

Summary of parallel 1B

Generic Processes & Materials

Monday 25th, 14:00 – 15:30

14:00	Updates on Par04 and fast simulation <i>Hokkaido University, Room B</i>	<i>Anna Zaborowska et al.</i>	14:00 - 14:30
	Progress with using geant-val <i>Hokkaido University, Room B</i>	<i>Igor Semeniouk</i>	14:30 - 14:45
	Update on command-based scorers (TBC) <i>Hokkaido University, Room B</i>	<i>Makoto Asai</i>	14:45 - 15:05
15:00	Discussion time <i>Hokkaido University, Room B</i>		15:05 - 15:30



Updates on Par04 and fast simulation

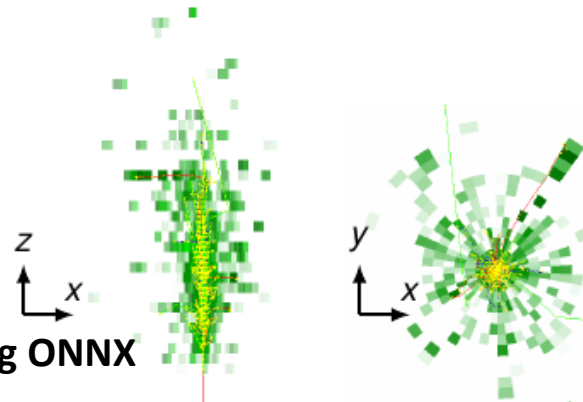
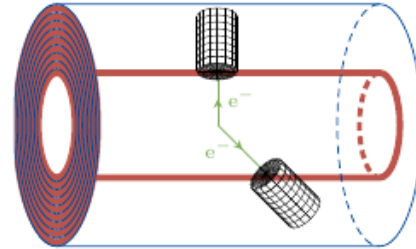
Anna Zaborowska
Dalila Salamani

Geant4 Collaboration Week, 25.09.2023

Geant4 example Par04

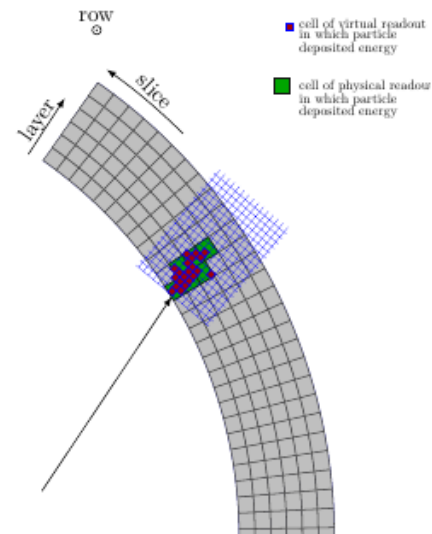
- ▶ Extended example Par04 shows how to use Machine Learning (ML) models within GEANT4.
- ▶ Distributed with a Variational Autoencoder (VAE) model of showers used in fast simulation, updated to a recent version.
- ▶ Demonstrates how to incorporate inference libraries: ONNX runtime, pyTorch (new since last release), lwttn.

→ Will provide GPU support too for running ONNX



Geant4 example Par04: realistic readout

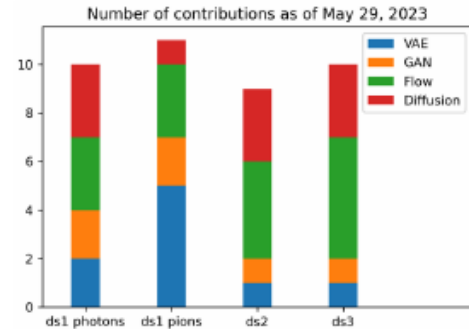
- ▶ Cylindrical scoring mesh for energy deposits is key in obtaining same showers independently of position in the detector.
- ▶ Recent additions of physical detector readout for performance benchmarking (for ML model speed-up calculation, to account the overhead time on top of inference time)
- ▶ Serves also an example of how to place back fast-simulation hits (even if they fall inside absorbers) via parallel world geometry.



