Fast Simulation of LHCb Time Sensor Using Machine Learning

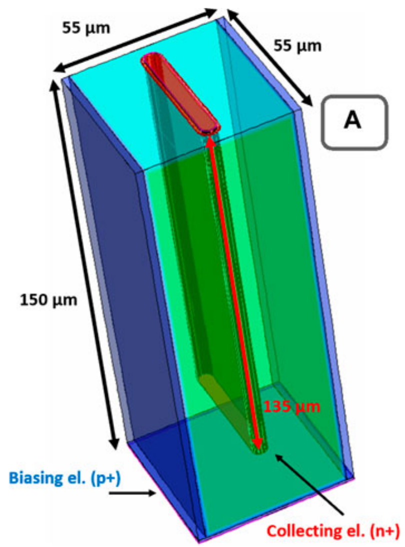
Zhihua Liang, Angelo Loi, Adriano Lai, Andrea Contu, Rolf Oldeman, Andrea Lampis
INFN Cagliari

ICSC and Spoke 2 - Where are we now?

11-12-2024

Introduction & Motivation

- **TimeSPOT**, a 3D trench silicon pixel sensor, is key for the VELO U2 upgrade, promising improved timing and radiation hardness. Efficient simulation is crucial for design optimization.
- Current Geant4/TCoDe simulations are computationally expensive, limiting large-scale design studies for VELO U2.
- **Goal:** Develop a fast, accurate ML-based TimeSPOT simulation to accelerate VELO U2 design and analysis, enabling:
  - **Rapid design evaluation.**
  - **Robust performance projections.**



3D trench pixel geometry (Borgato et al., 2024)

Technical Approach: Data & Features

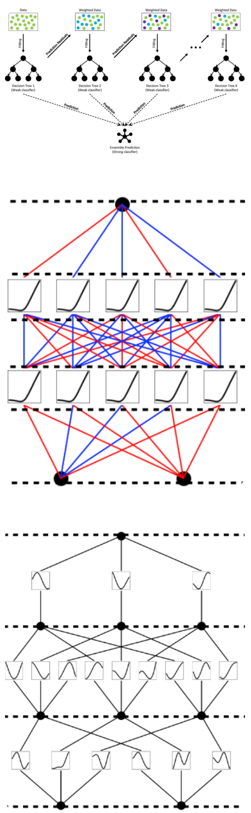
- **Dataset:** 400,000 simulated TimeSPOT events (Geant4/TCoDe).
- **Key Features (Input to ML Model):**
  - Particle impact & exit points (x, y, z).
  - Momentum (px, py, pz).
  - Kinetic & total energy.
  - Origin vertex (x, y, z).
  - Time.
- **Data Format:** HDF5 (efficient data handling).

TCODE
TimeSPOT Code for Detector simulation

GEANT4
A SIMULATION TOOLKIT

Technical Approach: ML Models & Training

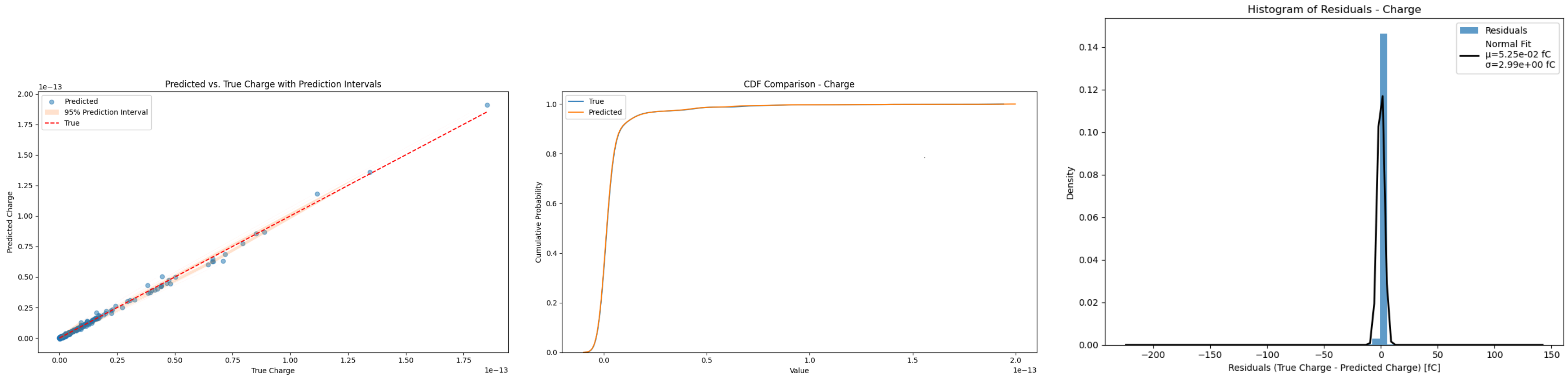
- **Models Explored:** Evaluated XGBoost, Multi-Layer Perceptron (MLP), and Edge-Activated Adaptive Function Network (EAAFN) architectures.
- **Selected Model:** MLP offered the optimal balance of predictive accuracy and inference speed.
- **Training Methodology:**
  - **Dataset Splitting:** 97% training, 3% validation.
  - **Optimization:** Adam optimizer with cosine annealing learning rate scheduler.
  - **Batch Size:** 1024.
- **Gauss Integration:** Trained model converted from PyTorch to ONNX and then to DAT format using ROOT TMVA's SOFIE module for optimized inference within Gauss.



