Experience with ILC detector model for Delphes

Aleksander Filip Żarnecki

Faculty of Physics, University of Warsaw

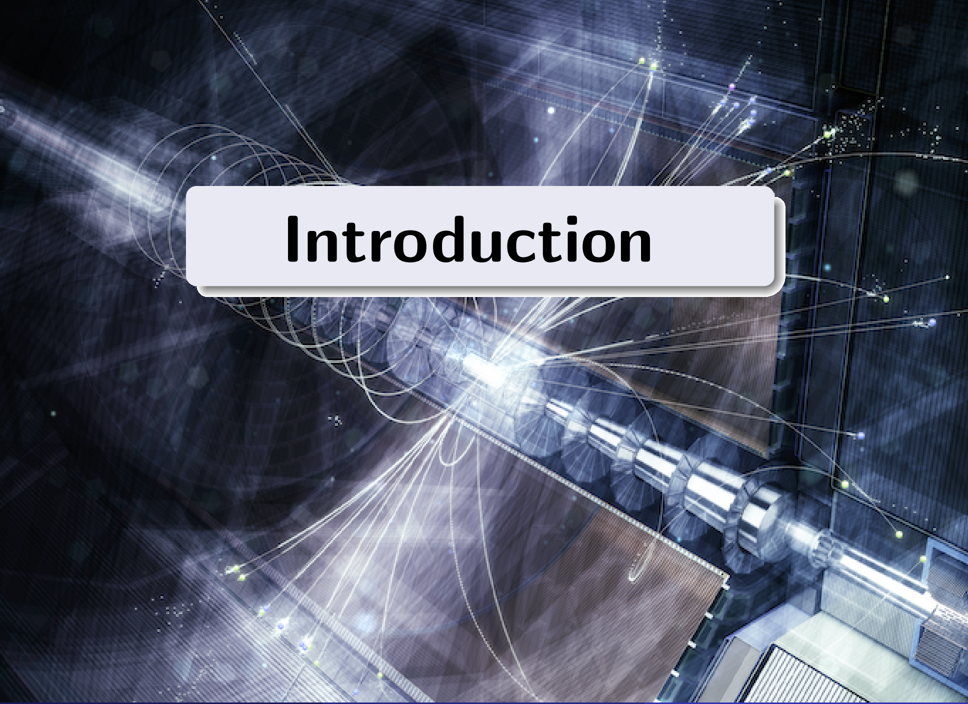
ECFA Higgs Factories

1st Topical Meeting on Simulation

Outline

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

Introduction



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.

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.

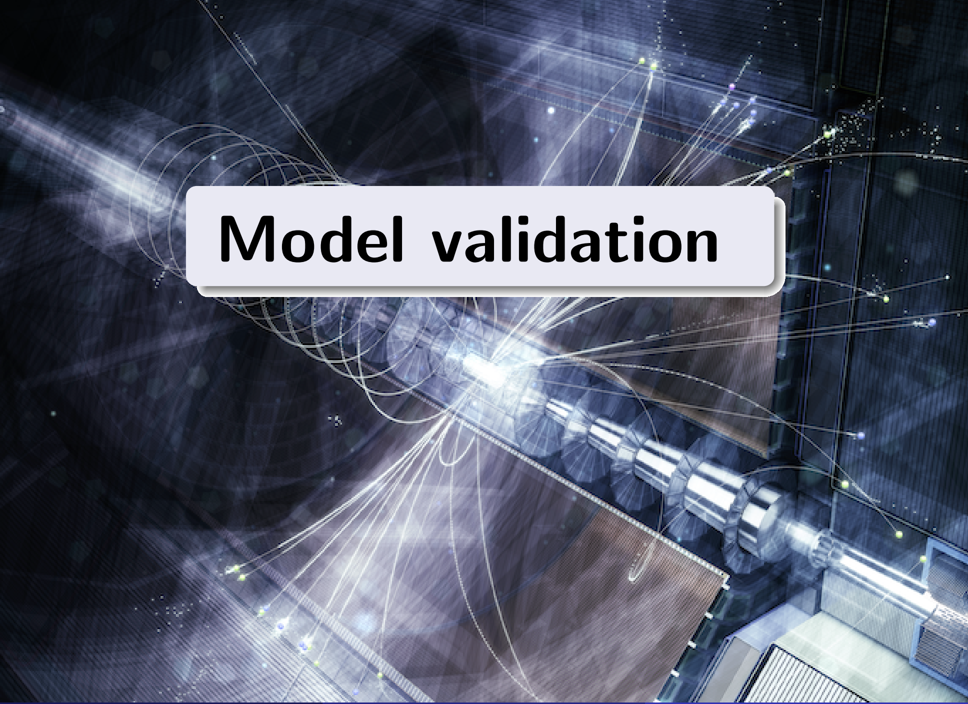
As a part of the Snowmass'2021 activities, it was suggested to prepare a new generic ILC detector model for DELPHES, which would better reproduce the expected detector performance.

