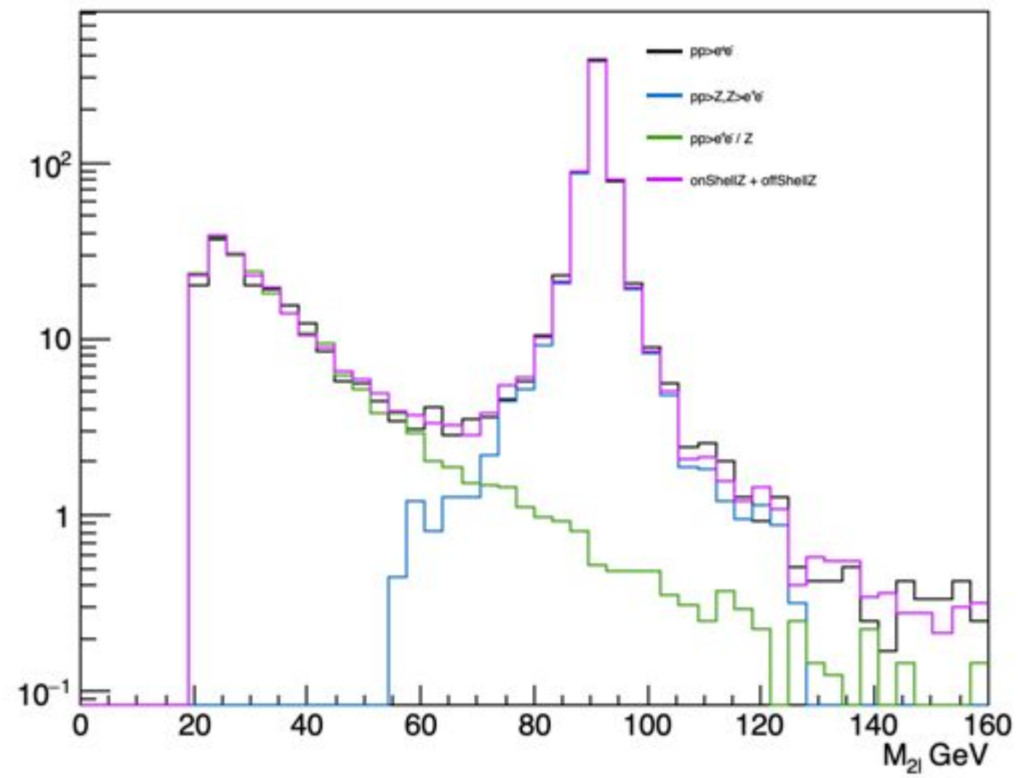
- cd YOURPATH/MG5_aMC_v2_7_3
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/ YOURPATH/MG5_aMC_v2_7_3/../

LO:

- cp ../ppeem_EZ/Events/run_01/unweighted_events.lhe.gz ./ppeem_EZ_unweighted_events.lhe.gz
- cp ../ppeem_ESZ/Events/run_01/unweighted_events.lhe.gz ./ppeem_ESZ_unweighted_events.lhe.gz
- cp ../ppeem_SZ/Events/run_01/unweighted_events.lhe.gz ./ppeem_SZ_unweighted_events.lhe.gz
- cp ../ppeem_Z/Events/run_01/unweighted_events.lhe.gz ./ppeem_Z_unweighted_events.lhe.gz
- cp ../ppeem/Events/run_01/unweighted_events.lhe.gz ./ppeem_unweighted_events.lhe.gz
- gunzip *.lhe.gz
- python lhe.py plotsppeem_unweighted_events
- python lhe.py plotsppeem_Z_unweighted_events
- python lhe.py plotsppeem_SE_unweighted_events
- python lhe.py plotsppeem_EZ_unweighted_events
- python lhe.py plotsppeem_ESZ_unweighted_events
- root -l -b -q draw_1f.cc\(\"ppeem_unweighted_events\"\)
- root -l -b -q draw_1f.cc\(\"ppeem_Z_unweighted_events\"\)
- root -l -b -q draw_1f.cc\(\"ppeem_EZ_unweighted_events\"\)
- root -l -b -q draw_1f.cc\(\"ppeem_SE_unweighted_events\"\)
- root -l -b -q draw_1f.cc\(\"ppeem_ESZ_unweighted_events\"\)

Exercise1

- sed -i "s/^.*= cut_decays / True = cut_decays /" ppeem_Z/Cards/run_card.dat
- ./ppeem_Z/bin/generate_events
- cd plots; cp ../ppeem_Z/Events/run_02/unweighted_events.lhe.gz ./ppeem_Z_2_unweighted_events.lhe.gz
- gunzip ppeem_Z_2_unweighted_events.lhe.gz
- python lhe.py ppeem_Z_2_unweighted_events
- root -l -b -q draw_1f.cc\(\"ppeem_Z_2_unweighted_events\"\)
- root -l -b -q ./draw_1f_plus.cc

