# A Library for ML-based Fast Calorimeter Shower Simulation at Future Collider Experiments and Beyond

**Peter McKeown**[1,*], Thorsten Buss[2], Frank Gaede[3], Gregor Kasieczka[2], Anatolii Korol[3], Katja Kruger[3], Thomas Madlener[3], Piyush Raikwar[1], Anna Zaborowska[1]

[1] **CERN**

[2] University of Hamburg, UHH

[3] Deutsches Elektronen-Synchrotron, DESY

*peter.mckeown@cern.ch

**7.11.2024, ML4Jets 2024, Paris**

# Evaluating Generative Models for Fast Simulation

- Lots of generative models explored for this task
  - Largely evaluated for **single shower** performance- numerous evaluation metrics explored

Review Article

**CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation**

“Calorimeter Simulation”, generated via midjourney, 2022

See talk from Claudius on the CaloChallenge

# Evaluating Generative Models for Fast Simulation

- Lots of generative models explored for this task
  - Largely evaluated for **single shower** performance- numerous evaluation metrics explored

- However, need to evaluate in **real physics events**: many particles, overlapping showers
  - Fold in effects from **reconstruction** chain
  - Ultimately have to judge models in terms of **physics performance after reconstruction**
  - Inherently dependent on the specifics of an experiment

- Need to have a way to **interface models with full simulation** infrastructure provided by Geant4, as well as the broader **reconstruction tools** present in the **software ecosystems** used in HEP

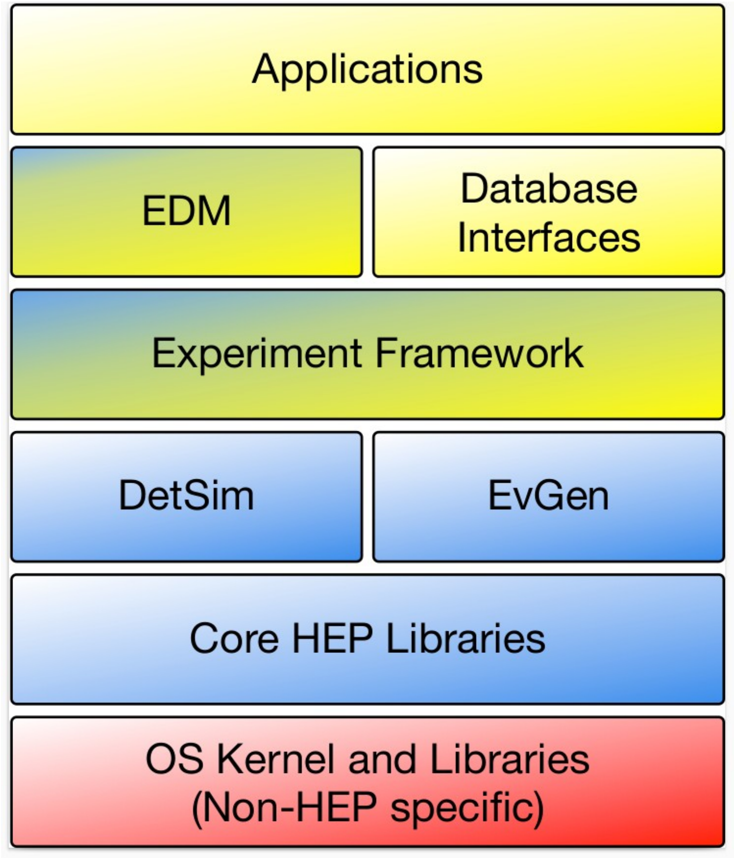
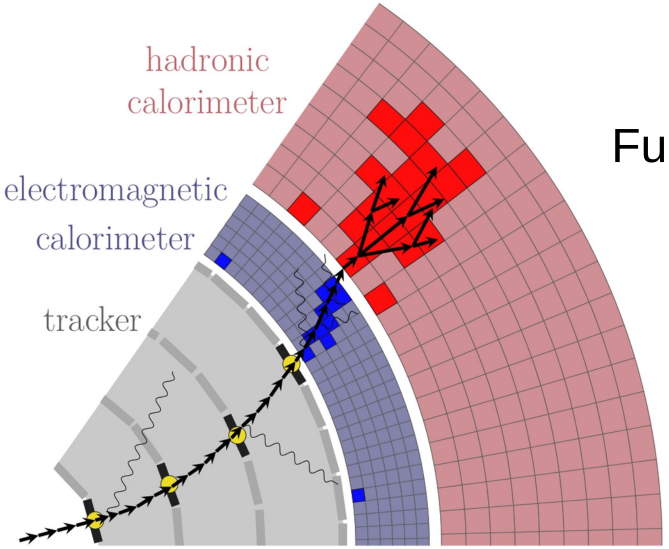



Figure credit: G. Stewart/ A. Sailer

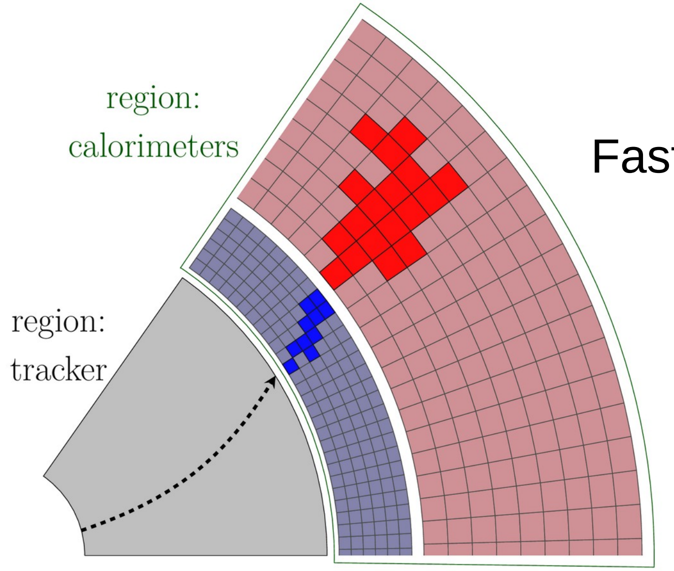
# Fast Simulation in Geant4

- '**Full simulation**' provides MC transportation of individual particles though the detector geometry

- Geant4 provides **fast simulation hooks**
  - Associated with a specific detector '**region**'
  - **Terminate full simulation**, dependent on particle type, kinematics etc.
  - Pass particle information to a **separate process** to simulate detector response- e.g. parameterization