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.

Model validation



Model validation

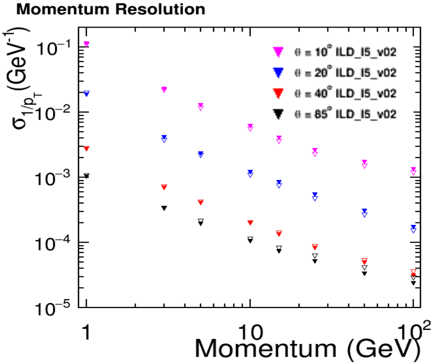
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

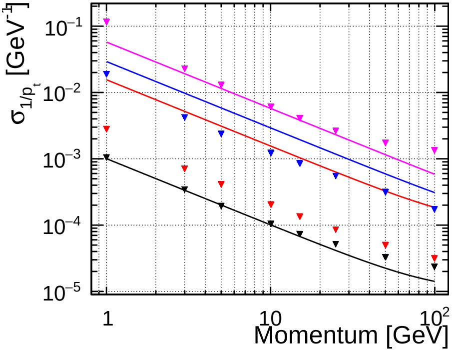
Same efficiency and resolution applied to all charged particles!

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

ILD IDR Fig. 8.1 a (muons)



Old DELPHES model



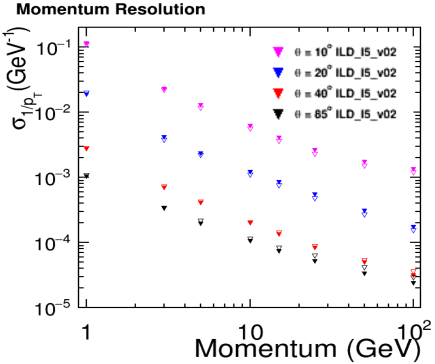
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

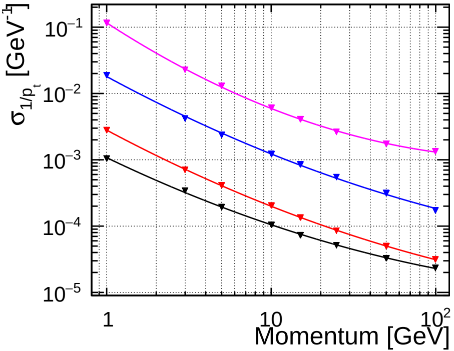
Same efficiency and resolution applied to all charged particles!

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

ILD IDR Fig. 8.1 a (muons)



New DELPHES model



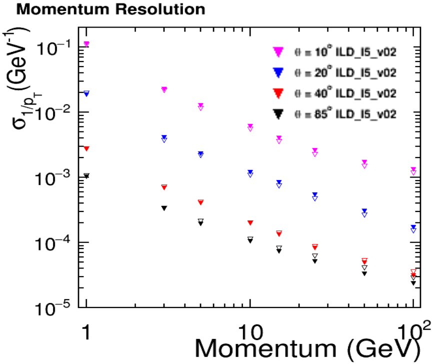
Tracking performance

Track momentum resolution taken from ILD IDR arXiv:2003.01116

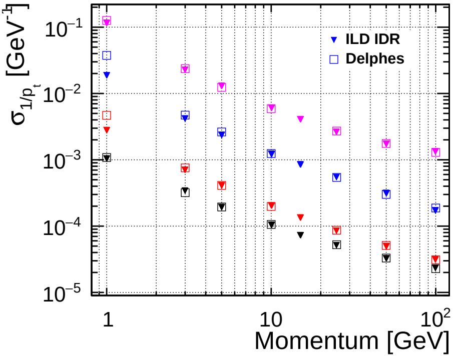
Same efficiency and resolution applied to all charged particles!

