Tutorials of Physics Analyses on CEPC ref-TDR

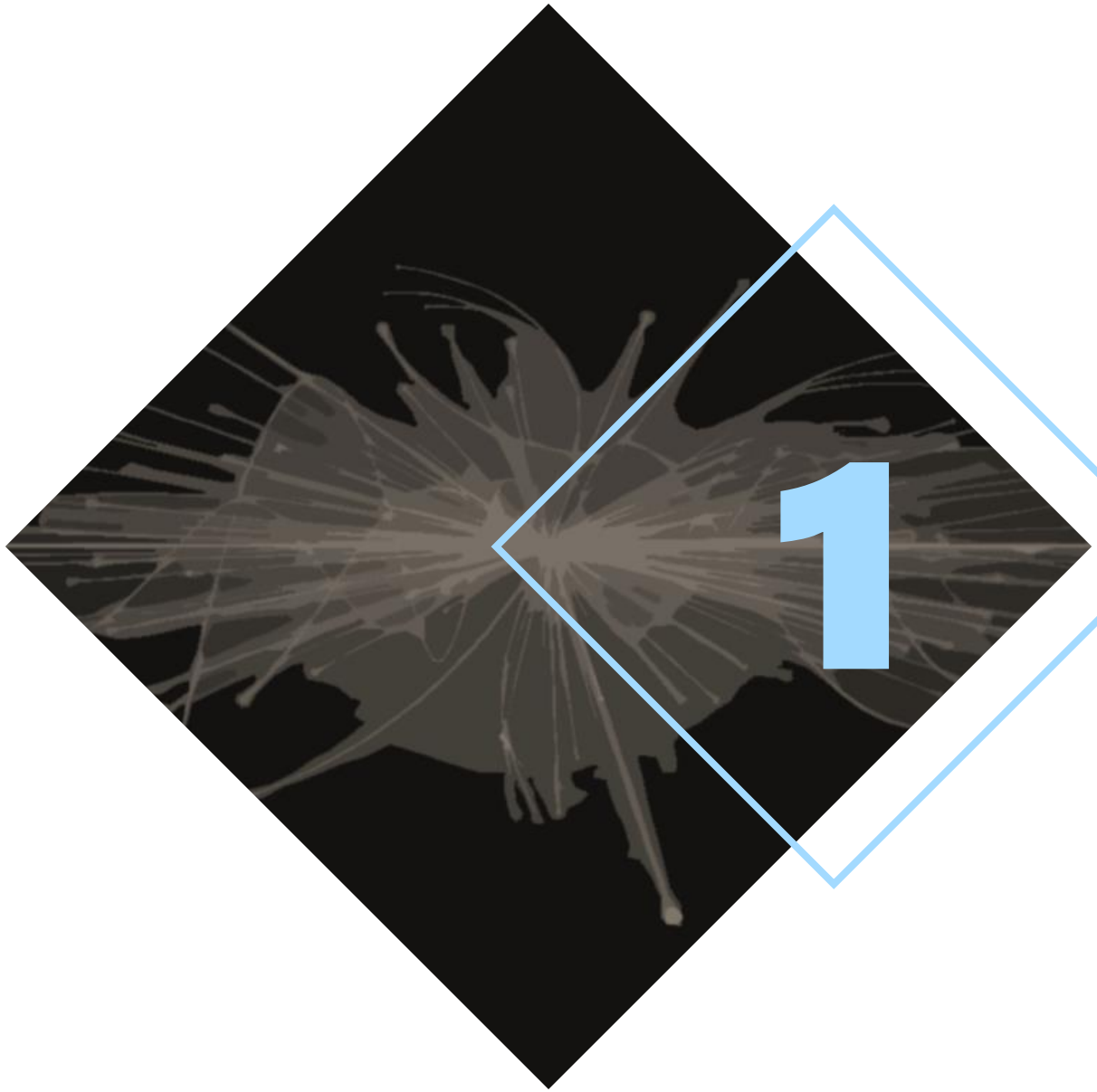
中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

Geliang Liu (刘格良)

Mar. 4th, 2025

Outlines

- CEPCSW
- Physics analyses for ref-Detector TDR
- Sample production
- Event selection
- Statistical model and inferences



CEPCSW

- Offline software prototype

Introduction to CEPCSW

An offline software prototype for CEPC ref-detector

- Available on “IHEP gitlab”: https://code.ihep.ac.cn/cepc/CEPCSW

Functions

- Perform full simulation of the CEPC ref-detector
  Perform offline event reconstruction
- Perform physics analyses, accompanied with tools you are familiar with

Structure

- Detector: detector geometry
- Generator: simulate physics processes
- Simulation: simulate detector responses
- Digitization: digitize the analog signals
- Reconstruction: particle flow algorithm, particle ID
- Analysis: analyze data after event reconstruction
- Examples: simple examples to get familiar

Framework

- Based on key4hep: Turnkey Software for Future Colliders
- Gaudi framework to build data processing application
- Let’s keep it short and easy:
  - **Write the algorithm in C++** which contains how you would like to process the data
  - **Build the algorithm** as a module with cmake
  - Use a python script to **call and run the module**
  - Get the output

Let's get start

Beforhand

- ssh -XY username@lxlogin.ihep.ac.cn
- cd /cefs/higgs/username/
- Don't forget to create a SSH key and put it in your account on code.ihep.ac.cn and github

Installation

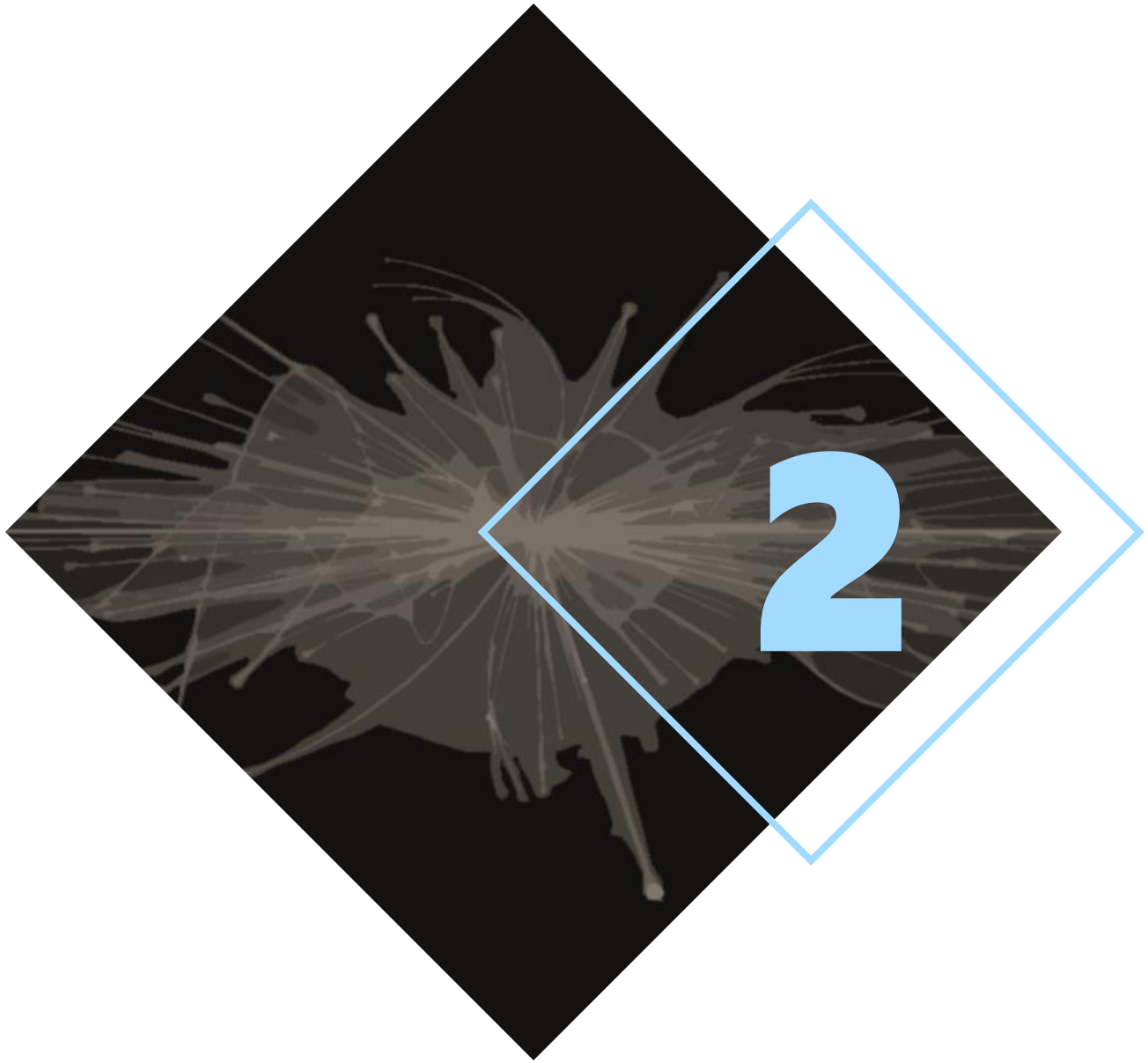
- **The version for this tutorial based on 25.1.2 and updates of Fangyi:**
  - git clone git@code.ihep.ac.cn:glliu/CEPCSW.git
  - cd CEPCSW
  - git checkout CyberPFA-5.0.2-dev
- **A new official version (25.3.0) has been available**

Build

- source setup.sh
- ./build.sh
- source setup.sh
- Every time you change something, don't forget to run ./build.sh

Example

- **Run the python script:** ./run.sh Examples/options/helloalg.py
- Very simple: print out an int number

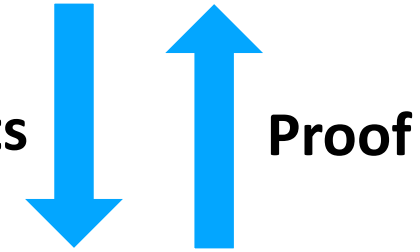


Physics Analyses for CEPC ref-Detector TDR

Physics performance

➢ Detector performances

- Tracking / vertexing / PID / jet clustering / jet flavor tagging



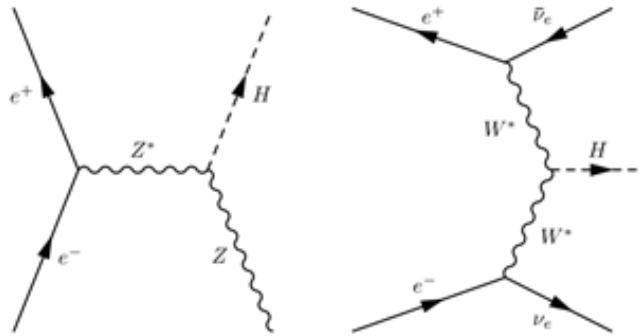
➢ Physics performances

- **Analyses of physics benchmarks**
- **Proof of detector performances**
  - H→qq: jet clustering / flavor tagging
  - H→γγ: ECAL energy measurements
  - H→invisible: overall PFA performances
- **Ability to measure/search for important physics**
  - Higgs boson recoil mass: inclusive measurements; pure measurement on gZZ
  - Z / top precise measurements
  - .....