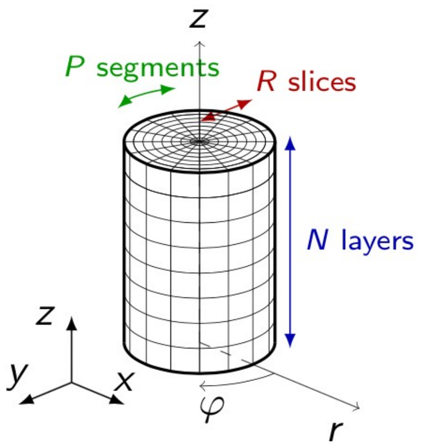



Full Sim

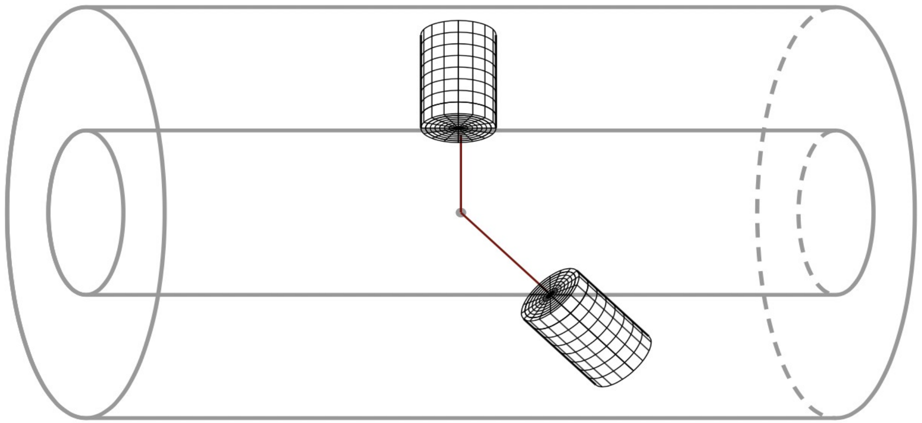



Fast Sim

# Fast Simulation in Geant4



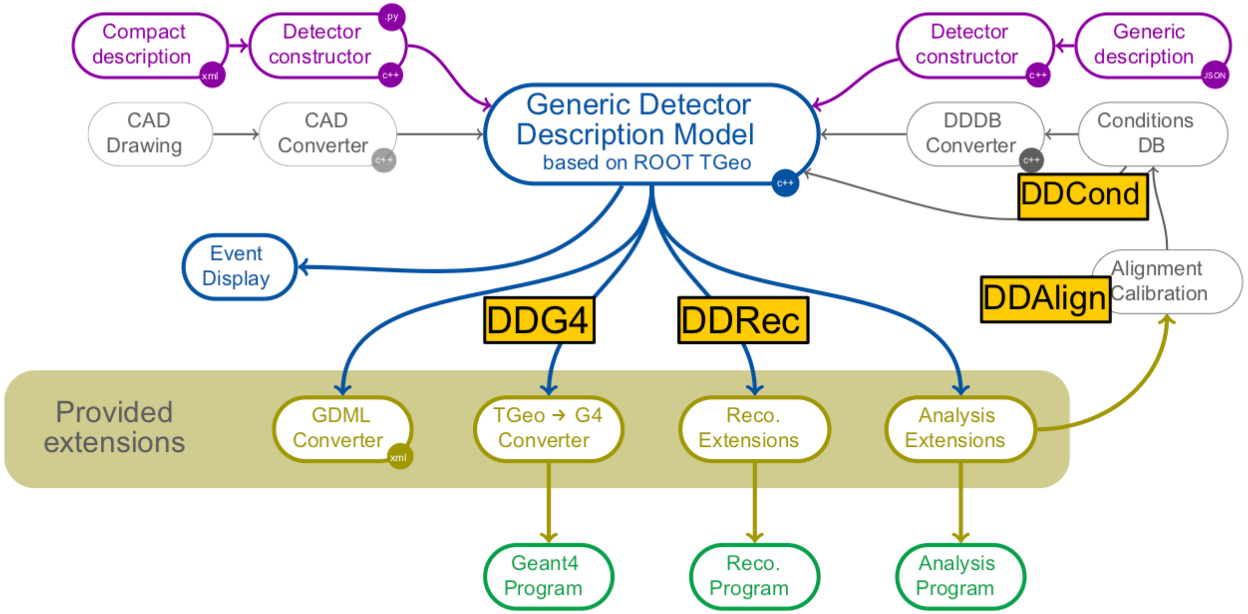

- '**Full simulation**' provides MC transportation of individual particles though the detector geometry
- Geant4 provides **fast simulation hooks**
  - Associated with a specific detector '**region**'
  - **Terminate full simulation**, dependent on particle type, kinematics etc.
  - Pass particle information to a **separate process** to simulate detector response- e.g. parameterization
- Geant4 provides the extended [Par04](Par04) example showing how to use **ML models in Geant4**
  - Energy deposits recorded in virtual scoring mesh
  - Inference libraries: ONNX Runtime, libTorch, lwtnn
  - Can also be run on GPU (currently batch size 1)
  - Provided datasets 2 and 3 for the CaloChallenge