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

ILD IDR Fig. 8.1 a (muons)



New DELPHES simulation results



Model validation

Jet reconstruction

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 not relevant in DELPHES

Central (ECAL, HCAL) and Forward (LumiCal, LHCal) calorimeters **combined** in Particle Flow reconstruction

$\Rightarrow$ subsequent particle identification and jet clustering

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

Model validation

Jet reconstruction

In the ILC full simulation studies, Durham algorithm (ee_kt_algorithm in FastJet) is used as the default choice.

It has not been implemented in DELPHES (!)

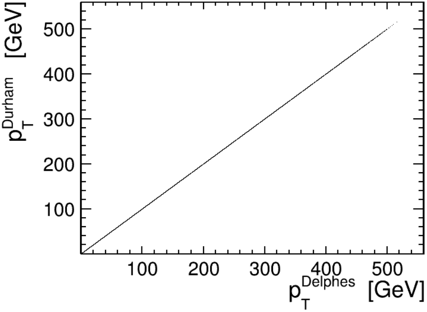
Jet reconstruction

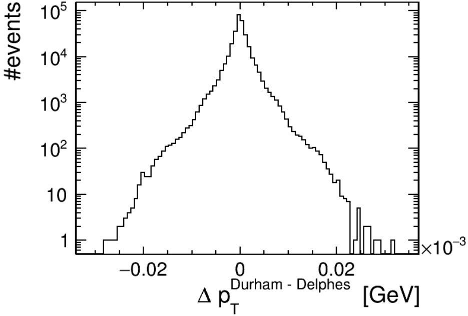
In the ILC full simulation studies, Durham algorithm (ee_kt_algorithm in FastJet) is used as the default choice.

It has not been 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



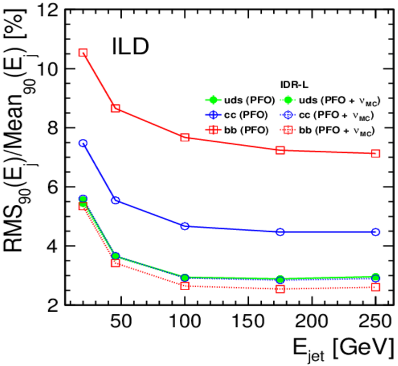


Jet energy resolution

Surprisingly well reproduced with DELPHES (very simplified) Particle Flow

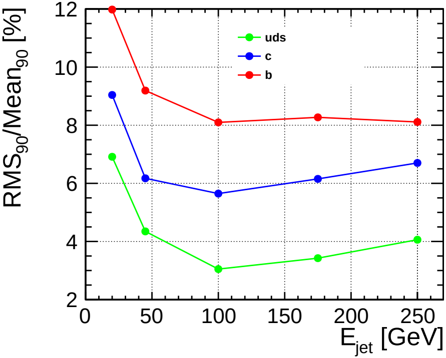
Calorimeter granularity and energy response thresholds important!

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



DELPHES simulation

tests



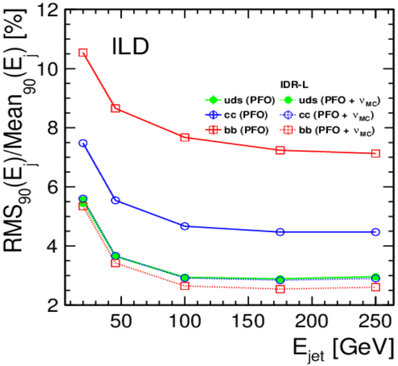
$E^{ECAL}_{min} = 0.5$ GeV, $E^{HCAL}_{min} = 1.0$ GeV

Jet energy resolution

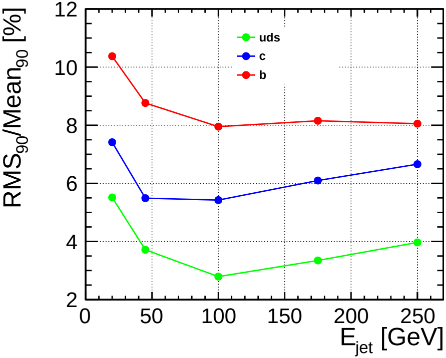
Surprisingly well reproduced with DELPHES (very simplified) Particle Flow

Calorimeter granularity and energy response thresholds important!