Focus: Higgs boson invisible decay

➢ Higgs boson production

- Higgs strahlung: ee→Z*→ZH (dominant)
- W fusion: ee→$\nu_e\nu_e$H (sub-dominant)

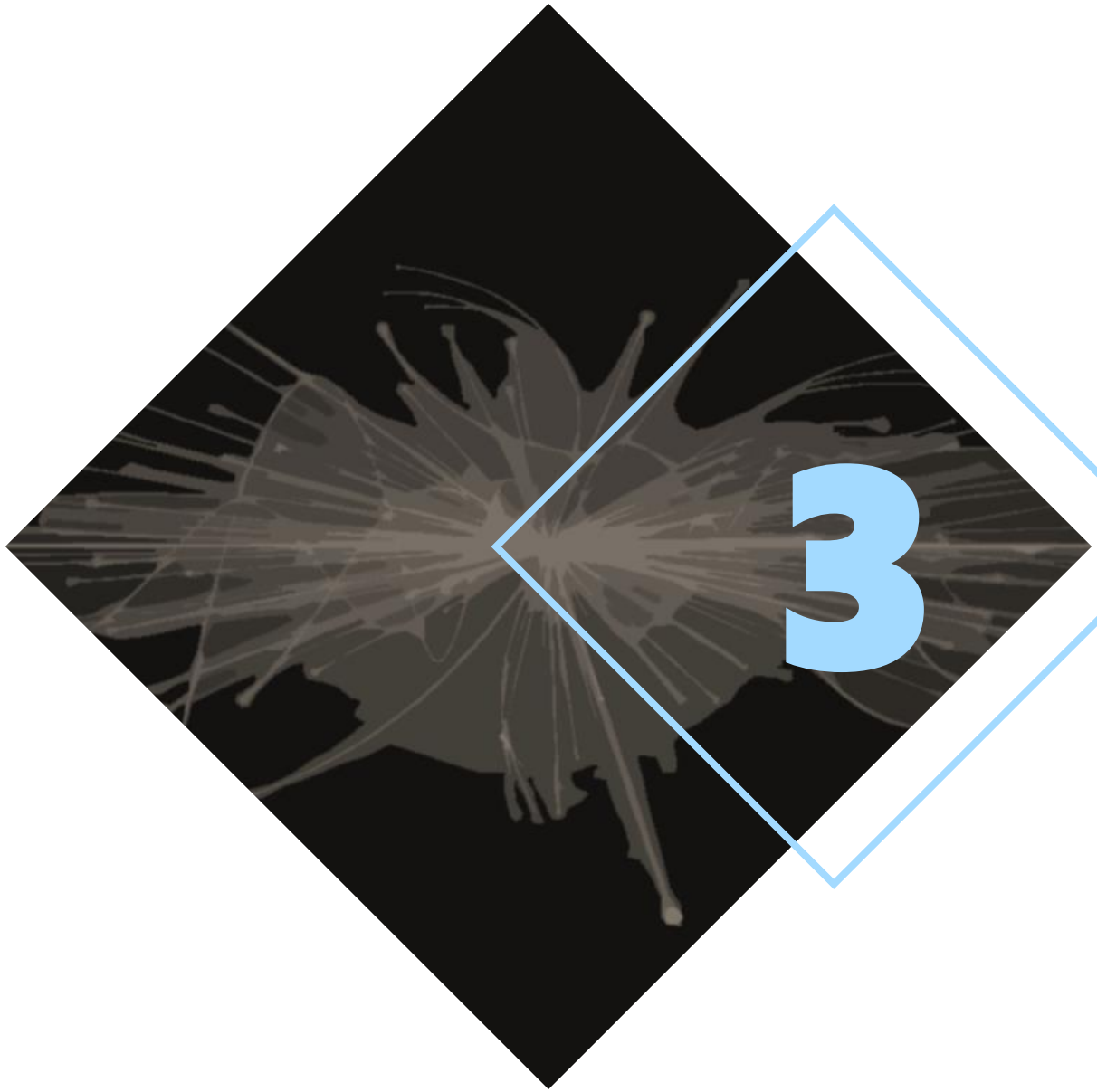


➢ Invisible decays

- ee→Z(→ee/μμ/qq)H(→invisible)
- In the SM: H→ZZ*→4ν
  - BR(H→4ν)=0.106%
- BSM: H→sparticles / dark matter / LLPs, ...

➢ Previous studies

Exps	Data	Results	Publication
ATLAS	LHC Run 2	UL on BR(H→inv): 10%	JHEP08(2022)104
CMS	LHC Run 2	UL on BR(H→inv): 10%	PRD 105 (2022) 092007
ILC	250, 350, 500 GeV; 250, 350, 500 fb-1	UL on BR(H→inv): 0.26%	arXiv:1909.07537
FCC-ee	240+365 GeV; 10.8+3 ab-1	3.9 σ on BR(H→ZZ→4ν)	Presentation
CEPC	240 GeV, 5.6 ab-1	UL on BR(H→inv): 0.26%	Chinese Phys. C 44 123001

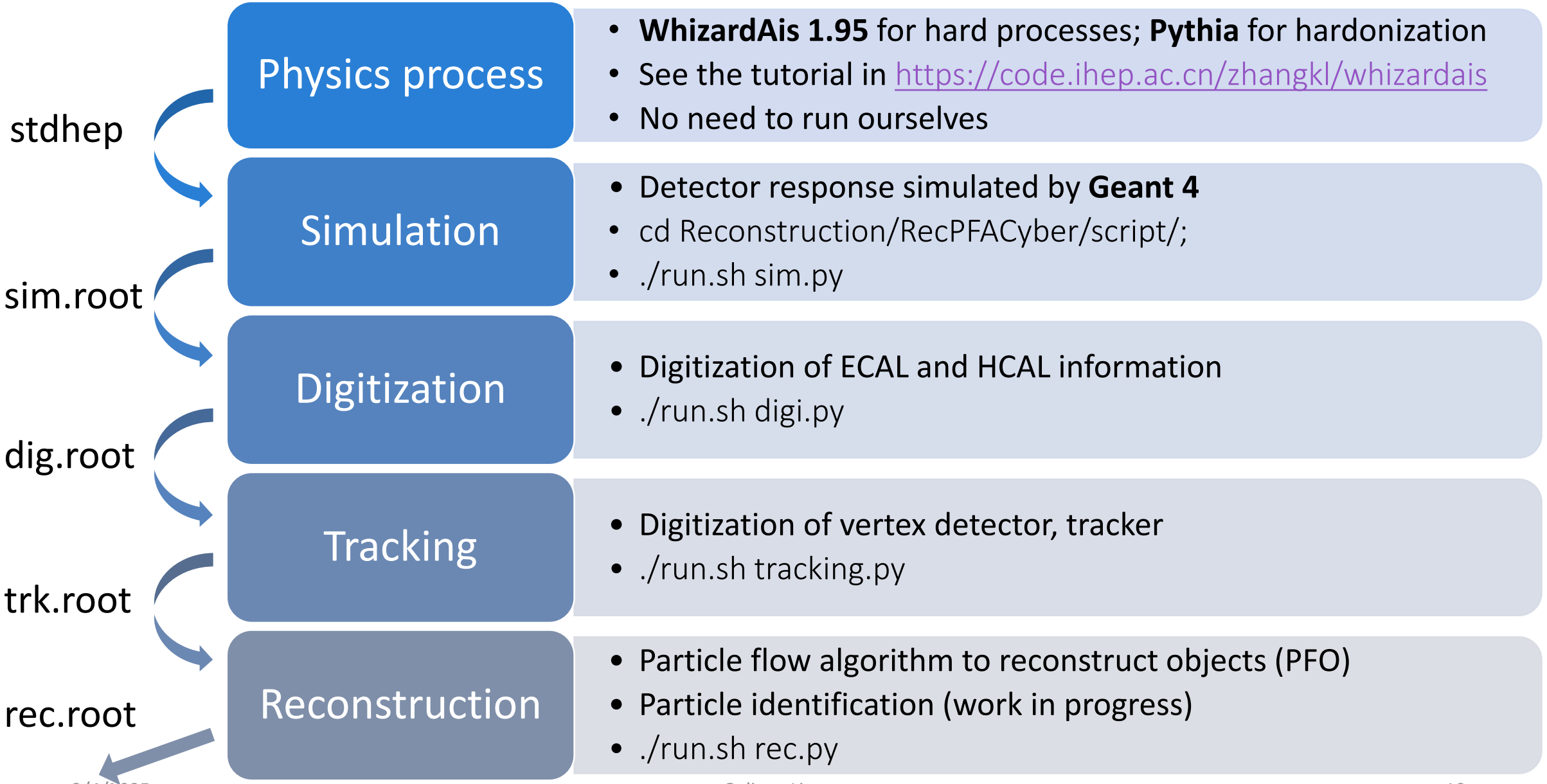


3

Sample Production

- What you need first to do analyses

Steps (See Kaili's tutorial)



Job submission

➢ Download the package

- git clone git@code.ihep.ac.cn:glliu/cepcsw_tutorial.git
- cd cepcsw_tutorial
- git checkout master

➢ Understand what this is doing

- check jobs/CEPC_Hinvi.sh
  - Take the inputs of stdhep from Kaili (each file 1000 events, split into 5 jobs; iFile defines N_Files to be read)
  - Five steps: sim → digi → trk → rec → tre (TTree production, to read rec.root, introduced later)
  - Write sh files to source CEPCSW environment and run the job
  - Use hep_sub to submit jobs
- Modify yourself:
  - Change CurrPath to the folder you want to save the samples
  - Change the CEPCSW environment to your own path

➢ Job launching

- cd /path/to/your/CEPCSW/repository; source setup.sh
- cd /path/to/your/cepc_tutorial/jobs
- source CEPC_Hinvi.sh
- Check job status: hep_q –g higgs –u username

➢ WARNING !!!

- Do NOT direct run!
- Make sure you made the changes!
- Make sure iFile=1 for check first!

Available samples

➢ stdhep (hard process + hadronization):

- Produced by Kaili
- On lxlogin: /cefs/higgs/zhangkl/stdhep/
- No need to produce ourselves

➢ Full simulated and reconstructed samples:

- I have produced some for both **Higgs boson invisible decays** and backgrounds
- Under git@code.ihep.ac.cn:glliu/CEPCSW.git
- On lxlogin: /cefs/higgs/liugeliang/CEPC/202501/Production
  - Hinvi/: **signal samples**
  - 4fermions/, 2fermions/: **4-fermion and 2-fermion-final-state background samples**
  - HX/: **inclusive Higgs boson decays (another background)**
- With limited statistics
  - Production en masse after new CEPCSW release

➢ Sample name conventions

- For explanation: /cefs/higgs/zhangkl/stdhep/cepc_sample_note_latest.pdf

How to read samples

Output of reconstruction

- In the repository for each process: Combined/rec*root are the output from reconstruction.
- As a TTree, and can be directly read.
- A more convenient way is to read it in the format of edm4hep: documentation

Example

- https://code.ihep.ac.cn/glliu/CEPCSW/-/blob/CyberPFA-5.0.2-dev/Analysis/MissingET

Understand the python script

- cd Analysis/MissingET/
- Check MissingET.py
- Input files:

```
podioevent = k4DataSvc("EventDataSvc", input="/cefs/higgs/liugeliang/CEPC/202501/Production/Hinvi/E2
```

- Input collections from the files:

```
from Configurables import PodioInput
inp = PodioInput("InputReader")
inp.collections = [ "CyberPFOPID", "MCParticle" , "RecTofCollection", "DndxTracks"]
```

- Inherit the algorithm:

```
from Configurables import MissingET
missingET = MissingET("MissingET")
```

- Output:

```
missingET.OutputFile = "E240_e1e1Hinvi_test.root"
```

How to read samples

➢ Understand the algorithm

- https://code.ihep.ac.cn/glliu/CEPCSW/-/blob/CyberPFA-5.0.2-dev/Analysis/MissingET/src/:
- Input collections:

```
DataHandle<edm4hep::ReconstructedParticleCollection> m_PFOColHdl{"CyberPFOPID", Gaudi::DataHandle::Reader, this};
DataHandle<edm4hep::MCParticleCollection> m_MCParticleGenHdl{"MCParticle", Gaudi::DataHandle::Reader, this};
DataHandle<edm4hep::RecTofCollection> m_inTofCol{"RecTofCollection", Gaudi::DataHandle::Reader, this};
DataHandle<edm4hep::RecDqdxCollection> m_inDqdxCol{"DndxTracks", Gaudi::DataHandle::Reader, this};
```