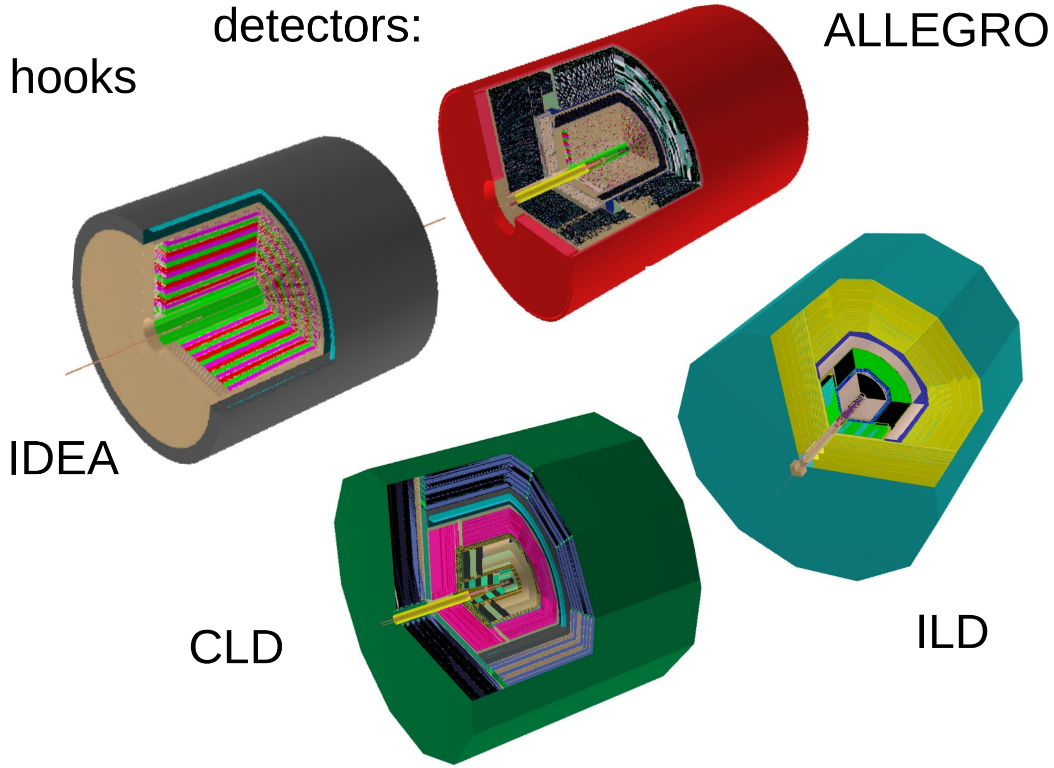
# The DD4hep Toolkit

- Provides a means for defining a **common detector geometry** that can be used for purposes of **simulation**, **reconstruction**, conditions, alignment, visualisation and analysis
- Used by **LHCb** and **CMS** experiments at **LHC**
- Part of the **Key4hep** turnkey software stack for **future colliders**, e.g. : CEPC, CLIC, EIC, FCCee, FCChh, ILC, LUXE, Muon Collider …

- **DDG4**- provides interface to Geant4, including the Fast Sim hooks




A few examples of detectors:



# The DDFastShowerML Library

- **Generic library** for running ML-based fast sim models in DD4hep: ***DDFastShowerML*** https://gitlab.desy.de/ilcsoft/ddfastshowerml
  - Uses fast sim hooks in Geant4 via DDG4
  - Follow Par04 example for interfacing with Geant4
  - Can be used with **realistic, detailed detector geometries**
- Aim for easy to use library which can **accommodate all types of ML architectures**
- **Components** of library **decoupled** as far as possible

**Trigger**

- Fast Sim trigger
  - e.g. particle type, energy, geometry

**Model**

- Model-specific implementation of ML architecture
  - e.g. BIB-AE, Flow, Diffusion model

**Inference**

- Concrete inference in C++
  - ONNX, LibTorch etc…

**Geometry**

- Concrete placement in detector geometry
  - Endcap, barrel etc…

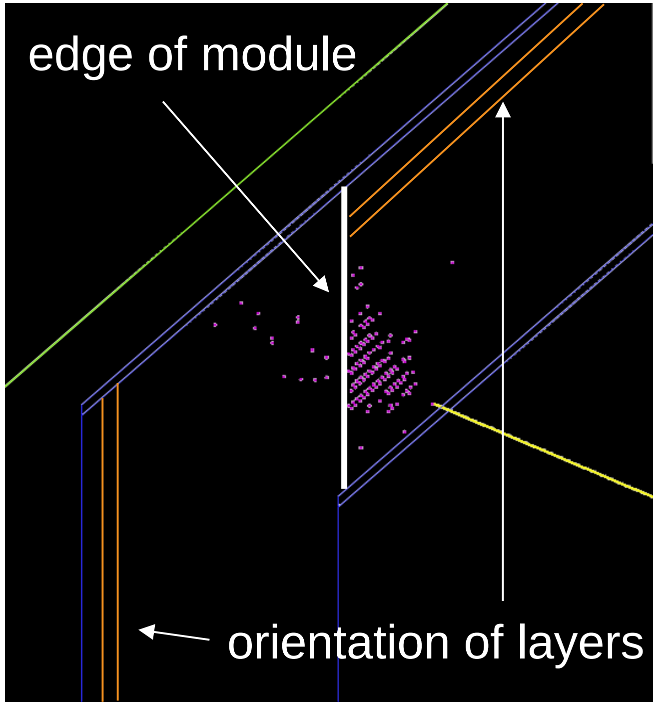
# Dealing With Realistic Geometries

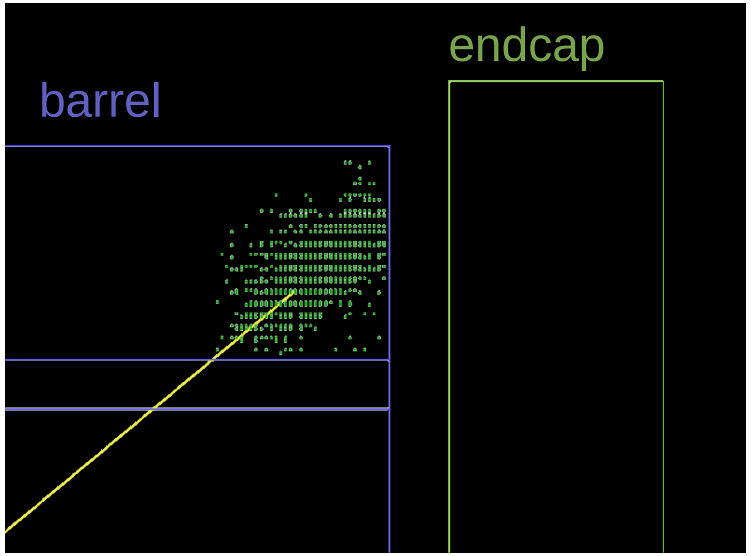
- **Physical structure** of calorimeter can change in certain regions
  - May want to run **full rather than fast sim**
  - May wish to select **different models** for **different geometrical regions** of the detector

- This can be implemented, for a given geometrical configuration, via the generic **Trigger Interface**

- E.g. for ILD excluding (i.e. running full sim):
  - **Corners** of the octagonal **barrel**
  - The **transition** between **barrel and endcap**

