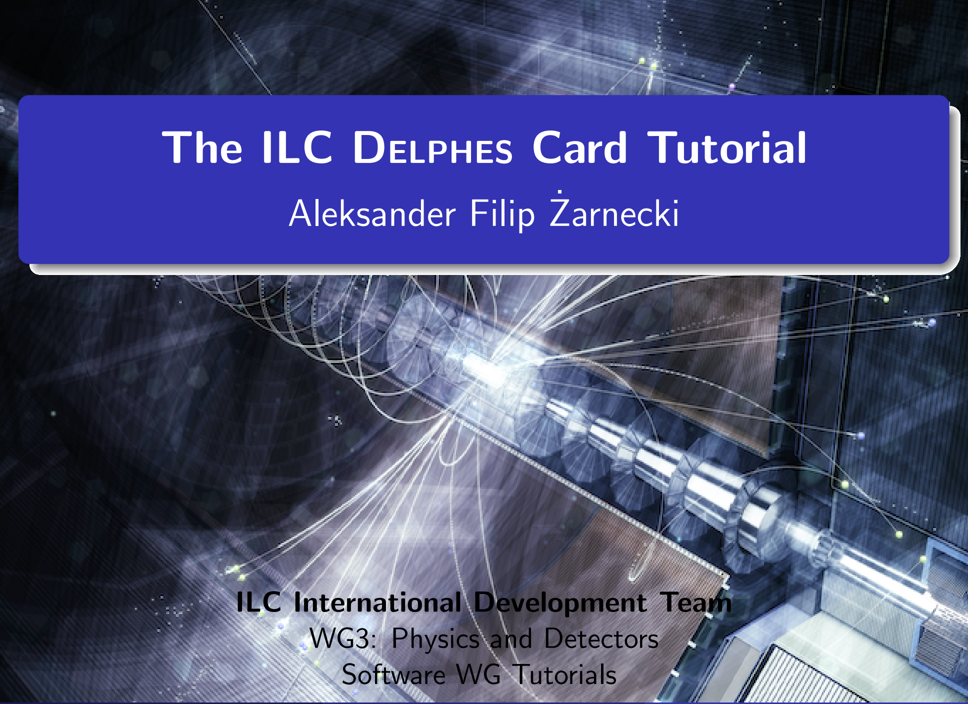


The ILC Delphes Card Tutorial

Aleksander Filip Żarnecki

ILC International Development Team

WG3: Physics and Detectors

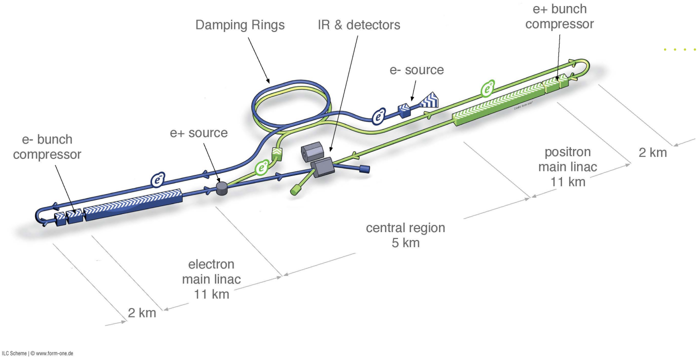
Software WG Tutorials

Outline

Cover image: Rey.Hori (copied from ILC Newsline)

ILC

International Linear Collider



Technical Design (TDR) completed in 2013 arXiv:1306.6328

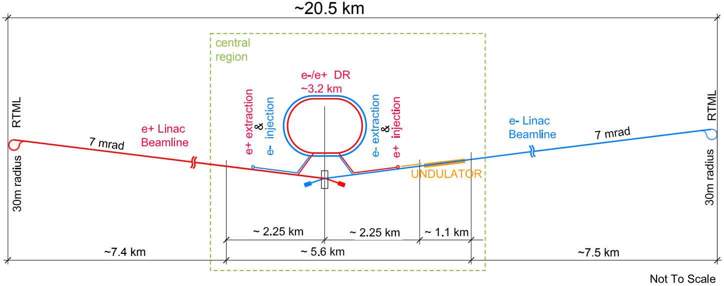
- superconducting accelerating cavities
- 250 – 500 GeV c.m.s. energy (baseline), 1 TeV upgrade possible
- footprint 31 km
- polarisation for both e^- and e^+ (80%/30%)

ILC-250

The discovery of a Higgs Boson with a mass of 125 GeV opened the possibility of reducing ILC cost by starting at a centre-of-mass energy of 250 GeV with the possibility of future upgrades to 500 GeV or even 1 TeV.

arXiv:1711.00568

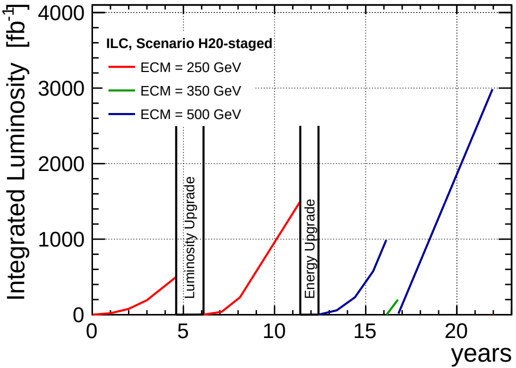
"Higgs-factory" layout 250 GeV optimal for Higgs production



arXiv:1903.01629

Running scenario

Baseline running scenario for staged ILC construction



arXiv:1903.01629

Total integrated luminosities same as in original H-20 proposal for ILC-500!

Polarisation

The unique feature of the ILC is the possibility of having both electron and positron beams polarised! This is crucial for many precision measurements as well as BSM searches. Four independent measurements instead of one:

- increase accuracy of precision measurements
- remove ambiguity in many BSM studies
- reduce sensitivity to systematic effects

Polarisation

The unique feature of the ILC is the possibility of having both electron and positron beams polarised! This is crucial for many precision measurements as well as BSM searches. Four independent measurements instead of one:

- increase accuracy of precision measurements
- remove ambiguity in many BSM studies
- reduce sensitivity to systematic effects

Integrated luminosity planned with different polarisation settings [fb^{-1}]

$\sqrt{s}$	$\mathrm{sgn}(P(e^-), P(e^+))$			
	(−,+)	(+,−)	(−,−)	(+,+)
250 GeV	900	900	100	100
350 GeV	135	45	10	10
500 GeV	1600	1600	400	400

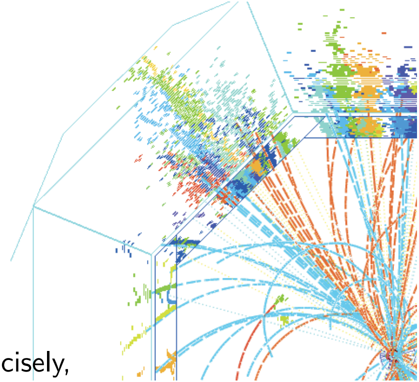
arXiv:1903.01629

Particle Flow concept

Jet energy resolution crucial for precision physics and background rejection

Typical jet composition:

- 60% charged particles
- 30% photons
- 10% neutral hadrons



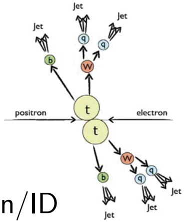
Jet energy poorly measured in calorimeters, large flactuations.