Architecture comparisons: BDT (top), MLP (middle), EAAFN (bottom)

Results: Charge Prediction

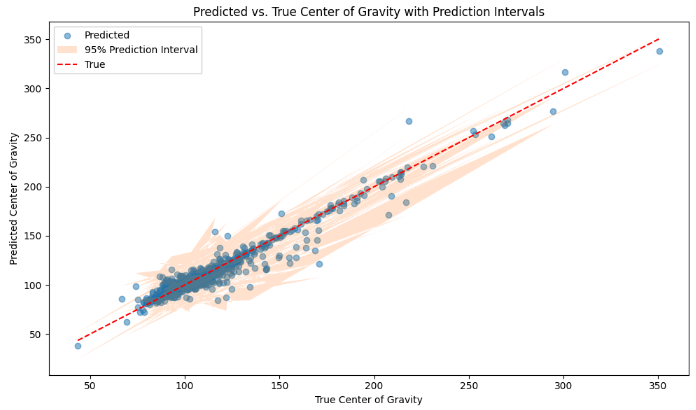
- **Performance Metrics:**
  - R^2: > 0.99 (Excellent agreement)
  - Relative MAE: ~0.0125 (Low average error)
  - Relative MSE: ~0.0056 (Small variance in errors)
- These metrics demonstrate the model's high accuracy in predicting the collected charge.

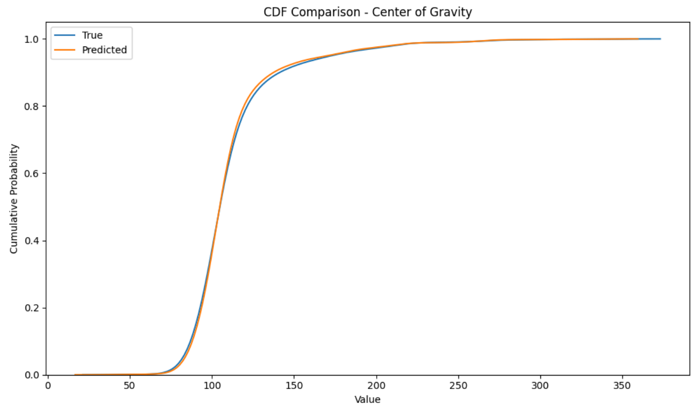


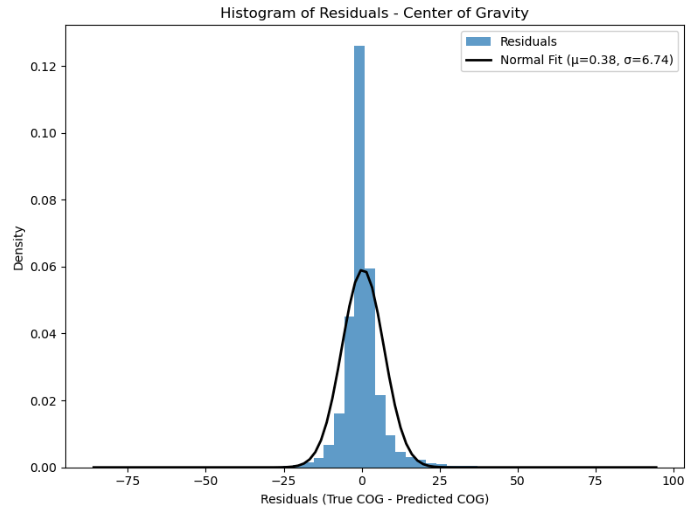
Left: Prediction intervals. Center: CDF comparison (predicted vs. true charge). Right: Residual distribution.

