Application of Matrix Element Method

Junping Tian, Keisuke Fujii (KEK)

Nov. 1 @ 39th General Physics Meeting of ILC Subgroup

recent development: matrix element method

(approach the true likelihood of each event)

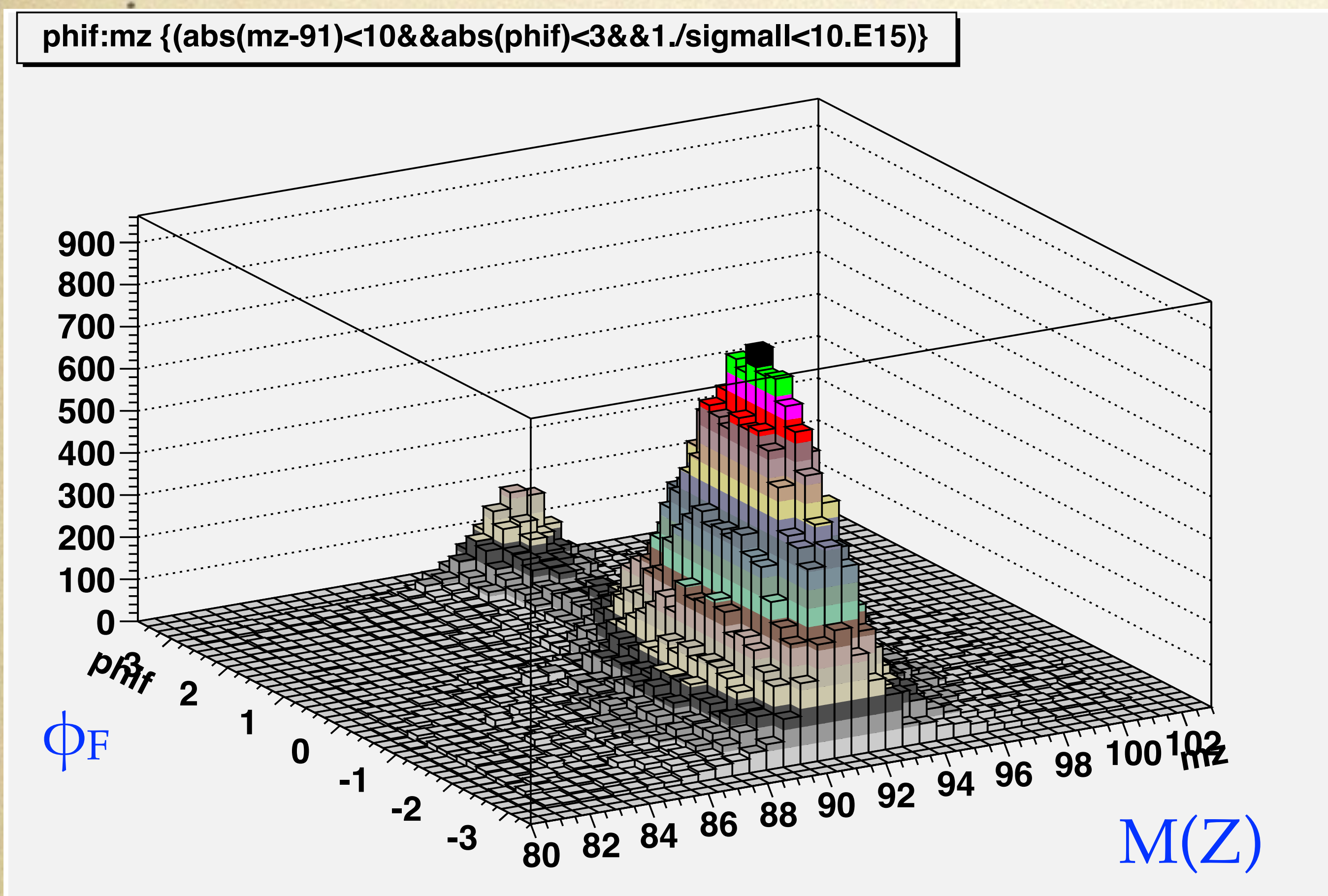
- @ ILC: well defined initial states, precisely measured final states, MEM perfectly fits for precision measurement
- MEM tools for full simulation released with latest ilcsoft-v01-17-06: Physsim-v00-01
- basic verification done
- principle demonstrated in first application for e+e- —> eeH via ZZ-fusion
- main focus now is to apply for other major Higgs measurement (today for recoil mass)

svn co https://svnsrv.desy.de/basic/physsim/Physsim/trunk

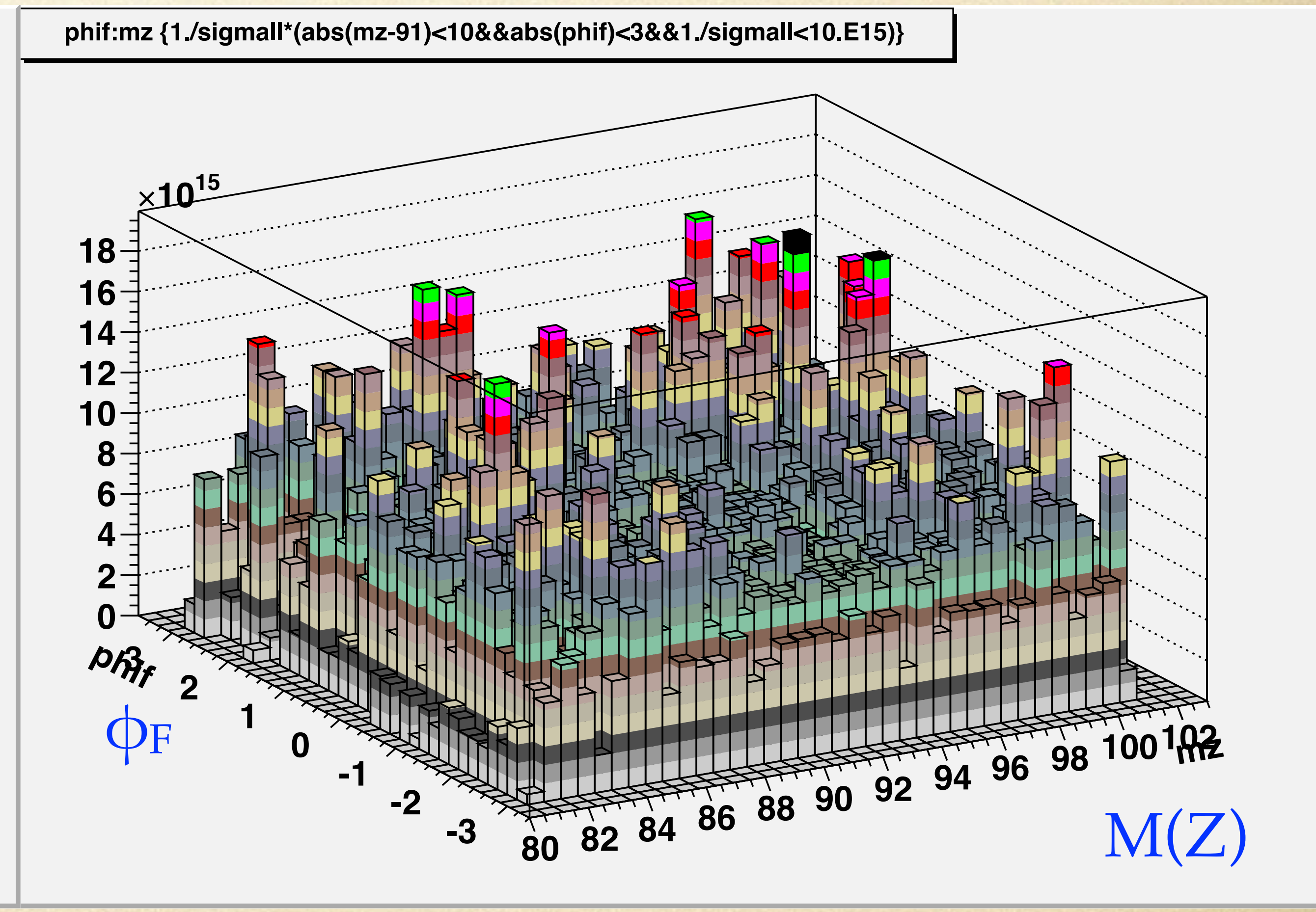
reminder: verification of calculated matrix element

(ZHH events)