But we can measure:

- charged particle momenta very precisely,
- photon energy quite well,
- only neutral hadrons are a problem...

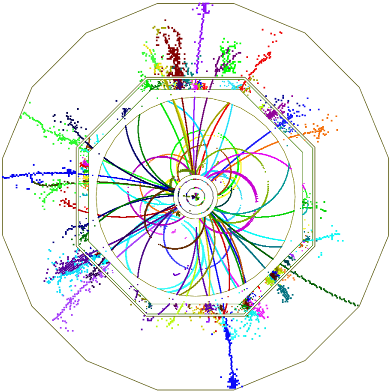
Detector Requirements

"Particle Flow" concept:
try to measure energy
particle by particle



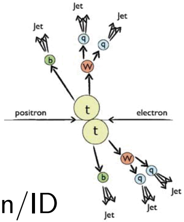
Single particle reconstruction/ID
$\Rightarrow$ high calorimeter granularity

Benchmark reaction
$e^+e^- \to t\bar{t} \to 6j$



Detector Requirements

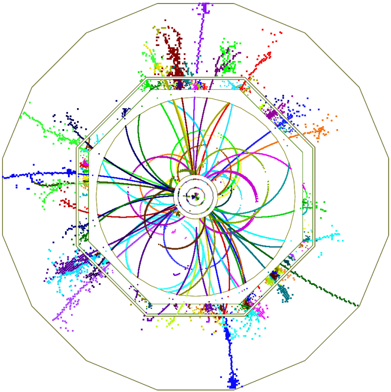
"Particle Flow" concept:
try to measure energy
particle by particle



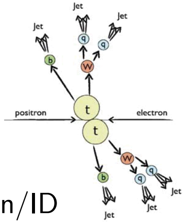
Single particle reconstruction/ID
⇒ high calorimeter granularity

Best energy estimate
for charged particles
⇒ precise momentum measurement

Benchmark reaction
$e^+e^- \to t\bar{t} \to 6j$



Detector Requirements



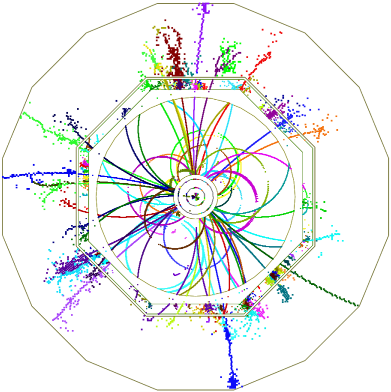
"Particle Flow" concept:
try to measure energy
particle by particle

Single particle reconstruction/ID
⇒ high calorimeter granularity

Best energy estimate
for charged particles
⇒ precise momentum measurement

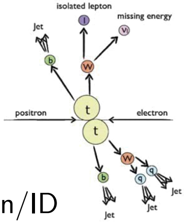
Very efficient flavour tagging
⇒ high precision vertex detector

Benchmark reaction
$e^+e^- \rightarrow t\bar{t} \rightarrow 6j$



Detector Requirements

"Particle Flow" concept:
try to measure energy
particle by particle



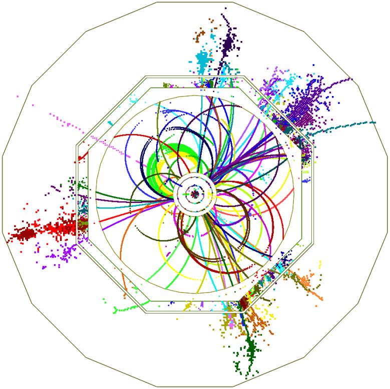
Single particle reconstruction/ID
⇒ high calorimeter granularity

Best energy estimate
for charged particles
⇒ precise momentum measurement

Very efficient flavour tagging
⇒ high precision vertex detector

Missing energy measurement
⇒ hermecity

Benchmark reaction
$e^+e^- \rightarrow t\bar{t} \rightarrow 4j + l + \nu$

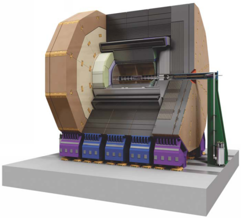


Detector Requirements

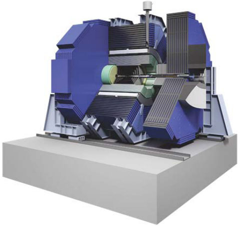
- Track momentum resolution: $\sigma_{1/p} < 5 \cdot 10^{-5}\ \text{GeV}^{-1}$
- Impact parameter resolution: $\sigma_d < 5\mu m \oplus 10\mu m \frac{1\ \text{GeV}}{p\ \sin^{3/2}\Theta}$
- Jet energy resolution: $\sigma_E/E = 3 - 4\%$ (for highest jet energies)
- Hermecity: $\Theta_{min} = 5$ mrad

Two detailed ILC detector concepts: arXiv:1306.6329

ILD



SiD



Different in design, but giving very similar performance

Simulation

Detector concepts for ILC were based on very detailed studies involving full simulation of detector response based on Geant 4.

Generated event samples, as will also be the case for the actual data collected at the ILC, need to be reconstructed, step by step, to extract full information about the underlying physics event.

Simulation

Detector concepts for ILC were based on very detailed studies involving full simulation of detector response based on Geant 4.

Generated event samples, as will also be the case for the actual data collected at the ILC, need to be reconstructed, step by step, to extract full information about the underlying physics event.

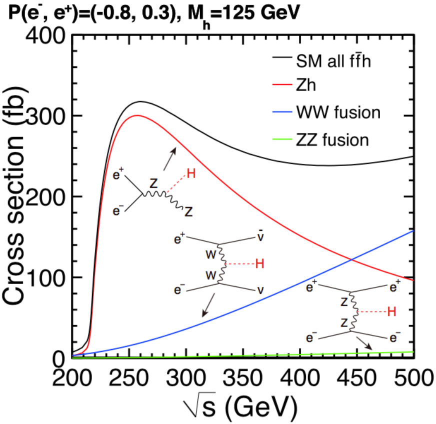
We need to use detailed event simulation and reconstruction to:

- evaluate detector performance figures,
- verify and optimise detector design,
- study influence of different systematic effects,
- establish expected precision of future measurements,
- and sensitivity to possible BSM scenarios.

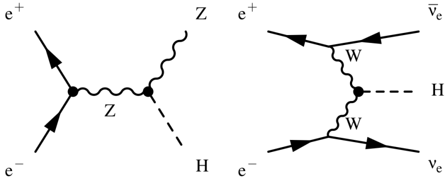
Unfortunately, "full" event simulation and reconstruction is very time consuming. There are cases, where we can use simplified methods...

