



LIETUVOS MOKSLO TARYBOS FINANSUOTO IMPRESS-U PROGRAMOS PROJEKTO

„IRIS-HEP: MOKSLO PROGRAMINĖS ĮRANGOS PLĖTIMAS DALELIŲ FIZIKOS REIKMĖMS UKRAINAI, LENKIJAI IR LIETUVA“

**(Nr. S-IMPRESSU-24-4, reg. data 2024-06-26 d.,
Pakeista Nr. S-IMPRESSU-26-2, reg. Data 2026-03-19 d.)**

2026 M. STUDENTŲ STAŽUOČIŲ PATVIRTINTI PROJEKTAI

Projekto vykdytojas: Vilniaus Universitetas

Projekto vadovas: dr. Andrius Juodagalvis

2026 Vilnius

IRIS-HEP Fellow: Paulius Balčiūnas



Fellowship dates: Feb – Apr, 2026

Home Institution: Vilnius University Faculty of Mathematics and Informatics

Project: Lossy Data Compression for Simulated CMS Pile-up Pixel Subdetector Datasets

CMS datasets pose significant challenges for data storage and processing due to their large size and complexity. The primary goal of this project is to investigate the possibilities of applying Vector Quantization Variational Autoencoder (VQ-VAE) model for CMS pileup dataset lossy compression. VQ-VAE model aims to encode the data into a compact latent representation, which can later be reconstructed using VQ-VAE learned codebooks with minimal information loss

More information: [My project proposal](#)

Mentors:

- Valdas Rapševičius (Vilnius University)
- Tomas Raila (Vilnius University)

Presentations and Publications

- 5 Mar 2026 - "[Lossy Compression algorithm for simulated datasets](#)", Paulius Balčiūnas, Lithuanians@CERN Recording: [Lossy Compression algorithm for simulated datasets](#)

Current Status

A placeholder for status updates

Contact me:

✉ [Send me an email](#)

🔗 [Visit my GitHub Repo](#)

🌐 [Visit my LinkedIn Profile](#)



This project is supported by the National Science Foundation under Cooperative Agreements OAC-1836650 and PHY-2323298. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Applicant: Ernesta Idzelytė
Mentors: Angira Rastogi, Simone Pagan Griso, Sergo Jindariani
Project Duration: 12 Weeks
Proposed Start Date: 1st July 2026

IRIS-HEP Fellows Program Project Proposal

Charged-Particles Reconstruction at Muon Colliders: 3D Sensor Charge Digitization Model

Abstract

Muon colliders are promising candidates for future experiments due to their ability to reach multi-TeV collision energies with more compact and cheaper designs. However, muon decays produce a large beam-induced background (BIB). This project aims to successfully implement the hybrid 3D pixel sensor digitization model into the Key4Hep-based muon collider software stack. Investigation of hit timing capabilities and detector occupancy in the presence of BIB can improve "noise" rejection in the innermost layers of the detector with 3D sensor geometry.

1 Project Description and Deliverables

A muon collider can combine the strengths of hadron and electron colliders. With a bigger mass, muons emit less synchrotron radiation, so a muon collider could reach higher energies. Furthermore, muons, unlike protons, are fundamental particles that put all their energy into a collision. Thus a muon collider could be much smaller and cheaper, while still reaching multi-TeV collision energies. Muon collisions are cleaner and more likely to produce important particles like the Higgs boson. Moreover, muon collider's sensitivity to weak interactions could help detect weakly interacting massive particles.

The biggest challenge while dealing with muon colliders is the high rate of beam-induced background (BIB) produced from muon decays due to their short lifetime. Therefore, colliders must operate quickly and efficiently before muons decay. At the same time, hits must be precisely detected so the particles from the real collision can be separated from the large multiplicity of BIB particles entering the detector concurrently. Hence we need to try for advanced technologies for high spatial and temporal resolution as well as which have high radiation tolerance in the innermost layers of the detector for good tracking performance of charged particles.

One way for a muon collider to cope with BIB would be to implement 3D sensors into the innermost layers of pixel detectors. In 3D sensors electrodes are deposited in columns, opposed to the usual planar configuration with electrodes deposited on the faces of the sensor. Therefore, 3D geometry reduces charge drift inside the detector and provides better timing precision. In addition, they are also more radiation tolerant due to the lower probability of charge trapping in the silicon defects over time. The goal of this project is to include 3D sensor charge digitization model in the muon collider software stack to study the hit occupancies and timing capabilities for better BIB mitigation techniques. This implementation will simulate timing response and pixel cluster formation in 3D pixel sensors. The main idea is to evaluate the tracker detector performance by the hit occupancy, timing and BIB rejection.