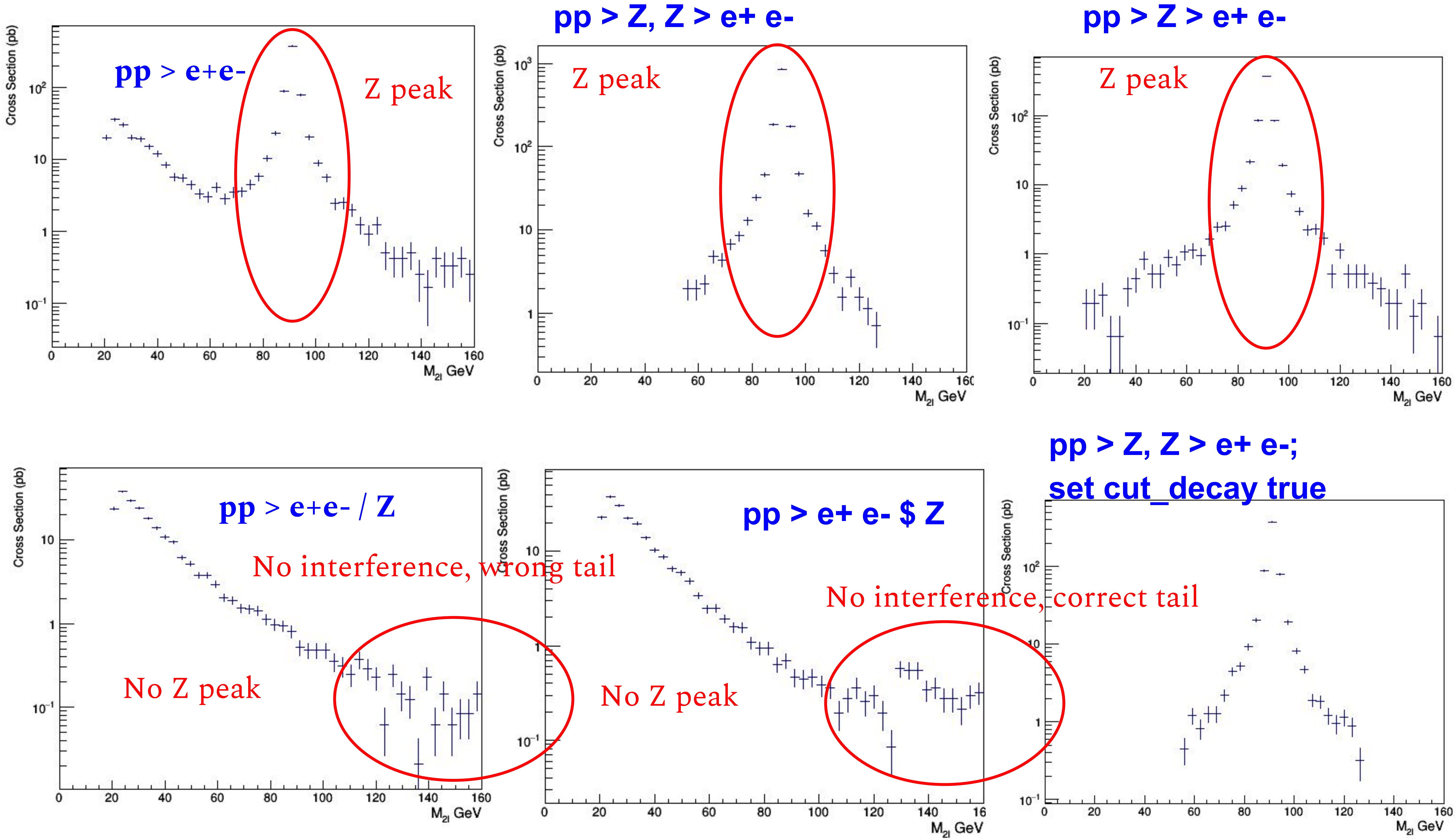


1. p p > e+ e- (all contributions)
2. p p > z, z > e+ e- (z is on-shell)
3. p p > e+ e- $z (forbids s-channel z to be on-shell)
4. p p > e+ e- /z (forbids any z)

From the definition above: process 1 = process 2 + p

	Process 1	Process 2 (False = cut_decays)	Process 2 (True = cut_decays)	Process 3	Process 4
XS (with default cut)	843.0 pb	1420.7 pb	631.7 pb	213.1 pb	207.8 pb
contributions	Onshell Z Offshell Z Virtual photon Interference	Onshell Z (but no cut on the decayed electrons)	Onshell Z (with cuts on the decayed electrons)	Offshell Z Virtual photon Interference	Virtual photon

