The mini-DST: a high-level LCIO format

Shin-ichi Kawada (DESY)

LCWS2021 @ Online

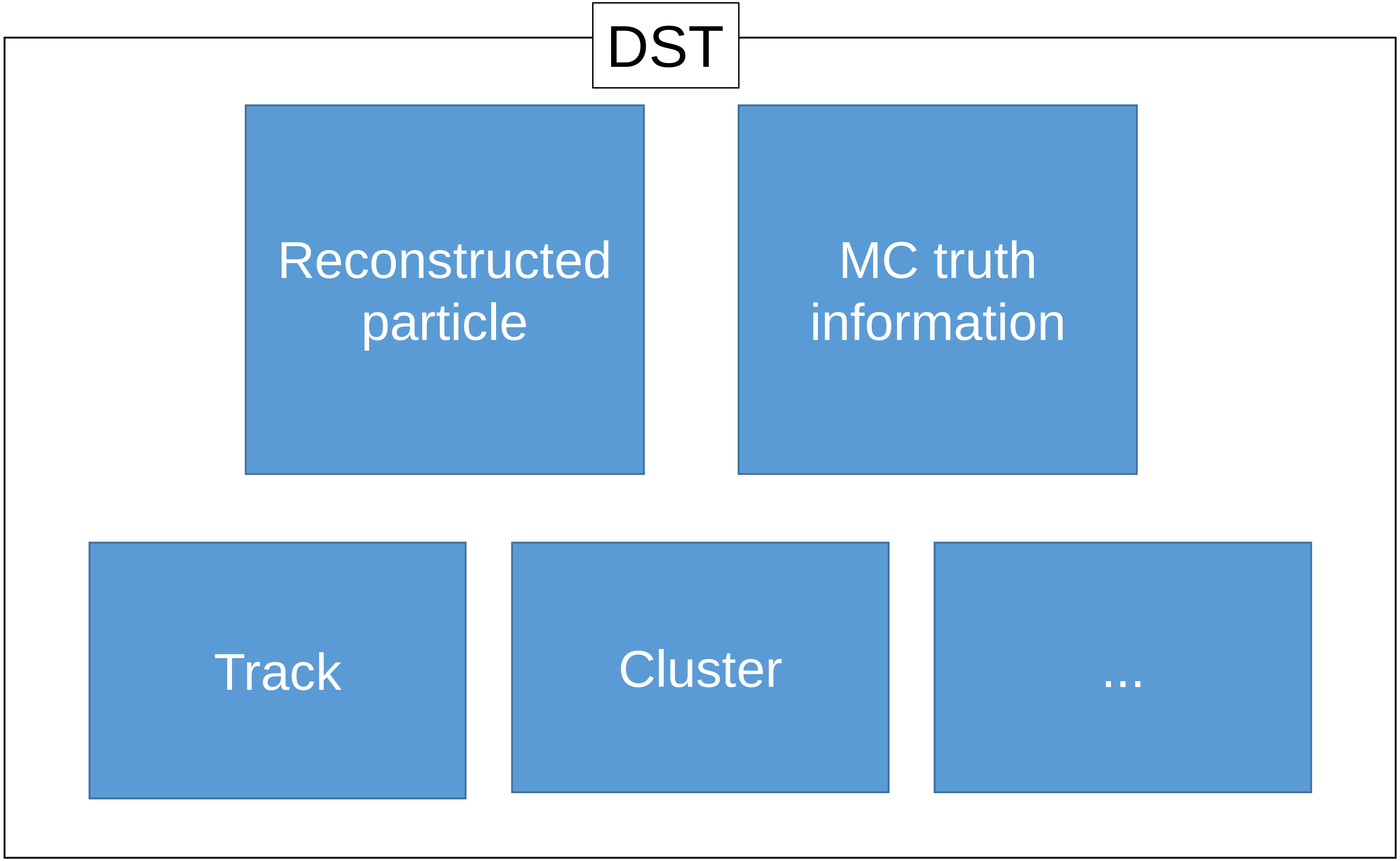
2021/March/15 - 18

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Introduction (1)

- The mini-DST is originally triggered by Snowmass 2021 study.
- In general, we should use MC samples with full simulation (or reliable fast simulation) when available to increase realism of the analysis.
- In ILD MC production, events are stored in DST format.
- However, DST contains a lot of collections.
  - MC samples produced for ILD-IDR: more than 20 collections
- These are maybe too much complicated, especially for beginners and newcomers.


DST
Reconstructed particle
MC truth information
Track
Cluster
...