original events

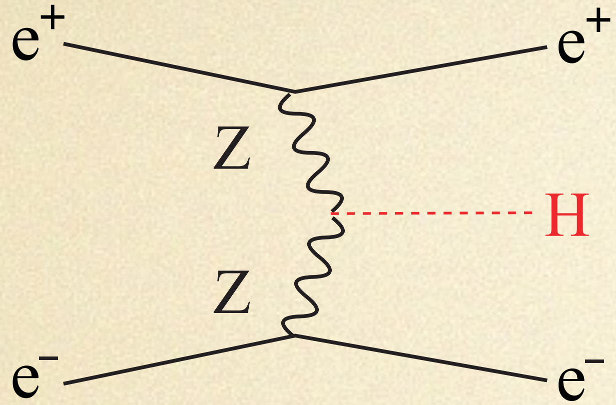


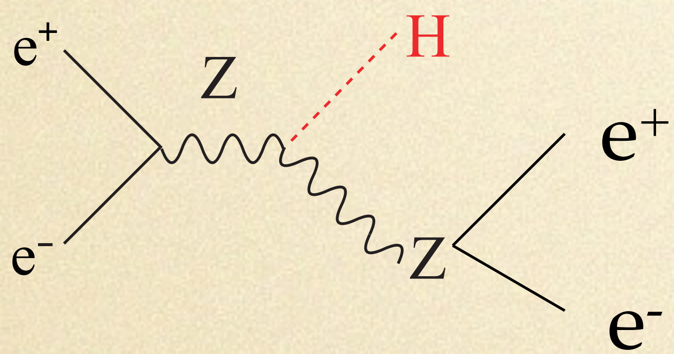
weighted by $\frac{1}{|\mathrm{ME}|^2}$

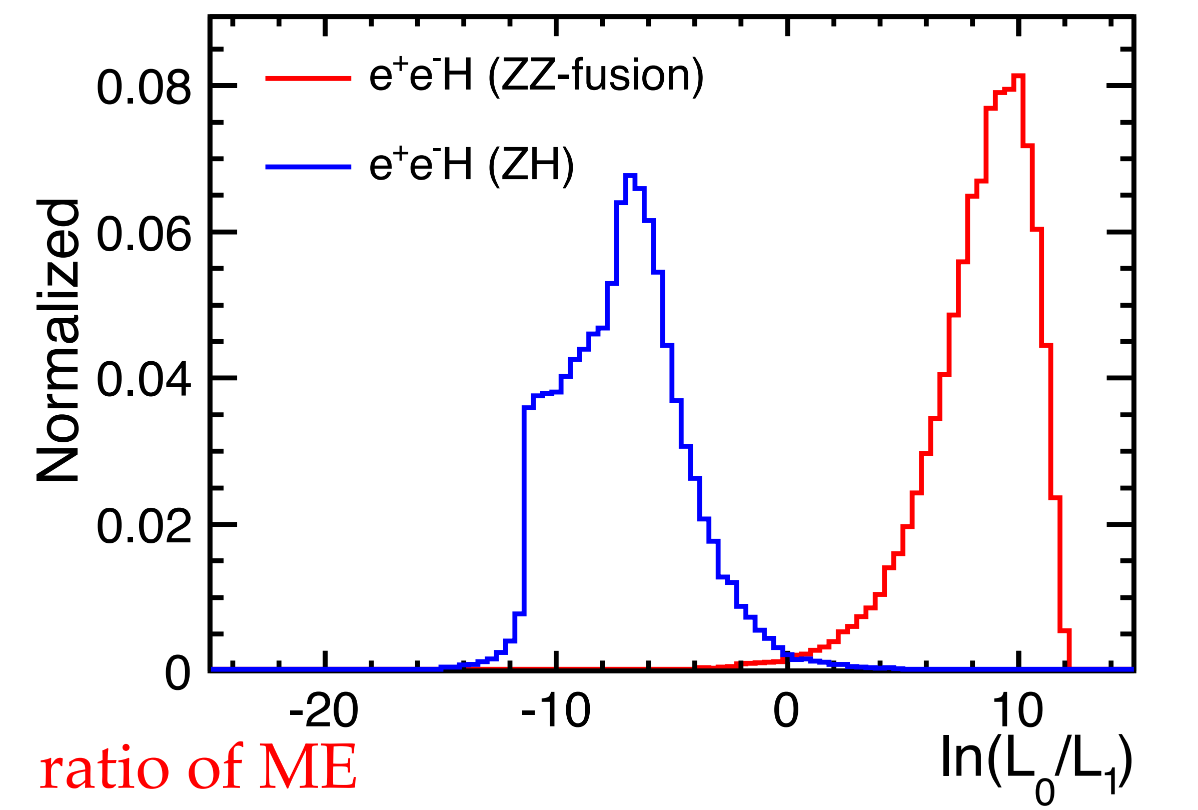


ϕ_F: azimuthal angle defined in Z rest frame of fermion from Z—>ff

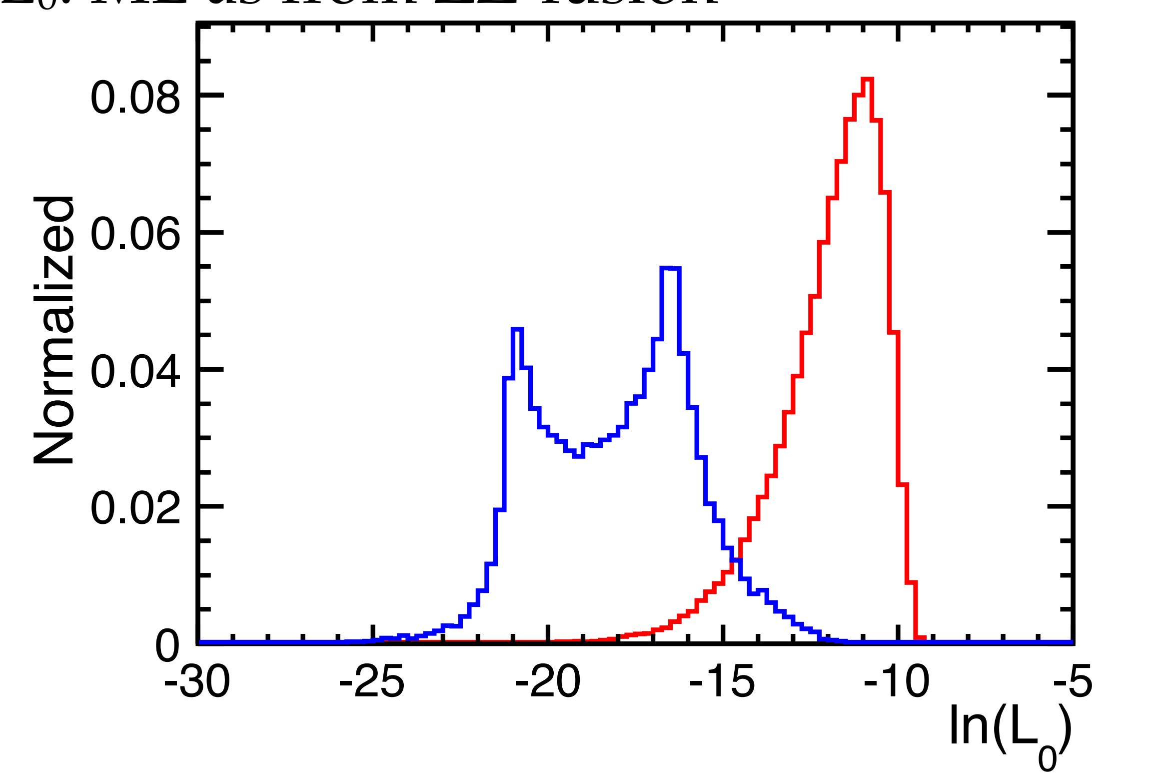
first application: e+e- —> e+e-H



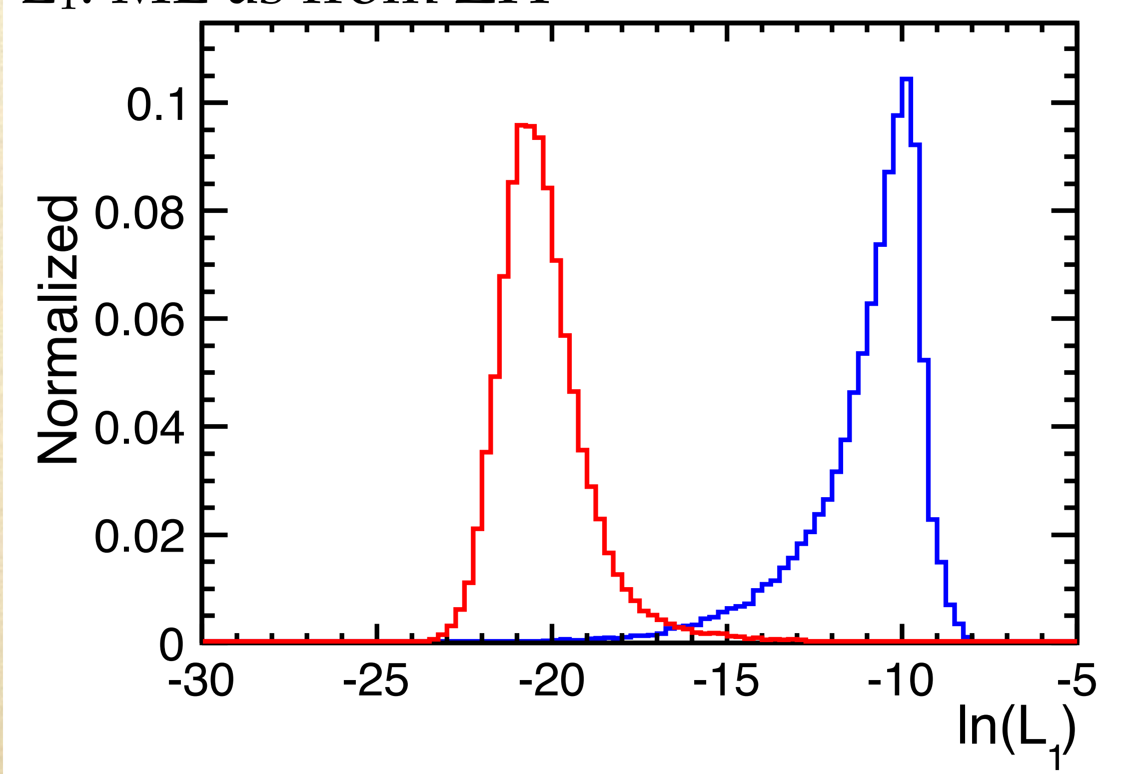




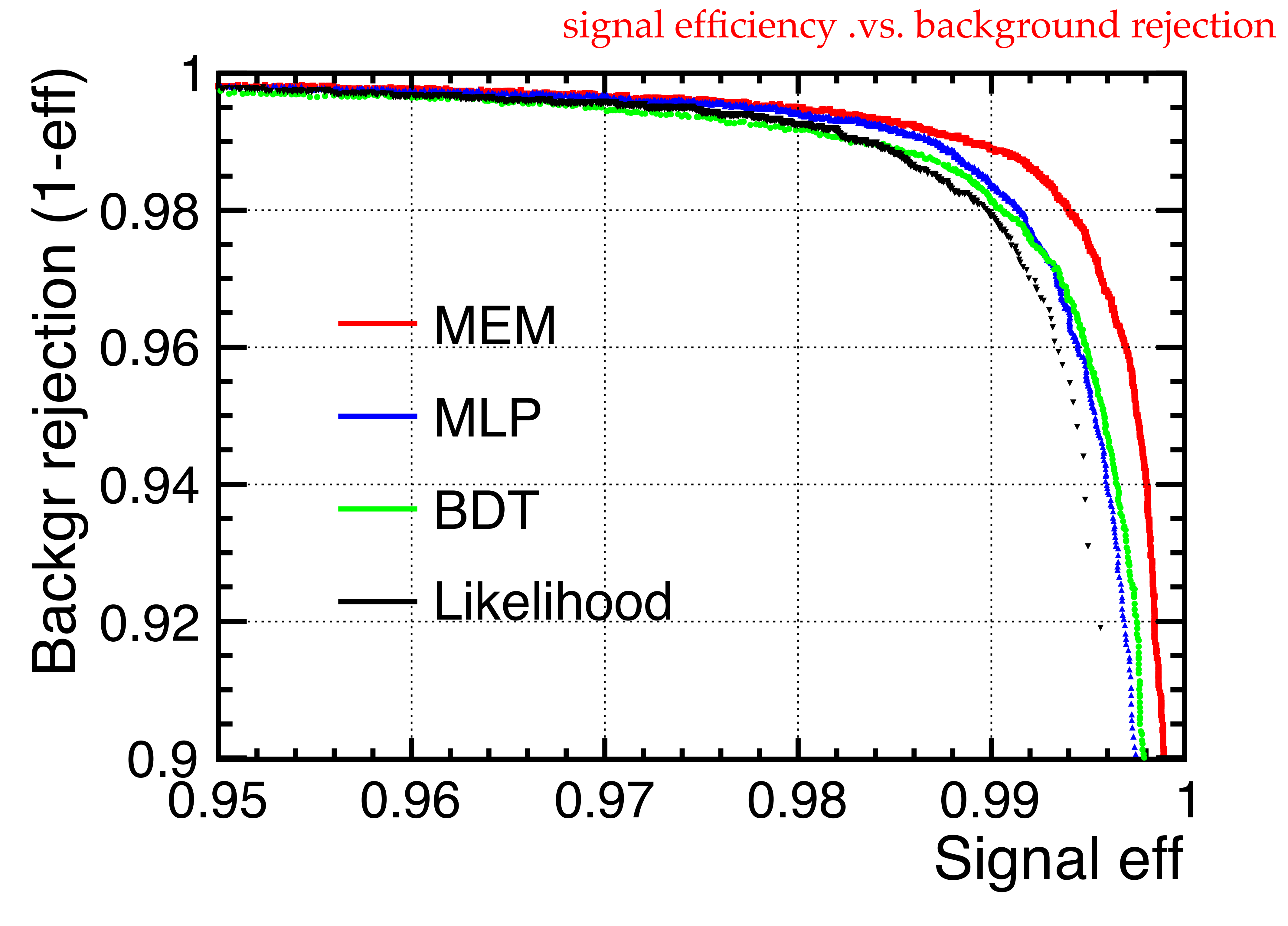
L$_0$: ME as from ZZ-fusion



L$_1$: ME as from ZH



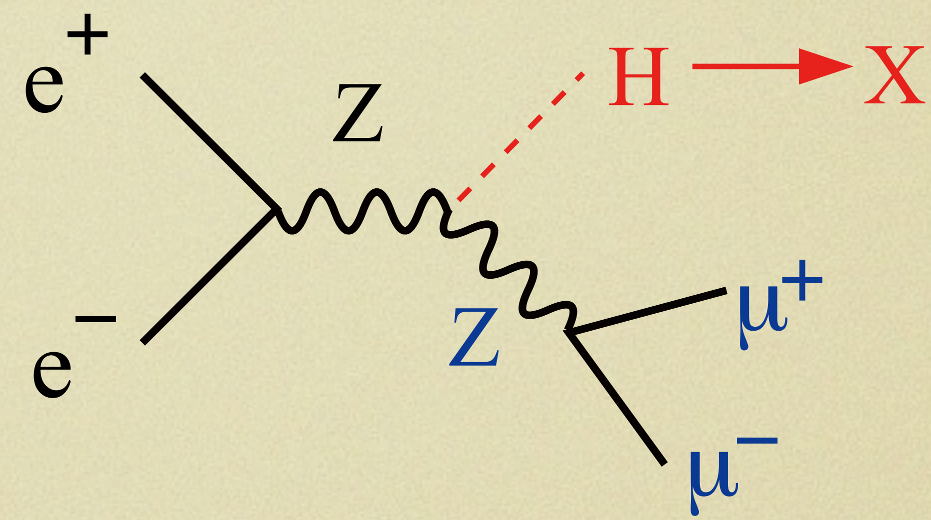
principle demonstrated: physicist .vs. statistician



