

Your first ILC analysis: Higgs recoil mass from miniDST via root

J. List (DESY) on behalf of the IDT-WG3 Software & Computing Group

ILCDelphes & miniDST Tutorial June 23, 2021

Before we start - who are you?

- please raise your zoom hand for any category that you consider a match!

- are you:
 - an undergraduate student?
 - a post-graduate student?
 - a postdoc?
 - a senior scientist / professor?

- are you / have you been working in
 - an e^+e^- experiment?
 - a pp experiment?
 - other particle physics experiments?
 - particle physics theory?

- why are you here:
 - you plan to do an e^+e^- study for ILC / Higgs factory in general
 - have already topic in mind
 - still looking for a topic
 - just curious, but no concrete plan to do a study
 - co-organiser / convener

Before we start - technical prerequisites

- **general pre-requisites:**

- some basic knowledge of C++/root, linux shell, etc
- a linux system (or X OS) with rather recent versions of gcc, python, cmake, root

- **there are 3 possibilities to set up the environment:**

1) you have mounted /cvmfs or /afs - just **source ilcsoft setup** - done

- you have mounted /cvmfs :
 - `/cvmfs/ilc.desy.de/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh`
- or you have mounted /afs :
 - `/afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh`

2) you have installed docker - **download docker image** - also contains Delphes

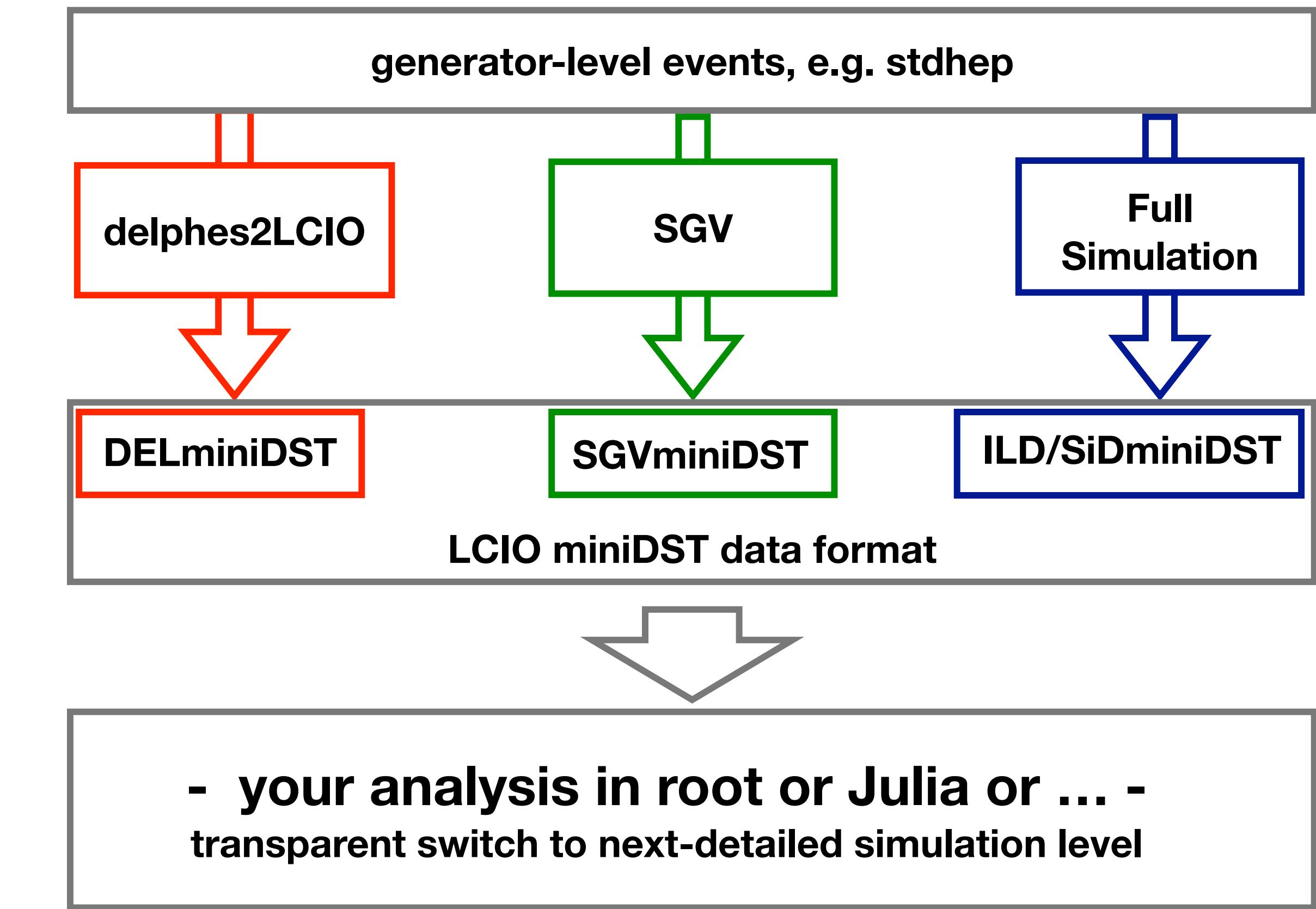
<https://github.com/EnginEren/delphes2lcio-docker>

3) you have gcc, python, cmake, root on your local computer, and **install LCIO yourself** as described here:

<https://github.com/iLCSoft/LCIO>

Parametrised, fast and full (=geant4-based) simulations

- **delphes2lcio**: an lcio application which makes Delphes (parametrised detector simulation) write out LCIO (<https://github.com/iLCSoft/LCIO/tree/master/examples/cpp/delphes2lcio>)
- **SGV**: **S**imulation a **G**rande **V**itesse (https://www.desy.de/~berggren/sgv_ug/sgv_ug.html) - detailed fast simulation from “first principles” (nearly no parametrisations!)
- **iLCSoft** (<https://github.com/iLCSoft>): software suite for full simulation and reconstruction of ILC & CLIC detectors



Parametrised, fast and full (=geant4-based) simulations

- **delphes2lcio**: an lcio application which makes Delphes (parametrised detector simulation) write out LCIO (<https://github.com/iLCSoft/LCIO/tree/master/examples/cpp/delphes2lcio>)
- **SGV**: Simulation a Grande Vitesse (https://www.desy.de/~berggren/svg_ug/svg_ug.html) - detailed fast simulation from “first principles” (nearly no parametrisations!)
- **iLCSoft** (<https://github.com/iLCSoft>): software suite for full simulation and reconstruction of ILC & CLIC detectors

**NOTE: precision e^+e^- studies often require full simulation
- see arXiv:2007.03650 for discussion of
study topics vs level of detail in simulation!**

DELminiDST

SGVminiDST

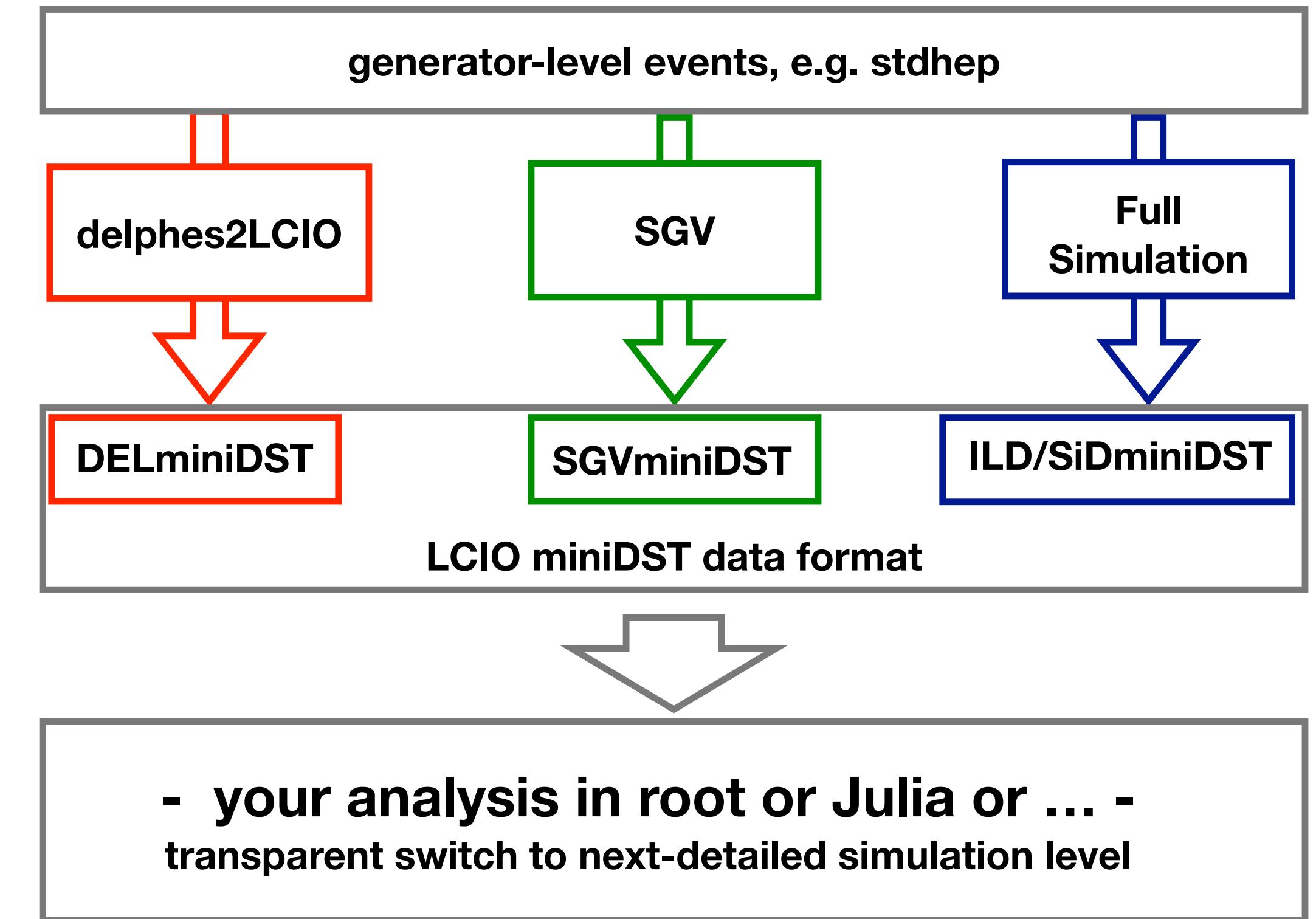
ILD/SiDminiDST

LCIO miniDST data format

- your analysis in root or Julia or ... -
transparent switch to next-detailed simulation level