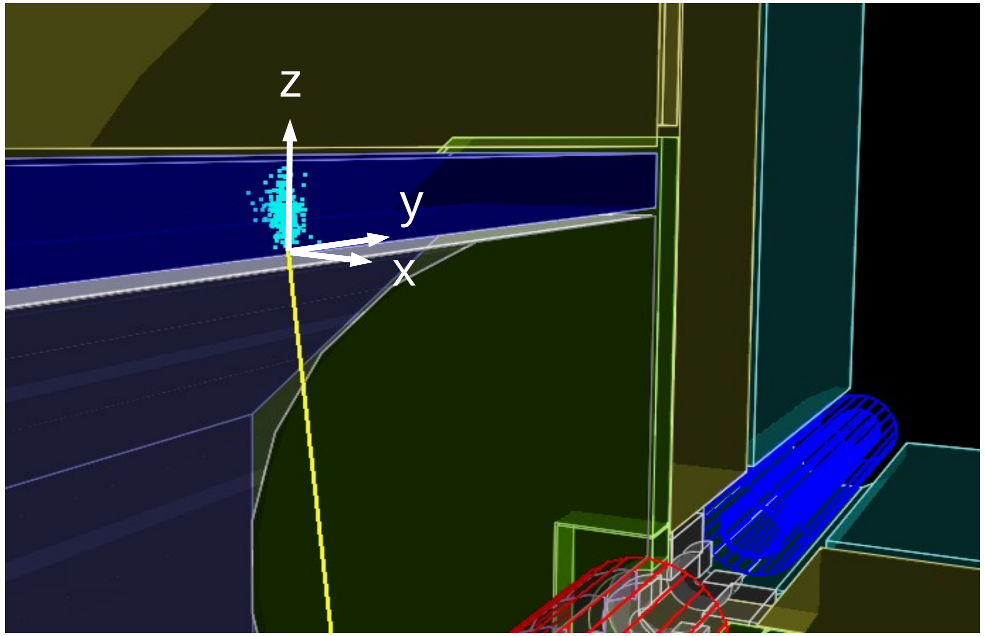
- The **geometry** placement and **trigger** necessarily have to be implemented on a per detector basis



# Conventions and How to Integrate a New Model

- Define **local coordinate system** for consistent placement
  - **Right-handed** coordinate system **centered on incident position**
  - Z-axis **perpendicular** to the calorimeter face



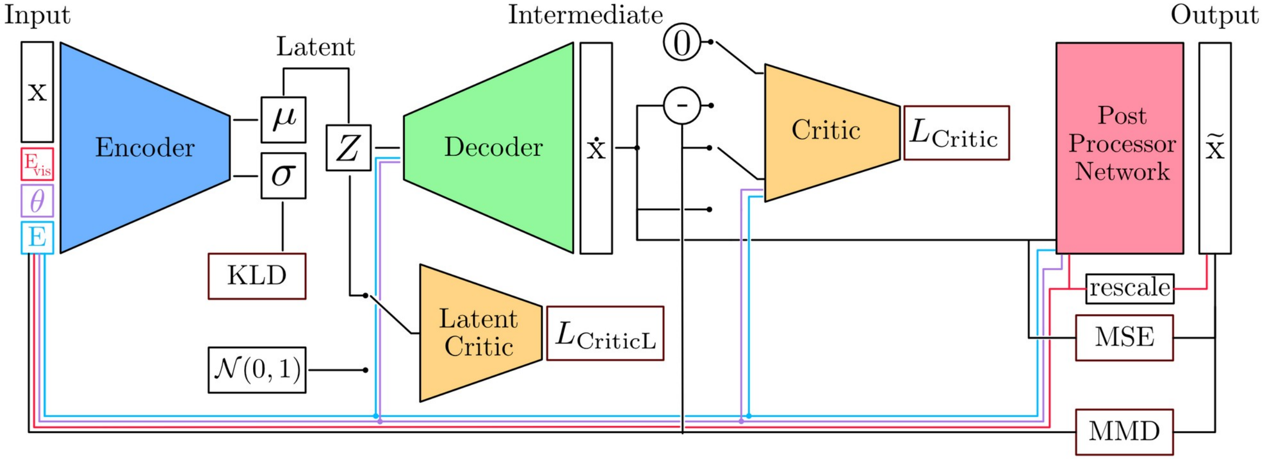

- Assuming the training dataset conforms to the conventions, adding a new model involves ***only*** adding a new class (via the **Model Interface**)
  - Ensuring the **correct input** is provided to the model
  - Ensure **correct conversion** of model output to spacepoints

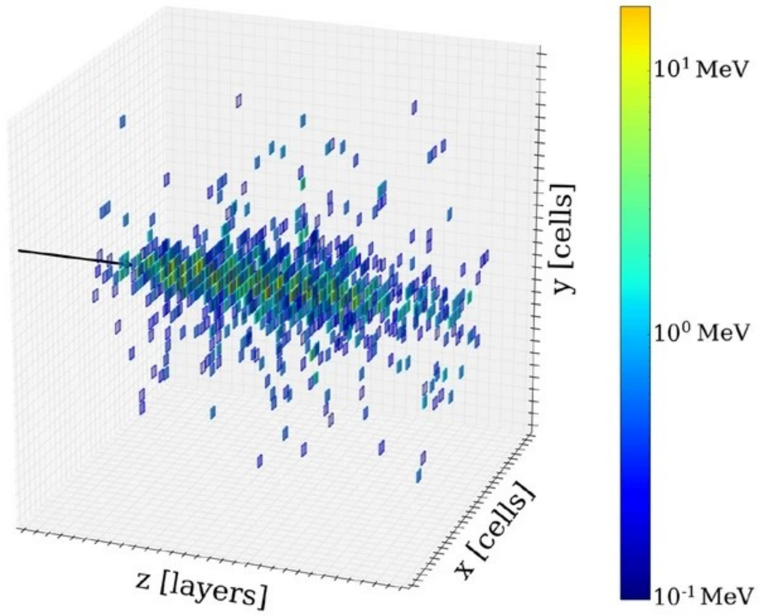
- Must be able to **convert model** into a format that can be called in **C++**
- **Inference Interface** currently supports **libTorch** and **Onnx Runtime**

- Purely for development purposes- provide functionality to **read showers** from a **library** (e.g. HDF5 file)
  - This is not intended as a solution for production!