2 Project Timeline

- Weeks 1–4: study muon collider detector concepts, silicon detector digitization process and 3D sensor technology; get familiar with the muon collider software stack; run the whole pipeline from detector simulation to object reconstruction.
- Weeks 5–8: implement 3D sensor charge digitization model into the software; validate the results through hit energy and timing distributions, detector occupancy, and comparison between planar and 3D sensors.
- Weeks 9–12: study track reconstruction performance with new tracker design, analyze pixel cluster shapes for better BIB mitigation techniques; write a report detailing the development process, improvements, and analysis results.

References

- [1] Adrian Cho. The Dream Machine. *Science*, 2024. Available: <https://www.science.org/content/article/muon-collider-could-revolutionize-particle-physics-if-it-can-be-built>.
- [2] The International Muon Collider Collaboration. Supplementary Report to the European Strategy for Particle Physics. 2026. Available: <https://arxiv.org/pdf/2504.21417>.
- [3] Alberto Cervelli. Status and Perspectives of Silicon Detectors. 2023. Available: https://www.researchgate.net/publication/376360792_Status_and_Perspectives_of_Silicon_Detectors.
- [4] Angira Rastogi. Pixel-based BIB Suppression and DAQ Impact at a Multi-TeV Muon Collider. Lepton-Photon 2025 Conference, 2025. Available: <https://indico.cern.ch/event/1493037/timetable/?view=standard#244-pixel-based-bib-suppressio>.

Modifying existing TTrees in Uproot

Applicant: Jokūbas Mačiulis

Mentor: Andres Rios-Tascon

Project duration: June 22, 2026 – September 2026 (10 weeks)

Project Description

ROOT is the standard data storage and analysis framework used in High Energy Physics (HEP), developed at CERN [1]. Data is stored in .root files as TTrees — tree-like structures where branches represent data columns (e.g. particle properties) and entries represent individual collision events. Uproot is a widely-used Python library in the scikit-hep ecosystem that enables reading and writing ROOT files without requiring CERN's ROOT software, making HEP data accessible in modern Python data science workflows [2].

While Uproot supports creating new ROOT files and writing new TTrees, it currently lacks the ability to modify existing TTrees in place. This limitation affects workflows such as backfilling data — for example, adding a new trigger branch with default values to previously recorded data. Uproot does provide `uproot.update` for adding new objects to existing ROOT files [3], however modifying TTrees themselves is not yet supported.

As a previous IRIS-HEP fellow, Zoë Bilodeau took on the challenge of implementing the ability to add new branches (columns) to existing TTrees, resulting in PR #1155 [4]. This contribution brought the feature close to completion, but a few corner cases remain that need to be tested and resolved before the PR can be merged. Adding new entries (rows) to existing TTrees is a separate, entirely new problem with no prior implementation to build on.

Planned Work

The project has two phases:

Phase 1 — Completing TBranch addition (columns): Review and understand Zoë's implementation in PR #1155 [4]. Identify remaining corner cases, write comprehensive tests, debug issues, and work toward merging the PR into the main Uproot codebase.

Phase 2 — Implementing TTree entry addition (rows): Design and implement the ability to append new entries/events to existing TTrees. This involves understanding ROOT's binary basket format, correctly appending data, and updating all relevant metadata. This is entirely new functionality with no prior implementation to build on.

Both phases will include writing tests, documentation, and ensuring compatibility with CERN's ROOT software.

Software Deliverables

- Merged PR #1155 (adding columns to existing TTrees) with full test coverage.
- New implementation for adding rows/entries to existing TTrees.
- Unit tests for all new functionality.
- Documentation and usage examples.

- If time permits: equivalent functionality for RNTuples, ROOT's next-generation data format.

Timeline

Weeks 1–2: Set up development environment, study Uproot codebase and PR #1155.

Weeks 3–4: Debug and test remaining corner cases in PR #1155, prepare for merge.

Weeks 5–6: Design approach for row-addition feature, study ROOT basket format.

Weeks 7–9: Implement and test row-addition functionality.

Week 10: Final testing, documentation, and cleanup. If time permits: equivalent functionality for RNTuples.

References

[1] CERN ROOT framework: <https://root.cern>

[2] Uproot documentation: <https://uproot.readthedocs.io>

[3] uproot.update: <https://uproot.readthedocs.io/en/latest/uproot.writing.writable.update.html>

[4] PR #1155 — Add branches to existing TTrees: <https://github.com/scikit-hep/uproot5/pull/1155>