Results: Center of Gravity (Time) Prediction

- **Performance Metrics:**
  - R^2 Score: > 0.93 (Excellent agreement with true values)
  - Relative MAE: ~0.0357 (Low average error)
  - Relative MSE: ~0.0063 (Small variance in errors)



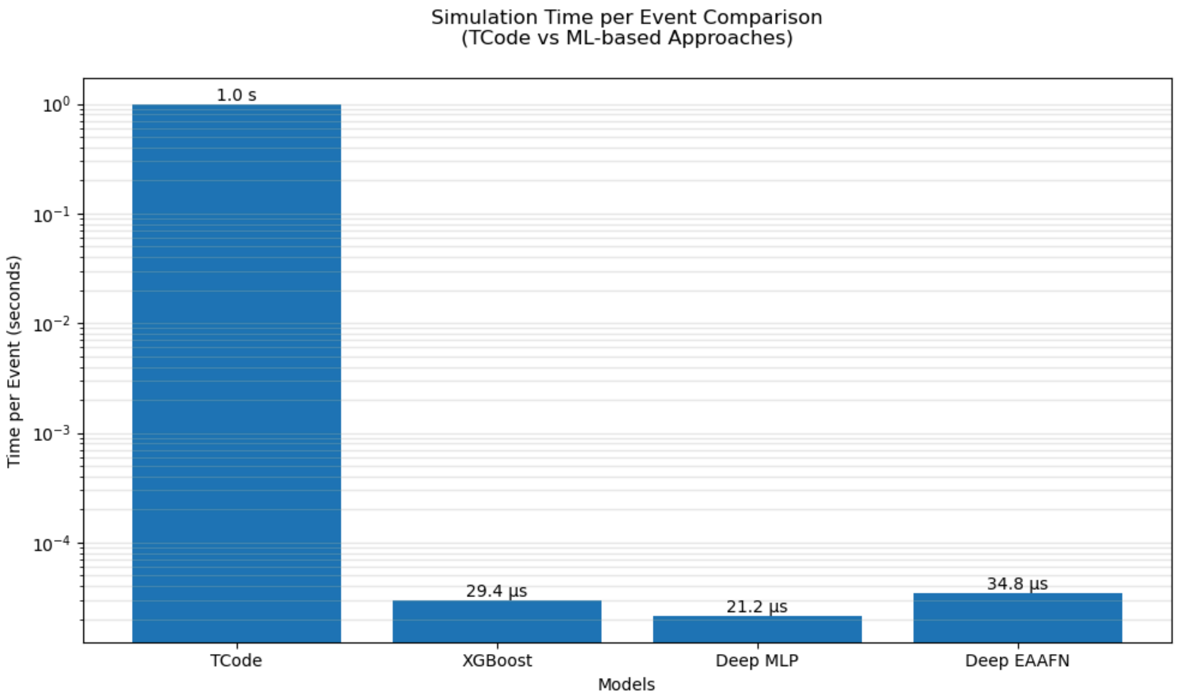




Left: Prediction intervals (model uncertainty). Center: CDF comparison (predicted vs. true). Right: Residual distribution.

Performance Benchmark: Simulation Speedup

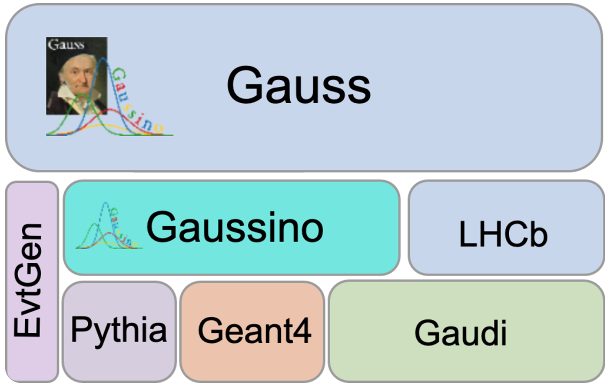
- **Speedup:** The ML-based simulation achieves a remarkable speedup of 10^4 - 10^5 compared to the traditional TCoDe simulation.



Simulation time per event: TCoDe (≈1s) vs. ML models. The ML models demonstrate a dramatic reduction in simulation time, enabling high-throughput studies.

