# MG5aMC tutorial

Eleni Vryonodou, Cen Zhang, Ilaria Brivio
And O. Mattelaer

# Avoid internet !

- We are many on the room.
  - ➡ External network is bounded to be slow
- Setup Madgraph to use the local network
  - ➡ export MG5aMC_WWW="http://192.168.XXX.YYY:8000"
  - ➡ This requires version 2.6.5
- Need to download MG5aMC
  - ➡ http://192.168.XXX.YYY:8000/MG5
- For PDF:
  - ➡ http://192.168.XXX.YYY:8000/PDF

# Ex. I: Install MadGraph 5!

- http://192.168.000.000:8000/MG5/
- untar it (tar -xzpvf MG5_XXX.tgz)
- launch it ( $ ./bin/mg5_amc)

- learn it!
  - ➡ Type tutorial and follow instructions

- install external package
  - ➡ install pythia8
  - ➡ install MadAnalysis5
- Be sure that you have run export cmd in that shell
  - ➡ export MG5aMC_WWW="http://192.168.XXX.YYY:8000"

# Where to find help (after the school)?

- Type tutorial
- Use the command "help" / "help XXX"
  - ➡ "help" tell you the next command that you need to do.
- Launchpad:
  - ➡ https://answers.launchpad.net/madgraph5
  - ➡ FAQ: https://answers.launchpad.net/madgraph5/+faqs

# Ex. II : Order

Goal

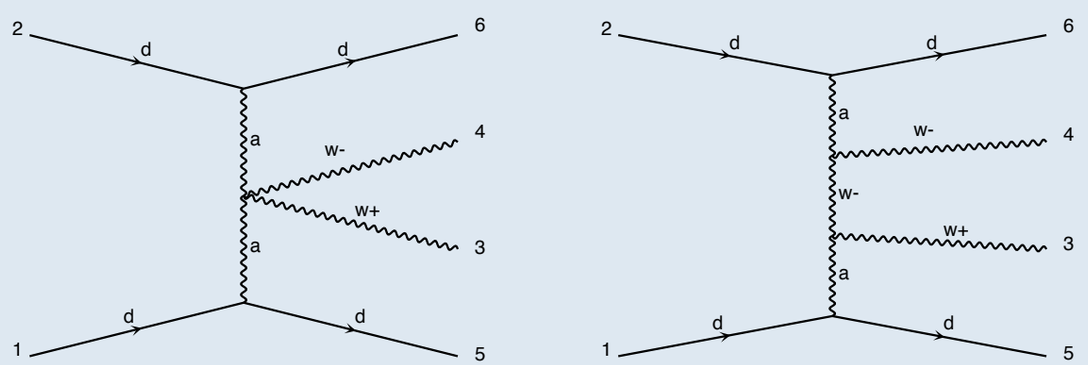
- What's the default choice for QED/QCD order

Learn

- What's the difference between
  - ➡ p p > t t~
  - ➡ p p > t t~ QED=0
  - ➡ p p > t t~ QED<=2
- Compute the cross-section for each of those and check the diagram

Check

- Generate VBF process ( two jet + two W in final state) only the diagram!
- check that you have the QED diagram that you want:



# Solution I : Syntax

- What's the meaning of the order QED/QCD
  - ➡ By default MG5 takes the lowest order in QED!

```
INFO: Trying coupling order WEIGHTED<=2: WEIGTHED IS 2*QED+QCD
```

  - ➡ p p > t t~ IS the same as p p > t t~ QED=0
  - ➡ p p > t t~ QED<=2 has additional diagrams (photon/z exchange)

p p > t t~

| Cross section (pb) |
| --- |
| 555 ± 0.84 |

p p > t t~ QED=2

| Cross section (pb) |
| --- |
| 555.8 ± 0.91 |

No significant QED contribution

Number computed here with cteq6l1 PDF set (old default)

# Solution I Syntax

- generate p p > w+ w- j j
  - ➡ 76 processes
  - ➡ 1432 diagrams
  - ➡ None of them are VBF

- generate p p > w+ w- j j QED<=2
  - ➡ 76 processes
  - ➡ 1432 diagrams
  - ➡ None of them are VBF

- generate p p > w+ w- j j QED<=4
  - ➡ 76 processes
  - ➡ 5332 diagrams
  - ➡ VBF present! + those not VBF

- generate p p > w+ w- j j QCD=0
  - ➡ 60 processes
  - ➡ 3900 diagrams
  - ➡ VBF present!

- generate p p > w+ w- j j QCD<=2
  - ➡ 76 processes
  - ➡ 5332 diagrams

- generate p p > w+ w- j j QCD<=4
  - ➡ 76 processes
  - ➡ 5332 diagrams

# Ex III: What are those cards?

- Read the Cards and identify what they do
  - ➡ param_card
  - ➡ run_card:
- To see such cards run:
  - ➡ Generate p p > t t~
  - ➡ Output
  - ➡ Launch
    - ✦ Type enter to the first question
    - ✦ Now you can type 1 or 2 to see the files

# Exercise III: Cards Meaning

- How do you change
  - ➡ top mass
  - ➡ top width
  - ➡ W mass
  - ➡ beam energy
  - ➡ pt cut on the lepton

# Ex III: What are those cards? (Solution)

- Read the Cards and identify what they do
  - ➡ param_card: model parameters
    - ✦ Note aS is not typically not read from the param_card but from the PDF set chosen (if any)
  - ➡ run_card: beam/run parameters and cuts
    - ✦ https://answers.launchpad.net/madgraph5/+faq/2014

# Exercise II: Cards Meaning (Solution)

- How do you change
  - ➡ top mass
    - ✦ Set mt 180 # or edit param_card
  - ➡ top width
    - ✦ Set wt 2.1 # or edit param_card
  - ➡ W mass
    - ✦ Set mZ 80 # or change GF/aEW !! MW is not free!
  - ➡ beam energy
    - ✦ set ebeam 7000 # or change run_card
  - ➡ pt cut on the lepton
    - ✦ set ptl 20 # or change run_card

# Ex. IV: Syntax

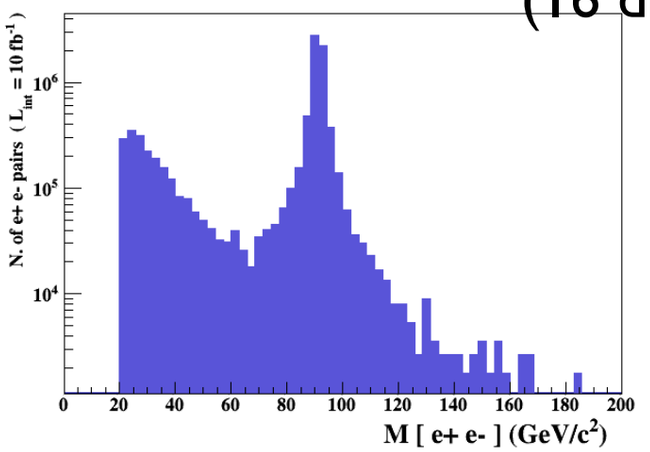
- Generate the cross-section and the distribution (invariant mass) for
  - ➡ p p > e+ e-
  - ➡ p p > z , z > e+ e-
  - ➡ p p > z > e+ e-
  - ➡ p p > e+ e- $ z
  - ➡ p p > e+ e- / z

Hint :To plot automatically distributions:
mg5> install MadAnalysis5

- Use the invariant mass distribution to determine the meaning of each syntax.

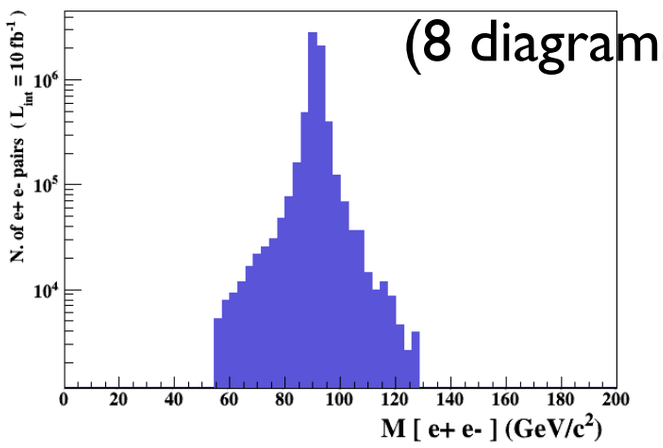
# p p > e+ e-

(16 diagrams)



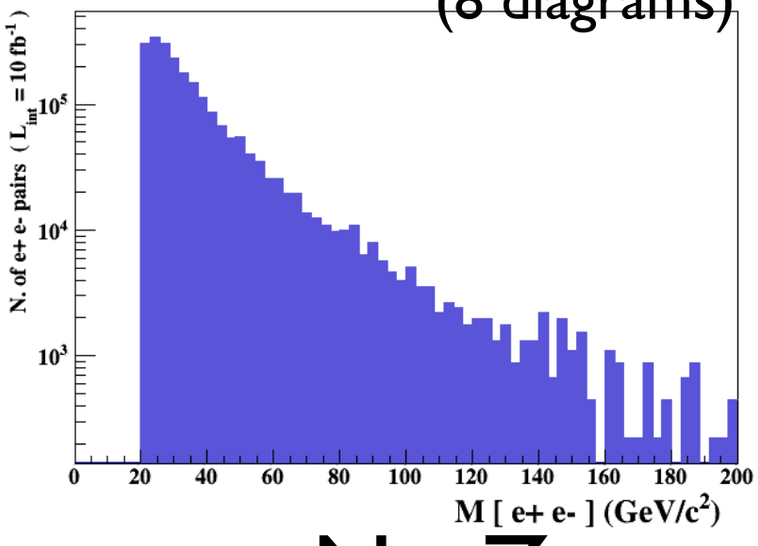

# p p >z , z > e+ e-

(8 diagrams)




# p p > e+ e- /z

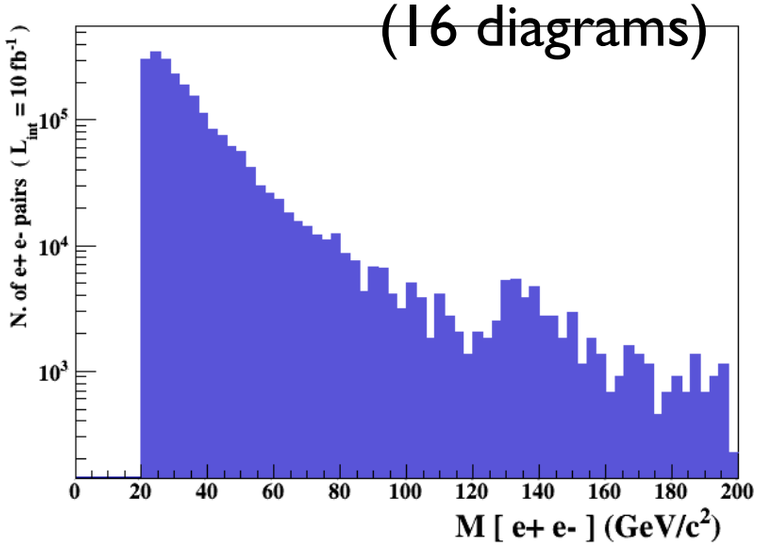
(8 diagrams)




No Z

# p p > e+ e- $ z

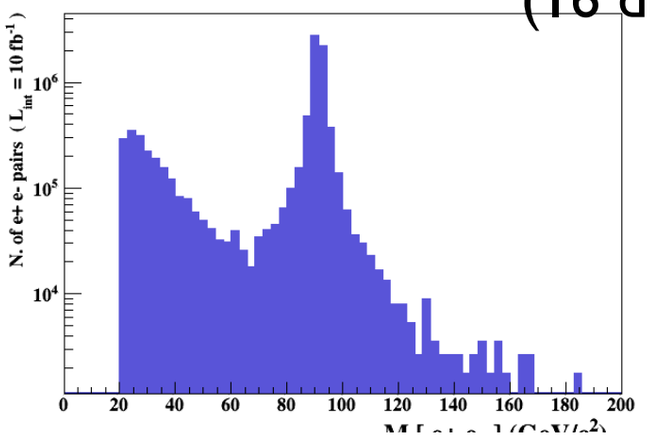
(16 diagrams)




Z- onshell veto

p p > e+ e-

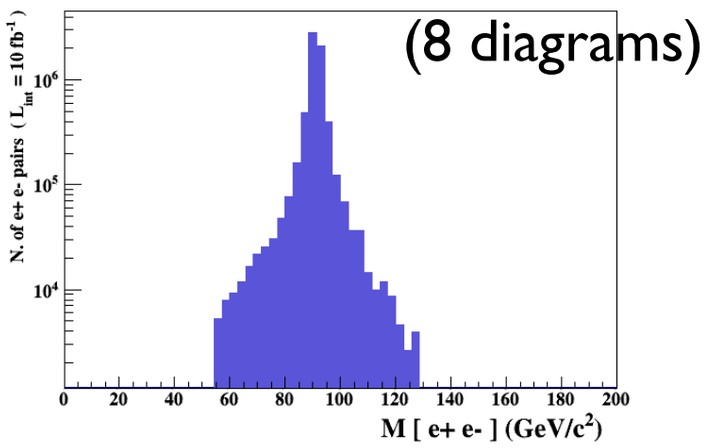
(16 diagrams)