Exercise1


pp > e+e-
Z peak
pp > Z, Z > e+ e-
Z peak
pp > Z > e+ e-
Z peak
pp > e+e- / Z
No interference, wrong tail
No Z peak
pp > e+ e- $ Z
No interference, correct tail
No Z peak
pp > Z, Z > e+ e-;
set cut_decay true
Cross Section (pb)
M2l GeV

Exercise2

- cd YOURPATH/MG5_aMC_v2_7_3
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/ YOURPATH/MG5_aMC_v2_7_3/../

NLO:

- ./bin/mg5_aMC ../cards/ppllm_NLO/proc_card.dat
- sed -i "s/HERWIG6   = parton_shower/PYTHIA8   = parton_shower/" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= fixed_fac_scale/ True   = fixed_fac_scale/" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= fixed_ren_scale/ True   = fixed_ren_scale/" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= nevents / 100 = nevents /" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= jetalgo / -1.0 = jetalgo /" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= jetradius / 0.4   = jetradius /" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= ptj / 5.0   = ptj /" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= ptgmin / 0.0   = ptgmin /" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*hadronize   = T /hadronize   = F /" ppllm_NLO/Cards/shower_card.dat
- sed -i "s/^.*njmax   = 0 /njmax   = -1.0 /" ppllm_NLO/Cards/shower_card.dat
- sed -i "s/extralibs.*/extralibs   = stdhep Fmcfio dl   # Extra-libraries (not LHAPDF)/" ppllm_NLO/Cards/shower_card.dat
- sed -i "s/EXTRALIBS.*/EXTRALIBS   = stdhep Fmcfio dl   # Extra-libraries (not LHAPDF)/" ppllm_NLO/Cards/shower_card.dat
- ./ppllm_NLO/bin/generate_events
- ./ppllm_NLO/bin/calculate_xsect

Exercise2

- cd YOURPATH/MG5_aMC_v2_7_3
- cp -r /data/pubfs/pku_visitor/guoqianying/public/cards/ YOURPATH/MG5_aMC_v2_7_3/../

NLO:

- ./bin/mg5_aMC ../cards/ppllm_NLO/proc_card.dat
- sed -i "s/HERWIG6 = parton_shower/PYTHIA8 = parton_shower/" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= fixed_fac_scale/ True = fixed_fac_scale/" ppllm_NLO/Cards/run_card.dat
- sed -i "s/^.*= fixed_ren_scale/ True = fixed_ren_scale/" ppllm_NLO/Cards/run_card.dat
- s
- s
- s
- s
- s
- s
- s
- s card.dat
- s
  ppllm_NLO/Cards/shower_card.dat
- ./ppllm_NLO/bin/generate_events
- ./ppllm_NLO/bin/calculate_xsect

```
--------------------------------------------------------------
    Final results and run summary:
    Process p p > lep+ lep- QCD=0 QED=2 [QCD]
    Run at p-p collider (6500.0 + 6500.0 GeV)
    Total cross section: 6.835e+03 +- 2.7e+01 pb
--------------------------------------------------------------
    Scale variation (computed from histogram information):
        Dynamical_scale_choice 0 (envelope of 9 values):
            6.835e+03 pb  +3.0% -6.6%
--------------------------------------------------------------
```

Additional slides

generate p p > lep+ lep- a j j QCD=0

```
INFO: Process b~ b > ta+ ta- a b b~ added to mirror process b b~ > ta+ ta- a b b~
INFO: Crossed process found for b~ b~ > e+ e- a b~ b~, reuse diagrams.
INFO: Crossed process found for b~ b~ > mu+ mu- a b~ b~, reuse diagrams.
INFO: Crossed process found for b~ b~ > ta+ ta- a b~ b~, reuse diagrams.
765 processes with 105948 diagrams generated in 120.187 s
Total: 765 processes with 105948 diagrams
```

Very complicated process, 100k Feynman diagram! Now we have all the process, we need to make a output of the process. **Don't quit MG before the output, you will lose what you have run.**

output ZA_EW

```
save configuration file to /home/Documents/MG5_aMC_v2_7_3/bin/ZA_EW/Cards/me5_configuration.txt
INFO: Use Fortran compiler gfortran
INFO: Use c++ compiler g++
INFO: Generate jpeg diagrams
INFO: Generate web pages
Output to directory /home/Documents/MG5_aMC_v2_7_3/bin/ZA_EW done.
Type "launch" to generate events from this process, or see
/home/Documents/MG5_aMC_v2_7_3/bin/ZA_EW/README
Run "open index.html" to see more information about this process.
```

"Ctrl + d"

EW Zγ

Now in the "bin" path, you will have repository "ZA_EW"

⏭ ls ZA_EW

```
✔ bin ls ZA_EW
bin     Events  index.html  madevent.tar.gz  README                Source        TemplateVersion.txt
Cards   HTML    lib         MGMEVersion.txt  README.systematics    SubProcesses
```