signal: eeH via ZZ-fusion; background: eeH via ZH

apply for more critical measurement

recoil mass analysis

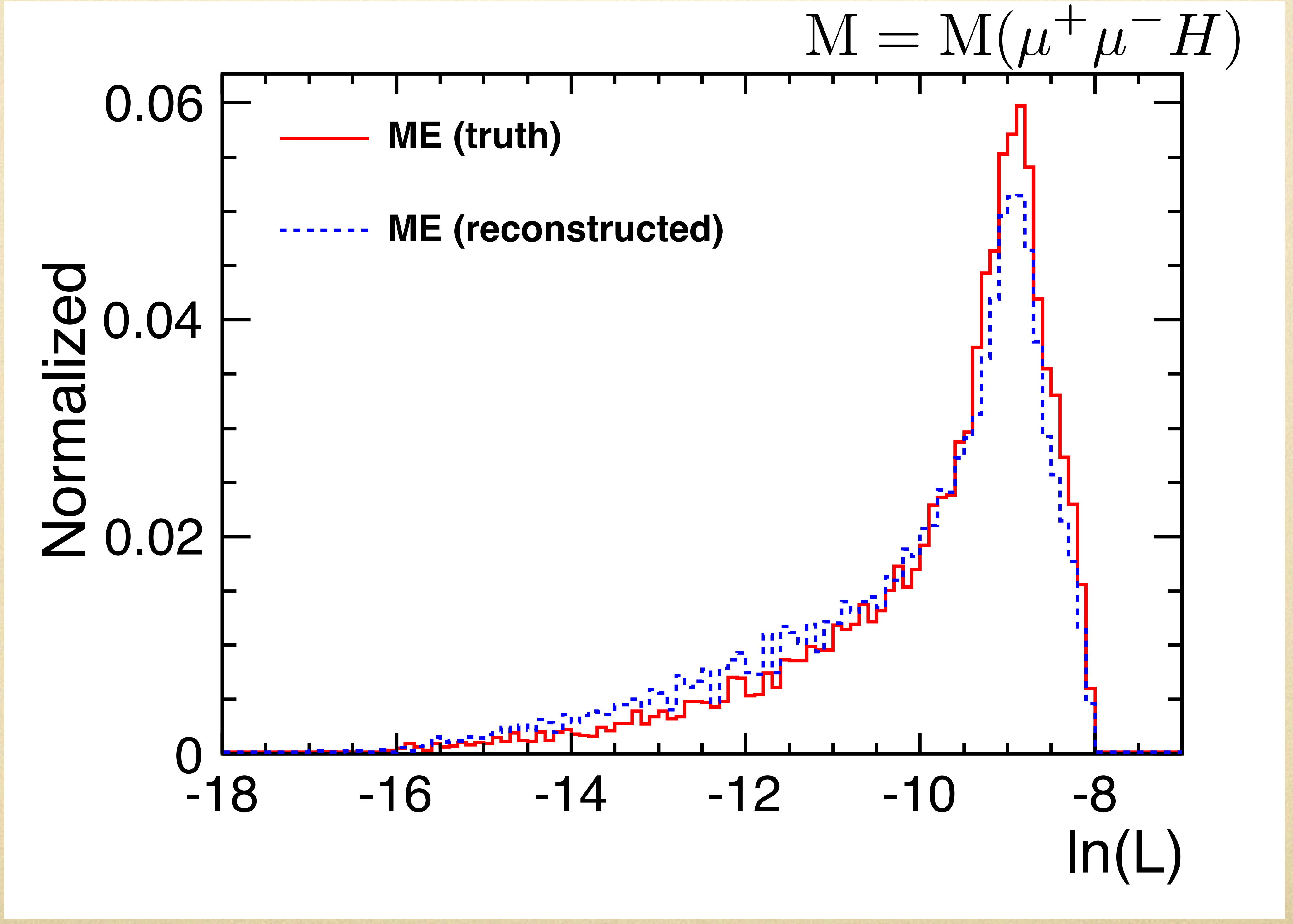


- momentum of muon are well measured
- since ECM is well defined, momentum of Higgs can be well determined as well
- therefore matrix element can be precisely calculated without assuming any Higgs decay!

$$\mathrm{M} = \mathrm{M}(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H}$$

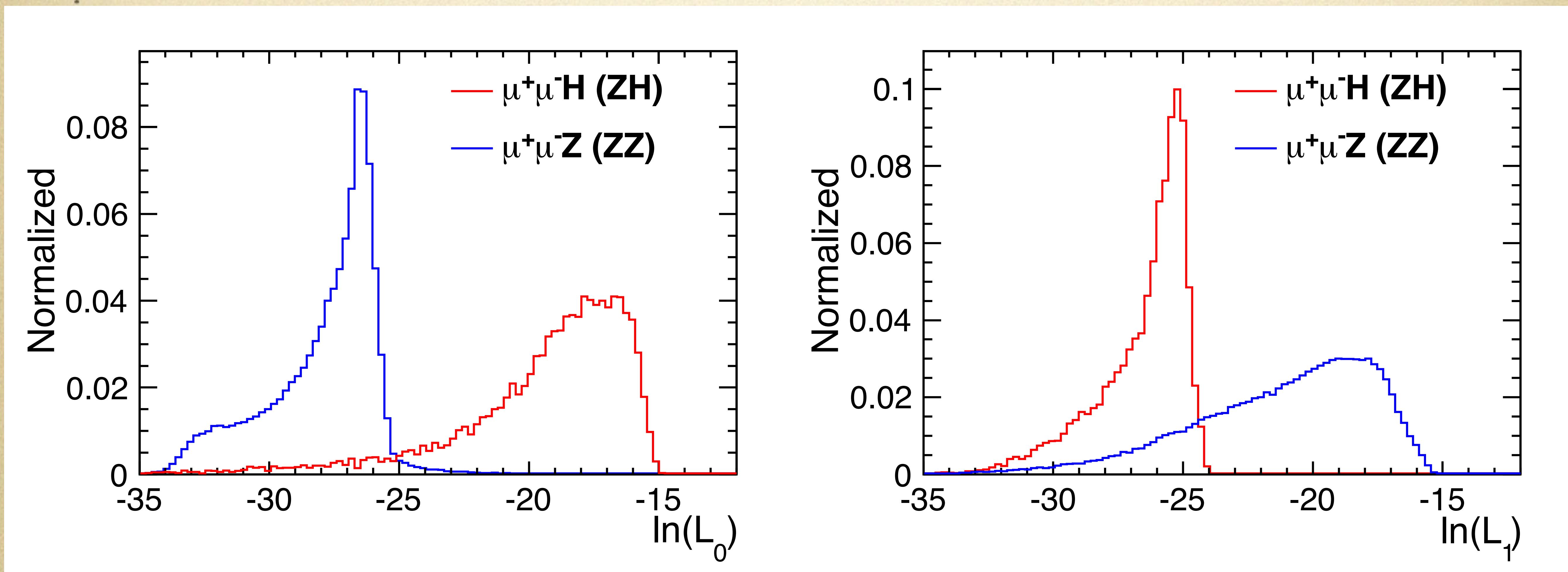
matrix element: reconstructed .vs. truth

(step-by-step test: samples are generated w/o ISR and beam strahlung)