Correct Distribution

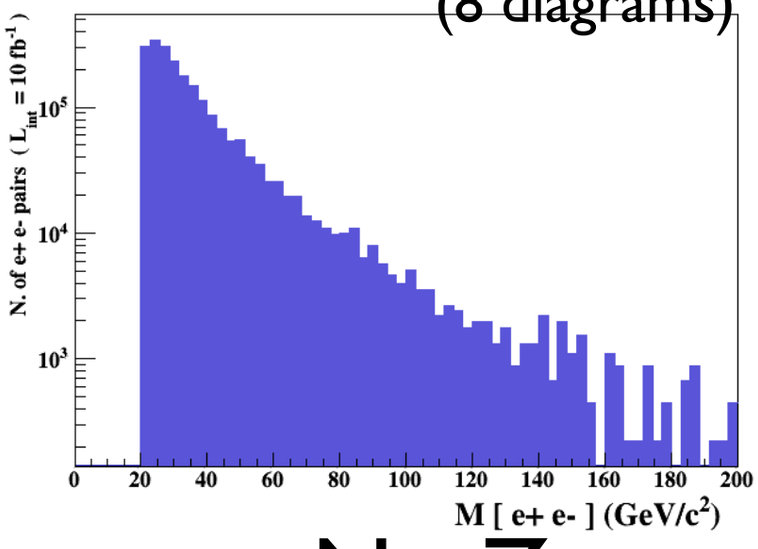
p p >z , z > e+ e-

(8 diagrams)




p p > e+ e- /z

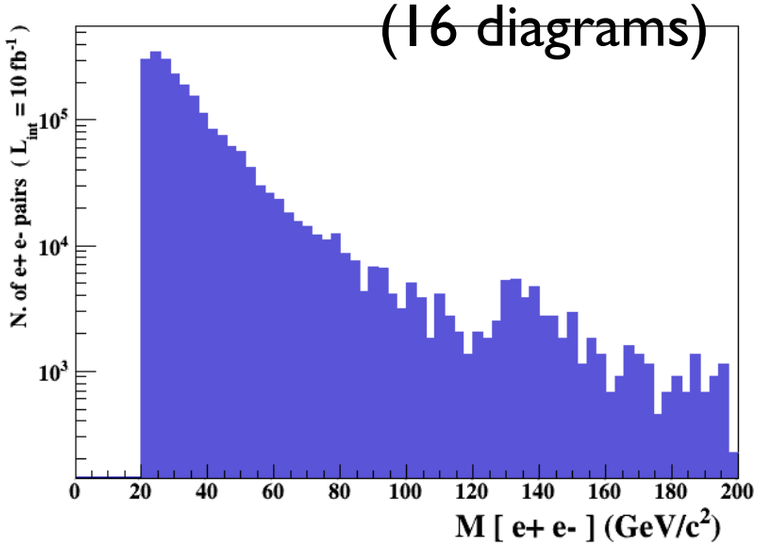
(8 diagrams)




No Z

p p > e+ e- $ z

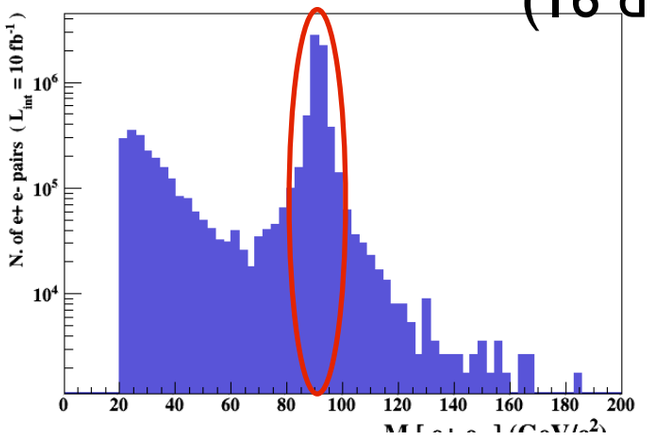
(16 diagrams)




Z- onshell veto

p p > e+ e-

(16 diagrams)

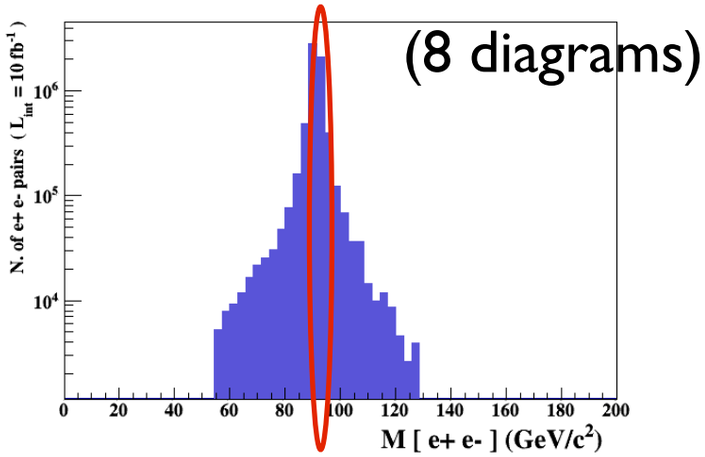



Correct Distribution

Z Peak

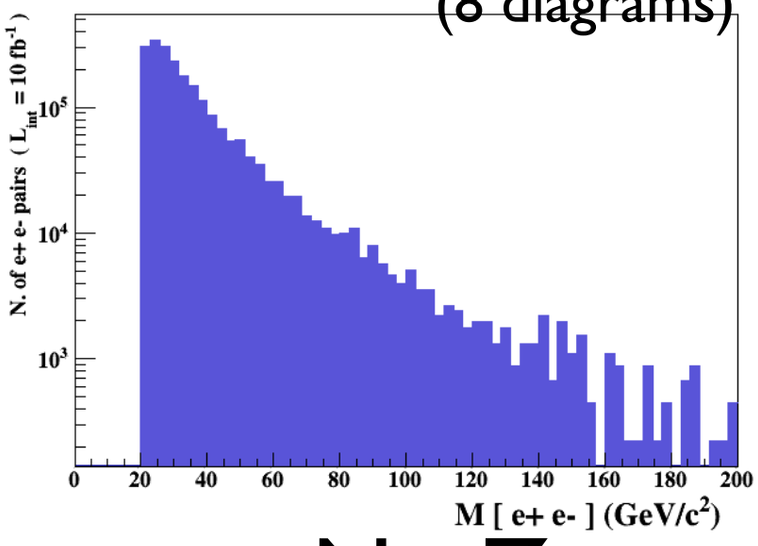
p p >z , z > e+ e-

(8 diagrams)




p p > e+ e- /z

(8 diagrams)

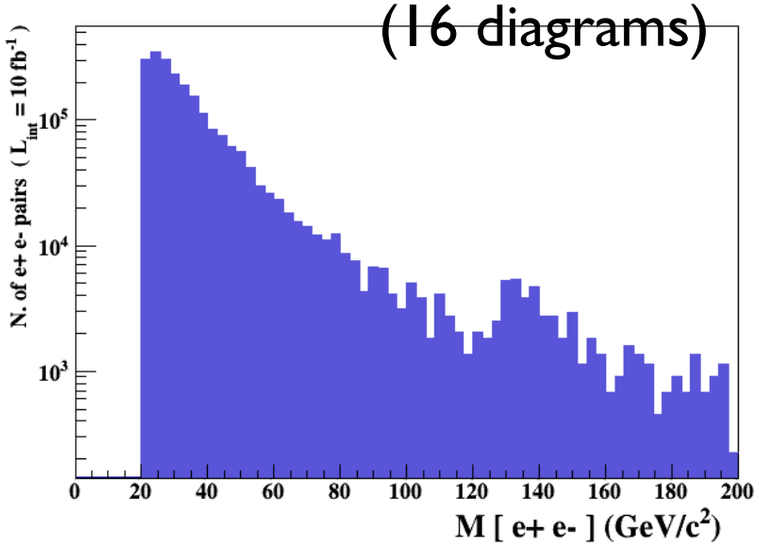



No Z

NO Z Peak

p p > e+ e- $ z

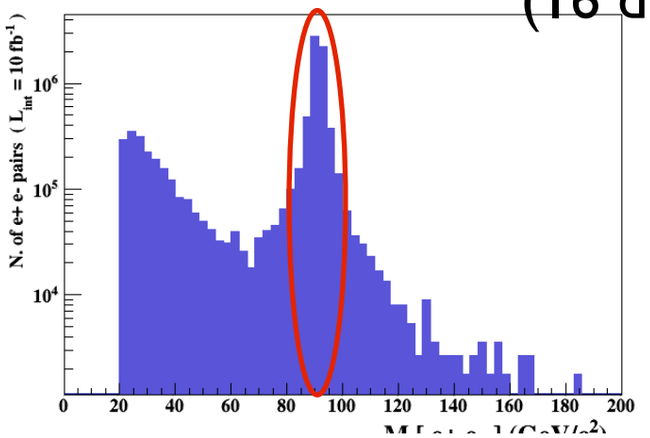
(16 diagrams)




Z- onshell veto

p p > e+ e-

(16 diagrams)

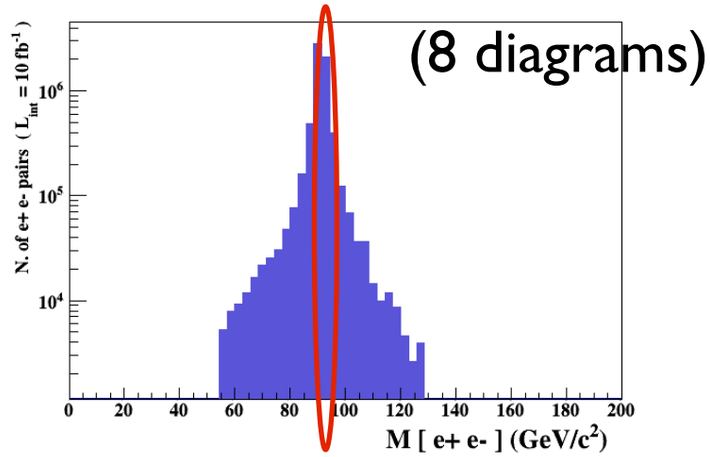



Correct Distribution

Z Peak

p p >z , z > e+ e-

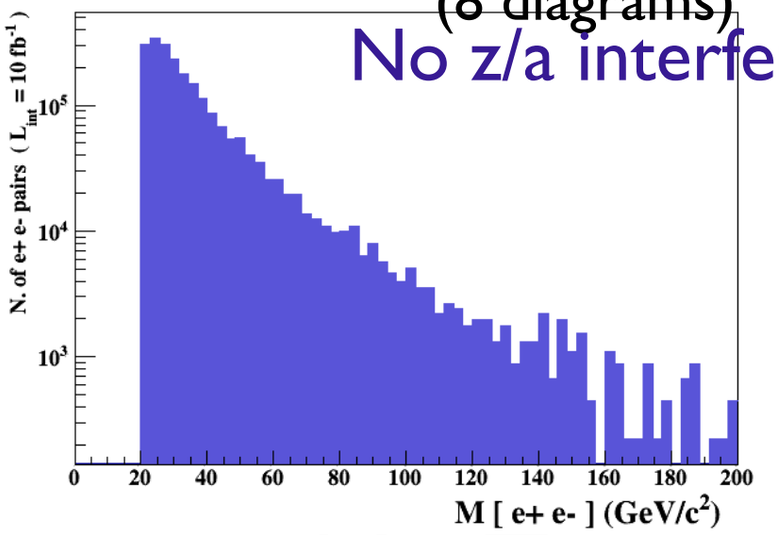
(8 diagrams)




p p > e+ e- /z

(8 diagrams)

No z/a interference



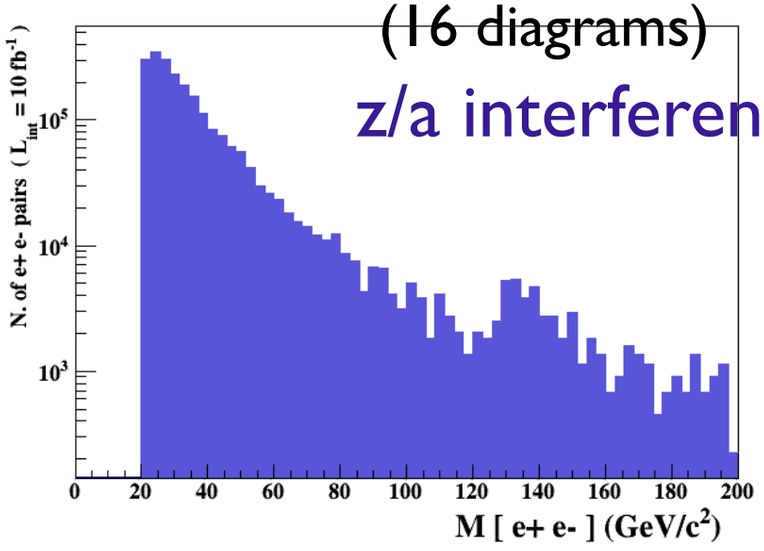

No Z

NO Z Peak

p p > e+ e- $ z

(16 diagrams)