Introduction (2)

- This mini-DST project is to reduce such complexity for beginners and provide the starting point for the analysis.
- Idea: provide simple data format with high-level object and analysis environment without installing iLCSoft
  - high-level object: isolated objects, jets, ...
  - directly readable in ROOT
  - can be applied commonly for both full/fast simulation
  - allows seamless transition to full output format of the ILD/SiD full simulations

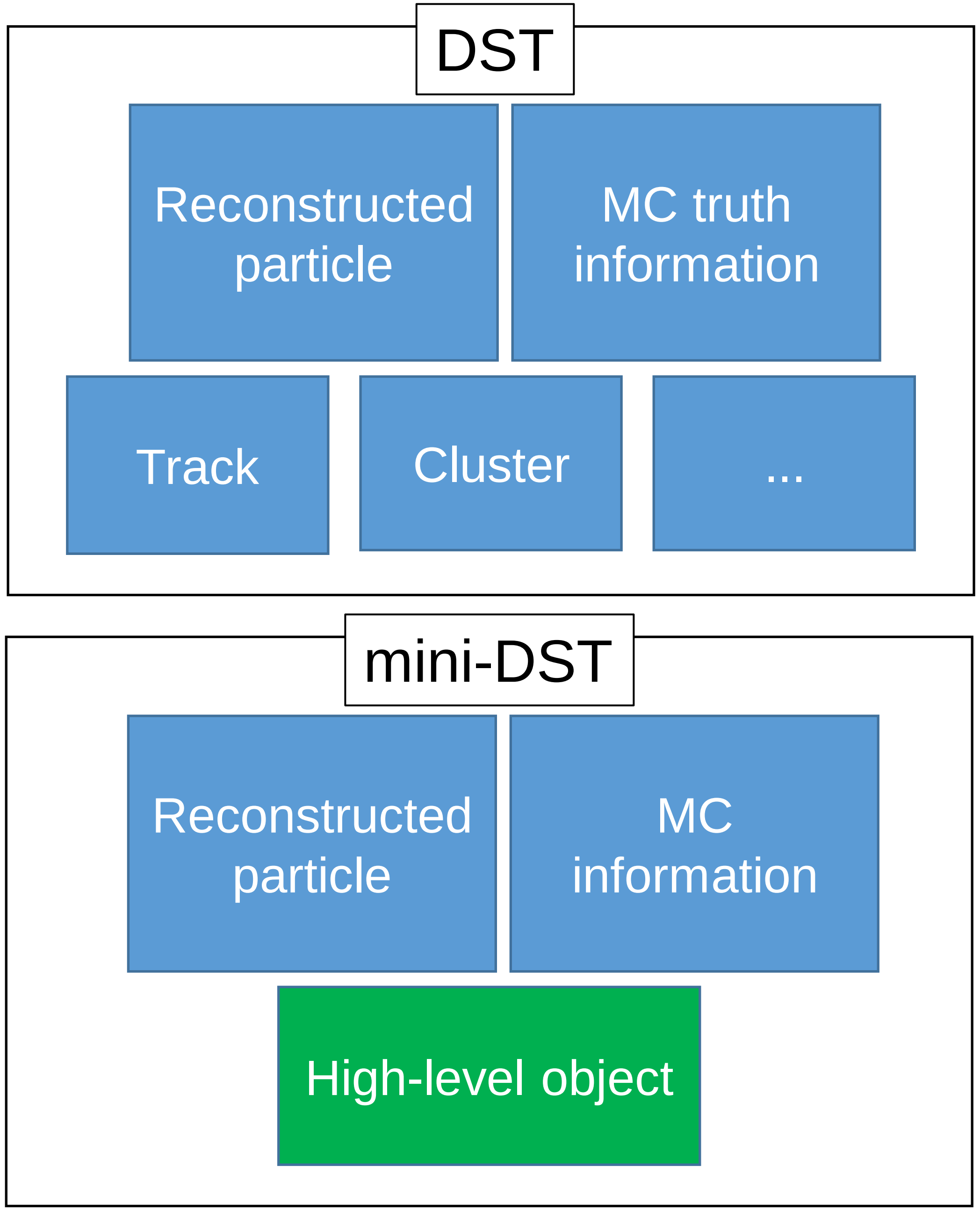
mini-DST in a nutshell

DST input file

Add high-level object
event shape, isolated objects, jets, flavor tagging,,,
(with typical parametrization)