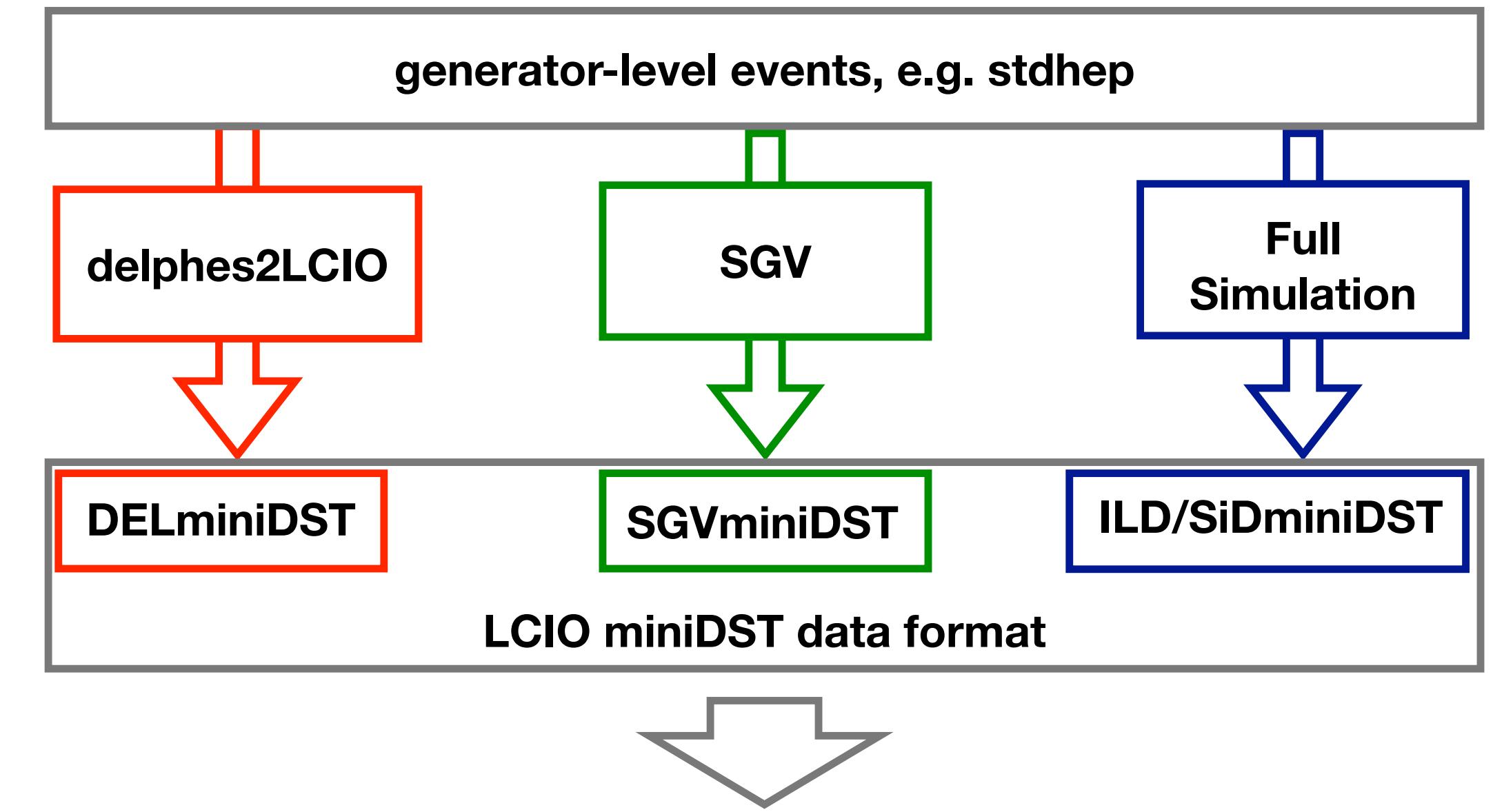
LCIO & miniDST

- **LCIO** (**L**inear **C**ollider **I/O**, part of iLCSoft, <https://github.com/iLCSoft/LCIO>):
 - event data model and persistency framework
 - implemented for C++, Fortran, Java, Go, Python
=> will see root examples today!
- **miniDST**, <https://github.com/ILDAAnaSoft/miniDST>:
 - high-level LCIO-file containing information very similar to Delphes root tree
 - can be filled from Delphes, SGV and full simulation
- **analyses based on miniDST can easily switch between parametrised, fast and full simulation!**



LCIO & miniDST

- **LCIO** (**L**inear **C**ollider **I/O**, part of iLCSoft, <https://github.com/iLCSoft/LCIO>):
 - event data model and persistency framework
 - implemented for C++, Fortran, Java, Go, Python
=> will see root examples today!
- **miniDST**, <https://github.com/ILDAAnaSoft/miniDST>:
 - high-level LCIO-file containing information very similar to Delphes root tree
 - can be filled from Delphes, SGV and full simulation
- **analyses based on miniDST can easily switch between parametrised, fast and full simulation!**



- **your analysis in root or Julia or ... -**
 transparent switch to next-detailed simulation level

**LCIO is *not* ILC-specific -
 it's used by all future e⁺e⁻ colliders
 in one way or the other!**

The miniDST format

COLLECTION NAME (SGV / ILD full sim)	COLLECTION NAME (Delphes)	COLLECTION TYPE
PandoraPF0s	PF0s	ReconstructedParticle
IsolatedElectrons	IsolatedElectrons	ReconstructedParticle
IsolatedMuons	IsolatedMuons	ReconstructedParticle
IsolatedTaus	IsolatedTaus	ReconstructedParticle
IsolatedPhotons	IsolatedPhotons	ReconstructedParticle
Refined2Jets	Durham2Jets	ReconstructedParticle
Refined3Jets	Durham3Jets	ReconstructedParticle
Refined4Jets	Durham4Jets	ReconstructedParticle
Refined5Jets	Durham5Jets	ReconstructedParticle
Refined6Jets	Durham6Jets	ReconstructedParticle
BCalPF0s	N/A	ReconstructedParticle
PrimaryVertex	N/A	LCVertex
PrimaryVertex_RP	N/A	ReconstructedParticle
MCParticlesSkimmed	MCParticles	MCParticle
MCTruthRecoLink	MCTruthRecoLink	LCRelation
RecoMCTruthLink	RecoMCTruthLink	LCRelation

full documentation of
ReconstructedParticle class

full documentation of
MCParticle class

The miniDST format

COLLECTION NAME (SGV / ILD full sim)	COLLECTION NAME (Delphes)	EXPLANATION
PandoraPF0s	PF0s	particle flow objects from the main detector, incl. event shape variables
IsolatedElectrons	IsolatedElectrons	
IsolatedMuons	IsolatedMuons	
IsolatedTaus	IsolatedTaus	
IsolatedPhotons	IsolatedPhotons	
Refined2Jets	Durham2Jets	PandoraPF0s minus "IsolatedX" forced into 2 jets (Durham algorithm, plus flavour tag)
Refined3Jets	Durham3Jets	PandoraPF0s minus "IsolatedX" forced into 3 jets (Durham algorithm, plus flavour tag)
Refined4Jets	Durham4Jets	PandoraPF0s minus "IsolatedX" forced into 4 jets (Durham algorithm, plus flavour tag)
Refined5Jets	Durham5Jets	PandoraPF0s minus "IsolatedX" forced into 5 jets (Durham algorithm, plus flavour tag)
Refined6Jets	Durham6Jets	PandoraPF0s minus "IsolatedX" forced into 6 jets (Durham algorithm, plus flavour tag)
BCalPF0s	N/A	particle flow objects from the most forward calorimeter
PrimaryVertex	N/A	
PrimaryVertex_RP	N/A	"reconstructed particle" representing the primary vertex
MCParticlesSkimmed	MCParticles	
MCTruthRecoLink	MCTruthRecoLink	links from MCParticles to PandoraPF0s
RecoMCTruthLink	RecoMCTruthLink	links from PandoraPF0s to MCParticles

Getting started

- open <https://github.com/ILDAAnaSoft/minIDST> in your browser - all the following parts of the tutorial are described there!
- use bash shell:
`bash`
- **Option 2)**
 - go to <https://github.com/EnginEren/delphes2lcio-docker> and follow instructions there
- **Option 1) & 3)**
 - download the examples:
`git clone https://github.com/ILDAAnaSoft/minIDST.git`
 - change to minIDST folder:
`cd minIDST`
 - **Option 1)**
 - source ilcsoft setup script
 - `/cvmfs/ilc.desy.de/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh` or
 - `/afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh`
 - **Option 3)**
 - set up versions of root, cmake, gcc, python etc - edit if you have different versions:
 - `setenv4LCIO.sh`
 - next: download and install LCIO => next slide

Getting started

- open <https://github.com/ILDAAnaSoft/minIDST> in your browser - all the following parts of the tutorial are described there!
- use bash shell:
`bash`
- **Option 2)**
 - go to <https://github.com/EnginEren/delphes2lcio-docker> and follow instructions there
- **Option 1) & 3)**
 - download the examples:
`git clone https://github.com/ILDAAnaSoft/minIDST.git`
 - change to minIDST folder:
`cd minIDST`
 - **Option 1)**
 - source ilcsoft setup script
 - `/cvmfs/ilc.desy.de/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh` or
 - `/afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_centos7/v02-02-02/init_ilcsoft.sh`
 - **Option 3)**
 - set up versions of root, cmake, gcc, python etc - edit if you have different versions:
 - `setenv4LCIO.sh`
 - next: download and install LCIO => next slide

Hint: replace
`x86_64_gcc82_centos7`
by the most suitable
configuration for your system!
(list the sw-directory to check
the available installations)

Option 3) Installing LCIO (incl. delphes2lcio)

- go to <https://github.com/iLCSoft/LCIO>

- go one directory up: `cd ..`

- download LCIO:

```
git clone https://github.com/iLCSoft/LCIO.git
```

- change to LCIO folder, create and change to build directory:

```
cd LCIO; mkdir build; cd build
```

- run cmake with option to build root dictionaries and C++17:

```
cmake -DBUILD_ROOTDICT=ON -D CMAKE_CXX_STANDARD=17 ..
```

- make it:

```
make -j 4 install
```

- test your installation:

```
make test
```

- set up paths etc:

```
cd ..; . ./setup.sh
```

Option 3) Installing LCIO (incl. delphes2lcio)

- go to <https://github.com/iLCSoft/LCIO>

- go one directory up: `cd ..`

- download LCIO:

```
git clone https://github.com/iLCSoft/LCIO.git
```

- change to LCIO folder, create and change to build directory:

```
cd LCIO; mkdir build; cd build
```

- run cmake with option to build root dictionaries and C++17:

```
cmake -DBUILD_ROOTDICT=ON -D CMAKE_CXX_STANDARD=17 ..
```

- make it:

```
make -j 4 install
```

- test your installation:

```
make test
```

- set up paths etc:

```
cd ..; . ./setup.sh
```

Hint: If you have (rather recent) gcc, python, root and cmake on your laptop / local desktop, try to install LCIO there for more convenient interactive work in root!