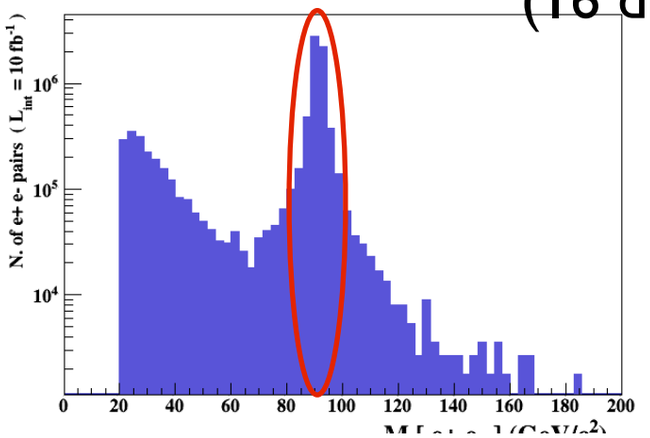
z/a interference




Z- onshell veto

p p > e+ e-

(16 diagrams)

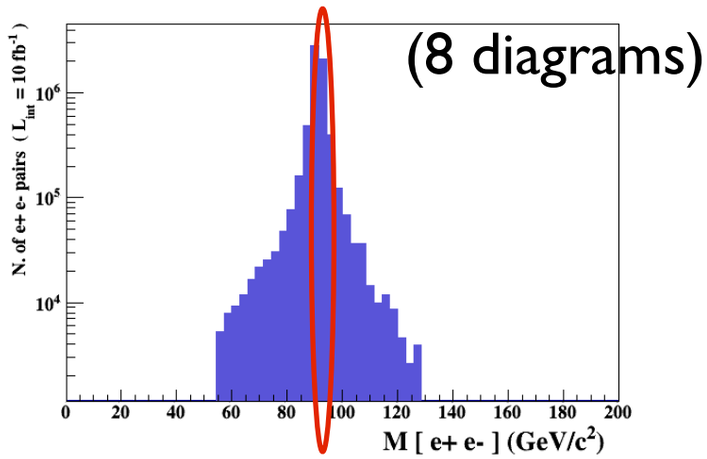



Correct Distribution

Z Peak

p p >z , z > e+ e-

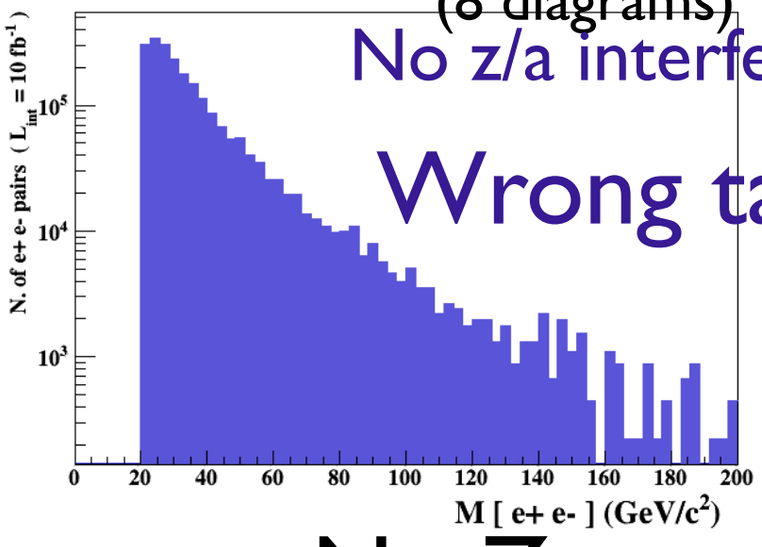
(8 diagrams)




p p > e+ e- /z

(8 diagrams)

No z/a interference




Wrong tail

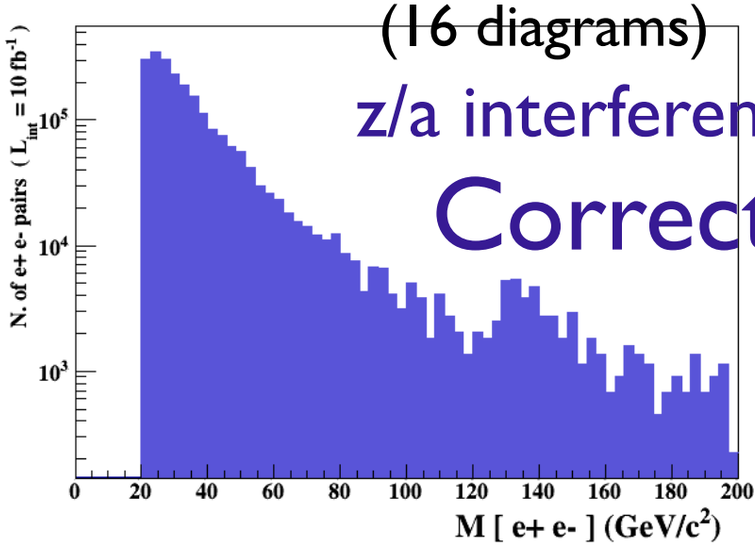
No Z

NO Z Peak

p p > e+ e- $ z

(16 diagrams)

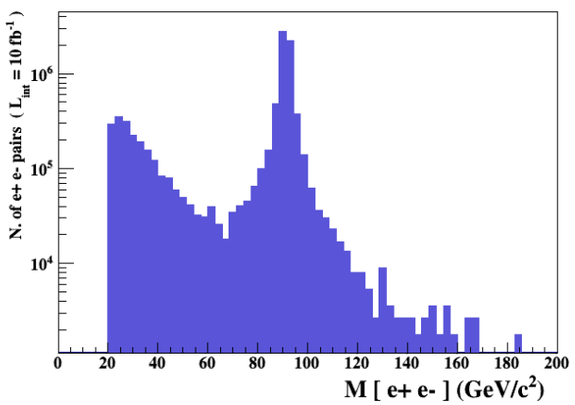
z/a interference




Correct tail

Z- onshell veto

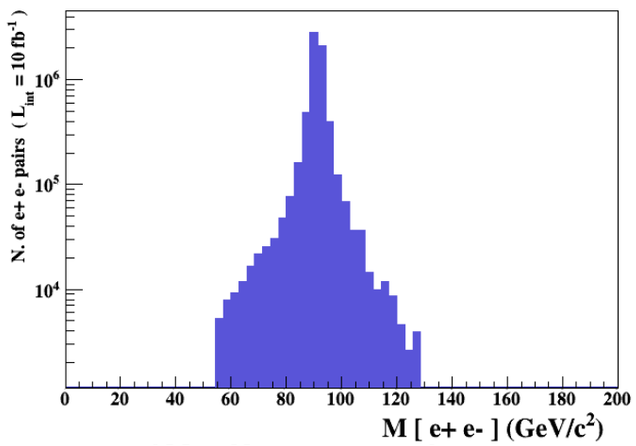
p p > e+ e-




(16 diagrams)

=

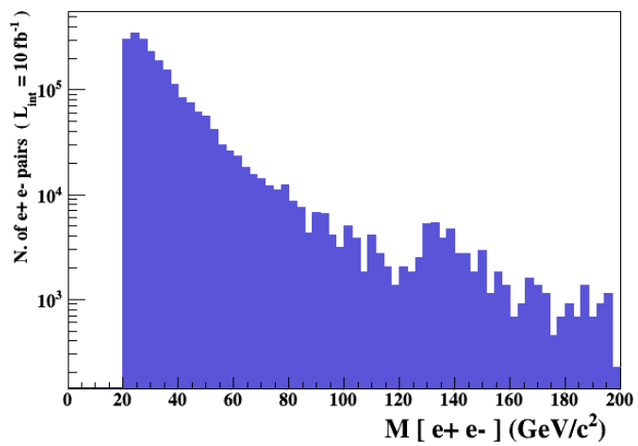
p p >z , z > e+ e-




(8 diagrams)

+

p p > e+ e- $ z




(16 diagrams)

# Onshell cut: BW_cut

$$|M^* - M| < BW_{cut} * \Gamma$$

- The Physical distribution is (very close to) exact sum of the two other one.
- The "$" forbids the Z to be onshell but the photon invariant mass can be at MZ (i.e. on shell substraction).
- The "/" is to be avoid if possible since this leads to violation of gauge invariance.

# WARNING

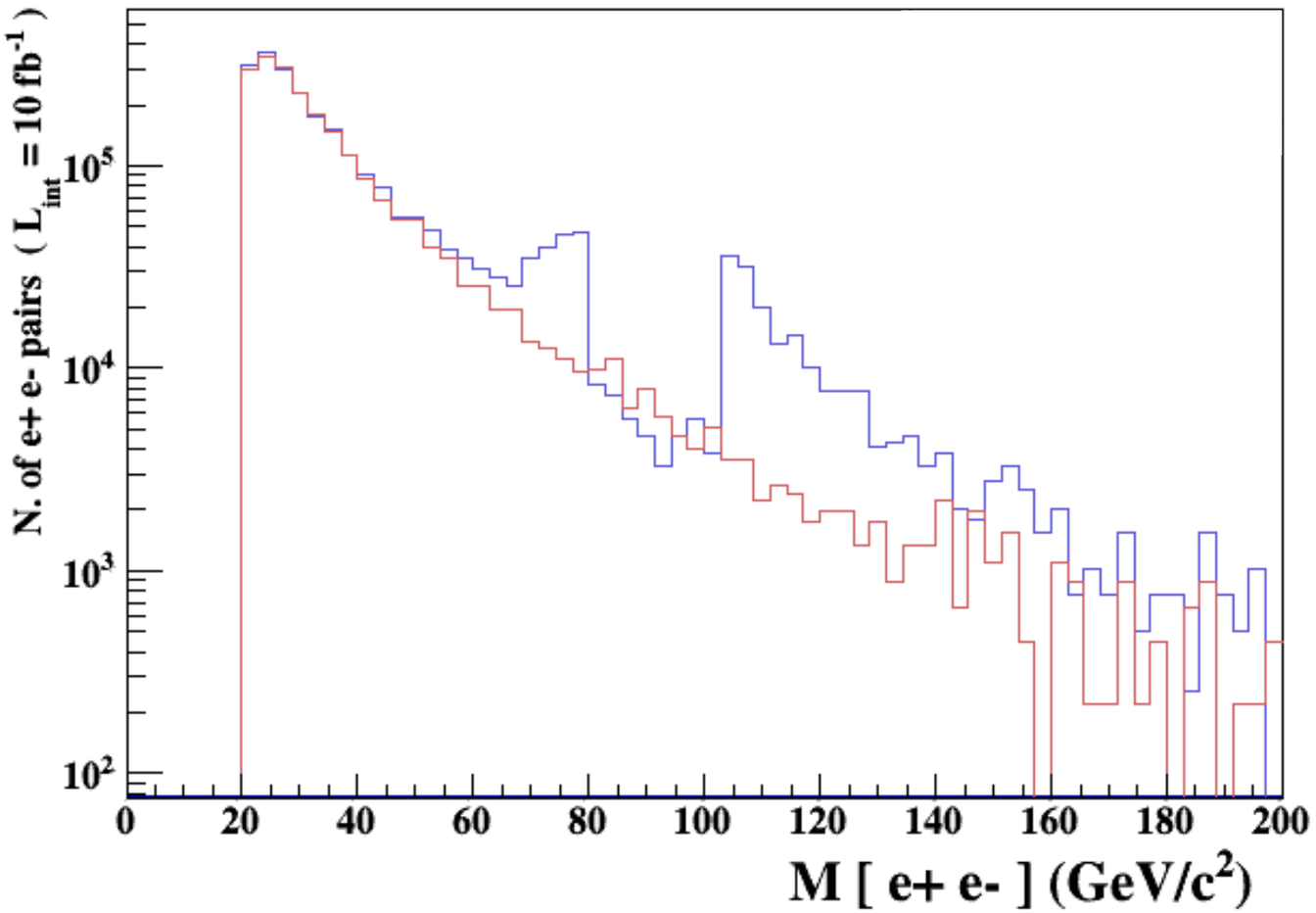
- NEXT SLIDE is generated with bw_cut =5
- This is TOO SMALL to have a physical meaning (15 the default value used in previous plot is better)
- This was done to illustrate more in detail how the "$" syntax works.