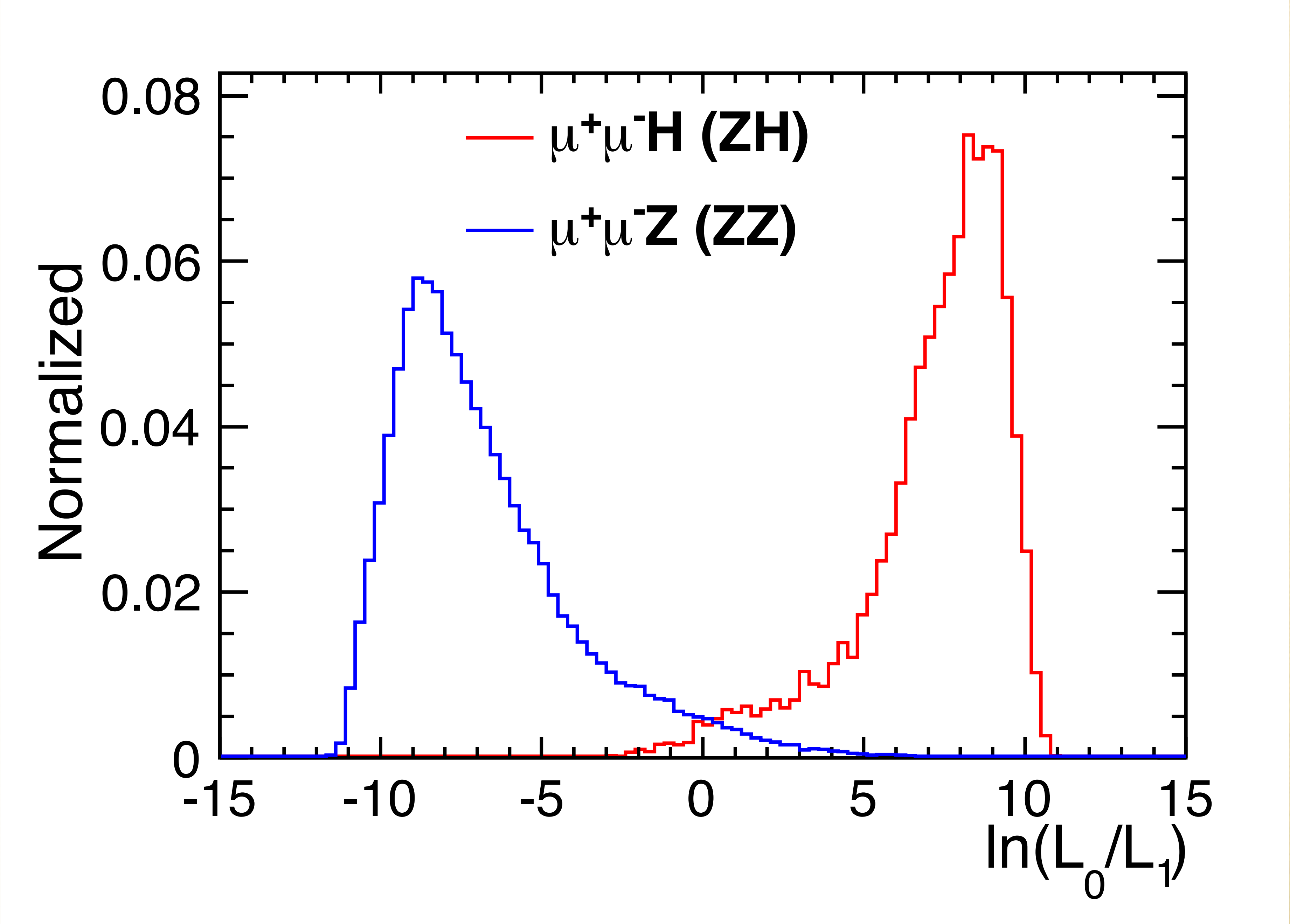
matrix element: signal .vs. background (ZZ)

$$\mathrm{M} = \mathrm{M}(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H} \qquad \mathrm{M} = \mathrm{M}(\mu^+\mu^- Z)\frac{1}{p_Z^2 - m_Z^2 + i m_Z \Gamma_Z}$$



matrix element: signal .vs. background (ZZ)

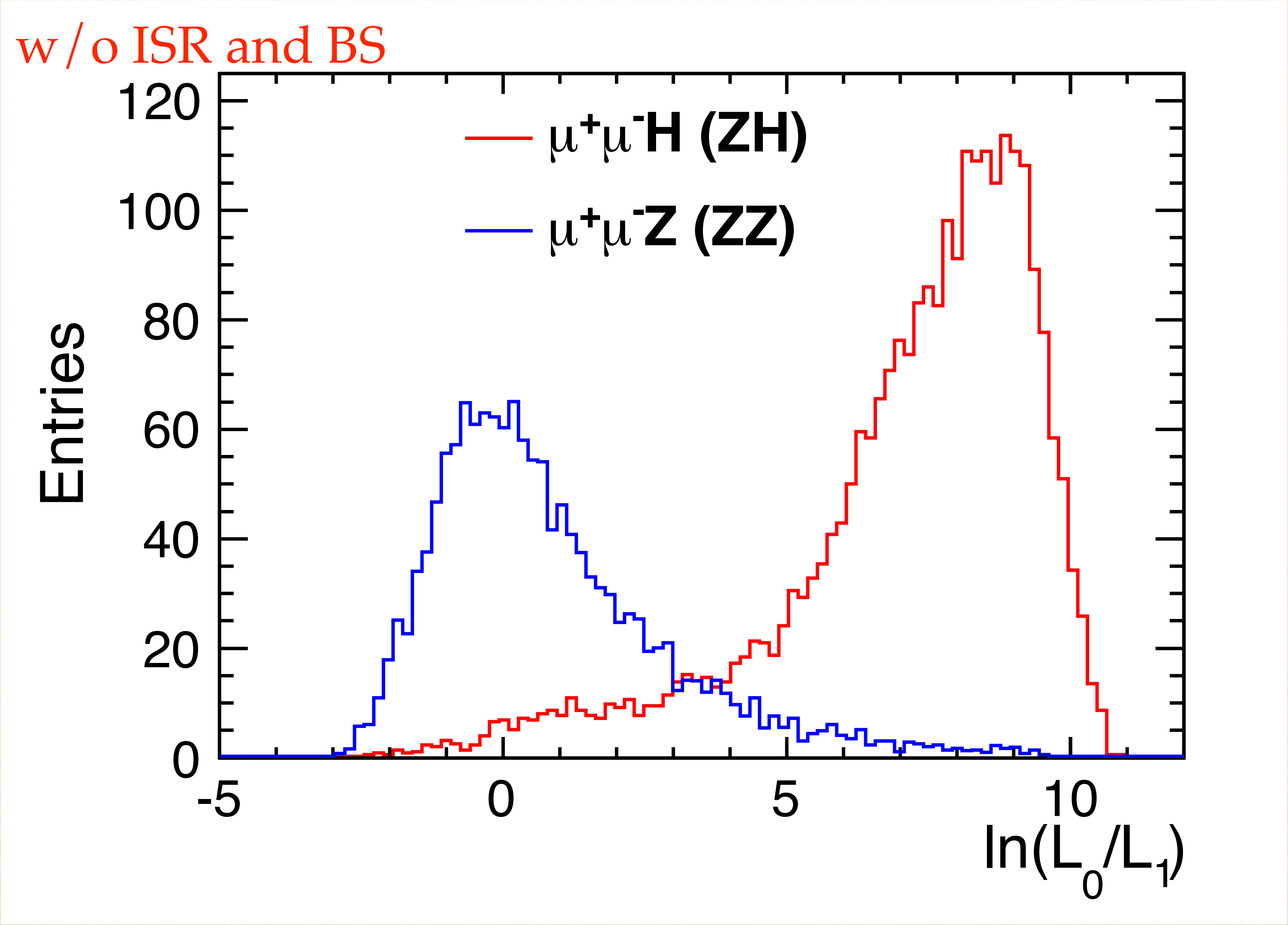
$$\mathrm{M} = \mathrm{M}(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H} \Big/ \mathrm{M} = \mathrm{M}(\mu^+\mu^- Z)\frac{1}{p_Z^2 - m_Z^2 + i m_Z \Gamma_Z}$$



matrix element: signal .vs. background (ZZ)

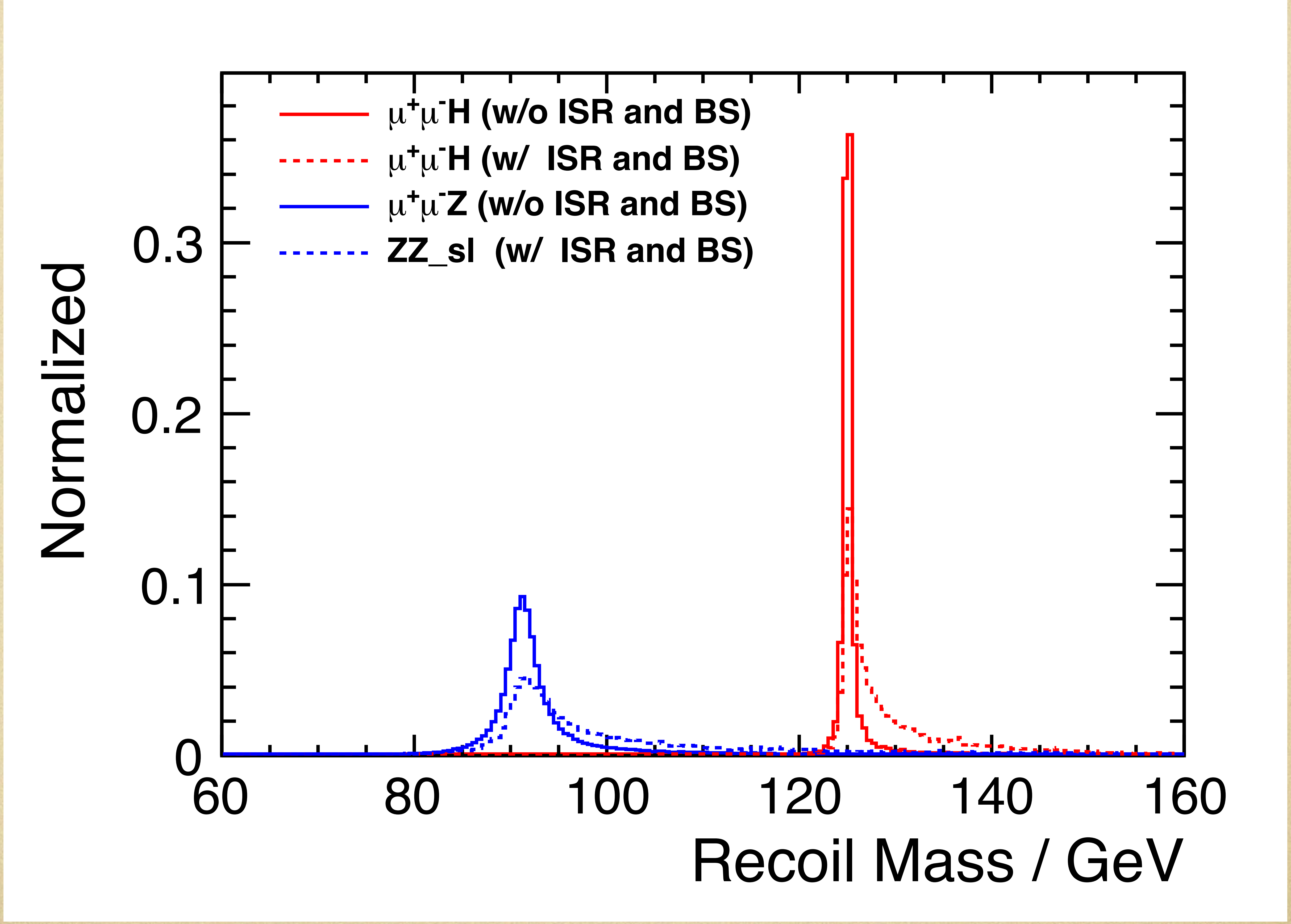
(normalized to expected number of events w/ 250 fb^{-1})

$$\mathrm{M} = \mathrm{M}(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H} \Bigg/ \mathrm{M} = \mathrm{M}(\mu^+\mu^- Z)\frac{1}{p_Z^2 - m_Z^2 + i m_Z \Gamma_Z}$$