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



DELPHES simulation final implementation

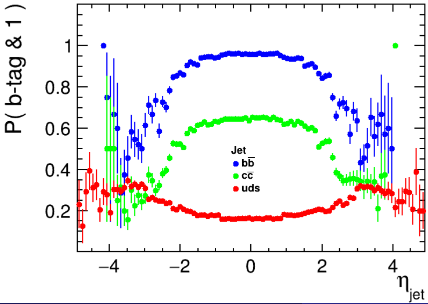


$E_{min}^{ECAL} = 0.1$ GeV, $E_{min}^{HCAL} = 0.25$ GeV

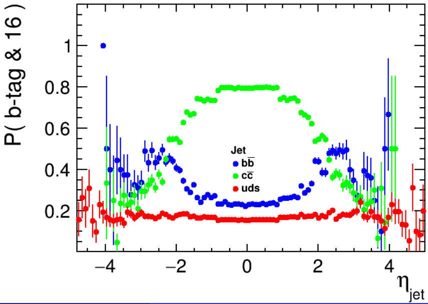
Jet flavour tagging

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.
"Signal" efficiency not fixed, but depends on the energy and rapidity !!!

Loose *b*-tagging



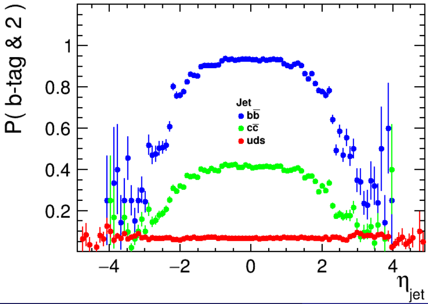
Loose *c*-tagging



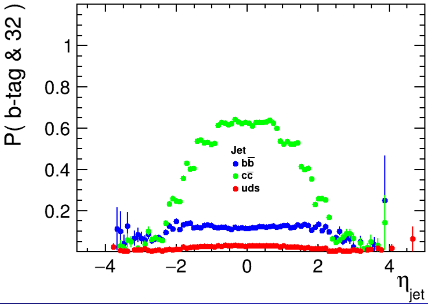
Jet flavour tagging

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.
"Signal" efficiency not fixed, but depends on the energy and rapidity !!!

Medium *b*-tagging



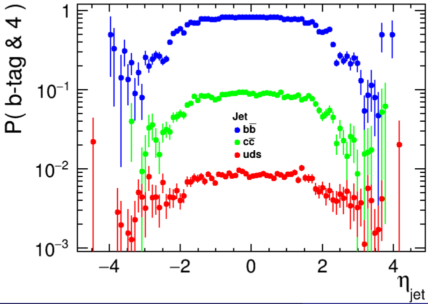
Medium *c*-tagging



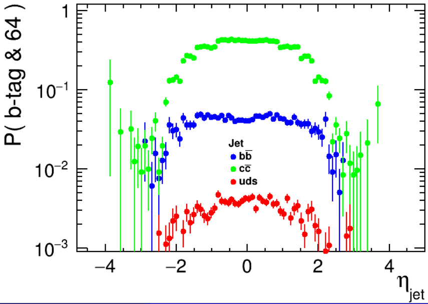
Jet flavour tagging

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.
"Signal" efficiency not fixed, but depends on the energy and rapidity !!!

Tight *b*-tagging



Tight *c*-tagging



BeamCal description

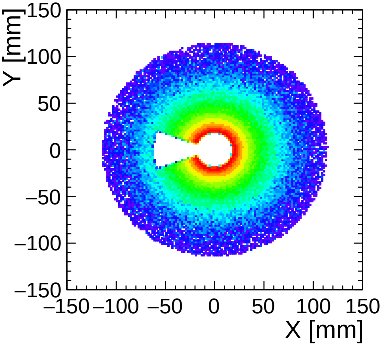
Beam crossing angle not taken into account in DELPHES