Control output collection

mini-DST only selected collections
~30-50% file size



Details of mini-DST

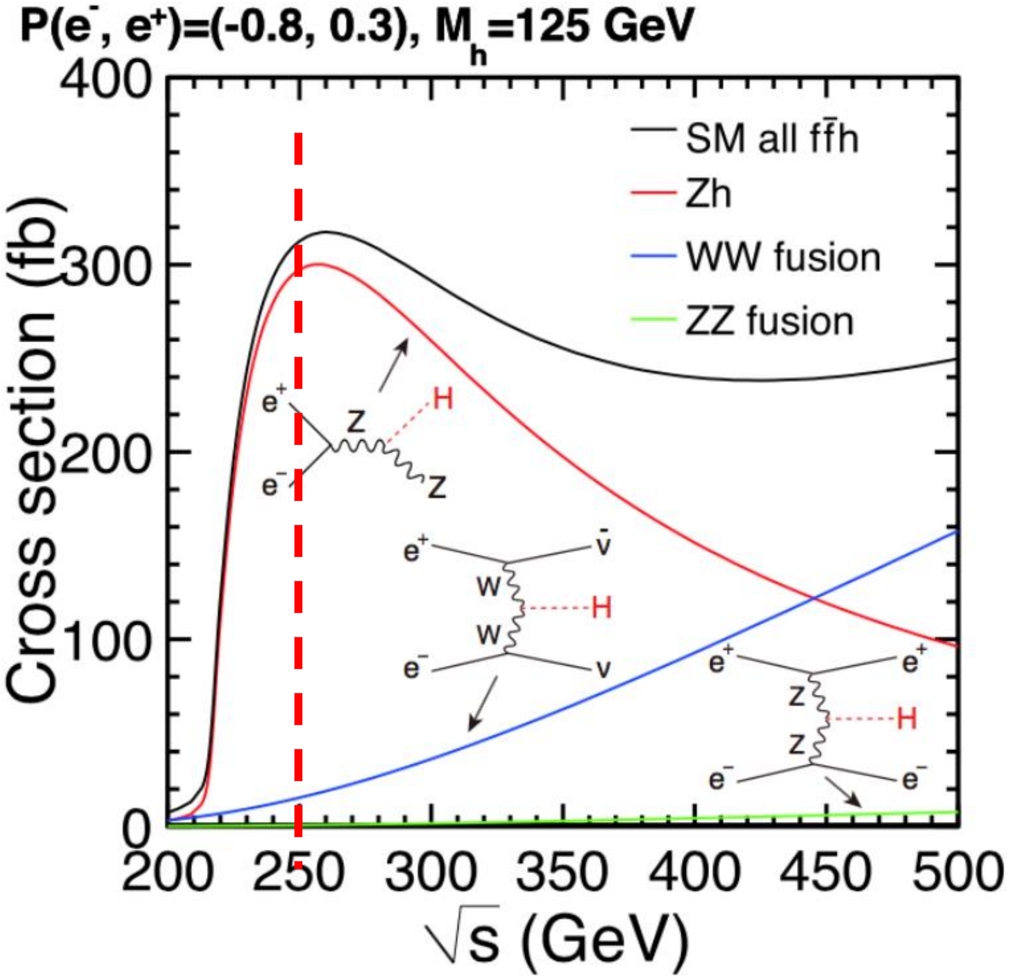
- Particle flow object (reconstructed particle)
  - including event shape variables, proton/kaon/pion ID, particles reconstructed with forward detectors
- MC truth information, links between reconstructed particle and MC truth
- High-level object
  - Isolated electrons/muons/taus/photons
  - Jets (N = 2, 3, 4, ... up to 10)
    - Durham algorithm applied to remained particle flow objects
  - Flavor tagging
    - Sophisticated MVA approach, giving three output classifications (b-, c-, other tag)

How to use mini-DST

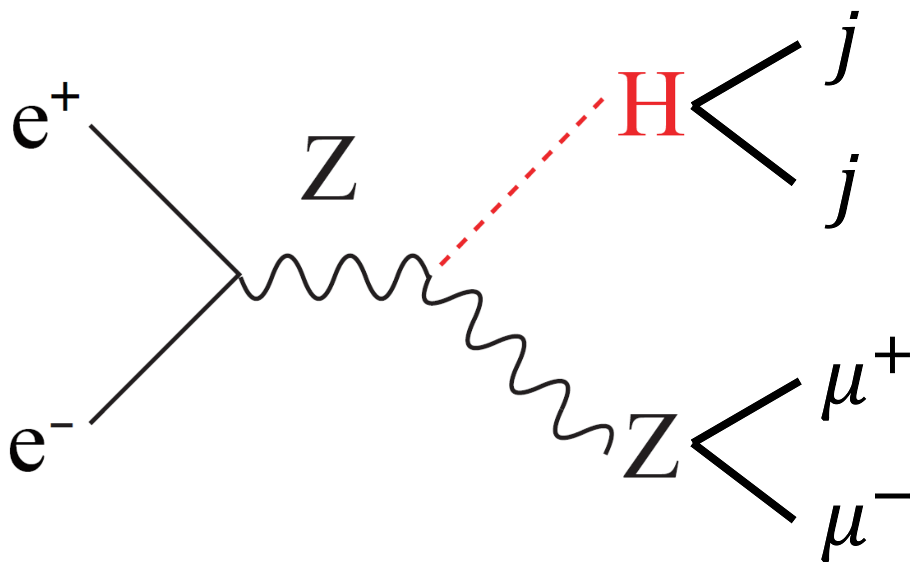
- We developed simple macro to create histograms.
  - Only require ROOT environment and LCIO (Linear Collider I/O) libraries. You don't need to install entire iLCSoft!

```
/*
put this into your .rootlogon.C file

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

for the LCIO API documentation see:
http://lcio.desy.de/v02-09/doc/doxygen_api/html/index.html
*/
```