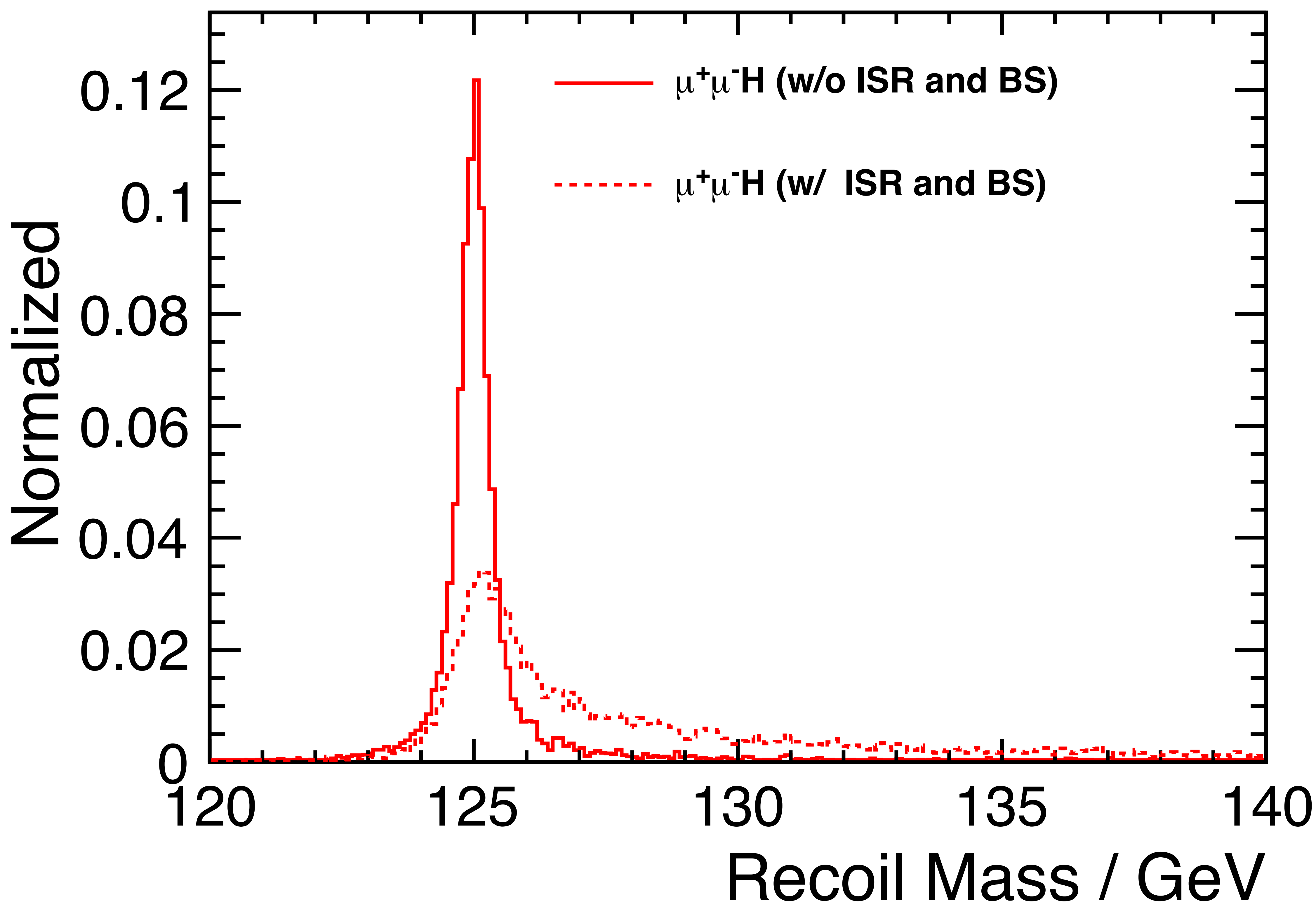


only cut on recoil mass > 110 GeV applied, already very well separated

including everything: ISR + beam spectrum



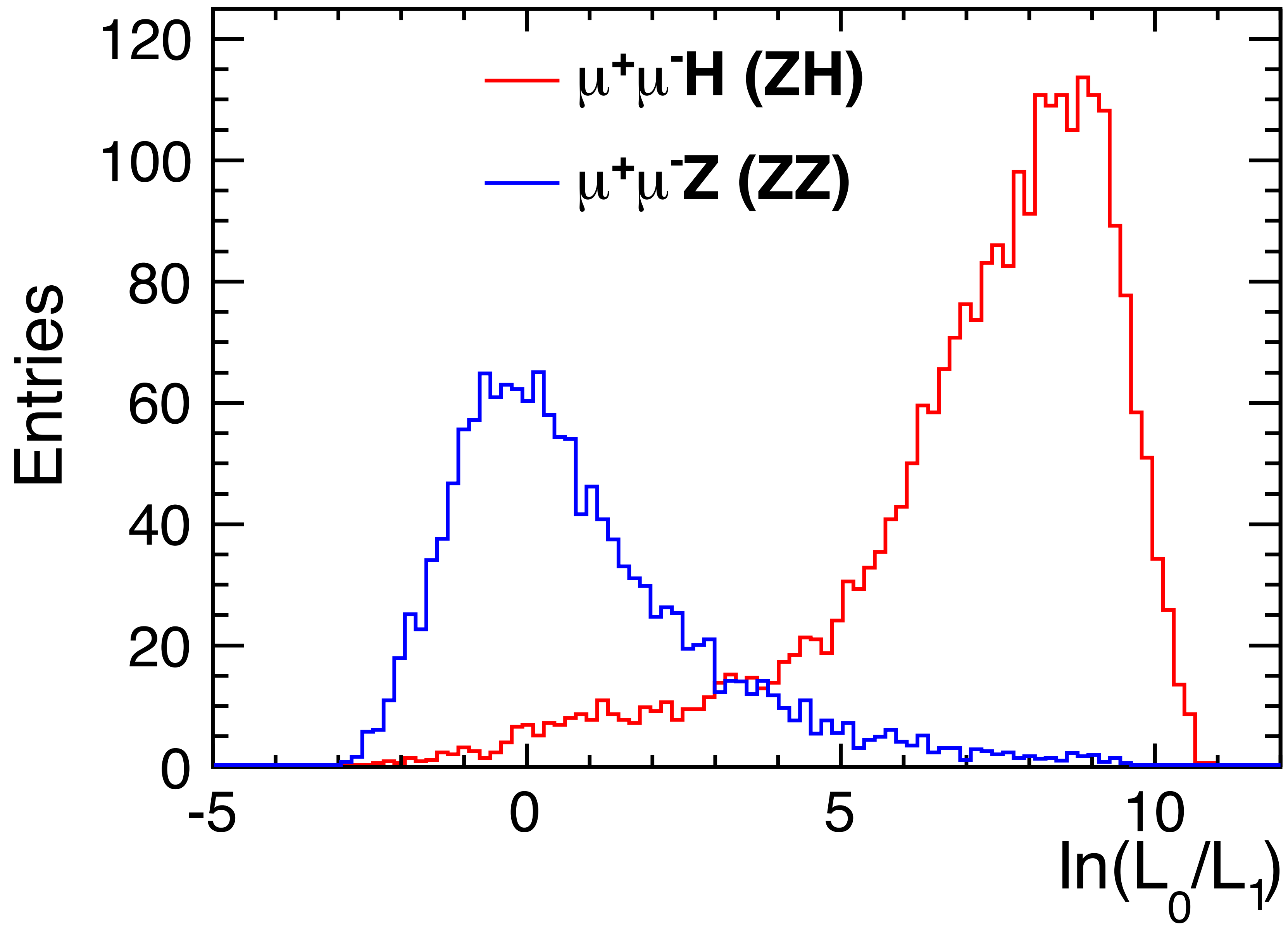
including everything: ISR + beam spectrum



matrix element: signal .vs. background (ZZ)

$$M = M(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H} \Big/ M = M(\mu^+\mu^- Z)\frac{1}{p_Z^2 - m_Z^2 + i m_Z \Gamma_Z}$$

w/o ISR and BS

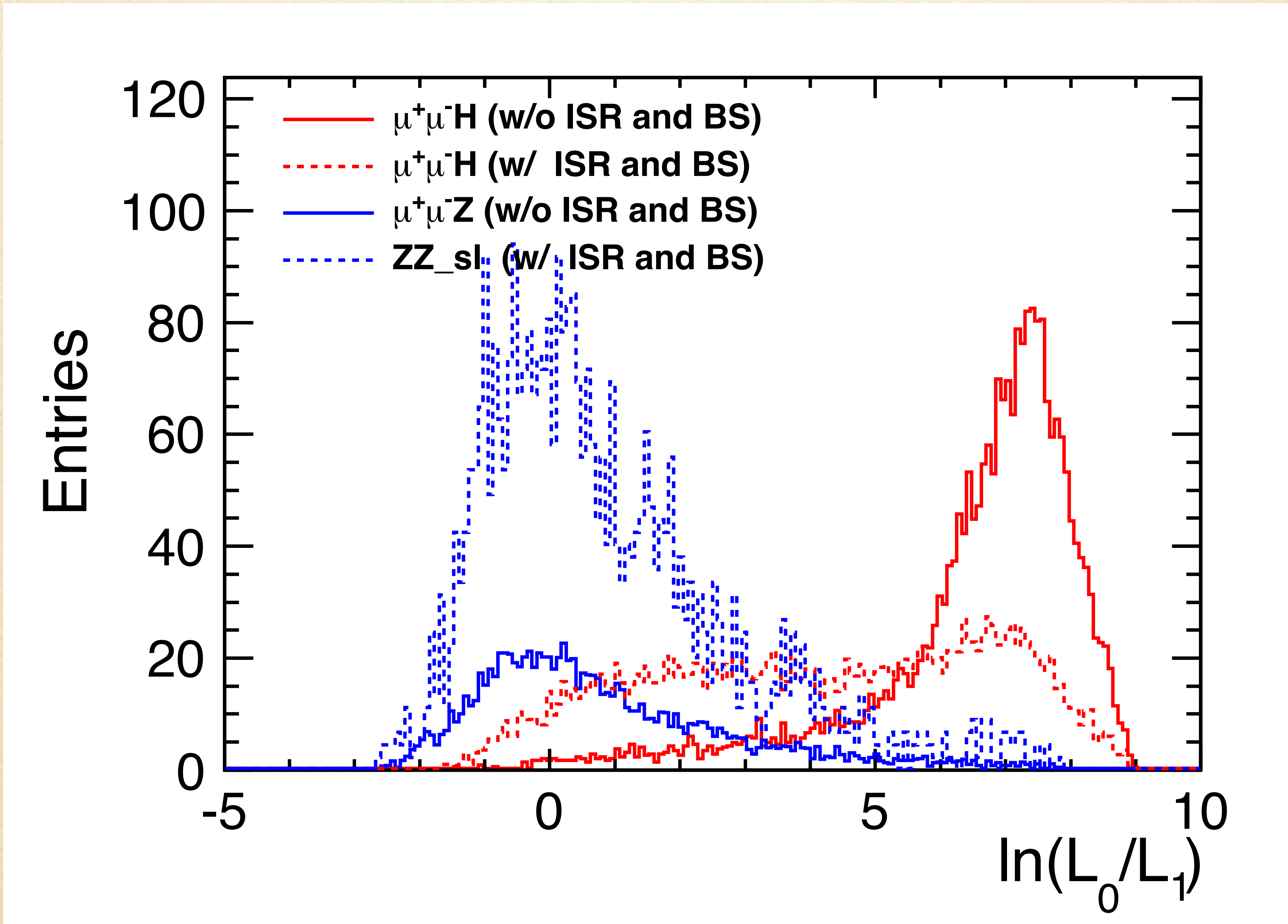


only cut on recoil mass > 110 GeV applied, already very well separated

matrix element: signal .vs. background (ZZ)