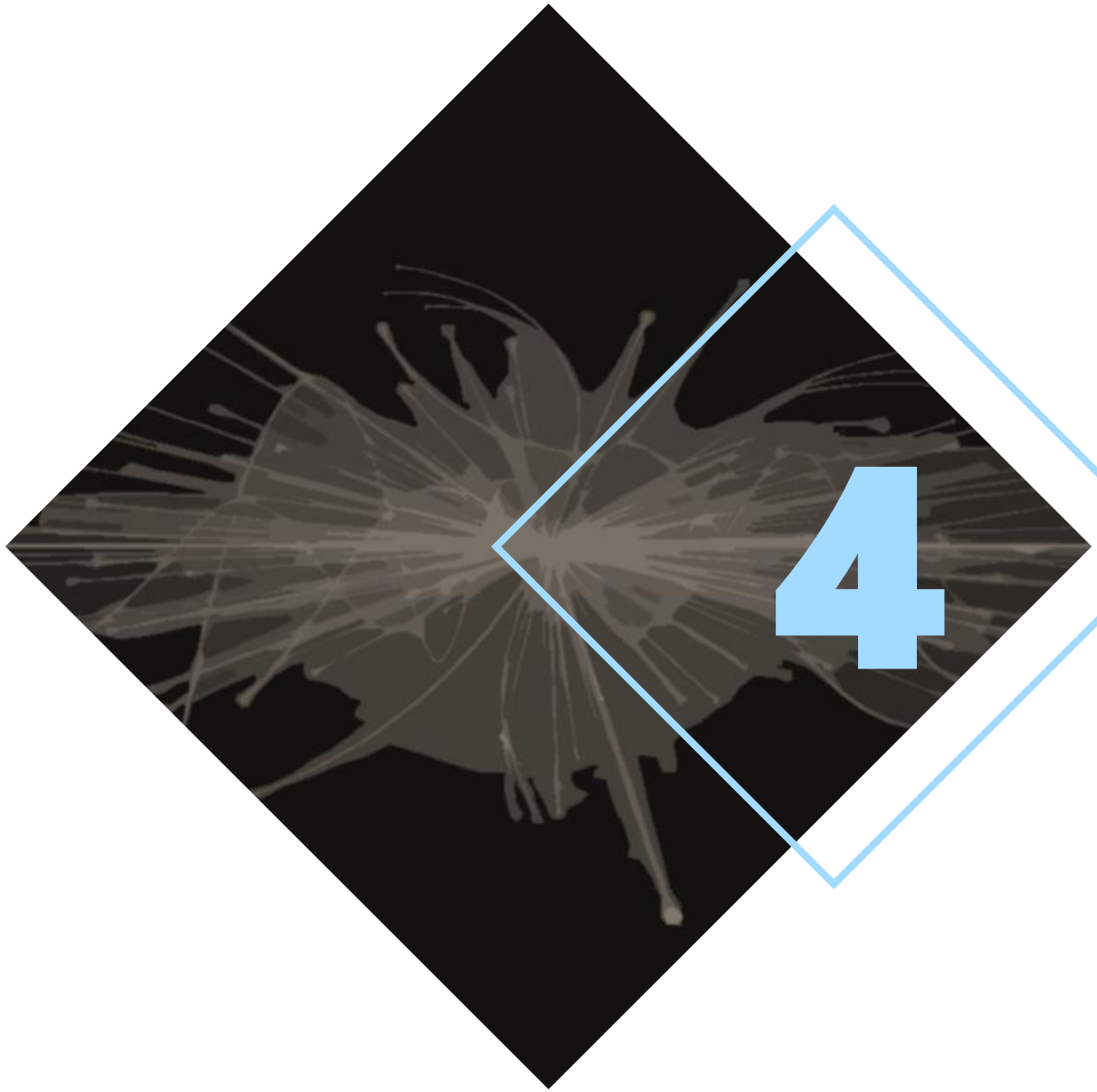
- Interferable properties:

```
Gaudi::Property<std::string> m_algo{this, "Algorithm", "ee_kt_algorithm"};
Gaudi::Property<int> m_nJets{this, "nJets", 2};
Gaudi::Property<double> m_R{this, "R", 0.6};
Gaudi::Property<std::string> m_outputFile{this, "OutputFile", "GenMatch.root"};
```

- What the algorithm does: compute/save some properties of simulation events in a new TTree and ROOT file.
  - MissingET.cpp#L195-277: loop over PFOs and compute/save their 4-momentum/isolation/charge/PID information..., as well as the visible 4-momentum.
  - MissingET.cpp#L279-396: jet clustering (ee-kt algorithm with Njet=2), and compute variables related to jet-substructure.
  - MissingET.cpp#L398-466: loop over MC particles and save their 4-momentum/charge/flavor/status...

➢ Try yourself

- ./run.sh Analysis/MissingET/MissingET.py
- Check the output!



4

Event selection

- Improve the sensitivity

Jupyter-notebook

- Feel free to choose your own tool/language to do analyses
- Let's use jupyter-notebook for this tutorial

➢ On lxlogin

- ssh -XY username@lxlogin.ihep.ac.cn -L 8888:localhost: 8888
- Change 8888 to your favorite number
- cd /path/to/your/CEPCSW/repository/
- source setup.sh
- export JUPYTER_DATA_DIR=$(mktemp -d)
- export JUPYTER_CONFIG_DIR=$(mktemp -d)
- jupyter-notebook --port 8888 (don't forget to change the number)
- You should be able to open the link it generates, starting with http://localhost:8888/?token=.....
- If it doesn't work, restart your terminal, and change to a new port number.

➢ Notebooks for tutorials

- Go to Analysis/HiggsInvisible/
- MissingET.ipynb: draw some plots of key variables in the Higgs boson invisible decay.
- LeptonID.ipynb: studies of temperary solutions for lepton identification.
- EventSelection.ipynb: perform event selection of the invisible decay analysis and build the statistical model.

Understand the signal

The signal features

- Signal process: ee→Z(→ee/μμ/qq)H(→invisible)
  - Z→ττ is not considered for now
- 2-fermion final state
- A large missing energy / momentum: can we reconstruct the Higgs boson as the missing mass?

Key variables

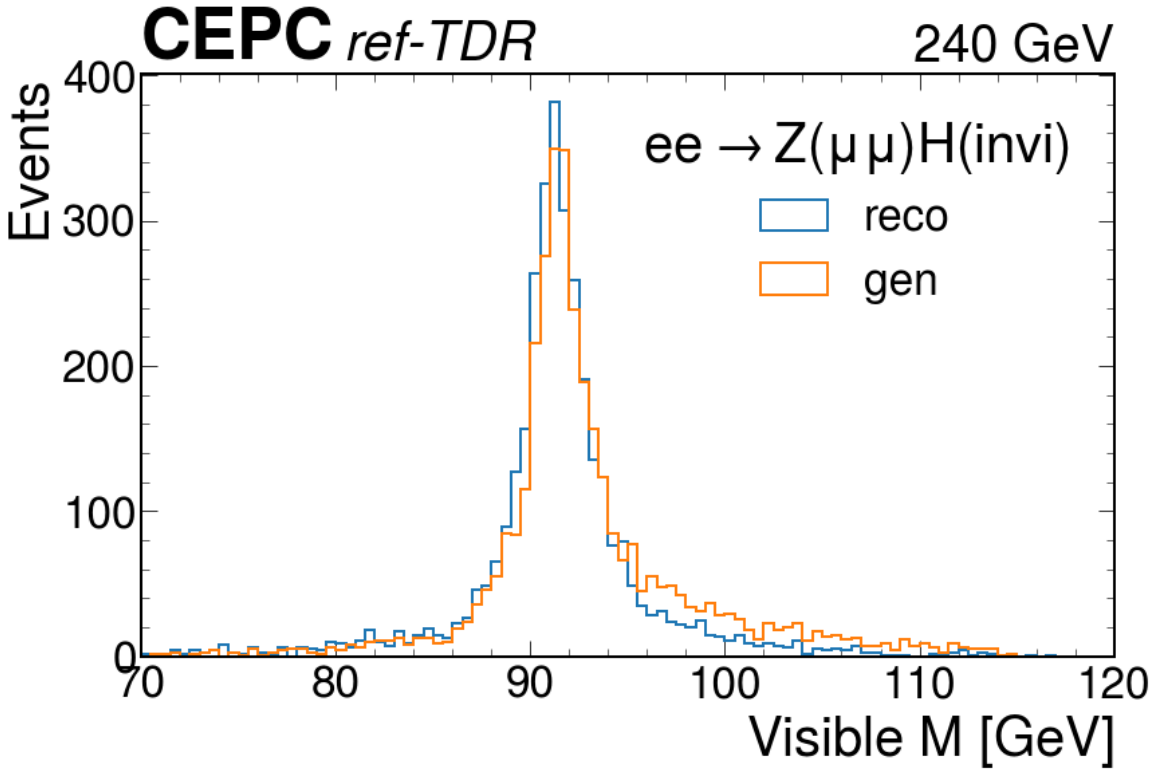
- Visible 4-momentum: $\mathrm{p}^{\mathrm{vis}} = \sum_{\mathrm{i}}^{\mathrm{PFO}} \mathrm{p}_{\mathrm{i}}$
- Missing 4-momentum: $\mathrm{p}^{\mathrm{mis}} = \mathrm{p}^{\mathrm{tot}} - \mathrm{p}^{\mathrm{vis}}$, $\mathrm{p}^{\mathrm{tot}} = (0, 0, 0, 240\ \mathrm{GeV})$

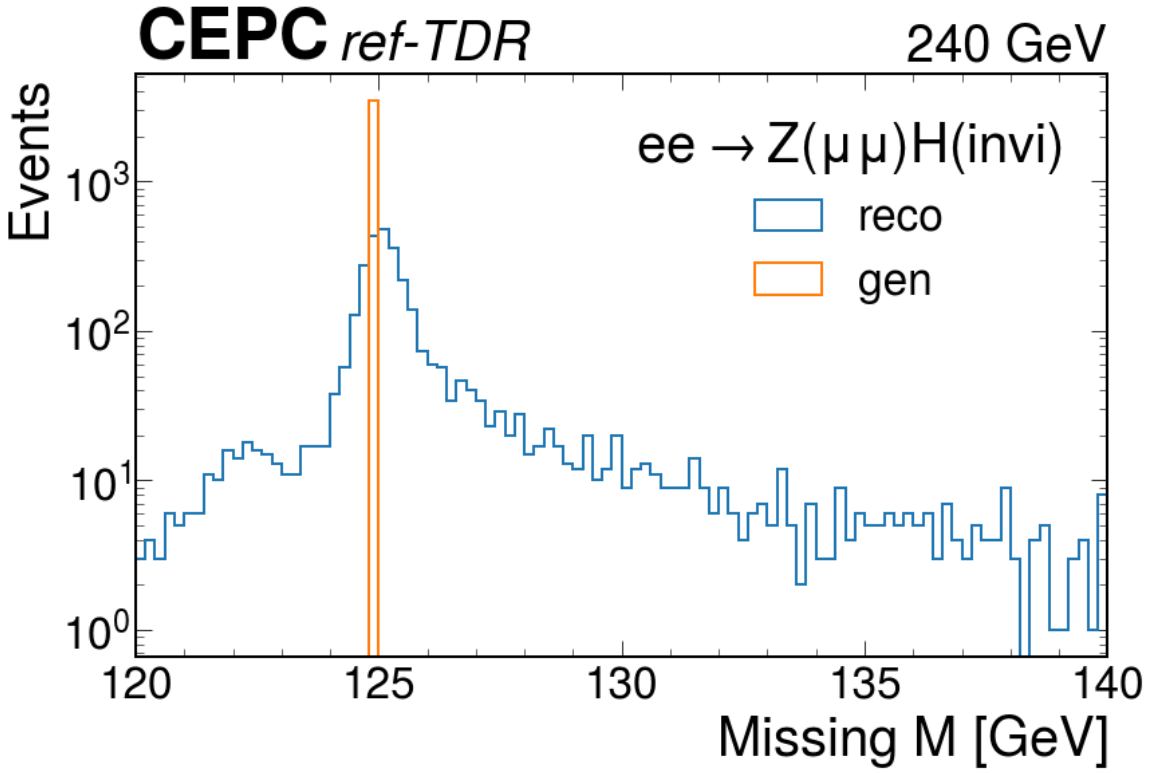
Computation

- Check Analysis/HiggsInvisible/MissingET.ipynb for computing px, py, pz, pt, p, E or M of these 4-momenta
- Reco-level: loop over PFOs
- Gen-level: loop over MC particles with status==1 (final state) and PDG!=12/14/16 (no neutrinos)
- **Run the notebook!**