⏭ ls ZA_EW/SubProcesses/

```
✔ bin ls ZA_EW/SubProcesses/
addmothers.f  done           lhe_event_infos.inc  P1_qq_taptamaqq          run_config.inc  survey.sh
cluster.f     dummy_fct.f    makefile             proc_characteristics     run.inc         symmetry.f
cluster.inc   epsterms.f     maxconfigs.inc       procdef_mg5.dat          setcuts.f       transform.f
coupl.inc     finiteterms.f  maxparticles.inc     projection.f             setscales.f     transformint.f
cuts.f        genps.f        message.inc          randinit                 shrinktops.f    unwgt.f
cuts.inc      genps.inc      MGVersion.txt        refine.sh                subproc.mg
dipole.inc    idenparts.f    myamp.f              refine_splitted.sh       subproc.txt
dipolesub.f   initcluster.f  P1_qq_llaqq          reweight.f               sudakov.inc
```

In the repository "P1_qq_llaqq" and "P1_qq_taptamaqq", there are many plots (jpg and ps) which are the process Feynman diagrams, you can download them to your local laptop to play around.

EW Zγ

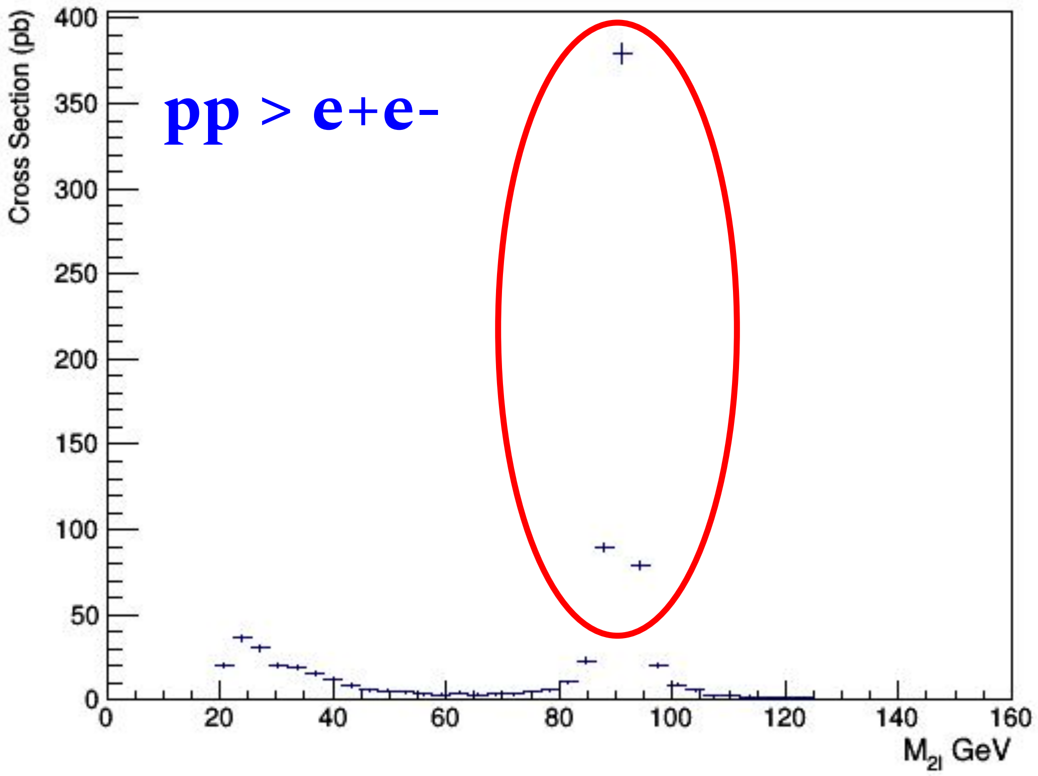
Besides the process definition, we need to make further requirements on the objects, e.g., the energy, transverse momentum and the angular requirements. These values can be set in “ZA_EW/Cards/run_card.dat”