The first plot

- go to back to directory with macros:

```
cd ../../miniDST/examples (O1/O3)  
cd /home/ilc/scripts (O2)
```

- get the data:

- download from links given on <https://github.com/ILDAnaSoft/miniDST> (O1/O3)
 - or from links given on <https://github.com/EnginEren/delphes2lcio-docker> (O2)

- start root:

```
root
```

this reads the provided .rootlogon.C

=> most of it is optional and a matter of taste - the crucial part is:

```
gInterpreter->AddIncludePath("$LCIO");  
gSystem->Load("${LCIO}/lib/liblcio.so");  
gSystem->Load("${LCIO}/lib/liblcioDict.so");
```

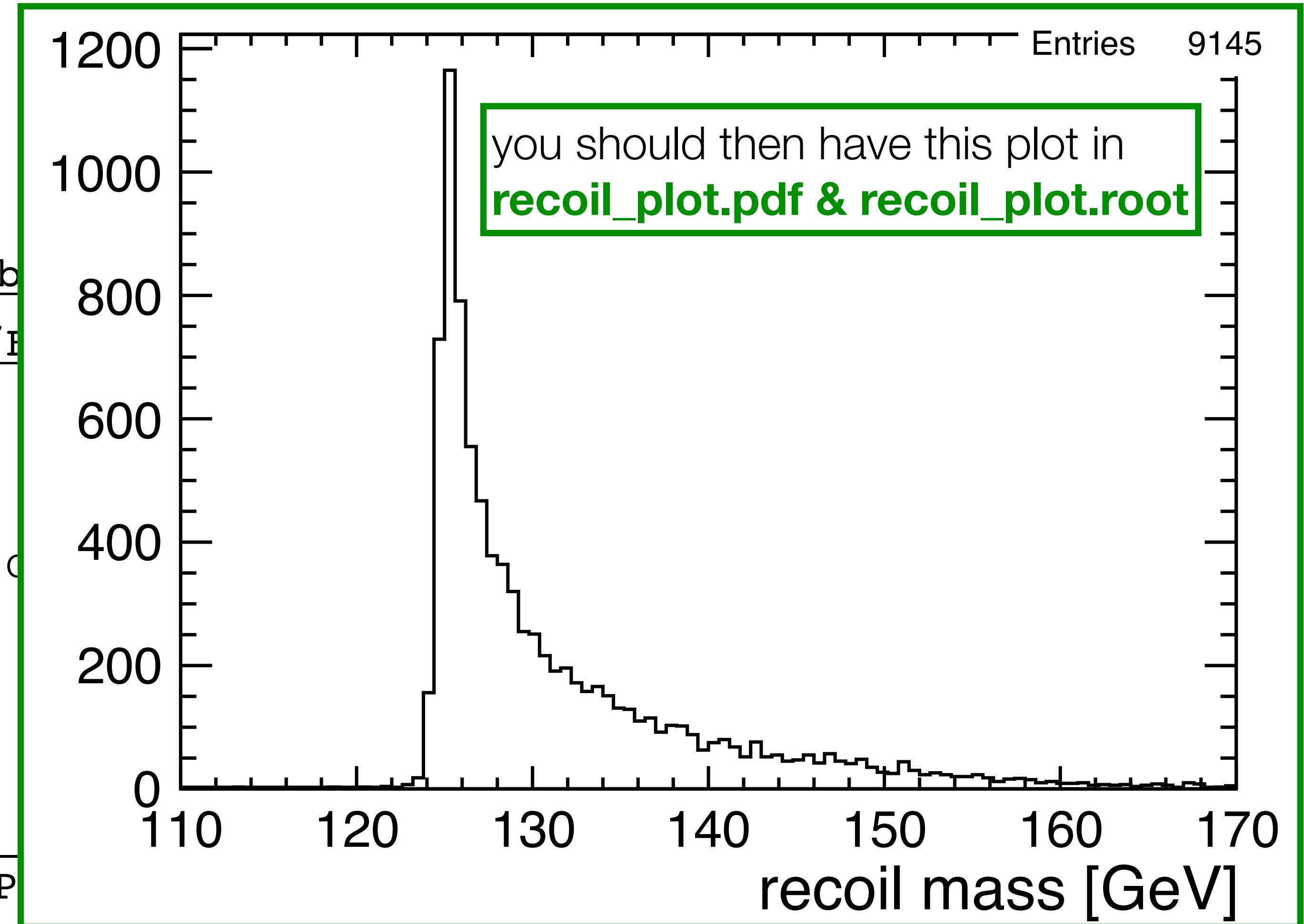
- run first macro:

```
.x higgs_recoil.C ("data/rv01-16-p10_250.sv01-14-01-  
p00.mILD_o1_v05.E250-TDR_ws.I106479.Pe2e2h.eL.pR-00001-  
ILDminiDST.slcio");
```

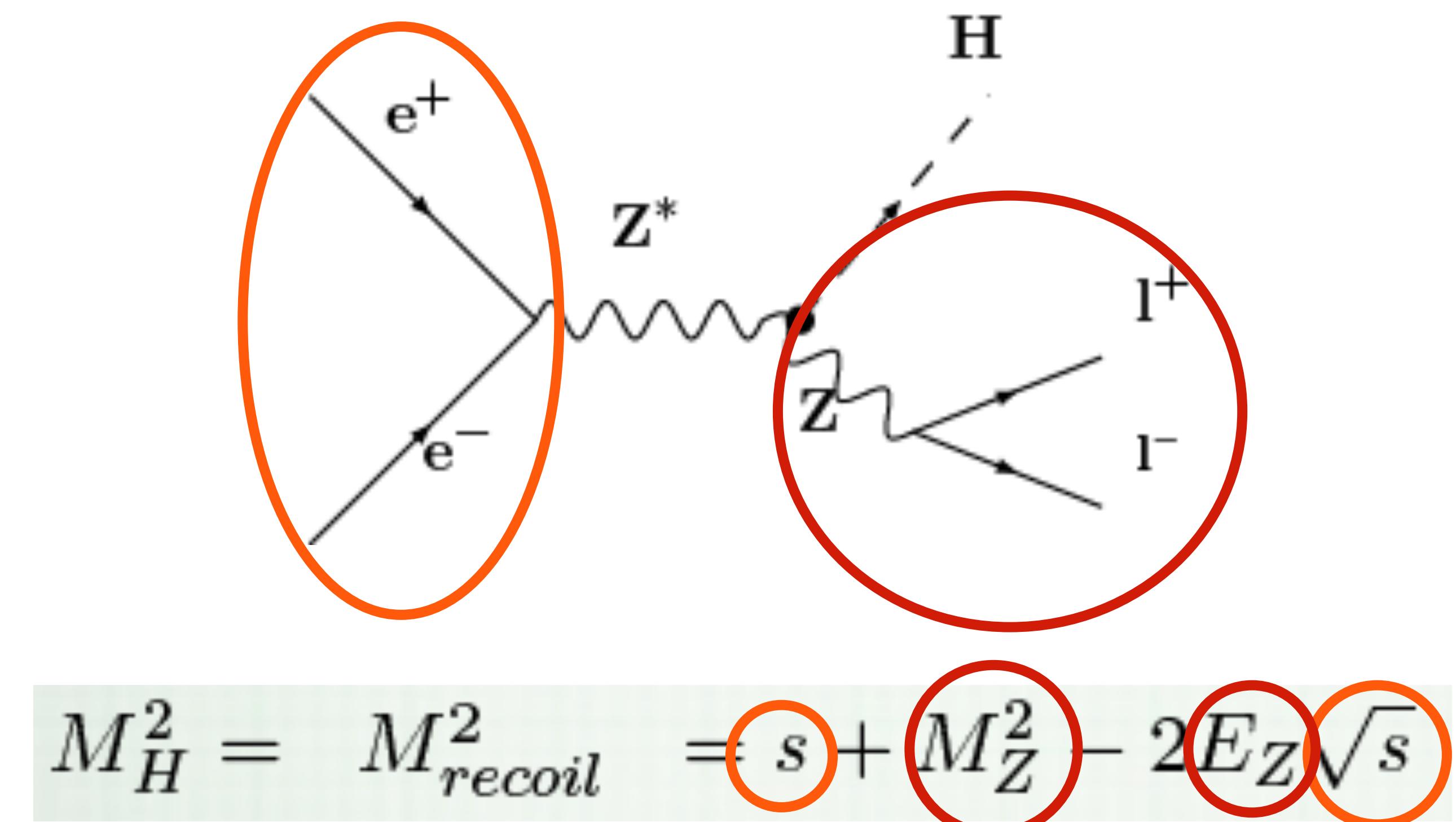
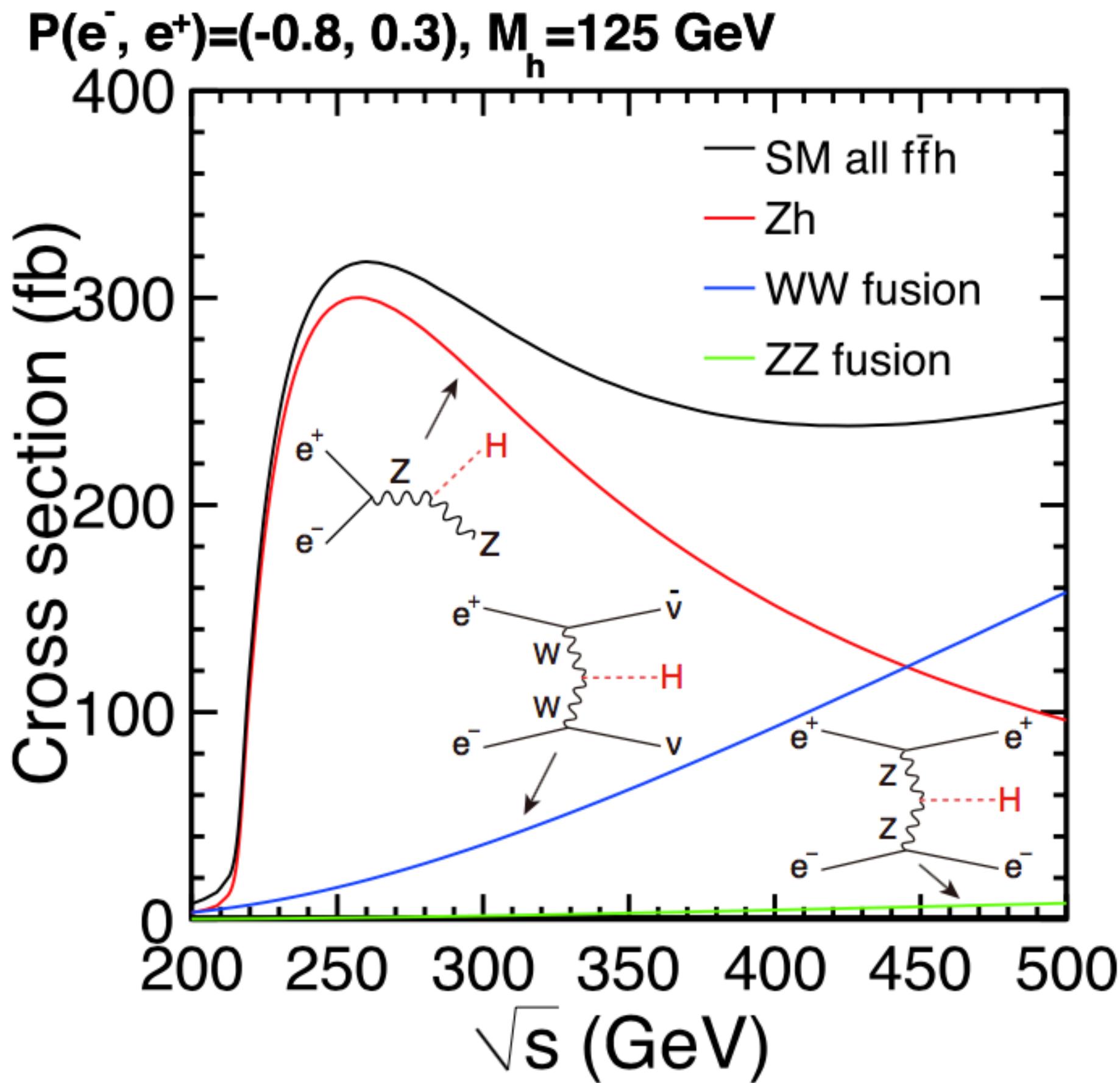
The first plot