(including everything; normalized to expected number of events w/ 250 fb^{-1})

$$\mathrm{M} = \mathrm{M}(\mu^+\mu^- H)\frac{1}{p_H^2 - m_H^2 + i m_H \Gamma_H} \Big/ \mathrm{M} = \mathrm{M}(\mu^+\mu^- Z)\frac{1}{p_Z^2 - m_Z^2 + i m_Z \Gamma_Z}$$



ISR + BS significantly degraded the separation power against ZZ

proposal to remove ISR

- ☑ ISR enters detector: identification (see Tomita-san's study, eff ~ 90%)
- ☐ what if ISR goes to beam pipe? (dominant)

we can resolve it!

$$|P_z(\gamma)| = |P(\gamma)|$$

$$P_t(H) = P_t(Z)$$

$$P_z(H) = -(P_z(Z) + P_z(\gamma)$$

$$E(Z) + \sqrt{P_t^2(H) + P_z^2(H) + m^2(H)} + |P(\gamma)| = \sqrt{s}$$

→ $$P(\gamma) = \frac{s - 2\sqrt{s}E(Z) + m^2(H) - m^2(Z)}{2(P(Z) \pm (\sqrt{s} - E(Z)))}$$

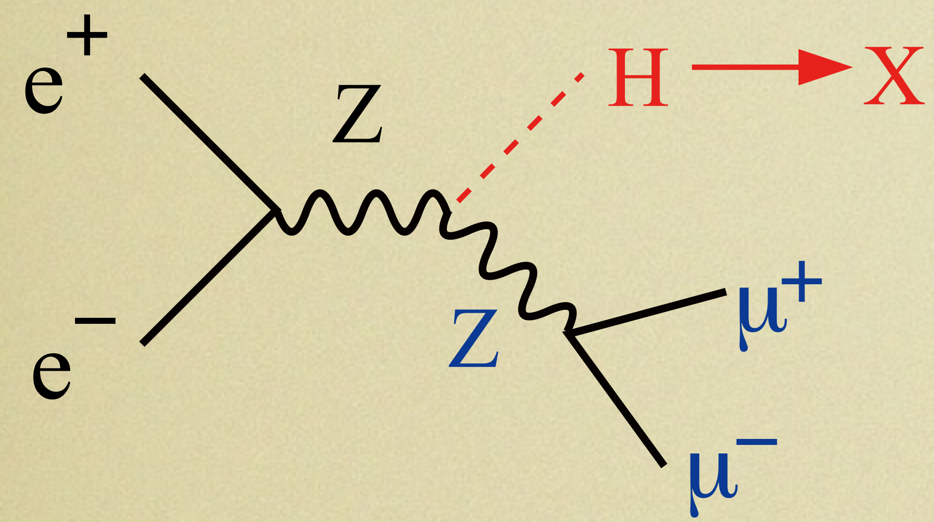
summary and next step

- ME tools have been developed and verified by first look.
- ME indeed can be reconstructed very precisely at the ILC.
- selection will be much simplified using MEM.
- to use Higgs propagator, in principle we need know Higgs mass first, which can be obtained by fitting recoil mass (note: to get mass more cuts on decay part can be applied); to improve $\sigma(ZH)$ measurement would then be main purpose of using MEM.
- complete analysis is needed to incorporate MEM for recoil mass study, propose to be done by Watanuki-san, (including CP study)
- going to have a look at next application for Higgs self-coupling

back up

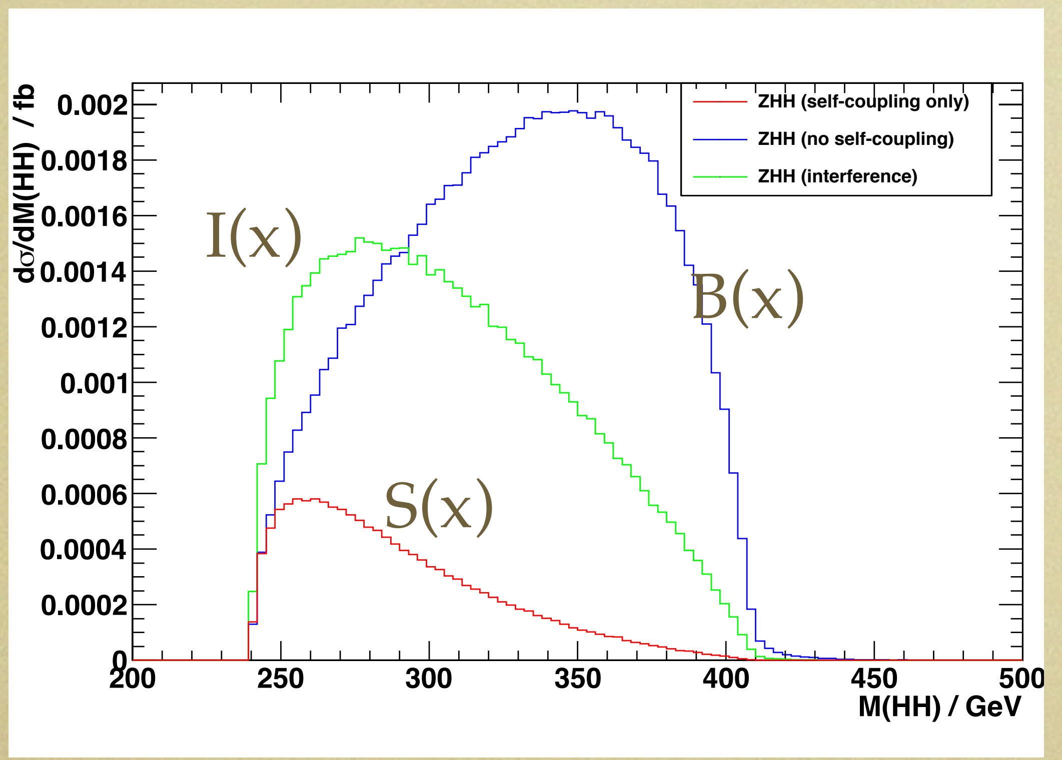
more potential applications:

recoil mass analysis



- momentum of muon are well measured
- since ecm is well defined, momentum of Higgs can be well measured as well
- hence matrix element can be well calculated without assuming any Higgs decay!

Higgs self-coupling



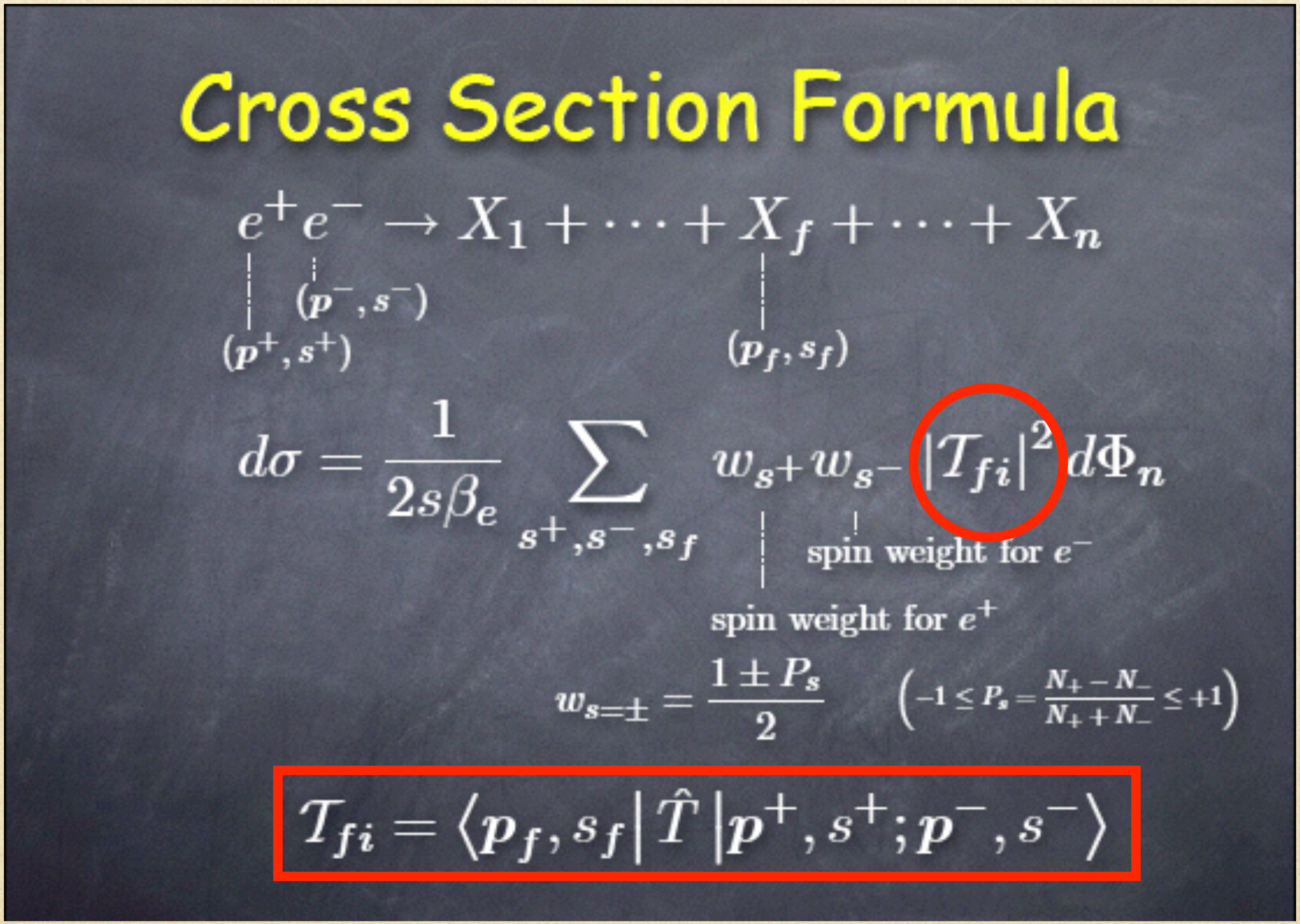
$$\frac{\mathrm{d}\sigma}{\mathrm{d}x} = B(x) + \lambda I(x) + \lambda^2 S(x)$$

$$w_0(x) = c \cdot \frac{I(x) + 2S(x)}{\sigma(x)}$$

what if we use ME as weighting?