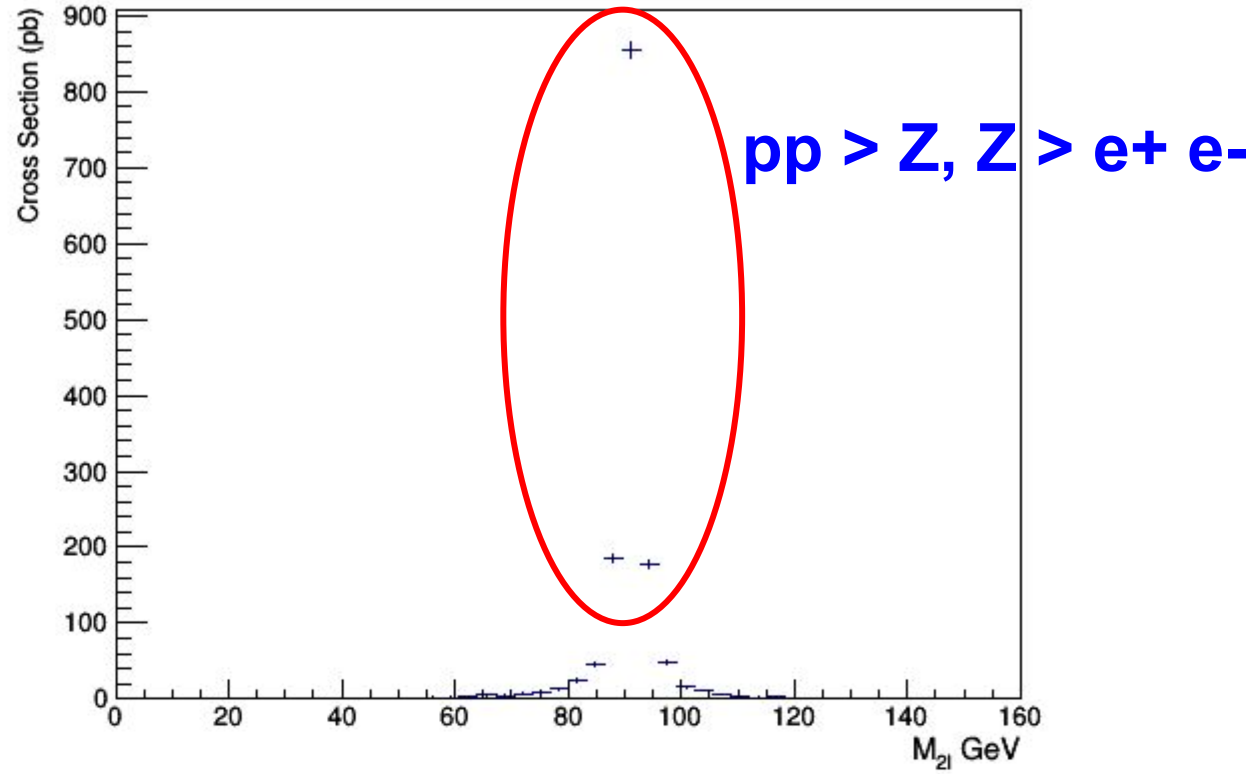
To make our private sample consistent with the official one, we need use library “LHAPDF” and specify the pdf.

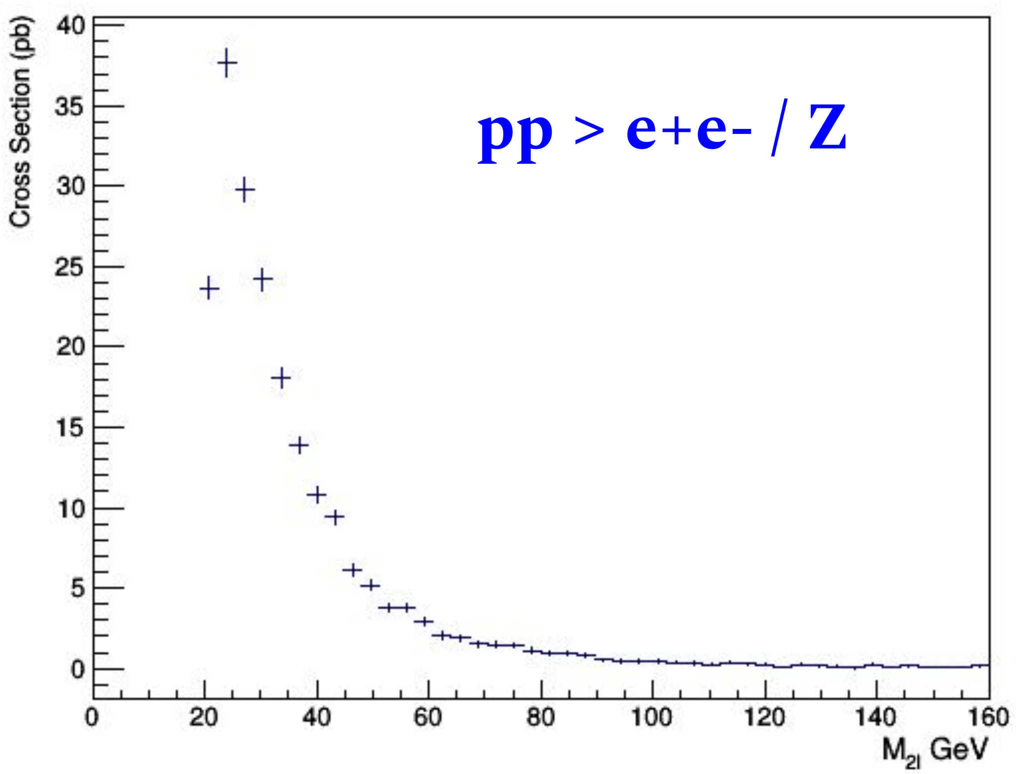
- export LD_LIBRARY_PATH=/home/pku/lum/LHAPDF/lib:$LD_LIBRARY_PATH
- export PATH=/home/pku/lum/LHAPDF/bin:$PATH
- cat ZA_EW/Cards/run_card.dat