Calo Challenge calochallenge.github.io

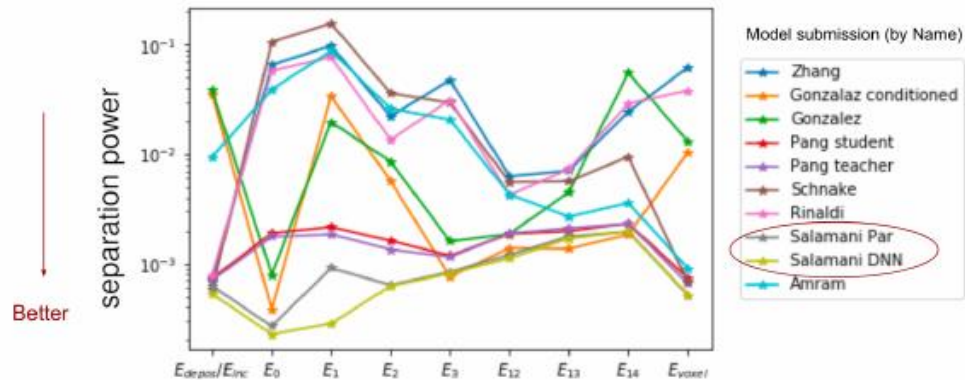
Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska

- ▶ Lots of on-going effort with ML models for fast shower simulation, but with **no place for comparison or discussions** of different methods.
- ▶ This challenge was released with three datasets with increasing dimension of input: first one comes from ATLAS open data, **the 2nd and 3rd from Par04**.
- ▶ Set of **common datasets** and **common validation metrics** allows to productively discuss different approaches.
- ▶ No winner is expected, it's a compilation of alternative solutions, often with different architectures.
- ▶ Calo Challenge Workshop held on 30-31 May 2023 in Frascati gathered many contributors and was a place for fruitful discussions, and first benchmarks.
- ▶ Final benchmarking and paper write-up is in progress.

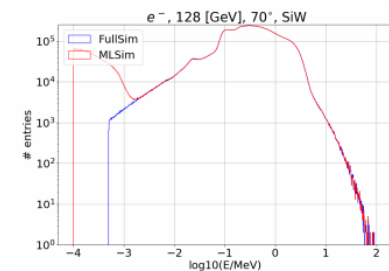


Calo Challenge: VAE model

Variational autoencoder (VAE) model developed for ATLAS pion dataset by Dalila Salamani. It seems to be very attractive, both comparing shower observables (histograms),

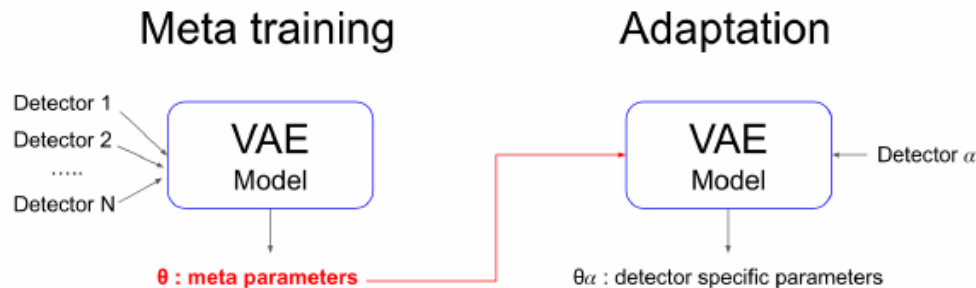


Work also to better describe individual cell energies



not yet a final result form a generative model, a interim prototype

MetaHEP



Finalised publication
MetaHEP: Meta learning for fast shower simulation of high energy physics experiments

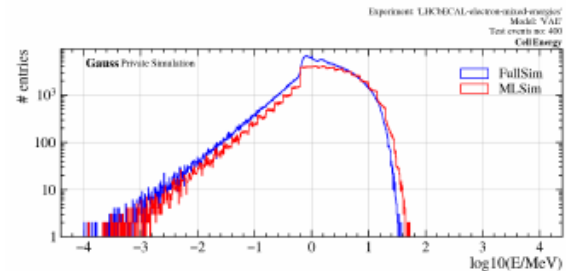
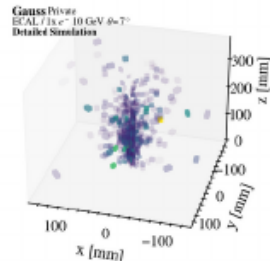
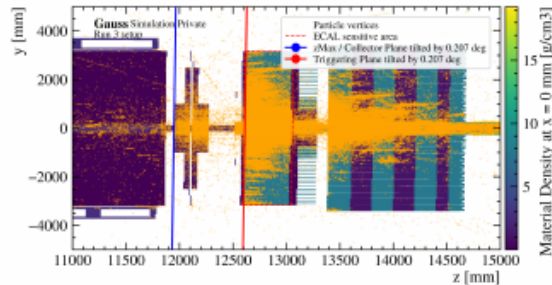
[10.1016/j.physletb.2023.138079](https://doi.org/10.1016/j.physletb.2023.138079)

- ▶ Covered at [the last Geant4 Collaboration Week](#) by Dalila
- ▶ Extensive instruction on how to use MetaHEP for any detector described at our website: g4fastsim.web.cern.ch/
- ▶ Documentation based on Par04 and the first experiment-framework implementation: in key4HEP simulation toolkit k4SimGeant4
- ▶ Inference part of Par04 was also integrated into DD4hep by DESY group (Frank Gaede et al) as DDML