First ILC running stage will clearly be focused on Higgs measurements

Higgs production



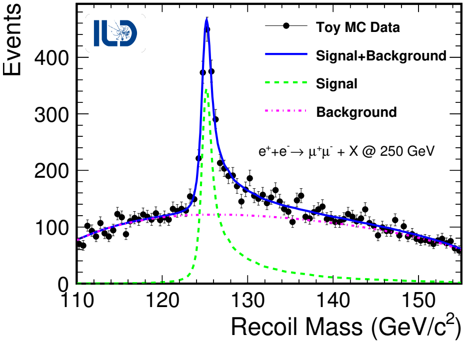
At 250 GeV dominated by Higgs-strahlung (ZH production)



but we still profit from combining two production channels
⇒ model independent analysis

Higgs production

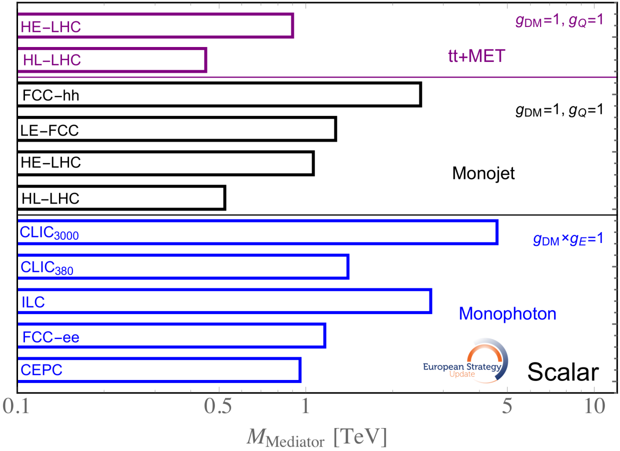
In the ZH production channel (dominating below 450 GeV) we can use "Z-tagging" for unbiased selection of Higgs production events



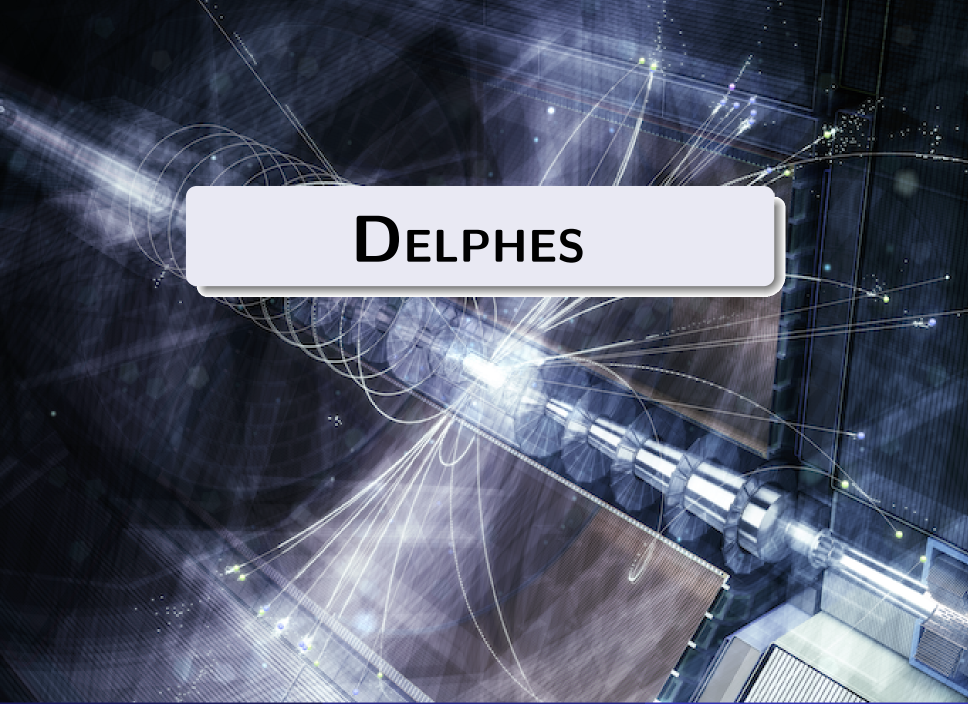
We avoid any dependence on the Higgs decay channel!

Indirect BSM searches

Comparison of extracted dark sector mediator mass limits



ILC mass reach comparable with that of FCC-hh !!!



DELPHES

Overview

Delphes implements the simplest possible approach to fast event simulation of collision events.

Simulation steps (1):

- Use four-momenta of all stable particles as input
  - ⇒ propagate particles in the magnetic field

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

Simulation steps (1):

- Use four-momenta of all stable particles as input
  - ⇒ propagate particles in the magnetic field
- Tracking detector modeling
  - ⇒ reconstruction efficiency (acceptance) and momentum resolution
  - ⇒ separately for electrons, muons and charged hadrons (based on true ID)

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

DELPHES
fast simulation

Simulation steps (1):

- Use four-momenta of all stable particles as input
  - ⇒ propagate particles in the magnetic field
- Tracking detector modeling
  - ⇒ reconstruction efficiency (acceptance) and momentum resolution
  - ⇒ separately for electrons, muons and charged hadrons (based on true ID)
- Calorimeter modeling
  - ⇒ geometrical acceptance and tower structure
  - ⇒ tower energy threshold and resolution for ECAL and HCAL
  - ⇒ energy sharing between ECAL and HCAL fixed for given particle ID!

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

Simulation steps (1):

- Use four-momenta of all stable particles as input
  - ⇒ propagate particles in the magnetic field
- Tracking detector modeling
  - ⇒ reconstruction efficiency (acceptance) and momentum resolution
  - ⇒ separately for electrons, muons and charged hadrons (based on true ID)
- Calorimeter modeling
  - ⇒ geometrical acceptance and tower structure
  - ⇒ tower energy threshold and resolution for ECAL and HCAL
  - ⇒ energy sharing between ECAL and HCAL fixed for given particle ID!
- Particle Flow (EFlow) reconstruction
  - ⇒ collects all reconstructed tracks and calorimeter towers not matched to tracks (or with excess energy)

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

Simulation steps (2):

- Isolated objects
  - ⇒ select electrons, muons and photons based on their true ID (!)
  - ⇒ apply identification efficiency and isolation criteria

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

Simulation steps (2):

- Isolated objects
  - ⇒ select electrons, muons and photons based on their true ID (!)
  - ⇒ apply identification efficiency and isolation criteria
- Jet clustering
  - ⇒ use EFlow objects, after reconstructed isolated objects are removed
  - ⇒ different clustering configurations can be simultaneously used

Overview

DELPHES implements the simplest possible approach to fast event simulation of collision events.

DELPHES
fast simulation

Simulation steps (2):

- Isolated objects
  - ⇒ select electrons, muons and photons based on their true ID (!)
  - ⇒ apply identification efficiency and isolation criteria
- Jet clustering
  - ⇒ use EFlow objects, after reconstructed isolated objects are removed
  - ⇒ different clustering configurations can be simultaneously used
- Flavour tagging
  - ⇒ "true" jet flavour is defined by looking for the closest MC parton
  - ⇒ flavour tag bits are generated taking into account tagging efficiency and mistagging probabilities
  - ⇒ multiple flavour tag bits can be defined for each jet

Reliability

DELPHES approach is very simplified, but it can still give reliable results provided adequate detector description is provided

DELPHES uses detector geometry, thresholds, resolution formula and selection efficiencies, as defined in detector model cards.

Reliability

DELPHES approach is very simplified, but it can still give reliable results provided adequate detector description is provided

DELPHES fast simulation

DELPHES uses detector geometry, thresholds, resolution formula and selection efficiencies, as defined in detector model cards.