Example: Recoil mass at ILC250



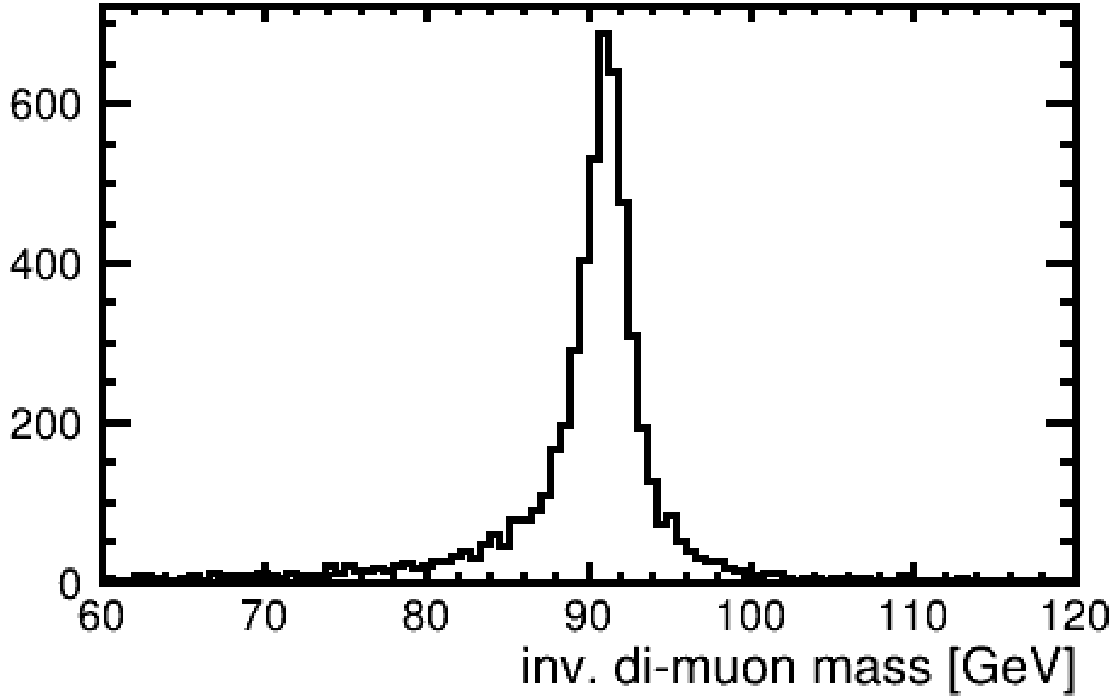
$$e^+e^- \to ZH \to (\mu^+\mu^-)(jj)$$



$M_H^2 = \left(p_{\text{initial}} - (p_{\mu^+} + p_{\mu^-})\right)^2$ (p: four-momentum)