# Example: BIB-AE

- Update at ML4Jets last year
- **Unification** of common **VAE + GAN** models, with additional application specific post processing
- Using **regular grid** (image) representation of shower




**Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed**, Buhmann et al., Comput Softw Big Sci 5, 13 (2021)

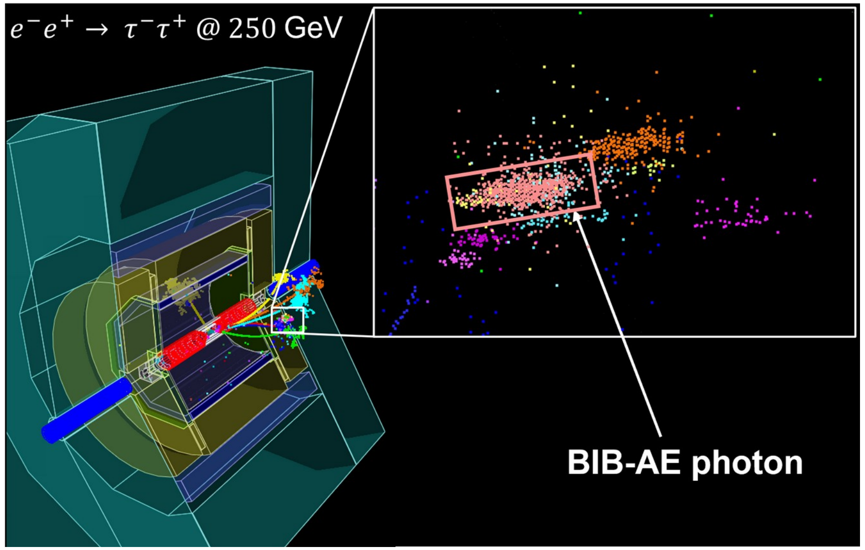
**Hadrons, Better, Faster, Stronger**
Buhmann, P.M. et al,
MLST 3 2, 025014 (2022)

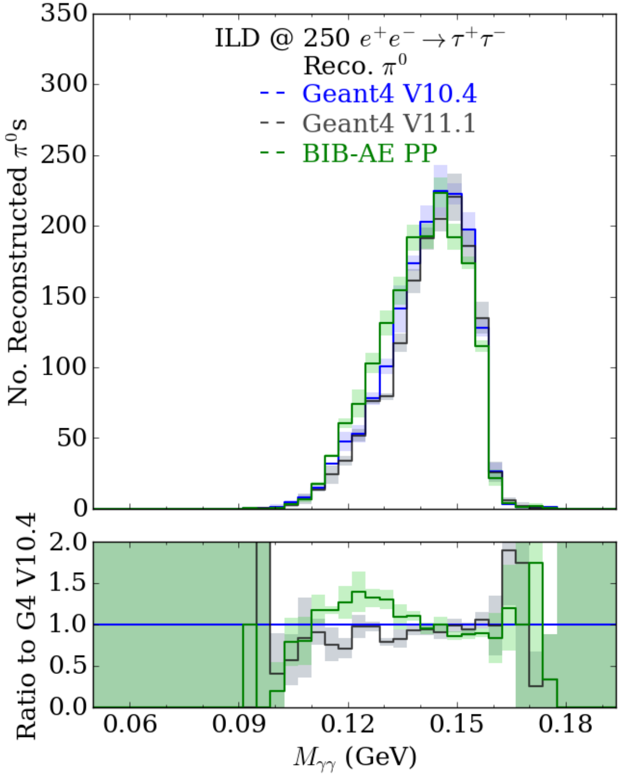
**New Angles on Fast Calorimeter Shower Simulation**, Diefenbacher, P.M. et al. MLST 4 035044 (2023)

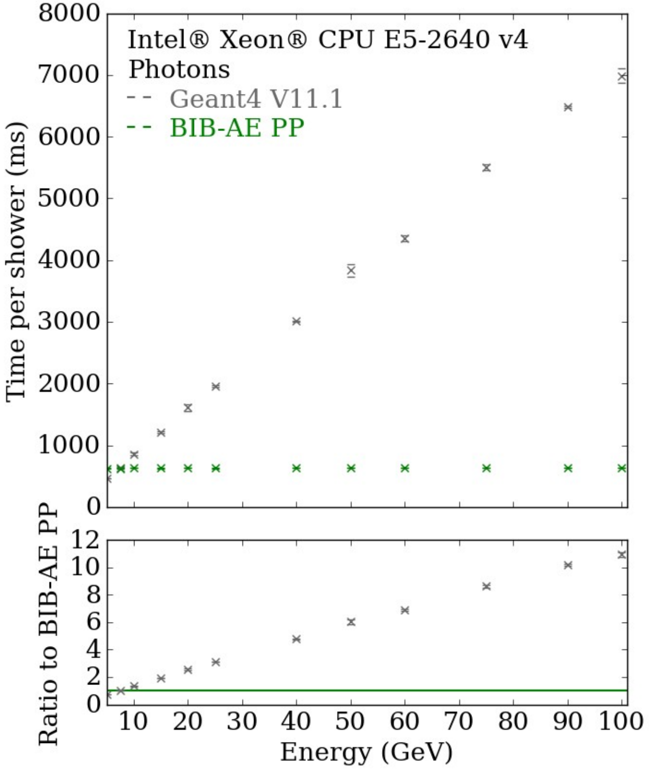
# Example: BIB-AE

- Update at ML4Jets last year
- **Unification** of common **VAE + GAN** models, with additional application specific post processing
- Using **regular grid** (image) representation of shower

- Study use for simulation of photon showers from **pi0 decays** in the process $e^{+}e^{-} \rightarrow \tau^{+}\tau^{-}$ for ILD @ 250 GeV
- Integration also allows for a **more realisitic timing** comparison to Geant4, including overheads (e.g. geometry placement etc.)