Implementation Overview

- Developed the fast timeSPOT sensor simulation under `Sim/GaussFastSim` in the Gauss package
- Main code files:
  - `Sim/GaussFastSim/src/FastsimTimeSensorModel.cpp`
    - Implementation of the `FastsimTimeSensorModel` class
  - `Sim/GaussFastSim/src/FastsimTimeSensorModel.h`
    - Header file for the `FastsimTimeSensorModel` class
  - `Sim/GaussFastSim/src/400k_mlp_model.h`
    - TMVA SOFIE generated model header
  - `Sim/GaussFastSim/src/400k_mlp_model.dat`
    - Model weights and biases data file
- Updated `Sim/GaussFastSim/CMakeLists.txt`
  - Specified dependencies and libraries for compilation



Integration into Gauss: FastsimTimeSensorModel

- **Seamless Integration:** The `FastsimTimeSensorModel` class integrates seamlessly within the Gauss framework by inheriting from `G4VFastSimulationModel`.
- **Key Responsibilities:**
  - **Model Applicability:** Determines if the fast simulation should be applied based on particle type, energy, or other relevant criteria.
  - **Triggering:** Selects specific particle tracks for fast simulation, optimizing performance by applying the model only when necessary.
  - **Response Generation:** Extracts relevant particle properties, performs inference using the loaded ML model, and generates the simulated TimeSPOT sensor response (charge, COG).

```
// FastsimTimeSensorModel.cpp (simplified example)
void FastsimTimeSensorModel::DoIt(const G4FastTrack& aFastTrack, G4FastStep& aFastStep) {
  // ... (Extract particle properties)

  // Perform inference using the loaded ML Model
```

Integration into Gauss: Model and Workflow

- **Model Conversion and Loading:** The trained MLP model is converted from PyTorch to optimized C++ using TMVA SOFIE, ensuring efficient inference within the Gauss framework. Model weights and biases are stored separately and loaded during initialization.
- **Fast Simulation Workflow:**

  i. Particle enters the sensitive region of the TimeSPOT detector.

  ii. `FastsimTimeSensorModel` evaluates applicability and trigger conditions.

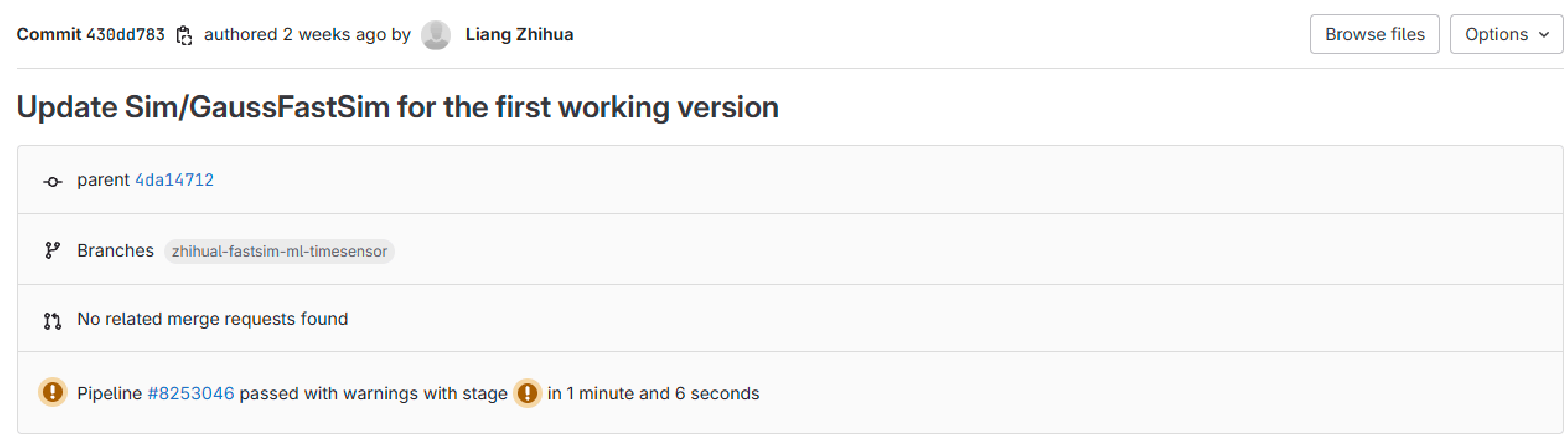
  iii. If triggered, relevant particle properties are extracted.

  iv. The pre-trained MLP model predicts the sensor response (charge, COG).

  v. The simulated response is used in downstream processing, replacing the computationally expensive Geant4 simulation.