# See previous slide warning

## p p > e+ e- / Z

(red curve)

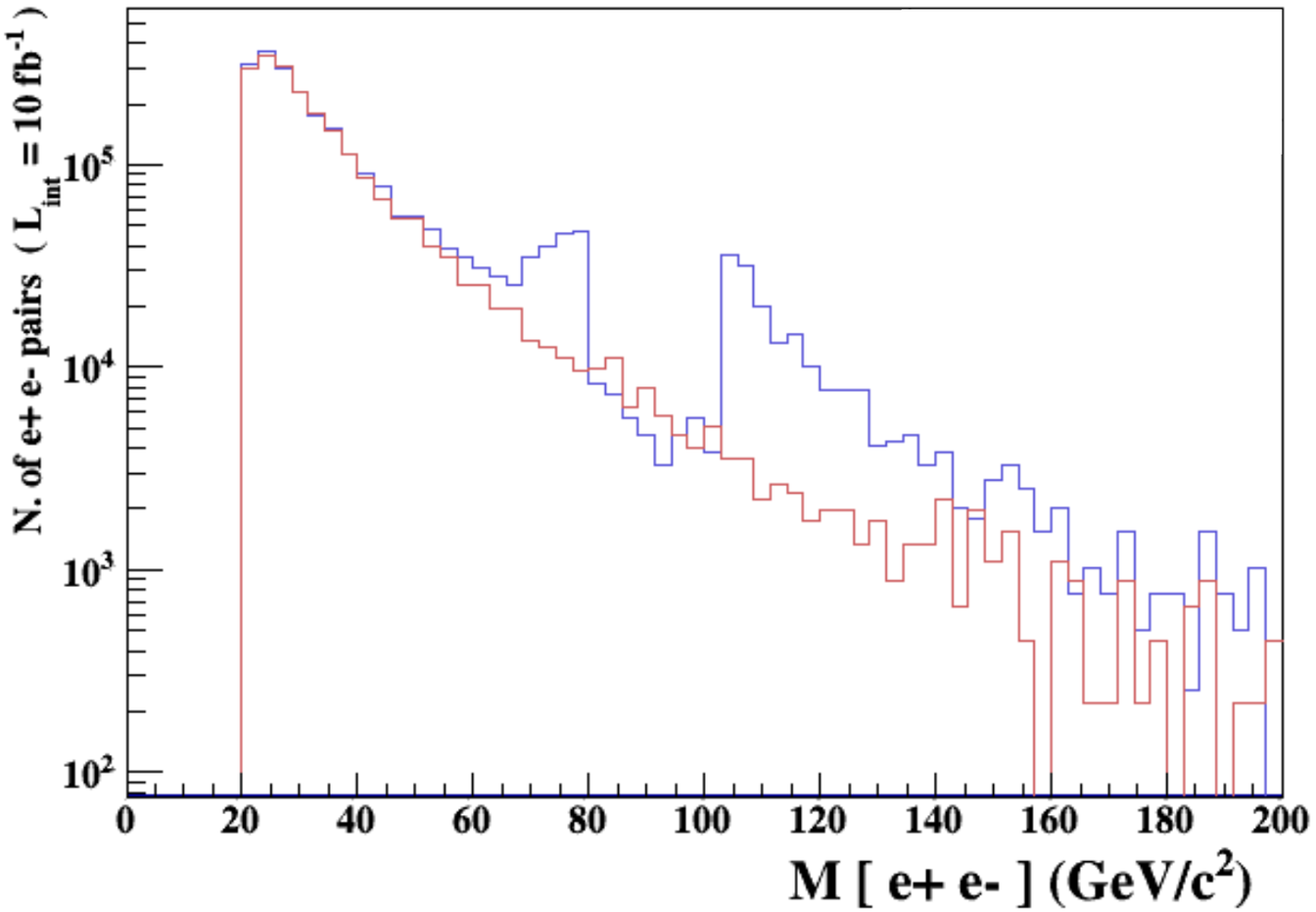
(blue curve)



# See previous slide warning

p p > e+ e- / Z (red curve)

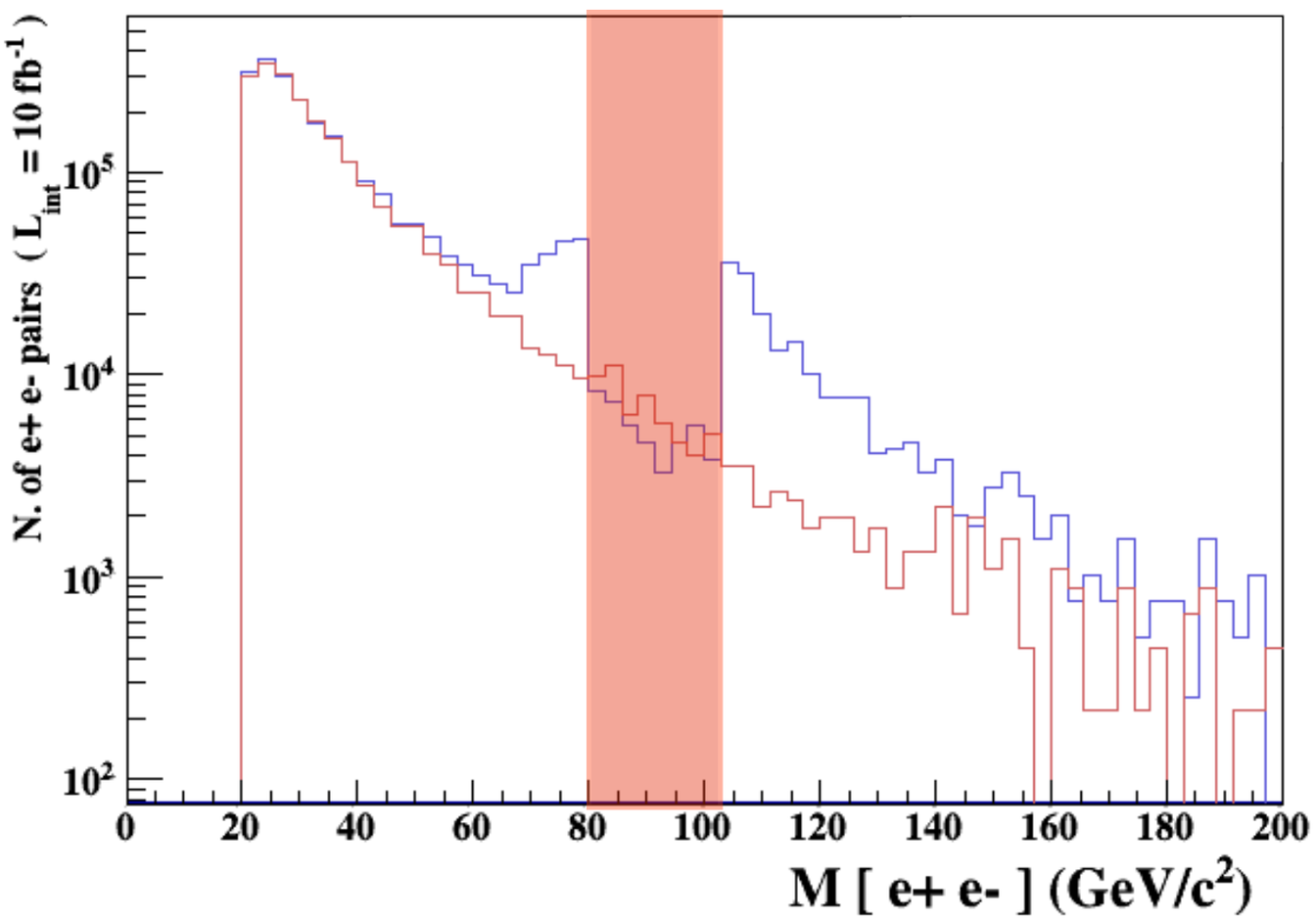
adding p p > e+ e- $ Z (blue curve)



See previous slide warning

p p > e+ e- / Z (red curve)

adding p p > e+ e- $ Z (blue curve)



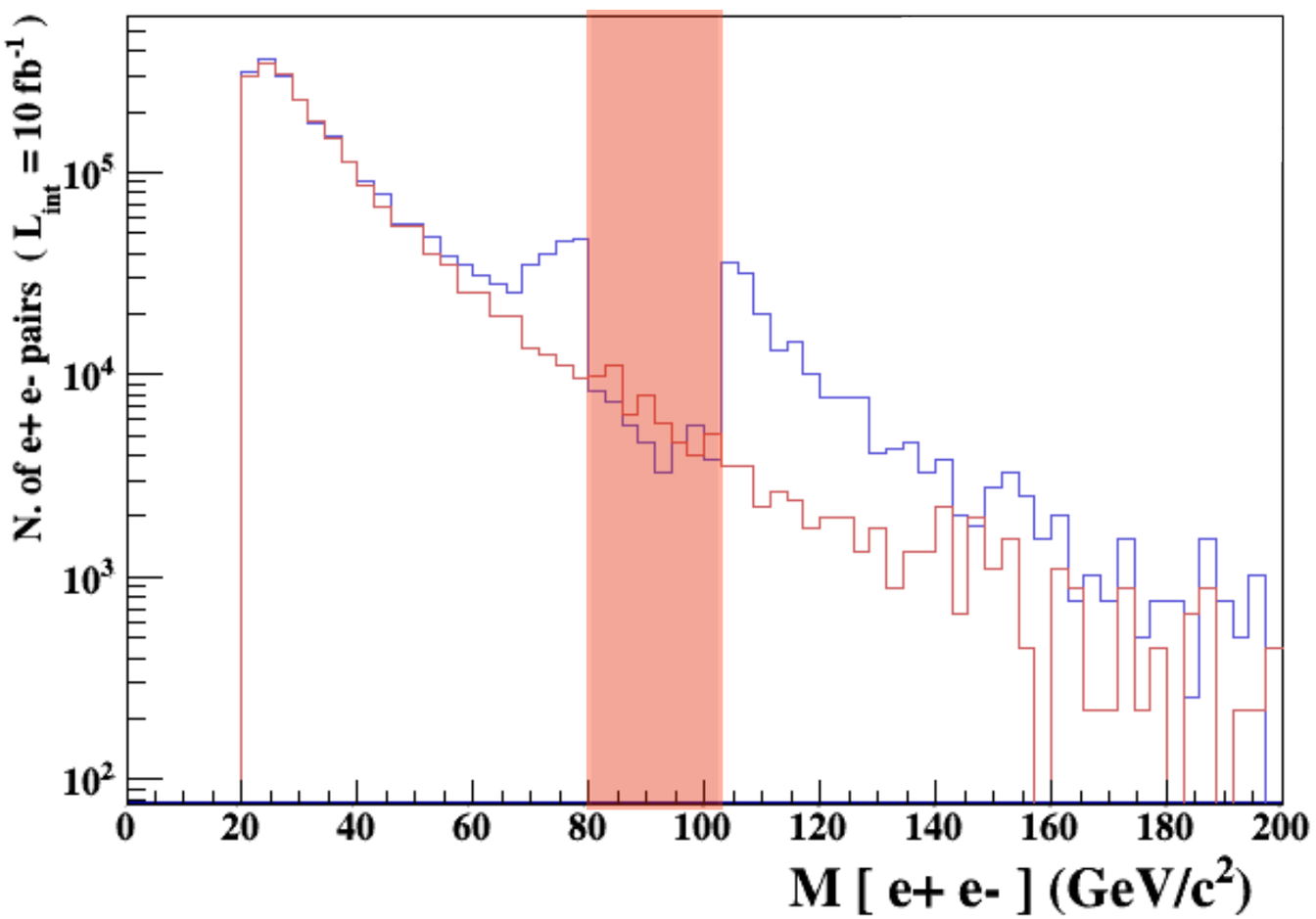

- Z onshell veto

5 times width area

See previous slide warning

p p > e+ e- / Z (red curve)

adding p p > e+ e- $ Z (blue curve)



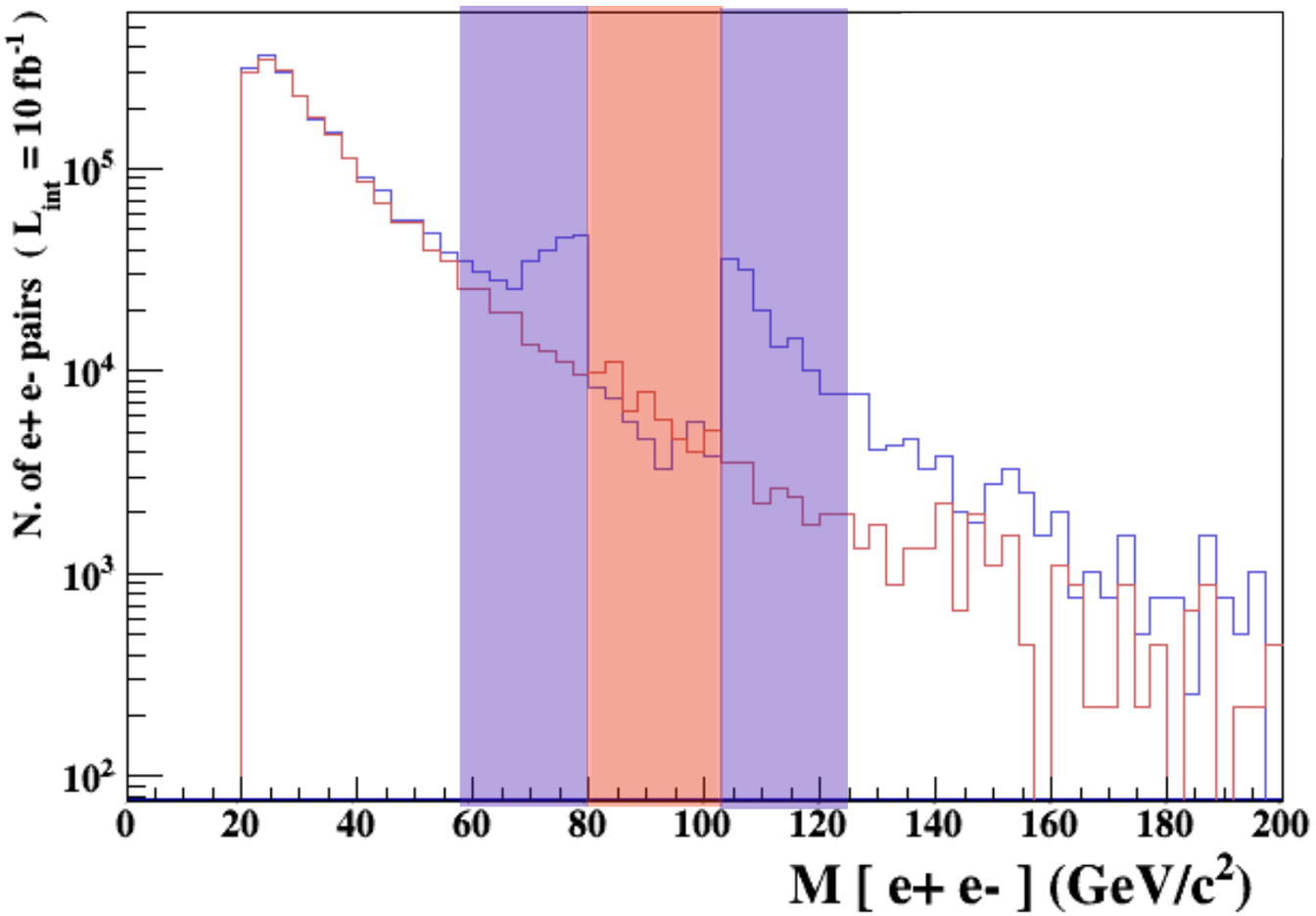

- Z onshell veto
- In veto area only photon contribution

5 times width area

See previous slide warning

p p > e+ e- / Z (red curve)

adding p p > e+ e- $ Z (blue curve)