- go to back to directory with macros:
`cd ../../miniDST/examples` (O1/O3)
`cd /home/ilc/scripts` (O2)
- get the data:
 - download from links given on <https://github>
 - or from links given on <https://github.com/I>
- start root:
`root`
 this reads the provided .rootlogon.C
 => most of it is optional and a matter of taste - the code:

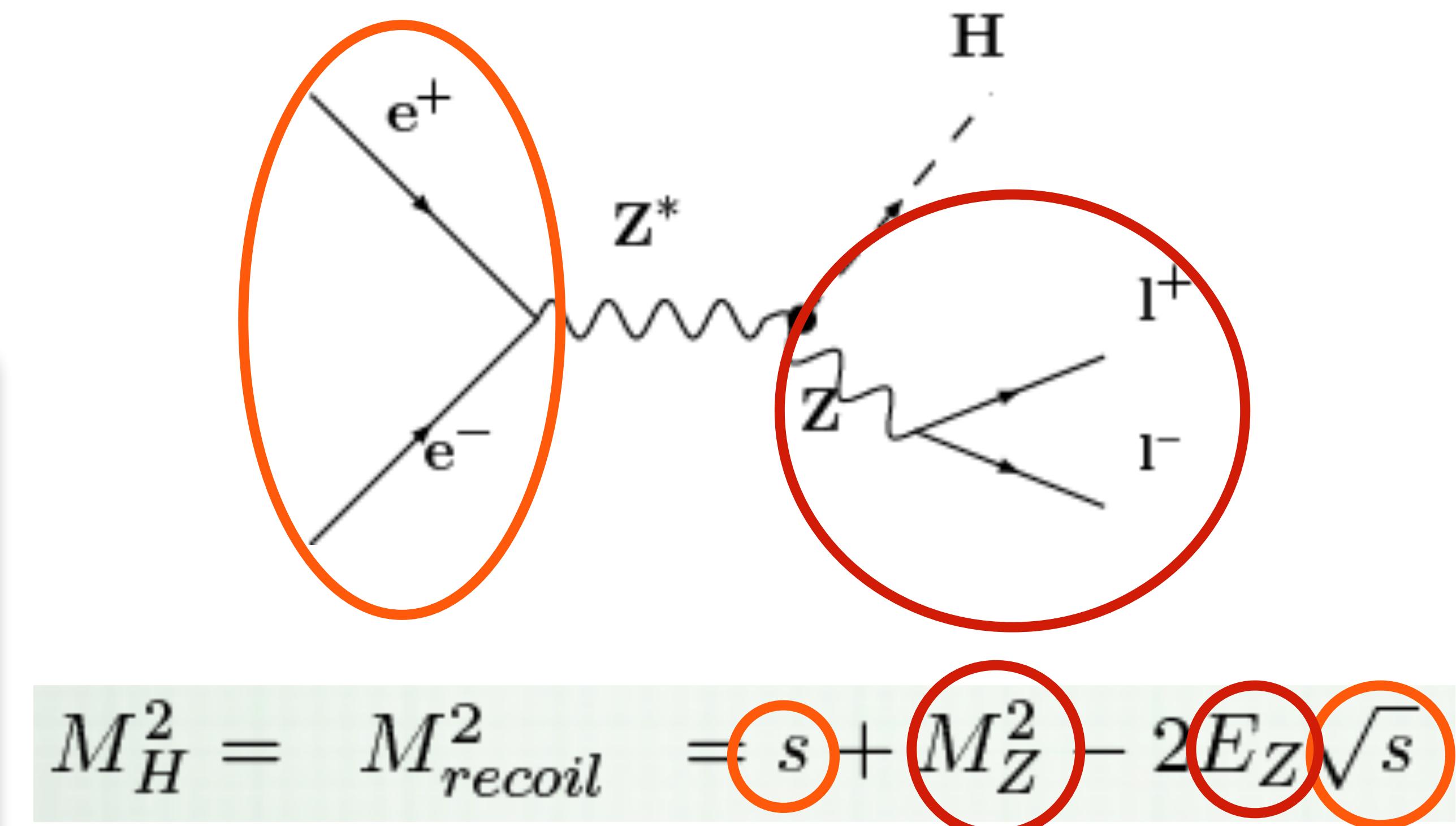
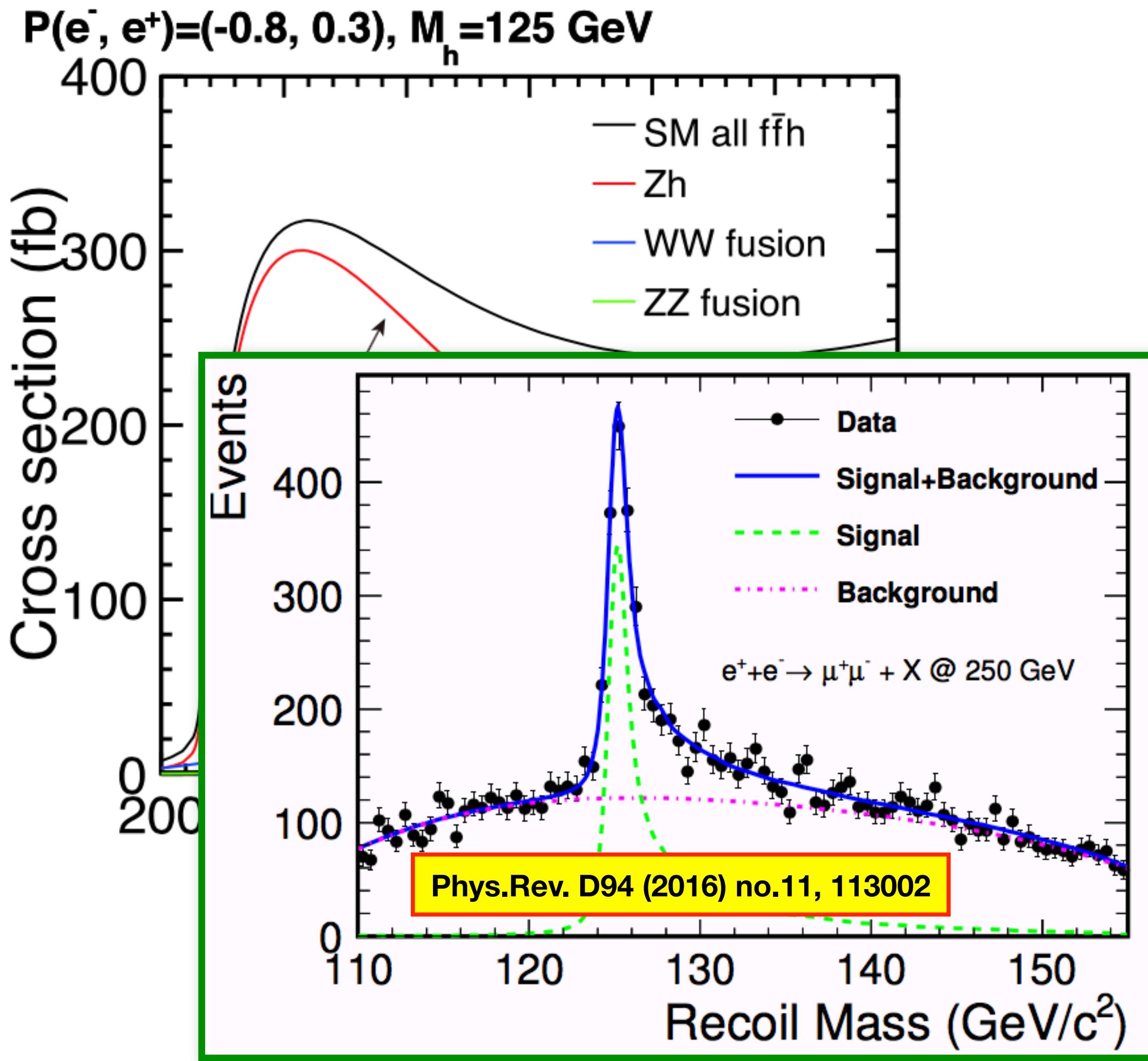
```
glInterpreter->AddIncludePath("$LCIO");
gSystem->Load("${LCIO}/lib/liblcio.so");
gSystem->Load("${LCIO}/lib/liblcioDict.so");
```
- run first macro:
`.x higgs_recoil.c ("data/rv01-16-p10_
p00.mILD_o1_v05.E250-TDR_ws.I106479.P
ILDminiDST.slcio");`