Understand the signal

➢ What you may get (take μμ final state as an example)





- The visible mass is around **91 GeV**, while the missing mass is around **125 GeV**.
- **Check other variables and other signals!**

Baseline selection

- Check Analysis/HiggsInvisible/EventSelection.ipynb

➢ Prerequisites

- Necessary imports and settings (lumi=20000)

➢ Read files

- You need to read a complete set of samples, including 1) signals; 2) backgrounds, including 4-fermion, 2-fermion and Higgs boson visible decays.
- corresponding weights are computed: w = cross section × lumi / number of events produced
- **Feel free to run them**

➢ Lepton ID

- Since lepton ID is not mature enough, I did a coarse study on it, and use a temporary solution in this analysis. A combination of TOF + TPC + E_ECAL/p_track + E_HCAL/p_track information.
- Check the functions Chisq_E and Chisq_Mu about how χ^2 for electron and muon is computed.
- Check the functions eleID and muID for lepton ID, to be selected from PFOs:
  - $|\cos\theta| < 0.99$
  - Charged PFO
  - χ^2 90% efficiency working point: you may change the working point later to check the effect.

Baseline selection

➢ Baseline selection: ee μμ

- For the μμ selection, require exactly 2 muons passing muon ID, with opposite charge.
- For the ee selection, require exactly 2 electrons passing electron ID, with opposite charge.

➢ Variables: ee μμ

- A number of variables are computed and saved in numpy arrays which may be used for further selection.
- Include: lepton pt / p; number of charged / neutral PFOs; ΔR, Δϕ of the two leptons; di-lepton 4-momentum and their recoil; visible and missing 4-momentum; ID score and isolation.
- **Fill in variables**: you may try only the signals, but it takes very long time.
- Skip to **Read variables from saved outputs**: outputs saved in EventSection/, which can be directly read.
- **Make plots**: for each variable and final state, make a plot composing all backgrounds stacked, and each signal separated.
  - Used to check their distributions, and how discriminating between signals and backgrounds.

➢ Baseline selection: qq

- No such selection. The jet clustering algorithm always clusters two jets.

➢ Variables : qq

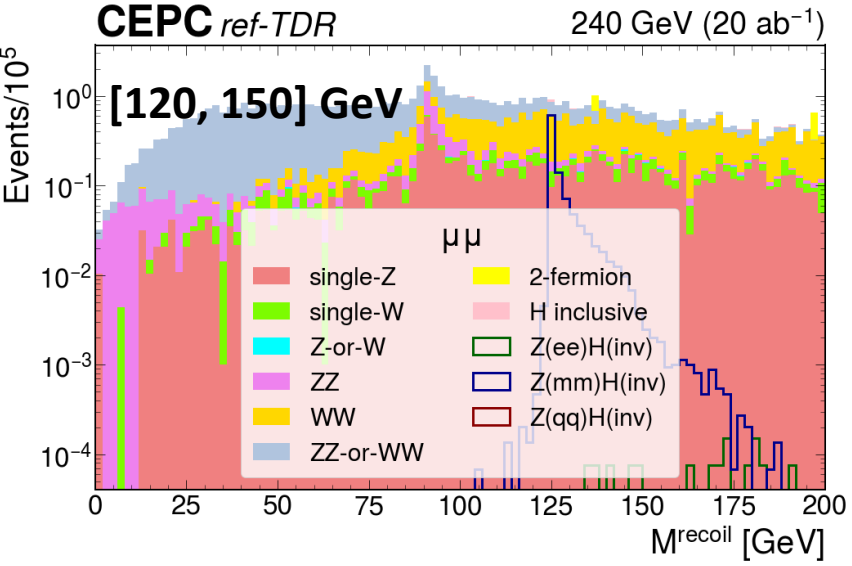
- Additional variables are computed: jet thrust, energy correlation functions (ECF), N-subjettinesses.

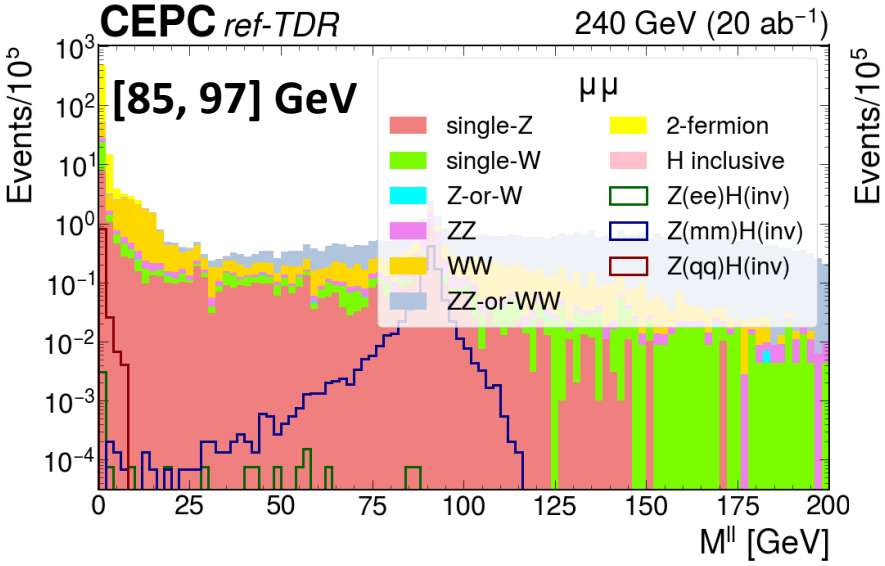
Kinematic selection

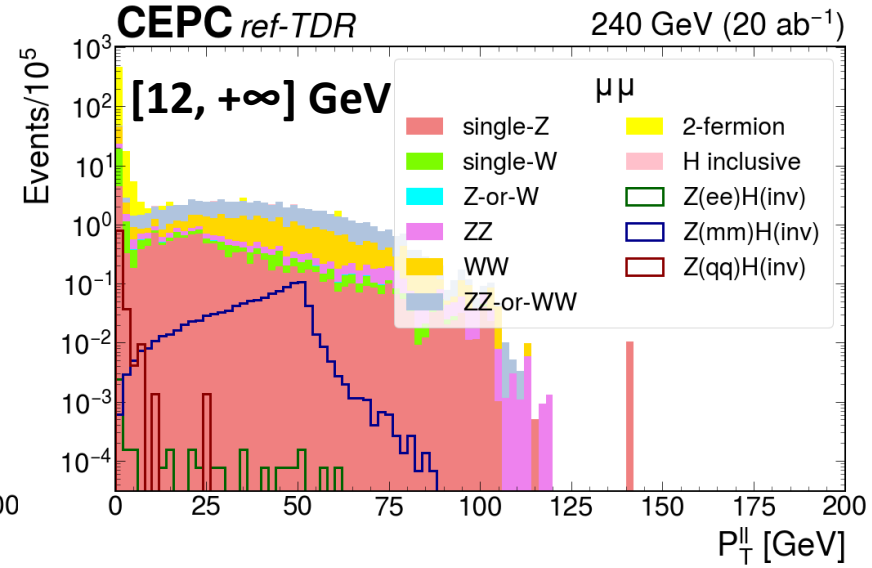
Kinematic selection

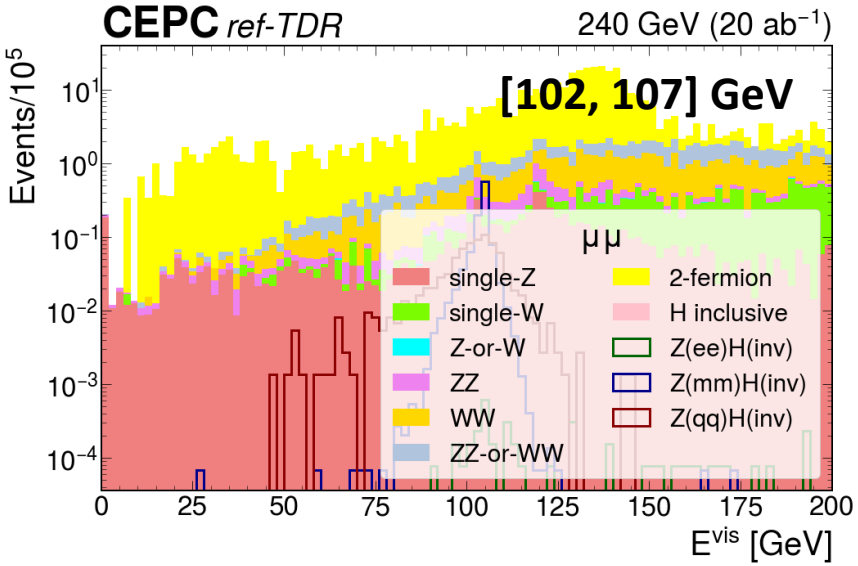
- Check function KelSel: different selection criteria are applied to multiple variables in different final states.
- Obtained from Chinese Phys. C 44 123001: H→invisible with CEPC CDR detector.
  - Selections to be **optimized**; but before doing it yourself, it is worth using the **selections in previous analyses** as starting points.
- **Print yields**: you may compute the yields and selection efficiencies. Compare with Chinese Phys. C 44 123001.
- **Plots after kinematic selection**: similar plots after kinematic selection. Check the effects!