- Z onshell veto
- In veto area only photon contribution
- area sensitive to z-peak

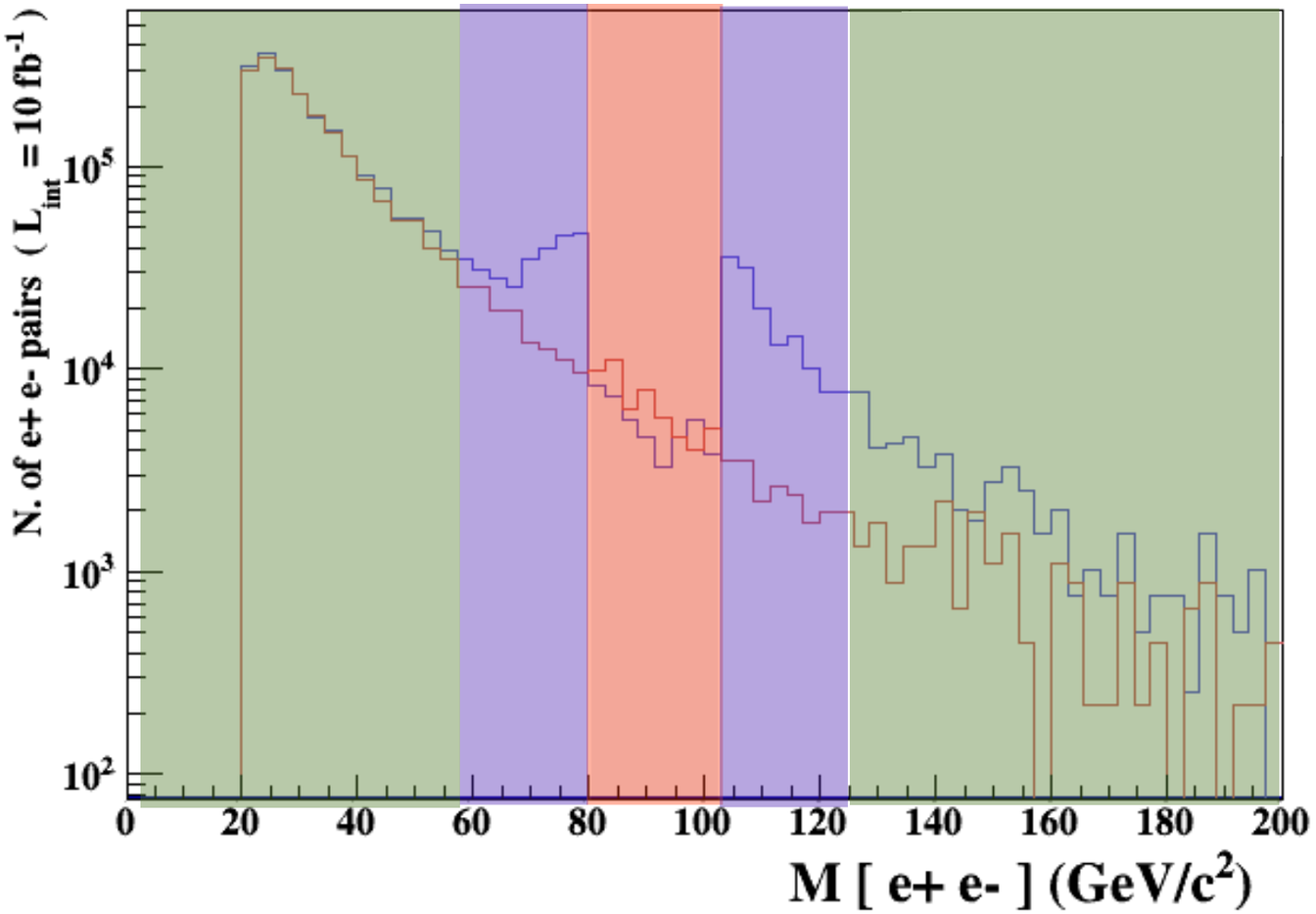
5 times width area

15 times width area

# See previous slide warning

p p > e+ e- / Z (red curve)

adding p p > e+ e- $ Z (blue curve)




- Z onshell veto
- In veto area only photon contribution
- area sensitive to z-peak
- very off-shell Z, the difference between the curve is due to interference which are need to be KEPT in simulation.

5 times width area

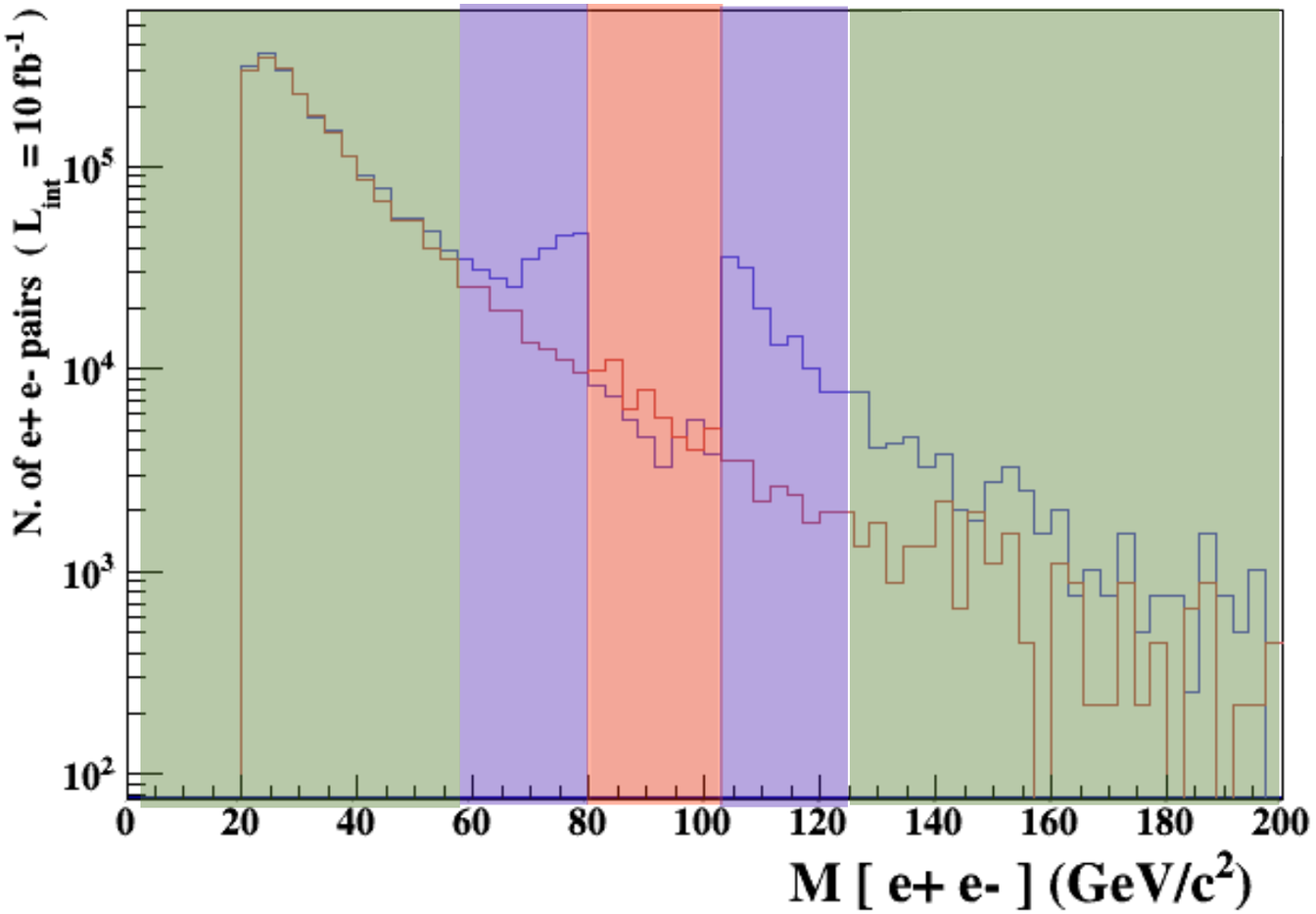
15 times width area

>15 times width area

See previous slide warning

p p > e+ e- / Z (red curve)

adding p p > e+ e- $ Z (blue curve)




- Z onshell veto
- In veto area only photon contribution
- area sensitive to z-peak
- very off-shell Z, the difference between the curve is due to interference which are need to be KEPT in simulation.

5 times width area

15 times width area

>15 times width area

The "$" can be use to split the sample in BG/SG area

- Syntax Like
  - ➡ p p > z > e+ e- (ask one S-channel z)
  - ➡ p p > e+ e- / z (forbids any z)
  - ➡ p p > e+ e- $$ z (forbids any z in s-channel)
- ARE NOT GAUGE INVARIANT !
- forgets diagram interference.
- can provides un-physical distributions.

- Syntax Like
  - ➡ p p > z > e+ e- (ask one S-channel z)
  - ➡ p p > e+ e- / z (forbids any z)
  - ➡ p p > e+ e- $$ z (forbids any z in s-channel)
- ARE NOT GAUGE INVARIANT !
- forgets diagram interference.
- can provides un-physical distributions.

# Avoid Those as much as possible!

- Syntax Like
  - ➡ p p > z > e+ e- (ask one S-channel z)
  - ➡ p p > e+ e- / z (forbids any z)
  - ➡ p p > e+ e- $$ z (forbids any z in s-channel)
- ARE NOT GAUGE INVARIANT !
- forgets diagram interference.
- can provides un-physical distributions.

# Avoid Those as much as possible!

check physical meaning and gauge/Lorentz invariance if you do.

- Syntax like
  - p p > z, z > e+ e-  (on-shell z decaying)
  - p p > e+ e- $ z  (forbids s-channel z to be on-shell)
- Are linked to cut $|M^* - M| < BW_{cut} * \Gamma$
- Are more safer to use
- Prefer those syntax to the previous slides one

# Exercise V

- Generate top pair production at LO,
- Do the fully leptonic decay of the top pair
- Shower event with pythia8
- Plot the pt distribution of the first jet

- How to improve the simulation
  - Of the cross-section
  - Of the pt of the first/second jet
  - …

# Two methods for the decay

- Generate p p > t t~, (t > w+ b, w+ > e+ ve), (t~ > w- b~, w-> e- ve~)
- output
- launch
  - ➡ Ask for Pythia8 and MA5 (rest keep on OFF)
  - ➡ set mpi OFF # This is for speed issue for the tuto

---

- Generate p p > t t~
- Output; Launch
  - ➡ Ask for MadSpin and Pythia8 and MA5
  - ➡ set mpi OFF # This is for speed issue for the tuto
  - ➡ decay t > w+ b, w+ > e+ ve
  - ➡ decay t~ > w- b~, w- > e- ve~

# Two methods for the decay

- Generate p p > t t~, (t > w+ b, w+ > e+ ve), (t~ > w- b~, w-> e- ve~)
- Full phase-space integration
  - ➡ Does not rely on the Branching ratio
  - ➡ Rely on the full width
  - ➡ cut-off to avoid be too much off-shell

---

- Generate p p > t t~ + Madspin
  - ➡ Rely on the Branching ratio
  - ➡ Keep the full spin-correlation
  - ➡ Keep off-shell effects: cut-off to avoid  be too much off-shell

# Improve Precision

- cross-section
  - ➡ Need to go to NLO
    - ✦ No decay chain syntax (only MadSpin option)
    - ✦ generate p p > t t~ [QCD]
  - ➡ To generate events we need to know which Parton-Shower, you will use!!
    - ✦ Events generated for that specific PS
      - Using another will break NLO accuracy

➡ MadSpin decay is based on LO and NWA.