What is Matrix Element (Amplitude)

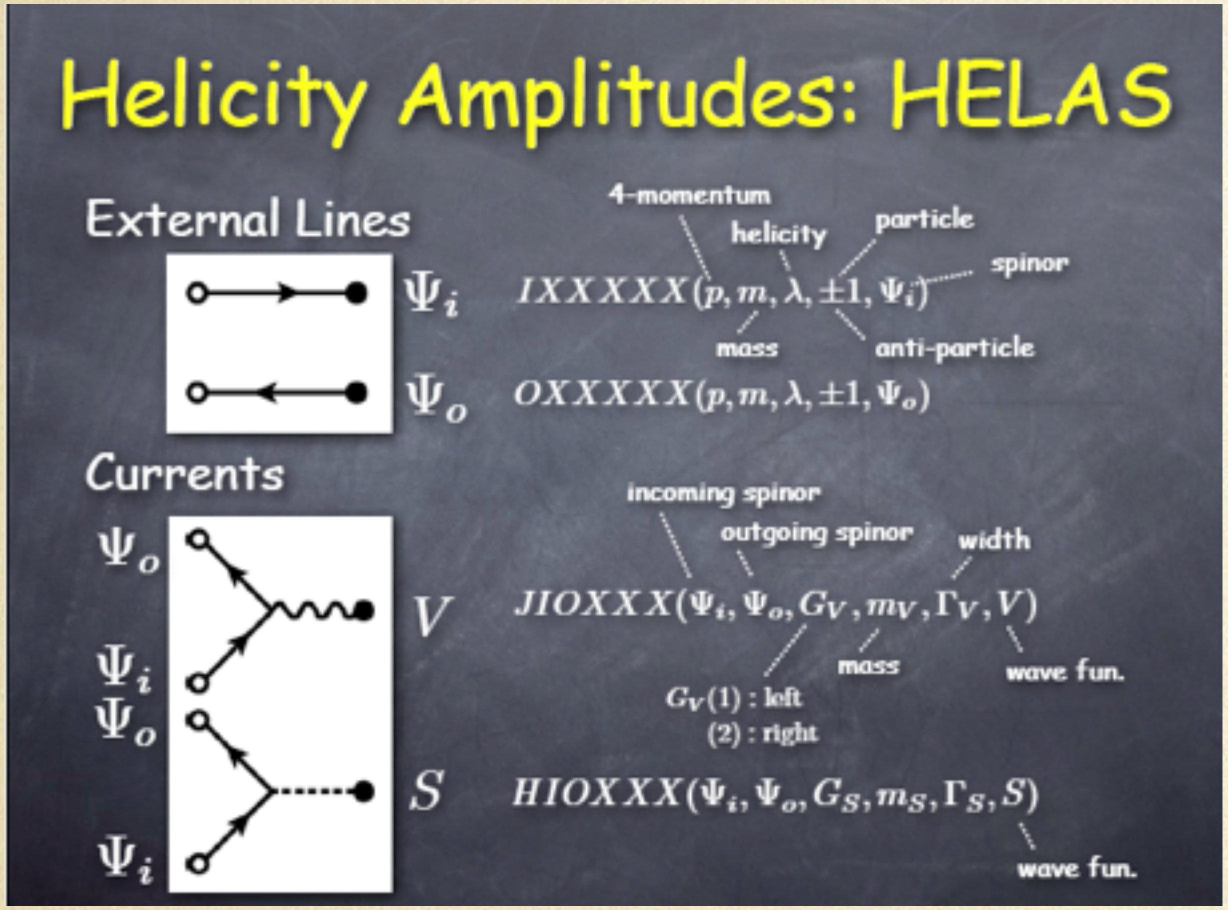
(squared ~ differential cross section)

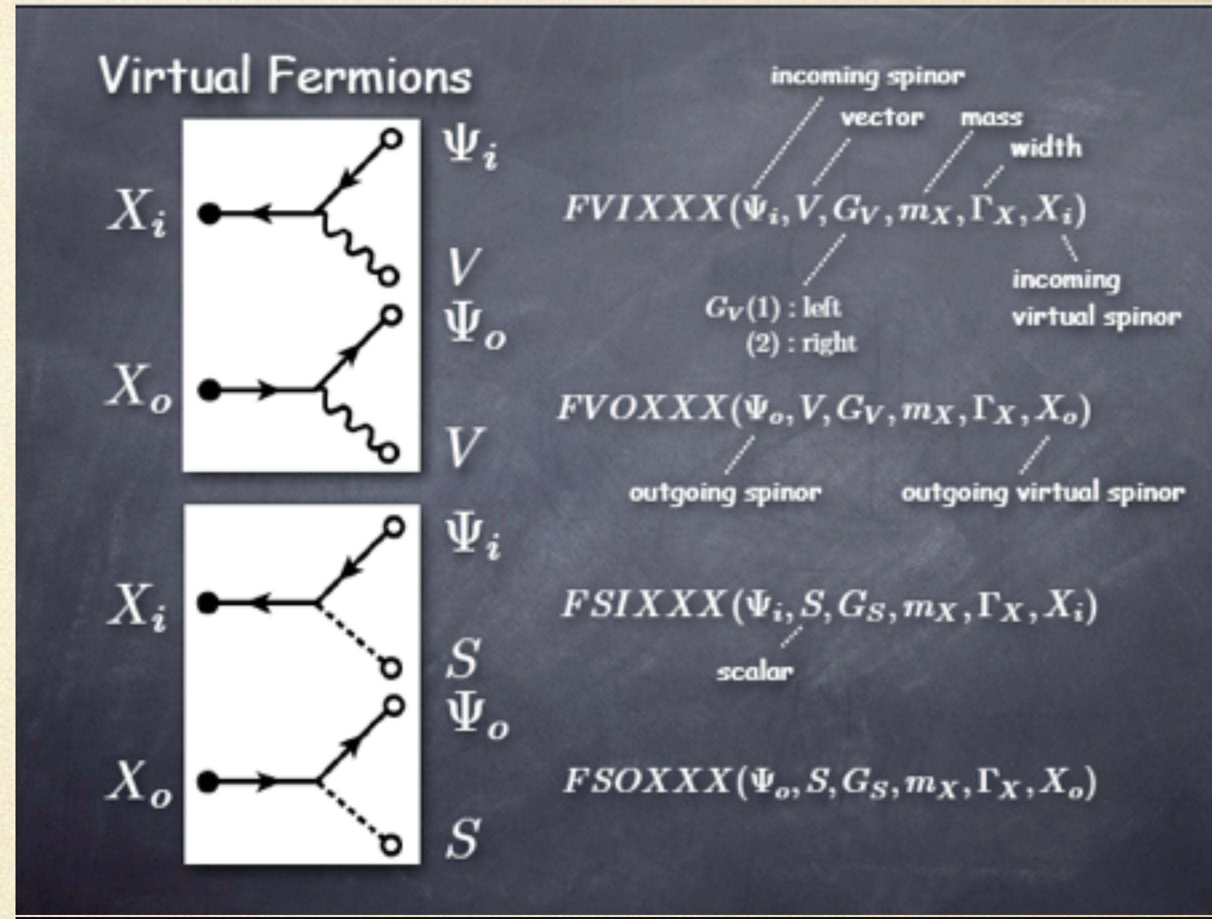


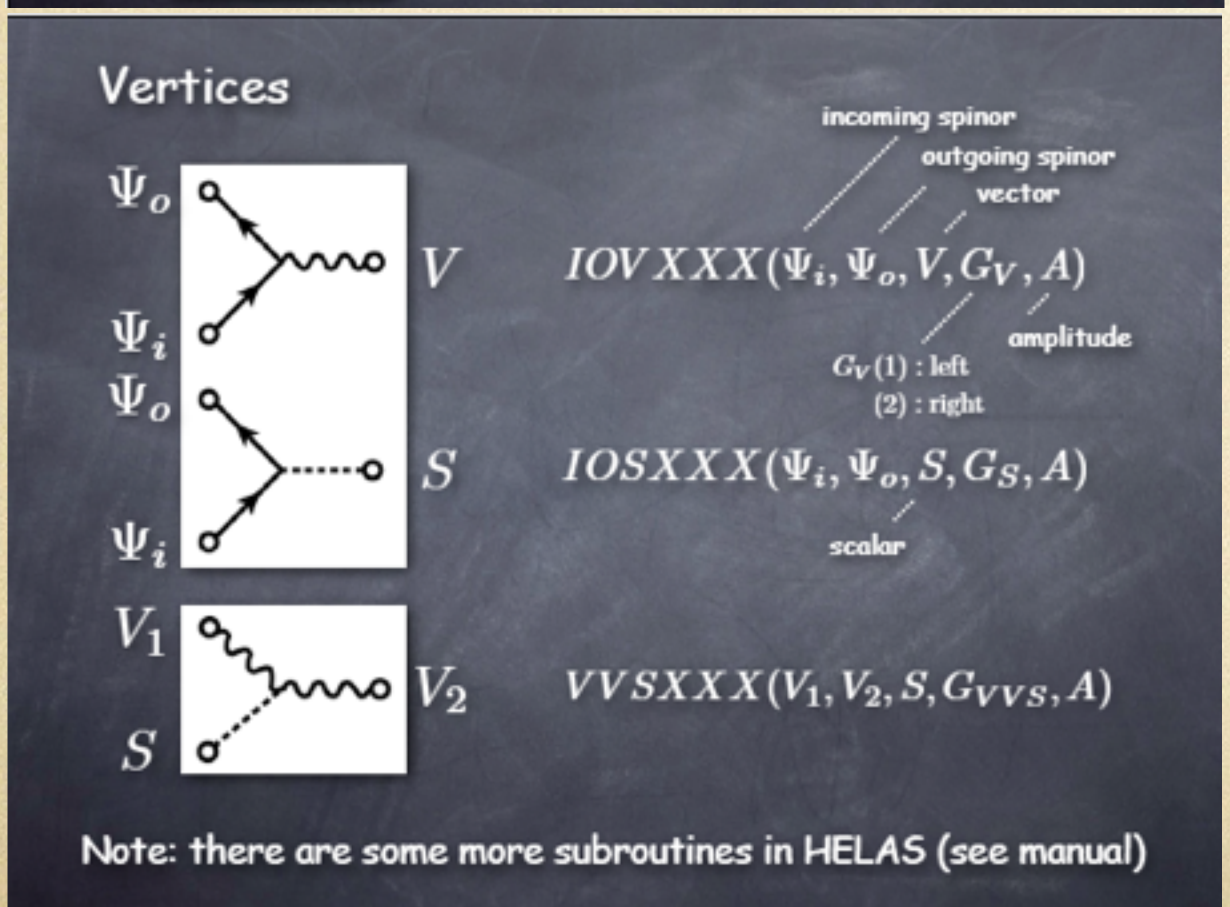
(technically, $|\mathrm{ME}|^2$ is the weight of each phase space point)