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

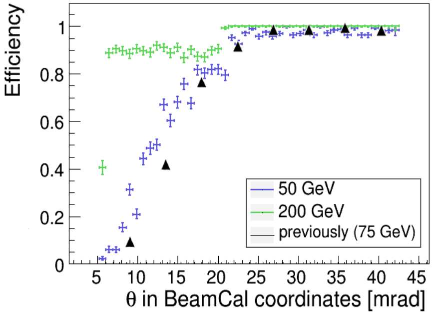
BeamCal tower hit positions for Bhabha event sample (log scale)



BeamCal description

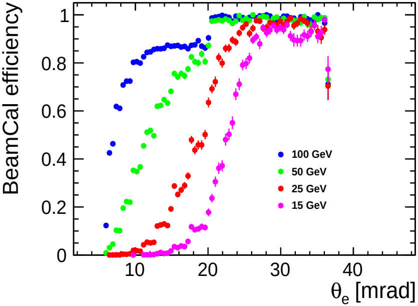
Electron/Photon reconstruction in BeamCal significantly affected by beam background. Taken into account in the photon reconstruction efficiency, depending on both the energy and position (η) of electron/photon

Example of full simulation results



arXiv:2001.03011

DELPHES simulation



fit to 500 GeV ILC full simulation results

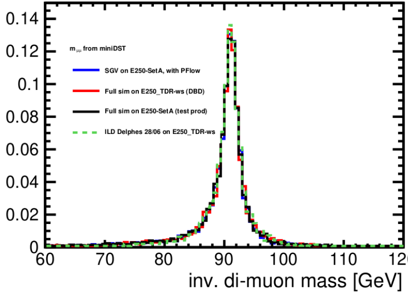
Example results

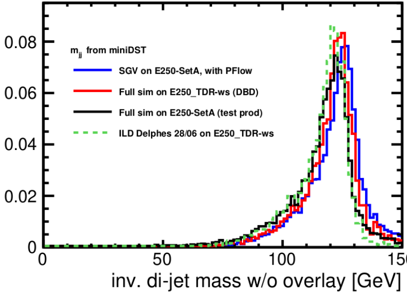
Higgs production at 250 GeV first checks, July 2020

Comparison of new Delphes model to SGV and full simulation results for

$$e^{+}e^{-} \rightarrow Z\,H \rightarrow \mu^{+}\mu^{-}q\,\bar{q}$$

Almost perfect agreement...





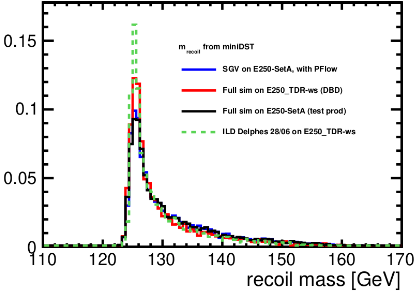
Plots prepared by Jenny List

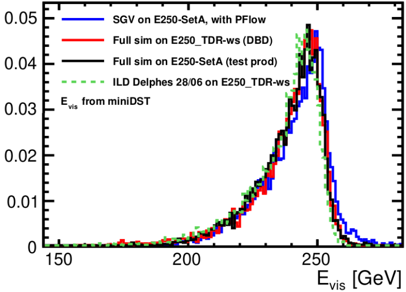
Higgs production at 250 GeV first checks, July 2020

Comparison of new Delphes model to SGV and full simulation results for

$$e^+e^- \rightarrow ZH \rightarrow \mu^+\mu^- q\,\bar{q}$$

Almost perfect agreement...





Plots prepared by Jenny List

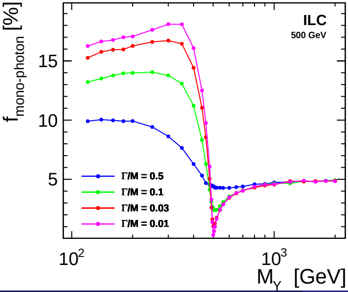
Mono-photon study

arXiv:2107.11194

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

$$e^+e^- \rightarrow Y \rightarrow \chi\,\chi\,\gamma$$

Delphes: Signal acceptance



Example results

Mono-photon study

arXiv:2107.11194

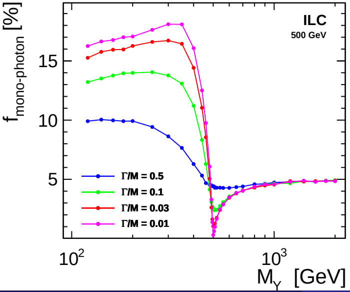
Search for dark matter pair-production at ILC via light mediator exchange