If detector geometry and full simulation results are properly implemented in these cards, we can expect good modeling of:

- global event properties,
- (true) lepton identification and reconstruction,
- jet distributions, including energy, invariant masses etc,
- cuts involving single flavour tagging criteria.

DELPHES allows also to take soft $\gamma\gamma \to$ *had* overlay background into account, but this is not yet implemented in the ILC detector model...

Reliability

Simulation procedure implemented in Delphes also includes "cheating", looking at the MC particle table for particle identification or jet flavour tagging.

Reliability

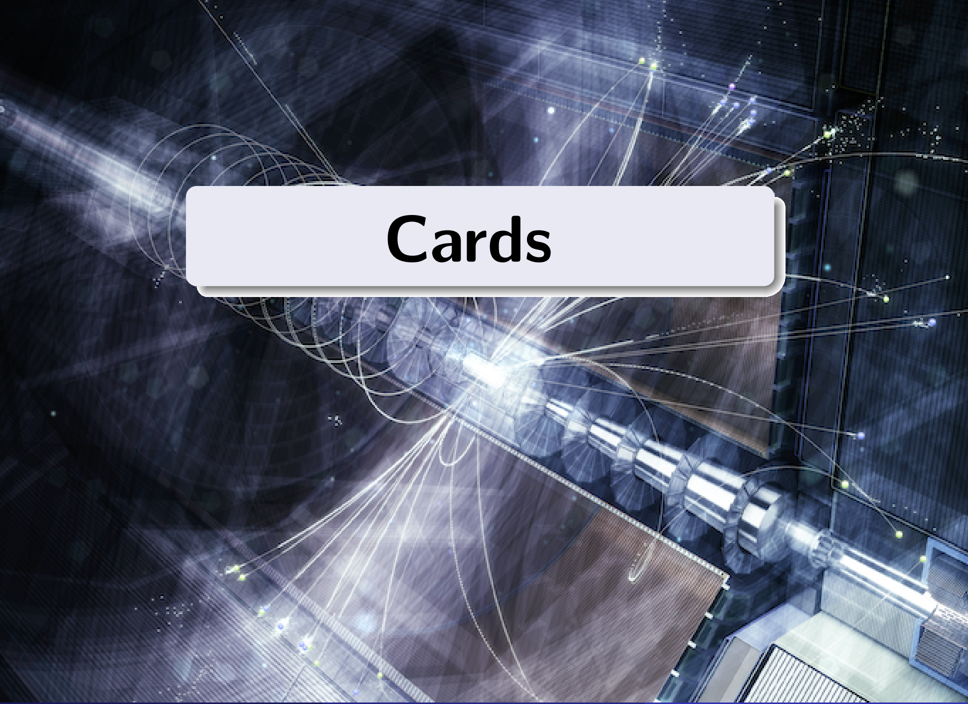
Simulation procedure implemented in DELPHES also includes "cheating", looking at the MC particle table for particle identification or jet flavour tagging.

DELPHES
fast simulation

You should never expect DELPHES to give correct results when you want to look at:

- "fake objects", e.g. fake electrons or muons in the detector
- effects of particle interactions in detector, e.g. photon radiation
- details of the detector response, e.g. ECAL – HCAL energy sharing
- impact of detector design, e.g. tower boundaries, gaps etc.
  as shower development and energy sharing between towers is not modeled, this will not work even if the geometry is precisely defined.
- correlations between different properties,
  e.g. tau tagging result and a number of particles in a jet
- correlation between different flavour taggs

Cards



Introduction

Detector models for ILC were implemented in Delphes in 2015 (ILC) and 2016 (SiD). However, the description was very simplified and only very general event properties could be reproduced.

Many features (like forward detectors, c-tag) were missing.

Introduction

Detector models for ILC were implemented in DELPHES in 2015 (ILC) and 2016 (SiD). However, the description was very simplified and only very general event properties could be reproduced.
Many features (like forward detectors, c-tag) were missing.

As a part of the Snowmass'2021 activities, it was suggested to prepare a new generic ILC detector model for DELPHES, which would better reproduce the expected detector performance.

Introduction

Detector models for ILC were implemented in DELPHES in 2015 (ILC) and 2016 (SiD). However, the description was very simplified and only very general event properties could be reproduced.
Many features (like forward detectors, c-tag) were missing.

As a part of the Snowmass'2021 activities, it was suggested to prepare a new generic ILC detector model for DELPHES, which would better reproduce the expected detector performance.

ILCgen detector model was implemented in June-July 2020.
Based on the ILD detector concept and simulation results, as presented in ILD IDR arXiv:2003.01116. It can be considered a generic ILC detector model, as expected performances of both ILD and SiD are very similar.
Since July 2020 ILCgen is included in the official DELPHES repository.

Input, contributions and support received from many people: Jenny List, Marcel Vos, Pawel Sopicki, Frank Gaede, Carl Mikael Berggren, Daniel Jeans, Ryo Yonamine, Tomohiko Tanabe, André Sailer, Remi Ete, Shin-ichi Kawada, Christopher Potter, Katja Krüger.

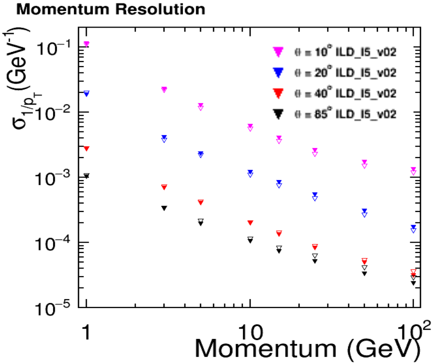
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

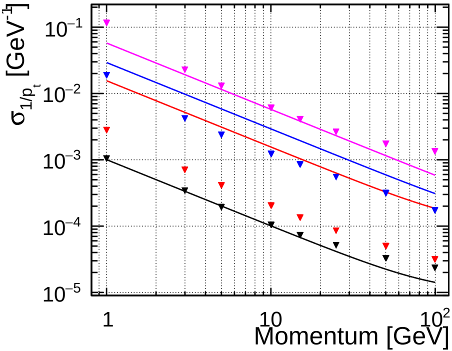
Same efficiency and resolution applied to all charged particles!

Dedicated parametrisation used instead of simple (p_T,η) bins.

ILD IDR Fig. 8.1 a (muons)



Old DELPHES model



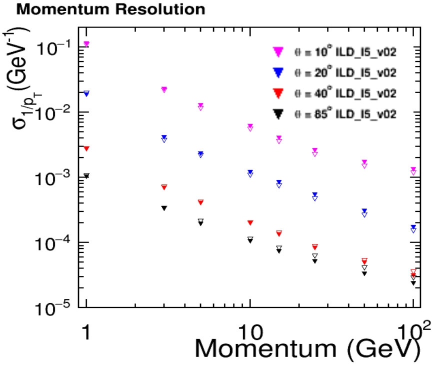
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

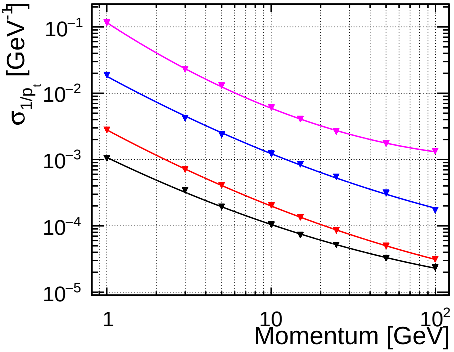
Same efficiency and resolution applied to all charged particles!