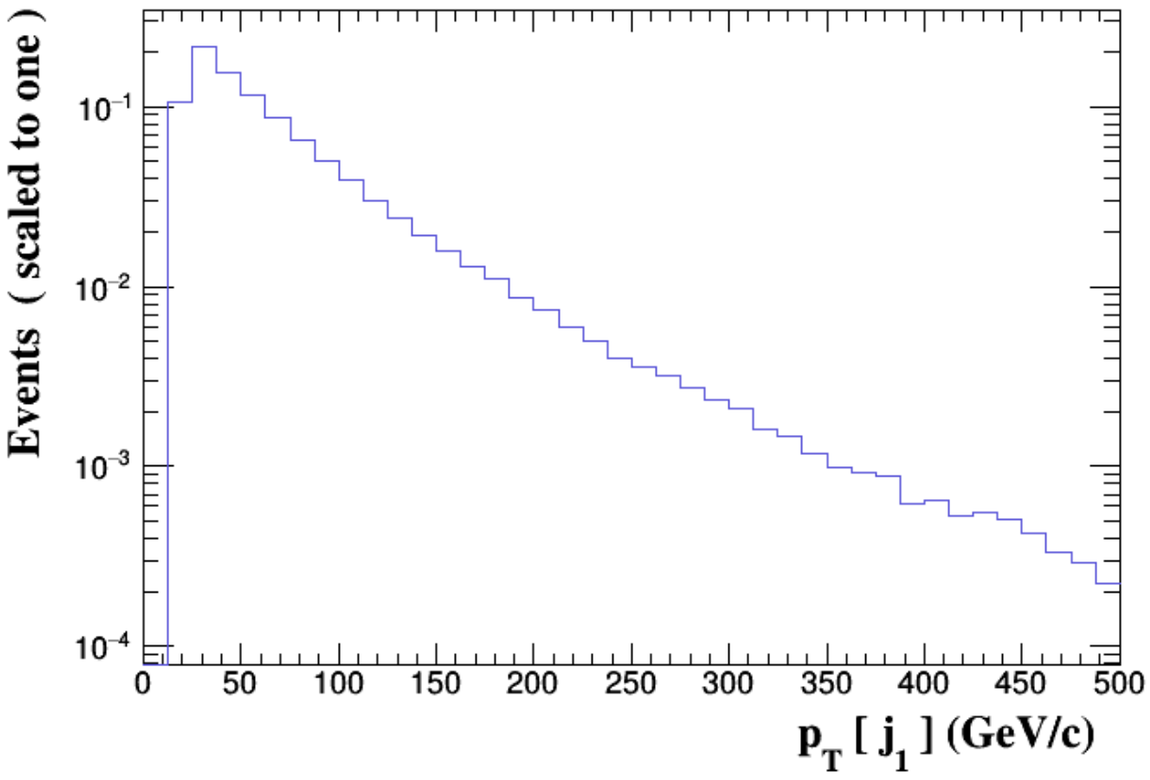
# Improve Precision

- Pt of the first jet
  - ➡ Add the jet at LO:
    - ✦ generate p p > t t~ j
      - Valid for hard jet only!
  - ➡ Going to NLO: “generate p p > t t~ [QCD]”
    - ✦ As accurate at p p > t t~ j
      - But if you do “generate p p > t t~ j [QCD]”

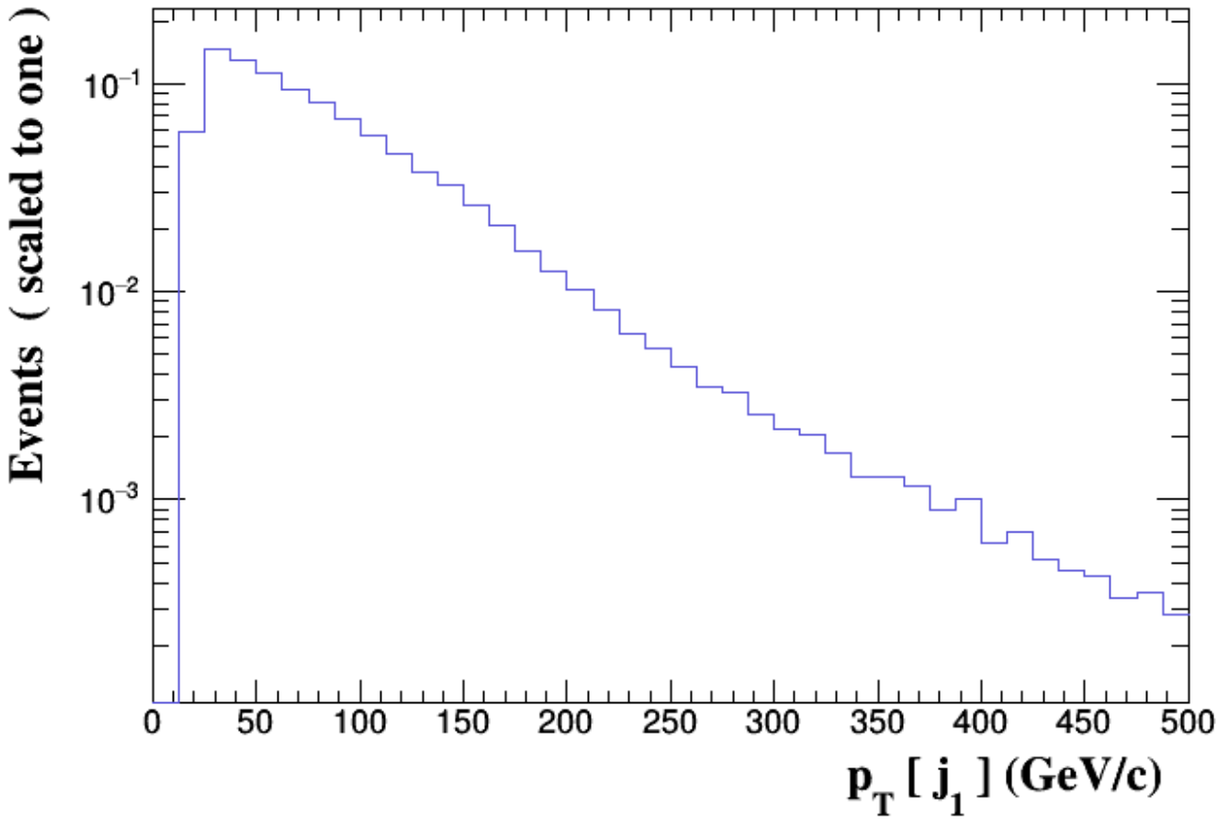
# Improve Precision

- Pt of the second jet
  - ➡ Need matching/merging method
    - ✦ generate p p > t t~
    - ✦ add process p p > t t~ j
    - ✦ add process p p > t t~ j j
  - ➡ Use MLM or CKKW-L scheme (or any variation)
  - ➡ You can also use matching/merging at NLO
    - ✦ FxFx or UNLOPS
      - generate p p > t t~ [QCD]
      - add process p p > t t~ j [QCD]
      - add process p p > t t~ j j [QCD]

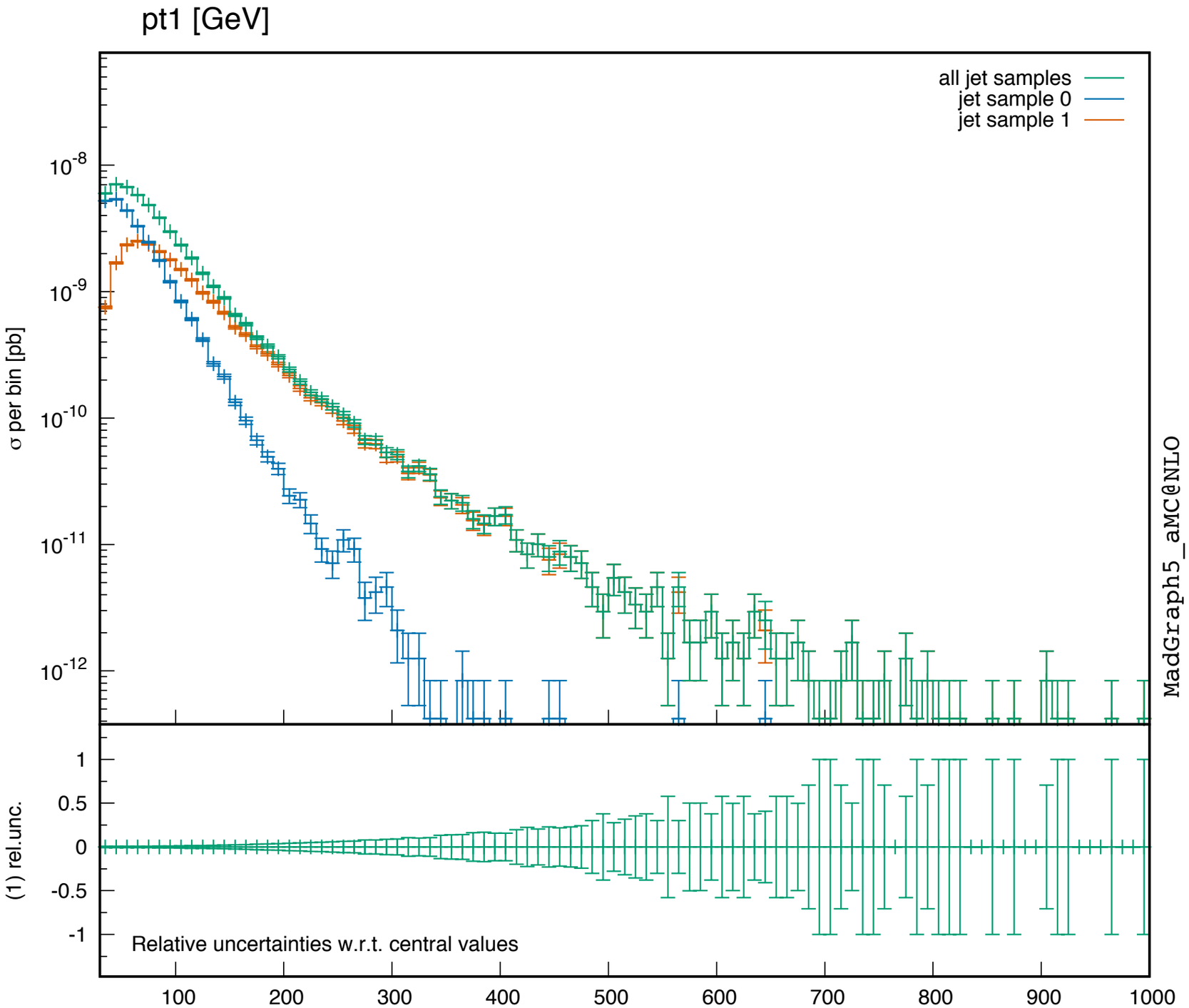
# tt@LO


Events ( scaled to one )
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
0
50
100
150
200
250
300
350
400
450
500
$p_T$ [ $j_1$ ] (GeV/c)

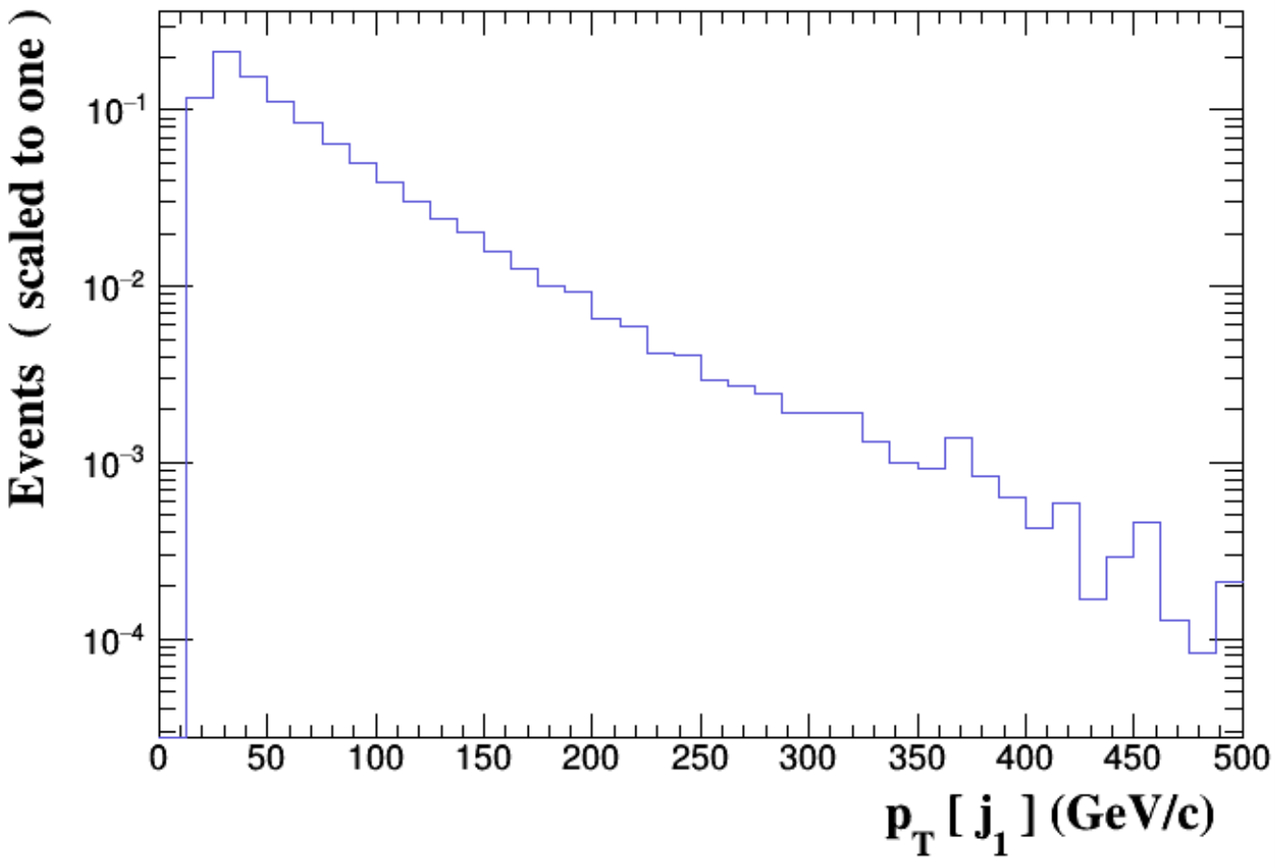
# ttj@LO


Events ( scaled to one )
$10^{-1}$
$10^{-2}$
$10^{-3}$
0
50
100
150
200
250
300
350
400
450
500
$p_T$ [ $j_1$ ] (GeV/c)