$$e^+e^- \rightarrow Y \rightarrow \chi\,\chi\,\gamma$$

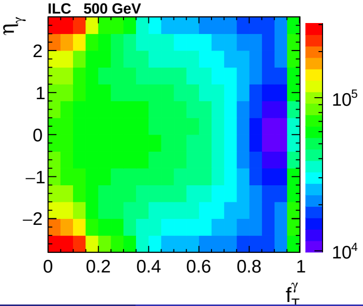
Huge background expected from SM processes:

$$e^+e^- \rightarrow e^+e^-\gamma \qquad\qquad e^+e^- \rightarrow \nu\,\bar{\nu}\,\gamma$$

Delphes: Signal acceptance



Background distribution



Example results

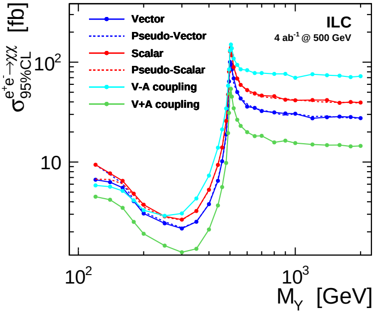
Mono-photon study

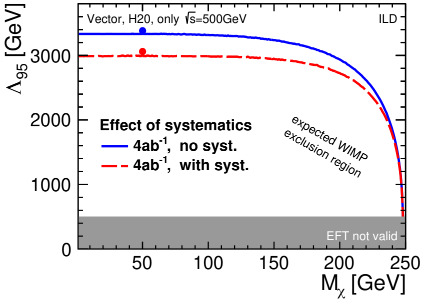
arXiv:2107.11194

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

DELPHES simulation crucial for scanning model parameter space

Results (•) verified by comparison with full simulation study (—)





Over 500 signal scenarios considered

×4 polarisation configurations ×100'000 events generated per scenario

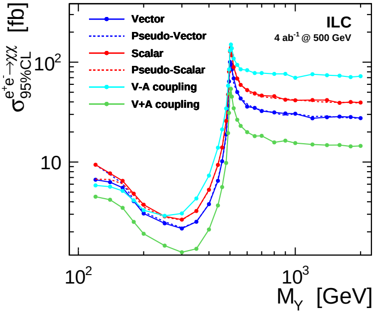
Mono-photon study

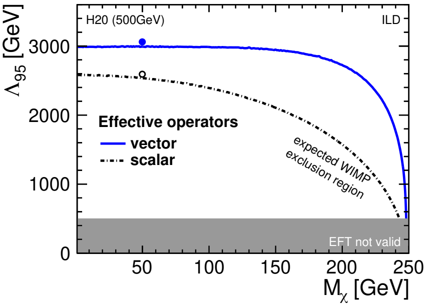
arXiv:2107.11194

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

DELPHES simulation crucial for scanning model parameter space

Results (•) verified by comparison with full simulation study (—)





Over 500 signal scenarios considered
×4 polarisation configurations ×100'000 events generated per scenario

Search for heavy neutrino production in preparation

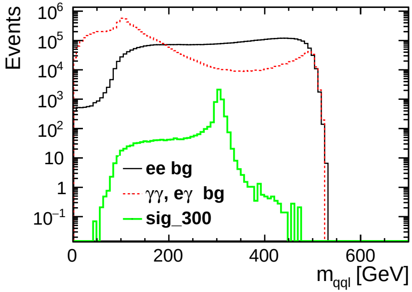
Search for heavy neutrino production:

$$e^+e^- \rightarrow N\,\nu \rightarrow W\,l\,\nu \rightarrow q\,q\,l\,\nu$$

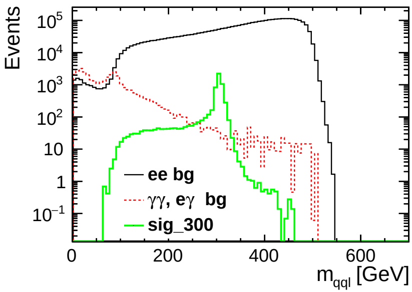
Main background contributions expected from SM processes:

$$e^+e^- \rightarrow W^+W^- \rightarrow q\,q\,l\,\nu \qquad\qquad \gamma\, e^\pm \rightarrow q\,q\,l$$