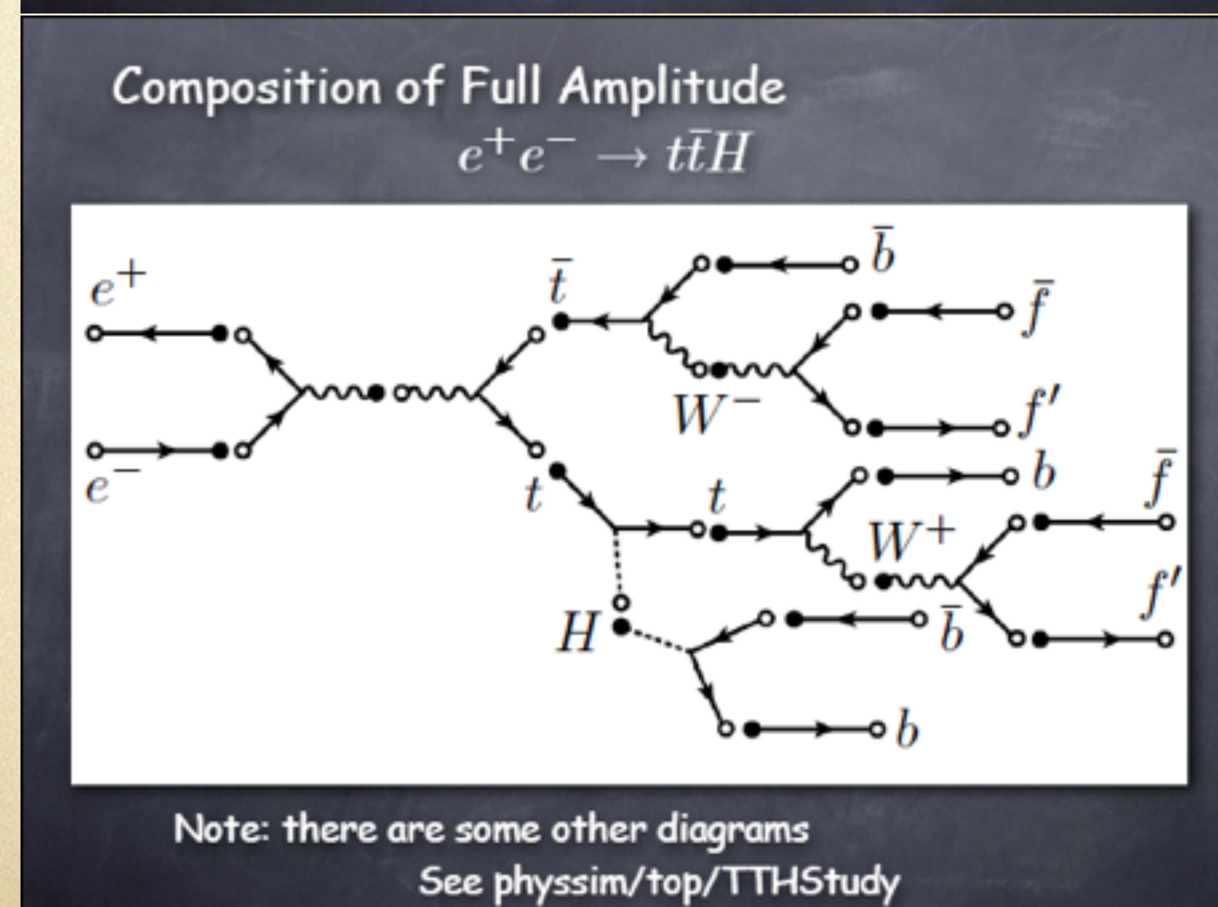
tool to calculate ME — HELAS (one of many others)

original fortran version by H. Murayama, etc.; c++ version by K. Fujii









development of ME tools for physics analysis: status

(for full detector simulation, within Marlin framework)

- basic idea: provide some libraries to calculate ME for given processes, which can be directly used in analyses by just passing four momentum.
- core libraries implemented: HELLib (C++ HELAS, helicity amplitude subroutines for feynman diagrams).
- LCME (linear collider matrix element libs), so far major Higgs production implemented: LCMEZH, LCMENNH, LCMEEEH, LCMEZHH, LCMENNHH, LCMEEEZ.
- verified by using MC truth information.
- Physsim v1.0 released, available now on svn.
- typical way to use it: check out; compile; include libPhyssim.so in your $MARLIN_DLL; using namespace lcme; follow example marlin processor (included in the package).

svn co https://svnsrv.desy.de/basic/physsim/Physsim/trunk

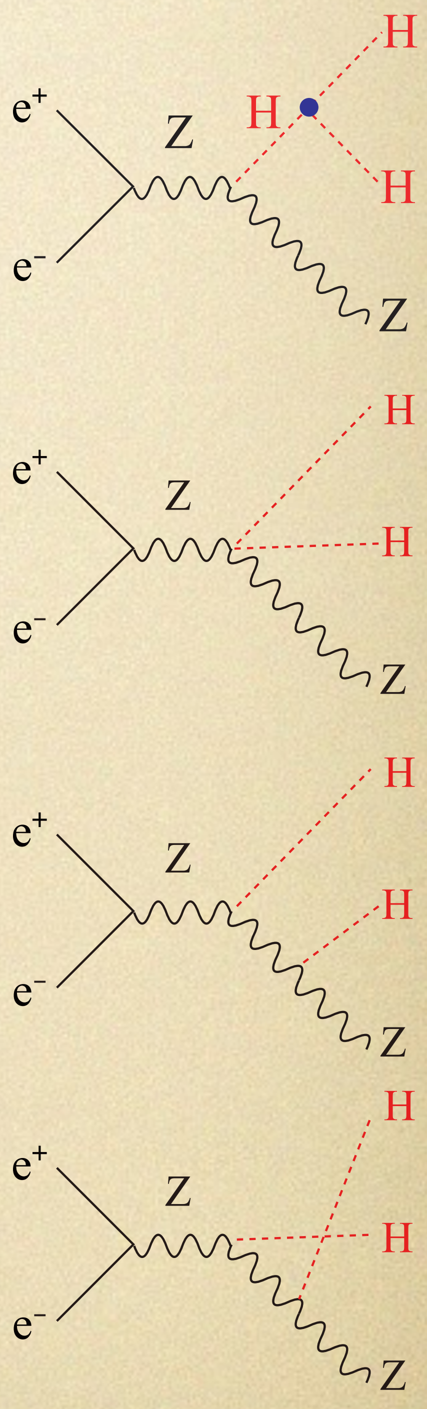
to be implemented: Detector Effect

- the four momenta we measured have resolution —> we need detector transfer function (jet-energy resolution, momentum resolution, etc.) and integrate all possible truth four momenta.
- some four momenta can not be measured (missing neutrinos) —> integrate all possible truth four momenta.

$$L(\mathbf{p}_i^{\mathrm{vis}}|\mathbf{a}) = \frac{1}{\sigma_{\mathbf{a}}}[\prod_{j\in\mathrm{inv.}}\int\frac{d^3p_j}{(2\pi)^3 2E_j}][\prod_{k\in\mathrm{vis.}}\int\frac{d^3p_k}{(2\pi)^3 2E_k}W_i(\mathbf{p}_i^{\mathrm{vis}}|p_k,\mathbf{a})]|M(p_j,p_k;\mathbf{a})|^2$$

example core code to calculate matrix element

```
//----------------------------
// Double Higgs Production Amplitude
//———————————————
HELFermion em(fK[0], kM_e, fHelInitial[0], +1, kIsIncoming);
HELFermion ep(fK[1], kM_e, fHelInitial[1], -1, kIsOutgoing);
HELScalar  h1(fP[0]);
HELScalar  h2(fP[1]);
HELFermion f (fP[2], fM[2], fHelFinal [2], +1, kIsOutgoing);
HELFermion fb(fP[3], fM[3], fHelFinal [3], -1, kIsIncoming);

Double_t v     = 2.*kM_w/kGw;
Double_t ghhh  = -TMath::Power(fMass,2)/v*3.;
Double_t gzzh  = kGz*kM_z;
Double_t gzzhh = kGz*kGz/2.;

HELVector zf(fb, f, glz, grz, kM_z, gamz);
HELVector zs(em, ep, glze, grze, kM_z, gamz);

HELScalar hh(h1, h2, ghhh, fMass, 0.);
HELVertex amp1(zs, zf, hh, gzzh);          // HHH self-coupling

HELVertex amp2(zs, zf, h1, h2, gzzhh);     // ZZHH 4-point

HELVector vz1(zf, h1, gzzh, kM_z, gamz);
HELVertex amp3(zs, vz1, h2, gzzh);         // double H-strahlung

HELVector vz2(zf, h2, gzzh, kM_z, gamz);
HELVertex amp4(zs, vz2, h1, gzzh);         // double H-strahlung

Complex_t amp  = amp1 + amp2 + amp3 + amp4;
```



example code in your marlin processor

```
  // initialize LCMEZHH with Higgs mass of 125 GeV and beam
polarisations P(e-,e+) = (0.,0.)
  _zhh = new LCMEZHH("LCMEZHH","ZHH",125.,0.,0.);
  // set mode of Z decay
  _zhh->SetZDecayMode(5);

  // ----------------------------------
  // calculate the matrix element
  // ----------------------------------
  // put four-momenta of final states to an array
  TLorentzVector vLortzMC[4] = {lortzLep1MC, lortzLep2MC, lortzH1MC, lortzH2MC};
  // pass kinematics to ME object
  _zhh->SetMomentumFinal(vLortzMC);
  // matrix element can be given for each combination of initial and final helicities
  Int_t vHelLL[2] = {-1,-1};
  Int_t vHelLR[2] = {-1,1};
  Int_t vHelRL[2] = {1,-1};
  Int_t vHelRR[2] = {1,1};
  Double_t dSigmaLL = _zhh->GetMatrixElement2(vHelLL);
  Double_t dSigmaLR = _zhh->GetMatrixElement2(vHelLR);
  Double_t dSigmaRL = _zhh->GetMatrixElement2(vHelRL);
  Double_t dSigmaRR = _zhh->GetMatrixElement2(vHelRR);
  // if no combination of helicities specified, final combinations are summed
  // and initial combinations are weighted by beam polarisations
  Double_t dSigma   = _zhh->GetMatrixElement2();
  // that's all need to do to get matrix elment for each event
  // ----------------------------------
```

verification

$$L(\mathbf{p}_i^{\mathrm{vis}}|\mathbf{a}) = \frac{1}{\sigma_{\mathbf{a}}}[\prod_{j\in\mathrm{inv.}}\int\frac{d^3p_j}{(2\pi)^3 2E_j}][\prod_{k\in\mathrm{vis.}}\int\frac{d^3p_k}{(2\pi)^3 2E_k}W_i(\mathbf{p}_i^{\mathrm{vis}}|p_k,\mathbf{a})]|M(p_j,p_k;\mathbf{a})|^2$$

- stdhep events are generated without any ISR and BS, helicity combinations of both initial and final states can be controlled.
- detector transfer function ($W(p_i | p_k,a)$) becomes a delta function, and no invisible variables. (test with MC information).
- ME calculated for each event in this way should be as same as the ME used in event generation (as event weights).
- to verify the calculated ME, if each event is weighted by ($1./|ME|^2$), all variables should be uniformly distributed.