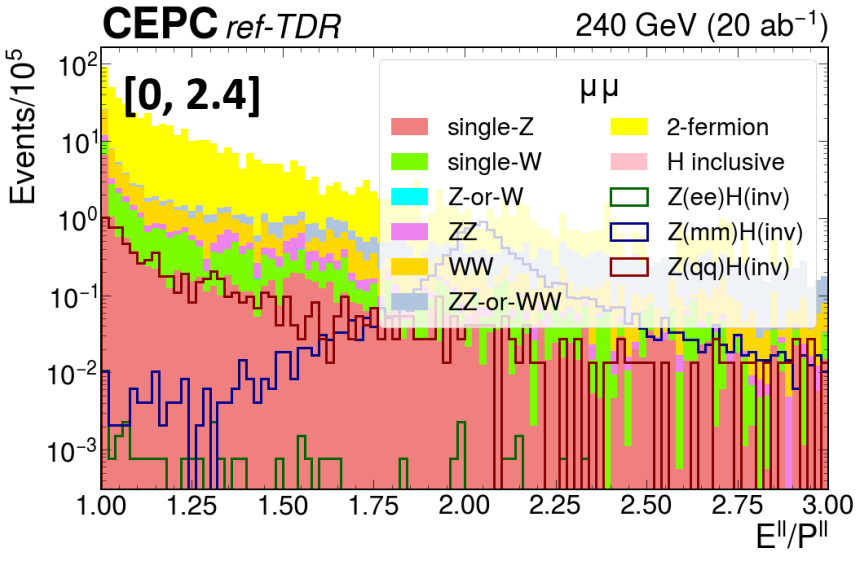
Kinematic selection: μμ







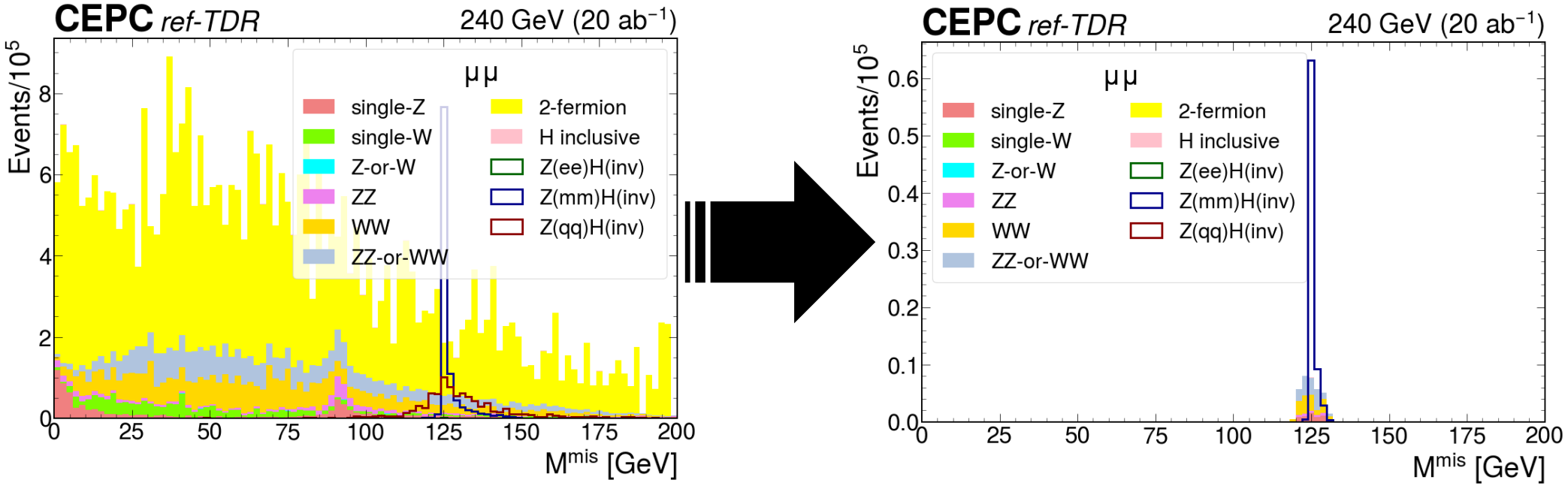




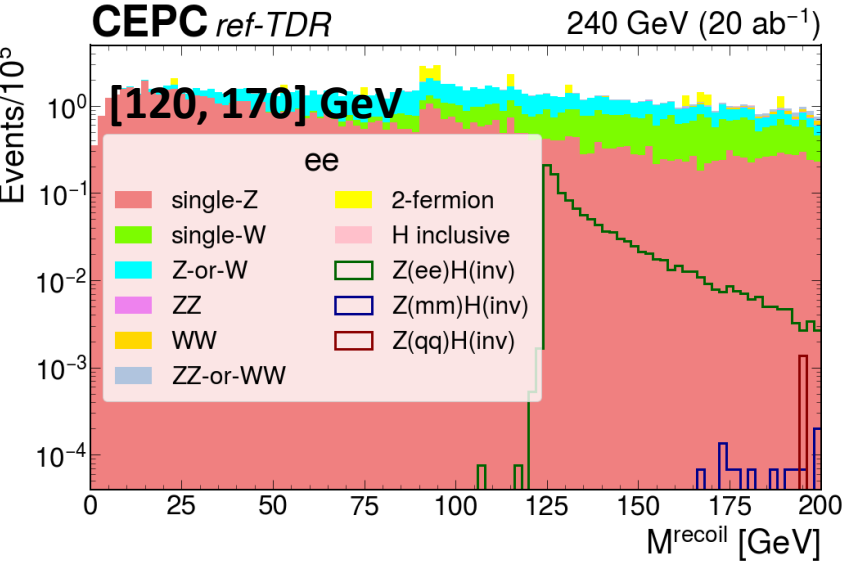
Kinematic selection: μμ

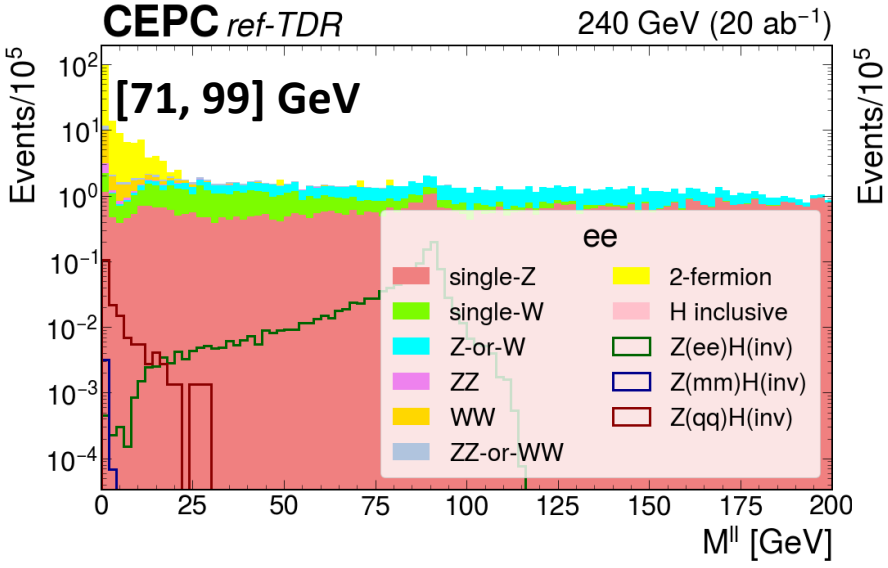
- μμ final state

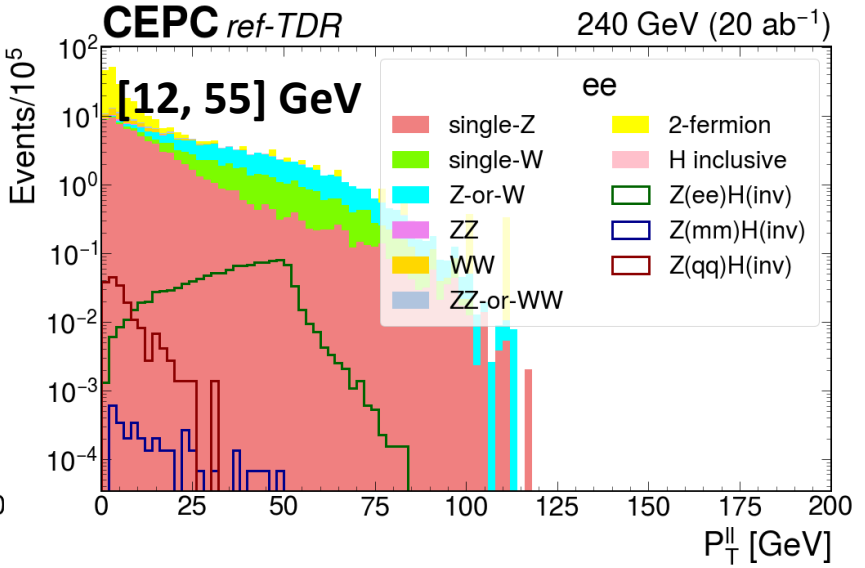
Process	eeH	mmH	qqH	sz	sw	szw	zz	ww	zzww	2f	Hincl	All bkg
Total yield	140800	135400	2736200	32403400	69705000	4989600	22819400	181522200	73003000	1082137200	4073200	1470653000
Base sel	406	106555	85917	1558339	1893099	14293	791039	4331596	3576077	44002567	34049	56201059
Kin sel	0	76004	0	5358	0	0	2188	12917	13525	0	47	34035
eff (%)	0.000	56.133	0.000	0.017	0.000	0.000	0.010	0.007	0.019	0.000	0.001	0.002
eff CDR (%)	-	59.17	-	0.01	0.00	0.00	0.01	0.01	0.02	0.000	0.00	0.00