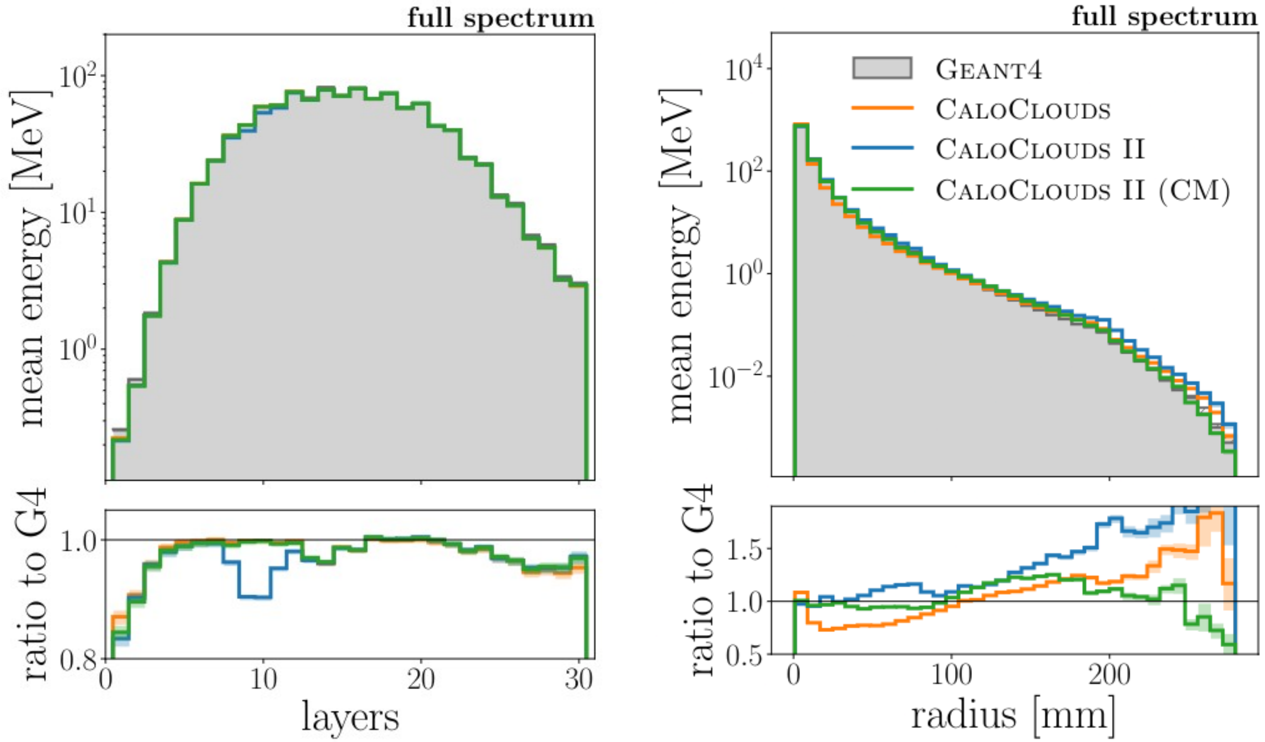




**Development and Performance of a Fast Simulation Tool for Showers in High Granularity Calorimeters based on Deep Generative Models**, P. M. (Doctoral thesis, U. Hamburg), DESY-THESIS-2024-008

# Example: CaloClouds

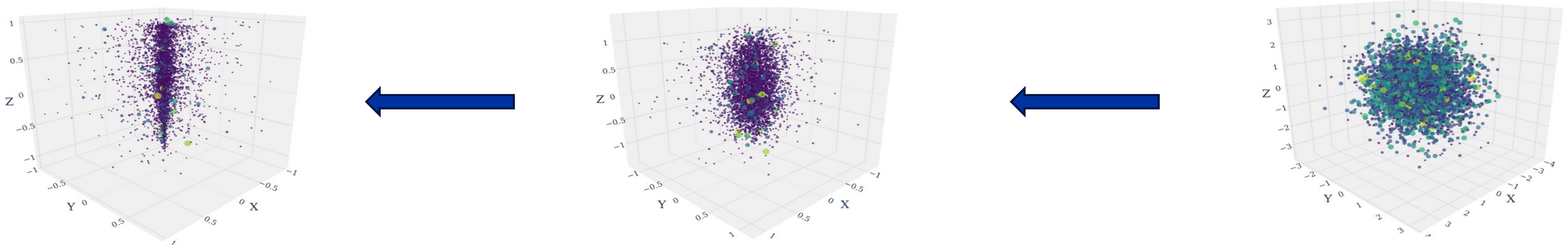
See Anatolii's upcoming talk

- **Point cloud diffusion model**
  - More efficient computation: ~**6x faster** than BIB-AE!
- Use full sim information at a **lower level than the detector readout**
  - Better handling of **irregular geometries**




**CaloClouds: Fast Geometry-Independent Highly-Granular Calorimeter Simulation**, Buhmann, P.M. et al., JINST 18 11, P11025 (2023)

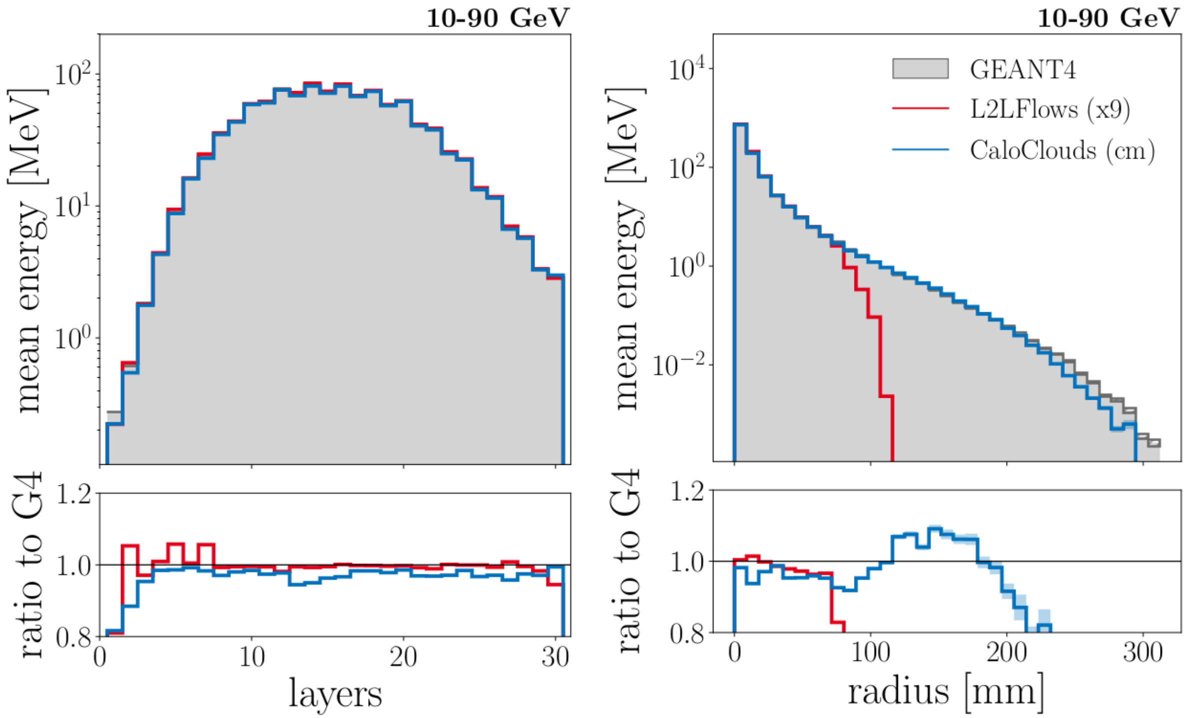
**CaloClouds II: Ultra-fast Geometry-Independent Highly-Granular Calorimeter Simulation**, Buhmann, P.M. et al., JINST 19 04, P04020 (2024)