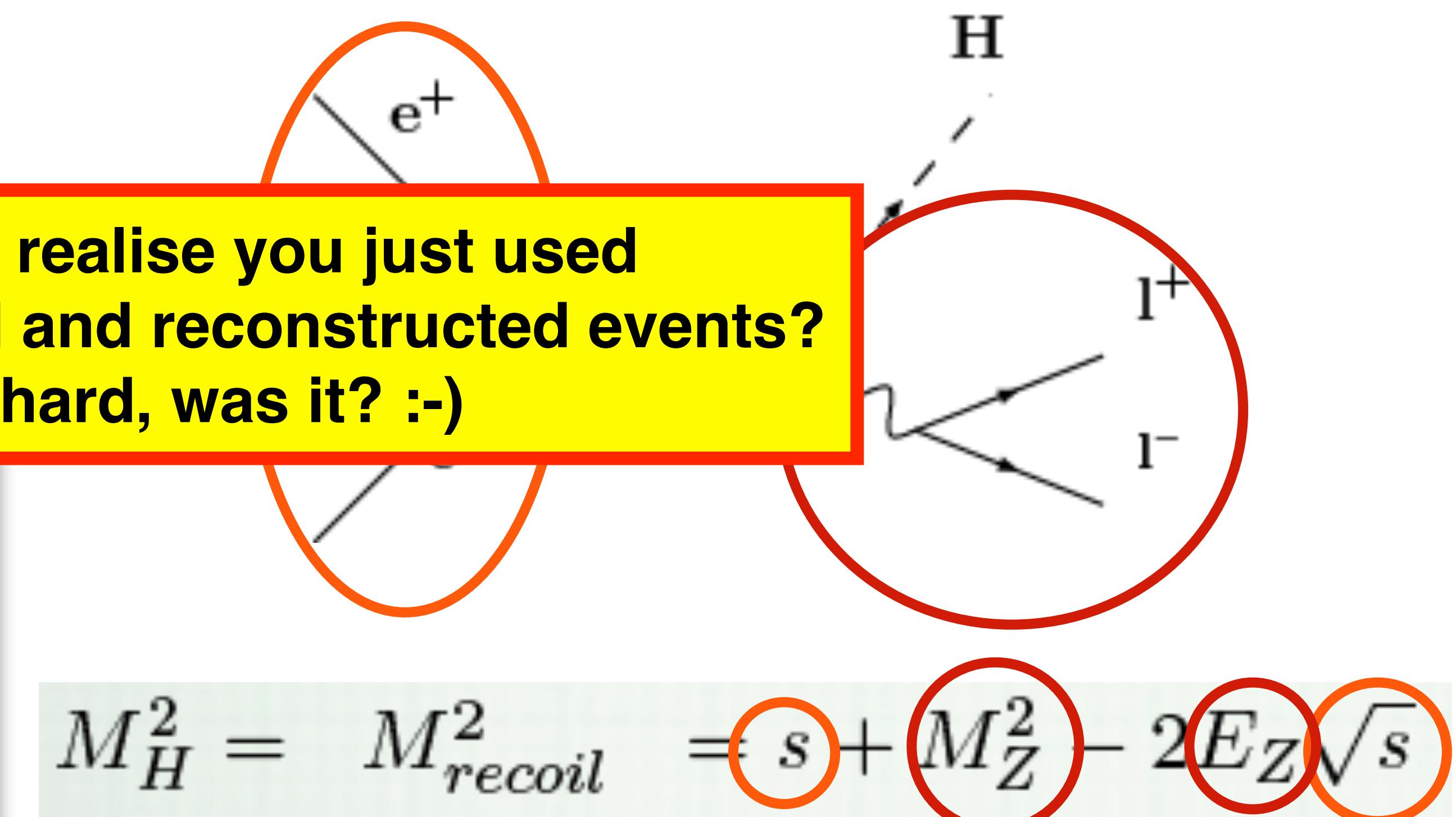
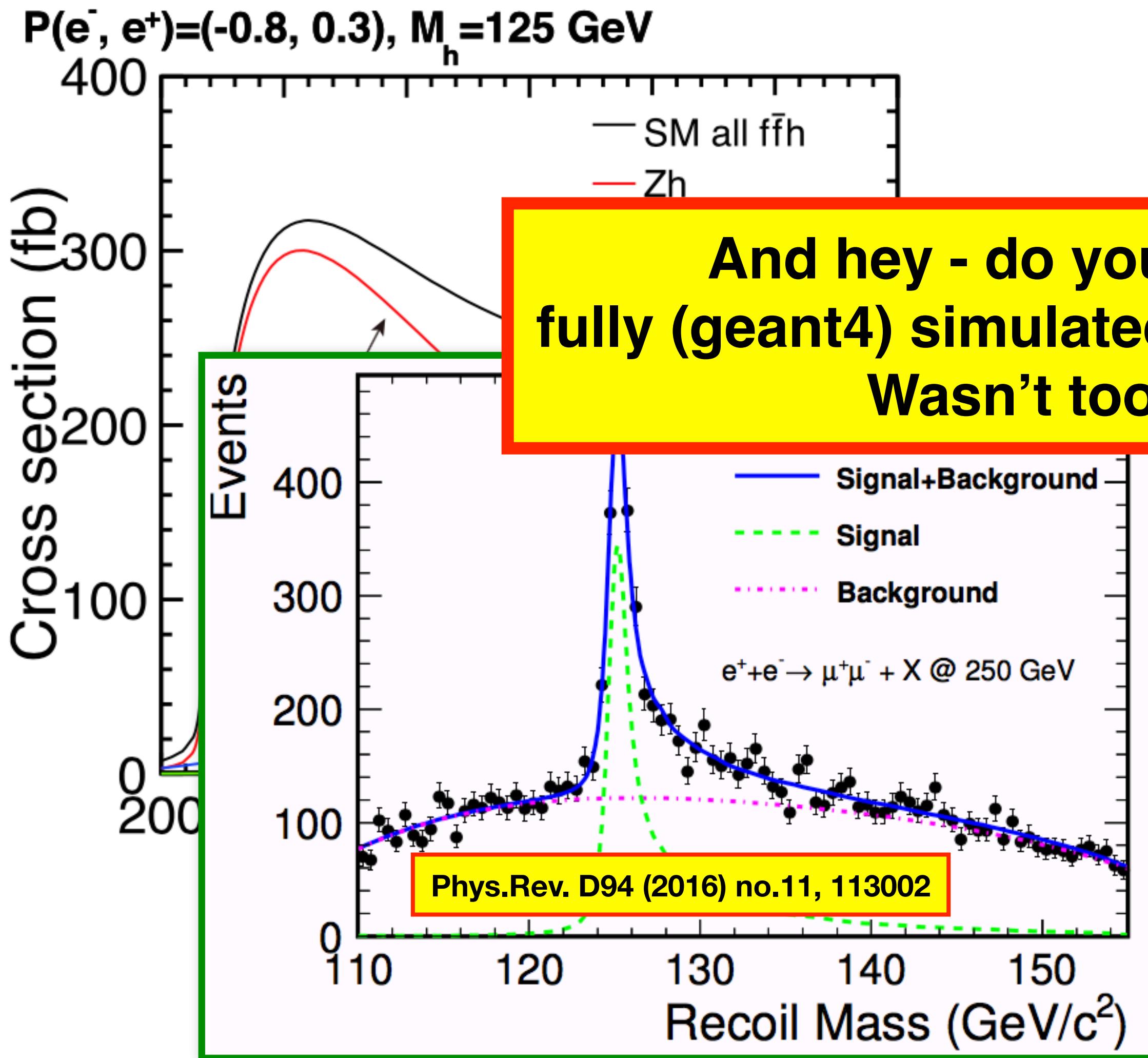
What are we seeing?



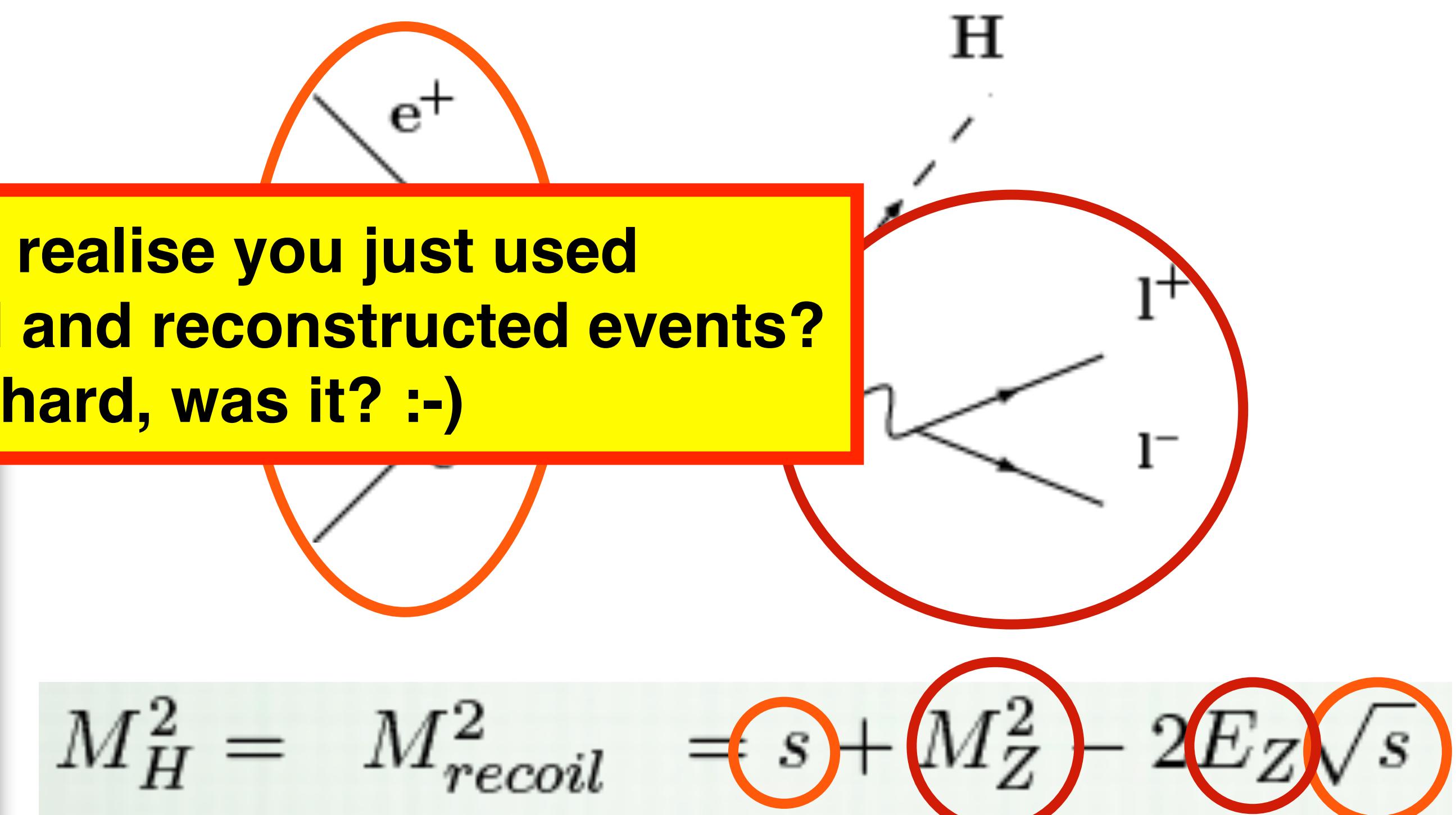
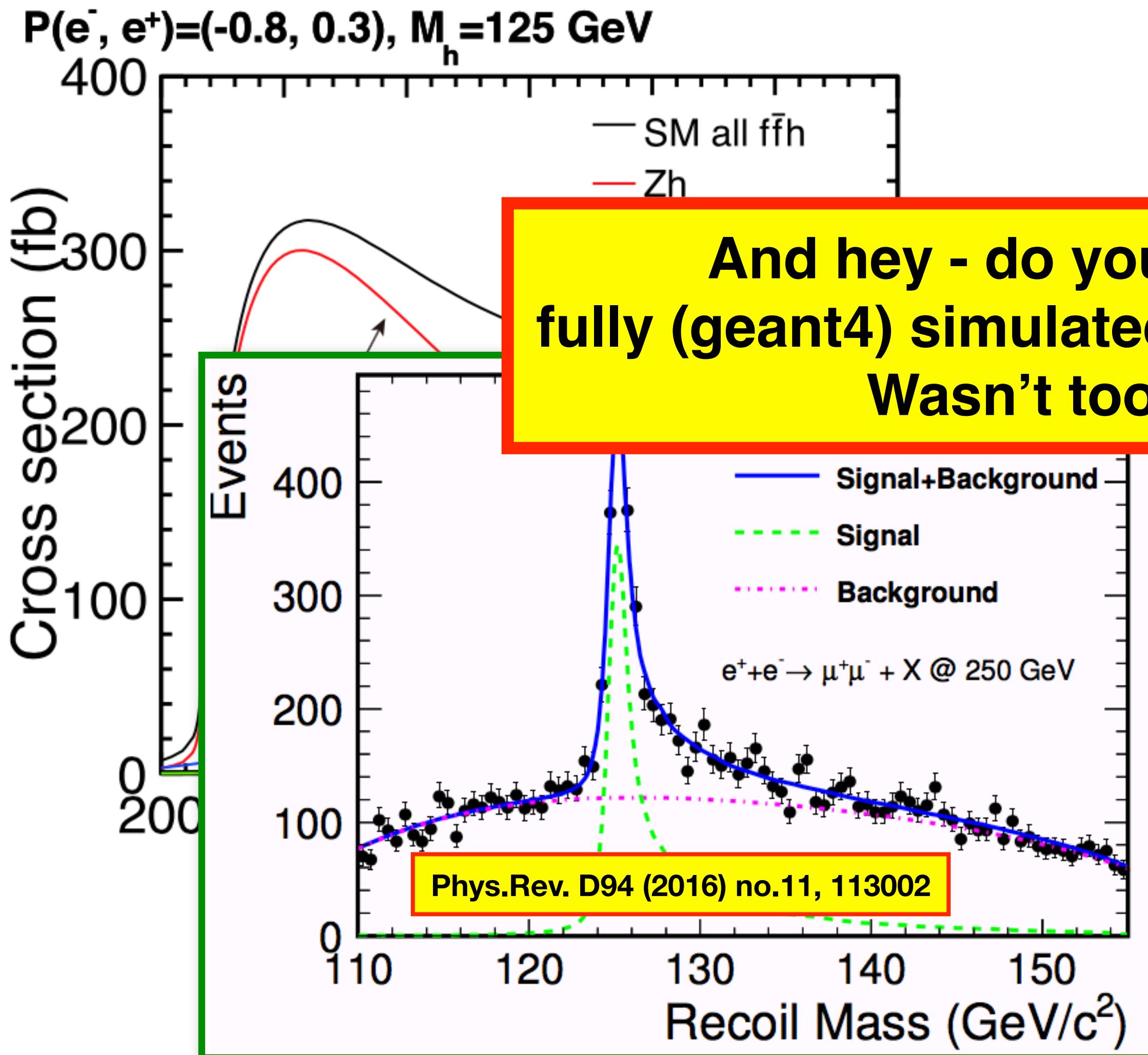
What are we seeing?