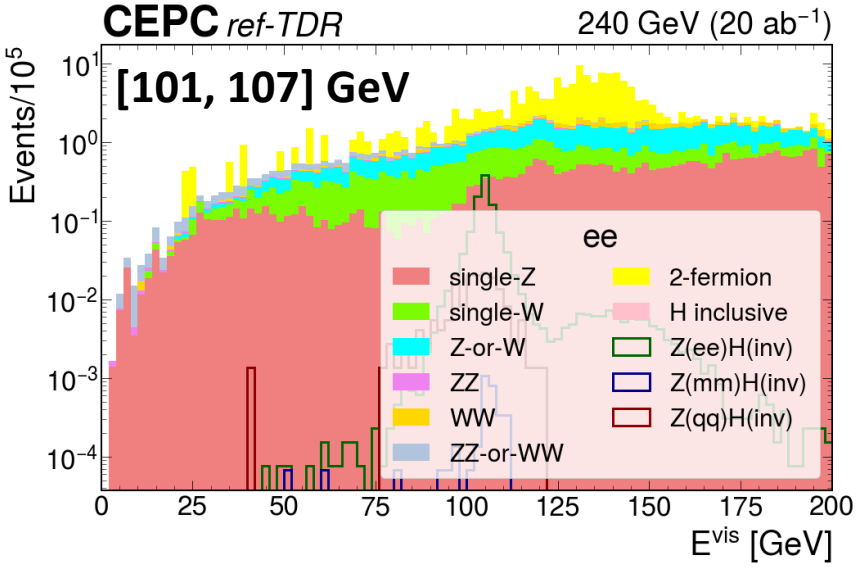


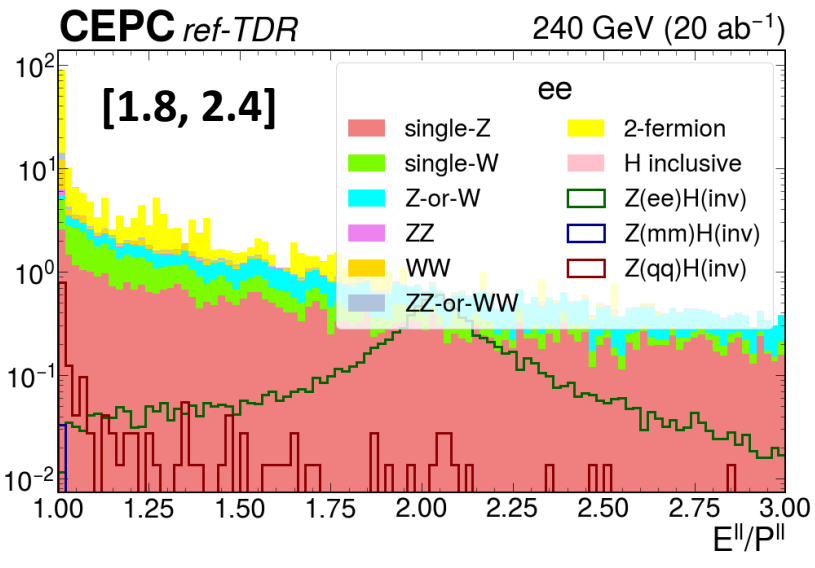
Kinematic selection: ee





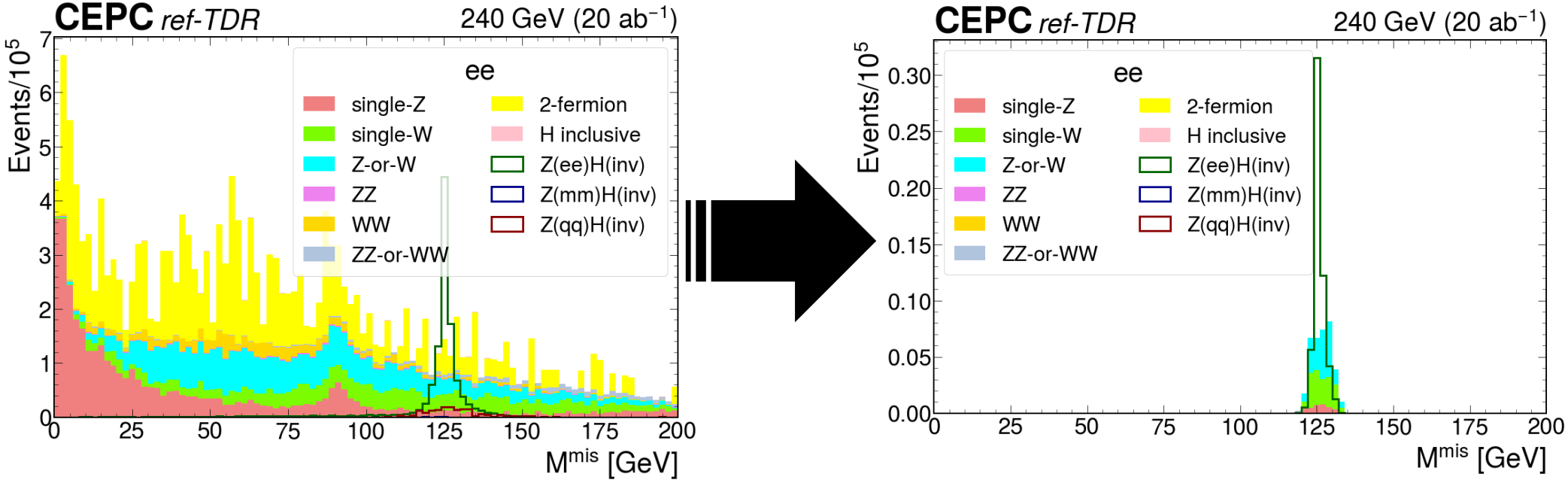




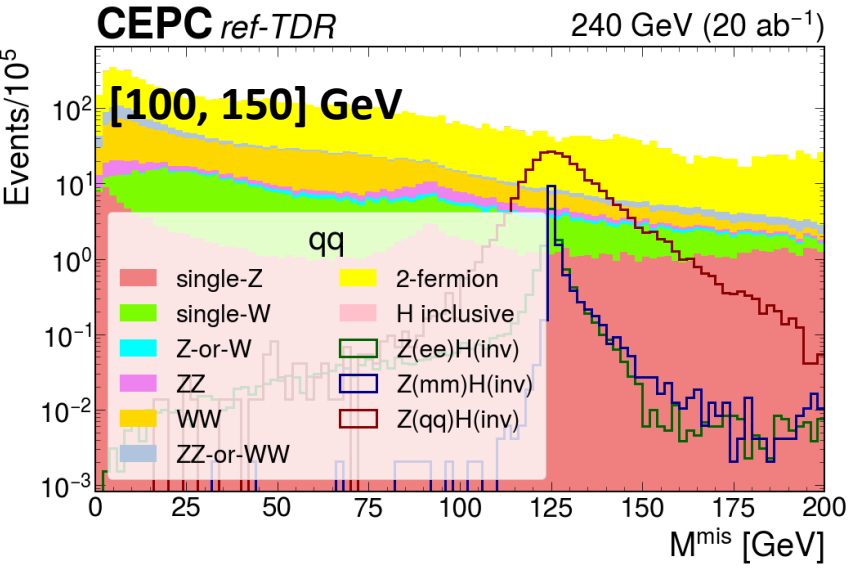


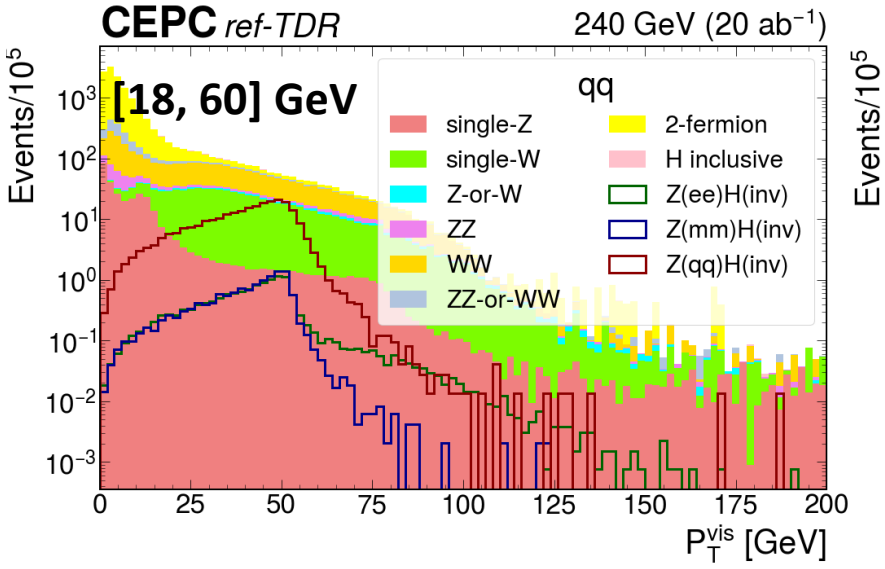
Kinematic selection: ee

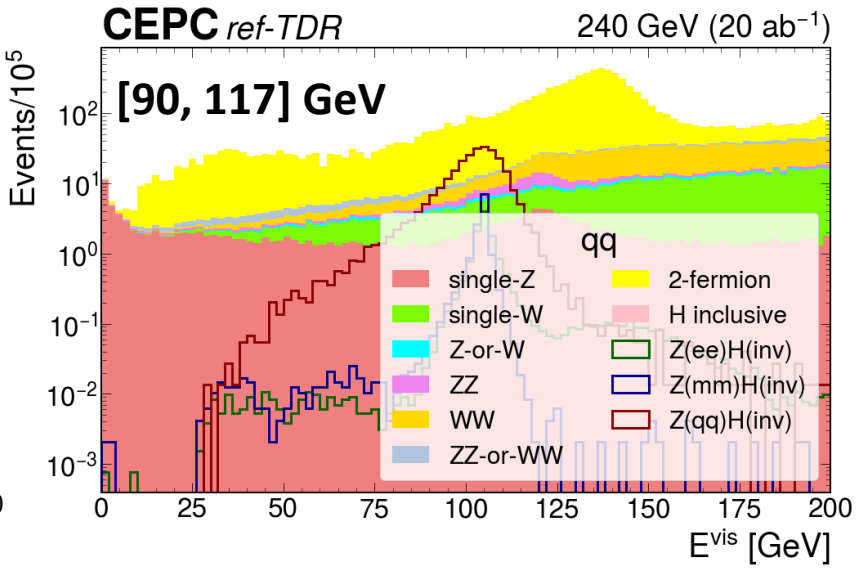
Process	eeH	mmH	qqH	sz	sw	szw	zz	ww	zzww	2f	Hincl	All bkg
Total yield	140800	135400	2736200	32403400	69705000	4989600	22819400	181522200	73003000	1082137200	4073200	1470653000
Base sel	118165	333	17785	7595862	2750212	4080038	161423	1092489	451954	12614069	23692	28769739
Kin sel	55517	0	0	3449	14227	17931	0	0	211	0	54	35872
eff (%)	39.429	0.000	0.000	0.011	0.020	0.359	0.000	0.000	0.000	0.000	0.001	0.002
eff CDR (%)	35.34	-	-	0.01	0.01	0.43	0.00	0.00	0.00	0.000	0.00	0.00