Implementation within Gaussino for the LHCb

work done by Michal Mazurek

- ▶ Implementation of Par04 approach in Gaussino, use on LHCb's calorimetry;
- ▶ Many caveats discovered on the way, as expected in realistic conditions;
- ▶ Implementation ready to be used, first model (VAE) is being tested;
- ▶ Very general implementation, allows to use MetaHEP, or any models submitted to CaloChallenge;



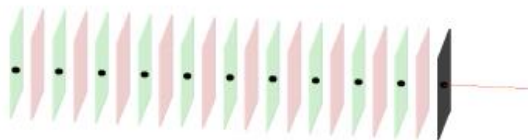
Preliminary results by Michal

→ Started for ATLAS too

In the context of classical fast simulation, still the one used in production, need for a fast track extrapolation. Idea : use a simple parallel geometry, to avoid the many boundaries of tracking one.

Track extrapolation: Par05 prototype

- ▶ We prototype in a standalone G4 application called Par05 as possibly could be a useful example of a simplified geometry (let's see how it evolves);



red = mass world, no points are created here

```
void Par05FastSimModel::do4PropagationStep(G4FieldTrack& fieldTrack){
    G4double retSafety = -1.0;
    G4double currentMinIonStep = 10.0*CLHEP::m;
    G4VPhysicalVolume * currentPhysVol = getCurrentG4VPhysicalVolume(fieldTrack);
    G4ThreeVector* direction = new G4ThreeVector();
    *direction = fieldTrack.GetMomentumDirection();
    // Must be called before calling the computeStep method
    m_SimplifiedGeoNavigator->locateGlobalPointAndSetup(fieldTrack.GetPosition(), direction);
    if(fieldTrack.GetCharge() == 0){
        /* Neutral particles: transport with simplified geometry navigator */
        // Compute the step length with the simplified geometry navigator
        G4double stepLength = m_SimplifiedGeoNavigator -> ComputeStep(fieldTrack.GetPosition(),
                                                                    fieldTrack.GetMomentumDirection(),
                                                                    currentMinIonStep);
    }
}
```

```
1 ----- EEEE ----- G4Exception-START ----- EEEE -----
2 *** G4Exception : ReadError
3   issued by : G4GDMLReadStructure::GetVolume()
4   Referenced volume 'LARMgr_LAR_Barrel_Cryostat_InnerWall_Vis0x4a60fb80' was not found!
5   *** Fatal Exception *** core dump ***
6   **** Track information is not available at this moment
7   **** Step information is not available at this moment
8 ----- EEEE ----- G4Exception-END ----- EEEE -----
```

Solution could not be ported back to ATLAS because a GDML problem (a feature indeed) : simply resolved during session ☺

Summary and Outlook

On-going work on finalisation of many of the topics:

- ▶ Par04 changes must be moved from our development repo to geant-dev;
- ▶ Many topics on the ML models: Calo Challenge finalisation, next evolution, ...
- ▶ Exciting results from LHCb who tests our (and in future Calo Challenge community!) models;

More long-term plans:

- ▶ Implementation of MetaHEP for other experiments than LHCb;
- ▶ Development of ML models based on transformers, applicable within above-mentioned implementations to frameworks;
- ▶ Helping to decouple the ATLAS classical parameterisation to act as a standalone package;

By Igor Semeniouk, LLR

Generic Processes & Materials examples in geant-val

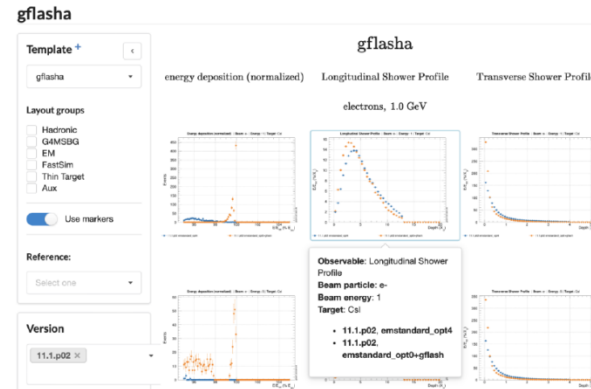
25/09/2023 Parallel session Generic Processes & Materials

28th Geant4 Collaboration Meeting

Hokkaido University, Sapporo (Japan)

Processes & Materials examples

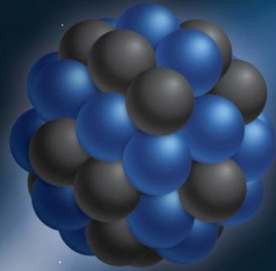
- Gflash
 - First preliminary histograms in geant-val
 - The results very promising
 - Working on final configuration
- Biasing
 - Work in progress
 - Adding histograms to the examples
- ReverseMC
 - Discussion started