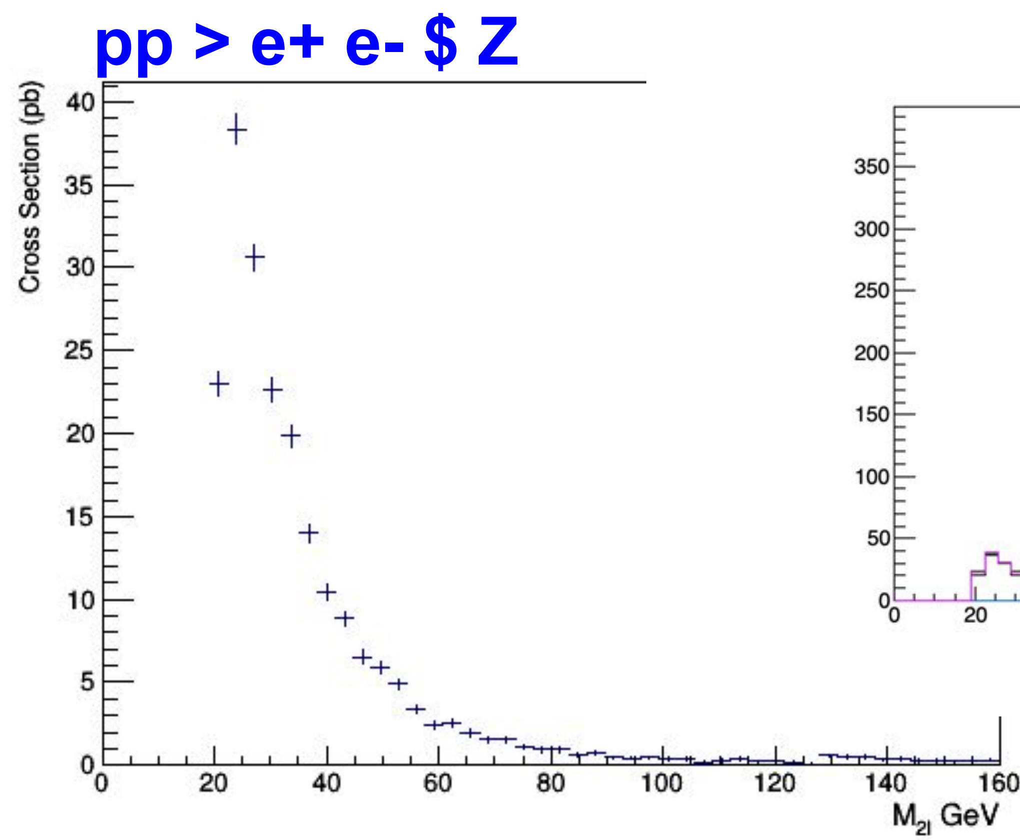
There are many cuts we need to modify, use “vim” or anyother text editor you like to modify them.