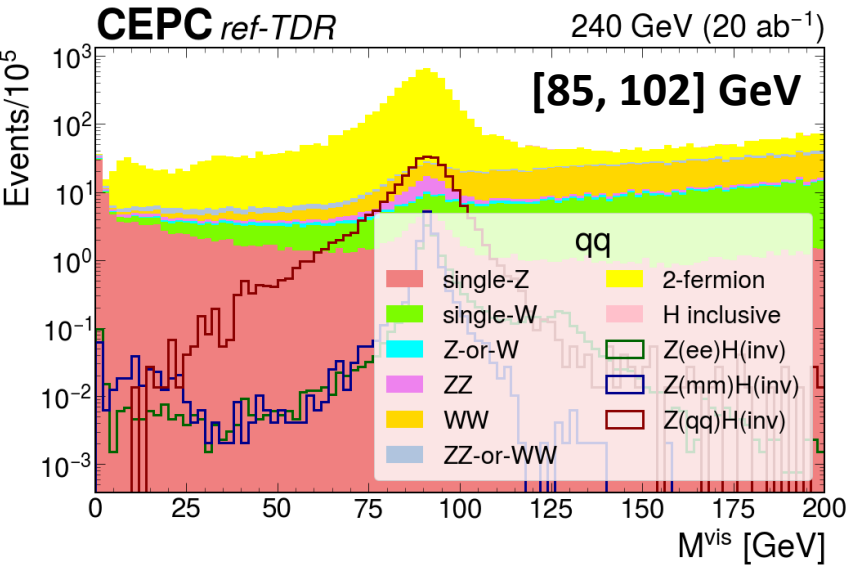


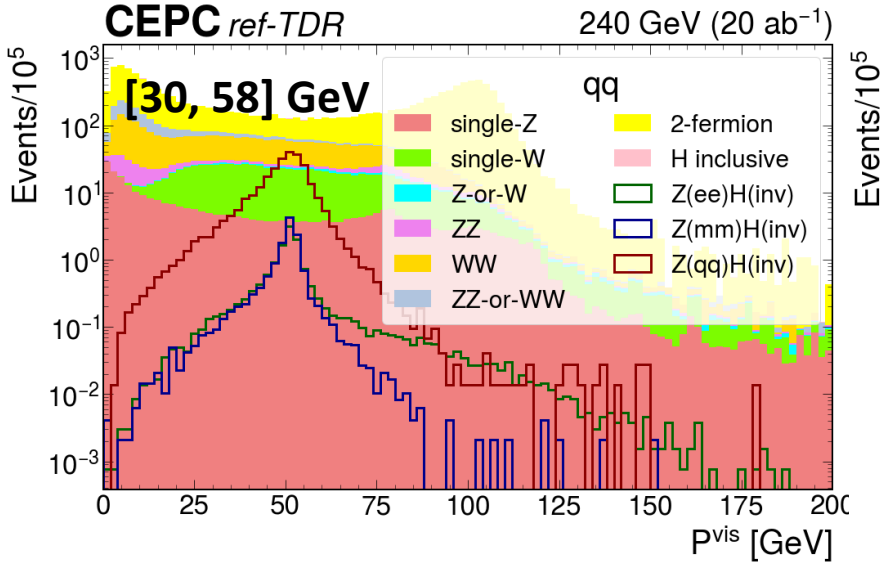
Kinematic selection: qq

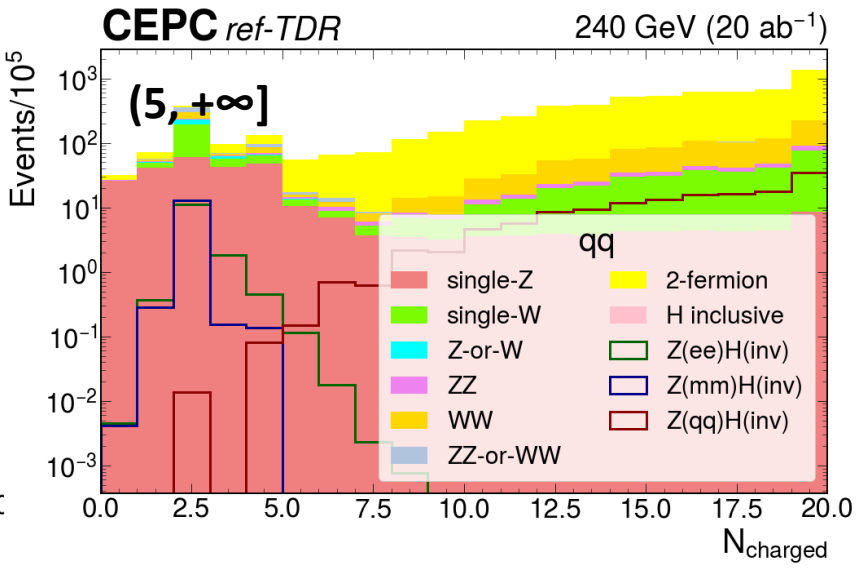






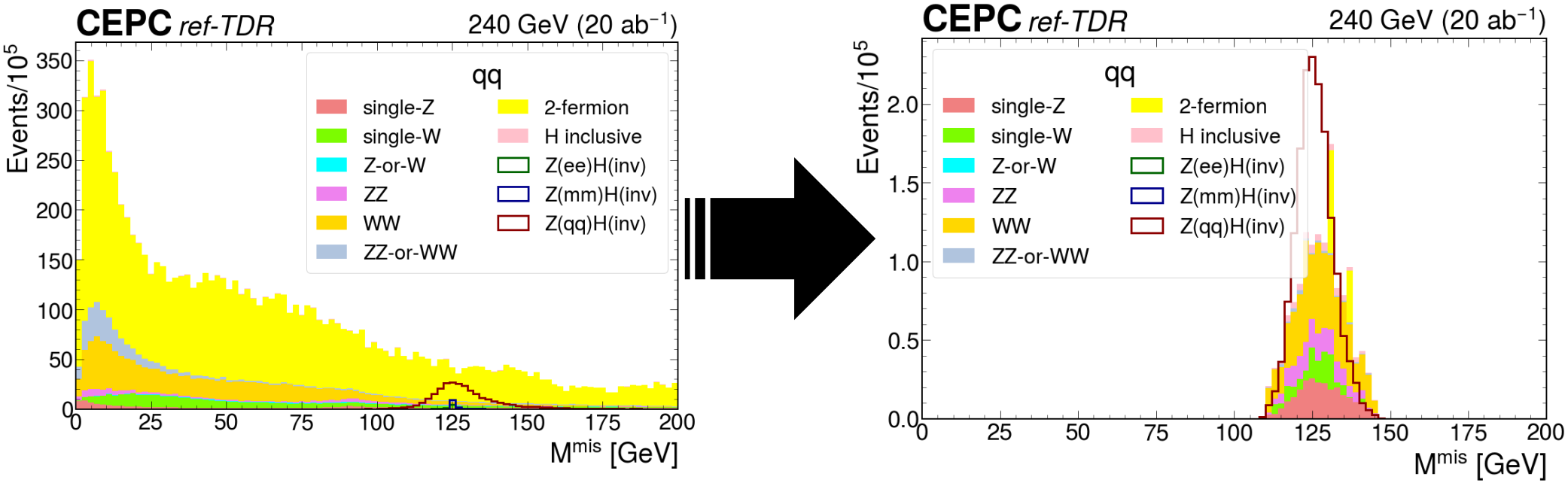


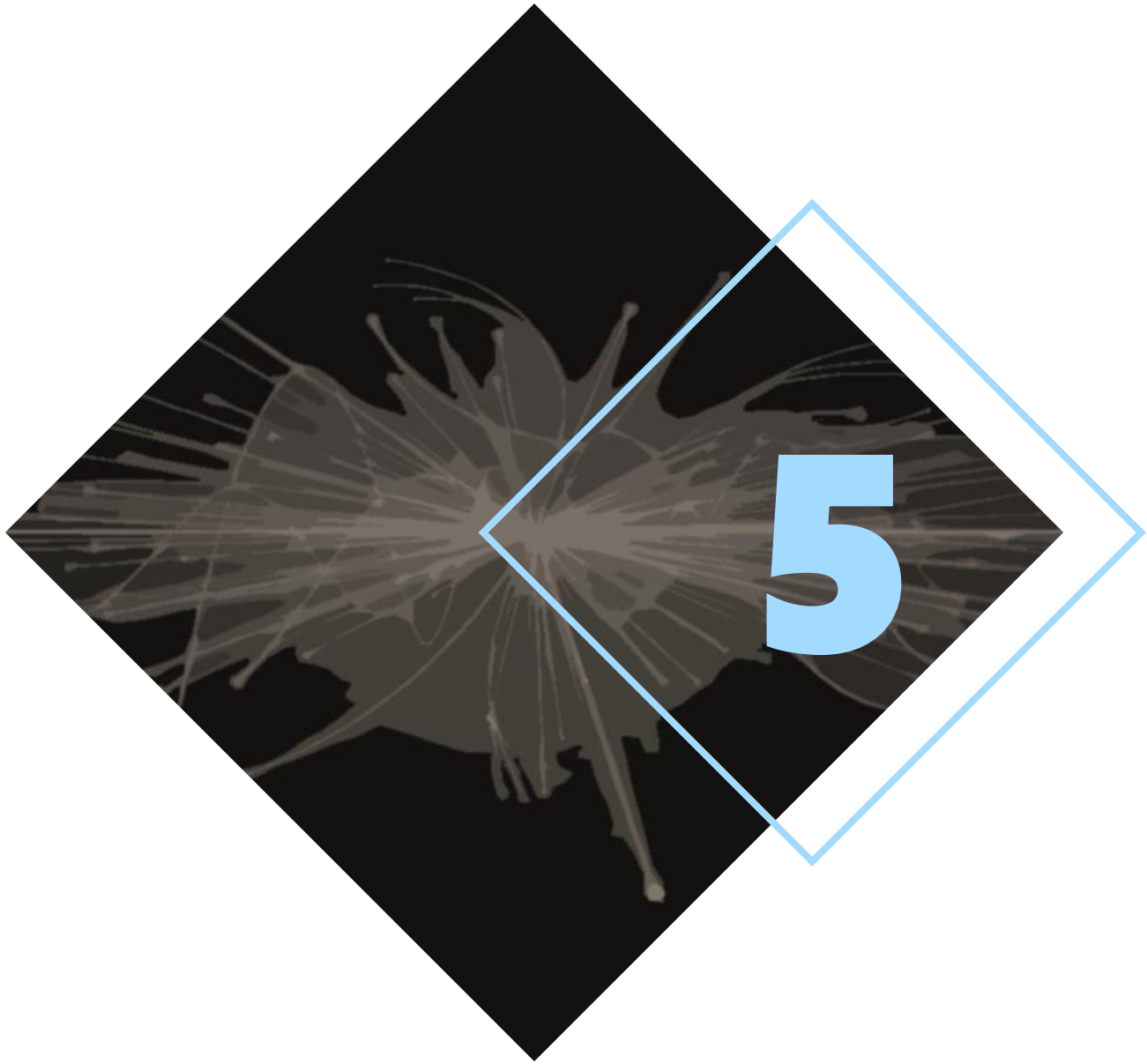




Kinematic selection: qq

Process	eeH	mmH	qqH	sz	sw	szw	zz	ww	zzww	2f	Hincl	All bkg
Total yield	140800	135400	2736200	32403400	69705000	4989600	22819400	181522200	73003000	1082137200	4073200	140800
Kin sel	23	0	1626945	244397	128022	0	182434	601706	20246	97783	55033	23
eff (%)	0.0163	0.000	59.460	0.754	0.184	0.000	0.800	0.332	0.028	0.009	1.351	0.0
eff CDR (%)	-	-	60.81	0.66	0.06	0.00	0.64	0.21	0.02	0.00	0.97	0.03