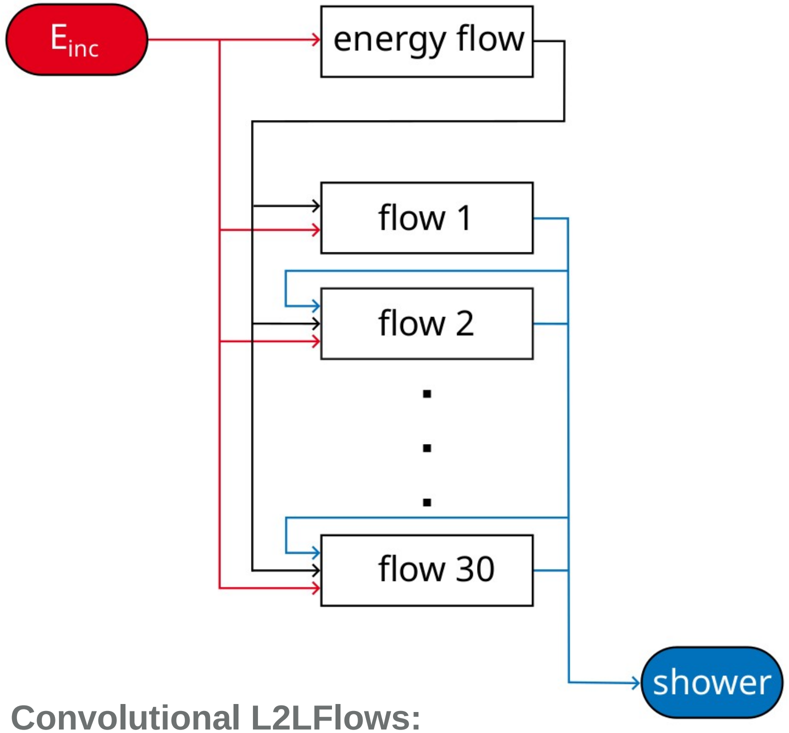


# Example: L2LFlows

See [Thorsten's talk](#) from Tuesday

- **Sequentially** produces shower shape in each layer
- Now also including
  - **Convolutions**
  - **Angular conditioning**
  - **More granular** regular grid



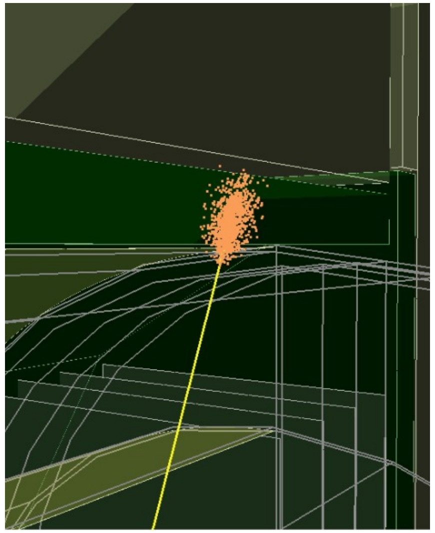

**Convolutional L2LFlows: Generating Accurate Showers in Highly Granular Calorimeters Using Convolutional Normalizing Flows**, Buss et al., JINST 19 09, P09003 (2024)

# Example: CaloDiT and Cylindrical Scoring Mesh Placements

See Piyush's upcoming talk

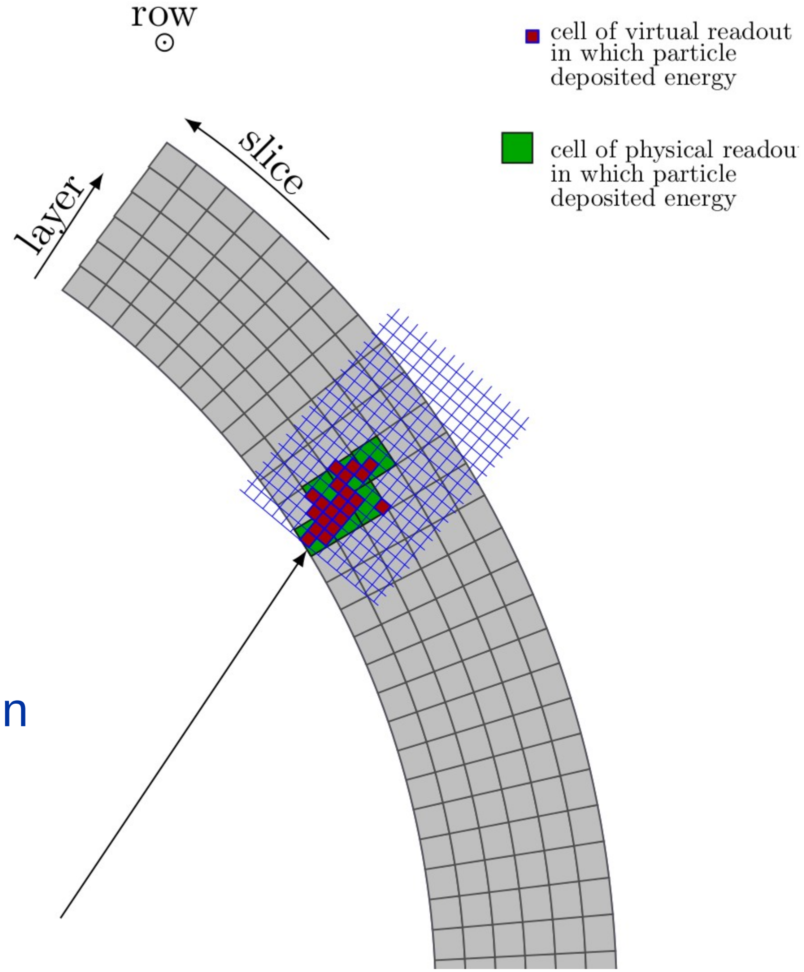
- CaloDiT **Diffusion Transformer** model
- Also explored in the context of adaption to **different detector geometries**

- Alternative fast simulation approach
  - **Score energy** deposits in mesh **not directly attached** to **detector readout**
- Validation of initial integration performed for **CLD** by C. Zhu