What are we seeing?



What are we seeing?



Let's add some backgrounds...

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account **all** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are **not** included
 - instead have separately: ee -> ffbar h
- file name contains:

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account **all** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are **not** included
 - instead have separately: ee -> ffbar h
- file name contains:

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

process:
e2 = muon, so
mumu h
(I dentification number)

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account **all** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are **not** included
 - instead have separately: ee -> ffbar h
- file name contains:

polarisation:
electron L (eft-handed)
positron R (ight-handed)

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

process:
e2 = muon, so
mumu h
(I dentification number)

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account ***all*** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are ***not*** included
 - instead have separately: ee -> ffbar h
- file name contains:

polarisation:
electron L (eft-handed)
positron R (ight-handed)

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

process:
e2 = muon, so
mumu h
(I dentification number)

serial file number
& file format

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account ***all*** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are ***not*** included
 - instead have separately: ee -> ffbar h
- file name contains:

polarisation:
electron L (eft-handed)
positron R (ight-handed)

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

**energy and
beam parameters:**
250 GeV,
TDR_ws

process:
e2 = muon, so
mumu h
(I dentification number)

**serial file number
& file format**

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account ***all*** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are ***not*** included
 - instead have separately: ee -> ffbar h
- file name contains:

detector model

polarisation:
electron L (eft-handed)
positron R (ight-handed)

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

**energy and
beam parameters:**
250 GeV,
TDR_ws

process:
e2 = muon, so
mumu h
(I dentification number)

**serial file number
& file format**

Interlude 1: nomenclature of processes

- more details c.f. <https://indico.cern.ch/event/868940/contributions/3814465/>
- classify physics processes by the number of (fermions + antifermions) in the final state:
 - ee -> 2f: ee -> f fbar (f = e, mu, tau, u, d, s, c, b, nu)
 - ee -> 4f: mostly WW / ZZ, but taking into account ***all*** contributing matrix elements and their interference
 - ee -> 6f: mostly ttbar, some ZZZ, WWZ, but again, all MEs + interference considered!
 - “SM” samples: mass of Higgs is set to huge value, so that Feynman diagrams containing the Higgs are ***not*** included
 - instead have separately: ee -> ffbar h
- file name contains:

detector model

polarisation:
electron L (eft-handed)
positron R (ight-handed)

rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.l106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio

**iLCSoft versions for
reconstruction and simulation**

**energy and
beam parameters:**
250 GeV,
TDR_ws

process:
e2 = muon, so
mumu h
(I dentification number)

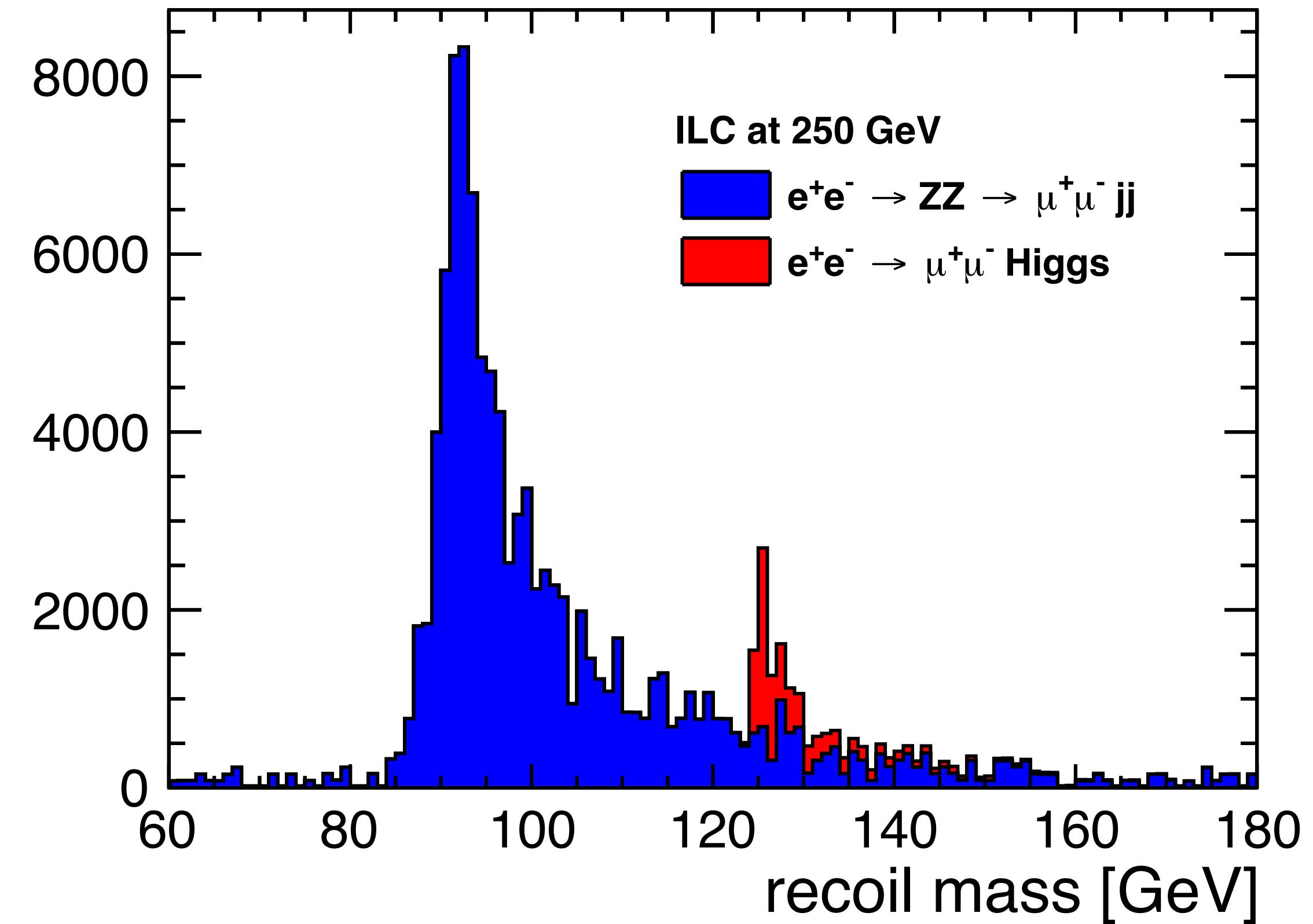
**serial file number
& file format**

And now with background

- restart root:

```
root
```
- ```
.x higgs_recoil_with_bkg.C ("../data/"); (O1/O3)
```

  
`.x higgs_recoil_with_bkg.C (); (O2)`
- you should then have this plot in
  - **recoil\_plot\_with\_bkg.pdf & recoil\_plot\_with\_bkg.root**
- macro takes as further (optional) arguments:
  - double lumi\_target=900., // 900 fb-1
  - double epol\_target=-0.8, // P(e-) =-80%
  - double ppol\_target=+0.3. // P(e+)=+30%
- try to change these settings to the opposite polarisation signs and redo the plot!