Electron channel



Muon channel



Tips and tricks

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:

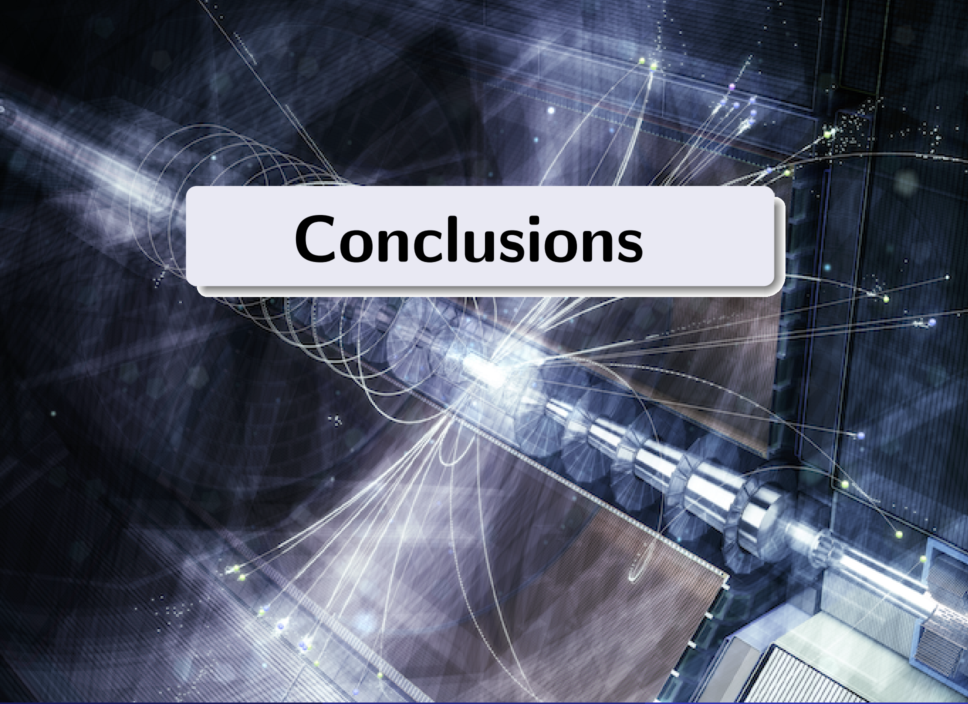
```
module PdgCodeFilter NeutrinoFilter {

    ...

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

  add PdgCode {35}

}
```



Conclusions

Conclusions

ILCgen detector model for Delphes based on parameterisation of full simulation results gives realistic description of the ILC experiment(s)

Detailed description of the detector acceptance, including forward region

Reliable predictions of the detector response for "standard" final states.

Precise jet flavour tagging parameterisation
Note: correlations between tagging results are not modeled !!!

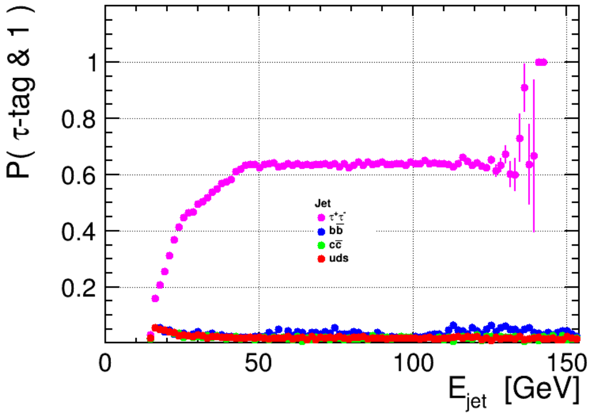
Can be used to get realistic results

- when very detailed detector response simulation is not relevant
- before more involved full simulation studies are undertaken
- to extrapolate full simulation results
  e.g. when scanning BSM model parameter space
- for studies where use of full simulation is not feasible

ILCgen model documentation: https://github.com/iLCSoft/ILCDelphes

Jet flavour tagging

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



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