Dedicated parametrisation used instead of simple (p_T,η) bins.

ILD IDR Fig. 8.1 a (muons)



New DELPHES model



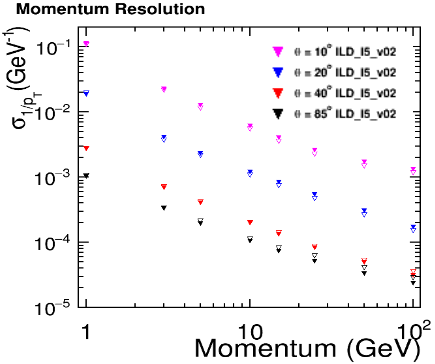
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

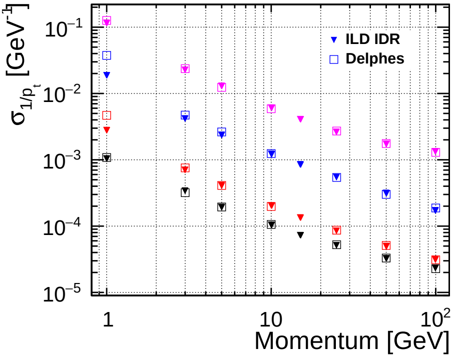
Same efficiency and resolution applied to all charged particles!

Dedicated parametrisation used instead of simple (p_T,η) bins.

ILD IDR Fig. 8.1 a (muons)



New Delphes simulation results



Calorimeters

Calorimeter coverage in $|\eta|$ assumed in ILCgen model

	EM	HAD
Central	up to 3.0	up to 2.8
Forward	3.0 – 4.0	2.8 – 3.8
BeamCal	4.0 – 5.8	

Tower structure defined in (η,ϕ)
$\Rightarrow$ tower size $(\Delta\eta,\Delta\phi)$ changing with rapidity range

Depth and longitudinal structure is not relevant in DELPHES

Central (ECAL, HCAL) and Forward (LumiCal, LHCal) calorimeters included in Particle Flow reconstruction
$\Rightarrow$ subsequent particle identification and jet clustering

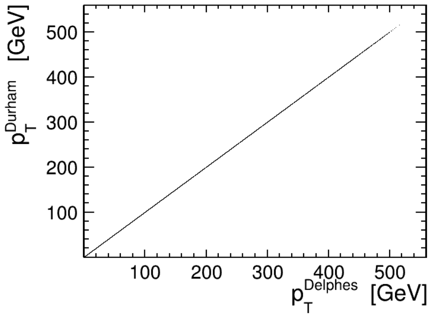
BeamCal response stored in separate collections for consistency with full simulation approach

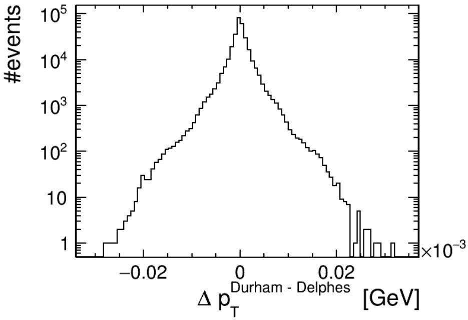
Jet clustering

Durham (ee_kt_algorithm in FastJet) not implemented in DELPHES (!)

Results reproduced with proper VLC configuration (R=2, $\beta = 1$, $\gamma = 0$)

Comparison of DELPHES jets (N=4) with Durham clustrisation in FastJet



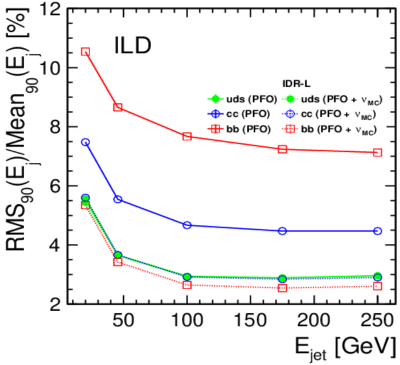


Jet energy resolution

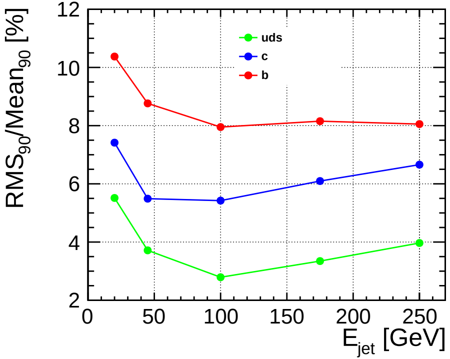
Surprisingly well reproduced with very simplified Particle Flow

Calorimeter granularity and energy response thresholds important!

ILD IDR Fig. 8.3 d ($Z \to q\bar{q}$)



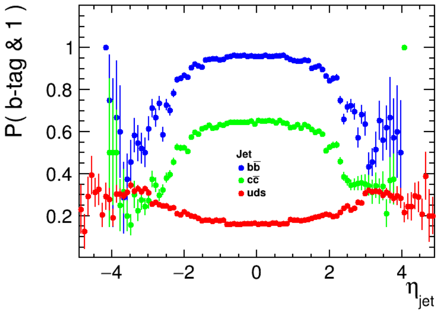
DELPHES simulation



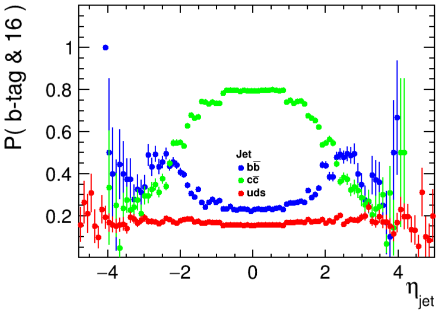
Jet flavour tagging

Both *b*- and *c*-tagging is implemented for all jet collections with 3 working points (loose, medium and tight selection).
They correspond (approximately) to 80%, 70% and 50% efficiency of *b*-tagging and 55%, 30% and 20% efficiency of *c*-tagging.

Loose *b*-tagging



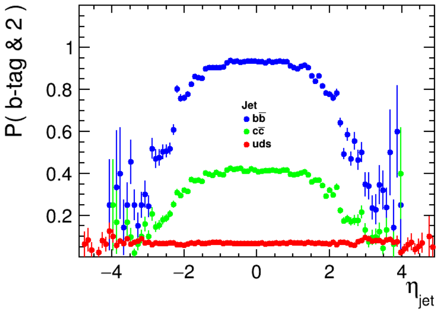
Loose *c*-tagging



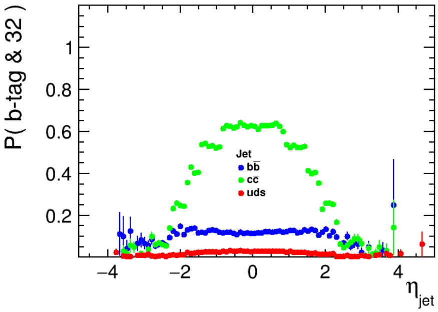
Jet flavour tagging

Both *b*- and *c*-tagging is implemented for all jet collections with 3 working points (loose, medium and tight selection).
They correspond (approximately) to 80%, 70% and 50% efficiency of *b*-tagging and 55%, 30% and 20% efficiency of *c*-tagging.