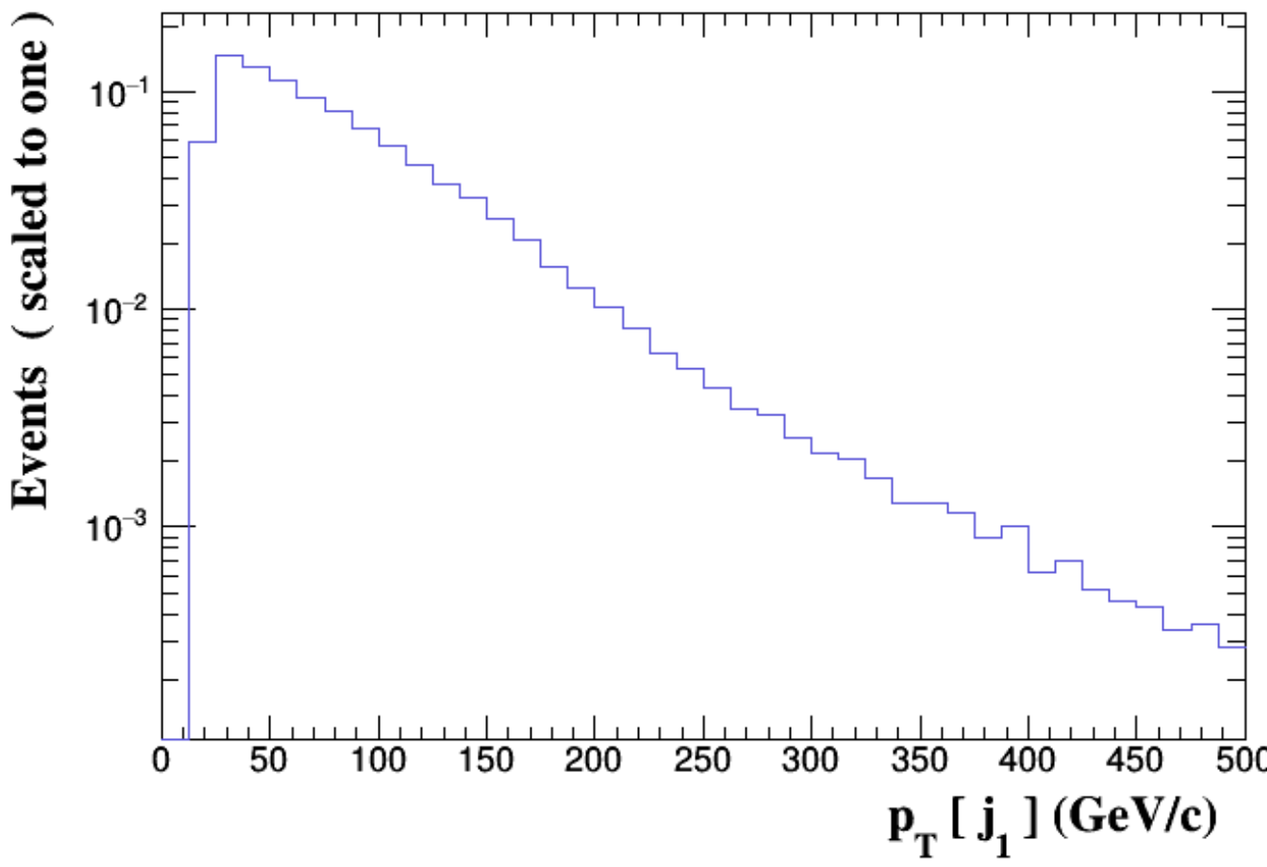
# PT distribution (MLM 0+1j)



# tt@NLO


Events ( scaled to one )
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
0
50
100
150
200
250
300
350
400
450
500
$p_T$ [ $j_1$ ] (GeV/c)

# ttj


Events ( scaled to one )
$10^{-1}$
$10^{-2}$
$10^{-3}$
0
50
100
150
200
250
300
350
400
450
500
$p_T$ [ $j_1$ ] (GeV/c)

# MG5aMC tutorial II

# BSM

Eleni Vryonodou, Cen Zhang, Ilaria Brivio
And O. Mattelaer

# Exercise I:  Restrict Model

- Run the "export command" in your shell!
- import model EWDim6
  - ➡ This downloads it on disk. (and change model to that one for the diagram generation)
  - ➡ This model contains 8 dimension operator

- We want to **RESTRICT** the model to only keep one (Owww)
- Such that Feynman diagram corresponding to other operator are **NOT** generated
  - Makes more optimal code !

# Exercise I: Restrict Model

- Go to models/EWdim6 directory
- Run the script
  - ➡ Python write_param_card.py
- cp param_card.dat restrict_owww.dat
  - ➡ The owww part can be changed to ANY string you want [but default and full].
- Edit that file
  - ➡ Put the c mass and b mass to zero
  - ➡ Put all the dim6 operator at 0 but CWWWL2
  - ➡ Put CWWWL2 to 9.999999e-1
- Go back to MG5_aMC
  - ➡ Import model EWdim6-owww

# Restrict Model

- When importing the model with the flag

```
MG5_aMC>import model EWdim6-owww
INFO: model loaded from PYTHONPATH: /Users/omattelaer/Desktop/UFOMODEL/EWdim6
INFO: Restrict model EWdim6 with file ../../../Desktop/UFOMODEL/EWdim6/restrict_owww.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- ve~ vm~ vt~ e+ t t~ z
MG5_aMC>
```

➡ MG5 mode pass to 5 flavour

➡ Less Feynman diagram generated

```
###################################
## INFORMATION FOR DIM6
###################################
Block dim6
    1 1.000000e+00 # CWWWL2

###################################
## INFORMATION FOR MASS
###################################
Block mass
    6 1.720000e+02 # MT
   13 1.056600e-01 # MM
   15 1.777000e+00 # MTA
   23 9.118760e+01 # MZ
   25 1.250000e+02 # set of param :1*MH, 1*MP
```

➡ Less parameter in the param_card

- ✦ No b/c mass option
- ✦ One Dim6 operator
- ✦ No CKM block

# Restrict Model

- What's happening

1. All coupling are evaluated for that param_card
2. All vertex associated to zero coupling (exactly or very small) are **removed** from the model
3. All zero/one value of the param_card are frozen to such value (use 0.000001e-99,9.999999e-1 to avoid that)
4. If two parameters are equal (or opposite) in the same block
   - ✦ Remove one of the two parameters
   - ✦ Freeze the second one accordingly
5. If a file default_XXX.dat exists use that one as default param_card. Otherwise use the restrict_XXX.dat itself
   - ✦ can be used for benchmark
6. restrict_default.dat is automatically loaded by MG5aMC
   - ✦ Use import model EWdim6-full to bypass it

# Exercise II: Validate Model

- Validate a Model/Process is always nice !!
  - ➡ You will sound like a MG5 expert
- Import model EW-dim6
- check p p > z h a

```
Lorentz invariance results:
Process          Min element       Max element       Relative diff.    Result
g g > z h a      3.0245789272e-01  3.0245789272e-01  0.0000000000e+00  Passed
u u~ > z h a     4.1915242516e-03  4.1915242516e-03  2.0693229620e-15  Passed
d d~ > z h a     1.2414404109e-03  1.2414404109e-03  2.6200262928e-15  Passed
Summary: 3/3 passed, 0/3 failed
Not checked processes: c c~ > z h a, s s~ > z h a
Gauge results:
Process          matrix            BRS               ratio             Result
g g > z h a      3.4921781373e-01  4.9684750757e-42  1.4227438809e-41  Passed
u u~ > z h a     4.9543423043e-03  8.8574527892e-34  1.7878160703e-31  Passed
d d~ > z h a     2.8216312492e-03  2.0405124807e-34  7.2316766455e-32  Passed
Summary: 3/3 passed, 0/3 failed
Process permutation results:
Process          Min element       Max element       Relative diff.    Result
g g > z h a      3.7207324869e-01  3.7207324869e-01  1.4919414773e-16  Passed
u u~ > z h a     1.2564293427e-02  1.2564293427e-02  2.7613546055e-16  Passed
d d~ > z h a     1.3180098875e-02  1.3180098875e-02  1.3161687879e-16  Passed
Summary: 3/3 passed, 0/3 failed
```

- Lorentz
  - ➡ Very sensitive to gauge
- Gauge
  - Epsilon replaced
- MG5 consistency
  - Change num method

# Exercise III: Width

- Compute p p > w+ w- b b~
  - ➡ Change the top quark width
  - ➡ How the cross-section changes (and why)
- compute p p > t t~, t > w+ b, t~ > w- b~
  - ➡ Change the top quark width
  - ➡ How the cross-section changes (and why)
- compute p p > t t~ + Madspin decay
  - ➡ Change the top quark width (but keep BR to 1)
  - ➡ How the cross-section changes (and why)

# Exercise III: Width

- Compute p p > w+ w- b b~
  - ➡ Cross-section as 1/ Gamma
- compute p p > t t~, t > w+ b, t~ > w- b~
  - ➡ Cross-section as 1/Gamma
- compute p p > t t~ + Madspin decay
  - ➡ Constant (use the Branching ratio information)
    - ✦ If MadSpin does not re-compute the width
- The width is consider as a free parameter in the computation.
  - ➡ Need to be provided correctly for the cross-section/ shape

# Exercise III: Width - Part II

- Compare
  - ➡ generate p p > w+ j
  - ➡ generate p p > w+ j, w+ > e+ ve
- Compare
  - ➡ generate p p > e+ ve j
  - ➡ generate p p > w+ j, w+ > e+ ve
- Redo the comparison when modifying the run_card parameter “cut_decays”

# Width Solution

Goal • understanding decay-chain handling

generate p p > w+ j
output; launch

generate p p > w+ j, w+ > e+ ve
output; launch

generate p p > e+ ve j
output; launch

| Wrong width | Correct width | +cut_decays=T |
|---|---|---|
| 21437 pb * BR<br>2304 pb | 21437 pb * BR<br>2304 pb | 21437 pb * BR<br>2304 pb |
| 32514 pb | 2329 pb | 1588 pb |
| 33095 pb | 1606 pb | 1606 pb |

Remember

- We do not use the BR information. The cross-section depends of the total width
- particle from on shell decay do not have cut by default

# Exercise III: Width - Part II

- Compare
  - ➡ generate p p > w+ j
  - ➡ generate p p > w+ j, w+ > e+ ve
- Compare
  - ➡ generate p p > e+ ve j
  - ➡ generate p p > w+ j, w+ > e+ ve

- They are not good default for cut_decays parameter. Some people expect that parameter to be True by default and some other to be False

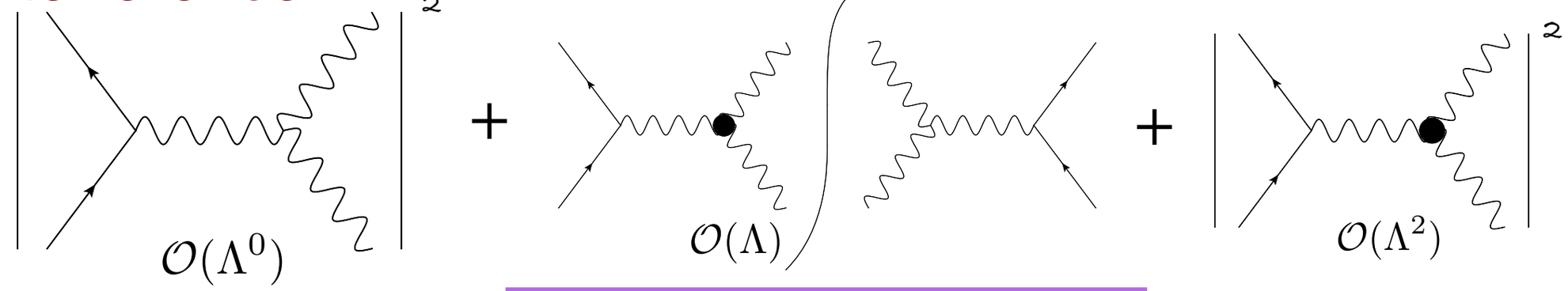
# Width: Trick

- Width are consider as free parameter
  - ➡ Not really True
- We can compute them automatically !!
  - ➡ “set wt Auto” # or inside the param_card
  - ➡ Tree-Level computation
    - ✦ Not valid for the Higgs (but for heft model)
  - ➡ Include 3 body decay (bypass them if not relevant)
- **Check it for the top/W/Z**