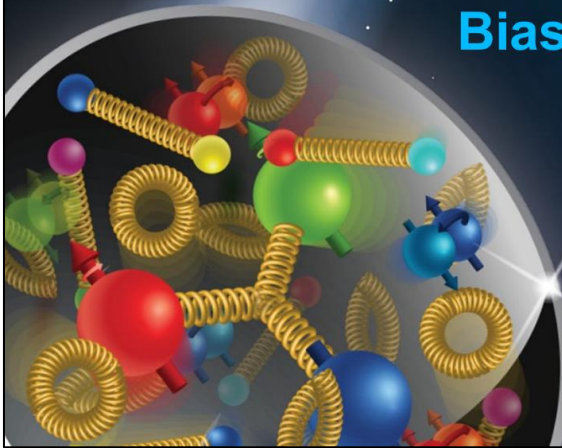
Status of ExGFlasha example

- Not usual use case: the results (fast monte carlo) has to be compared with full simulation. Noy only for gflash but other GSM examples.
- The first results was published
 - Not a final configuration
 - The parameter calorimeter length in X_0 missing
 - The histograms limits and energy points must be corrected
 - Long jobs for high energies need to be splitted
- The some extension of geant-val may be useful
 - The ability to display text/tables not only histograms
 - More selectors like material, energy, etc to choose what will be displayed
- Long publication cycle
- Missing of test site

→ Plan to visit CERN to clarify with Dmitri & Lorenzo



Biassing and Scoring



Makoto Asai (JLab/SCT)
asai@jlab.org

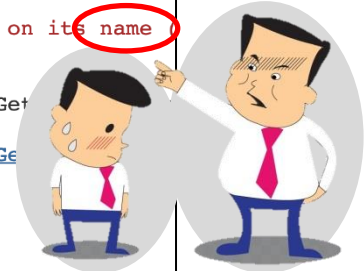
GFlashHitMaker::make() – version 11.0.p02

```
95
96 G4VPhysicalVolume* pCurrentVolume = fTouchableHandle()->GetVolume();
97 G4VSensitiveDetector* pSensitive;
98 if( pCurrentVolume != 0 )
99 {
100     pSensitive = pCurrentVolume->GetLogicalVolume()->GetSensitiveDetector();
101     G4VFlashSensitiveDetector * gflashSensitive =
102         dynamic_cast<G4VFlashSensitiveDetector * > (pSensitive);
103     if( gflashSensitive )
104     {
105         gflashSensitive->Hit(&theSpot); ← GFlashEnergySpot object
106     }
107     else if (( pSensitive ) &&
108             ( pCurrentVolume->GetLogicalVolume()->GetFastSimulationManager() ) )
109         // Using gflash without implementing the
110         // gflashSensitive detector interface -> not allowed!
111
112     {
113         G4cerr << "ERROR - GFlashHitMaker::make()" << G4endl
114             << "    It is required to implement the " << G4endl
115             << "    G4VFlashSensitiveDetector interface in " << G4endl
116             << "    addition to the usual SensitiveDetector class."
117             << G4endl;
118         G4Exception("GFlashHitMaker::make()", "InvalidSetup", FatalException,
119             "G4VFlashSensitiveDetector interface not implemented.");
120     }
121 }
```

Original GFlash uses a specialized SD, but became unnecessary (G4Step) and Hit Maker should be updated → compliant with scoring and Vis

GFlashHitMaker::make() : parallel world for scoring

```
00
69 void GFlashHitMaker::make(GFlashEnergySpot * aSpot, const G4FastTrack * aT)
70 {
71     // Locate the spot
72     if (!fNaviSetup)
73     {
74         // Choose the world volume that contains the sensitive detector based on its name
75         G4VPhysicalVolume* worldWithSD = nullptr;
76         if(fWorldWithSdName.empty()) {
77             worldWithSD = G4TransportationManager::GetTransportationManager()->Get
78         } else {
79             worldWithSD = G4TransportationManager::GetTransportationManager()->Ge
80         }
81         fpNavigator->SetWorldVolume(worldWithSD);
82         fpNavigator->
83             LocateGlobalPointAndUpdateTouchable(aSpot->GetPosition(),
84                                                 fTouchableHandle(), false);
85         fNaviSetup = true;
86     }
87     else
88     {
89         fpNavigator->
90             LocateGlobalPointAndUpdateTouchable(aSpot->GetPosition(),
91                                                 fTouchableHandle());
92     }
93 }
```



Makoto reminded me about a “feature” to be fixed in old GFlash code