Medium *b*-tagging



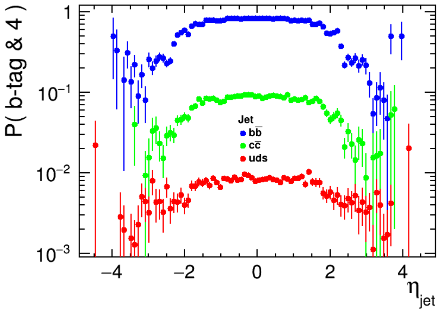
Medium *c*-tagging



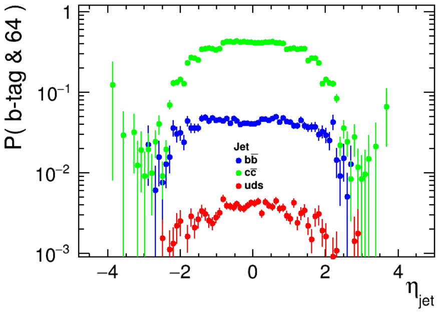
Jet flavour tagging

Both *b*- and *c*-tagging is implemented for all jet collections with 3 working points (loose, medium and tight selection).
They correspond (approximately) to 80%, 70% and 50% efficiency of *b*-tagging and 55%, 30% and 20% efficiency of *c*-tagging.

Tight *b*-tagging

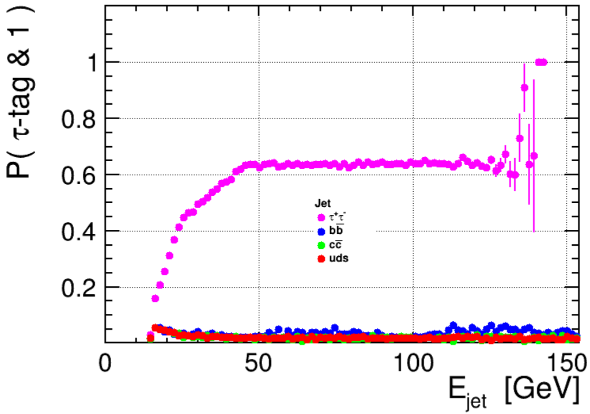


Tight *c*-tagging



Jet flavour tagging

ILCgen model includes also parametrisation of tau jet tagging results
only one working point implemented



BeamCal description

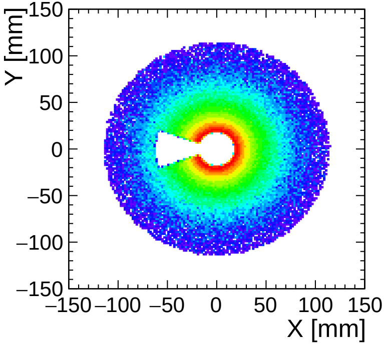
Beam crossing angle not taken into account in Delphes

Still, outgoing beam opening included in the BeamCal description.

⇒ best way to model efficiency drop for $\theta \leq 20$ mrad.

⇒ proper description of possible Rear-Forward correlations

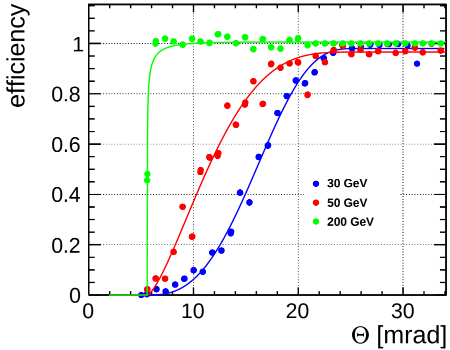
BeamCal tower hit positions for Bhabha event sample (log scale)



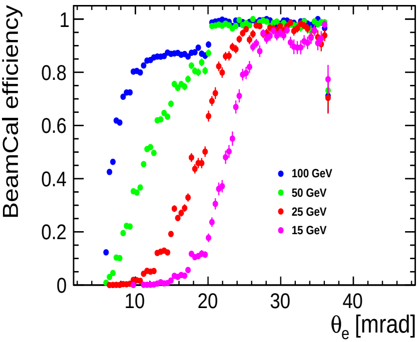
BeamCal description

Electron/Photon reconstruction in BeamCal significantly affected by background. Taken into account in the photon reconstruction efficiency, depending on both the energy and position (η) of electron/photon

Fit to ILD IDR results



DELPHES simulation



Corrected for keyhole opening

Output data structures

Results of the Delphes simulation are available at three levels:

- "raw" detector response:
  reconstructed tracks and deposits in calorimeter towers
- particle flow objects:
  reconstructed particle flow tracks, photons and neutral hadrons
- final reconstruction results:
  isolated electrons, muons and photons, and hadronic jets

We strongly recommend use of the final reconstruction results in analysis based on Delphes simulation!

Delphes simulation results can be stored in mini-DST format (LCIO), as well as in the native root format.

"Raw" detector response is not stored in the output file by default.
You can uncomment it in the model file, if you know what you are doing...

For more details on ILCgen detector model see github repository

Tips and tricks

When generating your sample

When using WHIZARD to generate your samples you should remember to include proper beam settings for ILC

```
beams = "e-", "e+" => circe2 => isr

$circe2_file = "250_SetA_ee024.circe"
$circe2_design = "ILC"
?circe2_polarized = false

beams_pol_density = @(-1), @(+1)
beams_pol_fraction = 80%, 30%
```

where the beams spectra file can be downloaded from WHIZARD repository

Instead of generating 4 events samples, for 4 polarisation considered, it is in most cases more effective to generate only two samples, with (−100%,+100%) and (+100%,−100%) polarisation settings, and mix the accordingly.

(−100%,−100%) and (+100%,+100%) cross sections vanish for majority of SM processes

How to install DELPHES

You need to have CERN root installed first (preferably from source).
root and its libraries need to be visible in the environment:

```
export ROOTSYS=/opt/root/default
export PATH=${PATH}:${ROOTSYS}/bin
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${ROOTSYS}/lib
```

DELPHES is then installed in just few simple steps
(run as root, if you want to install in system catalog)

```
mkdir -p /opt/Delphes
cd /opt/Delphes

git clone https://github.com/delphes/delphes.git

cd delphes
make
```

Refer to DELPHES wiki for more information

How to install Delphes2LCIO

Delphes2LCIO provides an option for creating an LCIO file from Delphes simulation output in the mini-DST format. See more details on github.

You need to have CERN root and DELPHES installed.

You need to have LCIO package installed, see git repository for details
Note that ROOT dictionary for LCIO that allows to read LCIO files in ROOT is not created by default, you need to add an option in cmake!