- known initial state
- recoil technique: only need to measure muons

Example: Recoil mass at ILC250

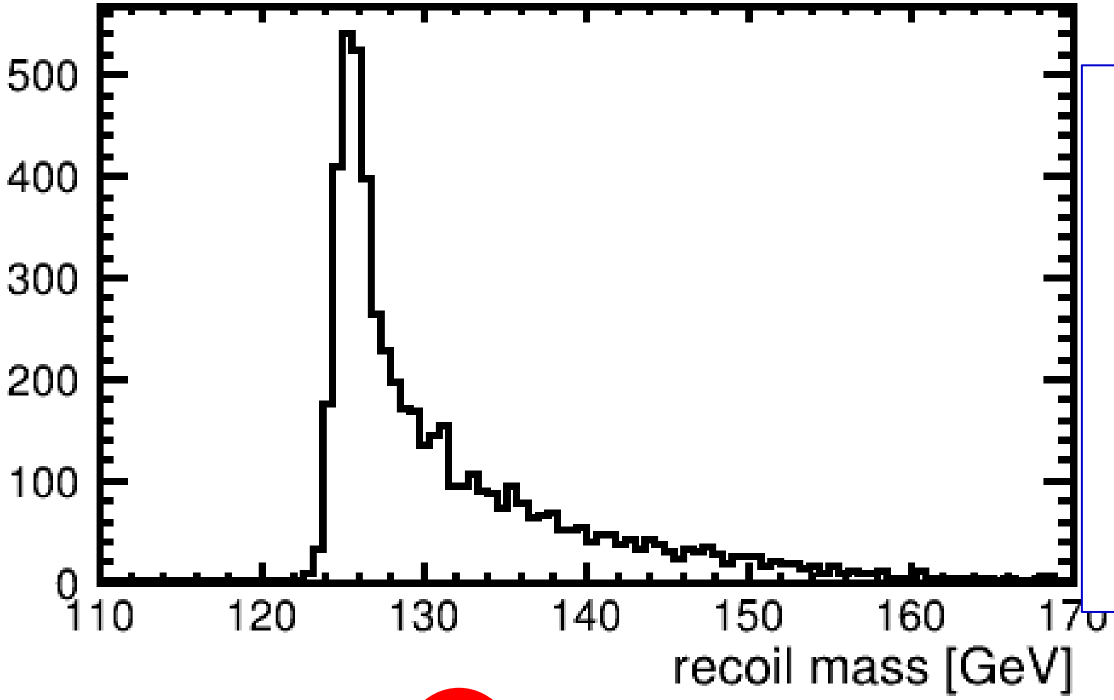


$$e^+e^- \rightarrow ZH \rightarrow (\mu^+\mu^-)(jj)$$

di-muon mass created with
one mini-DST file

```
//----------  the actual event processing

LCIterator<ReconstructedParticle> jets( evt, "Jets" ) ;
LCIterator<ReconstructedParticle> muons( evt, "Muons" ) ;

if( jets.size() != 2)
  continue;

if( muons.size() != 2)
  continue;

auto mu1 = muons.next();
auto mu2 = muons.next();
hmuonmass->Fill( inv_mass( mu1, mu2) ) ;
```

Example: Recoil mass at ILC250

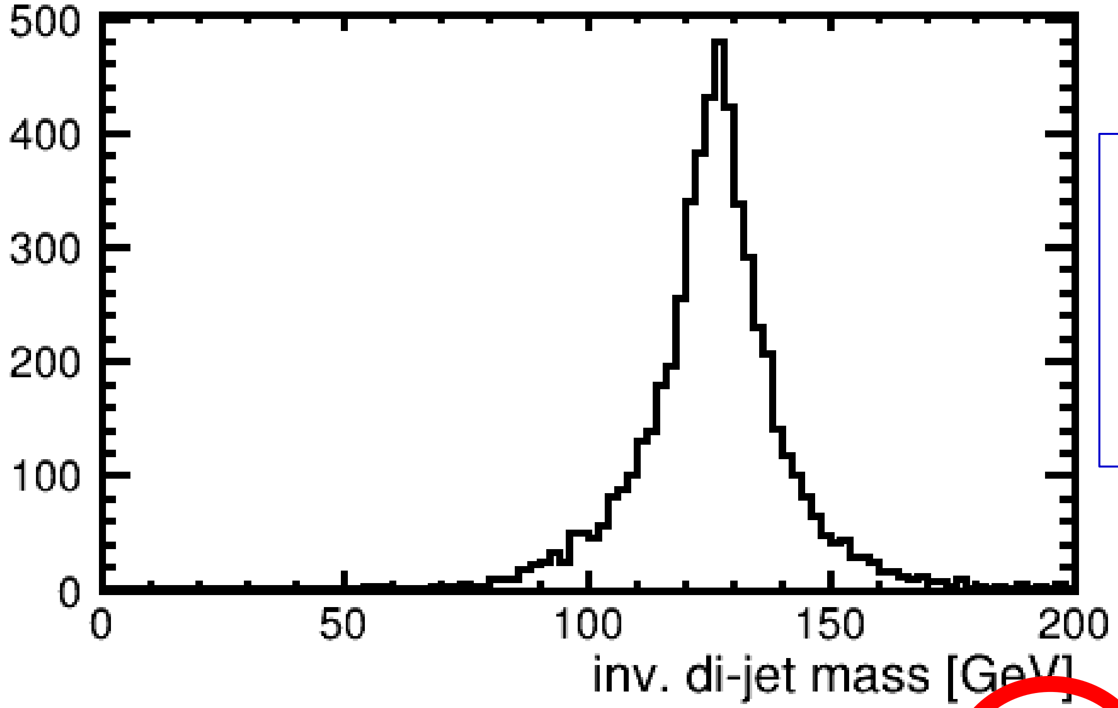



```
// the recoil mass
const auto& vm1 = v4(mu1) ;
const auto& vm2 = v4(mu2) ;
TLorentzVector ecms(0.,0.,0.,250.) ;
TLorentzVector recoil = ecms - ( vm1 + vm2 ) ;
hrecoilm->Fill( recoil.M() ) ;
```

$$e^+e^- \rightarrow ZH \rightarrow (\mu^+\mu^-)(jj)$$

recoil mass created with
one mini-DST file

Example: Recoil mass at ILC250




```
auto j1 = jets.next();
auto j2 = jets.next();
hjetmass->Fill( inv_mass( j1, j2) ) ;
```