# Interlude 2: Why these funny values?

## - The ILC Strawman Running Scenario & Polarisation



- beam polarisation absolute values:
  - Electron beam:  $|P(e^-)| \geq 80\%$
  - Positron beam:  $|P(e^+)| = 30\%$ ,  
at 500 GeV upgradable to 60%  
at 1 TeV assume 20%
- **Notation:** (  $P(e^-)$  ,  $P(e^+)$  )
- sharing of luminosity between polarisation signs:

| $\sqrt{s}$ | $\int \mathcal{L} dt$ | -+                   | +-                   | ++                   | --                   |
|------------|-----------------------|----------------------|----------------------|----------------------|----------------------|
| 250 GeV    | 2 ab <sup>-1</sup>    | 0.9 ab <sup>-1</sup> | 0.9 ab <sup>-1</sup> | 0.1 ab <sup>-1</sup> | 0.1 ab <sup>-1</sup> |
| 350 GeV    | 200 fb <sup>-1</sup>  | 135 fb <sup>-1</sup> | 45 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  |
| 500 GeV    | 4 ab <sup>-1</sup>    | 1.6 ab <sup>-1</sup> | 1.6 ab <sup>-1</sup> | 0.4 ab <sup>-1</sup> | 0.4 ab <sup>-1</sup> |
| 1 TeV      | 8 ab <sup>-1</sup>    | 3.2 ab <sup>-1</sup> | 3.2 ab <sup>-1</sup> | 0.8 ab <sup>-1</sup> | 0.8 ab <sup>-1</sup> |
| 91 GeV     | 100 fb <sup>-1</sup>  | 40 fb <sup>-1</sup>  | 40 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  |
| 161 GeV    | 500 fb <sup>-1</sup>  | 340 fb <sup>-1</sup> | 110 fb <sup>-1</sup> | 25 fb <sup>-1</sup>  | 25 fb <sup>-1</sup>  |

all up-to-date numbers  
in ILC input document  
to the European strategy

detailed reasoning c.f.  
[arXiv:1506.07830](https://arxiv.org/abs/1506.07830)

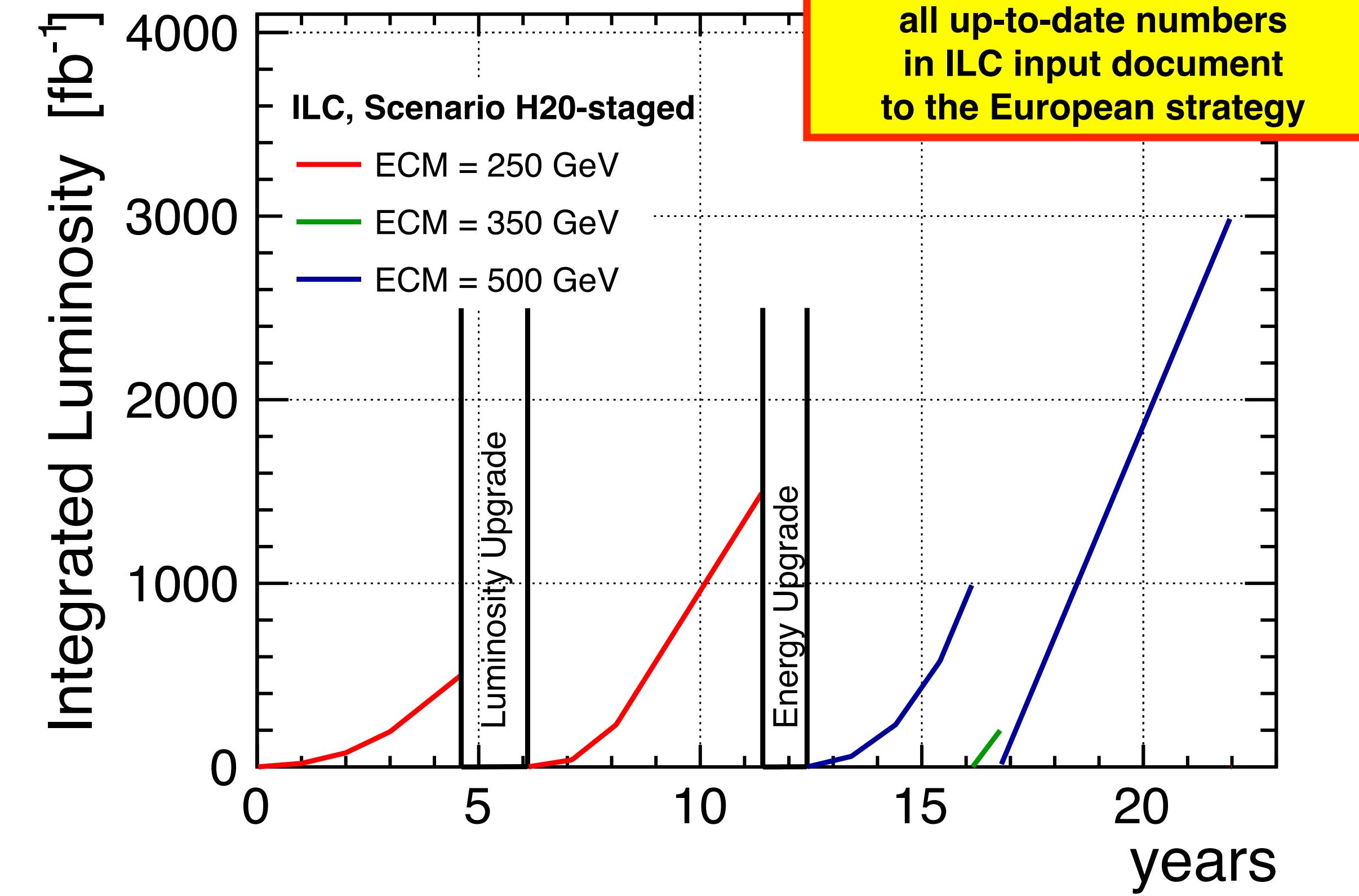
# Interlude 2: Why these funny values?

## - The ILC Strawman Running Scenario & Polarisation



- beam polarisation absolute values:
  - Electron beam:  $|P(e^-)| \geq 80\%$
  - Positron beam:  $|P(e^+)| = 30\%$ ,  
at 500 GeV upgradable to 60%  
at 1 TeV assume 20%
- Notation:**  $( P(e^-), P(e^+) )$
- sharing of luminosity between polarisation signs:

| $\sqrt{s}$ | $\int \mathcal{L} dt$ | -+                   | +-                   | ++                   | --                   |
|------------|-----------------------|----------------------|----------------------|----------------------|----------------------|
| 250 GeV    | 2 ab <sup>-1</sup>    | 0.9 ab <sup>-1</sup> | 0.9 ab <sup>-1</sup> | 0.1 ab <sup>-1</sup> | 0.1 ab <sup>-1</sup> |
| 350 GeV    | 200 fb <sup>-1</sup>  | 135 fb <sup>-1</sup> | 45 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  |
| 500 GeV    | 4 ab <sup>-1</sup>    | 1.6 ab <sup>-1</sup> | 1.6 ab <sup>-1</sup> | 0.4 ab <sup>-1</sup> | 0.4 ab <sup>-1</sup> |
| 1 TeV      | 8 ab <sup>-1</sup>    | 3.2 ab <sup>-1</sup> | 3.2 ab <sup>-1</sup> | 0.8 ab <sup>-1</sup> | 0.8 ab <sup>-1</sup> |
| 91 GeV     | 100 fb <sup>-1</sup>  | 40 fb <sup>-1</sup>  | 40 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  | 10 fb <sup>-1</sup>  |
| 161 GeV    | 500 fb <sup>-1</sup>  | 340 fb <sup>-1</sup> | 110 fb <sup>-1</sup> | 25 fb <sup>-1</sup>  | 25 fb <sup>-1</sup>  |



all up-to-date numbers  
in ILC input document  
to the European strategy

detailed reasoning c.f.  
[arXiv:1506.07830](https://arxiv.org/abs/1506.07830)

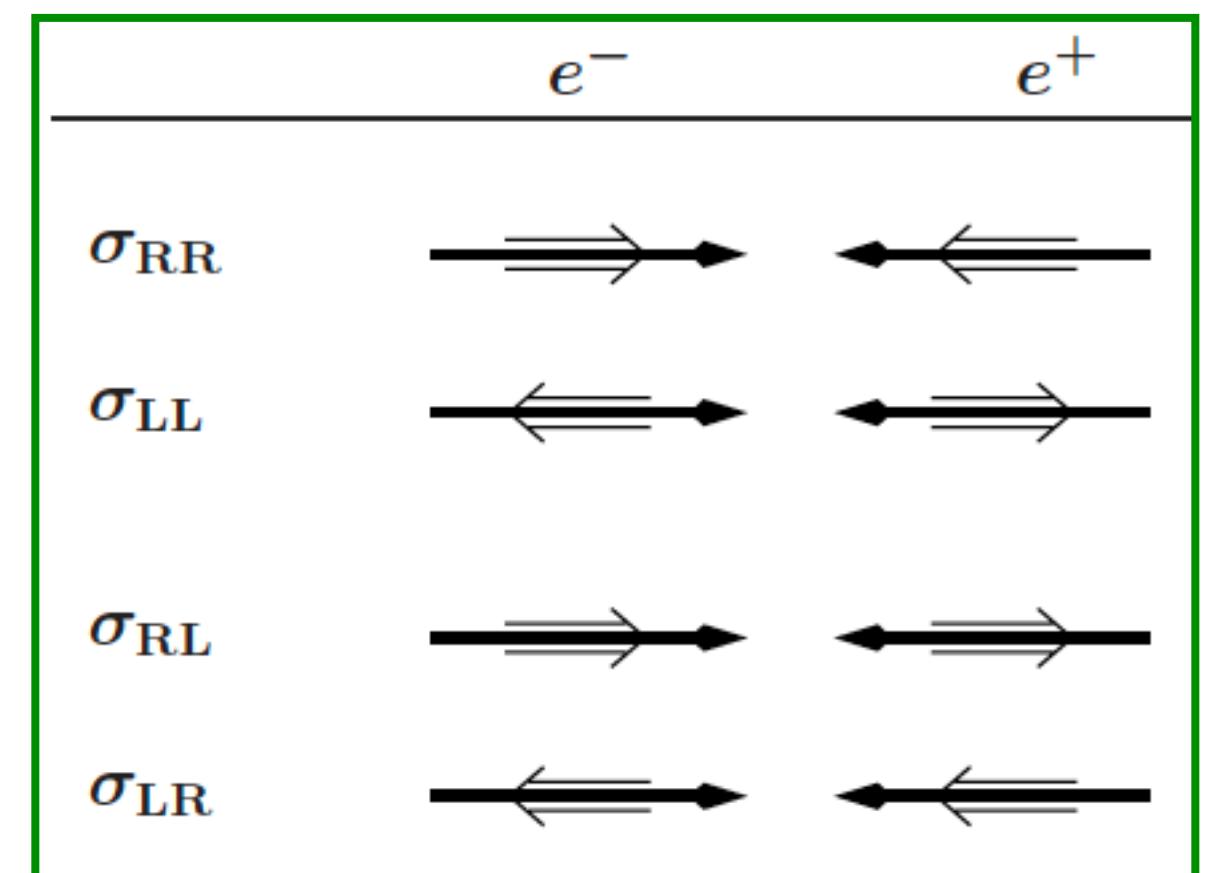
# Future $e^+e^-$ Colliders and (longitudinally) Polarised Beams

- Longitudinally **polarised beams** are a special feature of **Linear  $e^+e^-$  Colliders**:
  - SLC:  $P(e^-) = \pm 80\%$ ,  $P(e^+) = 0\%$
  - ILC:  $P(e^-) = \pm 80\%$ ,  $P(e^+) = \pm 30\%$  (upgrade 60%)
  - CLIC:  $P(e^-) = \pm 80\%$ ,  $P(e^+) = 0\%$
- Electroweak interactions highly sensitive to chirality of fermions:  $SU(2)_L \times U(1)$ 
  - every cross section depends on beam polarisations
  - with both its beams polarised, ILC is “four colliders in one”:**