Code Integration & Validation

- Successfully integrated the fast simulation code into the Gauss framework.
- Code submitted to the official Gauss repository (branch: *zhihual-fastsim-ml-timesensor*).
- Successful compilation confirmed. Validation tests are planned for the next phase of development.



Impact on VELO U2 Design & Studies

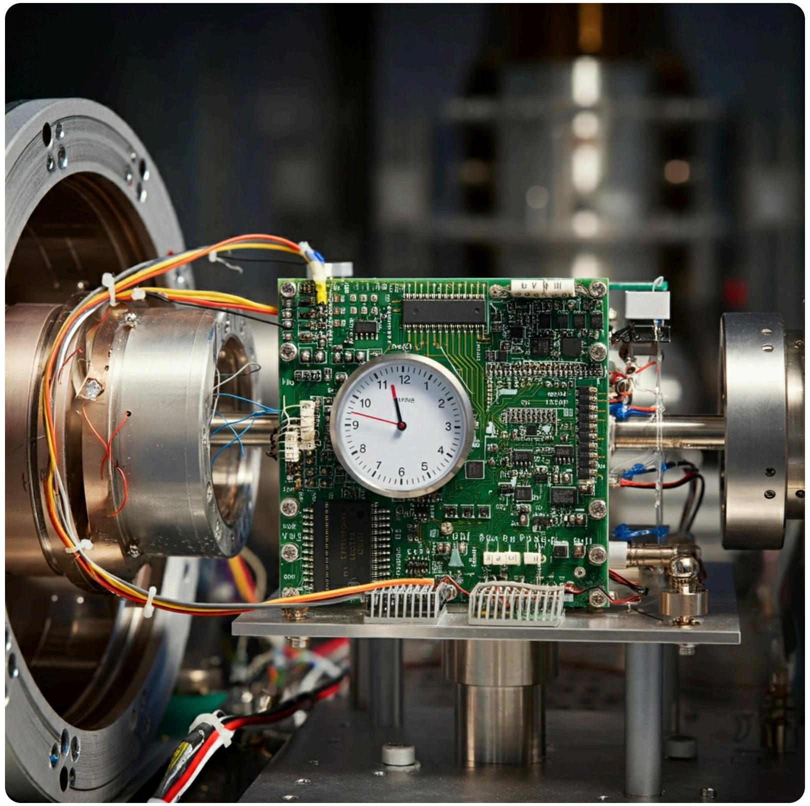
- **Current Capabilities:** The fast simulation, using an MLP trained on particle parameters, significantly accelerates studies related to particle interaction within the TimeSPOT sensor. This enables rapid optimization of:
  - **Sensor Placement:** Maximize geometric acceptance and minimize material budget for improved track reconstruction efficiency.
- **Future Potential (with enhanced models):** Future development of more sophisticated ML models (e.g., diffusion models) will enable direct exploration of sensor characteristics and readout architectures:
  - **Material Studies:** Evaluate different sensor materials (e.g., silicon, diamond) and support structures, considering trade-offs between radiation hardness, performance, and cost.
  - **Readout Optimization:** Investigate the impact of different readout architectures and technologies on timing performance and data rates.

Conclusion & Future Directions

- **Achievements:**
  - Developed a fast and accurate ML-based simulation for the LHCb TimeSPOT sensor, demonstrating a speedup of 10^4-10^5 compared to traditional methods.
  - Successfully integrated the fast simulation model into the Gauss framework using TMVA SOFIE.
  - Enabled rapid exploration of TimeSPOT sensor placement within VELO U2 design studies.
- **Next Steps & Community Input:**
  - **Advanced ML Models:** Explore more sophisticated models (e.g., diffusion models) to enable direct simulation of sensor material properties and readout architectures.
  - **Validation and Benchmarking:** Perform comprehensive validation against full simulations and, if possible, real data. Benchmark the impact of the fast simulation on physics performance.
  - **Community Engagement:** I'm eager to receive feedback and input from the LHCb community on the future direction of this work. What are the most critical needs and priorities for fast simulation within VELO U2 design and analysis?

Q&A

Thank you for your attention!



An AI-generated artistic rendering of the TimeSPOT sensor in the LHCb detector.