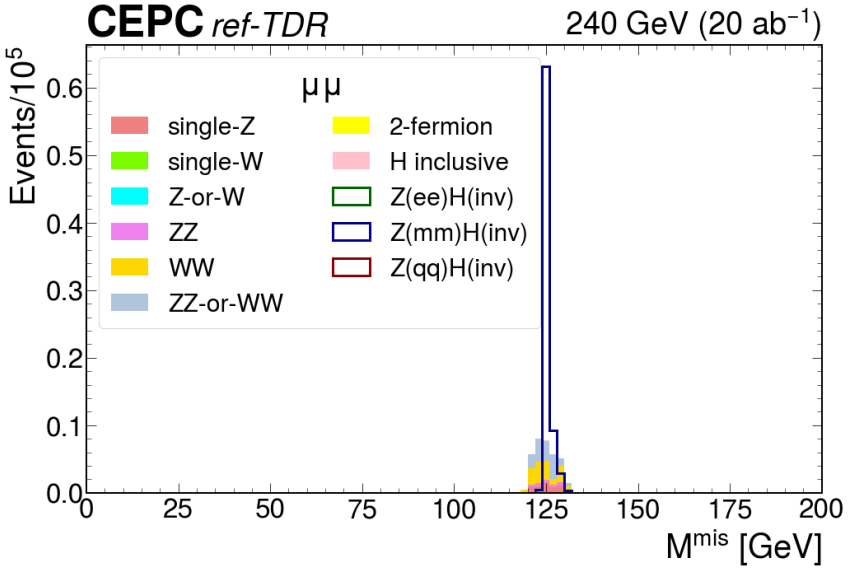


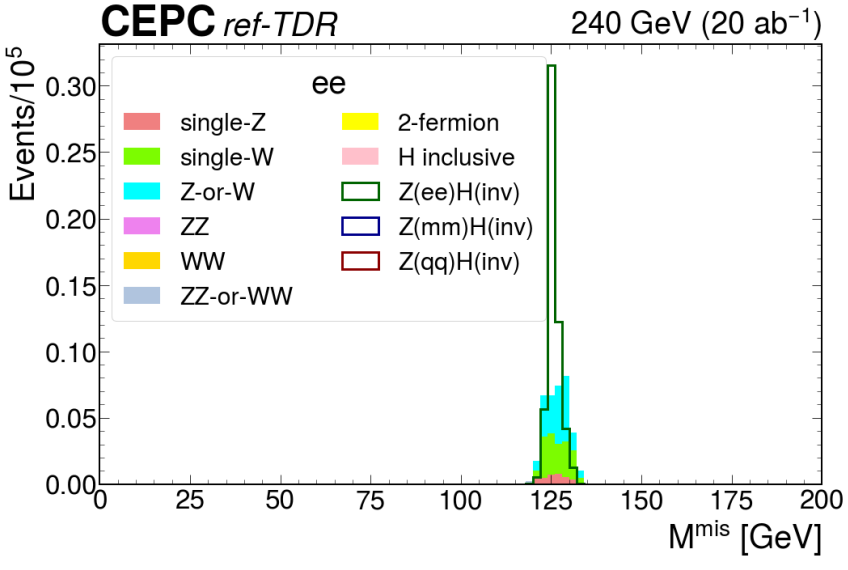
Statistical Model and inferences

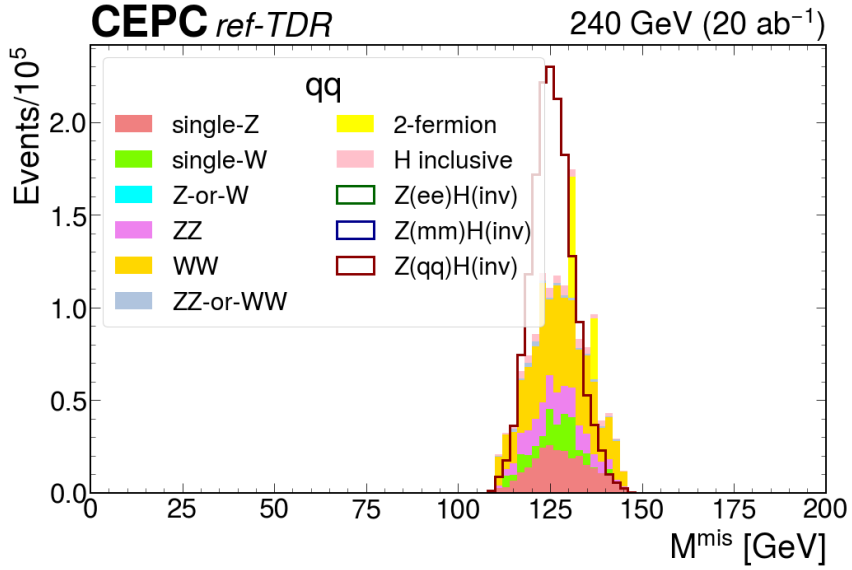
- Obtain the results

Discriminating variable

- You will need to fit on a certain variable to compute the results you want.
- It should have strong separation power between signals and backgrounds.
  - Typically, **a score from multivariate analysis**, or a **mass variable** (if resolution is good).
- In this tutorial, we choose the **missing mass**.
  - For signals, it is concentrated around 125 GeV.
  - For backgrounds, it is mostly flat, or concentrated somewhere else.







Statistical model

➢ Parameter of interest

- The variable you want to compute: branching ratio of H→invisible
- Practically, define the signal strength: $\boldsymbol{\mu = BR(H \to invisible)/BR_{SM}(H \to invisible)}$

➢ Binned likelihood fits

- You are fitting histograms, not functions!

$$L(\mu|D) = \prod_{i=1}^{N} \frac{(\mu S_i + B_i)^{D_i}}{D_i} e^{-(\mu S_i + B_i)}$$

Note: no systematic uncertainties are considered yet !!!

- N: number of bins.
- $\mu S_i + B_i$: expected yields in a bin, composing signals and backgrounds.
- D, D_i: the data to be fitted on.
- Aim: find the μ that minimizes $L(\mu|D)$.

➢ Expected results

- We don't have real data; instead, we fit on Asimov dataset: $D_i = S_i + B_i$
- We must obtain $\mu_{best} = 1$, but we care about its uncertainty.

The CMS Combine tool

➢ Introduction

- A compact tool for statistical analyses, with multiple usages.
- Open source, with an official publication: Comput Softw Big Sci 8, 19 (2024)
- Tutorial: https://cms-analysis.github.io/HiggsAnalysis-CombinedLimit/
- Again, feel free to choose the tool you are familiar with.

➢ Installation

- ssh -XY username@lxlogin.ihep.ac.cn
- cd /cefs/higgs/username/
- Note: do not enter a CEPCSW environment !!!
- CMS environment:
  - source /cvmfs/cms.cern.ch/cmsset_default.sh
  - cmsrel CMSSW_14_1_0_pre4
  - cd CMSSW_14_1_0_pre4/src
  - cmsenv
  - git clone https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit.git HiggsAnalysis/CombinedLimit
  - cd HiggsAnalysis/CombinedLimit
  - git fetch origin
  - git checkout v10.1.0
  - cd -
- Supplements
  - git clone https://github.com/cms-analysis/CombineHarvester.git CombineHarvester
- Compile
  - scramv1 b clean
  - scramv1 b –j 4

Prepare for datacards

- Go to the last section of Analysis/HiggsInvisible/EventSelection.ipynb
- Make sure your datacard folder exists.
- Fill in the histograms (TH1F) of missing mass for each process in each final state (channel).
- Compute the yield.
- Write in datacards in txt, and shapes in ROOT format, for each channel.

➢ Datacards

```
Datacard for Higgs boson invisible decay in ee final state.
imax 1 number of bins
jmax * number of processes minus 1
kmax * number of nuisance parameters
--------------------------------
shapes * ee shape_ee.root $PROCESS $PROCESS_$SYSTEMATIC
--------------------------------
bin ee
observation -1
--------------------------------
bin ee  ee  ee  ee  ee
process eeH sz  sw  szw Hincl
process -1  4   5   6   11
rate    15.545  959.805 4032.894    5020.785    15.163
--------------------------------
test    lnN 1.01    1.01    1.01    1.01    1.01
```

Statement, not necessary

Number of channels

Number of processes - 1

For syst uncertainties

The file you put the histograms, and their names.

Name of channels

Name of processes

Label of processes: signal <= 0, backgrounds > 0

Yields: should be compatible with the integral of histograms

Syst uncertainty templates

Run combine

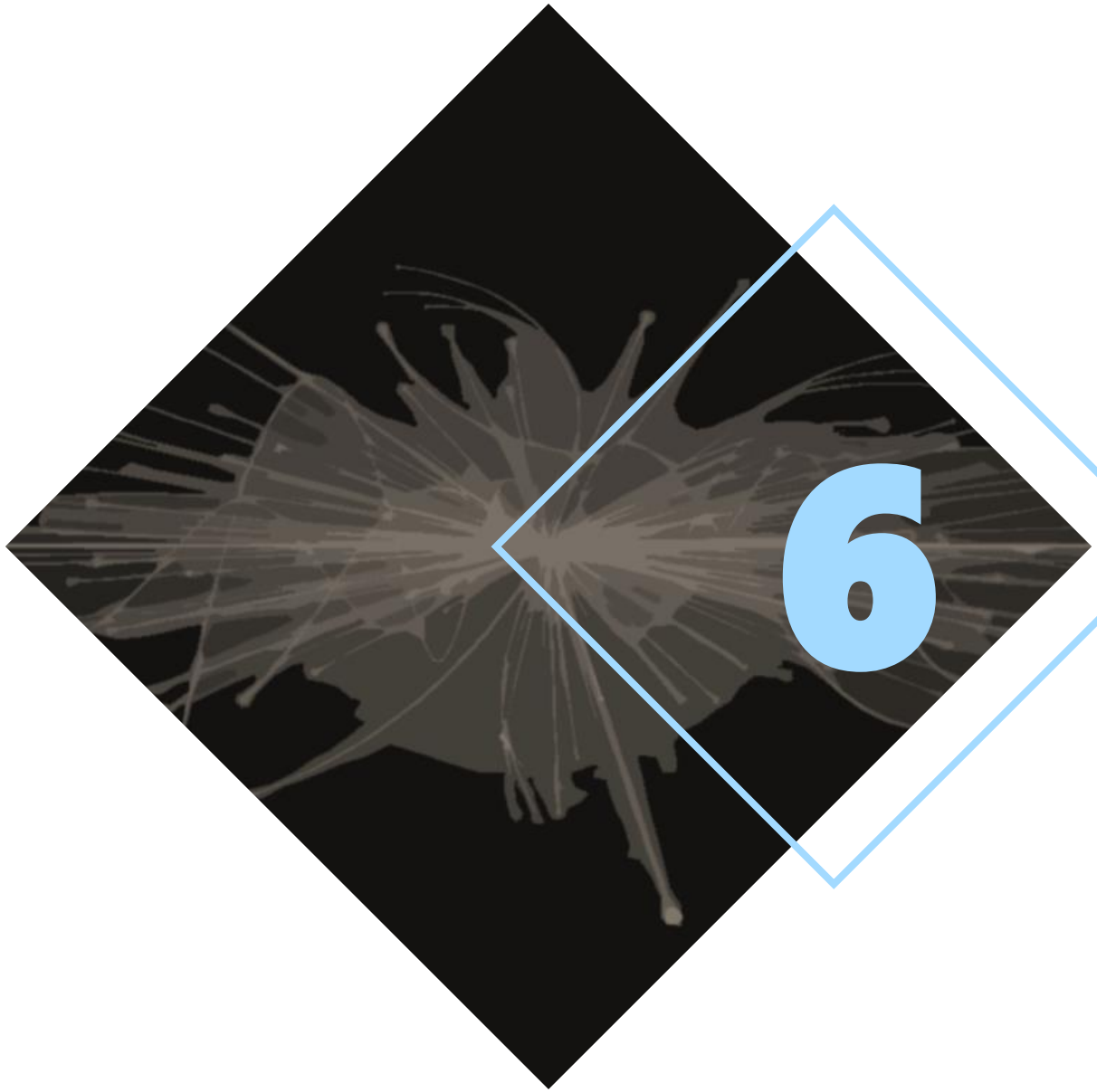
- cd /path/to/your/CMSSW/repository
- cmsenv
- cd /path/to/your/datacard/folder/
- combineCards.py dc*txt > combine.txt
  - **Combine datacards in different channels into one single datacard.**
- text2workspace.py combine.txt -o workspace.root -m 125 --X-allow-no-background
  - **Convert the datacards into a RooWorkspace, containing the complete statistical model.**
- combine -M MultiDimFit workspace.root -t -1 --algo=singles --X-rtd TMCSO_AdaptivePseudoAsimov -m 125 --expectSignal=1 --freezeParameters test
  - **You will get a fitted result with up and down uncertainty at 68% confidence level.**
- combine -M AsymptoticLimits workspace.root --run expected -m 125 --cminDefaultMinimizerStrategy=0 --cminFallback Minuit2,0:0.01 --cminFallback Minuit,0:0.001 --minosAlgo=stepping --freezeParameters test --expectSignal 1
  - **You will get the upper limit at 95% confidence level.**
- **To compute results in each channel, directly run from text2workspace.py to each datacard.**

➢ **Can be difficult to understand, but once you are able to run them, let's understand them little by little!**

Results

- After running all steps, you should be able to get the following results, assuming data of 20 ab-1.

Final state	Uncertainty	Upper limit
ee	$(1.00^{+2.32}_{-1.00}) \cdot 0.1\%$	0.454%
μμ	$(1.00^{+1.22}_{-0.84}) \cdot 0.1\%$	0.220%
qq	$(1.00^{+0.60}_{-0.60}) \cdot 0.1\%$	0.118%
All	$\left(1.00^{+0.52}_{-0.50}\right) \cdot 0.1\%$	0.965%



6 Conclusion

Conclusion

- **The physics analysis at CEPC ref-detector follows similar procedures as other experiments**
  - Produce your samples
  - Study the features of signals
  - Design the event selection criteria
  - Build the statistical model
  - Compute the result

- **There can be some specific features**
  - Unique particle identification
  - Design event selection based on detector efficiency and resolution of CEPC ref-detector.
  - ……

- **The tutorial**
  - Play with the materials provided in the tutorial!
  - In all steps: make sure you won’t write to my (liugeliang) folder, thanks! :)

Following steps

➢ For H→invisible studies

- Selection criteria are not optimized: isolation? Jet substructure? Tau reconstruction? Vertex information?
- Discriminating variable needs to be checked: is missing mass the best one? Do we want to use machine learning?
- Details to be checked: contributions from other Higgs boson decays,

➢ For physics analyses in general

- Just begins, a lot of works are needed for different topics.
  - Other Higgs boson decay channels.
  - Z→μμ studies.
- Performances of physics objects waiting to be checked under the new release.