**General references on polarised  $e^+e^-$  physics:**

- [arXiv:1801.02840](https://arxiv.org/abs/1801.02840)
- [Phys. Rept. 460 \(2008\) 131-243](https://doi.org/10.1016/j.physrep.2008.02.001)

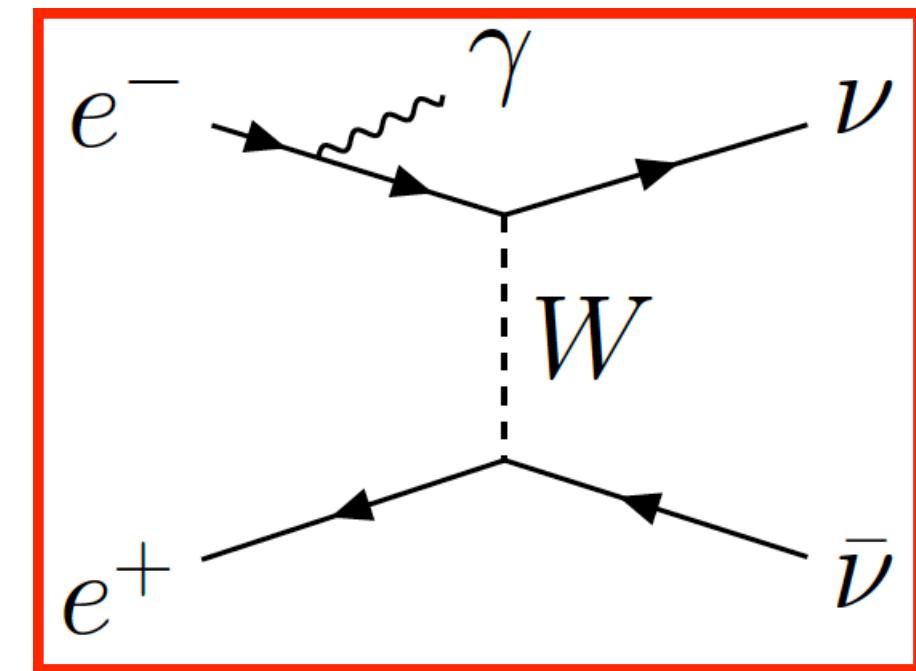
$$P = \frac{N_R - N_L}{N_R + N_L}$$



# Physics benefits of polarised beams

## background suppression:

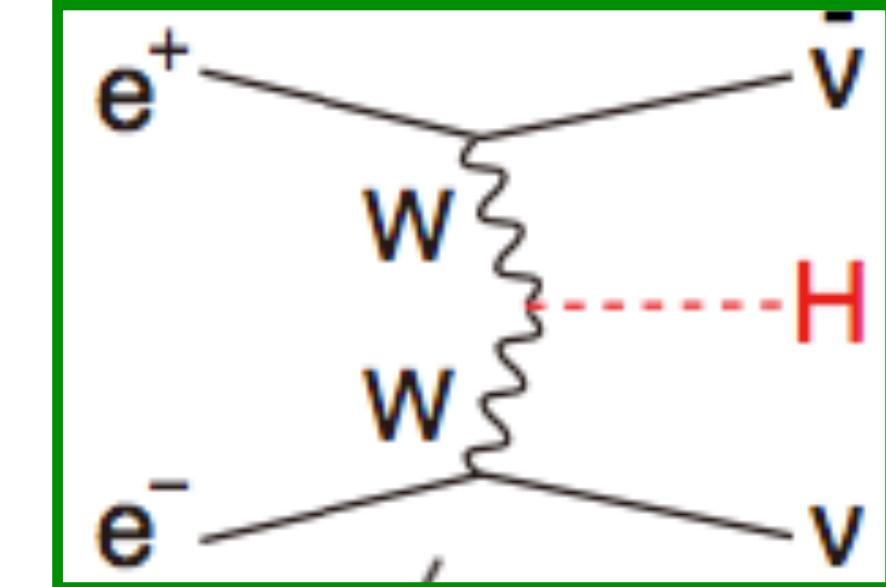
- $e^+e^- \rightarrow WW / \nu\nu$   
strongly P-dependent  
since t-channel only  
for  $e^-_L e^+_R$



## signal enhancement:

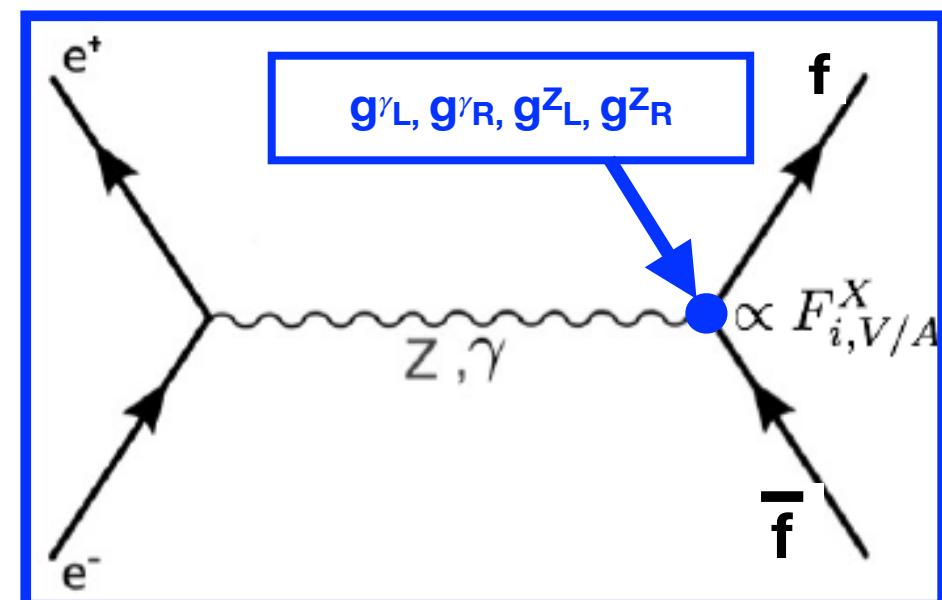
- Higgs production  
in  $WW$  fusion
- many BSM processes

have strong polarisation dependence => higher S/B



## chiral analysis:

- SM:  $Z$  and  $\gamma$  differ in  
couplings to left- and  
right-handed fermions
- BSM:  
chiral structure unknown, needs to be determined!



## redundancy & control of systematics:

- “wrong” polarisation yields “signal-free” control sample
- flipping *positron* polarisation controls nuisance effects on observables relying on *electron* polarisation
- essential: fast helicity reversal for *both* beams!

# Polarised cross sections

$$\sigma_{P_{e^-} P_{e^+}} = \frac{1}{4} \left\{ (1 + P_{e^-})(1 + P_{e^+})\sigma_{RR} + (1 - P_{e^-})(1 - P_{e^+})\sigma_{LL} \right. \\ \left. + (1 + P_{e^-})(1 - P_{e^+})\sigma_{RL} + (1 - P_{e^-})(1 + P_{e^+})\sigma_{LR} \right\}.$$

- For  $\sigma_{RR}$ ,  $\sigma_{LR}$  etc, use the generator cross sections given in the event header:

```
float xsection = evt->parameters().getFloatVal("CrossSection_fb");
```

- pre-factors are the respective event weights:

```
// polarisation weights for {LR, RL, LL, RR} events, as example for target P(e-,e+)=(-80%,+30%):
// LR: polweight = (1-epol_target)*(1+ppol_target)/4.; // -80%,+30% => 1.8 * 1.3 / 4. = 0.585
// RL: polweight = (1+epol_target)*(1-ppol_target)/4.; // -80%,+30% => 0.2 * 0.7 / 4. = 0.035
// LL: polweight = (1-epol_target)*(1-ppol_target)/4.; // -80%,+30% => 1.8 * 0.7 / 4. = 0.315
// RR: polweight = (1+epol_target)*(1+ppol_target)/4.; // -80%,+30% => 0.2 * 1.3 / 4. = 0.065
```

- Note: data sets with  $(\text{sign}(P_{e^-}), \text{sign } P_{e^+}) = (-,+)$  and  $(+, -)$  often have

- different initial S/B ratio
- different background composition => different kinematics etc

=> analyse data sets with different polarisation signs separately, different cut optimisation - either combine results afterwards, or exploit polarisation dependence in interpretation

# Useful tools

---

- **anajob [your .slcio file] | less**  
prints (after some header information) the list of collections available on each event, incl. their number of elements

=> try it - what do you see?

- **dumpevent [your .slcio file] [event number] | less**  
prints the content of all collections on the given event

=> try it - and find

- the IsolatedMuons collection
- the Refined2Jet collection

# Now it is your turn !

---

- Try to improve the signal-to-background ratio by applying a cut on the sum of the b-likeness values of the two jets.
- For this, read in the **Refined2Jets** collection, check that it is there and contains 2 jets.
- Then get the b-likeness values (MVA output between 0 and 1).
- You find an example of how to access jets and b-tag information in [./examples/jet\\_btag.C](#) ([./scripts/jet\\_btag.C](#))
- Take a look at this (of course you can also run it if you like!) and modify your **higgs\_recoil\_with\_bkg.C** such that the recoil mass histograms are only filled if the sum of the two b-likeness values  $> 1$ .

# How to continue

---

- how to get more data:
  - regularly check <http://ilcsnowmass.org> - large data sets (SM + Higgs) will appear there soon:
    - Delphes-miniDSTs of for 250 GeV, 350 GeV, 500 GeV, 1 TeV (from ILC TDR MC production, Whizard 1.95)
    - SGV-miniDSTs of new 250 GeV Whizard 2.8.4 samples
    - ILD-miniDSTs (full simulation) of TDR MC production - prioritisation depending on user requests !
  - if you need additional samples - eg BSM signals: **contact us!**  
=> depending on size/ complexity of request, we'll either produce them or teach you how to produce them
- choose a topic: take a look into [arXiv:2007.03650](https://arxiv.org/abs/2007.03650) - and don't hesitate to **contact us** if you have questions!
- next tutorial:
  - Introduction to ilcsoft (DD4HEP / Marlin) -> July 21 cf <https://agenda.linearcollider.org/event/9272/>

# Contact information

<https://linearcollider.org/team/wg3/>

## WG3 Subgroups

- Speakers Bureau
- Machine-Detector Interface Subgroup
- Detector and Technology R&D Subgroup
- [Software and Computing Subgroup](#) ↗
- [Physics Potential and Opportunities Subgroup](#)

## Links

- [IDT-WG3 Mandate and Workplan](#) ↗
- [WG3 meeting pages](#) ↗
- [Documents for Physics & Detectors](#) ↗

## Contact for IDT-WG3

- Chair: [Hitoshi Murayama](#) ↗, UC Berkeley/U. Tokyo
- Deputy: [Jenny List](#) ↗, DESY
- Deputy: [Claude Vallée](#) ↗, CPPM-IN2P3