$$e^+e^- \rightarrow ZH \rightarrow (\mu^+\mu^-)(jj)$$

di-jet mass created with
one mini-DST file

Simple Macro

- The plots in previous pages were made by such a simple macro.
  - Only require ROOT, LCIO libraries, and input mini-DST file
  - Usage of mini-DST format enables a seamless transition among DELPHES, SGV, and full simulation.

```
//----------  the actual event processing

LCIterator<ReconstructedParticle> jets( evt, "Jets" ) ;
LCIterator<ReconstructedParticle> muons( evt, "Muons" ) ;

if( jets.size() != 2)
  continue;

if( muons.size() != 2)
  continue;

auto mu1 = muons.next();
auto mu2 = muons.next();
hmuonmass->Fill( inv_mass( mu1, mu2) ) ;

auto j1 = jets.next();
auto j2 = jets.next();
hjetmass->Fill( inv_mass( j1, j2) ) ;


// the recoil mass
const auto& vm1 = v4(mu1) ;
const auto& vm2 = v4(mu2) ;
TLorentzVector ecms(0.,0.,0.,250.) ;
TLorentzVector recoil = ecms - ( vm1 + vm2 ) ;
hrecoilm->Fill( recoil.M() ) ;
```

Limitation of mini-DST

- The mini-DST only have “minimum” set of collections (reconstructed particle, MC truth information) plus high-level object with typical parametrization of processors.

want to see different collections (Track, Cluster, ...) ⇨ switch to DST or special samples

e.g.: check muon track information

want to optimize analysis ⇨ use processors with own parametrization

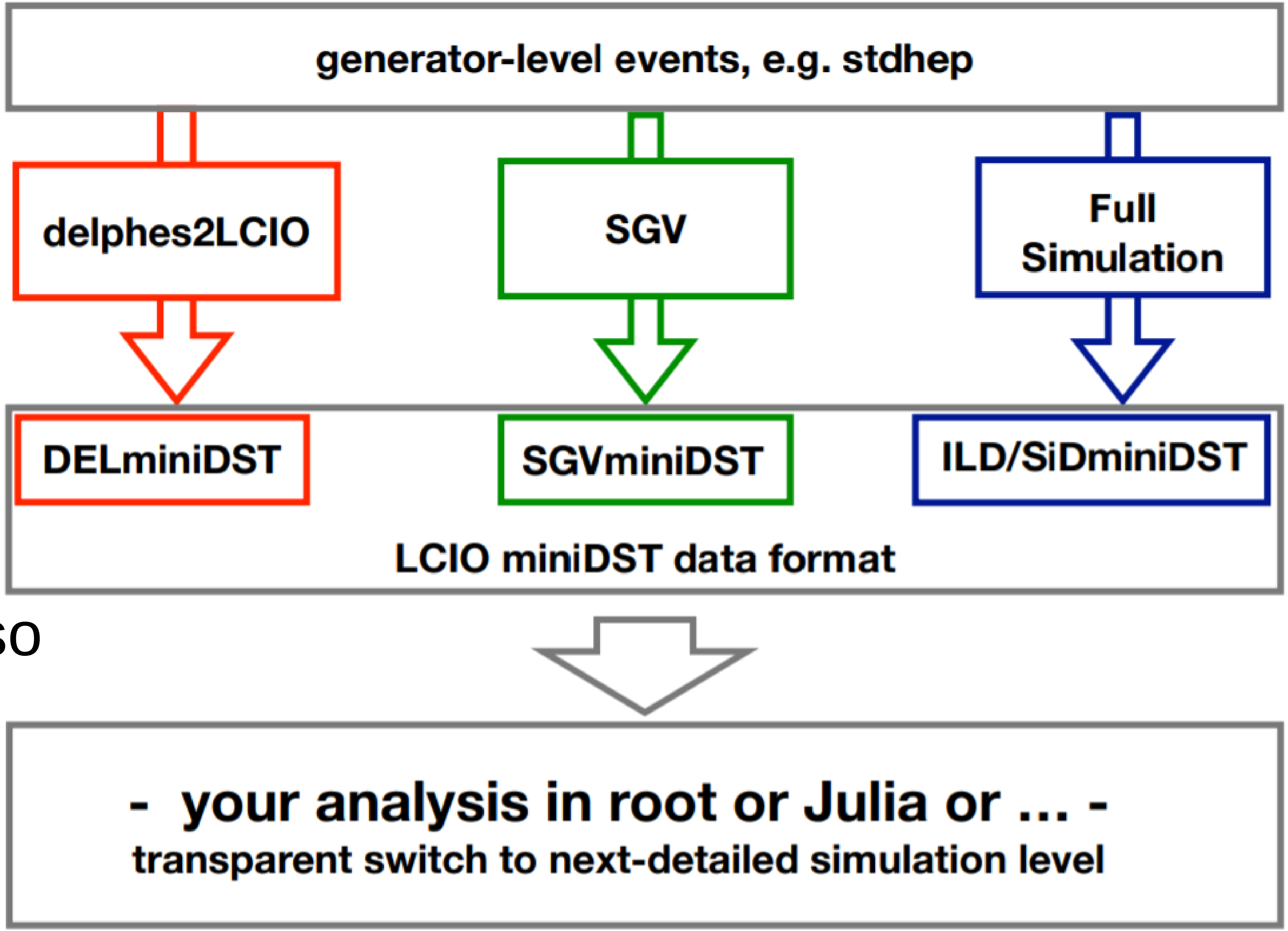
e.g.: want to use different jet clustering