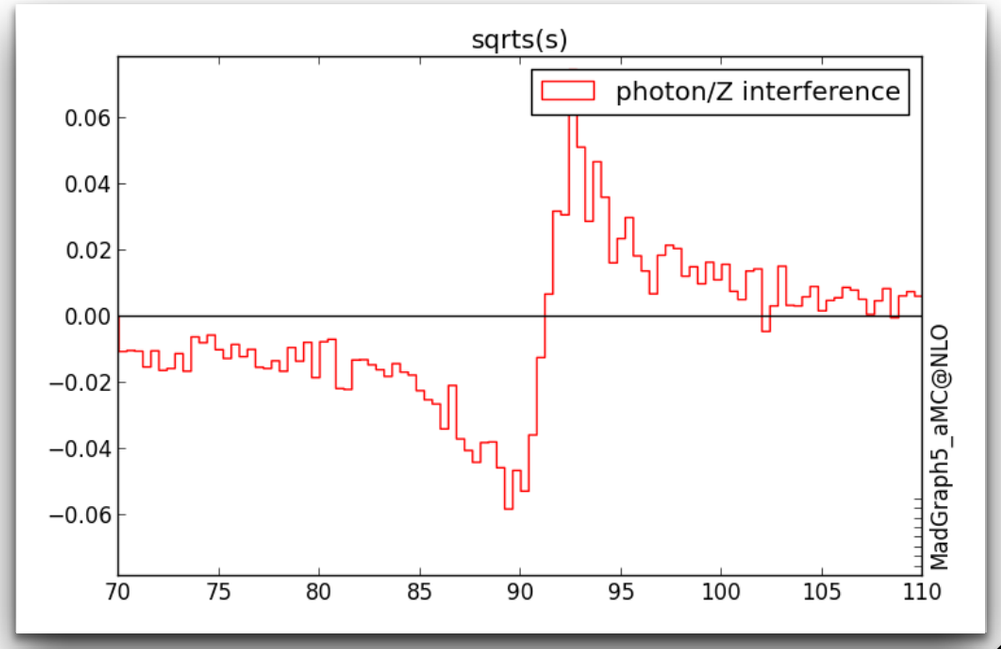
- 2 body computation can be done analytically
  - Fasten the computation (need recent UFO model)

# Exercise IV: Interference

## Interference




## Example




## Exercise

- Use your EWDIM6 model
- Compute cross-section without the square part

# Exercise V: Automation

- 2 Goals:
  - ➡ How to do a parameter scan
  - ➡ How to avoid the cli (command line interface

# Parameter scan

Parameter scan:

- compute the cross-section for a couple of mass

```
generate p p > go go
```

- for that you can enter for the go mass:

```
set mgo scan:[100,200, 300]
```

```
set mgo scan:[100*i for i in range(1,4)]
```

Any python syntax is valid!!

Width -> no problem

```
set mgo scan:[100,200, 300]
```

```
set wgo Auto
```

Recomputed for each benchmark

Output

- One additional output file scan_XX.dat

```
#run_name              mass#1000021          cross
run_01                 5.000000e+01          1.004913e+06
run_02                 1.000000e+02          5.471439e+04
run_03                 1.500000e+02          8.679740e+03
```

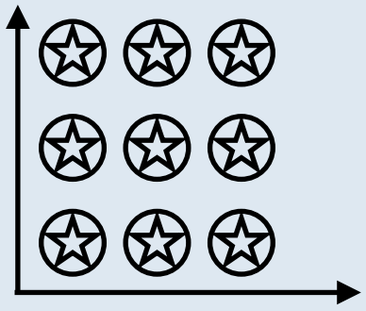
# More than one parameter

## 2D scan

- No correlation

```
set mgo scan:[100,200, 300]
```

```
set mneu1 scan:[100*i for i in range(1,4)]
```

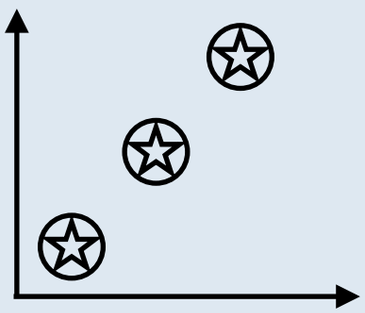


## 1D scan

- No correlation

```
set mgo scan1:[100,200, 300]
```

```
set mneu1 scan1:[100*i for i in range(1,4)]
```



## EFT operator scan

```
set dim6 1 scan1:[1 if i==0 else 0 for i in range(3)]
```

```
set dim6 2 scan1:[1 if i==1 else 0 for i in range(3)]
```

```
set dim6 3 scan1:[1 if i==2 else 0 for i in range(3)]
```

# Automation

## scripting

- write in a file (./MYFILE)
- run it as ./bin/mg5_aMC ./MYFILE

```
import model EWdim6
generate p p > z h
ouput TUTO
launch
  set nevents 5000
  set LHC 13
launch
  set LHC 14
```

## Comment on scripting

- Do not use ./bin/mg5_aMC < ./MYFILE
- If an answer to a question is not present: Default is taken automatically
- EVERYTHING that you type can be put in the entry file

# EFT related trick!

- If you specify one coupling order
  - ➡ Generate p p > t t~ QED<=2
  - ➡ All other coupling will be assume to be infinite
    - ✦ Some model restrict EFT operator to one
    - ✦ So their maximum will be one
- This can be changed with
  - ➡ set default_unset_couplings 0
    - ✦ (before the generate command)
- Useful for EFT model when they have plenty of coupling order

UCLouvain
Institut de recherche en mathématique et physique
Centre de Cosmologie, Physique des Particules et Phénoménologie

MCnet

# BSM Tutorial

Olivier Mattelaer

# Ex. IV : BSM

**Goal**

- Handling model

**Import**

- type: `import model MC4BSM`
- Automatic switch to 4/5 flavor computation according to the mass of the b quark
- MGaMC renames sm/susy particles such that their names are identical for any model. This can be avoid by `import model MC4BSM –modelname`

**Check**

- type:

```
define bsm = uv uv~ ev ev~ p1 p2
check p p > bsm bsm
```

- checks internal validity of the BSM part and consistency of the model (lorentz/gauge)

# Ex. V: Decay-Chain

Goal
- understanding decay-chain handling

Exercise
- Compare the cross-section for

```
define evdec = bsm / ev ev~
generate p p > ev ev~
output; launch
```

```
generate p p > ev ev~, ev > evdec all
output; launch
```

This is called the decay chain syntax

```
generate p p > ev > evdec all ev~
output; launch
```

- Use Automatic width computation (for all 3 cases)

```
set width wev Auto
```

To enter at the time of the edition of the cards

- Change the "cut_decays" parameter

```
set cut_decays T
```

# Decay-Chain Solution

**Goal** • understanding decay-chain handling

define bsm = bsm / ev ev~
generate p p > ev ev~
output; launch

generate p p > ev ev~, ev > bsm all
output; launch

generate p p > ev > bsm all ev~
output; launch

| Default | Correct width | +cut_decays=T |
|---|---|---|
| 19.7 pb | 19.6 pb | 19.7 pb |
| 0.1 pb | 19.3 pb | 11.8 pb |
| 0.07 pb | 11.9 pb | 11.9 pb |

**Remember**

- We do not use the BR information. The cross-section depends of the total width
- particle from on shell decay do not have cut by default

# Ex VI: Width Computation

Goal
- present the various way to compute the width

- Check with MG the width computed with FR:
  - ➡ generate uv > all all; output; launch
  - ➡ generate ev > all all; output; launch
  - ➡ generate p1 > all all; output; launch
  - ➡ generate p2 > all all; output; launch

| FR Number |
|---|
| 0.0706 GeV |
| 0.00497 GeV |
| 0 GeV |
| 0.0224 GeV |

- Compare with `compute_widths bsm`
- Why the width of uv is zero here? Function called when width on Auto

- Muv = 400 GeV  Mev = 50 GeV  λ=0.1
- m1 = 1GeV  m2 = 100GeV  m12 = 0.5 GeV

# Ex VII: Automation

Goal
- script and scan

Parameter scan:
- compute the cross-section for a couple of mass

```
generate p p > ev ev~
```

- for that you can enter for the ev mass:

```
set mev scan:[100,200, 300]
```

```
set mev scan:[100*i for i in range(1,4)]
```

Any python syntax is valid!!

scripting/ other scan:
- write in a file (./MYFILE)
- run it as ./bin/mg5_aMC ./MYFILE

```
import model MC4BSM
generate p p > ev ev~
ouput TUTO
launch
  set nevents 5000
  set LHC 13
launch
  set LHC 14
```

# Automation

Goal
- script and scan

Parameter scan:
- compute the cross-section for a couple of mass

```
generate p p > ev ev~
```

- for that you can enter for the ev mass:

```
set mev scan:[100,200, 300]
```

```
set mev scan:[100*i for i in range(1,4)]
```

Any python syntax is valid!!

Comment:
- ONLY for param_card entry!! Use scripting for other type of parameters (run_card,…)
- synchronized scan can be done via

```
set mev scan1:[100,200, 300]
set muv scan1:[200,300,400]
```

Three value will be computed!!

# Automation

scripting/ other scan:

- write in a file (./MYFILE)
- run it as ./bin/mg5_aMC ./MYFILE

```
import model MC4BSM
generate p p > ev ev~
ouput TUTO
launch
  set nevents 5000
  set LHC 13
launch
  set LHC 14
```

Comment on scripting

- Do not use ./bin/mg5_aMC < ./MYFILE
- If an answer to a question is not present: Default is taken automatically
- EVERYTHING that you type can be put in the entry file

# Exercise VIII: MadSpin

**Goal**

- Learn MadSpin for Onshell Decay

**What is MadSpin**

arXiv:1212.3460

- Program to decay on-shell particles
  - Use the NWA and the Branching-ratio
  - keep full spin-correlation
  - keep off-shell effect (up to cut-off)
  - keep unweighted event

**Exercise**

- generate all decay from ev pair production via MadSpin (and compare with decay-chain syntax)

# Exercise VIII: MadSpin

## Goal

- Learn MadSpin for Onshell Decay

## How to

```
The following switches determine which programs are run:
/-------------------------------------------------------------------------------------\
|  1. Choose the shower/hadronization program:              shower = OFF              |
|  2. Choose the detector simulation program:              detector = OFF             |
|  3. Run an analysis package on the events generated:     analysis = MADANALYSIS_5   |
|  4. Decay particles with the MadSpin module:              madspin = OFF             |
|  5. Add weights to events for different model hypothesis: reweight = OFF            |
\-------------------------------------------------------------------------------------/
```

When you see this text, type madspin=ON

- Then edit the madspin_card and include

  `decay ev > all all`

- You are done

## Note

- Also valid for NLO processes
- sometimes faster/slower than decay-chain

# Ex IX: Loop-Induced

Goal

- Learn loop-induced syntax

Ex.

- Compare Large stop limit and full loop

```
import model heft
generate g g > h
output; launch
```

```
import model sm
generate g g > h [QCD]
output; launch
```

```
import model sm-no_b_mass
generate g g > h [QCD]
output; launch
```

Note

- Interface fully identical to LO one
- No decay-chain/MadSpin allowed

# Ex 10: NLO

Goal
- Learn NLO syntax

Ex.
- Run the pair-production at NLO

```
import model MC4BSM
generate p p > ev ev~ [QCD]
output; launch
```

Note
- Interface close but different to LO one
  - different options
  - different cuts
- No decay-chain but MadSpin allowed
- Need dedicated model (not all model valid@NLO)

# NLO

```
The following switches determine which operations are executed:
 1 Perturbative order of the calculation:                              order=NLO
 2 Fixed order (no event generation and no MC@[N]LO matching):    fixed_order=OFF
 3 Shower the generated events:                                        shower=ON
 4 Decay particles with the MadSpin module:                          madspin=OFF
 5 Add weights to the events based on changing model parameters:     reweight=OFF
  Either type the switch number (1 to 5) to change its default setting,
  or set any switch explicitly (e.g. type 'order=LO' at the prompt)
  Type '0', 'auto', 'done' or just press enter when you are done.
 [0, 1, 2, 3, 4, 5, auto, done, order=LO, ... ][60s to answer]
```

order=LO / order=NLO

- Use this switch to compute K-factor with the exact same settings

fixed_order=ON / fixed_order=OFF

- if ON, we perform a pure NLO computation of the cross-section — no event generation—
- if OFF, we run NLO+PS, with the MC counter-term for a given parton shower —with event generation

# Exercise XI: Matching

1. Generate p p > w+ with 0 jets, 0,1 jets and 0,1,2 jets
   (Each on different computers - use the most powerful computer for 0,1,2 jets)
   a. Generate 20,000 events for a couple of different xqcut values.
   b. Compare the distributions (before and after Pythia) and cross sections (before and after Pythia) between the different processes, and between the different xqcut values.
   c. Summarize: How many jets do we need to simulate? What is a good xqcut value? How are the distributions affected?

# How to.

- generate the diagram with
  - ➡ generate
  - ➡ add process
- output
- launch
  - ➡ ask to run pythia
  - ➡ In run_card: put icckw=1
    - ✦ set the value for xqcut
  - ➡ In pythia_card set a value for qcut

- Qcut is the matching scale (the separation between the shower and the matrix element)
- xqcut should be strictly lower (by at least 10-15GeV) than qcut