Delphes2LCIO is shipped with LCIO, you just need to compile it:

```
cd $LCIO/examples/cpp/delphes2lcio
mkdir build
cd build
cmake -D LCIO_DIR=$LCIO ..
make -j 4 install
```

and copy (or link) the executable file built to main DELPHES catalog

How to check your sample (and DELPHES)

DELPHES distribution includes many different detector models, but also few utility cards, which can sometimes be useful.

Before running DELPHES its components need to be visible:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/Delphes/delphes
```

```
ln -s /opt/Delphes/delphes .
ln -s delphes/external .
ln -s delphes/classes .
```

To check the generated sample one can use gen_card.tcl cards:

```
delphes/DelphesSTDHEP delphes/cards/gen_card.tcl [output_name.root]\
                                                  [input_name.stdhep]
```

You can then start root session to see inside the output file:

```
gSystem->Load("delphes/libDelphes.so")
TFile *fin = TFile::Open("ee2ZH_250GeV_example.root")
TTree *gen = (TTree *)gROOT->FindObject("Delphes")
gen->Scan("Particle.PID:Particle.Status:Particle.E")
```

How to run DELPHES

To process the generated event sample with ILCgen detector model:

```
delphes/DelphesSTDHEP delphes/cards/delphes_card_ILCgen.tcl \
                      [output_name.root] [input_name.stdhep]
```

to produce root format output file or, for Mini-DST format (LCIO).

```
delphes/DelphesSTDHEP2LCIO  delphes/cards/delphes_card_ILCgen.tcl \
                      [output_name]     [input_name.stdhep]
```

You are likely to get many error messages from FastJet:

```
fastjet::Error:  Requested 3 exclusive jets, but there were only 2 particles in the event
fastjet::Error:  Requested 4 exclusive jets, but there were only 2 particles in the event
fastjet::Error:  Requested 5 exclusive jets, but there were only 2 particles in the event
fastjet::Error:  Requested 6 exclusive jets, but there were only 2 particles in the event
```

but this is OK!

You can test your Mini-DST file with standard LCIO tools, e.g.

```
anajob ee2ZH_250GeV_example.slcio | less
dumpevent ee2ZH_250GeV_example.slcio  1 | less
```

Checking DELPHES output

The advantage of the output root format is that you can directly check the output file in root "by hand":

```
gSystem->Load("delphes/libDelphes.so")
TFile *fin = TFile::Open("ee2ZH_250GeV_example.root")
TTree *del = (TTree *) gROOT->FindObject("Delphes")
del->Scan("Jet_size:Electron_size:Muon_size:Photon_size:
    BCalPhoton_size")
```

```
************************************************************************
*    Row   *  Jet_size * Electron_ * Muon_size * Photon_si * BCalPhoto *
************************************************************************
*        0 *         2 *         0 *         2 *         0 *         0 *
*        1 *         2 *         0 *         2 *         0 *         0 *
*        2 *         2 *         0 *         1 *         0 *         0 *
*        3 *         2 *         0 *         2 *         1 *         0 *
*        4 *         2 *         0 *         2 *         0 *         0 *
*        5 *         1 *         0 *         2 *         0 *         0 *
*        6 *         3 *         0 *         2 *         0 *         0 *
*        7 *         2 *         0 *         2 *         0 *         0 *
*        8 *         3 *         0 *         2 *         0 *         0 *
*        9 *         2 *         0 *         2 *         0 *         0 *
*       10 *         2 *         0 *         2 *         0 *         0 *
...
```

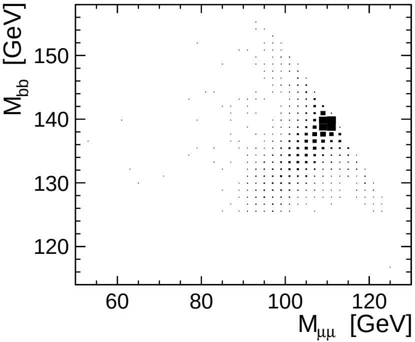
Checking Delphes output

Simple distributions you could plot by hand:

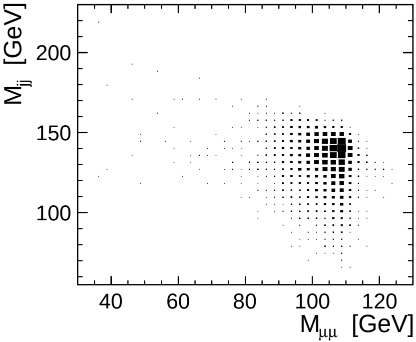
```
del->Draw("Jet_N2[0].P4().E()+Jet_N2[1].P4().E():Muon[0].P4().E()+
    Muon[1].P4().E()","","box")
```

Test sample of $e^+e^- \rightarrow ZH \rightarrow \mu^+\mu^- b\bar{b}$

Generator level



Delphes simulation



Dark matter particles

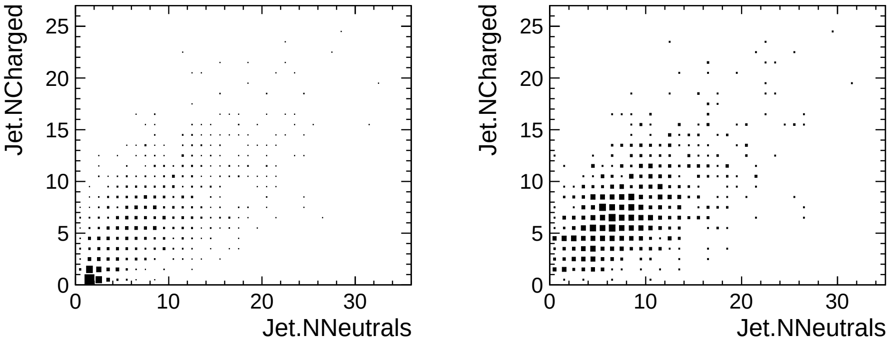
DELPHES assumes only SM particles in input event file!

If there are any "exotic" states, e.g. dark matter particles, which should remain invisible in the detector, you need to modify the main model file!

Otherwise DM is treated as neutral hadron ⇒ additional hadronic jet

Uncorrected

Corrected



Dark matter particles

DELPHES assumes only SM particles in input event file!

If there are any "exotic" states, e.g. dark matter particles, which should remain invisible in the detector, you need to modify the main model file!

Assuming my DM particle has ID=35:

```
module SimpleCalorimeter HCal {

    ...

    source ILCgen/ILCgen_HCAL_Binning.tcl
    source ILCgen/ILCgen_HCAL_EnergyFractions.tcl

    add EnergyFraction {35} {0.0}

    source ILCgen/ILCgen_HCAL_Resolution.tcl
}
```

Dark matter particles

DELPHES assumes only SM particles in input event file!

If there are any "exotic" states, e.g. dark matter particles, which should remain invisible in the detector, you need to modify the main model file!

Assuming my DM particle has ID=35:

```
module SimpleCalorimeter LHCalR {

    ...

    source ILCgen/ILCgen_LHCalR_Binning.tcl
    source ILCgen/ILCgen_HCAL_EnergyFractions.tcl

    add EnergyFraction {35} {0.0}

    source ILCgen/ILCgen_HCAL_Resolution.tcl
}
```

Dark matter particles

DELPHES assumes only SM particles in input event file!

If there are any "exotic" states, e.g. dark matter particles, which should remain invisible in the detector, you need to modify the main model file!

Assuming my DM particle has ID=35:

```
module PdgCodeFilter NeutrinoFilter {

    ...

  add PdgCode {-14}
  add PdgCode {-16}

  add PdgCode {35}
}
```

Use flavour tagging information

Both *b*- and *c*-tagging is implemented for all jet collections with 3 working points (loose, medium and tight selection).
However, DELPHES Jet class has only single variable: UInt_t BTag

b- and *c*-tagging results are thus stored as separate bits in a BTag word:

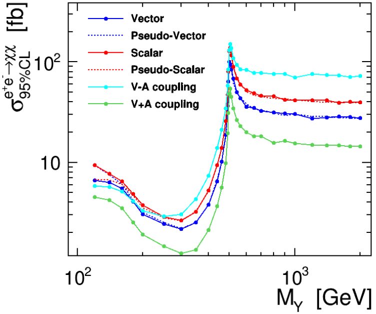
bit	expression (returning 0 or 1)	tag	level
0	jet.BTag&1	b-tag	loose
1	(jet.BTag&2)/2	b-tag	medium
2	(jet.BTag&4)/4	b-tag	tight
3	not used		
4	(jet.BTag&16)/16	c-tag	loose
5	(jet.BTag&32)/32	c-tag	medium
6	(jet.BTag&64)/64	c-tag	tight

Tips and tricks

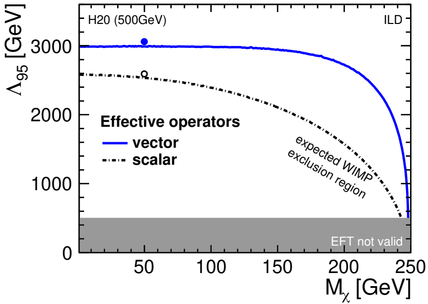
Example presented at IDT-WG3-Physics Open Meeting on June 17

Search for dark matter pair-production at ILC with light mediator exchange

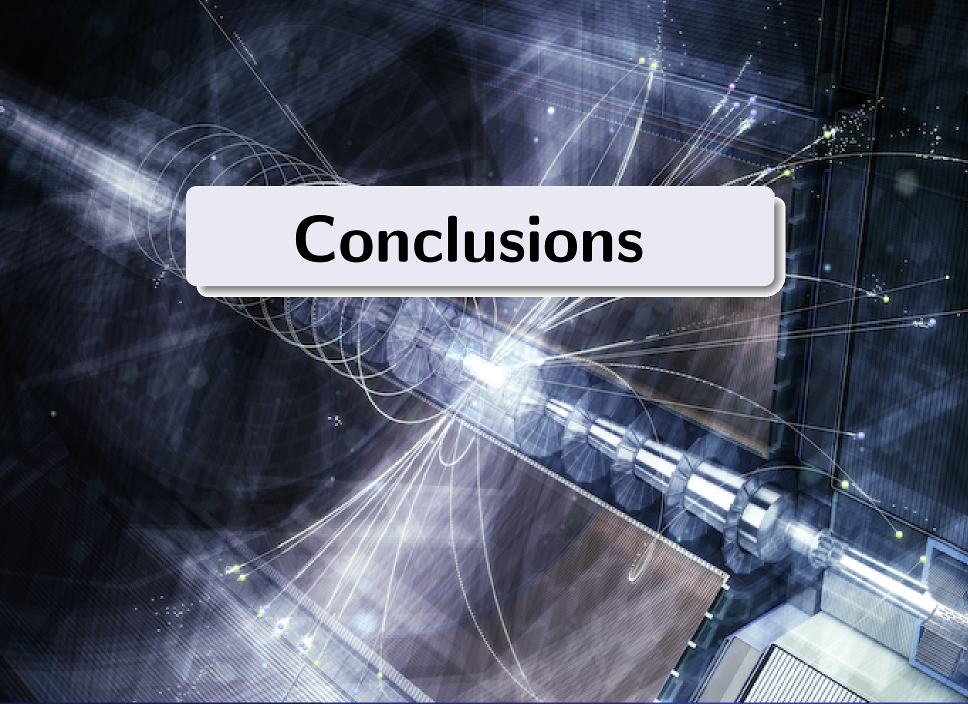
Delphes simulation used to scan model parameter space



Results verified by comparison with full simulation study



Over 500 signal scenarios considered
$\times$4 polarisation configurations $\times$100'000 events generated per scenario



Conclusions

Conclusions

Delphes is a simple and easy-to-use framework
suitable for getting immediately involved in ILC studies

ILCgen detector model for Delphes gives realistic description of the ILC experiment(s):

- It can be used to get preliminary estimates,
  before more involved full simulation studies are undertaken.
- Or to extrapolate full simulation results,
  e.g. when scanning BSM model parameter space.

When using mini-DST data format (LCIO) your analysis code will also be ready for use with full simulation data samples...

This will be covered in more detail in the second tutorial

Links

General

- ILC International Development Team https://linearcollider.org/
- ILC Newsline http://newsline.linearcollider.org/
- ILC IDT Working Group 3 (Physics and Detectors) https://linearcollider.org/team/wg3/
  also including many links to subgroups, indico sites etc.
- ILC Simulation Resources for Snowmass 2021 http://ilcsnowmass.org/
  including links to past tutorials and large sets of generated events samples
- SiD detector concept for ILC http://silicondetector.org
- ILD detector concept for ILC https://www.ilcild.org/
  https://confluence.desy.de/display/ILD/ILD

Software tools

- WHIZARD repository https://whizard.hepforge.org/
- ILC beam spectra files for WHIZARD https://whizard.hepforge.org/circe_files/ILC/
- DELPHES repository https://github.com/delphes/delphes
- DELPHES wiki https://cp3.irmp.ucl.ac.be/projects/delphes
- ILCgen model documentation https://github.com/iLCSoft/ILCDelphes
- LCIO package at github https://github.com/iLCSoft/LCIO
- Delphes2LCIO documentation
  https://github.com/iLCSoft/LCIO/tree/master/examples/cpp/delphes2lcio

References

Recent documents

- Proposal for the ILC Preparatory Laboratory (Pre-lab) arXiv:2106.00602
- ILC Study Questions for Snowmass 2021 arXiv:2007.03650
- International Large Detector: Interim Design Report arXiv:2003.01116
- Tests of the Standard Model at the International Linear Collider arXiv:1908.11299

European Strategy submissions

- The International Collider. A Global Project submission, arXiv:1903.01629
- The International Collider. An European perspective submission
- The ILD Detector at the ILC submission, arXiv:1912.04601

Other reports

- The role of positron polarization for the inital 250 GeV stage of the International Linear Collider arXiv:1801.02840
- The International Linear Collider Machine Staging Report 2017 arXiv:1711.00568
- Physics Case for the 250 GeV Stage of the International Linear Collider arXiv:1710.07621
- The Potential of the ILC for Discovering New Particles arXiv:1702.05333
- The International Linear Collider Technical Design Report Volume 3.II: Accelerator Baseline Design arXiv:1306.6328
- The International Linear Collider Technical Design Report Volume 4: Detectors arXiv:1306.6329