Other application

mini-DST is a data format

not only full simulation but also can use for fast simulation

allow user to use the same analysis code to compare fast/full simulation



Summary

- The mini-DST data format has been developed to reduce the complexity for beginners and to give easy starting point of the analysis.
- Contains output of high-level objects, particle flow objects, MC truth with relations to reconstructed objects; good enough for most analysis.
- Enables seamless transfer of analysis code among DELPHES, SGV and full simulation.
- Only need ROOT and LCIO libraries to use mini-DST.
- More developments and documentation are work in progress.

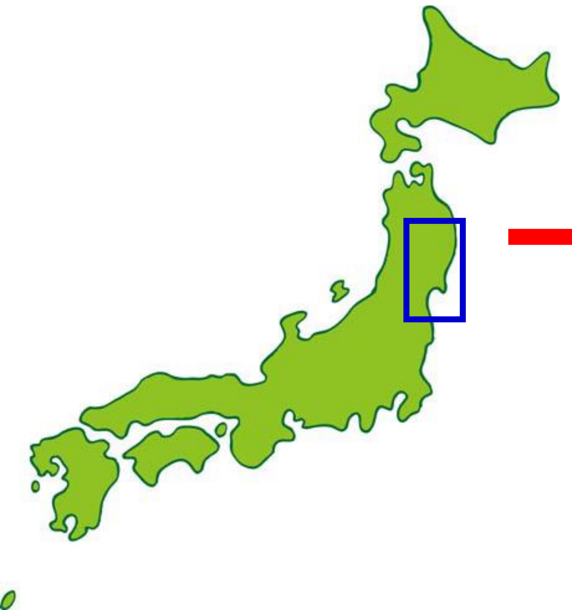
Links

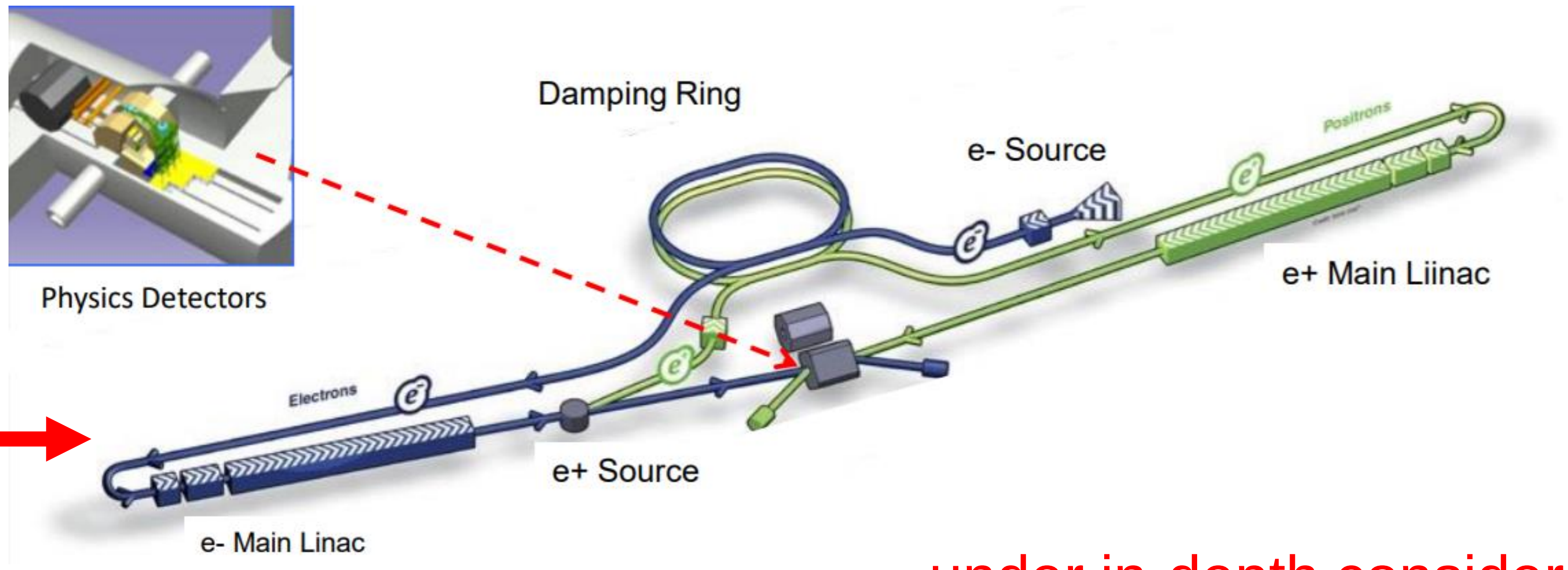
- My personal GitHub LINK
- Under ILDAnaSoft GitHub LINK
- Snowmass tutorial of ILC LINK