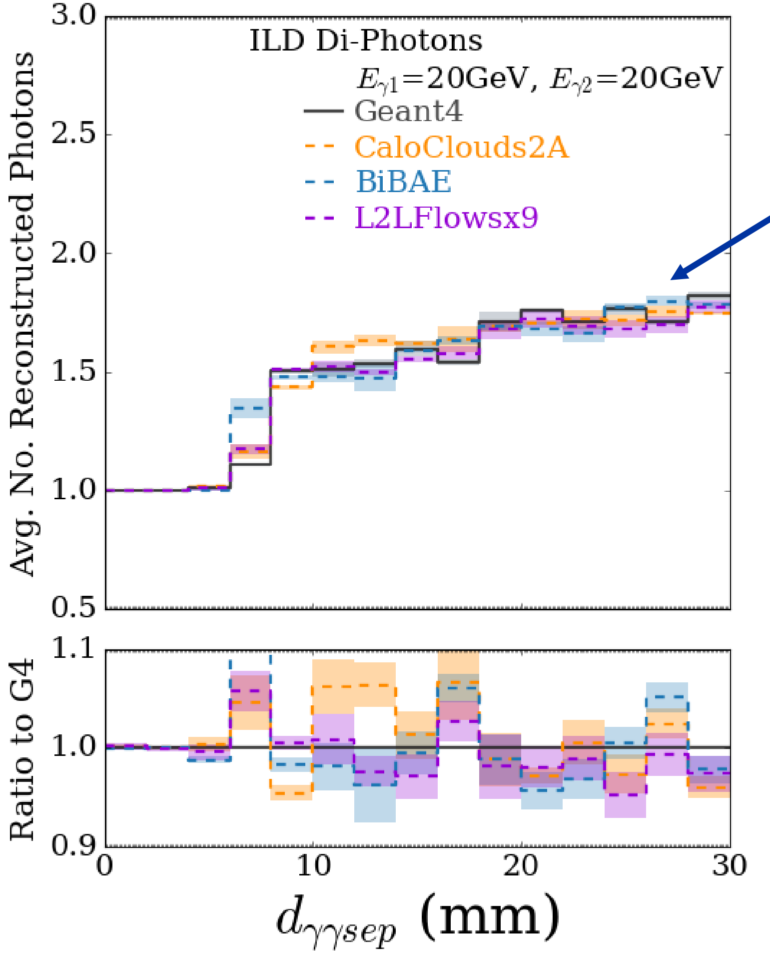
CaloDiT photon shower simulated in CLD with DDFastShowerML

Complete integration WIP

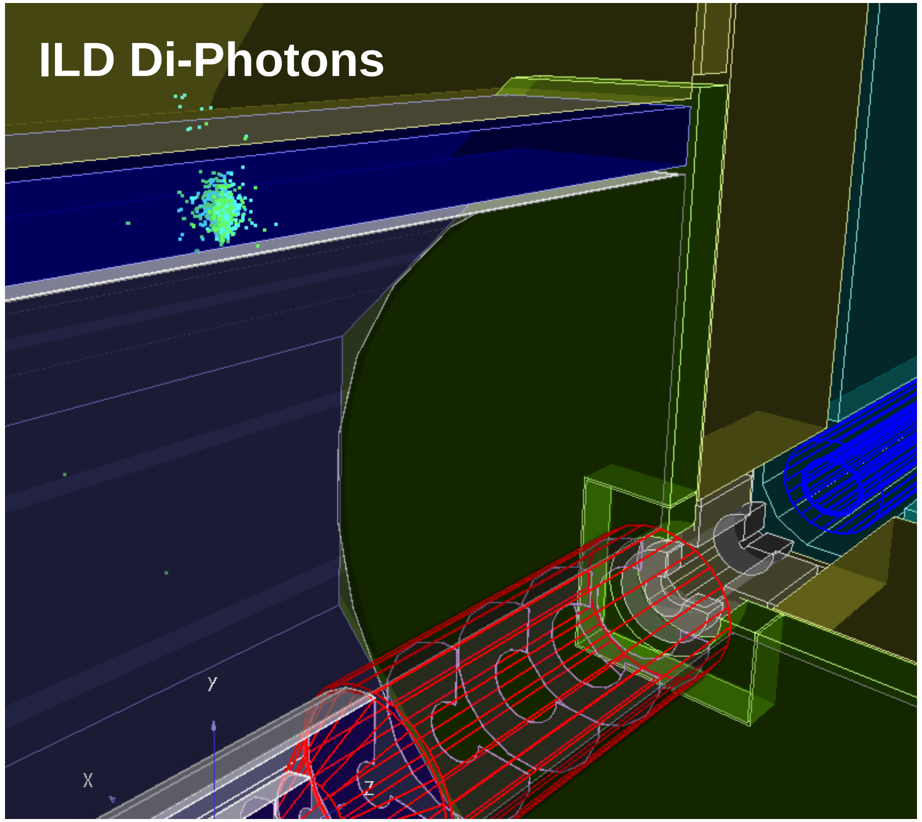


# Common Benchmarks Being Developed

- Common **Di-Photon** Benchmark (so far for CALICE Si-W calorimeters)
- Two photons orthogonal to face of ILD ECAL




Observable with **direct physical relevance** for quantifying model performance!



# Summary

***DDFastShowerML*** https://gitlab.desy.de/ilcsoft/ddfastshowerml

- Library for **incorporating ML models** with **Geant4** full simulation for realistic detector geometries
- Provides means of testing in **production-ready environment**
- Provides access to new **reconstruction** and **physics benchmarks**
- Can also **study/account** for **geometrical effects**- realisitic application
- **New models** can be **integrated** (and compared) in **as straightforward a manner as possible**

Outlook:

- Add **batching** and **GPU** support
- Add support for **additional detectors**
- Add examples for **hadronic shower** generation
- Release **datasets** for the community
- Collaborate with different experimental communities to **define new physics benchmarks**
- ...

# Backup

# DD4hep Integration: DDFastShowerML

**Algorithm 1** Pseudocode illustrating the order of operations for the core components of the *DDFastShowerML* library.

```
if Trigger.checkTrigger(track) == True then
    Kill full simulation of particle
    localDir = Geometry.getLocalDir(track)
    inputs = Model.prepareInputs(track, localDir)
    outputs = Inference.runInference(inputs)
    localSPs = Model.convertOutput(track, localDir, outputs)
    globalSPs = Geometry.localToGlobal(track, localSP)
    for (sp in globalSPs) do
        HitMaker.makeHit(sp, track)
    end for
else
    Full simulation of particle with GEANT4
end if
```