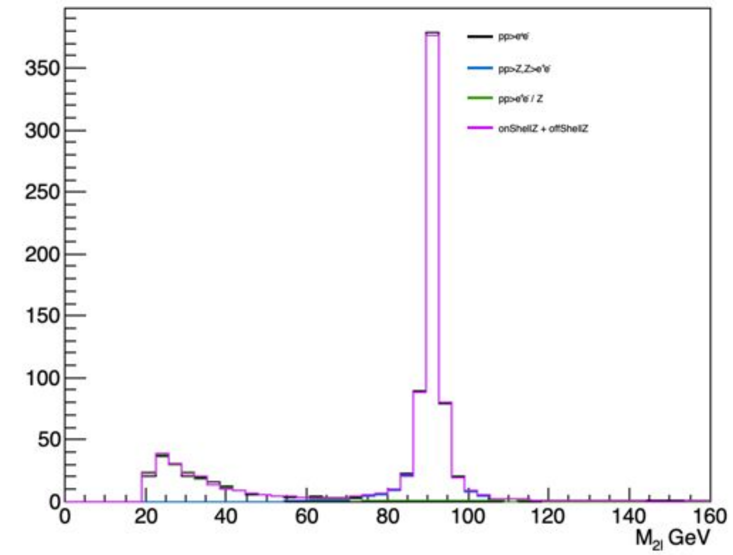
Exercise1


pp > e+e-
Cross Section (pb)
M2l GeV

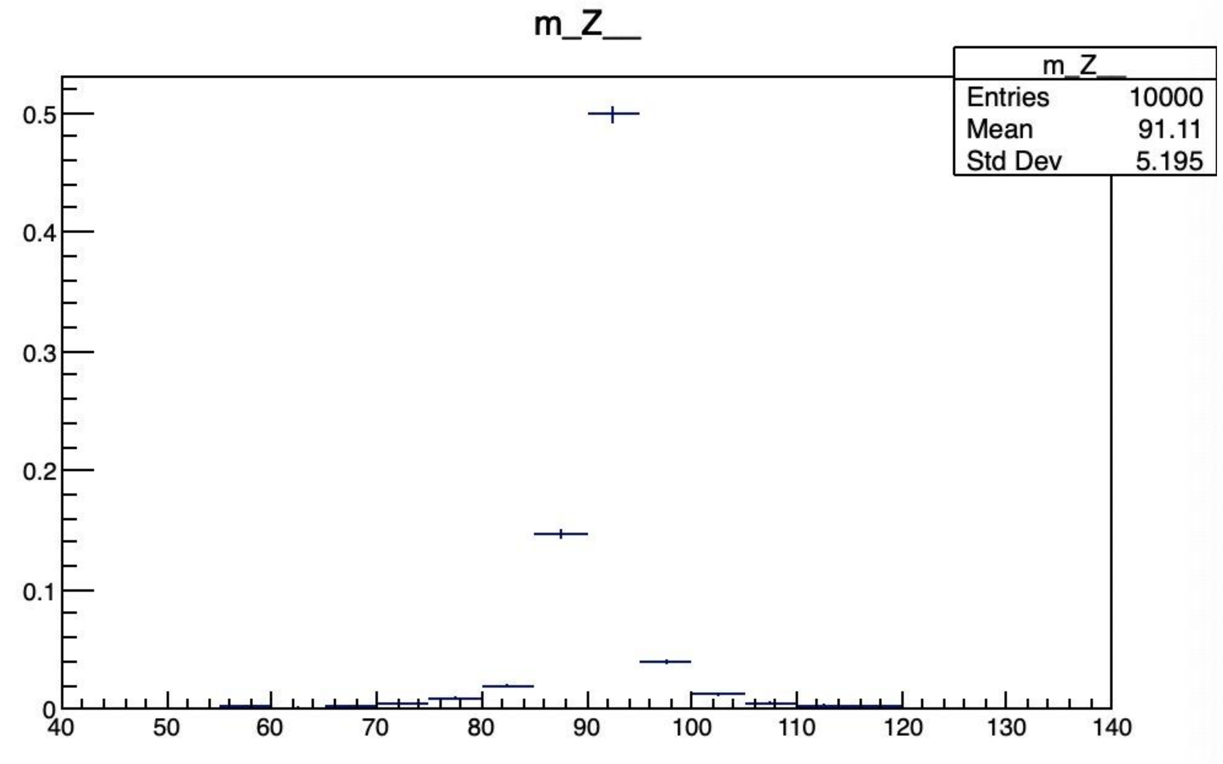

pp > Z, Z > e+ e-
Cross Section (pb)
M2l GeV

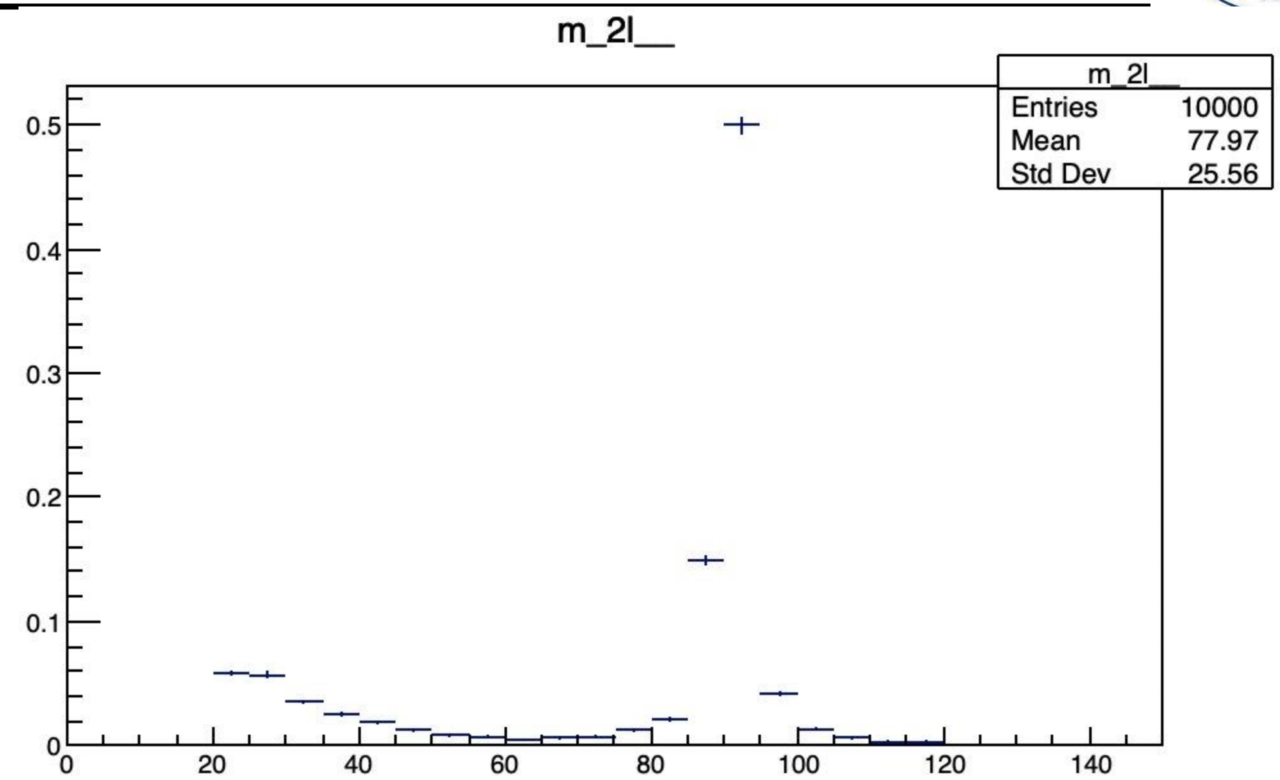

pp > e+e- / Z
Cross Section (pb)
M2l GeV

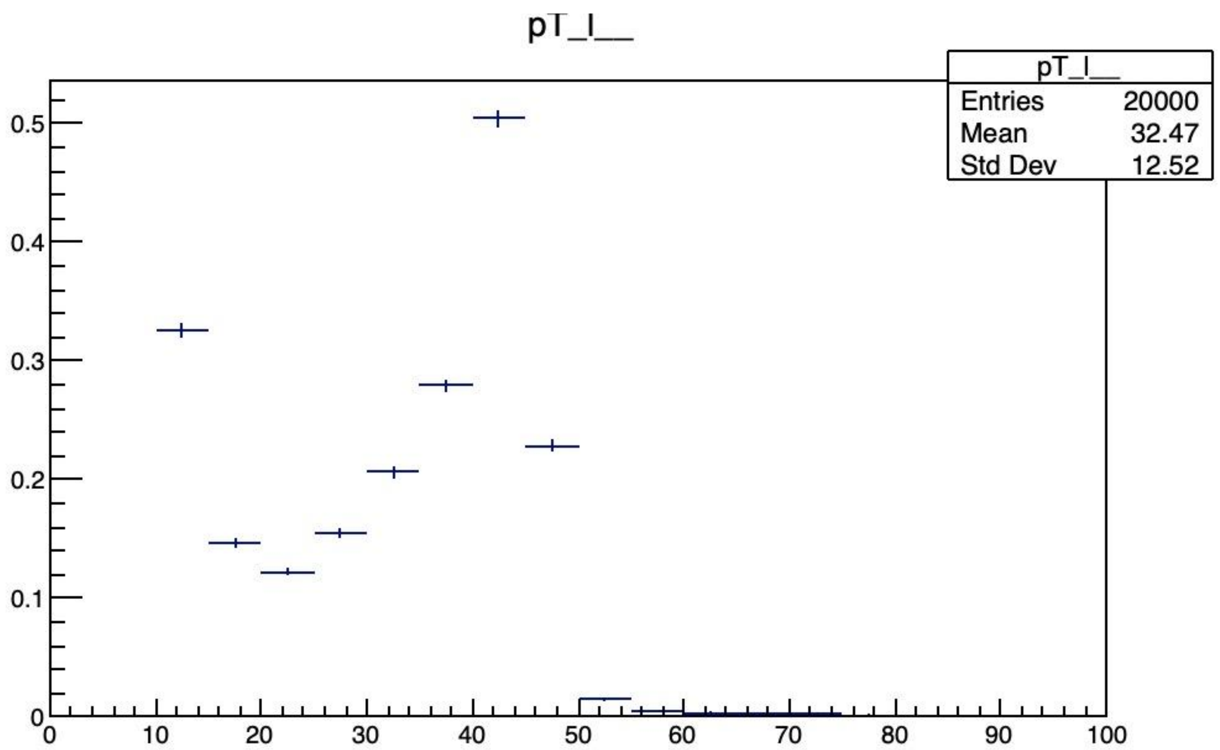

pp > e+ e- $ Z
Cross Section (pb)
M2l GeV


M2l GeV

Making plot







1) Z boson mass directly from the pdgid 23 and status 2
2) Z boson mass from the di-leptons reconstruction.
3) pT of leptons (e, muon, tau)

git clone https://github.com/qyguo/lhe_related.git -b noWeight
python lhe.py <input>.lhe <output>.root