BACKUP

The International Linear Collider (ILC)

- e^+e^- collider, E_{CM} = 250 GeV (upgradable to 500 GeV, 1 TeV)
- polarized beam (e^-: $\mp$80%, e^+: $\pm$30%)
- clean environment, known initial state

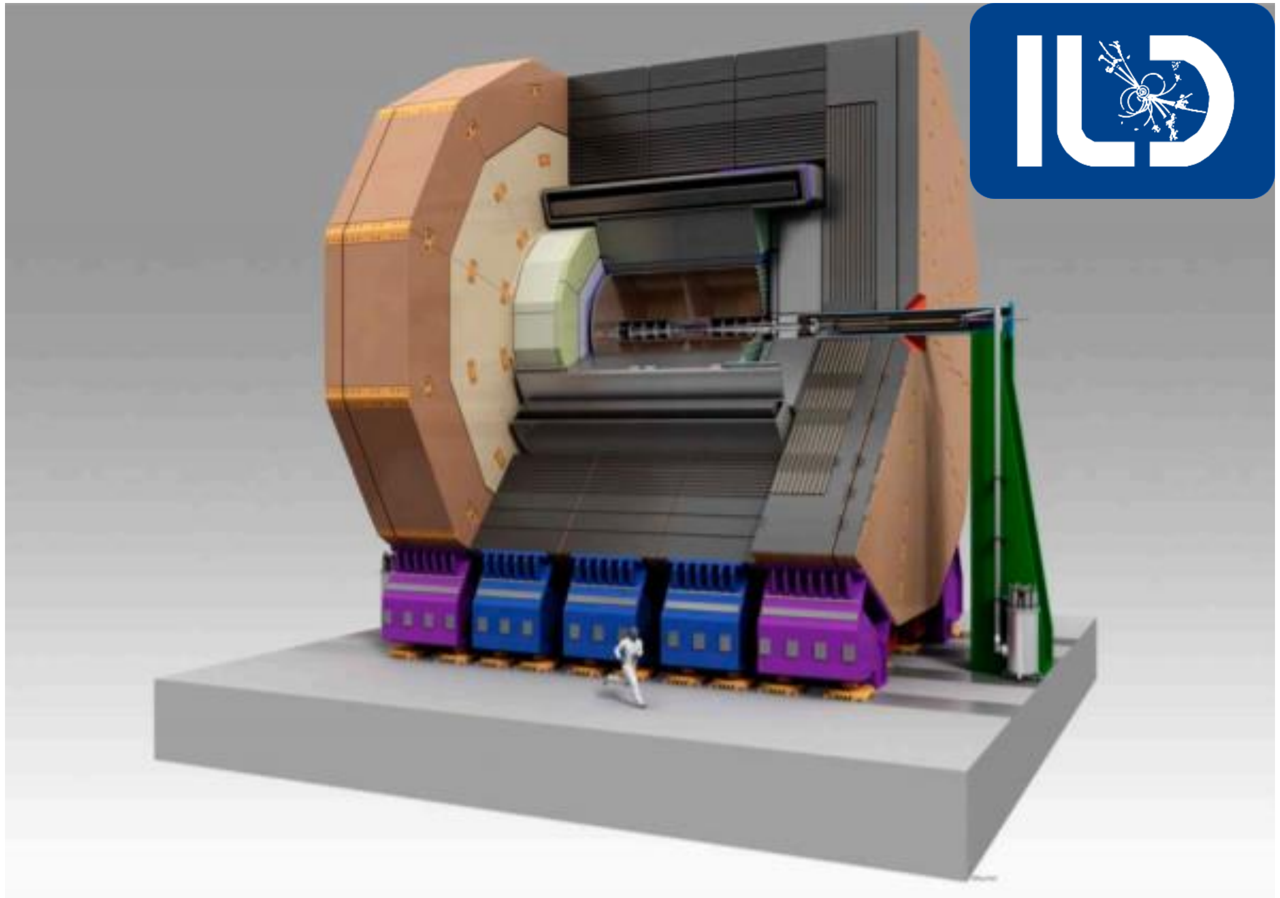




under in-depth consideration
by the Japanese government

Detector Concept at the ILC

ILD (**I**nternational **L**arge **D**etector)



Tracker: Vertex, TPC
Calorimeter: ECAL, HCAL
3.5T magnetic field
Yoke for muon, Forward system

Requirements:

- Impact parameter resolution

  $\sigma_{r\phi} < 5 \oplus \frac{10}{p\sin^{3/2}\theta}\ \mu\text{m}$

- **Momentum resolution**

  $\boldsymbol{\sigma_{1/p_T} < 2 \times 10^{-5}\ \mathrm{GeV}^{-1}}$

- Energy resolution

  $\sigma_E/E = 3 - 4\%$